

Personalized Surgical Risk Assessment Using
Population-based Data Analysis

A DISSERTATION
SUBMITTED TO THE FACULTY OF
UNIVERSITY OF MINNESOTA
BY

Ahmad Mohammad AbuSalah

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Dr. Terrence J. Adam

January 2013

Acknowledgements

I want to express my deeply-felt thanks to my advisor, Dr. Terrence Adam for his encouragement, thoughtful guidance and patience. His mentorship, at the right moments, helped me shape and present this work as it is today. Terry was not only an advisor but also a friend. I also extend my deep gratitude to Dr. Genevieve Melton-Meaux for her assistance and helpful discussions. I also thank the other members of my thesis committee: Dr. Stuart Speedie, Dr. Karen Monsen, and Dr. David Pieczkiewicz for their invaluable input and guidance.

Special thanks to my parents, Mohammad and Samira, for having instilled in me the confidence to face difficult challenges and come out as a winner. They provided me with solace in times of despair and encouraged me to carry on.

My heart-felt thanks to my wonderful wife Sajeda. Her support, quiet patience and unwavering devotion and love were undeniably the bedrock upon which my PhD years have been built. I would never forget the fun and joy my lovely daughters Layan and Raneem gave to me during the period of my study.

Special thanks are due to my brothers, colleagues, and friends who helped me during my graduate life.

Above all, I thank God for providing me with such a great people around me who have contributed generously to my success. I thank Him for His endless bounties (*And if you were to count Allah's (GOD) favors, you would not be able to number them; most surely Allah is Forgiving, Merciful. [Quran 16:18]*)

*To my parents, Mohammad and Samira, my wife, Sajeda, and my daughters,
Layan and Raneem*

Abstract

The volume of information generated by healthcare providers is growing at a relatively high speed. This tremendous growth has created a gap between knowledge and clinical practice that experts say could be narrowed with the proper use of healthcare data to guide clinical decisions and tools that support rapid information availability at the clinical setting.

In this thesis, we utilized population surgical procedure data from the Nationwide Inpatient Sample database, a nationally representative surgical outcome database, to answer the question of how can we use population data to guide the personalized surgical risk assessment process.

Specifically, we provided a risk model development approach to construct a model-driven clinical decision support system utilizing outcome predictive modeling techniques and applied the approach on a spinal fusion surgery which was selected as a use case.

We have also created The Procedure Outcome Evaluation Tool (POET); which is a data-driven system that provides clinicians with a method to access NIS population data and submit ad hoc multi-attribute queries to generate average and personalized data-driven surgical risks.

Both systems use patient demographics and comorbidities, hospital characteristics, and admission information data elements provided by NIS data to inform clinicians about inpatient mortality, length of stay, and discharge disposition status.

Finally, we conducted a subjective evaluation by clinicians to measure their satisfaction with the usability of the POET data-driven system in terms of system usefulness as well as information and interface quality and compared it to the use of the model-driven system.

Our work reinforces the need for the next generation clinical decision support systems that extract knowledge from population data and present it to the clinician at the point of care in a timely fashion.

Table of Contents

List of Tables	vii
List of Figures.....	viii
Chapter 1: Introduction	1
1.1 Research Question and Aim	3
1.2 Significance and Scope.....	5
1.3 Thesis Outline.....	11
Chapter 2: Background.....	12
Chapter 3: Data Source	20
Chapter 4: Model-driven System for Personalized Surgical Risk Assessment	25
4.1 Methods.....	26
4.1.1 Inclusion Criteria	26
4.1.2 Data Preprocessing	26
4.1.3 Predictive Model Construction	28
4.2 Results.....	30
4.3 Model Validation.....	37
4.4 Discussion.....	38
4.5 System Limitations	39
Chapter 5: Data-driven System for Personalized Surgical Risk Assessment	41
5.1 POET's Architecture	42
5.1.1 POET's Presentation Tier.....	47
5.1.2 POET's Logic Tier	48
5.1.3 POET's Data Tier.....	49
5.2 Use Cases	53
5.3 Discussion.....	55
5.4 System Limitations	57

Chapter 6: Providers Evaluation of The Personalized Surgical Risk Assessment Systems.....	59
6.1 Evaluation Model.....	59
6.2 Evaluation Results.....	62
6.3 Discussion.....	65
6.4 Evaluation Limitations.....	69
Chapter 7: Conclusion and Future Directions	70
Bibliography	73
APPENDIX A : Source Code.....	86
A1. POET’s interface, and Modules for Query builder and dispatcher.	86
A2. Module for ICD-9-CM Code Conversion	132
A3. Module to Populate the Personalized Risk Layer and Average-Risk Layer of POET’s Database.....	134

List of Tables

Table 1 Nationwide Inpatient Sample Database.	23
Table 2 The list of independent data variables abstracted from NIS for population-based analysis	27
Table 3 Significant Predictors of inpatient mortality after Spinal Fusion.....	32
Table 4 Significant Predictors of the log transformed length of stay after Spinal Fusion.....	33
Table 5 Significant Predictors of Patient Disposition to a Care Facility versus Disposition to Home after Spinal Fusion Surgery	34
Table 6 Significant Predictors of Patient Disposition to HHC versus Disposition to Home after Spinal Fusion Surgery.....	35
Table 7 Surgical Procedure Outcomes	43
Table 8 The list of NIS data elements covered by the POET System.....	44
Table 9 Summary of POET’s main components and functionalities	47
Table 10 Spinal Surgery Query	54
Table 11 Spinal Surgery with Hemophilia.....	55
Table 12 CSUQ Satisfaction Scores ($N = 7$)	64
Table 13 Challenges of using the POET system	66

List of Figures

Figure 1 Loading NIS data into SAS and Oracle database	24
Figure 2 The closed-loop health information flow chart	38
Figure 3 The Procedure Outcomes Evaluation Tool.....	46
Figure 4 POET's Graphical User Interface.....	48
Figure 5 Data Architecture Layers in POET.....	50

Chapter 1: Introduction

Operative procedures constitute resource-intensive high impact encounters within healthcare. Therefore, operative procedures are usually preceded by an important clinical care process to assess the surgical risks surrounding the procedure. This is usually conducted as part of the patient's preoperative evaluation which contributes to determining the required anesthetic policy and resources needed during and after surgery and facilitates medical management prior to the physiologic and psychological stress of surgical intervention [1, 2,3].

With 45 million procedures occurring each year, there is a large and growing need for pre-procedural evaluations [4]. Such evaluations focus on several key areas including cardiology, pulmonary and hematology as well as medication management to assess risk and manage patient comorbidities [5, 6]. After completing the patient assessment, the consultation results require communicating to surgical and anesthetic providers [7] to address pertinent clinical concerns and to optimize procedural planning [8].

Preoperative evaluations have been shown to reduce preoperative delays, mortality, and address medical problems, especially problems that effect surgical timing and potential surgical cancellations [9-11]. Preoperative assessments facilitate clinical care to optimize medical conditions prior to surgical intervention and can help avoid cancellations, which can have substantial economic impacts [12,13]. In addition, preoperative assessments ultimately may influence the clinician's decision to offer surgery or the patient's decision to pursue a surgical procedure if the patient has high-risk medical co-morbidities [14]. Clinicians

might also advise patients to reschedule their surgery around needed rehabilitation or skilled nursing facility availability that better meets their recovery requirements based on patient's medical conditions [15].

The preoperative evaluation is challenging since it is a multispecialty evaluation with general practitioners, anesthesia, surgery, nursing, and pharmacy professionals providing input into the surgical planning process [8]. Frequently, the primary care provider provides the preoperative evaluation and communicates to the surgical team any recommendations on a patient's cardiac, pulmonary, hematologic and medication management. They also evaluate and give recommendations on pre-existing or new medical problems which may affect surgery. A major challenge of preoperative evaluation is to perform value-based, efficient, and effective preoperative assessments that result in safe patient care and maximum operating room efficiency [16].

There is a need for new tools which can better support preoperative evaluation, provide insight to clinicians about patient risks, facilitate procedural planning, and improve the clinical encounter [17]. Clinical decision support systems are usually accepted as important informatics tools that can enhance clinicians decision-making while performing complex tasks [18]. Almost 20 years ago, clinical decision support systems were defined as "active knowledge systems that use two or more items of patient data to generate case-specific advice" [19]. According to the American Medical Informatics Association, clinical decision support systems are installed at the point of care to provide clinicians, staff, patients, and other individuals with knowledge and person-specific information, filtered and

presented at appropriate times, to enhance health and health care [20]. The Institute of Medicine has long recognized problems with quality of care in the United States, and has advocated using effective health information technology, including clinical decision support systems, to improve quality.[21,22].

The Anesthesia Quality Institute (AQI) mentioned that clinical decision support system could help practitioners avoid unnecessary treatment in low-risk patients or, conversely, choose a different treatment modality for high-risk individuals [23]. These systems do not replace the clinician's judgment, but rather are used as knowledge extraction tools to provide a supplementary decision aid to promote consistency and improve accuracy at the clinical setting [24].

1.1 Research Question and Aim

The main goal of this thesis is to use healthcare population data about surgical outcomes to support clinical decisions for preoperative evaluation. Specifically, the research question is: can we use the Nationwide Inpatient Sample (NIS) data [25], a nationally representative population database on surgical outcomes, to support the process of personalized surgical risk assessment process [26].

To answer this question, we created, evaluated, and compared the usability of different architectures of population-based clinical decision support systems for the support of personalized surgical risk assessment. Initially, we started with the development of a data-driven clinical decision support system, which provides users the ability to access and compute data-driven surgical risks to support decision making, however data-driven systems are expensive to develop and implement in organizations and requires large resource commitments compared to

model-drive systems [27]. Therefore, our focus shifted towards the construction of a population-based model-driven system, which uses statistical modeling techniques such as predictive modeling to create clinical decisions support systems with lower infrastructure cost and development time compared to data-driven systems. The construction of a model-driven system would also provide us with in-depth insight on the quality of NIS data to be utilized for personalized surgical risk assessment. We analyzed NIS data elements and utilized them within the context of spinal surgery which was selected as a use case. We have considered the experience we gained during the construction of model-driven system as important preliminary knowledge to inform the design and development of a data-driven system for personalized surgical risk assessment where the underlying data-tier forms the de facto key component. The created model-driven and data-driven systems will inform clinicians about three surgical procedure outcomes: inpatient mortality, hospital length of stay, and the anticipated discharge disposition status of the patient. These three outcomes have been frequently used to perform patient preoperative evaluation and surgical risk assessment [15, 28 – 32].

The three specific aims of this thesis are:

1. Utilize outcome predictive modeling to construct a population-based model-driven clinical decision support system for personalized surgical risk assessment optimally tailored to patients and hospital sites for a high volume surgical procedure. The model construction approach provides direction for collecting and analyzing NIS data for surgical risk

assessment and sheds light on potential data issues and challenges associated with the development of a data-driven system.

2. Create a population-based data-driven clinical decision support system aims at producing personalized surgical risk assessments to clinicians in a timely fashion using NIS data and addresses challenges identified with model-driven systems. Describe the system architecture and key components in addition to use cases where the system can be useful.
3. Measure clinician's satisfaction with the usability of the data-driven system in terms of system usefulness as well as information and interface quality. Provide a comparison of clinician perceptions on the usability of the data-driven and model-driven systems.

The completion of these aims provides a combination of prototype population-based clinical decision support systems for surgical risk assessment.

This work was exempt from the Institutional Review Board (IRB) approval since the data utilized were sufficiently de-identified (study number: 1104E97854).

1.2 Significance and Scope

Surgical procedure risk assessment is an essential step of the patient's preoperative evaluation [26]. It is crucial that this work incorporates all information required to provide an assessment that is tailored to the conditions surrounding a procedure including patients, hospitals, and other characteristics [28]. The Institute of Medicine defines quality as 'the degree to which health services for individuals and populations increase the likelihood of desired health outcomes and are consistent with current professional knowledge' [33].

Consequently, desired health outcomes refer to health characteristics sought at the personalized level. The achievement of a specified outcome is regarded as evidence that ‘good’ care was delivered [34].

Personalized risk assessment is gaining attention from research groups around the world because of its potential to improve diagnosis and optimize clinical treatment by providing insights on patient-specific outcomes of therapies and surgical interventions [35]. Patients have multiple characteristics that can affect the likelihood of the outcome and the effect of therapy; therefore it is recommended that such parameters are considered during a patient-specific risk assessment process [36]. This approach addresses the “individual knowledge” management conundrum pointed out by Pantazi [37]. What patients are really after is “For me in particular, what are the expected outcomes of this surgical intervention or therapy?” [38].

Personalized surgical risk assessment is challenging since physicians have general familiarity on the expected outcomes of frequently encountered surgical procedures, but their knowledge of rare clinical cases, or the impact of unusual comorbidities or patient-specific parameters on surgical outcomes are limited [39]. Studies have demonstrated that surgical outcomes of patients with more extensive comorbid conditions undergoing more complex surgical procedures may be different than patients with less severe conditions or simpler procedures [40].

Physicians need to maintain their clinical knowledge base to maintain the standards of clinical practice, meet patient expectations, and communicate with their colleagues [41]. When the unfamiliar is encountered, physicians rely on their clinical training and experience and refer to clinical guidelines and trials on which to make decisions in circumstances of uncertainty [41]. Clinical guidelines are generally designed after a lengthy and thorough process, involving clinical topic selection, consultation with stakeholders, expert panel review, the generation of draft recommendations, submission to stakeholders or peer-review, and finally, issuance of final guidelines [42]. However, this source of information may fail to provide definitive answers since they typically only provide rough estimates of outcomes for a particular treatment [43]. There are many cases where the quality and availability of clinical trials and guidelines data is insufficient to guide definitive risk assessment [16]. Moreover, these results might not apply directly to individual patients [44] because they are based on averages. Furthermore, the overgeneralization of clinical trials findings to populations excluded from the clinical trial has been a serious concerns among clinicians [38].

Physicians treat individuals and not population means, creating concerns on whether the results of randomized clinical trials really apply Individual patients, even those within the trial [44]. The reported effects of a treatment or the clinical factors associated with the outcomes of a surgical intervention reflect the average of the study participant and do not necessarily apply to all individual patients within the population of the clinical trial or to any selected sub-group [38]. Multiple and varied examples have demonstrated that large subgroups of a

clinical trial population have results that differ substantially from the overall summary [45-47]. Besides, accurate estimates of clinical risk can be difficult to estimate for patients in whom their comorbidities and clinical risk are poorly categorized in clinical guidelines potentially making overall population guidelines inappropriate for individuals [48].

As a supportive source of information, physicians also seek guidance for their clinical questions by consulting colleagues [41]. In these cases, specialty care providers can be consulted to help inform the clinical decision making and better understand patient needs. Multi-specialty high risk panels can also be used to provide medicine, anesthesia, and surgical feedback to the patient and yield interdisciplinary recommendations, but these are time consuming solutions require significant provider input and clinical site coordination and are based on limited patient-specific published data for a particular patient or procedure [14]. Clinicians would prefer solutions that can rapidly answer their questions [49].

Recently, population clinical data showed the potential to be an additional source of information that is useful and acceptable by clinicians to answer clinical enquiries [50-52]. The work of John Wennberg has provided insights on provider selection patterns among alternative treatments and has resulted in a growing interest in evidence-based clinical decision making process that utilize healthcare data to assess clinical outcomes [53, 54]. The knowledge gained from analyzing nationally representative clinical data can greatly benefit care delivery and outcome [55]. Population-based data sets have high degrees of generalizability and can be used as tools to examine and identify quality of care concerns and

enhance practice guideline development [56, 57]. Several studies have shown that a substantial number of primary care physicians' questions could be answered by consulting electronic clinical databases [58, 59, 41].

Nowadays, there are several electronic national clinical databases which provide a rich clinical digital content for clinical research. NSQIP (National Surgical Quality Improvement Program) [60], VASQIP (Veterans Affairs National Surgical Quality Improvement Program) [61], and NIS data [25] are examples of large outcome registries used nationally to support various clinical decisions including surgical risk assessment. For example, NSQIP has been used to identify the preoperative factors that predict morbidity and mortality for distal pancreatectomy [50]. VASQIP has been used to assess the attributable risk and potential benefit of smoking cessation on surgical outcomes [51] and to study the postoperative complications in patients undergoing non-cardiac surgery [52]. We have utilized NIS data to support spinal fusion risk assessment and inpatient surgical planning [62].

Despite the recent increase in clinical digital content, we find that this source of information is still underutilized by physicians in the clinical setting while seeking answers to their clinical questions [41]. Traditional methods of face-to-face communication and use of hard-copy evidence still prevail amongst qualified medical staff in the clinical setting to address various clinical information needs and surgical risk assessment is not an exception [41, 39,63]. This is due to the lack of information content processing tools that facilitate knowledge extraction from these large repositories for direct clinical decision support [63-66]. Such

tools can provide a value-added assessment at the point of care which can 1) allow the informational query to potentially instruct clinicians and researchers on broader outcomes often excluded from randomized control trial published data and 2) generate information for providers, patients, families, and facilities for optimal planning surrounding a patient's surgical procedure in a more individualized way as compared to data derived from large clinical trials and guideline assessment. The use of new technologies embracing the new digital age in information provision may influence this in the future [39, 63]. However, for now, there is a need for more research to be undertaken to uncover the most effective methods of encouraging clinicians to use the best evidence in everyday practice [39].

The disparity between clinical evidence and practice demonstrates a critical need for preoperative clinical decision support systems that provides patient-specific surgical risk assessment [16]. There are insufficient resources to utilize existing surgical outcomes data for the support of patient-specific surgical risk assessment process despite the in-depth insight gained through the preoperative evaluation in understanding high acuity, cost, and potentially serious consequences of many operative procedures [1]. These resources could potentially 1) provide patient data mimicking clinical studies on a desktop computer for exploratory analysis and 2) create an estimate of surgical risk optimally tailored to the individual patient for every-day practice. Such work has been historically challenging due to the lack of available clinical population data and insufficient computing power to complete such analysis in the context of patient-specific

surgical risk assessment [67]. Nowadays, with the availability of present-day digital content of data on surgical outcomes, such work becomes more feasible if proper tools are used [58, 59, 41].

1.3 Thesis Outline

Chapter 2 of this thesis provides a background review of clinical decision support systems.

Chapter 3 describes NIS discharge dataset to be used with the development of the model-driven and data-driven systems.

In chapter 4, we constructed three model-driven systems that can be used to create personalized predictions of inpatient mortality, length of stay, and disposition status using outcome predictive modeling techniques. The model-driven systems were created for spinal fusion surgery which was selected as a clinical use case.

Chapter 5 describes the architecture of the Procedure Outcomes Evaluation Tool (POET); a data-driven clinical decision support system aimed at producing personalized surgical risk assessments for clinicians in a timely fashion using NIS data.

In chapter 6, we utilize the four-phase model for evaluating clinical decision support architectures described by Wright and Sitting [66] and the IBM Computer Systems Usability Questionnaire (CSUQ) instrument [68] to measure POET's utility. We also discuss clinician's perceptions on the usability of data-driven and model-driven systems at the point of care for personalized surgical risk assessment.

Finally, chapter 7 concludes the thesis and highlights potential future directions.

Chapter 2: Background

Some common applications of clinical decision support systems are: diagnostic decision making and identifying medication errors [69]. Diagnostic decision making systems assess the risk of developing some diseases such as diabetes and heart disease. For example, the national cholesterol education program's online heart disease risk assessment tool estimates risk based on patient age, gender, and smoking status [70]. Another example is the similarity-based disease risk assessment tool which supports personalized medicine by analyzing the similarity between a personal genome and a disease [71]. Data elements from patient profiles have been used by the PULSE tool (Personalization Using Linkages of SCORE and behavior change readiness to web-based Education) for personalized cardiovascular risk assessment [72]. Also, the InfoBot system supports the development of individualized patient care plans based on patient's medical conditions which might eventually improve the quality of care [73]. The Archimedes system in turn provides evidence-based decision making using the Archimedes Model; which is a disease-based proprietary model that has been extensively validated and has already found multiple applications in healthcare policy and research [74, 75]. It integrates clinical guidelines rules with data from several clinical trials, and epidemiological studies to generate patient-specific decisions for diagnostic decision making for a set of diseases including diabetes and cancer.

Clinical decision support systems for identifying medication errors are generally supported with some type of alerts or reminders that would identify potential

medication error or an adverse drug event [76]. Reducing medication errors in intensive care units (ICUs) not only improved clinical care but also led to considerable financial savings [77]. They can be also used to identify and prevent potential drug interaction related problems such as in the SFNIX system [78, 79]. The ATHENA system developed in collaboration between Stanford University and the U.S. Department of Veterans Affairs (VA) is used to increase guideline-adherent prescribing. It gives patient-specific recommendations to primary care providers at the point of care based on data in patients' computerized medical record and knowledge of the clinical guidelines [80].

Modern-day clinical decision support systems are a combination of hardware and software components designed to facilitate the automatic recording, storage, and retrieval of perioperative data for surgical patients [81].

The Agency for Healthcare Research and Quality has listed the most common features of clinical decision support systems that are designed to provide patient-specific guidance to include: 1) the knowledge base; which holds the clinical information about diagnoses, drug interactions, procedures, etc., 2) a program for combining that knowledge with patient-specific information, and 3) a mechanism of entering patient data (or importing it from the EMR) into the clinical decision support engine and providing relevant output (e.g., possible diagnoses, drug interaction alerts, risk measures, or preventive care reminders) back to the clinician [76].

Several detailed taxonomies have been provided of clinical decision support systems' functions and features [78, 82, 83]. There are five major types of clinical decision support systems:

- 1) Communication-driven systems emphasizing communication technology among groups to share information and support collaboration as in the use of telemedicine to guide clinical decisions.
- 2) Data-driven systems providing users the means to access and analyze an underlying powerful data-tier to support decision making. With the growing availability of clinical digital content and an increasing dominance of Electronic Medical Record (EMR) systems and personalized medicine, it is suggested that this type of system will have a great potential in supporting clinical decisions [84, 85].
- 3) Document-driven systems using unstructured data to support decisions like the SimulConsult system (<http://www.SimulConsult.com>) which extracts knowledge from peer-reviewed medical literature to answer clinician's enquires.
- 4) Knowledge-driven systems incorporating rule-based systems which format the knowledge of domain experts into rules that are tested and then utilized in the clinical decision support process. Oncocin is an example of rule-based CDSS designed to assist physicians with the treatment of cancer patients receiving chemotherapy [86].
- 5) Model-driven decision support systems emphasizing the use of statistical modeling techniques to create clinical decisions support systems such as

outcome predictive modeling techniques [87]. Iliad is an expert diagnostic system designed by the University of Utah and uses statistical modeling to calculate the posterior probabilities of various diagnoses under consideration, given the findings present in a case [88].

Recent reviews demonstrate a growing use of data-driven clinical decision support systems that provide real-time access to larger and better integrated databases [89]. For example, the Informatics for Integrating Biology and the Bedside tool (I2B2) can be used to rapidly identify a cohort of patients for subsequent analysis from large clinical datasets [90]. The Shared Health Research Information Network (SHRINE) systems are used to identify cohorts and perform preparatory discovery [91]. Finally, data-driven systems can aggregate single-institution registries to drive individualized risk estimates for patients requiring care at the time of the surgical intervention [67, 57], such as the hepatic surgery procedure risk assessment [92].

From an architectural perspective, there are four general classes of decision support architectures [93]:

- 1) Standalone systems: This type of system operates separately from other systems requiring clinicians to enter pertinent information that will help the system produce relevant findings.
- 2) Integrated systems: The clinical decision support functionality in this architecture is integrated with other clinical systems, such as EMR systems. It can be implemented as a module in another system or

receives/routes data from/to the other system through health information exchange protocols.

- 3) Standardized systems: This architecture focuses on standardizing computerized representation of clinical knowledge so that it can be shared with other clinical systems.
- 4) Service-oriented systems: The Service Oriented Architecture (SOA) presents the clinical system and the decision support systems as two different services that are separated but can be integrated if needed using service-based interfaces.

Those architectures can be embedded into a local EMR system, installed at a local personal computers or handheld device, or implemented as an internet-based system [76]. The literature reported improvements in the quality and cost of care from the use of EMR systems with proper implementation and use of embedded clinical decision support systems have been noted previously [94-96].

From the data source perspective, the clinical literature reported several systems that utilize clinical practice guidelines, clinical trial data, or healthcare data at the single-institution level or population level to support clinical decisions.

OSCAR (Optimizing Surgical Care and Assessment Record) clinical decision support system predicts surgical risks prior to surgery based on the National Institute for Health and Clinical Excellence (NICE) guidelines [24]. STAT Cardiac Clearance (STAT CC) is another guideline-based system that exclusively performs cardiac preoperative assessment using the American College of Cardiology/American Heart Association (ACC/AHA) guidelines [97].

The challenge in guideline-based systems is of how to apply clinical guidelines effectively on the ground and integrate them in clinical workflows and work practices [24, 42]. Such systems have some inherent limitations since recommendations issued by the systems can only be as good as the guidelines they model, and as a result, flaws in the guidelines will be systematically reproduced in the output of the system [78, 98].

Healthcare data at the single-institution and population levels have been also utilized to guide preoperative clinical decisions. Hybrid systems such as Picis PreOptimize [99] utilize preoperative guidelines and hospital-specific patient databases to generate reports containing patient demographic information and patient-specific preoperative evaluation. Picis also enables clinicians to query the hospital-specific patient database to assess preoperative quality measures including surgical risks as mortality and length of stay [99]. Similarly, Bouamrane et. al. used clinical guidelines and single-institution healthcare data to predict surgical outcomes in their clinical decision support system [42].

Healthcare data have been heavily used by model-driven systems to predict surgical risks [100], such as in surgical risk evaluation of colorectal cancer surgery [102] and spinal surgery [101]. Such systems analyze population healthcare data to study the impact of various factors on surgical outcomes [103-105]. Outcome predictive modeling techniques, used to construct model-driven systems, are usually sufficiently powered to identify important patient-specific outcomes by stratifying patients according to relevant clinical features [106]. Healthcare data have also been utilized in data-driven systems such as the

Population Health Information System (POPULIS) which has been employed in Canada to assess healthcare metrics for policymakers, clinicians, and patients using administrative data routinely collected as part of the national health insurance plan [107]. Data driven systems were also utilized to develop an intensive care clinical information system that provides patient-specific preoperative assessment [108]. “PatientsLikeMe” is another data-driven system that is based on patient’s experience. The system provides a means to query and analyze structured and unstructured data provided by the patients themselves to generate patient-specific recommendations for patients with similar medical conditions [109].

Data-driven systems that correlate readily available population data with outcomes such as mortality and length of stay are less common [110] and contributes to the underutilization of population data by clinicians at the point of care [63-66]. This may be due to uncertainty about the clinical value of such systems and therefore affect the return on investment as well as cost of acquisition, implementation, and maintenance of data-driven preoperative systems [110,111]. The usability of such systems may also have been negatively impacted by data source limitations. For example, the data source scope may cover specific geographic regions or specific medical conditions but not include important covariates that affect the outcomes of interest because they are unmeasured or unreported [112].

There is a need of every-day practice of evidence-based medicine which aims to apply the best evidence extracted from clinical data for clinical decisions making such as the case of using clinical decision support systems for patient-specific surgical risk assessment [113, 114]. However, there are several barriers that hinder the widespread adoption of decision support systems in the clinical setting such as financial barriers, complexities of system installation, and a perceived lack of proven benefits [115]. Timely access to appropriate information has been reported as a major challenge [64, 65]. Other challenges include: search question formulation, optimal search strategy development, and evidence interpretation [41]. Physicians also report problems understanding the output of clinical decision support systems because the output is not well presented whether portrayed numerically, graphically or pictorially [66].

Chapter 3: Data Source

The work of this thesis utilized clinical population data from the Nationwide Inpatient Sample (NIS); which is part of the Healthcare Cost and Utilization Project (HCUP), sponsored by the Agency for Healthcare Research and Quality (AHRQ). NIS database is the largest all-payer inpatient care database that is publicly available in the United States including persons covered by Medicare, Medicaid, private insurance, and the uninsured. It contains data from 5 to 8 million hospital stays from about 1,000 hospitals sampled to approximate a 20-percent stratified sample of U.S. community hospitals.

NIS data are released on an annual basis and include clinical and resource use information typically available from discharge abstracts. The data are stored in three ASCII files that can be loaded into SAS and Oracle database with some SAS, Java, and SQL programming:

- *Core data file (record = inpatient stay)*: An inpatient discharge-level file containing all the discharge data from the sample of hospitals in NIS. Pertinent data elements include patient demographics, procedure codes, and admission information.
- *Comorbidities file (record = inpatient stay)*: A discharge-level table containing information on patient comorbidities. These measures identify coexisting medical conditions which are not directly related to the main reason for admission, and are likely to have originated prior to the hospital stay.

- *Hospital Information file (record = hospital)*: A hospital-level table containing one record for each hospital included in the NIS database. The table describes hospital characteristics including location, region, bed-size, and teaching status.

NIS data uses the AHRQ comorbidity software [116] to populate a comprehensive set of 29 comorbidity measures reported by Elixhauser et.al. for use with large administrative inpatient datasets [117]. Comorbidity measures are modeled in NIS database as binary variables that have the value 1 (comorbidity exists) or 0 (comorbidity does not exist).

The AHRQ uses the NIS data, among many other data sources, to help construct two of its most popular products: the National Healthcare Quality Report [118] and the National Healthcare Disparities Report [119]. These two reports, published annually since 2003, together give a clear picture of the state of the nation's healthcare quality by tracking quality measures for the overall healthcare system and summarize differences in quality of and access to health care experienced by various racial and ethnic groups as well as other priority populations. [100].

NIS population data has been used to generate population-based estimates of surgical procedure outcomes using various predictors which can expose opportunities for quality improvement. For example, it has been used to estimate the inpatient mortality after a Pancreatic Resection procedure [120] or a Diaphragmatic Hernia Repair procedure [121]. It has been also used to assess the

inpatient-mortality, length of stay, and discharge disposition of weekend stroke admission compared with weekday stroke admission [122].

Model-driven systems have frequently utilized NIS data for clinical decision making, however, according to AHRQ, NIS data as well as other data from the HCUP project do not yet communicate directly to a critically important stakeholder cohort as providers and patients that need to make patient-specific decisions based on these data [100]. The AHRQ recommends that proper tools, that can properly query these data, are created to help consumers using the data to guide data-driven decisions [100].

Despite the fact that most of NIS-based studies have a retrospective nature; which has significant potential for selection bias because the outcomes and exposures are already known [123], it is believed that selection bias is minimized if the entire population is selected by the study [124]. Nevertheless, population-based studies may be affected by study design limitations and NIS data limitations [122]. One possible limitation of many NIS based studies is their focus on evaluating clinical outcomes from the perspective of research selected factors which may miss other factors contributing to an outcome [125]. Another limitation is caused by the way missing data are being handled which can cause bias and affect the overall results [126]. Also, the results may be impacted by coding error; in a large database such as the NIS, there is significant potential for coding error or variability in coding which may lead to misleading results as in the case of clinical outcomes of carotid endarterectomy and carotid stenting [125]. NIS data may not provide the right data-level granularity for some research

studies such as comparing clinical outcomes of weekend admission and Saturday night admission [122]. NIS data are de-identified data and do not provide any mean to link same-patient admissions which may limits its use to model hospital readmissions and generate 30-day risk estimates.

In this thesis, we used 5 years of NIS data for the years 2004 to 2008. Table (1) presents the number of discharges per year in NIS data that we are using to develop the model-driven and data-drive clinical decision support system for personalized surgical risk assessment.

Table 1 Nationwide Inpatient Sample Database.

Year	Number of States	NIS Sample hospitals	Hospital Universe	NIS Sample discharges	Discharge Universe
2004	37	1,004	4,906	8,004,571	39,065,000
2005	37	1,054	5,146	7,995,048	39,163,000
2006	38	1,045	5,124	8,074,825	39,450,216
2007	40	1,044	5,099	8,043,415	39,541,948
2008	42	1,056	5,162	8,158,381	39,885,120

We used SAS programming to load the data into SAS for the construction of the model-driven system and used PL/SQL and Java scripts to load the data into Oracle database for the development of the data-driven system as shown in Figure (1). This process is also reffered to as ETL where we extract data from source data file, transform the data to match the target database and finally load it into the database.

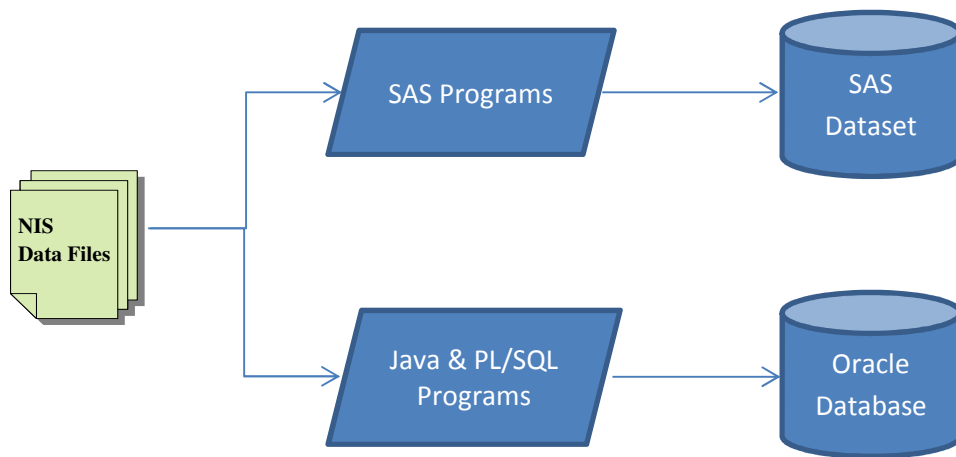


Figure 1 Loading NIS data into SAS and Oracle database

Chapter 4: Model-driven System for Personalized Surgical Risk Assessment¹

This chapter describes the design of population-based model-driven clinical decision support system utilizing outcome predictive modeling techniques for the support of personalized surgical risk assessment to achieve the aims stated in section 1.1 of Chapter 1. The system construction approach serves as a direction for collecting and analyzing NIS data for the support of surgical risk assessment. In addition, we want to make the analytic model results available to clinicians and researchers for preoperative surgical risk assessment, hospital resource allocation, and hypothesis generation for future research without an individual patient data management burden. Finally, the construction of model-driven system will inform the design of the data-driven system presented in Chapter 5 of this thesis.

Specifically, we utilized population-based data analysis to construct surgical outcome predictive models using hospital, patient and admission characteristics to identify data elements affecting inpatient mortality, length of stay, and disposition status. We applied the risk model development approach on patients receiving spinal fusion surgery in the years 2004-2008. Spinal fusion was the selected prototype procedure due to it being a high volume and typically inpatient procedure where patient risk factors will likely affect clinical outcomes. Spinal surgeries also present substantial clinical management challenges and represent a rapidly growing and high volume procedure which is typically completed in the hospital with potential for substantial surgical complications [127]. Despite this

¹ Ahmad AbuSalah, Genevieve Melton-Meaux, Terrence Adam, "Patient-Specific Surgical Outcomes Assessment Using Population-Based Data Analysis for Risk Model Development," AMIA Annu Symp Proc 2012, in press.

trend, there is a paucity of literature addressing perioperative outcomes, particularly for specific patient comorbidities [128]. Spinal surgery has high financial costs and is often associated with increased bleeding and severe postoperative pain [129]. It is also one of the most invasive orthopedic surgeries performed on children [130]. Given the significance of the procedure, this clinical area was chosen as a clinical prototype for this study.

4.1 Methods

4.1.1 Inclusion Criteria

A total of 356,857 discharges were identified from the NIS for patients that underwent primary fusion spinal surgical procedures with International Classification of Diseases, Ninth Revision, Clinical Modification (ICD-9-CM) codes :81.00 - 81.09, 81.30- 81.39, 81.62 – 81.64, 84.51, 84.52, 84.62 - 84.64 in the U.S. from 2004 through 2008. The ICD-9-CM codes were identified after consultation with medicine and surgical procedure experts and through a review of surgical outcome literature [131, 132]. Since this dataset spans over several years, we had to consider any updates and modifications to the ICD-9-CM codes [133].

4.1.2 Data Preprocessing

The quality of NIS data, in terms of missing data values, was the main challenge we faced after loading NIS data into SAS prior to the construction of the model-driven clinical decision support system for personalized surgical risk assessment. Modern methods for handling missing data, including maximum likelihood parameter estimation [134] and multiple imputation [135] methods, enable researchers to derive appropriate parameter estimates and inference from

incomplete data. For a variety of reasons, multiple imputation method is not always practical, particularly with very large datasets [134, 136]. The maximum likelihood parameter estimation method is more efficient than multiple imputation methods with large datasets [136]. Besides, for a given set of data, maximum likelihood always produces the same result. The multiple imputation method on contrary gives a different result every time you use it [137]. Therefore, for the studied data elements, we used an expectation-maximization (EM), PROC EM in SAS, which provides maximum likelihood estimates to handle missing values [138]. We excluded 6% of NIS data elements that have more than 10% of missing values since this may impact the estimations of the PROC EM in SAS [139].

Table 2 The list of independent data variables abstracted from NIS for population-based analysis

<p>Patient Characteristics: <i>Demographics:</i> Age, Sex (male, female), Race (white, black, Hispanic, others) <i>Number of co-morbidities:</i> 0, 1, 2, ≥ 3 <i>Co-morbidities:</i> Acquired Immune Deficiency Syndrome (AIDS), Alcohol Abuse, Deficiency Anemia, Rheumatoid Arthritis, Blood Loss Anemia, Congestive Heart Failure, Pulmonary Disease, Coagulopathy, Depression, Diabetes, Drug Abuse, Hypertension, Hypothyroidism, Liver Disease, Lymphoma, Fluid and Electrolyte Disorders, Metastatic Cancer, Neurological Disorders, Obesity, Paralysis, Peripheral Vascular Disorders, Psychoses, Pulmonary Circulation Disorders, Renal Failure, Solid Tumor Without Metastasis, Valvular Heart Disease, Weight Loss.</p>
<p>Hospital Characteristics: <i>Annual Volume:</i> Very low, Low, Medium, High, Very High <i>Region:</i> Northeast, Midwest, South, West <i>Location:</i> Rural, Urban <i>Bed-size:</i> Small, Medium, Large <i>Teaching Status:</i> Yes, No</p>
<p>Admission characteristics: <i>Weekend Admission:</i> Yes, No <i>Elective Admission:</i> Yes, No <i>Discharge Year:</i> 2004, 2005, 2006, 2007, 2008</p>

Table (2) lists the abstracted NIS data elements in addition to few more computed data elements that previous studies used to predict spinal fusion

surgical outcomes. For example, we calculated the number of comorbidities per admission as it might influence inpatient mortality [140]. We also calculated the annual hospital volume based on prior studies suggesting its usefulness in predicting surgical outcomes [141, 142]. Hospital volume was specifically determined using the unique hospital identifier for each institution provided by NIS. We categorized the annual hospital volume into very low (<40 cases), low (40 to 99 cases), medium (100 to 199 cases), high (200 to 399), and very high volumes (>400) [141]. A number of other binary meta-variables and derived variables were also included to facilitate the analysis step.

4.1.3 Predictive Model Construction

Outcome predictive modeling is a process by which statistical or probabilistic models are created to predict the likelihood of future outcomes based on the knowledge extracted from historical data [143]. Generally, statistical or probabilistic models create access to underlying clinical data patterns which do not exactly reflect the underlying patient data thereby masking patient identities to address data agreements and HIPAA rules. Such modeling approaches allow us to construct a model-driven clinical decision support system at a cheaper cost in terms of computer infrastructure and development time compared to data-driven systems which are often very expensive to develop and implement in organizations and requires large resource commitments compared to model-drive system [27].

All data analyses were performed using SAS software (version 9.2; SAS Institute, Inc, Cary, NC). Descriptive statistics for all variables were conducted. One predictive model was created per assessed outcome. The inclusion method to

incorporate variables into the predictive model is carried out in a forward stepwise manner, testing for the statistical significance with variable inclusion at each stage requiring p-value to enter of 0.05. Finally, these models were validated internally and validation measures reported. The literature shows that internally validated clinical decision support systems are beneficial when deployed clinically [144]. For the validation step, we used the SAS PROC SURVEY method to randomly divide the set of identified admissions into two mutually exclusive datasets: the training set (80% of admissions, n = 285,485) and the validation set (20% of admissions, n = 71,372). Both sets were generated once and used to create and validate the three models.

Inpatient Mortality Predictive Model

We used logistic regression (SAS procedure PROC LOGISTIC) to predict inpatient mortality (i.e. the probability of patient death during the hospitalization where the surgical procedure was performed). The training set used to create the model has 693 events (i.e. deaths) which is large enough to meet the recommended value for minimum Events per Variable (EPV) to maintain our statistical model [145, 146].

Length of Stay Predictive Model

We used multivariate linear regression modeling (SAS procedure PROC REG) to estimate the length of stay for patients admitted for spinal fusion surgery. The length of stay is a continuous variable measured in days with NIS providing a calculated patient length of stay per admission by subtracting the admission date

from the discharge date. The exploratory analysis prior to the modeling step indicated that patient length of stay is not normally distributed. It follows a distribution that is skewed to the right (mean=3.9, median=3). Thus, it was transformed using its natural logarithm to provide an appropriately normal distribution for modeling as indicated by a quantile-quantile plot.

Disposition Status Predictive Model

We use SAS procedure PROC LOGISTIC with option link=glogit to fit a multinomial logistic regression model to identify the variables associated with disposition to nursing facility or Home Health Care (HHC) using discharge to home (routine) as the reference level. The NIS database also includes patients who were discharged to short-term hospital or discharged Against Medical Advice (AMA). We excluded admissions pertaining to those two types as they represent <0.6% of the training set used to create the disposition status predictive model.

4.2 Results

The inpatient mortality rate for the entire cohort was 0.3%. The mean and median patient length of stay were 3.9 and 3 days respectively, 77.4% of patients were discharged to home, 12.5% discharged to another care facility, and 9.5% were discharged to Home Health Care. Age followed a normal distribution (Shapiro-Wilk test, $p < 0.01$, mean=52 y, StdDev =15 y). Fifty-four percent of patients were women and 60.5% were white. About 34.3% of patients had no co-morbidities. The most frequent co-morbidity among the patients was hypertension (40%). The majority of patients had elective admissions (87.3%) and

been treated at hospitals with large bed-size (65.3%). Patients also tend to choose teaching hospitals (55.1%) for spinal fusion.

The significant predictors ($\alpha = 0.05$) and their association with the inpatient mortality and the log transformation of length of stay are presented in Tables (3 and 4). Tables (5 and 6) in turn presents the significant results ($\alpha = 0.05$) of the multinomial logistic regression for patient disposition after primary spinal fusion procedure. All predictors are binary variables that have the values (1= Yes, 0= No). The reference value is 0 for all.

Table 3 Significant Predictors of inpatient mortality after Spinal Fusion

Significant Predictors*	Training	Validation	Inpatient Mortality***	
	n=285,485 (80%)	n=71,372 (20%)	Estimate **	OR [95% CI]
Age			0.530	1.07 [1.05, 1.08]
Sex (<i>Female</i>)	152,907	40,661	-0.617	0.54 [0.47, 0.62]
Teaching Hospital (<i>Yes</i>)	157,822	38,906	0.575	1.78 [1.52, 2.08]
Weekend Admission (<i>Yes</i>)	8,175	992	0.572	1.77 [1.45, 2.17]
Elective Admission (<i>Yes</i>)	245,870	65,034	-1.321	0.27 [0.23, 0.31]
Race				
<i>White</i>	171,661	44,352	-0.204	0.82 [0.71, 0.94]
<i>Black</i>	16,168	3,515	Not Significant	
Patient Co-morbidities				
<i>AIDS</i>	152	34	1.883	6.58 [2.33, 18.57]
<i>Congestive Heart Failure</i>	4,329	1,246	1.110	3.04 [2.42, 3.80]
<i>Pulmonary Disease</i>	38,968	10,427	0.404	1.50 [1.24, 1.81]
<i>Coagulopathy</i>	3,307	851	1.236	3.44 [2.69, 4.41]
<i>Depression</i>	30,232	8,430	-0.484	0.62 [0.44, 0.86]
<i>Hypertension</i>	111,360	32,295	-0.641	0.53 [0.44, 0.63]
<i>Fluid and Electrolyte Disorders</i>	15,574	4,562	1.082	2.95 [2.46, 3.55]
<i>Metastatic Cancer</i>	1,720	153	1.493	4.45 [3.40, 5.83]
<i>Neurological Disorders</i>	8,870	2,170	0.853	2.35 [1.83, 3.01]
<i>Paralysis</i>	6,523	1,088	1.067	2.91 [2.34, 3.61]
<i>Peripheral Vascular Disorders</i>	3,892	1,246	0.380	1.46 [1.04, 2.06]
<i>Pulmonary Circulation Disorders</i>	865	211	1.684	5.39 [3.76, 7.73]
<i>Renal Failure</i>	3,169	880	1.302	3.68 [2.83, 4.77]
<i>Solid Tumor Without Metastasis</i>	1,256	218	0.911	2.49 [1.66, 3.74]
<i>Weight Loss</i>	1,565	252	1.429	4.18 [3.24, 5.39]
Number of co-morbidities				
0	100,960	21,259	-1.382	0.25 [0.18, 0.35]
1	83,152	20,751	-0.417	0.66 [0.53, 0.82]
2	55,257	15,550	Not Significant	
Hospital Annual Volume				
<i>Very low</i>	9,120	2,025	Not Significant	
<i>Very High</i>	116,075	31,022	-0.243	0.78 [0.67, 0.91]
Hospital Region				
<i>Midwest</i>	68,491	15,830	-0.354	0.70 [0.59, 0.84]
Hospital Bed Size				
<i>Small</i>	38,583	9,487	-0.356	0.70 [0.51, 0.90]
<i>Large</i>	186,418	46,668	0.281	1.32 [1.10, 1.56]
Discharge Year				
2007	57,443	15,577	-0.353	0.70 [0.58, 0.85]
2008	70,307	14,480	-0.278	0.76 [0.64, 0.90]

OR: Odds Ratio, CI: Confidence Interval,

* All predictors are binary variables that have the values (1= Yes, 0= No). The reference value is 0.

** The logit estimates of the Logistic regression,

*** We model the probability of the occurrence of inpatient mortality assuming the predictor exists.

**** Parameter estimate of linear regression

Table 4 Significant Predictors of the log transformed length of stay after Spinal Fusion

Significant Predictors*	Training	Validation	Log-LOS	
	n=285,485 (80%)	n=71,372 (20%)	Estimate ****	[95% CI]
Age			0.022	[0.02, 0.03]
Sex (<i>Female</i>)	152,907	40,661	0.055	[0.05, 0.06]
Teaching Hospital (<i>Yes</i>)	157,822	38,906	0.165	[0.160, 0.170]
Weekend Admission (<i>Yes</i>)	8,175	992	0.427	[0.41, 0.44]
Elective Admission (<i>Yes</i>)	245,870	65,034	-0.435	[-0.44, -0.43]
Race				
<i>White</i>	171,661	44,352	-0.045	[-0.05, -0.04]
<i>Black</i>	16,168	3,515	0.029	[0.02, 0.04]
Patient Co-morbidities				
<i>AIDS</i>	152	34	Not Significant	
<i>Congestive Heart Failure</i>	4,329	1,246	0.297	[0.27, 0.32]
<i>Pulmonary Disease</i>	38,968	10,427	0.541	[0.05, 0.06]
<i>Coagulopathy</i>	3,307	851	0.347	[0.33, 0.37]
<i>Depression</i>	30,232	8,430	-0.065	[-0.08, -0.06]
<i>Hypertension</i>	111,360	32,295	-0.073	[-0.08, -0.06]
<i>Fluid and Electrolyte Disorders</i>	15,574	4,562	0.517	[0.50, 0.53]
<i>Metastatic Cancer</i>	1,720	153	0.556	[0.52, 0.59]
<i>Neurological Disorders</i>	8,870	2,170	0.077	[0.06, 0.09]
<i>Paralysis</i>	6,523	1,088	0.440	[0.42, 0.46]
<i>Peripheral Vascular Disorders</i>	3,892	1,246	0.076	[0.06, 0.09]
<i>Pulmonary Circulation Disorders</i>	865	211	0.424	[0.38, 0.46]
<i>Renal Failure</i>	3,169	880	0.182	[0.15, 0.20]
<i>Solid Tumor Without Metastasis</i>	1,256	218	0.278	[0.24, 0.31]
<i>Weight Loss</i>	1,565	252	0.724	[0.69, 0.76]
Number of co-morbidities				
0	100,960	21,259	-0.319	[-0.34, -0.29]
1	83,152	20,751	-0.195	[-0.21, -0.18]
2	55,257	15,550	-0.084	[-0.09, -0.07]
Hospital Annual Volume				
<i>Very low</i>	9,120	2,025	0.105	[-0.01, 0.03]
<i>Very High</i>	116,075	31,022	-0.085	[-0.08, -0.07]
Hospital Region				
<i>Midwest</i>	68,491	15,830	-0.043	[-0.05, -0.04]
Hospital Bed Size				
<i>Small</i>	38,583	9,487	-0.031	[-0.04, -0.02]
<i>Large</i>	186,418	46,668	0.097	[0.09, 0.10]
Discharge Year				
2007	57,443	15,577	-0.067	[-0.07, -0.06]
2008	70,307	14,480	-0.098	[-0.10, -0.09]

OR: Odds Ratio, CI: Confidence Interval,

* All predictors are binary variables that have the values (1= Yes, 0= No). The reference value is 0.

** The logit estimates of the Logistic regression,

*** We model the probability of the occurrence of inpatient mortality assuming the predictor exists.

**** Parameter estimate of linear regression

Table 5 Significant Predictors of Patient Disposition to a Care Facility versus Disposition to Home after Spinal Fusion Surgery

Significant Predictors	Disposition to a Care Facility versus Disposition to Home		
	Coeff*	OR	95% CI
Age	0.081	1.08	[1.08, 1.09]
Sex	0.393	1.48	[1.45, 1.52]
Weekend Admission (<i>Yes</i>)	0.856	2.35	[2.21, 2.51]
Elective Admission (<i>Yes</i>)	-0.986	0.37	[0.36, 0.39]
Hospital Location (<i>Urban</i>)	0.473	1.60	[1.50, 1.71]
Teaching Hospital (<i>Yes</i>)	0.171	1.19	[1.16, 1.22]
Race	Not Significant		
<i>White</i>	Not Significant		
<i>Black</i>	0.406	1.50	[1.42, 1.58]
<i>Hispanic</i>	0.255	1.29	[1.21, 1.38]
Patient Co-morbidities	Not Significant		
<i>AIDS</i>	0.524	1.69	[1.10, 2.60]
<i>Alcohol Abuse</i>	0.460	1.58	[1.46, 1.72]
<i>Deficiency Anemia</i>	0.550	1.73	[1.66, 1.81]
<i>Rheumatoid Arthritis</i>	0.173	1.19	[1.12, 1.27]
<i>Chronic Blood Loss Anemia</i>	0.720	2.05	[1.86, 2.27]
<i>Congestive Heart Failure</i>	0.423	1.53	[1.42, 1.64]
<i>Coagulopathy</i>	0.661	1.94	[1.78, 2.11]
<i>Diabetes</i>	0.525	1.69	[1.55, 1.84]
<i>Drug Abuse</i>	0.551	1.73	[1.57, 2.00]
<i>Hypertension</i>	-0.189	0.83	[0.80, 0.86]
<i>Lymphoma</i>	0.344	1.41	[1.18, 1.69]
<i>Fluid and Electrolyte Disorders</i>	0.859	2.36	[2.26, 2.47]
<i>Metastatic Cancer</i>	0.375	1.46	[1.29, 1.65]
<i>Neurological Disorders</i>	0.566	1.76	[1.66, 1.87]
<i>Obesity</i>	0.238	1.27	[1.21, 1.33]
<i>Paralysis</i>	1.773	5.89	[5.50, 6.31]
<i>Peripheral Vascular Disorders</i>	Not Significant		
<i>Psychoses</i>	0.693	2.00	[1.85, 2.17]
<i>Pulmonary Circulation Disorders</i>	0.480	1.62	[1.38, 1.89]
<i>Renal Failure</i>	0.280	1.32	[1.22, 1.44]
<i>Weight Loss</i>	0.990	2.69	[2.37, 3.06]
Number of co-morbidities	Not Significant		
0	-0.842	0.43	[0.39, 0.48]
1	-0.464	0.63	[0.59, 0.68]
2	-0.192	0.83	[0.79, 0.87]
Hospital Annual Volume	Not Significant		
<i>Very Low</i>	0.244	1.28	[1.19, 1.37]
<i>Low</i>	0.147	1.16	[1.11, 1.22]
<i>High</i>	0.148	1.16	[1.13, 1.20]
Hospital Region	Not Significant		
<i>North</i>	0.706	2.03	[1.95, 2.11]
<i>Midwest</i>	-0.204	0.82	[0.79, 0.85]
Hospital Bed-size	Not Significant		
<i>Small</i>	-0.469	0.63	[0.60, 0.65]
<i>Medium</i>	-0.173	0.84	[0.82, 0.87]
Discharge Year	Not Significant		
2004	0.342	1.41	[1.36, 1.46]
2005	0.284	1.33	[1.28, 1.38]
2006	0.145	1.16	[1.12, 1.20]
2007	0.155	1.15	[1.13, 1.21]

OR: Odds Ratio, CI: confidence Interval, LOS: Length of Stay

*Coefficient = logistic regression coefficient.

Table 6 Significant Predictors of Patient Disposition to HHC versus Disposition to Home after Spinal Fusion Surgery

Significant Predictors	Disposition to HHC versus Disposition to Home		
	Coeff*	OR	95%CI
Age	Not Significant		
Sex	0.221	1.25	[1.22, 1.28]
Weekend Admission (<i>Yes</i>)	0.288	1.32	[1.22, 1.42]
Elective Admission (<i>Yes</i>)	-0.187	0.83	[0.80, 0.87]
Hospital Location (<i>Urban</i>)	Not Significant		
Teaching Hospital (<i>Yes</i>)	-0.210	0.81	[0.79, 0.83]
Race			
<i>White</i>	0.182	1.20	[1.17, 1.24]
<i>Black</i>	0.403	1.49	[1.41, 1.57]
<i>Hispanic</i>	0.605	1.82	[1.72, 1.93]
Patient Co-morbidities			
<i>AIDS</i>	Not Significant		
<i>Alcohol Abuse</i>	Not Significant		
<i>Deficiency Anemia</i>	0.557	1.73	[1.64, 1.81]
<i>Rheumatoid Arthritis</i>	0.111	1.12	[1.04, 1.20]
<i>Chronic Blood Loss Anemia</i>	0.662	1.94	[1.74, 2.16]
<i>Congestive Heart Failure</i>	0.242	1.27	[1.16, 1.38]
<i>Coagulopathy</i>	0.403	1.49	[1.35, 1.64]
<i>Diabetes</i>	0.274	1.31	[1.18, 1.45]
<i>Drug Abuse</i>	0.269	1.30	[1.17, 1.44]
<i>Hypertension</i>	-0.103	0.91	[0.87, 0.95]
<i>Lymphoma</i>	Not Significant		
<i>Fluid and Electrolyte Disorders</i>	0.580	1.78	[1.69, 1.88]
<i>Metastatic Cancer</i>	0.701	2.00	[1.75, 2.29]
<i>Neurological Disorders</i>	0.150	1.16	[1.09, 1.24]
<i>Obesity</i>	Not Significant		
<i>Paralysis</i>	0.733	2.08	[1.91, 2.26]
<i>Peripheral Vascular Disorders</i>	0.140	1.15	[1.05, 1.25]
<i>Psychoses</i>	0.271	1.31	[1.20, 1.43]
<i>Pulmonary Circulation Disorders</i>	0.361	1.43	[1.19, 1.72]
<i>Renal Failure</i>	0.142	1.15	[1.04, 1.27]
<i>Weight Loss</i>	0.409	1.49	[1.27, 1.76]
Number of co-morbidities			
<i>0</i>	-0.562	0.57	[0.51, 0.63]
<i>1</i>	-0.295	0.75	[0.69, 0.81]
<i>2</i>	-0.113	0.90	[0.85, 0.95]
Hospital Annual Volume			
<i>Very Low</i>	-0.202	0.82	[0.76, 0.88]
<i>Low</i>	-0.212	0.81	[0.77, 0.85]
<i>High</i>	Not Significant		
Hospital Region			
<i>North</i>	0.808	2.22	[2.14, 2.31]
<i>Midwest</i>	-0.382	0.67	[0.66, 0.71]
Hospital Bed-size			
<i>Small</i>	-0.314	0.73	[0.71, 0.76]
<i>Medium</i>	-0.200	0.82	[0.80, 0.85]
Discharge Year			
<i>2004</i>	0.164	1.17	[1.13, 1.22]
<i>2005</i>	Not Significant		
<i>2006</i>	0.105	1.11	[1.07, 1.15]
<i>2007</i>	0.177	1.19	[1.15, 1.23]

OR: Odds Ratio, CI: confidence Interval, LOS: Length of Stay

*Coefficient = logistic regression coefficient.

The multivariate analysis for patient comorbidities (Table 3) demonstrated that inpatient mortality was significantly lower for patients with zero or one comorbidity (OR=0.25 and OR=0.66). Pulmonary circulation disorders and AIDS comorbidities showed the highest impact on inpatient mortality when compared to other co-morbidities; the difference in log-odds for inpatient mortality is expected to change by 1.7 and 1.9 respectively, given the other variables in the model are held constant. Interestingly, the analysis showed significantly higher inpatient mortality for hospitals with large bed-size (OR=1.3) as well as teaching hospitals (OR=1.8) which may be partly due to patient selection since larger facilities are likely to have the capacity to manage higher risk patients.

Table (4) shows the length of stay is prolonged for patients with co-morbidities as weight loss (+70%), metastatic cancer (+60%), fluid and electrolyte disorders (+52%), and paralysis (+40%). In addition, patients without any comorbidities have a 31.9% reduction in the length of stay. The analysis also demonstrates that elective admissions tend to be shorter when compared to non-elective admissions given the other variables in the model are held constant. The same is also valid for patients discharged in years 2007 and 2008.

The results in Tables (5 and 6) describe a series of logistic regressions, each of which models post-hospital disposition type - Care Facility or Home Health Care (HHC) or discharge to home. Table 4 shows that the majority of comorbidities significantly affect the likelihood of discharge to another care facility or to Home Health Care. Paralysis was found to be the co-morbidity with the highest impact

on disposition to another care facility (OR=5.89) or to Home Health Care (OR=2.08) as compared to home dispositions. The results also show that fewer patients are disposed to another care facility in recent years (e.g. 2007, OR=1.15) than earlier years (e.g. 2004, OR=1.41) which may reflect improvements in surgical management or changes in practice with the implementation of clinical pathways.

4.3 Model Validation

Using the 20% validation data set, we assessed each model's performance on unseen data. For the inpatient mortality model, this was done by examining the discrimination performance as measured by the area under the receiver operating characteristic (ROC) curve [147] which is found to be 91.8%. This measures the ability of the predictive model to correctly classify those patients who will have died during their hospital stay. For the length of stay model, the training and validation data sets had R^2 of 0.25; final coefficients can therefore be produced from a full data set (R^2 0.25). Residuals were normally distributed, and there was no evidence of heteroskedasticity and the scatter plot of predicted and residual values adequately fit the data (R^2 [adjusted] = 0.26) which confirms that model assumptions had been satisfied. Finally, for the multinomial regression model of the patient disposition status was validated using the Lowess smooth of the deviance residuals versus predicted probabilities plot. The Lowess smooth is found to be approximately a horizontal line with zero intercept that suggests a goodness of the model fit.

4.4 Discussion

Population based analysis for this project utilizes data originating from healthcare providers (discharge data) to provide insight into surgical procedure outcome risk at the patient-level. The results can then be presented to healthcare providers caring for patients in the pre-operative setting to provide improved guidance and counseling. An overview of the closed-loop health information flow chart is illustrated in Figure (2).

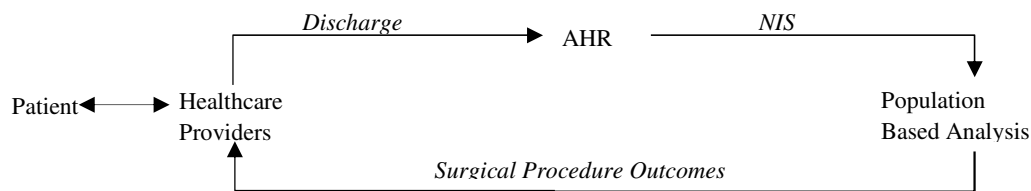


Figure 2 The closed-loop health information flow chart

We utilized the test case of spinal fusion surgery to demonstrate the utility and validity of this approach in guiding practitioners in the pre-operative setting. We found that outcome predictive modeling was effective in identifying statistically significant variables that affect outcomes following inpatient surgical procedures using an underutilized data source (NIS). This approach also has the advantage of being a data derivative in that it does not obtain original data from a direct query of discharge data as in data-driven clinical decisions support systems providing a reduced need to directly manage PHI data.

As a use case, our findings support observations reported previously in the surgery literature that decreased length of stay and decreased mortality are strongly associated with high-volume hospitals [148]. Similarly, co-morbidity data and risk stratification are commonly used and must be incorporated into

prognostic models of survival used to guide preoperative patient selection and patient counseling, such as outcomes used for prognostication associated with primary spinal surgery in the setting of spinal metastasis [141, 149]. As an additional point of face validity, inpatient mortality of 0.3% in our analysis was comparable to that reported by a national study examining selected spinal fusion procedures (0.25%) [150]. While factors that are associated with different disposition status are often not examined with single institution studies including in the setting of spinal surgery [151, 152], disposition status is a valuable factor for optimal surgical planning and may be beneficial to clinicians, facilities, and the patient's family in proper preoperative planning and counseling for anticipated outcomes. In addition to identifying a number of co-morbidities associated with home healthcare and disposition to another care facility, both the elderly and African Americans were observed to also be more likely to require home healthcare or disposition to another care facility. While elderly patients expectedly require greater resources to make a sufficient recovery, the higher rate of African American home healthcare utilization and care facility use support the large body of literature associated with racial disparities and represent an area for early social work and case management to best utilize family and social support in the establishment of post-operative surgery and recovery plans [15].

4.5 System Limitations

We did not consider possible interaction effects among the set of studied data elements since the number of variables increased dramatically. The interpretation of the data is limited since the underlying data is retrospective and limited to the inpatient clinical setting with limited generalizability to other settings. It is

believed that selection bias in retrospective studies is minimized if the entire population is selected by the study [124]. Given the modeling was established with spinal surgery results, there is also limited ability to generalize the resulting models to other procedures, however, the population based approach could be replicated with other selected procedures using the same analytic methods which might be challenging to do for each individual surgical procedure at the clinical setting due to limitations of time and resources. Besides, the analytic approach described in this chapter does not provide clinicians with the flexibility to define their own inclusion and exclusion criteria especially for which ICD9-CM codes to include. In large datasets as NIS, there is significant potential for coding error or variability in coding which may lead to misleading results due to misclassification [125]. All these limitations, may adversely impact the usability of population-based model-driven systems to support decisions at the clinical setting for the surgical risk assessment process.

Chapter 5: Data-driven System for Personalized Surgical Risk Assessment²

In this chapter we transition from the design of model-driven clinical decision support system to the design of population-based data-driven system for personalized surgical risk assessment adopted by this research to achieve the aims stated in section 1.1 of Chapter 1. We describe how the construction of model-driven system informed the design of data-driven system. Subsequently, we present the data-driven system architecture and describe clinical use cases where the system can be used with a focus on targeted rare disease states and perceived high complexity patients.

The main purpose of modern clinical decision support systems is to assist clinicians at the point of care [153]. Therefore, such systems should be flexible enough to cover the various cases at the point of care where patients have different characteristics and undergo different procedures especially that the majority of elderly patients have multiple comorbidities that must be addressed by their patient care team in a personalized way [154].

The limitations of the model-driven system described in chapter 4 motivated us to design a flexible system that provides clinicians with means to submit ad hoc queries against NIS population data. Such queries will enable the clinician to enter patient-specific data and choose the right ICD-9-CM code that better reflects the patient's surgical procedure.

²Ahmad AbuSalah, Genevieve Melton-Meaux, Terrence Adam, "A Personalized Surgical Risk Assessment Tool to Support Clinical Decision Making," the International Conference on Information Communication Technologies in Health, 2012, in press.

Also, during the selection of the ICD-9-CM codes needed for the construction of the model-driven system, we learned that such codes may vary over time especially when we use large datasets that span over multiple years. New codes may be created, some may be deleted, and some may be mapped to different codes [133]. We finally learned that it is very possible that we will need to analyze tens or hundreds thousands of discharge records to compute surgical risks for one procedural code. Therefore, simple database tuning may not be sufficient to provide timely response to clinician's queries.

5.1 POET's Architecture

We utilized information technology to create the Procedure Outcomes Evaluation Tool (POET); a data-driven system providing clinicians with timely access to NIS population data to generate surgical risks about inpatient mortality, patient's length of stay, and discharge disposition status tailored to the patient, hospital, and admission information (Table 7). The ability to filter decision support system recommendations to the clinician by using multi-attribute queries combining those recommendations for patients with comorbidities, and using large datasets for clinical recommendations have been identified as three grand challenges in clinical decision support [155].

Table 7 Surgical Procedure Outcomes

Outcome	Definition
Inpatient Mortality Rate	The rate of “patients like me” who died during the same hospitalization where the surgical procedure(s) was/were performed.
Average Length of Stay	Calculated by dividing the sum of inpatient days by the number of patients admitted for the same surgical procedure
Disposition Status	Patient's anticipated location or status following the procedure (Home, Home Health Care, Another care facility)

The development of the POET system followed an agile software engineering methodology in which the design has been refined several times in an iterative fashion based on clinical user requests. The modular design of POET subdivides the system into smaller parts (modules) that can be independently created and then used in different systems to drive multiple functionalities. In software engineering, this approach provides the ability to create future enhancements of some selected pieces without re-engineering the whole system.

The set of data elements to be included in POET was refined several times by domain experts to maintain content relevant for surgical risk assessment. The set was also compared to core dataset for preoperative assessment by Ahmadian et. al. [28]. Finally, we excluded data elements which had a significant percentage of missing values since such data elements will limit the ability to answer users’ queries. The data elements covered by the POET system comprise 96% of NIS data elements as listed in Table (8).

Table 8 The list of NIS data elements covered by the POET System

<p>Patient Characteristics:</p> <p><i>Demographics:</i> Age, Sex (male, female), Race (white, black, Hispanic, others)</p> <p><i>Comorbidities:</i> Acquired Immune Deficiency Syndrome (AIDS), Alcohol Abuse, Deficiency Anemia, Rheumatoid Arthritis, Blood Loss Anemia, Congestive Heart Failure, Pulmonary Disease, Coagulopathy, Depression, Diabetes, Drug Abuse, Hypertension, Hypothyroidism, Liver Disease, Lymphoma, Fluid and Electrolyte Disorders, Metastatic Cancer, Neurological Disorders, Obesity, Paralysis, Peripheral Vascular Disorders, Psychoses, Pulmonary Circulation Disorders, Renal Failure, Solid Tumor Without Metastasis, Valvular Heart Disease, Weight Loss.</p> <p><i>Primary diagnosis:</i> ICD-9-CM code</p> <p><i>Primary procedure:</i> ICD-9-CM code</p> <p><i>Secondary procedure:</i> Two ICD-9-CM secondary procedure codes.</p>
<p>Hospital Characteristics:</p> <p><i>Region:</i> Northeast, Midwest, South, West</p> <p><i>Location:</i> Rural, Urban</p> <p><i>Bed-size:</i> Small, Medium, Large</p> <p><i>Teaching Status:</i> Yes, No</p>
<p>Admission characteristics:</p> <p><i>Weekend Admission:</i> Yes, No</p> <p><i>Elective Admission:</i> Yes, No</p>

The technical specifications of the system can be summarized as follows:

- 1- The user should be able to query the NIS population database using all the identified data elements or any subset of those data elements.
- 2- All data elements are considered optional except the primary surgical procedure since the system aim is to generate surgical procedure risk assessment.
- 3- Since the underlying dataset spans over several years, the system should not expect the user to do any manual work to map ICD-9-CM codes for the primary and secondary procedures as well as the primary diagnosis.
- 4- The system should not assume user's knowledge of ICD-9-CM code of primary and secondary procedures as well as primary diagnosis.
- 5- The system should generate generalized surgical risk estimates as well as personalized surgical risk estimates for users.

- 6- The system should allow multi-selection of comorbidities from the list provided by NIS database.
- 7- Only authorized users will have access to the system through a user name and password.
- 8- The system should be performance tuned for user's multi-attribute queries.
- 9- The system should be designed in modules that are created at the data-tier level whenever possible since future enhancement may embed POET into an EMR system or create an internet-based version of it.
- 10- The system should comply with NIS data agreement.

The system architecture is shown in Figure (3).

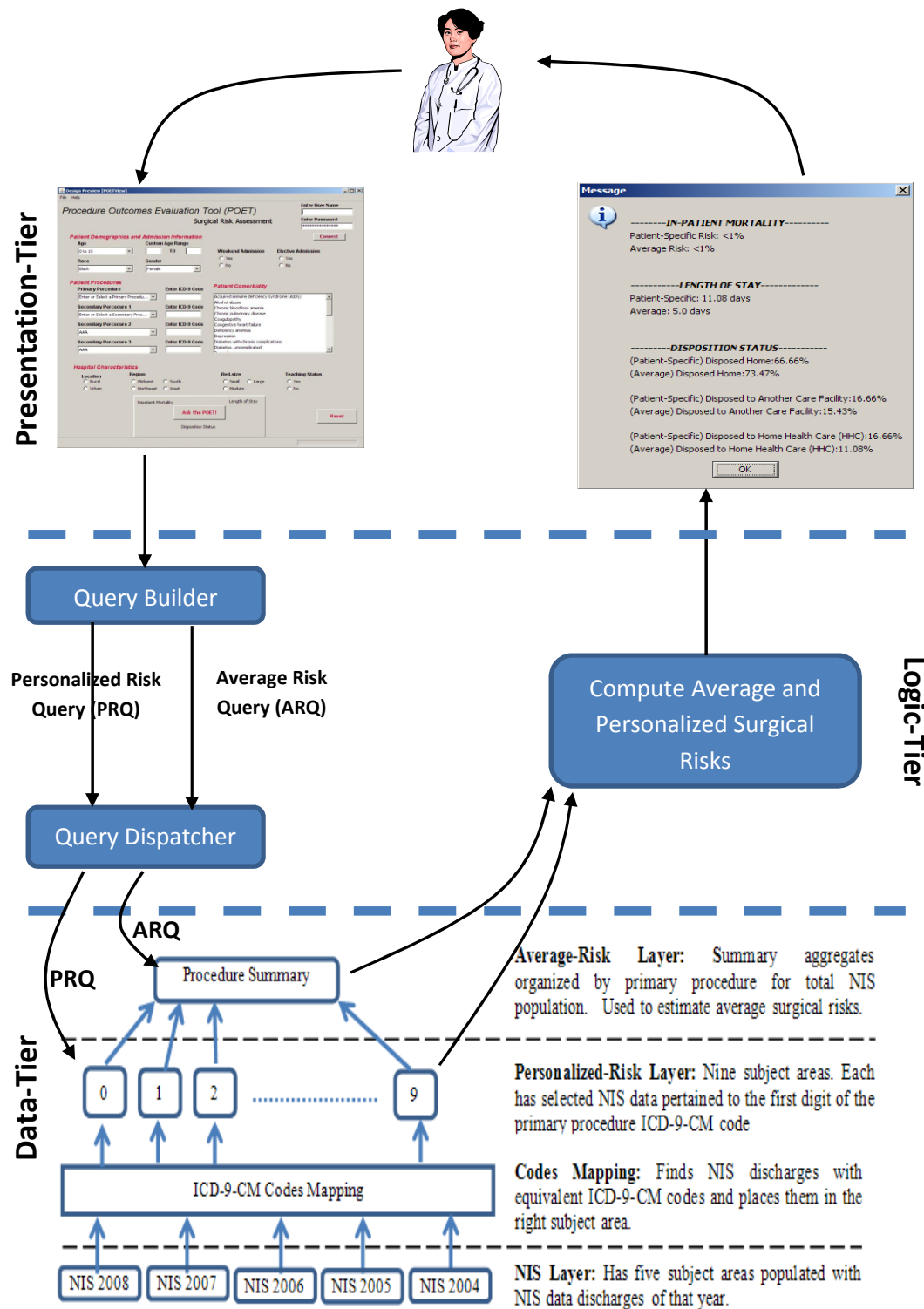


Figure 3 The Procedure Outcomes Evaluation Tool

Table (9) lists POET's main components.

Table 9 Summary of POET's main components and functionalities

Component	Description	Functionality
Presentation Tier	User Interface	Provides a user friendly interface that receives user's queries and displays results.
Logic Tier	Data Analysis	Processes user's queries and retrieved results.
Data Tier	Database	Stores POET's main data source (NIS)
ETL	Extracts, Transforms, and Load new Data	Updates POET's database with released NIS data.

5.1.1 POET's Presentation Tier

It provides the system interface (Figure 4, Source Code: Appendix A1) for ad-hoc queries about surgical procedures. The queries are usually submitted by the user who is performing the personalized surgical risk assessment process. This research gives minor focus on the interface design from the perspective of human-machine interaction between the POET system and the user.

Figure 4 POET's Graphical User Interface

The interface provides the user with a module to find ICD-9-CM code of interest based on complete or partial word entry. To do so, files with all ICD-9-CM codes and their definitions were downloaded, processed, and then persisted into the database of the system [133]. The interface covers all the data elements and enables multi-selection of comorbidities. The Help window shows basic information about the POET system and generated risks.

5.1.2 POET's Logic Tier

POET's logic-tier receives the user inputs, query builder and dispatcher functionalities (Source Code: Appendix A1) to build the average risk query (ARQ)

and personalized risk Query (PRQ) to dispatch the queries to the right table in the database to compute two types of surgical risks:

- Patient-specific risk assessment: The surgical risk is computed using the population of patients that match the input criteria.
- Average risk assessment: The surgical risk is computed using the population of patients that underwent the selected primary procedure regardless of any other factor.

The resulting surgical procedure outcomes are then routed to the presentation tier to inform the practitioner's decision-making process.

POET system is designed to comply with the NIS data agreement. From an implementation perspective, the system is designed to prohibit raw data disclosures and does not generate surgical risks for queries with result sets size less than 11 cases to meet requirements of the NIS data agreement.

5.1.3 POET's Data Tier

A well-designed data warehouse has a pyramid of summary tables with each layer containing additional detail until you arrive at the fact table at the bottom of the pyramid [156]. At the top of the pyramid, relatively small summary tables are used to facilitate rapid query execution. These design optimizations allow at least 90 percent of the queries to run against the summary tables to provide very fast system response [156].

For the design of POET Data Tier, the fact tables are populated with NIS discharge data, described in chapter 3, at the bottom level (NIS Layer) of the pyramid structure (Figure 5).

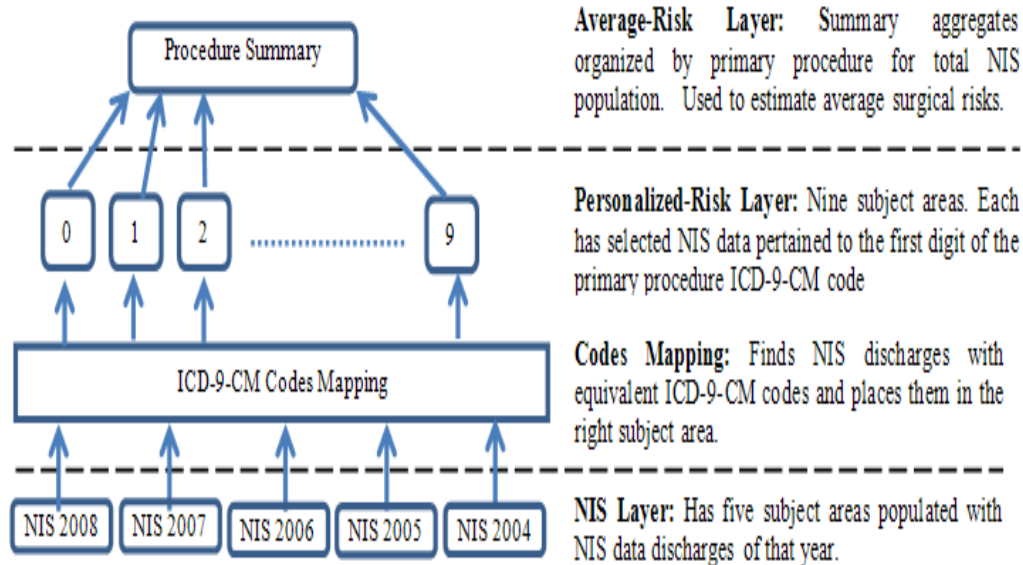


Figure 5 Data Architecture Layers in POET

Each subject area in the NIS layer contains one year of source NIS data loaded into three fact core data table (patient demographics and admission data elements), comorbidities data table, and hospital data table.

At the second level (Personalized-Risk Layer) of the pyramid, we grouped all discharges into subject areas according to the first digit of the primary procedure ICD-9-CM codes creating a “primary procedure-driven” database architecture. Each subject area has three tables: patient demographics and procedure information core table, patient comorbidities table, and hospital characteristics table. The tables at the Personalized-Risk Layer include the data elements that have been previously identified in Table (7) only; which are subset of the data elements in the fact tables at the NIS Layer. The motivation for this design is to create relatively smaller tables with a limited number of data elements from NIS

fact tables. This enhances the database performance and reduces query execution time.

To match the records from the NIS layer with the right subject group in the Personalized-Risk Layer, we need to find the equivalent ICD-9-CM codes over years to be grouped correctly. For example, ICD-9-CM procedure code 81.53 (valid until Oct 2005) is mapped to 00.70, 00.71, 00.72, and 00.73 starting Oct 2005. These mappings must be considered during the risk assessment process that spans over several years for various purposes, such as long-term clinical studies, epidemiological and other research, and calculating costs and other institutional financial measurements [157, 158] to ensure data consistency across different years of the data. The new codes as well as the conversion rules are issued by the ICD-9-CM Coordination and Maintenance Committee (CMC), which is sponsored jointly by The National Center for Health Statistics (NCHS) and the Centers for Medicare & Medicaid Services (CMS), every year since 1986 [133]. The CMC conversion files were downloaded, processed, and then persisted into the database of the system for this purpose.

We created a module (Source Code: Appendix A2) that automatically maps ICD-9-CM codes according to ICD-9-CM conversion rules issued by the CMC for procedure and diagnosis codes. The created module finds the 2008 equivalent ICD-9-CM codes for the primary procedure, secondary procedure, and primary diagnosis data elements over the 5 years of NIS data. The module enables risk estimates to be developed based on consistent coding across the different years of data. By creating an automated module, POET will work with any future releases

of conversion rules when they are incorporated into the conversion rules tables in the POET database to facilitate system sustainability.

Another module (Source Code: Appendix A3) was created that will extract data from the NIS layer and load it into the right subject area at the Personalized-Risk Layer utilizing the module that automatically finds the 2008 equivalent ICD-9-CM codes. This design eliminates the need to run the ICD-9-CM mapping module on-the-fly during the execution of the user queries.

The Personalized-Risk Layer of the pyramid is used to compute patient-specific inpatient mortality, length of stay, and disposition status because queries to compute patient-specific risk estimates will incorporate several data elements and from tuning perspective it is better for such complex queries to be submitted to a smaller size table that has all the data pertained to the primary procedure of interest and at the same time avoid any on-the-fly ICD-9-CM code conversion.

In the third layer (Average-Risk Layer), we computed the total counts of discharges, total number of patient deaths, total numbers of patients discharged to home, total numbers of patients discharged to Home Health Care (HHC), total numbers of patients discharged to another care facility, and the average hospital length of stay for each primary procedure from the Personalized-Risk Layer.

The Average-Risk Layer is used to compute the average inpatient mortality, length of stay, and disposition status since it uses the population of cases with similar primary procedure ICD-9-CM code regardless of any other factors. A module (Source Code: Appendix A3) is also created to populate the Average-Risk

Layer by looping on all primary procedures in nine subject areas in the Personalized-Risk Layer.

The modules that are used to populate The Personalized-Risk Layer and the Average-Risk Layer can be also used whenever a new table (for newly released NIS data) is loaded in the NIS-Layer.

5.2 Use Cases

- Case 1: POET as a Resource Planner

Case description: A surgical planner is interested in the mortality risk for nonagenarians white patients completing spinal surgery. The surgical planner is also interested in how long a nonagenarian patient usually stays in the hospital and what is the likelihood that those patients are being admitted to SNF (Skilled Nursing Facilities)?

Case Challenge: such a focused population in the oldest old frequently has limited published data available due to the paucity of patients surviving to such ages and the small numbers getting a particular procedure are even more infrequent.

POET results: With a couple of selections on the POET interface, the data is rapidly available to the researcher in Table (10).

Table 10 Spinal Surgery Query

Age Group	LOS (days)	Mortality (%)	NF Risk* (%)
21 to 30	5.6	0.17	0.96
31 to 40	3.31	0.12	0.70
41 to 50	3.15	0.13	1.14
51 to 60	3.76	0.24	2.66
61 to 70	4.2	0.30	6.14
71 to 80	4.95	0.70	14.19
81 to 90	6.36	2.33	25.24
91 to 99	7.9	11.11	37.04
Total population	4.05	0.34	4.71

* The risk of patient being disposed to a nursing Facility (NF)

Case Conclusion: This query could be used by the surgical planner to estimate how long the same nonagenarian patient will be in the hospital (average of 7.9 days) and help estimate the likelihood that nursing home admissions will need to be planned after the procedure (37%). These results can help the hospital manager do workforce planning and bed management. Finally for the provider participating in preoperative planning, they can obtain important information for the caregiver of the nonagenarian patient who is concerned their loved one may not survive the surgery. The provider can utilize the query results at the point of care and report that for other nonagenarians experiencing the same procedure there is an 11.1% risk of dying in the hospital which is higher than for younger age groups.

- Case 2: POET as a personalized surgical risk assessment tool for rare cases

Case description: how do patients who have hemophilia fare with high bleeding risk surgeries like spinal surgery?

Case Challenge: find “patients like me” who have unusual diseases and may not be supported by guidelines and published literature.

POET results: By selecting the appropriate procedure disease, the user finds there are 11 “patients like me” who underwent a spinal surgery and had hemophilia as shown in Table (11).

Table 11 Spinal Surgery with Hemophilia

Patient Groups	LOS (days)	Mortality (%)
Hemophilia Patients (Personalized-Risk)	5.6	0.0
Total Patients (Average Risk)	4.1	0.004

Case Conclusion: The results indicate that this rare population has a higher length of stay relative to the larger spinal surgery population but had no episodes of patient mortality during hospitalization.

5.3 Discussion

POET provides a data-driven solution to compute pertinent average and personalized surgical risks at the point of care to inform clinicians during preoperative evaluation process. Prior work has identified critical information needs at the point of care for preoperative assessment as mentioned in Chapter 1 of this thesis.

Patients with multiple comorbidities may not have their clinical risk well categorized in clinical guidelines. In such cases accurate estimates of clinical risk can be difficult to estimate and leave both patient and providers in the gray zone where the “art of medicine” is the only tool for difficult decision-making [159]. In this case, data-driven systems will grant clinicians access to valuable information of millions of patient cases over years to generate an evidence-based decision.

For a provider at a clinical site which lacks formal preoperative assessment, tools such as POET help the provider assess an individual patient's clinical risk and can also help with inpatient care planning as the results in Tables 10 and 11 highlight. Patient-specific information on length of stay and discharge disposition are important for helping providers and facilities manage patient resource utilization and plan for appropriate clinical referrals such as nursing home placement if this likelihood is high as noted in the results in Table 9.

Given the potential impact of accountable care organizations and potential for quality and outcomes accountability, it is important for surgical teams, hospital providers, social workers, and inpatient case managers to have information support tools to optimize surgical care needs and manage hospital workflows [160].

The development of population-based data-driven clinical decision support system is challenging due to the large amount of data that require management in order to generate personalized surgical risk measures using patient, hospital, and procedure data elements.

Compared to the construction of model-driven systems, the POET system took a significant amount of resources including software, hardware, and thousands lines of code. The construction of a model-driven system prior to the POET system enabled us to proactively design based on prior knowledge about the underlying NIS population data. Data profiling is another approach that can be used prior to the design and development of data-driven system in which data elements are examined for potential data quality issues [161].

The primary procedure-driven design of the database supports the tuning efforts to manage large datasets. The ARCHeS Simulator which implements the Archimedes Model follows the same concept but for disease-based design [112]. More specifically, the design of the database revolves around the main goal of the system. There have been attempts to create generalized clinical databases to represent diverse clinical data in a standard structure but such systems have not achieved wide spread acceptance especially with databases that have high volumes of transactions which require very fast query processing [162].

The POET system computes two types of risks: average risk and personalized risk. The ability to compare the average risk estimate to the patient-specific risk estimate is useful to identify high risk patients with more extensive comorbid conditions undergoing more complex surgical procedures based on real data [36, 40, 163].

POET is by no means a replacement for rigorous clinical trials. Rather, it contributes to a provider's ability to create risk projections for a clinical procedure, as well as estimate resource use and fills an important unmet clinical need.

5.4 System Limitations

We described the architecture of POET system as prototype solution for using population data to build data-driven surgical decision support for surgical risk assessment. The created system is neither formally nor thoroughly tested. Therefore, design and implementation flaws may be identified during testing. Besides, the architecture design and created modules are developed to fit NIS data structure and format creating challenges on reuse of the system. Therefore, re-

engineering of some components will be needed if other databases are incorporated. The POET system is dependent on NIS data, thus the limitations of NIS data described in chapter 3 also apply to POET. Finally, the current version of POET assumes the users to be clinicians. The presentation tier is not designed to be used by patients.

Chapter 6: Providers Evaluation of The Personalized Surgical Risk Assessment Systems

This chapter conducts a subjective evaluation to measure clinician's satisfaction with the usability of the data-driven system (POET) in terms of system usefulness as well as information and interface quality. Also, compare clinician perceptions on the usability of the data-driven and model-driven systems to achieve the aims stated in section 1.1 of Chapter 1.

6.1 Evaluation Model

We evaluated the POET architecture according to the four-phase model for evaluating clinical decision support architectures described by Wright and Sitting [93]. The framework is based on the clinical decision support roadmap [20] published by the American Medical Informatics Association. The framework is readily adoptable for the evaluation of clinical decision support architectures [164, 165].

The framework evaluates four general classes of decision support architectures: standalone, integrated, standardized and service-oriented. The evaluation framework has four key elements:

- Feature Determination:

The evaluation framework identifies desirable features of clinical decision support architectures as listed by Wright and Sitting [93]. Not all listed features apply to the POET system since some apply to rule-based or integrated clinical decision support systems.

- Existence and Use:

This element recommends that decision support systems have a degree of widespread use. However, this may not be feasible in the case of system prototype development and evaluation as in the case of POET.

- Utility:

This element tests clinical decision support system usability. For this purpose, POET was loaded on a dedicated ThinkPad laptop personal computer. Testing was conducted via a paper and pencil survey to assess clinicians' attitudes toward the use of POET. A convenience sample of seven physicians (three males and four females) was recruited to complete the assessment from the Department of surgery and Department of Medicine at the University of Minnesota. The number of clinicians matches the number used in other user evaluation studies of clinical decision support systems [166, 167]. The sample comprised clinicians with varied clinical experience. All subjects were experienced computer users in both the home and office settings and they reported daily computer use.

We used the IBM Computer Systems Usability Questionnaire (CSUQ) instrument to measure the system's utility [68]. It is designed to obtain subject satisfaction and usability data on information systems. The instrument has strong evidence of reliability ($\alpha = .95$) as well as content and construct validity [7]. It is commonly adapted to evaluate the usability of clinical decisions support systems [168-170]. It contains 19 questionnaire items with each item having a seven-point semantic distance scale (range of 1 to 7). The extremes were labeled "Strongly Agree" and "Strongly Disagree". Strongly Agree corresponds with the number

one and Strongly Disagree corresponds with the number seven. The 19 items of the CSUQ instrument are used to compute four factor scores: overall satisfaction score, System Usefulness, Information Quality, and Interface Quality. The items associated with each factor are averaged to generate the factor score value.

The IBM CSUQ was supplemented with one specific open ended question regarding the patient-specific risk assessment at the point of care. The goal behind the last open question was to identify scenarios where POET data-driven system provides clinical value when compared to population-based model-driven systems as the one presented in Chapter 3.

- Coverage

This element focuses on the ability of a decision support architecture to cover clinical knowledge when compared to other approaches. Generally, this element is used to compare rule-based clinical decision support systems. However, within the context of data-driven clinical decisions support system for personalized surgical risk assessment and for the purpose of this research, we define the coverage as the number of primary procedures that can be assessed using the POET system; i.e., the percentage of primary procedures that exists in POET's database compared to the total number of procedural ICD-9-CM codes for the year 2008 that users can use the system interface to generate personalized surgical risks.

6.2 Evaluation Results

The key elements of the evaluation framework are reviewed within the context of the POET system.

- Feature Determination:

The development of POET incorporated several of the evaluation framework's desirable features rendering a favorable evaluation. Specifically, POET:

- Avoids vocabulary issues: since POET is a prototype system, it does not require integration with other systems at this level and therefore it avoids vocabulary issues entirely.
- Is shareable: As a standalone system, POET is relatively well-suited to sharing [93]. It can simply be copied from one computer to another.
- Supports non-event driven CDS: POET is built to handle only request-driven decision support. It makes inferences only when explicitly called upon by their users making it a non-event driven system. However, it would be easy for POET to support an event-driven CDS in the future. For example, the system could send an email to a researcher on risk estimates related to a pre-determined list of surgical procedures whenever new data is populated.
- Provides population based decision support: POET generates patient-specific risk assessments for the population selected by the user based on patient, hospital, and admission characteristics. It also estimates average risks for all patients who have received a particular primary surgical procedure.

- Allows black-box services: POET makes inferences and responds to users' queries without exposing its underlying logic to the user.
 - Does not view decision support content: This point recommends decision support systems to format clinical knowledge (e.g. rules) to be both human and machine readable. It is usually used with rule-based systems to share the rule contents with similar systems. This is not applicable to POET.
 - Does not enable separation of responsibilities: It is recommended that the system developers are different from the clinical contents' developers (usually clinicians and informaticists). While this usually applies to rule-based system, in POET we followed a modular design in which some system components were designed in modules that can be easily copied to other systems or edited separately such as the module for ICD-9-CM code conversion.
 - Does not support free choice of knowledge representation syntax: This applies to integrated systems. It gives the user the freedom to view the clinical knowledge in his/her preferred syntax. This does not apply to POET system.
- Existence and Use:
A prototype of the POET system has been implemented. It is a platform independent Java desktop application that uses a Swing Framework and can be installed on any computer. In the future, POET system will be available online for users to download.
 - Utility:

The physicians used the system to get risk estimates on real patient cases and additional synthetic cases. The system was able to respond to most user requests with non-responses due to data granularity and data use agreement limitations.

The collected responses (Table 12) show mean CSUQ factor scores were: System Usefulness (M=1.80), Information Quality (M = 1.89), and Interface Quality (M=1.87).

Table 12 CSUQ Satisfaction Scores ($N = 7$)

Item	Description	Average	Std Dev
1	Overall, I am satisfied with how easy it is to use this system	1.6	1.3
2	It was simple to use this system	1.6	0.9
3	I can effectively complete my work using this system	1.8	0.8
4	I am able to complete my work quickly using this system	1.8	0.8
5	I am able to efficiently complete my work using this system	2.0	0.7
6	I feel comfortable using this system	1.6	0.6
7	It was easy to learn to use this system	1.4	0.6
8	I believe I became productive quickly using this system	2.6	1.1
<i>System Usefulness Score (Average the responses of items 1 to 8)</i>		1.80	
9	The system gives error messages that clearly tell me how to fix problems	1.8	0.8
10	Whenever I make a mistake using the system, I recover easily and quickly	1.8	0.8
11	The information (such as online help, on-screen messages, and other documentation) provided with this system is clear	1.8	0.8
12	It is easy to find the information I needed	2.2	1.1
13	The information provided for the system is easy to understand	1.6	0.9
14	The information is effective in helping me complete the tasks and scenarios	2.4	1.5
15	The organization of information on the system screens is clear	1.4	0.6
<i>Information Quality Score (Average the responses of items 9 to 15)</i>		1.89	
16	The interface of this system is pleasant	1.6	0.6
17	I like using the interface of this system	1.8	0.8
18	This system has all the functions and capabilities I expect it to have	2.6	0.8
<i>Interface Quality Score (Average the responses of items 16 to 18)</i>		1.87	
19	Overall, I am satisfied with this system	2.0	1.0
<i>Overall Satisfaction Score (Average the responses of items 1 to 19)</i>		1.85	

The average overall satisfaction was 1.89 indicating that users were generally satisfied with the system. The items “I believe I became productive quickly using this system” and “This system has all the functions and capabilities I expect it to have” had the highest CSUQ score, indicating that users’ productivity may be negatively impacted at the point of care and that the usefulness of the system will be improved if more features are added. The items “Easy to learn the system” and

“The organization of information on the system screens is clear” had the lowest mean scores indicating higher usability in these areas.

For the appended question, all the physicians were in favor of using POET data-driven system at the point of care over model-driven systems similar to the one presented in Chapter 3 of this thesis for personalized surgical risk assessment.

- Coverage:

The list of primary procedures included in POET’s NIS dataset comprise 78% of the total list of 2008 ICD-9-CM procedures [133]. However, the interface is able to generate risk aggregates for 71% of the 2008 ICD-9-CM procedures. That is due to the number of cases per procedure in the database; the system does not generate risk aggregate for result sets of less than 11 patients to meet requirements of the NIS data agreement.

6.3 Discussion

Prior research identified a lack of easy and simple tools for clinicians to access the clinical digital content on the internet despite its enormous growth [41, 66]. We evaluated POET’s architecture and usability and found that population-based clinical decision support systems have the potential to be useful and acceptable by clinicians to answer clinical enquiries.

We evaluated the POET system using the four-phase framework and model for evaluating clinical decision support architectures: feature determination, existence and use, utility, and coverage [93]. The framework also covers Nielsen’s major usability attributes of IT applications [171]. We showed that POET satisfies all

the desirable features of a stand-alone clinical decision support system listed in the feature determination phase.

For the utility phase, we evaluated the system's usefulness by measuring physician satisfaction using the IBM CSUQ questionnaire. Assessing clinician perceptions of an information system prior to dissemination is an essential step to evaluate the system's perceived usability and usefulness [172, 173]. The system answered most of the clinicians' inquiries but failed to answer some requests since the underlying population data lacked sufficient granularity to answer a clinician's inquiry specific type of clinical cases and the inherent limitations on raw data disclosure due to data agreements. Physicians also mentioned that such systems may cause workflow interruptions during system maintenance time. Table (13) summarizes the challenges we experienced with POET system to answer clinician's enquiries [62].

Table 13 Challenges of using the POET system

Challenge	Description	Impact
Cost and Workflow Interruption	<p><i>Infrastructure and Maintenance Cost:</i></p> <ul style="list-style-type: none"> Substantial computing analysis power is needed to provide timely response to users' inquiries especially when several data elements are used to generate patient-specific surgical outcome assessment. During system maintenance, the system may fail to perform its primary function. This can be due to database updates or other maintenance work. 	<ul style="list-style-type: none"> Adds extra financial, technical, and managerial burden to the clinical setting. Interrupts clinical workflow which may create substantial concerns for patient safety, workflow efficiency, and user satisfaction.
Data Issues	<p><i>Data Deficiency and Inconsistency:</i></p> <ul style="list-style-type: none"> Data quality issues may degrade the accuracy of procedure outcome assessment. The database may not have sufficient granularity for a clinician's inquiry about a specific type of cases. Limitations on raw data disclosure due to data agreements and HIPAA regulations 	<ul style="list-style-type: none"> A patient-specific procedure outcome assessment may not be possible. Limits the ability of cohort identification

All the usability questionnaire items got scores less than 3 indicating positive user responses. The four factors measured by CSUQ received scores less than 2. The lowest mean CSUQ factor score was 1.8 for “System Usefulness” factor; which indicates that physicians were generally comfortable with the system. They were able to effectively (POET is the right tool for surgical risk assessment), quickly (POET response times are fast), and efficiently (POET response does answer the clinicians inquiry) complete the patient-specific surgical risk assessment task.

To avoid impacts on physician productivity during the limited-time patient counseling session, physicians recommended that we support the Current Procedural Terminology (CPT) and ICD-10 codes in addition to ICD-9-CM codes in the POET system as some clinicians are more familiar with one coding approach than the other. They also recommended the POET system should be integrated into the Electronic Medical Record (EMR) system of the healthcare provider which some clinical decision support systems already do as mentioned in the background review in Chapter 2. In this case, surgical risk estimates will be computed automatically once patient data are populated in the EMR system. The computed estimates can augment the patient’s record. These identified clinical needs are under consideration for future work.

There are a number of methods by which risk can be assessed preoperatively related to the type of surgery, elective status of the procedure or factors related to the patient themselves. Risk assessment systems may query discharge data to estimate risk scores or use statistical modeling methods to predict risk of

developing certain outcomes such as using population-based data analysis approaches to estimate surgical risks as in our work in Chapter 4 of this thesis [62]. Physicians preferred systems such as POET at the point of care over model-driven system with enumerative statistical studies for personalized surgical risk assessment. The main reasons revolved around the limited time they have with the patient during counseling and the need of a timely response to their clinical risk assessment needs. Another reported reason is that it is not always feasible to find a model-driven system with a thorough statistical study incorporating large number of data elements that are of interest to the clinician in generating patient-specific risk assessment at the point of care. The physicians also mentioned that POET can be helpful with either clinically complicated or a rare case patient scenarios. In addition, most model-driven systems require individuals with strong mathematical and statistical background to assess patient-specific surgical risks. Moreover, extreme statistical accuracy is not usually required during the risk assessment process; it is generally preferred to estimate surgical risks that “support” clinical decisions rather than “make” clinical decisions. Therefore, querying raw data may provide sufficient insight for this focused purpose. The physicians also identified a circumstance where model-driven systems may have an advantage over systems like POET at the point of care; this case pertained to surgeons who perform a limited set of surgical procedures for most of his/her cases.

6.4 Evaluation Limitations

The four-phase evaluation framework may not have covered all dimensions needed to evaluate the system architecture and utility. Besides, the evaluation approach was applied for the prototype design of the POET system. Larger more complex systems would require more tests and as such benefit from tests requiring the collection of large sets of numerical data and statistical analysis, as well as more formal input of usability expertise.

Chapter 7: Conclusion and Future Directions

There is a growing interest in evidence-based clinical decision support systems that utilize healthcare data to assess clinical outcomes [53, 54]. Several studies demonstrated that population healthcare data are a useful and acceptable source of information for clinicians to answer various clinical enquiries, identify quality of care concerns, and enhance practice guideline development [41, 50-52, 56-59]. Despite its value, population data utilization in support of patient-specific surgical outcomes assessment is not commonly used in clinical decision-support to guide clinicians regarding likely outcomes after surgical interventions [16]. This can be due to the large number of data elements that are needed for the analysis to create patient-specific evaluations in addition to the challenges of using population-based clinical decision support systems.

In this thesis, we presented two prototype clinical decision support systems that bring population surgical outcomes data to clinicians to narrow the gap between knowledge and clinical practice within the arena of personalized surgical risk assessment which is an essential step of patient's preoperative evaluation.

The model-driven system construction approach provides a direction for collecting, preparing, and analyzing NIS population data. We selected spinal fusion procedure as a use case for the model-driven system since it is an inpatient high volume procedure that is rapidly increasing in the U.S. with potential for substantial surgical complications [127]. In turn, the creation of the POET data-driven system provides a means to access NIS population to compute average and personalized surgical risks. The modular design of the POET system is driven by

the primary procedure ICD-9-CM code to boost query performance. The system can filter decision support system recommendations to the clinician by enabling multi-attribute queries and combines those recommendations with patient's comorbidities. Both systems were used to support clinical decisions pertaining to inpatient mortality, length of stay, and discharge disposition status after a surgical procedure.

The POET data-driven system required relatively expensive infrastructure and resource commitments compared to the model-driven system. However, the data-driven system provides clinicians with the flexibility to define their own inclusion and exclusion criteria which was reflected in clinician's perceptions regarding the utility of both systems. In addition, the surgical risk values generated by the model-driven systems are predictions of surgical outcomes compared to the data-driven system where surgical risks are actually computed from the underlying NIS database. While both approaches are useful, we showed that surgical risks generated by the data-driven system will suffice to support personalized surgical risk evaluation according to clinician's feedback.

The design and evaluation of both prototype systems reinforce the need for the next generation clinical decision support systems which will improve information availability to providers to extract knowledge from population data at the point of care to provide clear expectations and optimal preoperative counseling to patients about surgical procedure outcomes.

A number of limitations existed during the design of the model-driven and data driven systems in addition to the evaluation approach. The limitations have been

identified in the last section of each chapter. We believe that the prototype design of the population-based systems we presented along with the utility evaluation study can be considered as a robust pilot study to proceed with a more rigorous research where additional resources will be needed to address those limitations.

Our next steps for potential future work are to expand this work to a more generalizable and scalable framework and interface for providers. The use of other large scale datasets which provide complementary information such as NSQIP and VASQIP databases, or a medical chart review of patients with surgeries could provide additional risk variable data needed to validate the model with actual clinical outcomes. We also anticipate that Medicare data from the Centers for Medicare & Medicaid Services (CMS) to add great value to the POET system. Medicare data will provide an important functional addition by facilitating the calculation of 30-day risk-adjusted mortality rates for surgical procedures. Finally, we will consider the integration of the POET system into the healthcare provider's EMR system so that risk estimates are embedded into the patient's EMR record.

Bibliography

- 1) Greer AE, Irwin MG. Implementation and evaluation of guidelines for preoperative testing in a tertiary hospital. *Anaesth Intensive Care* 2002; 30:326–330.
- 2) Van Klei WA, Grobbee DE, Rutten CL, et al. Role of history and physical examination in preoperative evaluation. *Eur J Anaesthesiol* 2003; 20: 612–618.
- 3) Ferrando A, Ivaldi C, Buttiglieri A, et al. Guidelines for preoperative assessment: impact on clinical practice and costs. *Int J Qual Health Care* 2005; 17:323–329.
- 4) Kozak LJ, DeFrances CJ, Hall MJ. National hospital discharge survey: 2004 annual summary with detailed diagnosis and procedure data. *Vital Health Stat* 13. 2006;1-209.
- 5) Woolger JM. Preoperative testing and medication management. *Clin Geriatr Med*. 2008;24:573-583, vii.
- 6) Pausjenssen L, Ward HA, Card SE. An internist's role in perioperative medicine: a survey of surgeons' opinions. *BMC Fam Pract*. 2008;9:4.
- 7) Cohn SL. The role of the medical consultant. *Med Clin North Am*. 2003;87:1-6.
- 8) Bock M, Wiedermann CJ. Minimizing perioperative risk - an interdisciplinary effort. *Wien Med Wochenschr*. 2008;158:590-594.
- 9) Kleinman B, Czinn E, Shah K, Sobotka PA, Rao TK. The value to the anesthesia-surgical care team of the preoperative cardiac consultation. *J Cardiothorac Anesth*. 1989;3:682-687.
- 10) Ferschl MB, Tung A, Sweitzer B, Huo D, Glick DB. Preoperative clinic visits reduce operating room cancellations and delays. *Anesthesiology*. 2005;103:855-859.
- 11) Correll DJ, Bader AM, Hull MW, Hsu C, Tsen LC, Hepner DL. Value of preoperative clinic visits in identifying issues with potential impact on operating room efficiency. *Anesthesiology*. 2006;105:1254-1259; discussion 1256A.
- 12) Dexter F, Marcon E, Epstein RH, Ledolter J. Validation of statistical methods to compare cancellation rates on the day of surgery. *Anesth Analg*. 2005;101:465-473.
- 13) Strum DP, Vargas LG, May JH. Surgical subspecialty block utilization and capacity planning: a minimal cost analysis model. *Anesthesiology*. 1999;90:1176-1185.
- 14) Adam, T., Parenti, C., Gioe, T., Ringsred, K. "Results of a Multidisciplinary Preoperative Assessment Process for High-Risk Orthopedic Patients" *Cleveland Clinic Journal of Medicine* March 2011 vol. 78 Electronic Suppl 1 eS37.
- 15) Falcone, D., and R. Broyles. Access to Long-Term Care: Race as a Barrier. *Journal of Health Politics, Policy and Law* 1994: 583-95.

- 16) Bader AM. Computer-based preoperative assessment. *Int Anesthesiol Clin*. 2002;40:193-199.
- 17) Sandberg EH, Sharma R, Wiklund R, Sandberg WS. Clinicians consistently exceed a typical person's short-term memory during preoperative teaching. *Anesth Analg*. 2008;107:972-978.
- 18) Garg AX, Adhikari NK, McDonald H, et al: Effects of computerized clinical decision support systems on practitioner performance and patient outcomes: a systematic review. *JAMA* 2005; 293:1223–38.
- 19) Wyatt J, Spiegelhalter D: Field trials of medical decision-aids: potential problems and solutions. *Proc Annu Symp Comput Appl Med Care* 1991: 3–7.
- 20) J. Osherooff, J. Teich, B. Middleton, E. Steen, A. Wright, D. Detmer, “A roadmap for national action on clinical decision support,” *J Am Med Inform Assoc* 2007;14(2):141–5.
- 21) Dick R, Steen E, Detmer DE. *The computer-based patient record: An essential technology for health care*, revised edition. Washington, DC: The National Academies Press; 1997.
- 22) Chassin MR, Galvin RW. The urgent need to improve health care quality. Institute of Medicine National Roundtable on Health Care Quality. *JAMA* 1998 Sep 16;280(11):1000-5.
- 23) The Anesthesia Quality Institute (AQI), <http://www.aqihq.com/benefits-aims.pdf> (last accessed Sept 9, 2012).
- 24) Murthy BV, Lake SP, Fisher AC. Evaluation of a decision support system to predict preoperative investigations. *Br J Anaesth*. 2008 Mar;100(3):315-21. Epub 2008 Jan 31.
- 25) HCUP Nationwide Inpatient Sample (NIS). Healthcare Cost and Utilization Project (HCUP). 2004-2008. Agency for Healthcare Research and Quality, Rockville, MD. www.hcup-us.ahrq.gov/nisoverview.jsp
- 26) Danielson D, Bjork K, Card R, Foreman J, Harper C, Roemer R, Stultz J, Sypura W, Thompson S, Webb B. Institute for Clinical Systems Improvement. Preoperative Evaluation. <http://bit.ly.Preop0712>. Updated July 2012.
- 27) Dan Power “What should managers know about business intelligence and data-driven decision support?”, *DSSNews* Vol. 13, No. 15, Aug 2012
- 28) Ahmadian, L., Cornet, R., Kalkman, C., de Keizer, N.F. (2009) Development of a national core dataset for preoperative assessment. *Methods of Information in Medicine* 48:155–161.
- 29) Kaafarani HM, Itani KM, Thornby J, Berger DH. Thirty-day and one-year predictors of death in noncardiac major surgical procedures. *Am J Surg* 2004; 188: 495–499.

- 30) Roizen MF. More preoperative assessment by physicians and less by laboratory tests. *N Engl J Med* 2000; 342: 204–205.
- 31) Halaszynski TM, Juda R, Silverman DG. Optimizing postoperative outcomes with efficient preoperative assessment and management. *Crit Care Med* 2004; 32: S76-S86.
- 32) Epps CD. Length of stay, discharge disposition and hospital charge predictors. *Assoc Operating Room Nurs J.* 2004;79:975–97.
- 33) Archer T, Macario A. The drive for operating room efficiency will increase quality of patient care. *Curr Opin Anaesthesiol* 2006; 19:171–176.
- 34) Balust J, Macario A. Can anesthesia information management systems improve quality in the surgical suite. *Curr Opin Anaesthesiol* 2009;22:215-22.[doi:10.1097/ACO.0b013e328324b9e6](https://doi.org/10.1097/ACO.0b013e328324b9e6).
- 35) Neal ML, Kerckhoffs R. Current progress in patient-specific modeling. *Brief Bioinform.* 2010; 11: 111–126.
- 36) Thune JJ, Hoefsten DE, Lindholm MG. et al. Simple risk stratification at admission to identify patients with reduced mortality from primary angioplasty. *Circulation.* 2005;112(13):2017-2021
- 37) Pantazi, SV., Arocha, JF, Moehr, JR. Case-based medical informatics. *BMC Medical Informatics and Decision Making.* 2004; 4: 19-39.
- 38) Rothwell PM. Treating individuals 1: external validity of randomised controlled trials: to whom do the results of this trial apply? *Lancet.* 2005;365:82-93.
- 39) Davies K. The information-seeking behavior of doctors: a review of the evidence. *Health Info Libr J*2007;24:78-94.
- 40) Daley J, Henderson WG, Khuri SF: Risk-adjusted surgical outcomes. *Ann Rev Med* 2001; 52:275–87.
- 41) H. Coumou, F. Meijman, “How do primary care physicians seek answers to clinical questions? a literature review,” *J Med Libr Assoc.* 2006. Jan; 94(1):55–60.
- 42) Bouamrane, M.M., Rector, A., Hurrell, M.: A Hybrid Architecture for a Preoperative Decision Support System using a Rule Engine and a Reasoner on a Clinical Ontology. In: Polleres, A. (ed.) *RR 2009. LNCS*, vol. 5837, pp. 242–253. Springer, Heidelberg (2009)
- 43) Ricotta JJ, Pagan J, Xenos M, et al. Cardiovascular disease management: the need for better diagnostics. *Med Biol Eng Comput* 2008;46:1059-68.
- 44) Kent DM, Hayward RA. Limitations of applying summary results of clinical trials to individual patients: the need for risk stratification. *JAMA* 2007;298:1209-12.
- 45) Thune JJ, Hoefsten DE, Lindholm MG. et al. Simple risk stratification at admission to identify patients with reduced mortality from primary angioplasty. *Circulation.* 2005;112(13):2017-2021

- 46) Kent DM, Ruthazer R, Selker HP. Are some patients likely to benefit from recombinant tissue-type plasminogen activator for acute ischemic stroke even beyond 3 hours from symptom onset? *Stroke*. 2003;34(2):464-467
- 47) Kent DM, Hayward RA, Griffith JL. et al. An independently derived and validated predictive model for selecting patients with myocardial infarction who are likely to benefit from tissue plasminogen activator compared with streptokinase. *Am J Med*. 2002;113(2):104-111
- 48) Woolf SH, Grol R, Hutchinson A, Eccles M, Grimshaw J. Potential benefits, limitations, and harms of clinical guidelines. *BMJ* 1999;318:527-30.
- 49) Alper BS, White DS, Ge B. Physicians answer more clinical questions and change clinical decisions more often with synthesized evidence: A randomized trial in primary care. *Ann Fam Med* 2005;3:507–513.
- 50) Kelly KJ, Greenblatt DY, Wan Y, Rettammel RJ, Winslow E, Cho CS, et al. Risk stratification for distal pancreatectomy utilizing ACS-NSQIP: preoperative factors predict morbidity and mortality. *J Gastrointest Surg*. 2011;15(2):250-9, discussion.
- 51) Hawn MT, Houston TK, Campagna EJ, Graham LA, Singh J, Bishop M, et al The attributable risk of smoking on surgical complications. *Ann Surg* 2011; 254: 914–20
- 52) Bradley, K. A., Rubinsky, A. D., Sun, H., Bryson, C. L., Bishop, M. J., Blough, D. K., . . . Kivlahan, D. R. (2011). Alcohol screening and risk of postoperative complications in male VA patients undergoing major noncardiac surgery. *Journal of General Internal Medicine*, 26, 162–169.
- 53) CM Clancy, Cronin K. Evidence-Based Decision Making: Global Evidence, Local Decisions. *HealthAff*. 2005;24:151–162
- 54) Wennberg J, Gittelsohn A. Small area variations in health care delivery. *Science* 1973 Dec 14;182(117):1102-8.
- 55) Lau LM, Banning PD, Monson K, Knight E, Wilson PS, Shakib SC. Mapping Department of Defense laboratory results to Logical Observation Identifiers Names and Codes (LOINC); *AMIA AnnuSympProc*; 2005.pp. 430–4.
- 56) Ko CY, Parikh J, Zingmond D. Secondary analyses of large population-based data sets: issues of quality, standards, and understanding. *Ann SurgOncol*. 2008;15:395–396. doi: 10.1245/s10434-007-9657-4.
- 57) Black C, Peterson S, Mansfield J, Thliveris M: Using population-based data to enhance clinical practice guideline development. *Med Care* 1999, 37(6 Suppl):JS254-JS263
- 58) Alper BS, Stevermer JJ, White DS, and Ewigmann BG. Answering family physicians' clinical questions using electronic medical databases. *J Fam Pract*. 2001. Nov; 50(1):960–5.

- 59) Chambliss ML, Conley J. Answering clinical questions. *J Fam Pract.* 1996. Aug; 43(2):140–4.
- 60) S. Khuri, “The NSQIP: a new frontier in surgery,” *Surgery.* 2005;138:837–843.
- 61) S. Khuri, “Daley J, Henderson W; et al, National VA Surgical Quality Improvement Program. The Department of Veterans Affairs' NSQIP: the first national, validated, outcome-based, risk-adjusted, and peer-controlled program for the measurement and enhancement of the quality of surgical care,” *Ann Surg.* 1998;228(4):491-507.
- 62) A. AbuSalah, G. Melton-Meaux, T. Adam, “Patient-Specific Surgical Outcomes Assessment Using Population-Based Data Analysis for Risk Model Development,” *AMIA Annu Symp Proc* 2012, in press.
- 63) Younger P. Internet-based information-seeking behaviour amongst doctors and nurses: a short review of the literature. *Health Info Libr J* 2010 Mar;27(1):2-10.
- 64) J. Dorsch, “Information needs of rural health professionals: a review of the literature,” *Bull Med Lib Assoc.* 2000. Oct; 88(4):346–54.
- 65) J. Ely et. al., “Obstacles to answering doctors' questions about patient care with evidence: qualitative study,” *BMJ.* 2002. Mar 23; 324(7339):710–3.
- 66) D Short, M. Frischer, and J. Bashford, “Barriers to the adoption of computerized decision support systems in general practice consultations: a qualitative study of GPs' perspectives,” *Int J Med Inform.* 2004. May; 73(4):357–62.
- 67) Courtright C., Crawford R. Klubert D. “Criteria for Developing Clinical Decision Support Systems”. *CBMS '01: Proceedings of the Fourteenth IEEE Symposium on Computer-Based Medical Systems*, 2001.
- 68) J. Lewis, “IBM Computer Usability Satisfaction Questionnaires: Psychometric Evaluation and Instructions for Use,” *International Journal of Human–Computer Interaction*, 7 (1) 1995, pp. 57–78.
- 69) Nimmo GR, Croskerry P. Better clinical decision making and reducing diagnostic error. *J R Coll Physicians Edinb* 2011;41 (2):155–62.
- 70) The National Cholesterol Education Programs online heart disease risk assessment tool. Available at: <http://hp2010.nhlbihin.net/atp/iii/calculator.asp?usertype=prof>.
- 71) J. Woo, A. Lai, W. Chung, C. Weng, “Similarity-based Disease Risk Assessment for Personal Genomes: Proof of Concept,” *AMIA Annu Symp Proc.* 2011; 2011:1524-31.
- 72) J. Li et. al., “TLC: an informatics approach to enable patients to initiate tailored lifestyle conversations with providers at the point of care,” *AMIA Annu Symp Proc.* 2009 Nov 14; 2009:364-8. Epub 2009 Nov 14.
- 73) D. Demner-Fushman, C. Seckman, C. Fisher, S. Hauser, J. Clayton, G. Thoma, “A prototype system to support evidence-based practice,” *AMIA Annu Symp Proc* 2008, pp. 151–155.

- 74) Eddy DM, Schlessinger L, Kahn R: Clinical outcomes and cost-effectiveness of strategies for managing people at high risk for diabetes. *Ann Intern Med* 2005, 143:251-264.
- 75) Eddy DM, Schlessinger L: Archimedes: a trial-validated model of diabetes. *Diab Care* 2003, 26:3093-3101.
- 76) Berner ES. Clinical decision support systems: State of the Art. AHRQ Publication No. 09-0069-EF. Rockville, Maryland: Agency for Healthcare Research and Quality. June 2009
- 77) Helmons PJ, Grouls RJE, Roos AN, et al: The potential value on medication safety of a clinical decision support system in intensive care patients with renal insufficiency. *Br J Clin Pharmacol* 2007; 63:504.
- 78) Sim I, Gorman P, Greenes RA, et al. Clinical decision support systems for the practice of evidence-based medicine. *J Am Med Inform Assoc* 2001 Nov;8(6):527-34.
- 79) Bottiger Y, Laine K, Andersson ML, Korhonen T, Molin B, Ovesjö ML, Tirkkonen T, Rane A, Gustafsson LL, Eiermann B. SFINX – a drug–drug interaction database designed for clinical decision support systems. *Eur J Clin Pharmacol* 2009; 65: 627–33.
- 80) Michel, J. Trafton, S. Martins, D. Wang, S. Tu, N. Johnson, M.K. Goldstein Improving patient safety using ATHENA-decision support system technology: the opioid therapy for chronic pain experience K. Henriksen, J.B. Battles, M.A. Keyes, M.L. Grady *Advances in Patient Safety: New Directions and Alternative Approaches (Vol. 4: Technology and Medication Safety)* 2008 Agency for Healthcare Research and Quality (US) Rockville (MD) [Aug].
- 81) Benson M, Junger A, Quinzio L, et al. Clinical and practical requirements of online software for anesthesia documentation an experience report. *Int J Med Inform* 2000;57(2/3):155–64.
- 82) Osheroff JA. Improving medication use and outcomes with clinical decision support: a step-by-step guide. Chicago, IL: The Healthcare Information and Management Systems Society; 2009.
- 83) Power, D. J. *Decision Support Systems: Concepts and Resources for Managers*. Westport, CT: Greenwood/Quorum, 2002.
- 84) Rinott R, Carmeli B, Kent C, Landau D, Maman Y, Rubin Y, Slonim N (2011) Prognostic data-driven clinical decision support-formulation and implications. *Stud Health Technol Inform* 169:140
- 85) Ketan Mane, Chris Bizon, Phillips Owen, Ken Gersing, Javed Mostafa, and Charles Schmitt (2011) Patient Electronic Health Data-Driven Approach to Clinical Decision Support, *Clinical and Translational Science Journal*, 4 (5), pp. 369-371

- 86) Knappe D, Piantavigna S, Hansen A, et al. Oncocin, a novel antibacterial peptide optimized against Gramnegative human pathogens, *J. Medicinal Chemistry*, 2010; 53 (14), 5240–5247.
- 87) An introduction to risk assessment and risk adjustment models, American Medical Association, 2009: 2, <http://www.ama-assn.org/ama1/pub/upload/mm/368/risk-assessment.pdf>, last accessed august 25, 2012
- 88) Warner HR Jr, Bouhaddou O. Innovation review: Iliad--a medical diagnostic support program. *Top Health Inf Manage*. 1994; (4):51-8.
- 89) Chai Z, Zhang X., On Decision Support Systems and It's Application to the Clinical Decisions. the International Conference on Innovation and Information Management (ICIIM 2012) IPCSIT vol. 36 (2012), IACSIT Press, Singapore
- 90) I2B2: Informatics for Integrating Biology and the Bedside. 2008 ; Available from:<https://www.i2b2.org/>
- 91) Weber,GM., Murphy SN., McMurry AJ., et al. The Shared Health Research Information Network (SHRINE): A Prototype Federated Query Tool for Clinical Data Repositories. *JAMIA*. 2009: 16(5):624-630.
- 92) M. Dugas, R. Schauer, A. Volk, H. Rau, "Interactive decision support in hepatic surgery," *BMC Medical Informatics and Decision Making*, 2002: 2, 5.
- 93) A. Wright, D. Sittig, "A framework and model for evaluating clinical decision support architectures," *J Biomed Inform*. 2008 Dec;41(6):982-90.
- 94) Nebeker JR, Hoffman JM, Weir CR, et al. High rates of adverse drug events in a highly computerized hospital. *Arch Intern Med* 2005 May;165(10):1111-6.
- 95) Stead WW, Lin HS. Computational technology for effective health care: immediate steps and strategic directions. Washington, DC: The National Academies Press; 2009.
- 96) Glaser JP, Davenport-Ennis N, Robertson RM,, et al. AHIC April 2008 meeting: clinical decision support recommendation letter. American Health Information Community; 2008 April 22. Available at: http://www.hhs.gov/healthit/documents/m20080422/6.2_cds_recs.html. Accessed August 25, 2012.
- 97) STAT Cardiac Clearance (STAT CC). Available from: <http://www.statcoder.com> (last accessed Sept 13, 2012).
- 98) M.-M. Bouamrane, A. Rector, and M. Hurrell. Experience of using owl ontologies for automated inference of routine pre-operative screening tests. In Proceedings of the 9th international semantic web conference on The semantic web - Volume Part II, ISWC'10, pages 50_65, Berlin, Heidelberg, 2010. Springer-Verlag.
- 99) De Mul M, Berg M, Hazelzet JA. Clinical information systems: CareSuite from Picis. *J Crit Care*2004;19:208-14.

- 100) Carolyn M. Clancy, Let the data be our guide: trends and tools for research on health care utilization. *Health Economics* 21: 19-23 (2012).
- 101) Bederman SS, Coyte PC, Kreder HJ, Mahomed NN, McIsaac WJ, Wright JG. Who's in the driver's seat? The influence of patient and physician enthusiasm on regional variation in degenerative lumbar spinal surgery: a population-based study. *Spine* 2011; 36(6):481-489.
- 102) Devon KM, Urbach DR, McLeod RS. Postoperative disposition and health services use in elderly patients undergoing colorectal cancer surgery: a population-based study. *Surgery*. 2011; 149: 705-712.
- 103) Eastham, J.A. Comparative effectiveness of prostate cancer surgical treatments: a population based analysis of postoperative outcomes. *J. Urol.* 2010; 183, 1366–1372.
- 104) Curtis JR, McClure LA, Delzell E, et al: Population-based fracture risk assessment and osteoporosis treatment disparities by race and gender. *J Gen Intern Med* 24. (8): 956-962.2009;
- 105) Power, D. J. (2007). A Brief History of Decision Support Systems. DSSResources.COM, URL: <http://DSSResources.COM/history/dsshhistory.html>, version 4.0.
- 106) Hayward RA, Kent DM, Vijan S, Hofer TP. Multivariable risk prediction can greatly enhance the statistical power of clinical trial subgroup analysis. *BMC Med Res Methodol.* 2006;618
- 107) Roos NP, Black C, Roos LL, Frohlich N, DeCoster C, Mustard C, Brownell MD, Shanahan M, Fergusson P, Toll F, Carriere KC, Burchill C, Fransoo R, MacWilliam L, Bogdanovic B, Friesen D: Managing health services: how the Population Health Information System (POPULIS) works for policymakers. *Med Care* 1999, 37(6 Suppl):JS27-41.
- 108) de Mul, M., Alons, P., van der Velde, P., Konings, I., Bakker, J., and Hazelzet, J., Development of a clinical data warehouse from an intensive care clinical information system. *Comput. Meth. Programs Biomed.* 105(1):22–30, 2012.
- 109) Frost JH, Massagli MP. Social uses of personal health information within PatientsLikeMe, an online patient community: what can happen when patients have access to one another's data. *J Med Internet Res.* 2008;10(3):e15. doi: 10.2196/jmir.1053
- 110) Lees N, Hall R. Information technology in anaesthesia and critical care. *Contin Educ Anaesth Crit Care Pain* 2011;11:104-7.
- 111) Muravchick S, Caldwell JE, Epstein RH, et al. Anesthesia information management system implementation: a practical guide. *Anesth Analg* 2008;107:1598-608. doi:10.1213/ane.0b013e318187bc8f.
- 112) David Eddy, Marc-david Cohen “Description of the Archimedes Model - ARChES Simulator 2.3”. Technical report available from:

<http://archimedesmodel.com/sites/default/files/Archimedes-Model-Description-ARChES-Simulator-2.3.pdf> Last accessed (Sept 15th, 2012)

- 113) Timmermans S, Mauck A (2005). "The promises and pitfalls of evidence-based medicine". *Health Aff (Millwood)* 24 (1): 18–28. doi:10.1377/hlthaff.24.1.18.PMID 15647212.
- 114) Murthy BV, Lake SP, Fisher AC: Evaluation of a decision support system to predict preoperative investigations. *Br J Anaesth* 2008; 100:315–21.
- 115) Ehrenfeld JM, Rehman MA. Anesthesia information management systems: a review of functionality and installation considerations. *J Clin Monit Comput* 2010.
- 116) HCUP Comorbidity Software. Healthcare Cost and Utilization Project (HCUP). 2000. Agency for Healthcare Research and Quality, Rockville, MD. Available from: www.hcup-us.ahrq.gov/toolssoftware/comorbidity/comorbidity.jsp.
- 117) Elixhauser A, Steiner C, Harris DR, Coffey RM. Comorbidity measures for use with administrative data. *Med Care* 1998;36:8-27.
- 118) Agency for Healthcare Research and Quality (AHRQ). National Healthcare Quality Report, 2011. Agency for Healthcare Research and Quality.
- 119) Agency for Healthcare Research and Quality (AHRQ). National Healthcare Disparities Report, 2011. Agency for Healthcare Research and Quality
- 120) J.S. Hill, J.T. McPhee, G.F. Whalen et al. In-hospital mortality after pancreatic resection for chronic pancreatitis: population-based estimates from the nationwide inpatient sample *J Am Coll Surg*, 209 (2009), pp. 468–476
- 121) Paul S, Nasar A, Port JL, et al. Comparative analysis of diaphragmatic hernia repair outcomes using the Nationwide Inpatient Sample database [published online March 19, 2012]. *Arch Surg*. 2012;;147(7):607-612
- 122) B.L. Hoh, Y.Y. Chi, M.F. Waters, J. Mocco, F.G. Barker II Effect of weekend compared with weekday stroke admission on thrombolytic use, in-hospital mortality, discharge disposition, hospital charges, and length of stay in the Nationwide Inpatient Sample Database, 2002 to 2007
- 123) Geneletti S, Richardson S, Best N. Adjusting for selection bias in retrospective case-control studies. *Biostatistics*. 2008;10(1):17–31
- 124) Sica GT. Bias in research studies. *Radiology* 2006;238(3):780–789.
- 125) Hertzner NR Reasons why data from the nationwide inpatient sample can be misleading for carotid endarterectomy and carotid stenting. *Semin Vasc Surg*. 2012;25:13-7.
- 126) Gorelick MH: Bias arising from missing data in predictive models. *J Clin Epidemiol* 2006, 59:1115-23

- 127) Deyo RA, Nachemson S, Mirza SK. Spinal-fusion surgery—the case for restraint. *N Engl J Med.* 2004;350(7):722-726.
- 128) Memtsoudis SG, Vougioukas VI, Ma Y, et al., Perioperative morbidity and mortality after anterior, posterior, and anterior/posterior spine fusion surgery, *Spine (Phila Pa 1976)*, 2010.
- 129) Yamashita K, Fukusaki M, Ando Y, et al. Preoperative administration of intravenous flurbiprofenaxetil reduces postoperative pain for spinal fusion surgery. *J Anesth*, 2006, 20(2):92- 95.
- 130) A.M. Kotzer. Factors predicting postoperative pain in children and adolescents following spine fusion. *Issues ComprPediaterNurs*, 23 (2000), pp. 83–102.
- 131) Olatilewa O. Awe, Mitchel G. Maltenfort, Srinivas Prasad, et al. Impact of total disc arthroplasty on the surgical management of lumbar degenerative disc disease: Analysis of the Nationwide Inpatient Sample from 2000 to 2008. *SurgNeurol Int.* 2011; 2: 139.
- 132) Cahill KS, Chi JH, Day A, Claus EB. Prevalence, complications, and hospital charges associated with use of bone-morphogenetic proteins in spinal fusion procedures. *JAMA* 2009;302(1):58e66
- 133) Centers for Disease Control and Prevention, National Center for Health Statistics. Conversion table of new ICD-9-CM Codes, Oct 2008 Available at: http://www.cdc.gov/nchs/icd/icd9cm_addenda_guidelines.htm. Accessed Sept 16, 2012
- 134) Truxillo, C. (2005). “Maximum Likelihood Parameter Estimation with Incomplete Data.” Proceedings of the Thirtieth Annual SAS(R) User Group International Conference.
- 135) Yuan , Y. C. (2000). Multiple imputation for missing data: Concepts and new development. In Proceedings of the Twenty-Fifth Annual SAS Users Group International Conference (Paper No. 267). Cary, NC: SAS Institute.
- 136) Allison PD: Handling Missing Data by Maximum Likelihood. SAS Global Forum, 2012:Paper 312-2012.
- 137) Sinharay , S. Stem , H. S. Russell , D. (2001). The use of multiple imputation for the analysis of missing data. *Psychological Methods*, 6, 317-329.
- 138) Schafer, J. L., & Olsen, M. K. (1998). Multiple imputation for multivariate missing-data problems: A data analyst’s perspective. *Multivariate Behavioral Research*, 33(4), 545-571
- 139) Rubin L. H., Witkiewitz K., St. Andre J., Reilly S. (2007). Methods for handling missing data in the behavioural neurosciences: don’t throw the baby rat out with the bath water. *J. Undergrad. Neurosci.* 5, A71–A77.
- 140) Shen Y, Silverstein JC, Roth S. In-hospital complications and mortality after elective spinal fusion surgery in the united states: A study of the nationwide inpatient sample from 2001 to 2005. *J Neurosurg Anesthesiol.* 2009;21:21–30.

- 141) Patil CG, Lad SP, Santarelli J, Boakye M. National Inpatient Complications and Outcomes After Surgery For Spinal Metastasis from 1993-2002. *Cancer*. 2007; Aug 1;110(3):625-30.
- 142) Lad SP, Patil CG, Berta S, Santarelli JG, Ho C, Boakye M. National trends in spinal fusion for cervical spondylotic myelopathy. *SurgNeurol* 2009;71(1):66-9.
- 143) Seymour G. *Predictive Inference: An Introduction*. New York: Chapman & Hall.1993.
- 144) Dreiseitl S, Binder M, Vinterbo S, Kittler H. Applying a decision support system in clinical practice: results from melanoma diagnosis. *AMIA AnnuSympProc* 2007. pp. 191-195.
- 145) Concato J, Peduzzi P, Holford TR, Feinstein AR. The importance of events per independent variable (EPV) in proportional hazards analysis: I. Background, goals and general strategy. *J ClinEpidemiol* 1995; 48: 1495-1501.
- 146) Peduzzi P, Concato J, Kemper E, Holford TR, Feinstein AR. A simulation study of the number of events per variable in logistic regression analysis *J ClinEpidemiol*, 49 (1996), pp. 1373–1379.
- 147) Hanley JA, McNeil BJ. The meaning and use of the area under the Receiver Operating Characteristic (ROC) curve. *Radiology* 1982 143 29-36.
- 148) Birkmeyer JD, Siewers AE, Finlayson EV, et al. Hospital volume and surgical mortality in the United States *N Engl J Med* 2002;346:1128-1137.
- 149) Tokuhashi Y, Matsuzaki H, Oda H, Oshima M, Ryu J. A revised scoring system for preoperative evaluation of metastatic spine tumor prognosis. *Spine*. 2005; 30: 2186–2191.
- 150) RajaeS. S., BaeH. W., KanimL. E.A., and DelamarterR. B., “Spinal fusion in the United States: analysis of trends from 1998 to 2008,” *Spine*, vol. 37, no. 1, pp. 67–76, 2012.
- 151) Faciszewski T, Winter RB, Lonstein JE, Denis F, Johnson L. The surgical and medical perioperative complications of anterior spinal fusion surgery in the thoracic and lumbar spine in adults. *Spine* 1995;20:1592-99.
- 152) Coe JD, et. al. Complications of spinal fusion for Scheuermann kyphosis: a report of the Scoliosis Research Society Morbidity and Mortality Committee. *Spine (Phila Pa 1976)*. 2009;35:99-103.
- 153) Arnott, D. and G. Pervan, A critical analysis of decision support systems research, *Journal of Information Technology*, 20, 2, 2005, 67-87.
- 154) Boyd CM, Darer J, Boulton C, Fried LP, Boulton L, Wu AW. Clinical practice guidelines and quality of care for older patients with multiple comorbid diseases: implications for pay for performance. *JAMA*.2005 Aug 10;294(6):716–24.

- 155) D.F. Sittig, A. Wright, J.A. Osherooff, B. Middleton, J.M. Teich, J.S. Ash et al. Grand challenges in clinical decision support J Biomed Inform, 41 (2) (2008), pp. 387–392.
- 156) L. Knutsen, “INFORMIX DBA: Building Fast Data Warehouse Schemas,” IBM Data Management Magazine. Oct, 2010.
- 157) A. Ketefian, J. Hu, A. Bartolucci, R. Azziz, “Fifteen-year trend in the use of reproductive surgery in women in the United States,” Fertil Steril. 2008 Sep 26.
- 158) E Graves, “Detailed diagnoses and procedures, National Hospital Discharge Survey, 1993,” Vital Health Stat Ser 13, 122, 1995, pp. 1–288.
- 159) Wachter RM .Expected and unanticipated consequences of the quality and information technology revolutions.JAMA2006;295:2780–2783
- 160) Rittenhouse DR, Shortell SM. The patient-centered medical home: will it stand the test of health reform? Jama. 2009;301(19):2038–40.
- 161) E. Rahm et al. Data cleaning: Problems and current approaches. IEEE Data Eng. Bull., 23(4):3–13, 2000.
- 162) Friedman C, Hripcsak G, Johnson S, Cimino J, Clayton P A generalized relational schema for an integrated clinical patient database. Proc 14th Annu Symp Comput Appl Med Care 1990:335–9.
- 163) Hani N. et al. Toward personalized care management of patients at risk: the diabetes case study. KDD 2011: 395-403.
- 164) A. Wright, D. Sittig, “SANDS: a service-oriented architecture for clinical decision support in a National Health Information Network,” J Biomed Inform. 2008 Dec;41(6):962–81.
- 165) C. Overby et. al., “An Evaluation of Functional and User Interface Requirements for Pharmacogenomic Clinical Decision Support,” HISB 2011: 134-141.
- 166) Carroll C et. al. Involving users in the design and usability evaluation of a clinical decision support system. Comput methods Programs Biomed 2002;69(2):123-135.
- 167) Peiris DP, Joshi R, Webster RJ, et al. An electronic clinical decision support tool to assist primary care providers in cardiovascular disease risk management: development and mixed methods evaluation.J Med Internet Res. 2009;11(4):e51.
- 168) J. Saleem, et al, “ Impact of clinical reminder redesign on learnability, efficiency, usability, and workload for ambulatory clinic nurses,” J Am Med Inform Assoc 2007;14(5):632–40.
- 169) K. Zheng, R. Padman, M. Johnson, H. Diamond, “Understanding technology adoption in clinical care: clinician adoption behavior of a point-of-care reminder system,” Int J Med Inform 2005;74:535-543.

170) T. Dillon, T. Ratcliffe, "Evaluating point-of-care technology with the IBM computer usability satisfaction questionnaire," *Issues in Information Systems*, 5 (2) 2004, pp. 441–446.

171) J. Nielsen What is usability? J. Nielsen (Ed.), *Usability Engineering*, Academic Press (1993) chapter 2.

172) K. Dansky, L. Gamm, J. Vasey, C. Barsukiewicz, "Electronic medical records: are physicians ready?," *Journal of Healthcare Management* 1999; 44 (6); 440–54.

173) D. Forsythe, "Using ethnography to build a working system: rethinking basic design assumptions," *Proceedings of the 15th Annual Symposium on Computer Applications in Medical Care*, American Medical Informatics Association, 505–9. New York: McGraw-Hill, 1993.

APPENDIX A : Source Code

A1. POET's interface, and Modules for Query builder and dispatcher.

```
/*
```

JAVA Source Code to create POET's interface, and Query builder and dispatcher modules.

Author: Ahmad AbuSalah

```
*/
```

```
/*
```

```
* POETView.java
```

```
*/
```

```
package poet;
```

```
import java.sql.SQLException;
import java.util.logging.Level;
import java.util.logging.Logger;
import org.jdesktop.application.Action;
import org.jdesktop.application.ResourceMap;
import org.jdesktop.application.SingleFrameApplication;
import org.jdesktop.application.FrameView;
import org.jdesktop.application.TaskMonitor;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;
import java.util.ArrayList;
import javax.swing.Timer;
import javax.swing.Icon;
import javax.swing.JDialog;
import javax.swing.JFrame;
import javax.swing.JOptionPane;
```

```
/**
```

```
* The application's main frame.
```

```
*/
```

```
public class POETView extends FrameView {
```



```
String url = "";

String pass_str = "";

Connection conn ;

//String primary_proc="";
//String sec_pr1="";
//String sec_pr2="";
//String sec_pr3="";
//String nis_code="";
String sec_pr1_nis_code="";
String sec_pr2_nis_code="";
String sec_pr3_nis_code="";
String primary_nis_code="";
String nis_query="";

String pat_age = "";
String pat_race="";
String pat_gender="";
String weekend_adm="";
String elective_adm="";
String hosp_teach="";
String hosp_loc="";
String hosp_region="";
String bed_size="";
String CM_AIDS="";
String CM_ALCOHOL="";
String CM_ANEMDEF="";
String CM_ARTH="";
String CM_BLDLOSS="";
String CM_CHF="";
String CM_CHRNLUNG="";
String CM_COAG="";
String CM_DEPRESS="";
String CM_DM="";
String CM_DMCX="";
String CM_DRUG="";
String CM_HTN_C="";
String CM_HYPOTHY="";
String CM_LIVER="";
String CM_LYMPH="";
String CM_LYTES="";
String CM_METS="";
String CM_NEURO="";
String CM_OBESE="";
String CM_PARA="";
```

```

String CM_PERIVASC="";
String CM_PSYCH="";
String CM_PULMCIRC="";
String CM_RENLFAIL="";
String CM_TUMOR="";
String CM_ULCER="";
String CM_VALVE="";
String CM_WGHTLOSS="";
String pat_como="";
String q_pr1="";
String q_pr2="";
String q_pr3="";
String q_dx1="";

public POETView(SingleFrameApplication app) {
    super(app);

    initComponents();
    jTextField1.setVisible(false);
    jTextField2.setVisible(false);
    jTextField4.setVisible(false);
    jTextField5.setVisible(false);
    jTextField6.setVisible(false);
    jTextField7.setVisible(false);
    jTextField14.setVisible(false);
    jTextField15.setVisible(false);
    jTextField16.setVisible(false);
    jTextField17.setVisible(false);

    jLabel3.setVisible(false);
    jLabel7.setVisible(false);

    //    jLabel27.setVisible(false);

    //    jLabel28.setVisible(false);

    //    jLabel29.setVisible(false);

    //    jLabel30.setVisible(false);
    jButton4.setVisible(false);
    jButton5.setVisible(false);
    jButton9.setVisible(false);
    jButton10.setVisible(false);
    //    AutoCompletion.enable(jComboBox4);
    //    AutoCompletion.enable(jComboBox5);
    //    AutoCompletion.enable(jComboBox6);
    //    AutoCompletion.enable(jComboBox7);
    buttonGroup1.add(jRadioButton1);

```

```

buttonGroup1.add(jRadioButton2);
buttonGroup2.add(jRadioButton3);
buttonGroup2.add(jRadioButton4);
buttonGroup2.add(jRadioButton5);
buttonGroup2.add(jRadioButton6);
buttonGroup3.add(jRadioButton7);
buttonGroup3.add(jRadioButton8);
buttonGroup3.add(jRadioButton9);
buttonGroup4.add(jRadioButton10);
buttonGroup4.add(jRadioButton12);
buttonGroup5.add(jRadioButton15);
buttonGroup5.add(jRadioButton16);
buttonGroup6.add(jRadioButton17);
buttonGroup6.add(jRadioButton18);
buttonGroup9.add(jRadioButton11);
buttonGroup9.add(jRadioButton13);
buttonGroup1.add(jRadioButton14);
buttonGroup1.add(jRadioButton19);
buttonGroup10.add(jRadioButton26);
buttonGroup10.add(jRadioButton27);
buttonGroup11.add(jRadioButton28);
buttonGroup11.add(jRadioButton29);

// status bar initialization - message timeout, idle icon and busy animation, etc
ResourceMap resourceMap = getResourceMap();
int messageTimeout = resourceMap.getInteger("StatusBar.messageTimeout");
messageTimer = new Timer(messageTimeout, new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        statusMessageLabel.setText("");
    }
});
messageTimer.setRepeats(false);
int busyAnimationRate = resourceMap.getInteger("StatusBar.busyAnimationRate");
for (int i = 0; i < busyIcons.length; i++) {
    busyIcons[i] = resourceMap.getIcon("StatusBar.busyIcons[" + i + "]");
}
busyIconTimer = new Timer(busyAnimationRate, new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        busyIconIndex = (busyIconIndex + 1) % busyIcons.length;
        statusAnimationLabel.setIcon(busyIcons[busyIconIndex]);
    }
});
idleIcon = resourceMap.getIcon("StatusBar.idleIcon");
statusAnimationLabel.setIcon(idleIcon);
progressBar.setVisible(false);

// connecting action tasks to status bar via TaskMonitor
TaskMonitor taskMonitor = new TaskMonitor(getApplication().getContext());

```

```

taskMonitor.addPropertyChangeListener(new java.beans.PropertyChangeListener()
{
    public void propertyChange(java.beans.PropertyChangeEvent evt) {
        String propertyName = evt.getPropertyName();
        if ("started".equals(propertyName)) {
            if (!busyIconTimer.isRunning()) {
                statusAnimationLabel.setIcon(busyIcons[0]);
                busyIconIndex = 0;
                busyIconTimer.start();
            }
            progressBar.setVisible(true);
            progressBar.setIndeterminate(true);
        } else if ("done".equals(propertyName)) {
            busyIconTimer.stop();
            statusAnimationLabel.setIcon(idleIcon);
            progressBar.setVisible(false);
            progressBar.setValue(0);
        } else if ("message".equals(propertyName)) {
            String text = (String)(evt.getNewValue());
            statusMessageLabel.setText((text == null) ? "" : text);
            messageTimer.restart();
        } else if ("progress".equals(propertyName)) {
            int value = (Integer)(evt.getNewValue());
            progressBar.setVisible(true);
            progressBar.setIndeterminate(false);
            progressBar.setValue(value);
        }
    }
});
}

```

```

@Action
public void showAboutBox() {
    if (aboutBox == null) {
        JFrame mainFrame = POETApp.getApplication().getMainFrame();
        aboutBox = new POETAboutBox(mainFrame);
        aboutBox.setLocationRelativeTo(mainFrame);
    }
    POETApp.getApplication().show(aboutBox);
}

```

```

/** This method is called from within the constructor to
 * initialize the form.
 * WARNING: Do NOT modify this code. The content of this method is
 * always regenerated by the Form Editor.
 */

```

```

@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">

```

```
private void initComponents() {  
  
    mainPanel = new javax.swing.JPanel();  
    jLabel3 = new javax.swing.JLabel();  
    jComboBox1 = new javax.swing.JComboBox();  
    jLabel4 = new javax.swing.JLabel();  
    jComboBox2 = new javax.swing.JComboBox();  
    jLabel5 = new javax.swing.JLabel();  
    jComboBox3 = new javax.swing.JComboBox();  
    jTextField1 = new javax.swing.JTextField();  
    jTextField2 = new javax.swing.JTextField();  
    jLabel6 = new javax.swing.JLabel();  
    jLabel7 = new javax.swing.JLabel();  
    jLabel9 = new javax.swing.JLabel();  
    jLabel12 = new javax.swing.JLabel();  
    jLabel13 = new javax.swing.JLabel();  
    jLabel14 = new javax.swing.JLabel();  
    jLabel15 = new javax.swing.JLabel();  
    jLabel16 = new javax.swing.JLabel();  
    jLabel18 = new javax.swing.JLabel();  
    jRadioButton1 = new javax.swing.JRadioButton();  
    jRadioButton2 = new javax.swing.JRadioButton();  
    jLabel20 = new javax.swing.JLabel();  
    jRadioButton3 = new javax.swing.JRadioButton();  
    jRadioButton4 = new javax.swing.JRadioButton();  
    jRadioButton5 = new javax.swing.JRadioButton();  
    jRadioButton6 = new javax.swing.JRadioButton();  
    jLabel21 = new javax.swing.JLabel();  
    jRadioButton7 = new javax.swing.JRadioButton();  
    jRadioButton8 = new javax.swing.JRadioButton();  
    jRadioButton9 = new javax.swing.JRadioButton();  
    jLabel22 = new javax.swing.JLabel();  
    jRadioButton10 = new javax.swing.JRadioButton();  
    jRadioButton12 = new javax.swing.JRadioButton();  
    jLabel23 = new javax.swing.JLabel();  
    jLabel24 = new javax.swing.JLabel();  
    jLabel25 = new javax.swing.JLabel();  
    jRadioButton15 = new javax.swing.JRadioButton();  
    jRadioButton16 = new javax.swing.JRadioButton();  
    jRadioButton17 = new javax.swing.JRadioButton();  
    jRadioButton18 = new javax.swing.JRadioButton();  
    jScrollPane18 = new javax.swing.JScrollPane();  
    jList14 = new javax.swing.JList();  
    jPanel1 = new javax.swing.JPanel();  
    jButton1 = new javax.swing.JButton();  
    jLabel2 = new javax.swing.JLabel();  
    jLabel8 = new javax.swing.JLabel();  
    jLabel17 = new javax.swing.JLabel();  
}
```

```
jButton2 = new javax.swing.JButton();
jTextField3 = new javax.swing.JTextField();
jLabel1 = new javax.swing.JLabel();
jLabel26 = new javax.swing.JLabel();
jPasswordField1 = new javax.swing.JPasswordField();
jButton3 = new javax.swing.JButton();
jRadioButton11 = new javax.swing.JRadioButton();
jRadioButton13 = new javax.swing.JRadioButton();
jTextField4 = new javax.swing.JTextField();
jTextField5 = new javax.swing.JTextField();
jButton4 = new javax.swing.JButton();
jRadioButton14 = new javax.swing.JRadioButton();
jRadioButton19 = new javax.swing.JRadioButton();
jTextField6 = new javax.swing.JTextField();
jTextField7 = new javax.swing.JTextField();
jButton5 = new javax.swing.JButton();
jLabel19 = new javax.swing.JLabel();
jLabel27 = new javax.swing.JLabel();
jScrollPane19 = new javax.swing.JScrollPane();
jList15 = new javax.swing.JList();
jRadioButton26 = new javax.swing.JRadioButton();
jTextField14 = new javax.swing.JTextField();
jTextField15 = new javax.swing.JTextField();
jRadioButton27 = new javax.swing.JRadioButton();
jButton9 = new javax.swing.JButton();
jRadioButton28 = new javax.swing.JRadioButton();
jTextField16 = new javax.swing.JTextField();
jTextField17 = new javax.swing.JTextField();
jRadioButton29 = new javax.swing.JRadioButton();
jButton10 = new javax.swing.JButton();
jLabel10 = new javax.swing.JLabel();
menuBar = new javax.swing.JMenuBar();
javax.swing.JMenu fileMenu = new javax.swing.JMenu();
javax.swing.JMenuItem exitMenuItem = new javax.swing.JMenuItem();
javax.swing.JMenu helpMenu = new javax.swing.JMenu();
javax.swing.JMenuItem aboutMenuItem = new javax.swing.JMenuItem();
statusPanel = new javax.swing.JPanel();
javax.swing.JSeparator statusPanelSeparator = new javax.swing.JSeparator();
statusMessageLabel = new javax.swing.JLabel();
statusAnimationLabel = new javax.swing.JLabel();
progressBar = new javax.swing.JProgressBar();
buttonGroup2 = new javax.swing.ButtonGroup();
buttonGroup3 = new javax.swing.ButtonGroup();
buttonGroup4 = new javax.swing.ButtonGroup();
buttonGroup5 = new javax.swing.ButtonGroup();
buttonGroup6 = new javax.swing.ButtonGroup();
buttonGroup7 = new javax.swing.ButtonGroup();
buttonGroup8 = new javax.swing.ButtonGroup();
```

```

buttonGroup9 = new javax.swing.ButtonGroup();
buttonGroup1 = new javax.swing.ButtonGroup();
buttonGroup10 = new javax.swing.ButtonGroup();
buttonGroup11 = new javax.swing.ButtonGroup();

mainPanel.setMaximumSize(new java.awt.Dimension(1000, 1000));
mainPanel.setName("mainPanel"); // NOI18N
mainPanel.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

org.jdesktop.application.ResourceMap resourceMap =
org.jdesktop.application.Application.getInstance(poet.POETApp.class).getContext().getR
esourceMap(poetView.class);
jLabel3.setFont(resourceMap.getFont("jLabel3.font")); // NOI18N
jLabel3.setText(resourceMap.getString("jLabel3.text")); // NOI18N
jLabel3.setName("jLabel3"); // NOI18N
mainPanel.add(jLabel3, new org.netbeans.lib.awtextra.AbsoluteConstraints(280,
130, 20, 20));

jComboBox1.setModel(new javax.swing.DefaultComboBoxModel(new String[] { "0
to 10", "11 to 20", "21 to 30", "31 to 40", "41 to 50", "51 to 60", "61 to 70", "71 to 80",
"81 to 90", "Above 90", "Custom Age Range" }));
jComboBox1.setName("jComboBox1"); // NOI18N
jComboBox1.addItemListener(new java.awt.event.ItemListener() {
    public void itemStateChanged(java.awt.event.ItemEvent evt) {
        jComboBox1ItemStateChanged(evt);
    }
});
mainPanel.add(jComboBox1, new
org.netbeans.lib.awtextra.AbsoluteConstraints(50, 130, 140, -1));

jLabel4.setFont(resourceMap.getFont("jLabel4.font")); // NOI18N
jLabel4.setText(resourceMap.getString("jLabel4.text")); // NOI18N
jLabel4.setName("jLabel4"); // NOI18N
mainPanel.add(jLabel4, new org.netbeans.lib.awtextra.AbsoluteConstraints(50,
160, 40, -1));

jComboBox2.setModel(new javax.swing.DefaultComboBoxModel(new String[] {
"Black", "Hispanic", "White", "Other" }));
jComboBox2.setName("jComboBox2"); // NOI18N
mainPanel.add(jComboBox2, new
org.netbeans.lib.awtextra.AbsoluteConstraints(50, 180, 140, -1));

jLabel5.setFont(resourceMap.getFont("jLabel5.font")); // NOI18N
jLabel5.setText(resourceMap.getString("jLabel5.text")); // NOI18N
jLabel5.setName("jLabel5"); // NOI18N
mainPanel.add(jLabel5, new org.netbeans.lib.awtextra.AbsoluteConstraints(220,
160, -1, -1));

```

```
    JComboBox3.setModel(new javax.swing.DefaultComboBoxModel(new String[] {
"Female", "Male" }));
    JComboBox3.setName("jComboBox3"); // NOI18N
    mainPanel.add(jComboBox3, new
org.netbeans.lib.awtextra.AbsoluteConstraints(220, 180, 140, -1));

    JTextField1.setText(resourceMap.getString("jTextField1.text")); // NOI18N
    JTextField1.setName("jTextField1"); // NOI18N
    mainPanel.add(jTextField1, new
org.netbeans.lib.awtextra.AbsoluteConstraints(320, 130, 40, -1));

    JTextField2.setText(resourceMap.getString("jTextField2.text")); // NOI18N
    JTextField2.setName("jTextField2"); // NOI18N
    mainPanel.add(jTextField2, new
org.netbeans.lib.awtextra.AbsoluteConstraints(220, 130, 40, -1));

    JLabel6.setFont(resourceMap.getFont("jLabel6.font")); // NOI18N
    JLabel6.setText(resourceMap.getString("jLabel6.text")); // NOI18N
    JLabel6.setName("jLabel6"); // NOI18N
    mainPanel.add(jLabel6, new org.netbeans.lib.awtextra.AbsoluteConstraints(50,
110, -1, -1));

    JLabel7.setFont(resourceMap.getFont("jLabel7.font")); // NOI18N
    JLabel7.setText(resourceMap.getString("jLabel7.text")); // NOI18N
    JLabel7.setName("jLabel7"); // NOI18N
    mainPanel.add(jLabel7, new org.netbeans.lib.awtextra.AbsoluteConstraints(220,
110, 120, -1));

    JLabel9.setFont(resourceMap.getFont("jLabel9.font")); // NOI18N
    JLabel9.setText(resourceMap.getString("jLabel9.text")); // NOI18N
    JLabel9.setName("jLabel9"); // NOI18N
    mainPanel.add(jLabel9, new org.netbeans.lib.awtextra.AbsoluteConstraints(50,
240, 120, -1));

    JLabel12.setFont(resourceMap.getFont("jLabel12.font")); // NOI18N
    JLabel12.setText(resourceMap.getString("jLabel12.text")); // NOI18N
    JLabel12.setName("jLabel12"); // NOI18N
    mainPanel.add(jLabel12, new org.netbeans.lib.awtextra.AbsoluteConstraints(60,
600, 60, 20));

    JLabel13.setFont(resourceMap.getFont("jLabel13.font")); // NOI18N
    JLabel13.setForeground(resourceMap.getColor("jLabel13.foreground")); // NOI18N
    JLabel13.setText(resourceMap.getString("jLabel13.text")); // NOI18N
    JLabel13.setName("jLabel13"); // NOI18N
    mainPanel.add(jLabel13, new org.netbeans.lib.awtextra.AbsoluteConstraints(30,
90, 340, -1));

    JLabel14.setFont(resourceMap.getFont("jLabel14.font")); // NOI18N
```



```
jLabel14.setText(resourceMap.getString("jLabel14.text")); // NOI18N
jLabel14.setName("jLabel14"); // NOI18N
mainPanel.add(jLabel14, new org.netbeans.lib.awtextra.AbsoluteConstraints(10,
10, 550, 30));

jLabel15.setFont(resourceMap.getFont("jLabel15.font")); // NOI18N
jLabel15.setText(resourceMap.getString("jLabel15.text")); // NOI18N
jLabel15.setName("jLabel15"); // NOI18N
mainPanel.add(jLabel15, new org.netbeans.lib.awtextra.AbsoluteConstraints(340,
40, 210, 30));

jLabel16.setFont(resourceMap.getFont("jLabel16.font")); // NOI18N
jLabel16.setForeground(resourceMap.getColor("jLabel16.foreground")); // NOI18N
jLabel16.setText(resourceMap.getString("jLabel16.text")); // NOI18N
jLabel16.setName("jLabel16"); // NOI18N
mainPanel.add(jLabel16, new org.netbeans.lib.awtextra.AbsoluteConstraints(450,
230, 150, -1));

jLabel18.setFont(resourceMap.getFont("jLabel18.font")); // NOI18N
jLabel18.setForeground(resourceMap.getColor("jLabel18.foreground")); // NOI18N
jLabel18.setText(resourceMap.getString("jLabel18.text")); // NOI18N
jLabel18.setName("jLabel18"); // NOI18N
mainPanel.add(jLabel18, new org.netbeans.lib.awtextra.AbsoluteConstraints(30,
220, 240, -1));

jRadioButton1.setText(resourceMap.getString("jRadioButton1.text")); // NOI18N
jRadioButton1.setName("jRadioButton1"); // NOI18N
mainPanel.add(jRadioButton1, new
org.netbeans.lib.awtextra.AbsoluteConstraints(60, 640, -1, -1));

jRadioButton2.setText(resourceMap.getString("jRadioButton2.text")); // NOI18N
jRadioButton2.setName("jRadioButton2"); // NOI18N
mainPanel.add(jRadioButton2, new
org.netbeans.lib.awtextra.AbsoluteConstraints(60, 620, -1, -1));

jLabel20.setFont(resourceMap.getFont("jLabel20.font")); // NOI18N
jLabel20.setText(resourceMap.getString("jLabel20.text")); // NOI18N
jLabel20.setName("jLabel20"); // NOI18N
mainPanel.add(jLabel20, new org.netbeans.lib.awtextra.AbsoluteConstraints(180,
600, 60, 20));

jRadioButton3.setText(resourceMap.getString("jRadioButton3.text")); // NOI18N
jRadioButton3.setName("jRadioButton3"); // NOI18N
mainPanel.add(jRadioButton3, new
org.netbeans.lib.awtextra.AbsoluteConstraints(180, 620, -1, -1));

jRadioButton4.setText(resourceMap.getString("jRadioButton4.text")); // NOI18N
jRadioButton4.setName("jRadioButton4"); // NOI18N
```

```
mainPanel.add(jRadioButton4, new
org.netbeans.lib.awtextra.AbsoluteConstraints(180, 640, -1, -1));

jRadioButton5.setText(resourceMap.getString("jRadioButton5.text")); // NOI18N
jRadioButton5.setName("jRadioButton5"); // NOI18N
mainPanel.add(jRadioButton5, new
org.netbeans.lib.awtextra.AbsoluteConstraints(260, 620, -1, -1));

jRadioButton6.setText(resourceMap.getString("jRadioButton6.text")); // NOI18N
jRadioButton6.setName("jRadioButton6"); // NOI18N
mainPanel.add(jRadioButton6, new
org.netbeans.lib.awtextra.AbsoluteConstraints(260, 640, -1, -1));

jLabel21.setFont(resourceMap.getFont("jLabel21.font")); // NOI18N
jLabel21.setText(resourceMap.getString("jLabel21.text")); // NOI18N
jLabel21.setName("jLabel21"); // NOI18N
mainPanel.add(jLabel21, new org.netbeans.lib.awtextra.AbsoluteConstraints(410,
600, 60, 20));

jRadioButton7.setText(resourceMap.getString("jRadioButton7.text")); // NOI18N
jRadioButton7.setName("jRadioButton7"); // NOI18N
mainPanel.add(jRadioButton7, new
org.netbeans.lib.awtextra.AbsoluteConstraints(410, 640, -1, -1));

jRadioButton8.setText(resourceMap.getString("jRadioButton8.text")); // NOI18N
jRadioButton8.setName("jRadioButton8"); // NOI18N
mainPanel.add(jRadioButton8, new
org.netbeans.lib.awtextra.AbsoluteConstraints(470, 620, -1, -1));

jRadioButton9.setText(resourceMap.getString("jRadioButton9.text")); // NOI18N
jRadioButton9.setName("jRadioButton9"); // NOI18N
mainPanel.add(jRadioButton9, new
org.netbeans.lib.awtextra.AbsoluteConstraints(410, 620, -1, -1));

jLabel22.setFont(resourceMap.getFont("jLabel22.font")); // NOI18N
jLabel22.setText(resourceMap.getString("jLabel22.text")); // NOI18N
jLabel22.setName("jLabel22"); // NOI18N
mainPanel.add(jLabel22, new org.netbeans.lib.awtextra.AbsoluteConstraints(570,
600, 110, 20));

jRadioButton10.setText(resourceMap.getString("jRadioButton10.text")); // NOI18N
jRadioButton10.setName("jRadioButton10"); // NOI18N
mainPanel.add(jRadioButton10, new
org.netbeans.lib.awtextra.AbsoluteConstraints(570, 640, -1, -1));

jRadioButton12.setText(resourceMap.getString("jRadioButton12.text")); // NOI18N
jRadioButton12.setName("jRadioButton12"); // NOI18N
```

```

    mainPanel.add(jRadioButton12, new
org.netbeans.lib.awtextra.AbsoluteConstraints(570, 620, -1, -1));

    jLabel23.setFont(resourceMap.getFont("jLabel23.font")); // NOI18N
    jLabel23.setForeground(resourceMap.getColor("jLabel23.foreground")); // NOI18N
    jLabel23.setText(resourceMap.getString("jLabel23.text")); // NOI18N
    jLabel23.setName("jLabel23"); // NOI18N
    mainPanel.add(jLabel23, new org.netbeans.lib.awtextra.AbsoluteConstraints(40,
580, 190, -1));

    jLabel24.setFont(resourceMap.getFont("jLabel24.font")); // NOI18N
    jLabel24.setText(resourceMap.getString("jLabel24.text")); // NOI18N
    jLabel24.setName("jLabel24"); // NOI18N
    mainPanel.add(jLabel24, new org.netbeans.lib.awtextra.AbsoluteConstraints(400,
130, 130, 20));

    jLabel25.setFont(resourceMap.getFont("jLabel25.font")); // NOI18N
    jLabel25.setText(resourceMap.getString("jLabel25.text")); // NOI18N
    jLabel25.setName("jLabel25"); // NOI18N
    mainPanel.add(jLabel25, new org.netbeans.lib.awtextra.AbsoluteConstraints(550,
130, 110, 20));

    jRadioButton15.setText(resourceMap.getString("jRadioButton15.text")); // NOI18N
    jRadioButton15.setName("jRadioButton15"); // NOI18N
    mainPanel.add(jRadioButton15, new
org.netbeans.lib.awtextra.AbsoluteConstraints(550, 170, -1, -1));

    jRadioButton16.setText(resourceMap.getString("jRadioButton16.text")); // NOI18N
    jRadioButton16.setName("jRadioButton16"); // NOI18N
    mainPanel.add(jRadioButton16, new
org.netbeans.lib.awtextra.AbsoluteConstraints(550, 150, -1, -1));

    jRadioButton17.setText(resourceMap.getString("jRadioButton17.text")); // NOI18N
    jRadioButton17.setName("jRadioButton17"); // NOI18N
    mainPanel.add(jRadioButton17, new
org.netbeans.lib.awtextra.AbsoluteConstraints(400, 170, -1, -1));

    jRadioButton18.setText(resourceMap.getString("jRadioButton18.text")); // NOI18N
    jRadioButton18.setName("jRadioButton18"); // NOI18N
    mainPanel.add(jRadioButton18, new
org.netbeans.lib.awtextra.AbsoluteConstraints(400, 150, -1, -1));

    jScrollPane18.setName("jScrollPane18"); // NOI18N

    jList14.setModel(new javax.swing.AbstractListModel() {
        String[] strings = { "Acquired immune deficiency syndrome (AIDS)", "Alcohol
abuse", "Chronic blood loss anemia", "Chronic pulmonary disease", "Coagulopathy",
"Congestive heart failure", "Deficiency anemias", "Depression", "Diabetes with chronic

```

```

complications", "Diabetes, uncomplicated", "Drug abuse", "Fluid and electrolyte
disorders", "Hypertension (combine uncomplicated and complicated)",
"Hypothyroidism", "Liver disease", "Lymphoma", "Metastatic cancer", "Obesity", "Other
neurological disorders", "Paralysis", "Peptic ulcer disease excluding bleeding",
"Peripheral vascular disorders", "Psychoses", "Pulmonary circulation disorders", "Renal
failure", "Rheumatoid arthritis/collagen vascular diseases", "Solid tumor without
metastasis", "Valvular disease", "Weight loss" };
    public int getSize() { return strings.length; }
    public Object getElementAt(int i) { return strings[i]; }
});
jList14.setToolTipText(resourceMap.getString("jList14.toolTipText")); // NOI18N
jList14.setName("jList14"); // NOI18N
jList14.addListSelectionListener(new javax.swing.event.ListSelectionListener() {
    public void valueChanged(javax.swing.event.ListSelectionEvent evt) {
        jList14ValueChanged(evt);
    }
});
jScrollPane18.setViewportView(jList14);

mainPanel.add(jScrollPane18, new
org.netbeans.lib.awtextra.AbsoluteConstraints(450, 260, 300, 290));

jPanel1.setBorder(new
javax.swing.border.SoftBevelBorder(javax.swing.border.BevelBorder.RAISED));
jPanel1.setName("jPanel1"); // NOI18N
jPanel1.setLayout(new org.netbeans.lib.awtextra.AbsoluteLayout());

jButton1.setFont(resourceMap.getFont("jButton1.font")); // NOI18N
jButton1.setForeground(resourceMap.getColor("jButton1.foreground")); // NOI18N
jButton1.setText(resourceMap.getString("jButton1.text")); // NOI18N
jButton1.setName("jButton1"); // NOI18N
jButton1.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton1ActionPerformed(evt);
    }
});
jPanel1.add(jButton1, new org.netbeans.lib.awtextra.AbsoluteConstraints(100, 30,
130, 40));

jLabel2.setText(resourceMap.getString("jLabel2.text")); // NOI18N
jLabel2.setName("jLabel2"); // NOI18N
jPanel1.add(jLabel2, new org.netbeans.lib.awtextra.AbsoluteConstraints(120, 80,
110, 20));

jLabel8.setText(resourceMap.getString("jLabel8.text")); // NOI18N
jLabel8.setName("jLabel8"); // NOI18N
jPanel1.add(jLabel8, new org.netbeans.lib.awtextra.AbsoluteConstraints(10, 10,
110, 20));

```

```

jLabel17.setText(resourceMap.getString("jLabel17.text")); // NOI18N
jLabel17.setName("jLabel17"); // NOI18N
jPanel1.add(jLabel17, new org.netbeans.lib.awtextra.AbsoluteConstraints(240, 10,
90, -1));

mainPanel.add(jPanel1, new org.netbeans.lib.awtextra.AbsoluteConstraints(190,
670, 330, 120));

jButton2.setFont(resourceMap.getFont("jButton2.font")); // NOI18N
jButton2.setForeground(resourceMap.getColor("jButton2.foreground")); // NOI18N
jButton2.setText(resourceMap.getString("jButton2.text")); // NOI18N
jButton2.setName("jButton2"); // NOI18N
jButton2.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton2ActionPerformed(evt);
    }
});
mainPanel.add(jButton2, new org.netbeans.lib.awtextra.AbsoluteConstraints(650,
710, 100, 40));

jTextField3.setText(resourceMap.getString("jTextField3.text")); // NOI18N
jTextField3.setName("jTextField3"); // NOI18N
mainPanel.add(jTextField3, new
org.netbeans.lib.awtextra.AbsoluteConstraints(610, 20, 130, -1));

jLabel1.setFont(resourceMap.getFont("jLabel1.font")); // NOI18N
jLabel1.setText(resourceMap.getString("jLabel1.text")); // NOI18N
jLabel1.setName("jLabel1"); // NOI18N
mainPanel.add(jLabel1, new org.netbeans.lib.awtextra.AbsoluteConstraints(610, 0,
-1, 20));

jLabel26.setFont(resourceMap.getFont("jLabel26.font")); // NOI18N
jLabel26.setText(resourceMap.getString("jLabel26.text")); // NOI18N
jLabel26.setName("jLabel26"); // NOI18N
mainPanel.add(jLabel26, new org.netbeans.lib.awtextra.AbsoluteConstraints(610,
40, -1, 20));

jPasswordField1.setText(resourceMap.getString("jPasswordField1.text")); //
NOI18N
jPasswordField1.setName("jPasswordField1"); // NOI18N
mainPanel.add(jPasswordField1, new
org.netbeans.lib.awtextra.AbsoluteConstraints(610, 60, 130, 20));

jButton3.setFont(resourceMap.getFont("jButton3.font")); // NOI18N
jButton3.setForeground(resourceMap.getColor("jButton3.foreground")); // NOI18N
jButton3.setText(resourceMap.getString("jButton3.text")); // NOI18N
jButton3.setName("jButton3"); // NOI18N

```

```

jButton3.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton3ActionPerformed(evt);
    }
});
mainPanel.add(jButton3, new org.netbeans.lib.awtextra.AbsoluteConstraints(640,
90, -1, 20));

jRadioButton11.setText(resourceMap.getString("jRadioButton11.text")); // NOI18N
jRadioButton11.setName("jRadioButton11"); // NOI18N
jRadioButton11.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        jRadioButton11MouseClicked(evt);
    }
});
mainPanel.add(jRadioButton11, new
org.netbeans.lib.awtextra.AbsoluteConstraints(240, 260, 140, -1));

jRadioButton13.setText(resourceMap.getString("jRadioButton13.text")); // NOI18N
jRadioButton13.setName("jRadioButton13"); // NOI18N
jRadioButton13.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        jRadioButton13MouseClicked(evt);
    }
});
mainPanel.add(jRadioButton13, new
org.netbeans.lib.awtextra.AbsoluteConstraints(80, 260, -1, -1));

jTextField4.setText(resourceMap.getString("jTextField4.text")); // NOI18N
jTextField4.setName("jTextField4"); // NOI18N
mainPanel.add(jTextField4, new org.netbeans.lib.awtextra.AbsoluteConstraints(90,
290, 120, 20));

jTextField5.setText(resourceMap.getString("jTextField5.text")); // NOI18N
jTextField5.setName("jTextField5"); // NOI18N
mainPanel.add(jTextField5, new
org.netbeans.lib.awtextra.AbsoluteConstraints(250, 290, 120, 20));

jButton4.setFont(resourceMap.getFont("jButton4.font")); // NOI18N
jButton4.setText(resourceMap.getString("jButton4.text")); // NOI18N
jButton4.setName("jButton4"); // NOI18N
jButton4.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton4ActionPerformed(evt);
    }
});
mainPanel.add(jButton4, new org.netbeans.lib.awtextra.AbsoluteConstraints(380,
290, 50, 20));

```

```

jRadioButton14.setText(resourceMap.getString("jRadioButton14.text")); // NOI18N
jRadioButton14.setName("jRadioButton14"); // NOI18N
jRadioButton14.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        jRadioButton14MouseClicked(evt);
    }
});
mainPanel.add(jRadioButton14, new
org.netbeans.lib.awtextra.AbsoluteConstraints(80, 340, -1, -1));

jRadioButton19.setText(resourceMap.getString("jRadioButton19.text")); // NOI18N
jRadioButton19.setName("jRadioButton19"); // NOI18N
jRadioButton19.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        jRadioButton19MouseClicked(evt);
    }
});
mainPanel.add(jRadioButton19, new
org.netbeans.lib.awtextra.AbsoluteConstraints(240, 340, 140, -1));

jTextField6.setName("jTextField6"); // NOI18N
mainPanel.add(jTextField6, new org.netbeans.lib.awtextra.AbsoluteConstraints(90,
370, 120, 20));

jTextField7.setName("jTextField7"); // NOI18N
mainPanel.add(jTextField7, new
org.netbeans.lib.awtextra.AbsoluteConstraints(250, 370, 120, 20));

jButton5.setFont(resourceMap.getFont("jButton5.font")); // NOI18N
jButton5.setText(resourceMap.getString("jButton5.text")); // NOI18N
jButton5.setName("jButton5"); // NOI18N
jButton5.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton5ActionPerformed(evt);
    }
});
mainPanel.add(jButton5, new org.netbeans.lib.awtextra.AbsoluteConstraints(380,
370, 50, 20));

jLabel19.setFont(resourceMap.getFont("jLabel19.font")); // NOI18N
jLabel19.setText(resourceMap.getString("jLabel19.text")); // NOI18N
jLabel19.setName("jLabel19"); // NOI18N
mainPanel.add(jLabel19, new org.netbeans.lib.awtextra.AbsoluteConstraints(50,
320, 150, -1));

jLabel27.setFont(resourceMap.getFont("jLabel27.font")); // NOI18N
jLabel27.setText(resourceMap.getString("jLabel27.text")); // NOI18N

```

```

jLabel27.setName("jLabel27"); // NOI18N
mainPanel.add(jLabel27, new org.netbeans.lib.awtextra.AbsoluteConstraints(50,
400, 150, -1));

jScrollPane19.setName("jScrollPane19"); // NOI18N

jList15.setModel(new javax.swing.AbstractListModel() {
    String[] strings = { "Acquired immune deficiency syndrome (AIDS)", "Alcohol
abuse", "Chronic blood loss anemia", "Chronic pulmonary disease", "Coagulopathy",
"Congestive heart failure", "Deficiency anemias", "Depression", "Diabetes with chronic
complications", "Diabetes, uncomplicated", "Drug abuse", "Fluid and electrolyte
disorders", "Hypertension (combine uncomplicated and complicated)",
"Hypothyroidism", "Liver disease", "Lymphoma", "Metastatic cancer", "Obesity", "Other
neurological disorders", "Paralysis", "Peptic ulcer disease excluding bleeding",
"Peripheral vascular disorders", "Psychoses", "Pulmonary circulation disorders", "Renal
failure", "Rheumatoid arthritis/collagen vascular diseases", "Solid tumor without
metastasis", "Valvular disease", "Weight loss" };
    public int getSize() { return strings.length; }
    public Object getElementAt(int i) { return strings[i]; }
});
jList15.setToolTipText(resourceMap.getString("jList15.toolTipText")); // NOI18N
jList15.setName("jList15"); // NOI18N
jList15.addListSelectionListener(new javax.swing.event.ListSelectionListener() {
    public void valueChanged(javax.swing.event.ListSelectionEvent evt) {
        jList15ValueChanged(evt);
    }
});
jScrollPane19.setViewportView(jList15);

mainPanel.add(jScrollPane19, new
org.netbeans.lib.awtextra.AbsoluteConstraints(450, 260, 300, 170));

jRadioButton26.setText(resourceMap.getString("jRadioButton26.text")); // NOI18N
jRadioButton26.setName("jRadioButton26"); // NOI18N
jRadioButton26.addMouseListener(new java.awt.event.MouseAdapter() {
    public void mouseClicked(java.awt.event.MouseEvent evt) {
        jRadioButton26MouseClicked(evt);
    }
});
mainPanel.add(jRadioButton26, new
org.netbeans.lib.awtextra.AbsoluteConstraints(80, 420, -1, -1));

jTextField14.setName("jTextField14"); // NOI18N
mainPanel.add(jTextField14, new
org.netbeans.lib.awtextra.AbsoluteConstraints(90, 450, 120, 20));

jTextField15.setName("jTextField15"); // NOI18N

```



```

    mainPanel.add(jTextField15, new
org.netbeans.lib.awtextra.AbsoluteConstraints(250, 450, 120, 20));

    jButton27.setText(resourceMap.getString("jRadioButton27.text")); // NOI18N
    jButton27.setName("jRadioButton27"); // NOI18N
    jButton27.addMouseListener(new java.awt.event.MouseAdapter() {
        public void mouseClicked(java.awt.event.MouseEvent evt) {
            jButton27MouseClicked(evt);
        }
    });
    mainPanel.add(jButton27, new
org.netbeans.lib.awtextra.AbsoluteConstraints(240, 420, 140, -1));

    jButton9.setFont(resourceMap.getFont("jButton9.font")); // NOI18N
    jButton9.setText(resourceMap.getString("jButton9.text")); // NOI18N
    jButton9.setName("jButton9"); // NOI18N
    jButton9.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            jButton9ActionPerformed(evt);
        }
    });
    mainPanel.add(jButton9, new org.netbeans.lib.awtextra.AbsoluteConstraints(380,
450, 50, 20));

    jButton28.setText(resourceMap.getString("jRadioButton28.text")); // NOI18N
    jButton28.setName("jRadioButton28"); // NOI18N
    jButton28.addMouseListener(new java.awt.event.MouseAdapter() {
        public void mouseClicked(java.awt.event.MouseEvent evt) {
            jButton28MouseClicked(evt);
        }
    });
    mainPanel.add(jButton28, new
org.netbeans.lib.awtextra.AbsoluteConstraints(80, 500, -1, -1));

    jTextField16.setName("jTextField16"); // NOI18N
    mainPanel.add(jTextField16, new
org.netbeans.lib.awtextra.AbsoluteConstraints(90, 530, 120, 20));

    jTextField17.setName("jTextField17"); // NOI18N
    mainPanel.add(jTextField17, new
org.netbeans.lib.awtextra.AbsoluteConstraints(250, 530, 120, 20));

    jButton29.setText(resourceMap.getString("jRadioButton29.text")); // NOI18N
    jButton29.setName("jRadioButton29"); // NOI18N
    jButton29.addMouseListener(new java.awt.event.MouseAdapter() {
        public void mouseClicked(java.awt.event.MouseEvent evt) {
            jButton29MouseClicked(evt);
        }
    });

```

```

});
mainPanel.add(jRadioButton29, new
org.netbeans.lib.awtextra.AbsoluteConstraints(240, 500, 140, -1));

jButton10.setFont(resourceMap.getFont("jButton10.font")); // NOI18N
jButton10.setText(resourceMap.getString("jButton10.text")); // NOI18N
jButton10.setName("jButton10"); // NOI18N
jButton10.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton10ActionPerformed(evt);
    }
});
mainPanel.add(jButton10, new org.netbeans.lib.awtextra.AbsoluteConstraints(380,
530, 50, 20));

jLabel10.setFont(resourceMap.getFont("jLabel10.font")); // NOI18N
jLabel10.setText(resourceMap.getString("jLabel10.text")); // NOI18N
jLabel10.setName("jLabel10"); // NOI18N
mainPanel.add(jLabel10, new org.netbeans.lib.awtextra.AbsoluteConstraints(50,
480, 130, -1));

menuBar.setName("menuBar"); // NOI18N

fileMenu.setText(resourceMap.getString("fileMenu.text")); // NOI18N
fileMenu.setName("fileMenu"); // NOI18N

javax.swing.ActionMap actionMap =
org.jdesktop.application.Application.getInstance(poet.POETApp.class).getContext().getA
ctionMap(POETView.class, this);
exitMenuItem.setAction(actionMap.get("quit")); // NOI18N
exitMenuItem.setName("exitMenuItem"); // NOI18N
fileMenu.add(exitMenuItem);

menuBar.add(fileMenu);

helpMenu.setText(resourceMap.getString("helpMenu.text")); // NOI18N
helpMenu.setName("helpMenu"); // NOI18N
helpMenu.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        helpMenuActionPerformed(evt);
    }
});

aboutMenuItem.setAction(actionMap.get("showAboutBox")); // NOI18N
aboutMenuItem.setName("aboutMenuItem"); // NOI18N
helpMenu.add(aboutMenuItem);

menuBar.add(helpMenu);

```

```

statusPanel.setName("statusPanel"); // NOI18N

statusPanelSeparator.setName("statusPanelSeparator"); // NOI18N

statusMessageLabel.setName("statusMessageLabel"); // NOI18N

statusAnimationLabel.setHorizontalAlignment(javax.swing.SwingConstants.LEFT);
statusAnimationLabel.setName("statusAnimationLabel"); // NOI18N

progressBar.setName("progressBar"); // NOI18N

    javax.swing.GroupLayout statusPanelLayout = new
javax.swing.GroupLayout(statusPanel);
    statusPanel.setLayout(statusPanelLayout);
    statusPanelLayout.setHorizontalGroup(

statusPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addComponent(statusPanelSeparator, javax.swing.GroupLayout.DEFAULT_SIZE,
798, Short.MAX_VALUE)
    .addGroup(statusPanelLayout.createSequentialGroup()
        .addComponent(statusMessageLabel)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
624, Short.MAX_VALUE)
        .addComponent(progressBar, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(statusAnimationLabel)
        .addContainerGap())
    );
    statusPanelLayout.setVerticalGroup(

statusPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(statusPanelLayout.createSequentialGroup()
        .addComponent(statusPanelSeparator,
javax.swing.GroupLayout.PREFERRED_SIZE, 2,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

    .addGroup(statusPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.
BASELINE)
        .addComponent(statusMessageLabel)
        .addComponent(statusAnimationLabel)
        .addComponent(progressBar, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGap(3, 3, 3))

```

```

);

setComponent(mainPanel);
setMenuBar(menuBar);
setStatusBar(statusPanel);
} // </editor-fold>

private void jComboBox1ItemStateChanged(java.awt.event.ItemEvent evt) {
    // TODO add your handling code here:

    if (jComboBox1.getSelectedItem().toString().equalsIgnoreCase("Custom Age
Range"))
    {
        jTextField1.setVisible(true);
        jTextField2.setVisible(true);
        jLabel3.setVisible(true);
        jLabel7.setVisible(true);
        jComboBox1.setEnabled(false);
    }

}

private void jList14ValueChanged(javax.swing.event.ListSelectionEvent evt) {

}

/*****
***
Query builder
*****/

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

    sec_pr1_nis_code="";
    sec_pr2_nis_code="";
    sec_pr3_nis_code="";
    primary_nis_code="";
    nis_query="";
    pat_age="";
    pat_race="";
    pat_gender="";
    weekend_adm="";
    elective_adm="";
    hosp_teach="";
    hosp_loc="";
    hosp_region="";
    bed_size="";

```

```

CM_AIDS="";
CM_ALCOHOL="";
CM_ANEMDEF="";
CM_ARTH="";
CM_BLDLOSS="";
CM_CHF="";
CM_CHRNLUNG="";
CM_COAG="";
CM_DEPRESS="";
CM_DM="";
CM_DMCX="";
CM_DRUG="";
CM_HTN_C="";
CM_HYPOTHY="";
CM_LIVER="";
CM_LYMPH="";
CM_LYTES="";
CM_METS="";
CM_NEURO="";
CM_OBESE="";
CM_PARA="";
CM_PERIVASC="";
CM_PSYCH="";
CM_PULMCIRC="";
CM_RENLFAIL="";
CM_TUMOR="";
CM_ULCER="";
CM_VALVE="";
CM_WGHTLOSS="";
pat_como="";
q_pr1="";
q_pr2="";
q_pr3="";
q_dx1="";

try {

    // get the age
    if (jComboBox1.isEnabled())
    {
        if (jComboBox1.getSelectedObjects().length >0 )
        {
            Object[] age_sel = jComboBox1.getSelectedObjects();
            pat_age = age_sel[0].toString();
            if (pat_age.equalsIgnoreCase("0 to 10"))
                pat_age = "0 and 10";
            else if (pat_age.equalsIgnoreCase("11 to 20"))

```

```

        pat_age = "11 and 20";
    else if (pat_age.equalsIgnoreCase("21 to 30"))
        pat_age = "31 and 40";
    else if (pat_age.equalsIgnoreCase("41 to 50"))
        pat_age = "51 and 60";
    else if (pat_age.equalsIgnoreCase("61 to 70"))
        pat_age = "71 and 80";
    else if (pat_age.equalsIgnoreCase("81 to 90"))
        pat_age = "81 and 90";
    else if (pat_age.equalsIgnoreCase("Above 90"))
        pat_age = "91 and 130";

    nis_query= nis_query+" and age_category='"+pat_age+"'";
}
}
else
    if (!jTextField2.getText().isEmpty() && !jTextField1.getText().isEmpty() )
    { pat_age = jTextField2.getText() + " and " + jTextField1.getText();
      nis_query= nis_query+" and age between "+pat_age+"'";
    }

System.out.println ("pat_age:"+pat_age);

// patient race
if (jComboBox2.getSelectedObjects().length >0 )
{
    Object[] race_sel = jComboBox2.getSelectedObjects();
    pat_race = race_sel[0].toString();
    if (pat_race.equalsIgnoreCase("white"))
        pat_race = "1";
    else if (pat_race.equalsIgnoreCase("black"))
        pat_race = "2";
    else if (pat_race.equalsIgnoreCase("Hispanic"))
        pat_race = "3";
    else if (pat_race.equalsIgnoreCase("other"))
        pat_race = "6";
    nis_query= nis_query+" and race='"+pat_race+"'";
}
else
    pat_race="";

System.out.println ("pat_race:"+pat_race);

// patient gender
if (jComboBox3.getSelectedObjects().length >0 )
{
    Object[] gender_sel = jComboBox3.getSelectedObjects();

```

```

    pat_gender = gender_sel[0].toString();
    if (pat_gender.equalsIgnoreCase("female"))
        pat_gender = "1";
    else if (pat_gender.equalsIgnoreCase("male"))
        pat_gender = "0";
    nis_query= nis_query+" and female="+pat_gender+"";
}
else
    pat_gender="";

System.out.println ("pat_gender:"+pat_gender);

//weekend adm
if (jRadioButton17.isSelected())
{
    weekend_adm = "0";
    nis_query= nis_query+" and weekend_adm="+weekend_adm+"";
}
else if (jRadioButton18.isSelected())
{
    weekend_adm = "1";
    nis_query= nis_query+" and weekend_adm="+weekend_adm+"";
}

else
    weekend_adm="";
System.out.println ("weekend_adm:"+weekend_adm);

// elective
if (jRadioButton16.isSelected())
{
    elective_adm = "1";
    nis_query= nis_query+" and elec_adm="+elective_adm+"";
}
else if (jRadioButton15.isSelected())
{
    elective_adm = "0";
    nis_query= nis_query+" and elec_adm="+elective_adm+"";
}

else
    elective_adm="";
System.out.println ("elective_adm:"+elective_adm);

// hosp location
if (jRadioButton2.isSelected())

```

```
{
    hosp_loc = "0"; // rural
    nis_query= nis_query+" and hosp_loc='"+hosp_loc+"'";
}
else if (jRadioButton1.isSelected())
{ hosp_loc = "1"; //urban
  nis_query= nis_query+" and hosp_loc='"+hosp_loc+"'";
}
else
  hosp_loc="";
System.out.println ("hosp_loc:"+hosp_loc);

// hosp teaching
if (jRadioButton12.isSelected())
{
  hosp_teach = "1"; // rural
  nis_query= nis_query+" and teaching_status='"+hosp_teach+"'";
}
else if (jRadioButton10.isSelected())
{
  hosp_teach = "0"; //urban
  nis_query= nis_query+" and teaching_status='"+hosp_teach+"'";
}
else
  hosp_teach="";
System.out.println ("hosp_teach:"+hosp_teach);

// hosp region
if (jRadioButton3.isSelected())
{
  hosp_region = "2"; //Midwest
  nis_query= nis_query+" and hosp_region='"+hosp_region+"'";
}
else if (jRadioButton4.isSelected())
{
  hosp_region = "1"; //Northeast
  nis_query= nis_query+" and hosp_region='"+hosp_region+"'";
}
else if (jRadioButton5.isSelected())
{
  hosp_region = "3"; //South
  nis_query= nis_query+" and hosp_region='"+hosp_region+"'";
}
else if (jRadioButton6.isSelected())
{
```



```

    hosp_region = "4";//West
    nis_query= nis_query+" and hosp_region=""+hosp_region+"";
}
else
    hosp_region="";

System.out.println ("hosp_region:"+hosp_region);

// hosp bed_size
if (jRadioButton7.isSelected())
{bed_size = "2"; //Miduim
nis_query= nis_query+" and hosp_bedsizes=""+bed_size+"";}
else if (jRadioButton8.isSelected())
{bed_size = "3"; //large
nis_query= nis_query+" and hosp_bedsizes=""+bed_size+"";}
else if (jRadioButton9.isSelected())
{bed_size = "1"; //Small
nis_query= nis_query+" and hosp_bedsizes=""+bed_size+"";}
else
    bed_size="";

System.out.println ("bed_size:"+bed_size);

// comorbidities
Object[] como = jList14.getSelectedValues();
System.out.println ("como length:"+como.length);

for (int i = 0; i<como.length; i++)
{
    if (como[i].toString().equalsIgnoreCase("Acquired immune deficiency
syndrome (AIDS)"))
        pat_como=pat_como + " AND CM_AIDS ='1'";
    else if (como[i].toString().equalsIgnoreCase("Alcohol abuse"))
        pat_como=pat_como + " AND CM_ALCOHOL ='1'";
    else if (como[i].toString().equalsIgnoreCase("Deficiency anemias"))
        pat_como=pat_como + " AND CM_ANEMDEF ='1'";
    else if (como[i].toString().equalsIgnoreCase("Rheumatoid arthritis/collagen
vascular diseases"))
        pat_como=pat_como + " AND CM_ARTH ='1'";
    else if (como[i].toString().equalsIgnoreCase("Chronic blood loss anemia"))
        pat_como=pat_como + " AND CM_BLDLOSS ='1'";
    else if (como[i].toString().equalsIgnoreCase("Congestive heart failure"))
        pat_como=pat_como + " AND CM_CHF ='1'";
    else if (como[i].toString().equalsIgnoreCase("Chronic pulmonary disease"))
        pat_como=pat_como + " AND CM_CHRNLUNG ='1'";
}

```

```

else if (como[i].toString().equalsIgnoreCase("Coagulopathy"))
    pat_como=pat_como + " AND CM_COAG ='1'";
else if (como[i].toString().equalsIgnoreCase("Depression"))
    pat_como=pat_como + " AND CM_DEPRESS ='1'";
else if (como[i].toString().equalsIgnoreCase("Diabetes, uncomplicated"))
    pat_como=pat_como + " AND CM_DM ='1'";
else if (como[i].toString().equalsIgnoreCase("Diabetes with chronic
complications"))
    pat_como=pat_como + " AND CM_DMCX ='1'";
else if (como[i].toString().equalsIgnoreCase("Drug abuse"))
    pat_como=pat_como + " AND CM_DRUG ='1'";
else if (como[i].toString().equalsIgnoreCase("Hypertension (combine
uncomplicated and complicated"))
    pat_como=pat_como + " AND CM_HTN_C ='1'";
else if (como[i].toString().equalsIgnoreCase("Hypothyroidism"))
    pat_como=pat_como + " AND CM_HYPOTHY ='1'";
else if (como[i].toString().equalsIgnoreCase("Liver disease"))
    pat_como=pat_como + " AND CM_LIVER ='1'";
else if (como[i].toString().equalsIgnoreCase("Lymphoma"))
    pat_como=pat_como + " AND CM_LYMPH ='1'";
else if (como[i].toString().equalsIgnoreCase("Fluid and electrolyte disorders"))
    pat_como=pat_como + " AND CM_LYTES ='1'";
else if (como[i].toString().equalsIgnoreCase("Metastatic cancer"))
    pat_como=pat_como + " AND CM_METS ='1'";
else if (como[i].toString().equalsIgnoreCase("Other neurological disorders"))
    pat_como=pat_como + " AND CM_NEURO ='1'";
else if (como[i].toString().equalsIgnoreCase("Obesity"))
    pat_como=pat_como + " AND CM_OBESE ='1'";
else if (como[i].toString().equalsIgnoreCase("Paralysis"))
    pat_como=pat_como + " AND CM_PARA ='1'";
else if (como[i].toString().equalsIgnoreCase("Peripheral vascular disorders"))
    pat_como=pat_como + " AND CM_PERIVASC ='1'";
else if (como[i].toString().equalsIgnoreCase("Psychoses"))
    pat_como=pat_como + " AND CM_PSYCH ='1'";
else if (como[i].toString().equalsIgnoreCase("Pulmonary circulation
disorders"))
    pat_como=pat_como + " AND CM_PULMCIRC ='1'";
else if (como[i].toString().equalsIgnoreCase("Renal failure"))
    pat_como=pat_como + " AND CM_RENLFAIL ='1'";
else if (como[i].toString().equalsIgnoreCase("Solid tumor without metastasis"))
    pat_como=pat_como + " AND CM_TUMOR ='1'";
else if (como[i].toString().equalsIgnoreCase("Peptic ulcer disease excluding
bleeding"))
    pat_como=pat_como + " AND CM_ULCER ='1'";
else if (como[i].toString().equalsIgnoreCase("Valvular disease"))
    pat_como=pat_como + " AND CM_VALVE ='1'";
else if (como[i].toString().equalsIgnoreCase("Weight loss"))
    pat_como=pat_como + " AND CM_WGHTLOSS ='1'";

```

```

}

System.out.println ("pat_como:"+pat_como);

nis_query= nis_query+pat_como;

// pr1

if (!jTextField4.getText().isEmpty() && jTextField5.getText().isEmpty() )
    q_pr1 = jTextField4.getText();
else if (!jTextField5.getText().isEmpty() && jTextField4.getText().isEmpty() )
    q_pr1 = jTextField5.getText();

if (!q_pr1.equalsIgnoreCase(""))
    nis_query= nis_query+" and pr1='"+q_pr1+"'";

// System.out.println ("pr1:"+q_pr1);
//pr2
if (!jTextField6.getText().isEmpty() && jTextField7.getText().isEmpty() )
    q_pr2 = jTextField6.getText();

else if
    (!jTextField7.getText().isEmpty() && jTextField6.getText().isEmpty() )
    q_pr2 = jTextField7.getText();

if (!q_pr2.equalsIgnoreCase(""))
    nis_query= nis_query+" and pr2='"+q_pr2+"'";

// System.out.println ("pr2:"+q_pr2);
//pr3
if (!jTextField14.getText().isEmpty() && jTextField15.getText().isEmpty() )
    q_pr3 = jTextField14.getText();
else if
    (!jTextField15.getText().isEmpty() && jTextField14.getText().isEmpty() )
    q_pr3 = jTextField15.getText();

// System.out.println ("pr3:"+q_pr3);

if (!q_pr3.equalsIgnoreCase(""))
    nis_query= nis_query+" and pr3='"+q_pr3+"'";
//dx1
if (!jTextField16.getText().isEmpty() && jTextField17.getText().isEmpty() )
    q_dx1 = jTextField16.getText();
else if
    (!jTextField17.getText().isEmpty() && jTextField16.getText().isEmpty() )
    q_dx1 = jTextField17.getText();

```

```

// System.out.println ("dx1:"+q_dx1);
// if (!q_dx1.equalsIgnoreCase(""))
//     nis_query= nis_query+" and dx1='"+q_dx1.toUpperCase()+"';
////
/*****
***
Query Disptacher
*****/

    /// table names
    String table_name = "";
    if (!q_pr1.equalsIgnoreCase(""))
    {
        if ((q_pr1.substring(0,1).equalsIgnoreCase("0")) ||
(q_pr1.substring(0,1).equalsIgnoreCase("1")) ||
(q_pr1.substring(0,1).equalsIgnoreCase("2")))
            table_name = "pat_pr_0_2 pr, pat_como_0_2 cm, hosp_info hi where
pr.key = cm.key and pr.hospid = hi.hospid ";
        else if (q_pr1.substring(0,1).equalsIgnoreCase("3"))
            table_name = "pat_pr_3 pr, pat_como_3 cm, hosp_info hi where pr.key =
cm.key and pr.hospid = hi.hospid ";
        else if (q_pr1.substring(0,1).equalsIgnoreCase("4"))
            table_name = "pat_pr_4 pr, pat_como_4 cm, hosp_info hi where pr.key =
cm.key and pr.hospid = hi.hospid ";
        else if (q_pr1.substring(0,1).equalsIgnoreCase("5"))
            table_name = "pat_pr_5 pr, pat_como_5 cm, hosp_info hi where pr.key =
cm.key and pr.hospid = hi.hospid ";
        else if (q_pr1.substring(0,1).equalsIgnoreCase("6"))
            table_name = "pat_pr_6 pr, pat_como_6 cm, hosp_info hi where pr.key =
cm.key and pr.hospid = hi.hospid ";
        else if (q_pr1.substring(0,1).equalsIgnoreCase("7"))
            table_name = "pat_pr_7 pr, pat_como_7 cm, hosp_info hi where pr.key =
cm.key and pr.hospid = hi.hospid ";
        else if (q_pr1.substring(0,1).equalsIgnoreCase("8"))
            table_name = "pat_pr_8 pr, pat_como_8 cm, hosp_info hi where pr.key =
cm.key and pr.hospid = hi.hospid ";
        else if (q_pr1.substring(0,1).equalsIgnoreCase("9"))
            table_name = "pat_pr_9 pr, pat_como_9 cm, hosp_info hi where pr.key =
cm.key and pr.hospid = hi.hospid ";
    }

    String mortality_query = "select count(distinct pr.key) from "+ table_name +
nis_query+" and died='1'";
    String los_query = "select avg(los) from "+table_name + nis_query;
    String pr_cnt_query = "select count(distinct pr.key) from "+table_name +
nis_query;

```

```

        String dispo_home_query = "select count(distinct pr.key) from "+ table_name +
nis_query+" and dispo='1'";
        String dispo_facility_query = "select count(distinct pr.key) from "+ table_name
+ nis_query+" and dispo In ('5')";
        String dispo_hhc_query = "select count(distinct pr.key) from "+ table_name +
nis_query+" and dispo='6'";

        String all_cnt_query ="select
TOT_CNT,TOT_DIED,AVG_LOS,TOT_DISPO_HOME,TOT_DISPO_FACILITY,TOT_DISPO_HH
C from pr_counts where pr1='"+q_pr1+"'";

        System.out.println ("mortality_query:"+mortality_query);
        System.out.println ("los_query:"+los_query);
        System.out.println ("all_cnt_query:"+all_cnt_query);
        System.out.println ("dispo_home_query:"+dispo_home_query);
        System.out.println ("dispo_facility_query:"+dispo_facility_query);
        System.out.println ("dispo_hhc_query:"+dispo_hhc_query);
        System.out.println ("pr_cnt_query:"+pr_cnt_query);
        /*
        nis_query = "select count(*) from nis_2008_core c28,nis_2008_severity
s28,nis_2008_hosp h28 where c28.key = s28.key"+
        " and c28.hospid = h28.hospid and pr1 ='"+ primary_nis_code+"' and pr2
='"+sec_pr1_nis_code+"' and pr3 ='"+sec_pr2_nis_code+"' and pr4
='"+sec_pr3_nis_code
        +"' "+pat_como+ " and age_num between "+ pat_age + " and race = '"+
pat_race+"' and female = '"+ pat_gender+"' "+" and aweekend ='"+
        weekend_adm + "' and elective ='"+ elective_adm + "' and hosp_teach ='"+
hosp_teach + "' and hosp_location ='"+hosp_loc+"' and hosp_region ='
        + hosp_region+"' and hosp_bedsizes ='"+ bed_size+"'";
        */

        double pr_avg_los=0;
        int all_cnt = 0;
        int all_mortality = 0;
        double all_avg_los = 0;
        int all_tot_dispo_home = 0;
        int all_tot_dispo_facility =0;
        int all_tot_dispo_hhc = 0;
        int pr_dispo_home=0;
        int pr_dispo_hhc=0;
        int pr_dispo_facility =0;

        Statement stmt = conn.createStatement();

        ResultSet rset_pr_cnt= stmt.executeQuery(pr_cnt_query);

```

```
int pr_cnt=0;
while (rset_pr_cnt.next()) {
    pr_cnt = rset_pr_cnt.getInt(1);
    //System.out.println ("IN pr_mortality:"+pr_mortality);
}

// pr_mortality
ResultSet rset_m = stmt.executeQuery(mortality_query);
int pr_mortality=0;
while (rset_m.next()) {
    pr_mortality = rset_m.getInt(1);
    //System.out.println ("IN pr_mortality:"+pr_mortality);
}

ResultSet rset_l = stmt.executeQuery(los_query);
//pr_avg_los
while (rset_l.next()) {
    pr_avg_los = rset_l.getDouble(1) ;

    int temp=(int)((pr_avg_los*Math.pow(10,2)));
    pr_avg_los = (((double)temp)/Math.pow(10,2));

}

ResultSet rset_d_home = stmt.executeQuery(dispo_home_query);

//pr_dispo_home
while (rset_d_home.next()) {
    pr_dispo_home = rset_d_home.getInt(1);

}

ResultSet rset_d_facility = stmt.executeQuery(dispo_facility_query);

while (rset_d_facility.next()) {
    pr_dispo_facility = rset_d_facility.getInt(1);

}

ResultSet rset_d_hhc = stmt.executeQuery(dispo_hhc_query);

while (rset_d_hhc.next()) {
    pr_dispo_hhc = rset_d_hhc.getInt(1);

}

ResultSet rset_all = stmt.executeQuery(all_cnt_query);
```

```

while (rset_all.next()) {
    all_cnt = rset_all.getInt(1);
    all_mortality = rset_all.getInt(2);
    all_avg_los = rset_all.getDouble(3);

    int temp1=(int)((all_avg_los*Math.pow(10,2)));
    all_avg_los = (((double)temp1)/Math.pow(10,2));

    all_tot_dispo_home = rset_all.getInt(4);
    all_tot_dispo_facility = rset_all.getInt(5);
    all_tot_dispo_hhc = rset_all.getInt(6);

}

System.out.println ("pr_mortality:"+pr_mortality);
System.out.println ("pr_avg_los:"+pr_avg_los);
System.out.println ("pr_dispo_home:"+pr_dispo_home);
System.out.println ("pr_dispo_hhc:"+pr_dispo_hhc);
System.out.println ("pr_dispo_facility:"+pr_dispo_facility);

    System.out.println ("pr1:"+q_pr1+", all_cnt:"+ all_cnt+", all_mortality:"+
all_mortality +", all_avg_los:"+ all_avg_los+ ", all_tot_dispo_home:"+
all_tot_dispo_home+", all_tot_dispo_facility:"+ all_tot_dispo_facility+",
all_tot_dispo_hhc:"+ all_tot_dispo_hhc);

    double pr_mortality_risk=0.0;
    String m_risk="";
    if (pr_cnt !=0)
    {pr_mortality_risk = pr_mortality / pr_cnt;

    if (pr_mortality_risk < 1)
        m_risk = "Patient-Specific"+" Risk: <1%";
    else if ((pr_mortality_risk >= 1) && (pr_mortality_risk <= 3))
        m_risk = "Patient-Specific"+" Risk: 1 - 3(%)";
    else if (pr_mortality_risk > 3)
        m_risk = "Patient-Specific"+" Risk: > 3%";
    }
    else
        m_risk="No patients with given crietria";

    double pr_all_mortality_risk=0.0;
    String m_all_risk="";

```

```

if (all_cnt !=0)
{
pr_all_mortality_risk = all_mortality / all_cnt;
if (pr_all_mortality_risk < 1)
    m_all_risk = "Average"+" Risk: <1%";
else if ((pr_all_mortality_risk >= 1) && (pr_all_mortality_risk <= 3))
    m_all_risk = "Average"+" Risk: 1 - 3(%)";
else if (pr_all_mortality_risk > 3)
    m_all_risk = "Average"+" Risk: > 3%";
}
else
    m_all_risk="No patients with given criteria";

String rr_mortality_msg="";

if (all_cnt!=0)
    {double rr_mortality = pr_mortality_risk/pr_all_mortality_risk;
    int temp2=(int)((rr_mortality*Math.pow(10,2)));
    rr_mortality = (((double)temp2)/Math.pow(10,2));
    rr_mortality_msg = "Relative Inpatient Mortality Risk (Patient-
Specific/Average): "+ rr_mortality;}
else
    rr_mortality_msg="No patients with given criteria";

//print m_risk, m_all_risk, rr_mortality_msg

// for LOS, show the pr_avg_los, and all_avg_los
String Los_msg = "Average: "+all_avg_los+" days";
String Los_msg2="Patient-Specific: "+pr_avg_los+" days";

double pr_dispo_home_risk=0.0;
String dispo_home_risk_msg="";
double avg_dispo_home_risk=0.0;
String avg_dispo_home_risk_msg="";
if (all_cnt!=0)
{
pr_dispo_home_risk = (((double)pr_dispo_home)/((double)pr_cnt))*100;
int temp5=(int)((pr_dispo_home_risk*Math.pow(10,2)));
pr_dispo_home_risk = (((double)temp5)/Math.pow(10,2));
dispo_home_risk_msg = "(Patient-Specific)+" Disposed
Home:"+pr_dispo_home_risk+"%";

avg_dispo_home_risk = (((double)all_tot_dispo_home)/((double)all_cnt))*100;
int temp15=(int)((avg_dispo_home_risk*Math.pow(10,2)));
avg_dispo_home_risk = (((double)temp15)/Math.pow(10,2));

```



```

    avg_dispo_home_risk_msg = "(Average)+" Disposed
Home:"+avg_dispo_home_risk+"%";

}
else
    dispo_home_risk_msg="No patients with given crietria";

double pr_dispo_facility_risk=0.0;
String dispo_facility_risk_msg="";

double avg_dispo_facility_risk=0.0;
String avg_dispo_facility_risk_msg="";

if (all_cnt!=0)
{
//System.out.println ("IN pr_dispo_facility="+pr_dispo_facility);
    pr_dispo_facility_risk = (((double)pr_dispo_facility)/((double)pr_cnt))*100;
    int temp4=(int)((pr_dispo_facility_risk*Math.pow(10,2)));
    pr_dispo_facility_risk = (((double)temp4)/Math.pow(10,2));
//System.out.println ("IN pr_dispo_facility_risk="+pr_dispo_facility_risk);

    dispo_facility_risk_msg = "(Patient-Specific)+" Disposed to Another Care
Facility:"+pr_dispo_facility_risk+"%";

    avg_dispo_facility_risk = (((double)all_tot_dispo_facility)/((double)all_cnt))*100;
    int temp16=(int)((avg_dispo_facility_risk*Math.pow(10,2)));
    avg_dispo_facility_risk = (((double)temp16)/Math.pow(10,2));
    avg_dispo_facility_risk_msg = "(Average)+" Disposed to Another Care
Facility:"+avg_dispo_facility_risk+"%";

}
else
    dispo_facility_risk_msg="No patients with given crietria";

double pr_dispo_hhc_risk=0.0;
String dispo_hhc_risk_msg="";

double avg_dispo_hhc_risk=0.0;
String avg_dispo_hhc_risk_msg="";

if (all_cnt!=0)
{
    pr_dispo_hhc_risk = (((double)pr_dispo_hhc)/((double)pr_cnt))*100;
    int temp10=(int)((pr_dispo_hhc_risk*Math.pow(10,2)));
    pr_dispo_hhc_risk = (((double)temp10)/Math.pow(10,2));

```

```

        dispo_hhc_risk_msg = "(Patient-Specific)"+" Disposed to Home Health Care
(HHC):"+pr_dispo_hhc_risk+"%";

        avg_dispo_hhc_risk = (((double)all_tot_dispo_hhc)/((double)all_cnt))*100;
        int temp17=(int)((avg_dispo_hhc_risk*Math.pow(10,2)));
        avg_dispo_hhc_risk = (((double)temp17)/Math.pow(10,2));
        avg_dispo_hhc_risk_msg = "(Average)"+" Disposed to Home Health Care
(HHC):"+avg_dispo_hhc_risk+"%";

    }
    else
        dispo_hhc_risk_msg="No patients with given crietria";

    // print los_msg, los_msg2
    JOptionPane.showMessageDialog(mainPanel,
        "\n<html><b><i>-----IN-PATIENT MORTALITY-----
</i></b></html>\n"+m_risk+"\n"+m_all_risk+"\n\n"+
        "\n<html><b><i>-----LENGTH OF STAY-----
</i></b></html>\n"+Los_msg2+"\n"+Los_msg+"\n\n"+
        "\n<html><b><i>-----DISPOSITION STATUS-----
</i></b></html>\n"+dispo_home_risk_msg+"\n"+avg_dispo_home_risk_msg+"\n\n"+
        dispo_facility_risk_msg+"\n"+avg_dispo_facility_risk_msg+"\n\n"+
        dispo_hhc_risk_msg+"\n"+avg_dispo_hhc_risk_msg);

    //todo:
    //1-check if counts are zero and when to create relative risk
    //2- put results in a dialog msg

    stmt.close();
    System.out.println ("Ok.");
    } catch (SQLException e) {
        JOptionPane.showMessageDialog(mainPanel,"SQL EXCEPTION: Can't connect
to the database, check VPN connection and your Username/Password");
        System.out.println ("SQLEXCEPTION:"+e);
    }

}

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:

    jTextField1.setVisible(false);
    jTextField2.setVisible(false);
    jLabel3.setVisible(false);
    jLabel7.setVisible(false);

```

```
jComboBox1.setEnabled(true);
jComboBox1.setSelectedIndex(0);
jTextField1.setText(null);
jTextField2.setText(null);
jTextField4.setText(null);
jTextField4.setVisible(false);
jComboBox2.setSelectedIndex(0);
jComboBox3.setSelectedIndex(0);
buttonGroup1.clearSelection();
buttonGroup2.clearSelection();
buttonGroup3.clearSelection();
buttonGroup4.clearSelection();
buttonGroup6.clearSelection();
buttonGroup5.clearSelection();
buttonGroup9.clearSelection();
buttonGroup10.clearSelection();
buttonGroup11.clearSelection();
jTextField5.setText(null);
jTextField5.setVisible(false);
jButton4.setVisible(false);

jButton5.setVisible(false);
jButton9.setVisible(false);
jButton10.setVisible(false);
jTextField6.setText(null);
jTextField6.setVisible(false);

jTextField7.setText(null);
jTextField7.setVisible(false);

jTextField14.setText(null);
jTextField14.setVisible(false);

jTextField15.setText(null);
jTextField15.setVisible(false);

jTextField16.setText(null);
jTextField16.setVisible(false);

jTextField17.setText(null);
jTextField17.setVisible(false);
//jRadioButton17.setSelected(false);
//jRadioButton18.setSelected(false);
//    jTextField4.setVisible(false);
//    jLabel27.setVisible(false);
//    jComboBox4.setEnabled(true);

//    jTextField5.setVisible(false);
```

```
//    jLabel28.setVisible(false);
//    jComboBox5.setEnabled(true);

//    jTextField6.setVisible(false);
//    jLabel29.setVisible(false);
//    jComboBox6.setEnabled(true);

//    jTextField7.setVisible(false);
//    jLabel30.setVisible(false);
//    jComboBox7.setEnabled(true);

//    jComboBox4.setSelectedIndex(0);
//    jComboBox5.setSelectedIndex(0);
//    jComboBox6.setSelectedIndex(0);
//    jComboBox7.setSelectedIndex(0);

jList14.clearSelection();
jList14.ensureIndexIsVisible(0);

sec_pr1_nis_code="";
sec_pr2_nis_code="";
sec_pr3_nis_code="";
primary_nis_code="";
nis_query="";
pat_age="";
pat_race="";
pat_gender="";
weekend_adm="";
elective_adm="";
hosp_teach="";
hosp_loc="";
hosp_region="";
bed_size="";
CM_AIDS="";
CM_ALCOHOL="";
CM_ANEMDEF="";
CM_ARTH="";
CM_BLDLOSS="";
CM_CHF="";
CM_CHRNLUNG="";
CM_COAG="";
CM_DEPRESS="";
CM_DM="";
CM_DMCX="";
CM_DRUG="";
CM_HTN_C="";
CM_HYPOTHY="";
CM_LIVER="";
```

```

CM_LYMPH="";
CM_LYTES="";
CM_METS="";
CM_NEURO="";
CM_OBESE="";
CM_PARA="";
CM_PERIVASC="";
CM_PSYCH="";
CM_PULMCIRC="";
CM_RENLFAIL="";
CM_TUMOR="";
CM_ULCER="";
CM_VALVE="";
CM_WGHTLOSS="";
pat_como="";

}

private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        // TODO add your handling code here:

        Class.forName("oracle.jdbc.OracleDriver");

        url = "jdbc:oracle:thin:@bmhidb.ahc.umn.edu:1521:bmhi";

        pass_str = new String (jPasswordField1.getPassword());

        conn = DriverManager.getConnection(url,jTextField3.getText(),pass_str);
//"Hom3coming_parade123"
    } catch (SQLException e) {
        JOptionPane.showMessageDialog(mainPanel, "SQL EXCEPTION: Can't connect
to the database, check VPN connection and your Username/Password");
        System.out.println ("SQLEXCEPTION:"+e);
    } catch (ClassNotFoundException e) {
        JOptionPane.showMessageDialog(mainPanel, "JDBC Issue: Can't Locate the
JDBC Driver");
        System.out.println ("SQLEXCEPTION:"+e);
    }
}

private void jButton13MouseClicked(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:

```

```

jTextField4.setText(null);
jTextField4.setVisible(true);
jTextField5.setText(null);
jTextField5.setVisible(false);
jButton4.setVisible(false);

}

private void jButton11MouseClicked(java.awt.event.MouseEvent evt) {

    // TODO add your handling code here:
    jTextField4.setText(null);
    jTextField4.setVisible(false);
    jTextField5.setText(null);
    jTextField5.setVisible(true);
    jButton4.setVisible(true);

    // get all matching codes with their description from code_desc table

}

private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:

    try {
        Statement stmt = conn.createStatement();
        String query = "select code, description from code_desc where flag='p' and
upper(description) like Upper('%"+jTextField5.getText()+"%')";
        //System.out.println (query);
        ResultSet rset =
        stmt.executeQuery(query);

        ArrayList<Object> q_res = new ArrayList<Object>();

        while (rset.next()) {
            q_res.add((rset.getString(1)+" "+rset.getString(2)));

            // System.out.println (rset.getString(1)+" "+rset.getString(2));

        }
        if (!q_res.isEmpty())
        {
            String [] query_res = q_res.toArray(new String[q_res.size()]);

```

```
String selection = (String) JOptionPane.showInputDialog(null, "Please choose a
code", "Code Finder", JOptionPane.QUESTION_MESSAGE, null, query_res, query_res[0]);
```

```
String[] res = selection.split(",");
```

```
textField5.setText(res[0].trim());
```

```
}
```

```
else
```

```
{
```

```
    JOptionPane.showMessageDialog(mainPanel, "Can't Find a Matching
Procedure");
```

```
    textField5.setText("");
```

```
}
```

```
stmt.close();
```

```
//System.out.println ("code:"+res[0].trim());
```

```
} catch (SQLException ex) {
```

```
    Logger.getLogger(POETView.class.getName()).log(Level.SEVERE, null, ex);
```

```
}
```

```
}
```

```
private void jButton14MouseClicked(java.awt.event.MouseEvent evt) {
```

```
    // TODO add your handling code here:
```

```
    textField6.setText(null);
```

```
    textField6.setVisible(true);
```

```
    textField7.setText(null);
```

```
    textField7.setVisible(false);
```

```
    jButton5.setVisible(false);
```

```
}
```

```
private void jButton19MouseClicked(java.awt.event.MouseEvent evt) {
```

```
    // TODO add your handling code here:
```

```
    textField6.setText(null);
```

```
    textField6.setVisible(false);
```

```
    textField7.setText(null);
```

```
    textField7.setVisible(true);
```

```
    jButton5.setVisible(true);
```

```
}
```

```
private void jButton5ActionPerformed(java.awt.event.ActionEvent evt) {
```

```
    // TODO add your handling code here:
```

```
try {
```

```

        Statement stmt = conn.createStatement();
        String query = "select code, description from code_desc where flag='p' and
upper(description) like Upper('%"+jTextField7.getText()+"%')";
        //System.out.println (query);
        ResultSet rset =
        stmt.executeQuery(query);

        ArrayList<Object> q_res = new ArrayList<Object>();

        while (rset.next()) {
            q_res.add((rset.getString(1)+" "+rset.getString(2)));

            //System.out.println (rset.getString(1)+" "+rset.getString(2));

        }

        if (!q_res.isEmpty())
        {
            String [] query_res = q_res.toArray(new String[q_res.size()]);

            String selection = (String) JOptionPane.showInputDialog(null, "Please choose a
code", "Code Finder", JOptionPane.QUESTION_MESSAGE, null, query_res, query_res[0]);

            String[] res = selection.split(" ");

            jTextField7.setText(res[0].trim());
        }
        else
        {
            JOptionPane.showMessageDialog(mainPanel, "Can't Find a Matching
Procedure");

            jTextField7.setText("");
        }
        stmt.close();

        //System.out.println ("code:"+res[0].trim());

    } catch (SQLException ex) {
        Logger.getLogger(POETView.class.getName()).log(Level.SEVERE, null, ex);
    }

}

private void jList15ValueChanged(javax.swing.event.ListSelectionEvent evt) {
    // TODO add your handling code here:
}

```



```

private void jButton10MouseClicked(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
    jTextField14.setText(null);
    jTextField14.setVisible(true);
    jTextField15.setText(null);
    jTextField15.setVisible(false);
    jButton9.setVisible(false);
}

private void jButton11MouseClicked(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
    jTextField16.setText(null);
    jTextField16.setVisible(true);
    jTextField17.setText(null);
    jTextField17.setVisible(false);
    jButton10.setVisible(false);
}

private void jButton12MouseClicked(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
    jTextField17.setText(null);
    jTextField17.setVisible(true);
    jTextField16.setText(null);
    jTextField16.setVisible(false);
    jButton10.setVisible(true);
}

private void jButton10ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:

    try {
        Statement stmt = conn.createStatement();
        String query = "select code, description from code_desc where flag ='d' and
upper(description) like Upper('%"+jTextField17.getText()+"%')";
        //System.out.println (query);
        ResultSet rset =
        stmt.executeQuery(query);

        ArrayList<Object> q_res = new ArrayList<Object>();

        while (rset.next()) {
            q_res.add((rset.getString(1)+" "+rset.getString(2)));

            // System.out.println (rset.getString(1)+" "+rset.getString(2));

```

```

    }
    if(!q_res.isEmpty())
    {
        String [] query_res = q_res.toArray(new String[q_res.size()]);

        String selection = (String) JOptionPane.showInputDialog(null, "Please choose a
code", "Code Finder", JOptionPane.QUESTION_MESSAGE, null, query_res, query_res[0]);

        String[] res = selection.split(",");

        jTextField17.setText(res[0].trim());
    }
    else
    {
        JOptionPane.showMessageDialog(mainPanel,"Can't Find a Matching
Diagnosis");

        jTextField17.setText("");
    }
    stmt.close();

    //System.out.println ("code:"+res[0].trim());

} catch (SQLException ex) {
    Logger.getLogger(POETView.class.getName()).log(Level.SEVERE, null, ex);
}

}

private void jButton27MouseClicked(java.awt.event.MouseEvent evt) {

    jTextField15.setText(null);
    jTextField15.setVisible(true);
    jTextField14.setText(null);
    jTextField14.setVisible(false);
    jButton9.setVisible(true);
    // TODO add your handling code here:}
}

private void jButton9ActionPerformed(java.awt.event.ActionEvent evt) {

    try {
        Statement stmt = conn.createStatement();
        String query = "select code, description from code_desc where flag ='p' and
upper(description) like Upper('%"+jTextField15.getText()+"%')";
        //System.out.println (query);
        ResultSet rset =
        stmt.executeQuery(query);
    }
}

```

```

ArrayList<Object> q_res = new ArrayList<Object>();

while (rset.next()) {
    q_res.add((rset.getString(1)+" "+rset.getString(2)));

    // System.out.println (rset.getString(1)+" "+rset.getString(2));

}

if (!q_res.isEmpty())
{
    String [] query_res = q_res.toArray(new String[q_res.size()]);

    String selection = (String) JOptionPane.showInputDialog(null, "Please choose a
code", "Code Finder", JOptionPane.QUESTION_MESSAGE, null, query_res, query_res[0]);

    String[] res = selection.split(" ");

    jTextField15.setText(res[0].trim());
}
else
{
    JOptionPane.showMessageDialog(mainPanel,"Can't Find a Matching
Procedure");

    jTextField15.setText("");
}
stmt.close();

//System.out.println ("code:"+res[0].trim());

} catch (SQLException ex) {
    Logger.getLogger(POETView.class.getName()).log(Level.SEVERE, null, ex);
}

// TODO add your handling code here:
}
private void helpMenuActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:
    JOptionPane.showMessageDialog(mainPanel,"POET HELP");

}
//}
// Variables declaration - do not modify
private javax.swing.ButtonGroup buttonGroup1;
private javax.swing.ButtonGroup buttonGroup10;

```

```
private javax.swing.ButtonGroup buttonGroup11;
private javax.swing.ButtonGroup buttonGroup2;
private javax.swing.ButtonGroup buttonGroup3;
private javax.swing.ButtonGroup buttonGroup4;
private javax.swing.ButtonGroup buttonGroup5;
private javax.swing.ButtonGroup buttonGroup6;
private javax.swing.ButtonGroup buttonGroup7;
private javax.swing.ButtonGroup buttonGroup8;
private javax.swing.ButtonGroup buttonGroup9;
private javax.swing.JButton jButton1;
private javax.swing.JButton jButton10;
private javax.swing.JButton jButton2;
private javax.swing.JButton jButton3;
private javax.swing.JButton jButton4;
private javax.swing.JButton jButton5;
private javax.swing.JButton jButton9;
private javax.swing.JComboBox jComboBox1;
private javax.swing.JComboBox jComboBox2;
private javax.swing.JComboBox jComboBox3;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel10;
private javax.swing.JLabel jLabel12;
private javax.swing.JLabel jLabel13;
private javax.swing.JLabel jLabel14;
private javax.swing.JLabel jLabel15;
private javax.swing.JLabel jLabel16;
private javax.swing.JLabel jLabel17;
private javax.swing.JLabel jLabel18;
private javax.swing.JLabel jLabel19;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel20;
private javax.swing.JLabel jLabel21;
private javax.swing.JLabel jLabel22;
private javax.swing.JLabel jLabel23;
private javax.swing.JLabel jLabel24;
private javax.swing.JLabel jLabel25;
private javax.swing.JLabel jLabel26;
private javax.swing.JLabel jLabel27;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;
private javax.swing.JLabel jLabel6;
private javax.swing.JLabel jLabel7;
private javax.swing.JLabel jLabel8;
private javax.swing.JLabel jLabel9;
private javax.swing.JList jList14;
private javax.swing.JList jList15;
private javax.swing.JPanel jPanel1;
```

```
private javax.swing.JPasswordField jPasswordField1;
private javax.swing.JRadioButton jRadioButton1;
private javax.swing.JRadioButton jRadioButton10;
private javax.swing.JRadioButton jRadioButton11;
private javax.swing.JRadioButton jRadioButton12;
private javax.swing.JRadioButton jRadioButton13;
private javax.swing.JRadioButton jRadioButton14;
private javax.swing.JRadioButton jRadioButton15;
private javax.swing.JRadioButton jRadioButton16;
private javax.swing.JRadioButton jRadioButton17;
private javax.swing.JRadioButton jRadioButton18;
private javax.swing.JRadioButton jRadioButton19;
private javax.swing.JRadioButton jRadioButton2;
private javax.swing.JRadioButton jRadioButton26;
private javax.swing.JRadioButton jRadioButton27;
private javax.swing.JRadioButton jRadioButton28;
private javax.swing.JRadioButton jRadioButton29;
private javax.swing.JRadioButton jRadioButton3;
private javax.swing.JRadioButton jRadioButton4;
private javax.swing.JRadioButton jRadioButton5;
private javax.swing.JRadioButton jRadioButton6;
private javax.swing.JRadioButton jRadioButton7;
private javax.swing.JRadioButton jRadioButton8;
private javax.swing.JRadioButton jRadioButton9;
private javax.swing.JScrollPane jScrollPane18;
private javax.swing.JScrollPane jScrollPane19;
private javax.swing.JTextField jTextField1;
private javax.swing.JTextField jTextField14;
private javax.swing.JTextField jTextField15;
private javax.swing.JTextField jTextField16;
private javax.swing.JTextField jTextField17;
private javax.swing.JTextField jTextField2;
private javax.swing.JTextField jTextField3;
private javax.swing.JTextField jTextField4;
private javax.swing.JTextField jTextField5;
private javax.swing.JTextField jTextField6;
private javax.swing.JTextField jTextField7;
private javax.swing.JPanel mainPanel;
private javax.swing.JMenuBar menuBar;
private javax.swing.JProgressBar progressBar;
private javax.swing.JLabel statusAnimationLabel;
private javax.swing.JLabel statusMessageLabel;
private javax.swing.JPanel statusPanel;
// End of variables declaration
private final Timer messageTimer;
private final Timer busyIconTimer;
private final Icon idleIcon;
private final Icon[] busyIcons = new Icon[15];
```

```

private int busyIconIndex = 0;

private JDialog aboutBox;
}

```

A2. Module for ICD-9-CM Code Conversion

/*****
 The SQL function assumes that the ICD-9-CM conversion rules provided by ICD-9-CM
 Coordination and Maintenance Committee (CMC) in the following link are already
 populated in the database:

Centers for Disease Control and Prevention, National Center for Health Statistics.
 Conversion table of new ICD-9-CM Codes, Oct 2008 Available at:
http://www.cdc.gov/nchs/icd/icd9cm_addenda_guidelines.htm. Accessed Sept 16, 2012

This function will be used during the population of Layer 2 and layer of POET's pyramid
 data architecture

Author: Ahmad AbuSalah

*****/

```

create or replace type v_array is table of varchar2(100);

```

```

set serveroutput on

```

```

CREATE OR REPLACE FUNCTION get_equiv
  (v_year IN varchar2, in_array IN v_array , v_flag IN varchar2)
  RETURN v_array
IS --| AS
  equiv_array v_array:=v_array();
  pr_cnt number;
  dx_cnt number;

```

```

i number:= 0;
BEGIN
if v_flag='p' then
  for elem in 1 .. in_array.count loop
    --dbms_output.put_line('orig_elem' || ':' || in_array(elem));

    select count(distinct pr_cur)
    INTO pr_cnt
    from pr_conv_rules
    where pr_year = v_year
    and replace(trim(pr_prev),',') = trim (in_array(elem));

    if pr_cnt =0 then
      equiv_array.extend();
      i:= i+1;
      equiv_array(i):= in_array(elem);

    else

      select distinct replace(trim(pr_cur),',')
      bulk collect
      into equiv_array
      from pr_conv_rules
      where pr_year = v_year
      and replace(trim(pr_prev),',') = trim (in_array(elem));

    end if;

  end loop;

else --v_flag ='d'
  for elem in 1 .. in_array.count loop
--    dbms_output.put_line('orig_elem' || ':' || in_array(elem));

    select count(distinct dx_cur)
    INTO dx_cnt
    from dx_conv_rules
    where dx_year = v_year
    and replace(trim(dx_prev),',') = trim (in_array(elem));

    if dx_cnt =0 then
      equiv_array.extend();
      i:= i+1;
      equiv_array(i):= in_array(elem);

    else

```

```

--dbms_output.put_line ('Exception!!!!');

select distinct replace(trim(dx_cur),',','')
bulk collect
into equiv_array
from dx_conv_rules
where dx_year = v_year
and replace(trim(dx_prev),',','') = trim (in_array(elem));

end if;
end loop;

end if; -- v_flag = 'p'
return equiv_array;

EXCEPTION WHEN OTHERS THEN
dbms_output.put_line ('Exception!!!!');
END get_equiv;

```

A3. Module to Populate the Personalized Risk Layer and Average-Risk Layer of POET's Database.

/*****
 The SQL code assumes that the bottom layer of the data tier is already populate with NIS data.

The code also uses function get_equiv which is responsible of matching the procedure and diagnosis ICD-9-CM codes for the 5 years of NIS data used by the POET system.

Author: Ahmad AbuSalah

```

*****/
/*
SET serveroutput on

BEGIN

--populate pat_pr
for pat_pr in (select nc_8.key, dqtr, age_num, female, race, aweekend, elective, pr1, pr2
, pr3, dx1, died, LOS, dispuniform, CM_AIDS, CM_ALCOHOL, CM_ANEMDEF,
CM_ARTH,CM_BLDLOSS,CM_CHF,CM_CHRNLUNG,CM_COAG,CM_DEPRESS,CM_DM,
CM_DMCX, CM_DRUG, CM_HTN_C, CM_HYPOTHY, CM_LIVER, CM_LYMPH, CM_LYTES,
CM_METS, CM_NEURO, CM_OBESE, CM_PARA, CM_PERIVASC,
CM_PSYCH,CM_PULMCIRC,CM_RENLFAIL,CM_TUMOR,CM_ULCER,CM_VALVE,CM_WG
HTLOSS
from nis_2008_core nc_8, nis_2008_severity ns_8 where nc_8.key = ns_8.key)

```


loop

```
if substr(pat_pr.pr1,1,1) IN ('0','1','2') then
  insert into pat_pr_0_2 (key,year,dqtr, age, female, race, weekend_adm,
  elec_adm,pr1,pr2,pr3,dx1, o_pr1,o_pr2,o_pr3,o_dx1, died, LOS, dispo)
  values(pat_pr.key,'2008', pat_pr.dqtr, pat_pr.age_num, pat_pr.female, pat_pr.race,
  pat_pr.aweekend,
  pat_pr.elective,pat_pr.pr1,pat_pr.pr2,pat_pr.pr3,pat_pr.dx1,pat_pr.pr1,pat_pr.pr2,pat_
  pr.pr3,pat_pr.dx1,pat_pr.died,decode(pat_pr.LOS,null,null,'A',null,'B',null,'C',null,to_nu
  mber(pat_pr.LOS)),pat_pr.dispuniform);
```

```
insert into pat_como_0_2
```

```
(key,CM_AIDS,CM_ALCOHOL,CM_ANEMDEF,CM_ARTH,CM_BLDLOSS,CM_CHF,CM_CHR
NLUNG,CM_COAG,CM_DEPRESS,CM_DM,
```

```
CM_DMCX,CM_DRUG,CM_HTN_C,CM_HYPOTHY,CM_LIVER,CM_LYMPH,CM_LYTES,CM_
METS,CM_NEURO,CM_OBESE,CM_PARA,CM_PERIVASC,
```

```
CM_PSYCH,CM_PULMCIRC,CM_RENLFAIL,CM_TUMOR,CM_ULCER,CM_VALVE,CM_WG
HTLOSS)
```

```
VALUES
```

```
(pat_pr.key,pat_pr.CM_AIDS,pat_pr.CM_ALCOHOL,pat_pr.CM_ANEMDEF,pat_pr.CM_A
RTH,pat_pr.CM_BLDLOSS,pat_pr.CM_CHF,pat_pr.CM_CHRNLUNG,pat_pr.CM_COAG,pat
_pr.CM_DEPRESS,pat_pr.CM_DM,
```

```
pat_pr.CM_DMCX,pat_pr.CM_DRUG,pat_pr.CM_HTN_C,pat_pr.CM_HYPOTHY,pat_pr.C
M_LIVER,pat_pr.CM_LYMPH,pat_pr.CM_LYTES,pat_pr.CM_METS,pat_pr.CM_NEURO,pa
t_pr.CM_OBESE,pat_pr.CM_PARA,
```

```
pat_pr.CM_PERIVASC,pat_pr.CM_PSYCH,pat_pr.CM_PULMCIRC,pat_pr.CM_RENLFAIL,p
at_pr.CM_TUMOR,pat_pr.CM_ULCER,pat_pr.CM_VALVE,pat_pr.CM_WGHTLOSS);
```

```
elsif substr(pat_pr.pr1,1,1) = '3' then
```

```
insert into pat_pr_3 (key,year,dqtr, age, female, race, weekend_adm,
elec_adm,pr1,pr2,pr3,dx1, o_pr1,o_pr2,o_pr3,o_dx1, died, LOS, dispo)
values(pat_pr.key,'2008', pat_pr.dqtr, pat_pr.age_num, pat_pr.female, pat_pr.race,
pat_pr.aweekend,
pat_pr.elective,pat_pr.pr1,pat_pr.pr2,pat_pr.pr3,pat_pr.dx1,pat_pr.pr1,pat_pr.pr2,pat_
pr.pr3,pat_pr.dx1,pat_pr.died,decode(pat_pr.LOS,null,null,'A',null,'B',null,'C',null,to_nu
mber(pat_pr.LOS)),pat_pr.dispuniform);
```

```
insert into pat_como_3
```

```
(key,CM_AIDS,CM_ALCOHOL,CM_ANEMDEF,CM_ARTH,CM_BLDLOSS,CM_CHF,CM_CHR
NLUNG,CM_COAG,CM_DEPRESS,CM_DM,
```

```
CM_DMCX,CM_DRUG,CM_HTN_C,CM_HYPOTHY,CM_LIVER,CM_LYMPH,CM_LYTES,CM_
METS,CM_NEURO,CM_OBESE,CM_PARA,CM_PERIVASC,
```

```
CM_PSYCH,CM_PULMCIRC,CM_RENLFAIL,CM_TUMOR,CM_ULCER,CM_VALVE,CM_WG
HTLOSS)
```

```
VALUES
```

```
(pat_pr.key,pat_pr.CM_AIDS,pat_pr.CM_ALCOHOL,pat_pr.CM_ANEMDEF,pat_pr.CM_A
RTH,pat_pr.CM_BLDLOSS,pat_pr.CM_CHF,pat_pr.CM_CHRNLUNG,pat_pr.CM_COAG,pat
_pr.CM_DEPRESS,pat_pr.CM_DM,
```

```
pat_pr.CM_DMCX,pat_pr.CM_DRUG,pat_pr.CM_HTN_C,pat_pr.CM_HYPOTHY,pat_pr.C
M_LIVER,pat_pr.CM_LYMPH,pat_pr.CM_LYTES,pat_pr.CM_METS,pat_pr.CM_NEURO,pa
t_pr.CM_OBESE,pat_pr.CM_PARA,
```

```
pat_pr.CM_PERIVASC,pat_pr.CM_PSYCH,pat_pr.CM_PULMCIRC,pat_pr.CM_RENLFAIL,p
at_pr.CM_TUMOR,pat_pr.CM_ULCER,pat_pr.CM_VALVE,pat_pr.CM_WGHTLOSS);
```

```
elsif substr(pat_pr.pr1,1,1) = '4' then
```

```
insert into pat_pr_4 (key,year,dqtr, age, female, race, weekend_adm,
elec_adm,pr1,pr2,pr3,dx1, o_pr1,o_pr2,o_pr3,o_dx1, died, LOS, dispo)
values(pat_pr.key,'2008', pat_pr.dqtr, pat_pr.age_num, pat_pr.female, pat_pr.race,
pat_pr.aweekend,
pat_pr.elective,pat_pr.pr1,pat_pr.pr2,pat_pr.pr3,pat_pr.dx1,pat_pr.pr1,pat_pr.pr2,pat_
pr.pr3,pat_pr.dx1,pat_pr.died,decode(pat_pr.LOS,null,null,'A',null,'B',null,'C',null,to_nu
mber(pat_pr.LOS)),pat_pr.dispuniform);
```

```
insert into pat_como_4
```

```
(key,CM_AIDS,CM_ALCOHOL,CM_ANEMDEF,CM_ARTH,CM_BLDLOSS,CM_CHF,CM_CHR
NLUNG,CM_COAG,CM_DEPRESS,CM_DM,
```

```
CM_DMCX,CM_DRUG,CM_HTN_C,CM_HYPOTHY,CM_LIVER,CM_LYMPH,CM_LYTES,CM_
METS,CM_NEURO,CM_OBESE,CM_PARA,CM_PERIVASC,
```

```
CM_PSYCH,CM_PULMCIRC,CM_RENLFAIL,CM_TUMOR,CM_ULCER,CM_VALVE,CM_WG
HTLOSS)
```

```
VALUES
```

```
(pat_pr.key,pat_pr.CM_AIDS,pat_pr.CM_ALCOHOL,pat_pr.CM_ANEMDEF,pat_pr.CM_A
RTH,pat_pr.CM_BLDLOSS,pat_pr.CM_CHF,pat_pr.CM_CHRNLUNG,pat_pr.CM_COAG,pat
_pr.CM_DEPRESS,pat_pr.CM_DM,
```

```
pat_pr.CM_DMCX,pat_pr.CM_DRUG,pat_pr.CM_HTN_C,pat_pr.CM_HYPOTHY,pat_pr.C
M_LIVER,pat_pr.CM_LYMPH,pat_pr.CM_LYTES,pat_pr.CM_METS,pat_pr.CM_NEURO,pa
t_pr.CM_OBESE,pat_pr.CM_PARA,
```

```
pat_pr.CM_PERIVASC,pat_pr.CM_PSYCH,pat_pr.CM_PULMCIRC,pat_pr.CM_RENLFAIL,pat_pr.CM_TUMOR,pat_pr.CM_ULCER,pat_pr.CM_VALVE,pat_pr.CM_WGHTLOSS);
```

```
elseif substr(pat_pr.pr1,1,1) = '5' then
  insert into pat_pr_5 (key,year,dqtr, age, female, race, weekend_adm,
  elec_adm,pr1,pr2,pr3,dx1, o_pr1,o_pr2,o_pr3,o_dx1, died, LOS, dispo)
  values(pat_pr.key,'2008', pat_pr.dqtr, pat_pr.age_num, pat_pr.female, pat_pr.race,
  pat_pr.aweekend,
  pat_pr.elective,pat_pr.pr1,pat_pr.pr2,pat_pr.pr3,pat_pr.dx1,pat_pr.pr1,pat_pr.pr2,pat_pr.pr3,pat_pr.dx1,pat_pr.died,decode(pat_pr.LOS,null,null,'A',null,'B',null,'C',null,to_number(pat_pr.LOS)),pat_pr.dispuniform);
```

```
insert into pat_como_5
```

```
(key,CM_AIDS,CM_ALCOHOL,CM_ANEMDEF,CM_ARTH,CM_BLDLOSS,CM_CHF,CM_CHRNLUNG,CM_COAG,CM_DEPRESS,CM_DM,
```

```
CM_DMCX,CM_DRUG,CM_HTN_C,CM_HYPOTHY,CM_LIVER,CM_LYMPH,CM_LYTES,CM_METS,CM_NEURO,CM_OBESE,CM_PARA,CM_PERIVASC,
```

```
CM_PSYCH,CM_PULMCIRC,CM_RENLFAIL,CM_TUMOR,CM_ULCER,CM_VALVE,CM_WGHTLOSS)
```

```
VALUES
```

```
(pat_pr.key,pat_pr.CM_AIDS,pat_pr.CM_ALCOHOL,pat_pr.CM_ANEMDEF,pat_pr.CM_ARTH,pat_pr.CM_BLDLOSS,pat_pr.CM_CHF,pat_pr.CM_CHRNLUNG,pat_pr.CM_COAG,pat_pr.CM_DEPRESS,pat_pr.CM_DM,
```

```
pat_pr.CM_DMCX,pat_pr.CM_DRUG,pat_pr.CM_HTN_C,pat_pr.CM_HYPOTHY,pat_pr.CM_LIVER,pat_pr.CM_LYMPH,pat_pr.CM_LYTES,pat_pr.CM_METS,pat_pr.CM_NEURO,pat_pr.CM_OBESE,pat_pr.CM_PARA,
```

```
pat_pr.CM_PERIVASC,pat_pr.CM_PSYCH,pat_pr.CM_PULMCIRC,pat_pr.CM_RENLFAIL,pat_pr.CM_TUMOR,pat_pr.CM_ULCER,pat_pr.CM_VALVE,pat_pr.CM_WGHTLOSS);
```

```
elseif substr(pat_pr.pr1,1,1) = '6' then
  insert into pat_pr_6 (key,year,dqtr, age, female, race, weekend_adm,
  elec_adm,pr1,pr2,pr3,dx1, o_pr1,o_pr2,o_pr3,o_dx1, died, LOS, dispo)
  values(pat_pr.key,'2008', pat_pr.dqtr, pat_pr.age_num, pat_pr.female, pat_pr.race,
  pat_pr.aweekend,
  pat_pr.elective,pat_pr.pr1,pat_pr.pr2,pat_pr.pr3,pat_pr.dx1,pat_pr.pr1,pat_pr.pr2,pat_pr.pr3,pat_pr.dx1,pat_pr.died,decode(pat_pr.LOS,null,null,'A',null,'B',null,'C',null,to_number(pat_pr.LOS)),pat_pr.dispuniform);
```

```
insert into pat_como_6
```

```
(key,CM_AIDS,CM_ALCOHOL,CM_ANEMDEF,CM_ARTH,CM_BLDLOSS,CM_CHF,CM_CHRNLUNG,CM_COAG,CM_DEPRESS,CM_DM,
```

```
CM_DMCX,CM_DRUG,CM_HTN_C,CM_HYPOTHY,CM_LIVER,CM_LYMPH,CM_LYTES,CM_METS,CM_NEURO,CM_OBESE,CM_PARA,CM_PERIVASC,
```

```
CM_PSYCH,CM_PULMCIRC,CM_RENLFAIL,CM_TUMOR,CM_ULCER,CM_VALVE,CM_WGHTLOSS)
```

```
VALUES
```

```
(pat_pr.key,pat_pr.CM_AIDS,pat_pr.CM_ALCOHOL,pat_pr.CM_ANEMDEF,pat_pr.CM_ARTH,pat_pr.CM_BLDLOSS,pat_pr.CM_CHF,pat_pr.CM_CHRNLUNG,pat_pr.CM_COAG,pat_pr.CM_DEPRESS,pat_pr.CM_DM,
```

```
pat_pr.CM_DMCX,pat_pr.CM_DRUG,pat_pr.CM_HTN_C,pat_pr.CM_HYPOTHY,pat_pr.CM_LIVER,pat_pr.CM_LYMPH,pat_pr.CM_LYTES,pat_pr.CM_METS,pat_pr.CM_NEURO,pat_pr.CM_OBESE,pat_pr.CM_PARA,
```

```
pat_pr.CM_PERIVASC,pat_pr.CM_PSYCH,pat_pr.CM_PULMCIRC,pat_pr.CM_RENLFAIL,pat_pr.CM_TUMOR,pat_pr.CM_ULCER,pat_pr.CM_VALVE,pat_pr.CM_WGHTLOSS);
```

```
elsif substr(pat_pr.pr1,1,1) = '7' then
```

```
insert into pat_pr_7 (key,year,dqtr, age, female, race, weekend_adm, elec_adm,pr1,pr2,pr3,dx1, o_pr1,o_pr2,o_pr3,o_dx1, died, LOS, dispo)
```

```
values(pat_pr.key,'2008', pat_pr.dqtr, pat_pr.age_num, pat_pr.female, pat_pr.race, pat_pr.aweekend,
```

```
pat_pr.elective,pat_pr.pr1,pat_pr.pr2,pat_pr.pr3,pat_pr.dx1,pat_pr.pr1,pat_pr.pr2,pat_pr.pr3,pat_pr.dx1,pat_pr.died,decode(pat_pr.LOS,null,null,'A',null,'B',null,'C',null,to_number(pat_pr.LOS)),pat_pr.dispuniform);
```

```
insert into pat_como_7
```

```
(key,CM_AIDS,CM_ALCOHOL,CM_ANEMDEF,CM_ARTH,CM_BLDLOSS,CM_CHF,CM_CHRNLUNG,CM_COAG,CM_DEPRESS,CM_DM,
```

```
CM_DMCX,CM_DRUG,CM_HTN_C,CM_HYPOTHY,CM_LIVER,CM_LYMPH,CM_LYTES,CM_METS,CM_NEURO,CM_OBESE,CM_PARA,CM_PERIVASC,
```

```
CM_PSYCH,CM_PULMCIRC,CM_RENLFAIL,CM_TUMOR,CM_ULCER,CM_VALVE,CM_WGHTLOSS)
```

```
VALUES
```

```
(pat_pr.key,pat_pr.CM_AIDS,pat_pr.CM_ALCOHOL,pat_pr.CM_ANEMDEF,pat_pr.CM_ARTH,pat_pr.CM_BLDLOSS,pat_pr.CM_CHF,pat_pr.CM_CHRNLUNG,pat_pr.CM_COAG,pat_pr.CM_DEPRESS,pat_pr.CM_DM,
```

```
pat_pr.CM_DMCX,pat_pr.CM_DRUG,pat_pr.CM_HTN_C,pat_pr.CM_HYPOTHY,pat_pr.C
```

```
M_LIVER,pat_pr.CM_LYMPH,pat_pr.CM_LYTES,pat_pr.CM_METS,pat_pr.CM_NEURO,pat_pr.CM_OBESE,pat_pr.CM_PARA,
```

```
pat_pr.CM_PERIVASC,pat_pr.CM_PSYCH,pat_pr.CM_PULMCIRC,pat_pr.CM_RENLFAIL,pat_pr.CM_TUMOR,pat_pr.CM_ULCER,pat_pr.CM_VALVE,pat_pr.CM_WGHTLOSS);
```

```
elseif substr(pat_pr.pr1,1,1) = '8' then
```

```
  insert into pat_pr_8 (key,year,dqtr, age, female, race, weekend_adm,
    elec_adm,pr1,pr2,pr3,dx1, o_pr1,o_pr2,o_pr3,o_dx1, died, LOS, dispo)
    values(pat_pr.key,'2008', pat_pr.dqtr, pat_pr.age_num, pat_pr.female, pat_pr.race,
    pat_pr.aweekend,
    pat_pr.elective,pat_pr.pr1,pat_pr.pr2,pat_pr.pr3,pat_pr.dx1,pat_pr.pr1,pat_pr.pr2,pat_pr.pr3,pat_pr.dx1,pat_pr.died,decode(pat_pr.LOS,null,null,'A',null,'B',null,'C',null,to_number(pat_pr.LOS)),pat_pr.dispuniform);
```

```
  insert into pat_como_8
```

```
(key,CM_AIDS,CM_ALCOHOL,CM_ANEMDEF,CM_ARTH,CM_BLDLOSS,CM_CHF,CM_CHRNLUNG,CM_COAG,CM_DEPRESS,CM_DM,
```

```
CM_DMCX,CM_DRUG,CM_HTN_C,CM_HYPOTHY,CM_LIVER,CM_LYMPH,CM_LYTES,CM_METS,CM_NEURO,CM_OBESE,CM_PARA,CM_PERIVASC,
```

```
CM_PSYCH,CM_PULMCIRC,CM_RENLFAIL,CM_TUMOR,CM_ULCER,CM_VALVE,CM_WGHTLOSS)
```

```
  VALUES
```

```
(pat_pr.key,pat_pr.CM_AIDS,pat_pr.CM_ALCOHOL,pat_pr.CM_ANEMDEF,pat_pr.CM_ARTH,pat_pr.CM_BLDLOSS,pat_pr.CM_CHF,pat_pr.CM_CHRNLUNG,pat_pr.CM_COAG,pat_pr.CM_DEPRESS,pat_pr.CM_DM,
```

```
pat_pr.CM_DMCX,pat_pr.CM_DRUG,pat_pr.CM_HTN_C,pat_pr.CM_HYPOTHY,pat_pr.CM_LIVER,pat_pr.CM_LYMPH,pat_pr.CM_LYTES,pat_pr.CM_METS,pat_pr.CM_NEURO,pat_pr.CM_OBESE,pat_pr.CM_PARA,
```

```
pat_pr.CM_PERIVASC,pat_pr.CM_PSYCH,pat_pr.CM_PULMCIRC,pat_pr.CM_RENLFAIL,pat_pr.CM_TUMOR,pat_pr.CM_ULCER,pat_pr.CM_VALVE,pat_pr.CM_WGHTLOSS);
```

```
elseif substr(pat_pr.pr1,1,1) = '9' then
```

```
  insert into pat_pr_9 (key,year,dqtr, age, female, race, weekend_adm,
    elec_adm,pr1,pr2,pr3,dx1, o_pr1,o_pr2,o_pr3,o_dx1, died, LOS, dispo)
    values(pat_pr.key,'2008', pat_pr.dqtr, pat_pr.age_num, pat_pr.female, pat_pr.race,
    pat_pr.aweekend,
    pat_pr.elective,pat_pr.pr1,pat_pr.pr2,pat_pr.pr3,pat_pr.dx1,pat_pr.pr1,pat_pr.pr2,pat_pr.pr3,pat_pr.dx1,pat_pr.died,decode(pat_pr.LOS,null,null,'A',null,'B',null,'C',null,to_number(pat_pr.LOS)),pat_pr.dispuniform);
```

```
  insert into pat_como_9
```

```

(key,CM_AIDS,CM_ALCOHOL,CM_ANEMDEF,CM_ARTH,CM_BLDLOSS,CM_CHF,CM_CHR
NLUNG,CM_COAG,CM_DEPRESS,CM_DM,

CM_DMCX,CM_DRUG,CM_HTN_C,CM_HYPOTHY,CM_LIVER,CM_LYMPH,CM_LYTES,CM_
METS,CM_NEURO,CM_OBESE,CM_PARA,CM_PERIVASC,

CM_PSYCH,CM_PULMCIRC,CM_RENLFAIL,CM_TUMOR,CM_ULCER,CM_VALVE,CM_WG
HTLOSS)
VALUES

(pat_pr.key,pat_pr.CM_AIDS,pat_pr.CM_ALCOHOL,pat_pr.CM_ANEMDEF,pat_pr.CM_A
RTH,pat_pr.CM_BLDLOSS,pat_pr.CM_CHF,pat_pr.CM_CHRNLUNG,pat_pr.CM_COAG,pat_
_pr.CM_DEPRESS,pat_pr.CM_DM,

pat_pr.CM_DMCX,pat_pr.CM_DRUG,pat_pr.CM_HTN_C,pat_pr.CM_HYPOTHY,pat_pr.C
M_LIVER,pat_pr.CM_LYMPH,pat_pr.CM_LYTES,pat_pr.CM_METS,pat_pr.CM_NEURO,pa
t_pr.CM_OBESE,pat_pr.CM_PARA,

pat_pr.CM_PERIVASC,pat_pr.CM_PSYCH,pat_pr.CM_PULMCIRC,pat_pr.CM_RENLFAIL,p
at_pr.CM_TUMOR,pat_pr.CM_ULCER,pat_pr.CM_VALVE,pat_pr.CM_WGHTLOSS);

end if;

end loop;

commit;
END;

/
*/
/*
create table debug_script
(script_name varchar2(1000),
comments varchar2(1000),
time_stamp timestamp
)
tablespace bmhi;
*/
/*
SET serveroutput on

declare
equiv_pr1 v_array:=v_array();
orig_pr1 v_array:=v_array();

```

```

equiv_pr2 v_array:= v_array();
orig_pr2 v_array:=v_array();

equiv_pr3 v_array:= v_array();
orig_pr3 v_array:=v_array();

equiv_dx1 v_array:= v_array();
orig_dx1 v_array:=v_array();

v_pr2_cnt NUMBER(10):=0;
v_pr3_cnt NUMBER(10):=0;
v_dx1_cnt NUMBER(10):=0;
i number(10):=0;

BEGIN

--populate pat_pr
for pat_pr in (select nc_7.key, dqtr, age_num, female, race, aweekend,
elective,pr1,pr2,pr3,dx1,died,LOS,dispuniform,

CM_AIDS,CM_ALCOHOL,CM_ANEMDEF,CM_ARTH,CM_BLDLOSS,CM_CHF,CM_CHRNLU
NG,CM_COAG,CM_DEPRESS,CM_DM,

CM_DMCX,CM_DRUG,CM_HTN_C,CM_HYPOTHY,CM_LIVER,CM_LYMPH,CM_LYTES,CM_
METS,CM_NEURO,CM_OBESE,CM_PARA,CM_PERIVASC,

CM_PSYCH,CM_PULMCIRC,CM_RENLFAIL,CM_TUMOR,CM_ULCER,CM_VALVE,CM_WG
HTLOSS
from nis_2007_core nc_7, nis_2007_severity ns_7 where nc_7.key = ns_7.key
and pr1 is not null and pr1 !='A' and pr1 !='B' and pr1 !='C')
loop
  i:= i+1;
  if i = 100 then
    insert into debug_script values('2007 data','100 records',SYSTIMESTAMP);
    i:= 0;
  end if;

  if (pat_pr.dqtr ='1') OR (pat_pr.dqtr ='2') OR (pat_pr.dqtr ='3')
  then
    -- get pr1 equiv
    orig_pr1.delete;
    equiv_pr1.delete;
    orig_pr1.extend;
    orig_pr1(1):= pat_pr.pr1;
    equiv_pr1 := get_equiv('2007',orig_pr1,'p');

-- for elem in 1 .. equiv_pr1.count loop

```

```

--      dbms_output.put_line('pr1' || ':' || pat_pr.pr1||', equiv_pr1' || ':' ||
trim(equiv_pr1(elem)));
-- end loop;

-- dbms_output.put_line('pr2' || ':' || pat_pr.pr2);
-- get pr2 equiv
if(pat_pr.pr2 is not null) and (pat_pr.pr2 !='A') and (pat_pr.pr2 !='B') and (pat_pr.pr2
!='C')
THEN
orig_pr2.delete;
equiv_pr2.delete;
orig_pr2.extend;
orig_pr2(1):= pat_pr.pr2;
equiv_pr2 := get_equiv('2007',orig_pr2,'p');

else
orig_pr2.delete;
equiv_pr2.delete;
equiv_pr2.extend;
equiv_pr2(1):= 'AAA';

end if; -- pr1 is null

-- for elem in 1 .. equiv_pr2.count loop
--      dbms_output.put_line('pr2' || ':' || pat_pr.pr2||', equiv_pr2' || ':' ||
trim(equiv_pr2(elem)));
-- end loop;

-- dbms_output.put_line('pr3' || ':' || pat_pr.pr3);
-- get pr3 equiv
if(pat_pr.pr3 is not null) and (pat_pr.pr3 !='A') and (pat_pr.pr3 !='B') and (pat_pr.pr3
!='C')
THEN
orig_pr3.delete;
equiv_pr3.delete;
orig_pr3.extend;
orig_pr3(1):= pat_pr.pr3;
equiv_pr3 := get_equiv('2007',orig_pr3,'p');

else
orig_pr3.delete;
equiv_pr3.delete;
equiv_pr3.extend;
equiv_pr3(1):= 'AAA';

end if; -- pr3 is null

-- for elem in 1 .. equiv_pr3.count loop

```



```

--      dbms_output.put_line('pr3' || ':' || pat_pr.pr3 || ', equiv_pr3' || ':' ||
trim(equiv_pr3(elem)));
--      end loop;

-- dbms_output.put_line('dx1' || ':' || pat_pr.dx1);
-- get dx1 equiv
if(pat_pr.dx1 is not null) and (pat_pr.dx1 !='A') and (pat_pr.dx1 !='B') and (pat_pr.dx1
!='C')
THEN
  orig_dx1.delete;
  equiv_dx1.delete;
  orig_dx1.extend;
  orig_dx1(1):= pat_pr.dx1;
  equiv_dx1 := get_equiv('2007',orig_dx1,'d');

else
  orig_dx1.delete;
  equiv_dx1.delete;
  equiv_dx1.extend;
  equiv_dx1(1):= 'AAA';

end if; -- dx1 is null
--- for elem in 1 .. equiv_dx1.count loop
--      dbms_output.put_line('$$$ equiv_elem_dx1' || ':' || equiv_dx1(elem));-
--      end loop;

for elem1 in 1 .. equiv_pr1.count loop
  for elem2 in 1 .. equiv_pr2.count loop
    for elem3 in 1 .. equiv_pr3.count loop
      for elemdx in 1 .. equiv_dx1.count loop

if substr(equiv_pr1(elem1),1,1) IN ('0','1','2') then
  insert into pat_pr_0_2 (key,year,dqtr, age, female, race, weekend_adm,
elec_adm,pr1,pr2,pr3,dx1, o_pr1,o_pr2,o_pr3,o_dx1, died, LOS, dispo)
  values(pat_pr.key,'2007', pat_pr.dqtr, pat_pr.age_num, pat_pr.female, pat_pr.race,
pat_pr.aweekend,
pat_pr.elective,trim(equiv_pr1(elem1)),trim(replace(equiv_pr2(elem2),'AAA','')),trim(re
place(equiv_pr3(elem3),'AAA','')),trim(replace(equiv_dx1(elemdx),'AAA','')),pat_pr.pr1,p
at_pr.pr2,pat_pr.pr3,pat_pr.dx1,pat_pr.died,decode(pat_pr.LOS,null,null,'A',null,'B',null,
'C',null,to_number(pat_pr.LOS)),pat_pr.dispuniform);

insert into pat_como_0_2

(key,CM_AIDS,CM_ALCOHOL,CM_ANEMDEF,CM_ARTH,CM_BLDLOSS,CM_CHF,CM_CHR
NLUNG,CM_COAG,CM_DEPRESS,CM_DM,

```

```
CM_DMCX,CM_DRUG,CM_HTN_C,CM_HYPOTHY,CM_LIVER,CM_LYMPH,CM_LYTES,CM_
METS,CM_NEURO,CM_OBESE,CM_PARA,CM_PERIVASC,
```

```
CM_PSYCH,CM_PULMCIRC,CM_RENLFAIL,CM_TUMOR,CM_ULCER,CM_VALVE,CM_WG
HTLOSS)
```

```
VALUES
```

```
(pat_pr.key,pat_pr.CM_AIDS,pat_pr.CM_ALCOHOL,pat_pr.CM_ANEMDEF,pat_pr.CM_A
RTH,pat_pr.CM_BLDLOSS,pat_pr.CM_CHF,pat_pr.CM_CHRNLUNG,pat_pr.CM_COAG,pat
_pr.CM_DEPRESS,pat_pr.CM_DM,
```

```
pat_pr.CM_DMCX,pat_pr.CM_DRUG,pat_pr.CM_HTN_C,pat_pr.CM_HYPOTHY,pat_pr.C
M_LIVER,pat_pr.CM_LYMPH,pat_pr.CM_LYTES,pat_pr.CM_METS,pat_pr.CM_NEURO,pa
t_pr.CM_OBESE,pat_pr.CM_PARA,
```

```
pat_pr.CM_PERIVASC,pat_pr.CM_PSYCH,pat_pr.CM_PULMCIRC,pat_pr.CM_RENLFAIL,p
at_pr.CM_TUMOR,pat_pr.CM_ULCER,pat_pr.CM_VALVE,pat_pr.CM_WGHTLOSS);
```

```
elseif substr(equiv_pr1(elem1),1,1) = '3' then
  insert into pat_pr_3 (key,year,dqtr, age, female, race, weekend_adm,
elec_adm,pr1,pr2,pr3,dx1, o_pr1,o_pr2,o_pr3,o_dx1, died, LOS, dispo)
  values(pat_pr.key,'2007', pat_pr.dqtr, pat_pr.age_num, pat_pr.female, pat_pr.race,
pat_pr.aweekend,
pat_pr.elective,trim(equiv_pr1(elem1)),trim(replace(equiv_pr2(elem2),'AAA','')),trim(re
place(equiv_pr3(elem3),'AAA','')),trim(replace(equiv_dx1(elemdx),'AAA','')),pat_pr.pr1,p
at_pr.pr2,pat_pr.pr3,pat_pr.dx1,pat_pr.died,decode(pat_pr.LOS,null,null,'A',null,'B',null,
'C',null,to_number(pat_pr.LOS)),pat_pr.dispuniform);
```

```
insert into pat_como_3
```

```
(key,CM_AIDS,CM_ALCOHOL,CM_ANEMDEF,CM_ARTH,CM_BLDLOSS,CM_CHF,CM_CHR
NLUNG,CM_COAG,CM_DEPRESS,CM_DM,
```

```
CM_DMCX,CM_DRUG,CM_HTN_C,CM_HYPOTHY,CM_LIVER,CM_LYMPH,CM_LYTES,CM_
METS,CM_NEURO,CM_OBESE,CM_PARA,CM_PERIVASC,
```

```
CM_PSYCH,CM_PULMCIRC,CM_RENLFAIL,CM_TUMOR,CM_ULCER,CM_VALVE,CM_WG
HTLOSS)
```

```
VALUES
```

```
(pat_pr.key,pat_pr.CM_AIDS,pat_pr.CM_ALCOHOL,pat_pr.CM_ANEMDEF,pat_pr.CM_A
RTH,pat_pr.CM_BLDLOSS,pat_pr.CM_CHF,pat_pr.CM_CHRNLUNG,pat_pr.CM_COAG,pat
_pr.CM_DEPRESS,pat_pr.CM_DM,
```

```
pat_pr.CM_DMCX,pat_pr.CM_DRUG,pat_pr.CM_HTN_C,pat_pr.CM_HYPOTHY,pat_pr.C
```

```
M_LIVER,pat_pr.CM_LYMPH,pat_pr.CM_LYTES,pat_pr.CM_METS,pat_pr.CM_NEURO,pat_pr.CM_OBESE,pat_pr.CM_PARA,
```

```
pat_pr.CM_PERIVASC,pat_pr.CM_PSYCH,pat_pr.CM_PULMCIRC,pat_pr.CM_RENLFAIL,pat_pr.CM_TUMOR,pat_pr.CM_ULCER,pat_pr.CM_VALVE,pat_pr.CM_WGHTLOSS);
```

```
elseif substr(equiv_pr1(elem1),1,1) = '4' then
  insert into pat_pr_4 (key,year,dqtr, age, female, race, weekend_adm,
    elec_adm,pr1,pr2,pr3,dx1, o_pr1,o_pr2,o_pr3,o_dx1, died, LOS, dispo)
  values(pat_pr.key,'2007', pat_pr.dqtr, pat_pr.age_num, pat_pr.female, pat_pr.race,
    pat_pr.aweekend,
    pat_pr.elective,trim(equiv_pr1(elem1)),trim(replace(equiv_pr2(elem2),'AAA','')),trim(
    replace(equiv_pr3(elem3),'AAA','')),trim(replace(equiv_dx1(elemdx),'AAA','')),pat_pr.pr1,
    pat_pr.pr2,pat_pr.pr3,pat_pr.dx1,pat_pr.died,decode(pat_pr.LOS,null,null,'A',null,'B',null,
    'C',null,to_number(pat_pr.LOS)),pat_pr.dispuniform);
```

```
insert into pat_como_4
```

```
(key,CM_AIDS,CM_ALCOHOL,CM_ANEMDEF,CM_ARTH,CM_BLDLOSS,CM_CHF,CM_CHRNLUNG,CM_COAG,CM_DEPRESS,CM_DM,
```

```
CM_DMCX,CM_DRUG,CM_HTN_C,CM_HYPOTHY,CM_LIVER,CM_LYMPH,CM_LYTES,CM_METS,CM_NEURO,CM_OBESE,CM_PARA,CM_PERIVASC,
```

```
CM_PSYCH,CM_PULMCIRC,CM_RENLFAIL,CM_TUMOR,CM_ULCER,CM_VALVE,CM_WGHTLOSS)
VALUES
```

```
(pat_pr.key,pat_pr.CM_AIDS,pat_pr.CM_ALCOHOL,pat_pr.CM_ANEMDEF,pat_pr.CM_ARTH,pat_pr.CM_BLDLOSS,pat_pr.CM_CHF,pat_pr.CM_CHRNLUNG,pat_pr.CM_COAG,pat_pr.CM_DEPRESS,pat_pr.CM_DM,
```

```
pat_pr.CM_DMCX,pat_pr.CM_DRUG,pat_pr.CM_HTN_C,pat_pr.CM_HYPOTHY,pat_pr.CM_LIVER,pat_pr.CM_LYMPH,pat_pr.CM_LYTES,pat_pr.CM_METS,pat_pr.CM_NEURO,pat_pr.CM_OBESE,pat_pr.CM_PARA,
```

```
pat_pr.CM_PERIVASC,pat_pr.CM_PSYCH,pat_pr.CM_PULMCIRC,pat_pr.CM_RENLFAIL,pat_pr.CM_TUMOR,pat_pr.CM_ULCER,pat_pr.CM_VALVE,pat_pr.CM_WGHTLOSS);
```

```
elseif substr(equiv_pr1(elem1),1,1) = '5' then
  insert into pat_pr_5 (key,year,dqtr, age, female, race, weekend_adm,
    elec_adm,pr1,pr2,pr3,dx1, o_pr1,o_pr2,o_pr3,o_dx1, died, LOS, dispo)
  values(pat_pr.key,'2007', pat_pr.dqtr, pat_pr.age_num, pat_pr.female, pat_pr.race,
    pat_pr.aweekend,
    pat_pr.elective,trim(equiv_pr1(elem1)),trim(replace(equiv_pr2(elem2),'AAA','')),trim(
    replace(equiv_pr3(elem3),'AAA','')),trim(replace(equiv_dx1(elemdx),'AAA','')),pat_pr.pr1,
    pat_pr.pr2,pat_pr.pr3,pat_pr.dx1,pat_pr.died,decode(pat_pr.LOS,null,null,'A',null,'B',null,
    'C',null,to_number(pat_pr.LOS)),pat_pr.dispuniform);
```

```
insert into pat_como_5
```

```
(key,CM_AIDS,CM_ALCOHOL,CM_ANEMDEF,CM_ARTH,CM_BLDLOSS,CM_CHF,CM_CHRNLUNG,CM_COAG,CM_DEPRESS,CM_DM,
```

```
CM_DMCX,CM_DRUG,CM_HTN_C,CM_HYPOTHY,CM_LIVER,CM_LYMPH,CM_LYTES,CM_METS,CM_NEURO,CM_OBESE,CM_PARA,CM_PERIVASC,
```

```
CM_PSYCH,CM_PULMCIRC,CM_RENLFAIL,CM_TUMOR,CM_ULCER,CM_VALVE,CM_WGHTLOSS)
```

```
VALUES
```

```
(pat_pr.key,pat_pr.CM_AIDS,pat_pr.CM_ALCOHOL,pat_pr.CM_ANEMDEF,pat_pr.CM_ARTH,pat_pr.CM_BLDLOSS,pat_pr.CM_CHF,pat_pr.CM_CHRNLUNG,pat_pr.CM_COAG,pat_pr.CM_DEPRESS,pat_pr.CM_DM,
```

```
pat_pr.CM_DMCX,pat_pr.CM_DRUG,pat_pr.CM_HTN_C,pat_pr.CM_HYPOTHY,pat_pr.CM_LIVER,pat_pr.CM_LYMPH,pat_pr.CM_LYTES,pat_pr.CM_METS,pat_pr.CM_NEURO,pat_pr.CM_OBESE,pat_pr.CM_PARA,
```

```
pat_pr.CM_PERIVASC,pat_pr.CM_PSYCH,pat_pr.CM_PULMCIRC,pat_pr.CM_RENLFAIL,pat_pr.CM_TUMOR,pat_pr.CM_ULCER,pat_pr.CM_VALVE,pat_pr.CM_WGHTLOSS);
```

```
elsif substr(equiv_pr1(elem1),1,1) = '6' then
```

```
insert into pat_pr_6 (key,year,dqtr, age, female, race, weekend_adm, elec_adm,pr1,pr2,pr3,dx1, o_pr1,o_pr2,o_pr3,o_dx1, died, LOS, dispo)
```

```
values(pat_pr.key,'2007', pat_pr.dqtr, pat_pr.age_num, pat_pr.female, pat_pr.race, pat_pr.aweekend, pat_pr.elective,trim(equiv_pr1(elem1)),trim(replace(equiv_pr2(elem2),'AAA','')),trim(replace(equiv_pr3(elem3),'AAA','')),trim(replace(equiv_dx1(elemdx),'AAA','')),pat_pr.pr1,pat_pr.pr2,pat_pr.pr3,pat_pr.dx1,pat_pr.died,decode(pat_pr.LOS,null,null,'A',null,'B',null,'C',null,to_number(pat_pr.LOS)),pat_pr.dispuniform);
```

```
insert into pat_como_6
```

```
(key,CM_AIDS,CM_ALCOHOL,CM_ANEMDEF,CM_ARTH,CM_BLDLOSS,CM_CHF,CM_CHRNLUNG,CM_COAG,CM_DEPRESS,CM_DM,
```

```
CM_DMCX,CM_DRUG,CM_HTN_C,CM_HYPOTHY,CM_LIVER,CM_LYMPH,CM_LYTES,CM_METS,CM_NEURO,CM_OBESE,CM_PARA,CM_PERIVASC,
```

```
CM_PSYCH,CM_PULMCIRC,CM_RENLFAIL,CM_TUMOR,CM_ULCER,CM_VALVE,CM_WGHTLOSS)
```

```
VALUES
```

```
(pat_pr.key,pat_pr.CM_AIDS,pat_pr.CM_ALCOHOL,pat_pr.CM_ANEMDEF,pat_pr.CM_A
```

```
RTH,pat_pr.CM_BLDLOSS,pat_pr.CM_CHF,pat_pr.CM_CHRNLUNG,pat_pr.CM_COAG,pat_pr.CM_DEPRESS,pat_pr.CM_DM,
```

```
pat_pr.CM_DMCX,pat_pr.CM_DRUG,pat_pr.CM_HTN_C,pat_pr.CM_HYPOTHY,pat_pr.CM_LIVER,pat_pr.CM_LYMPH,pat_pr.CM_LYTES,pat_pr.CM_METS,pat_pr.CM_NEURO,pat_pr.CM_OBESE,pat_pr.CM_PARA,
```

```
pat_pr.CM_PERIVASC,pat_pr.CM_PSYCH,pat_pr.CM_PULMCIRC,pat_pr.CM_RENLFAIL,pat_pr.CM_TUMOR,pat_pr.CM_ULCER,pat_pr.CM_VALVE,pat_pr.CM_WGHTLOSS);
```

```
elseif substr(equiv_pr1(elem1),1,1) = '7' then
  insert into pat_pr_7 (key,year,dqtr, age, female, race, weekend_adm,
    elec_adm,pr1,pr2,pr3,dx1, o_pr1,o_pr2,o_pr3,o_dx1, died, LOS, dispo)
    values(pat_pr.key,'2007', pat_pr.dqtr, pat_pr.age_num, pat_pr.female, pat_pr.race,
    pat_pr.aweekend,
    pat_pr.elective,trim(equiv_pr1(elem1)),trim(replace(equiv_pr2(elem2),'AAA','')),trim(
    replace(equiv_pr3(elem3),'AAA','')),trim(replace(equiv_dx1(elemdx),'AAA','')),pat_pr.pr1,
    pat_pr.pr2,pat_pr.pr3,pat_pr.dx1,pat_pr.died,decode(pat_pr.LOS,null,null,'A',null,'B',null,
    'C',null,to_number(pat_pr.LOS)),pat_pr.dispuniform);
```

```
insert into pat_como_7
```

```
(key,CM_AIDS,CM_ALCOHOL,CM_ANEMDEF,CM_ARTH,CM_BLDLOSS,CM_CHF,CM_CHRNLUNG,CM_COAG,CM_DEPRESS,CM_DM,
```

```
CM_DMCX,CM_DRUG,CM_HTN_C,CM_HYPOTHY,CM_LIVER,CM_LYMPH,CM_LYTES,CM_METS,CM_NEURO,CM_OBESE,CM_PARA,CM_PERIVASC,
```

```
CM_PSYCH,CM_PULMCIRC,CM_RENLFAIL,CM_TUMOR,CM_ULCER,CM_VALVE,CM_WGHTLOSS)
```

```
VALUES
```

```
(pat_pr.key,pat_pr.CM_AIDS,pat_pr.CM_ALCOHOL,pat_pr.CM_ANEMDEF,pat_pr.CM_ARTH,pat_pr.CM_BLDLOSS,pat_pr.CM_CHF,pat_pr.CM_CHRNLUNG,pat_pr.CM_COAG,pat_pr.CM_DEPRESS,pat_pr.CM_DM,
```

```
pat_pr.CM_DMCX,pat_pr.CM_DRUG,pat_pr.CM_HTN_C,pat_pr.CM_HYPOTHY,pat_pr.CM_LIVER,pat_pr.CM_LYMPH,pat_pr.CM_LYTES,pat_pr.CM_METS,pat_pr.CM_NEURO,pat_pr.CM_OBESE,pat_pr.CM_PARA,
```

```
pat_pr.CM_PERIVASC,pat_pr.CM_PSYCH,pat_pr.CM_PULMCIRC,pat_pr.CM_RENLFAIL,pat_pr.CM_TUMOR,pat_pr.CM_ULCER,pat_pr.CM_VALVE,pat_pr.CM_WGHTLOSS);
```

```
elseif substr(equiv_pr1(elem1),1,1) = '8' then
  insert into pat_pr_8 (key,year,dqtr, age, female, race, weekend_adm,
    elec_adm,pr1,pr2,pr3,dx1, o_pr1,o_pr2,o_pr3,o_dx1, died, LOS, dispo)
    values(pat_pr.key,'2007', pat_pr.dqtr, pat_pr.age_num, pat_pr.female, pat_pr.race,
    pat_pr.aweekend,
```

```

pat_pr.elective,trim(equiv_pr1(elem1)),trim(replace(equiv_pr2(elem2),'AAA','')),trim(replace(equiv_pr3(elem3),'AAA','')),trim(
replace(equiv_dx1(elemdx),'AAA','')),pat_pr.pr1,pat_pr.pr2,pat_pr.pr3,pat_pr.dx1,pat_pr.died,decode(pat_pr.LOS,null,null,'A',null,'B',null,'C',null,to_number(pat_pr.LOS)),pat_pr.dispuniform);

```

```

insert into pat_como_8

```

```

(key,CM_AIDS,CM_ALCOHOL,CM_ANEMDEF,CM_ARTH,CM_BLDLOSS,CM_CHF,CM_CHRNLUNG,CM_COAG,CM_DEPRESS,CM_DM,

```

```

CM_DMCX,CM_DRUG,CM_HTN_C,CM_HYPOTHY,CM_LIVER,CM_LYMPH,CM_LYTES,CM_METS,CM_NEURO,CM_OBESE,CM_PARA,CM_PERIVASC,

```

```

CM_PSYCH,CM_PULMCIRC,CM_RENLFAIL,CM_TUMOR,CM_ULCER,CM_VALVE,CM_WGHTLOSS)

```

```

VALUES

```

```

(pat_pr.key,pat_pr.CM_AIDS,pat_pr.CM_ALCOHOL,pat_pr.CM_ANEMDEF,pat_pr.CM_ARTH,pat_pr.CM_BLDLOSS,pat_pr.CM_CHF,pat_pr.CM_CHRNLUNG,pat_pr.CM_COAG,pat_pr.CM_DEPRESS,pat_pr.CM_DM,

```

```

pat_pr.CM_DMCX,pat_pr.CM_DRUG,pat_pr.CM_HTN_C,pat_pr.CM_HYPOTHY,pat_pr.CM_LIVER,pat_pr.CM_LYMPH,pat_pr.CM_LYTES,pat_pr.CM_METS,pat_pr.CM_NEURO,pat_pr.CM_OBESE,pat_pr.CM_PARA,

```

```

pat_pr.CM_PERIVASC,pat_pr.CM_PSYCH,pat_pr.CM_PULMCIRC,pat_pr.CM_RENLFAIL,pat_pr.CM_TUMOR,pat_pr.CM_ULCER,pat_pr.CM_VALVE,pat_pr.CM_WGHTLOSS);

```

```

elsif substr(equiv_pr1(elem1),1,1) = '9' then

```

```

insert into pat_pr_9 (key,year,dqtr, age, female, race, weekend_adm, elec_adm,pr1,pr2,pr3,dx1, o_pr1,o_pr2,o_pr3,o_dx1, died, LOS, dispo)

```

```

values(pat_pr.key,'2007', pat_pr.dqtr, pat_pr.age_num, pat_pr.female, pat_pr.race, pat_pr.aweekend,

```

```

pat_pr.elective,trim(equiv_pr1(elem1)),trim(replace(equiv_pr2(elem2),'AAA','')),trim(replace(equiv_pr3(elem3),'AAA','')),trim(
replace(equiv_dx1(elemdx),'AAA','')),pat_pr.pr1,pat_pr.pr2,pat_pr.pr3,pat_pr.dx1,pat_pr.died,decode(pat_pr.LOS,null,null,'A',null,'B',null,'C',null,to_number(pat_pr.LOS)),pat_pr.dispuniform);

```

```

insert into pat_como_9

```

```

(key,CM_AIDS,CM_ALCOHOL,CM_ANEMDEF,CM_ARTH,CM_BLDLOSS,CM_CHF,CM_CHRNLUNG,CM_COAG,CM_DEPRESS,CM_DM,

```

```

CM_DMCX,CM_DRUG,CM_HTN_C,CM_HYPOTHY,CM_LIVER,CM_LYMPH,CM_LYTES,CM_METS,CM_NEURO,CM_OBESE,CM_PARA,CM_PERIVASC,

```

```

CM_PSYCH,CM_PULMCIRC,CM_RENLFAIL,CM_TUMOR,CM_ULCER,CM_VALVE,CM_WGHTLOSS)

```

VALUES

```
(pat_pr.key,pat_pr.CM_AIDS,pat_pr.CM_ALCOHOL,pat_pr.CM_ANEMDEF,pat_pr.CM_ARTH,pat_pr.CM_BLDLOSS,pat_pr.CM_CHF,pat_pr.CM_CHRNLUNG,pat_pr.CM_COAG,pat_pr.CM_DEPRESS,pat_pr.CM_DM,
```

```
pat_pr.CM_DMCX,pat_pr.CM_DRUG,pat_pr.CM_HTN_C,pat_pr.CM_HYPOTHY,pat_pr.CM_LIVER,pat_pr.CM_LYMPH,pat_pr.CM_LYTES,pat_pr.CM_METS,pat_pr.CM_NEURO,pat_pr.CM_OBESE,pat_pr.CM_PARA,
```

```
pat_pr.CM_PERIVASC,pat_pr.CM_PSYCH,pat_pr.CM_PULMCIRC,pat_pr.CM_RENLFAIL,pat_pr.CM_TUMOR,pat_pr.CM_ULCER,pat_pr.CM_VALVE,pat_pr.CM_WGHTLOSS);
```

```
end if;
```

```
    end loop; -- dx1
```

```
    end loop; -- pr3
```

```
    end loop; -- pr2
```

```
end loop; -- pr1
```

```
elseif pat_pr.dqtr = '4' then
```

```
if substr(pat_pr.pr1,1,1) IN ('0','1','2') then
```

```
    insert into pat_pr_0_2 (key,year,dqtr, age, female, race, weekend_adm, elec_adm,pr1,pr2,pr3,dx1, o_pr1,o_pr2,o_pr3,o_dx1, died, LOS, dispo)
```

```
    values(pat_pr.key,'2007', pat_pr.dqtr, pat_pr.age_num, pat_pr.female, pat_pr.race, pat_pr.aweekend,
```

```
pat_pr.elective,pat_pr.pr1,pat_pr.pr2,pat_pr.pr3,pat_pr.dx1,pat_pr.pr1,pat_pr.pr2,pat_pr.pr3,pat_pr.dx1,pat_pr.died,decode(pat_pr.LOS,null,null,'A',null,'B',null,'C',null,to_number(pat_pr.LOS)),pat_pr.dispuniform);
```

```
    insert into pat_como_0_2
```

```
(key,CM_AIDS,CM_ALCOHOL,CM_ANEMDEF,CM_ARTH,CM_BLDLOSS,CM_CHF,CM_CHRNLUNG,CM_COAG,CM_DEPRESS,CM_DM,
```

```
CM_DMCX,CM_DRUG,CM_HTN_C,CM_HYPOTHY,CM_LIVER,CM_LYMPH,CM_LYTES,CM_METS,CM_NEURO,CM_OBESE,CM_PARA,CM_PERIVASC,
```

```
CM_PSYCH,CM_PULMCIRC,CM_RENLFAIL,CM_TUMOR,CM_ULCER,CM_VALVE,CM_WGHTLOSS)
```

VALUES

```
(pat_pr.key,pat_pr.CM_AIDS,pat_pr.CM_ALCOHOL,pat_pr.CM_ANEMDEF,pat_pr.CM_ARTH,pat_pr.CM_BLDLOSS,pat_pr.CM_CHF,pat_pr.CM_CHRNLUNG,pat_pr.CM_COAG,pat_pr.CM_DEPRESS,pat_pr.CM_DM,
```

```
pat_pr.CM_DMCX,pat_pr.CM_DRUG,pat_pr.CM_HTN_C,pat_pr.CM_HYPOTHY,pat_pr.CM
```

```
M_LIVER,pat_pr.CM_LYMPH,pat_pr.CM_LYTES,pat_pr.CM_METS,pat_pr.CM_NEURO,pat_pr.CM_OBESE,pat_pr.CM_PARA,
```

```
pat_pr.CM_PERIVASC,pat_pr.CM_PSYCH,pat_pr.CM_PULMCIRC,pat_pr.CM_RENLFAIL,pat_pr.CM_TUMOR,pat_pr.CM_ULCER,pat_pr.CM_VALVE,pat_pr.CM_WGHTLOSS);
```

```
elseif substr(pat_pr.pr1,1,1) = '3' then
```

```
  insert into pat_pr_3 (key,year,dqtr, age, female, race, weekend_adm,
  elec_adm,pr1,pr2,pr3,dx1, o_pr1,o_pr2,o_pr3,o_dx1, died, LOS, dispo)
  values(pat_pr.key,'2007', pat_pr.dqtr, pat_pr.age_num, pat_pr.female, pat_pr.race,
  pat_pr.aweekend,
  pat_pr.elective,pat_pr.pr1,pat_pr.pr2,pat_pr.pr3,pat_pr.dx1,pat_pr.pr1,pat_pr.pr2,pat_pr.pr3,pat_pr.dx1,pat_pr.died,decode(pat_pr.LOS,null,null,'A',null,'B',null,'C',null,to_number(pat_pr.LOS)),pat_pr.dispuniform);
```

```
  insert into pat_como_3
```

```
(key,CM_AIDS,CM_ALCOHOL,CM_ANEMDEF,CM_ARTH,CM_BLDLOSS,CM_CHF,CM_CHRNLUNG,CM_COAG,CM_DEPRESS,CM_DM,
```

```
CM_DMCX,CM_DRUG,CM_HTN_C,CM_HYPOTHY,CM_LIVER,CM_LYMPH,CM_LYTES,CM_METS,CM_NEURO,CM_OBESE,CM_PARA,CM_PERIVASC,
```

```
CM_PSYCH,CM_PULMCIRC,CM_RENLFAIL,CM_TUMOR,CM_ULCER,CM_VALVE,CM_WGHTLOSS)
```

```
  VALUES
```

```
(pat_pr.key,pat_pr.CM_AIDS,pat_pr.CM_ALCOHOL,pat_pr.CM_ANEMDEF,pat_pr.CM_ARTH,pat_pr.CM_BLDLOSS,pat_pr.CM_CHF,pat_pr.CM_CHRNLUNG,pat_pr.CM_COAG,pat_pr.CM_DEPRESS,pat_pr.CM_DM,
```

```
pat_pr.CM_DMCX,pat_pr.CM_DRUG,pat_pr.CM_HTN_C,pat_pr.CM_HYPOTHY,pat_pr.CM_LIVER,pat_pr.CM_LYMPH,pat_pr.CM_LYTES,pat_pr.CM_METS,pat_pr.CM_NEURO,pat_pr.CM_OBESE,pat_pr.CM_PARA,
```

```
pat_pr.CM_PERIVASC,pat_pr.CM_PSYCH,pat_pr.CM_PULMCIRC,pat_pr.CM_RENLFAIL,pat_pr.CM_TUMOR,pat_pr.CM_ULCER,pat_pr.CM_VALVE,pat_pr.CM_WGHTLOSS);
```

```
elseif substr(pat_pr.pr1,1,1) = '4' then
```

```
  insert into pat_pr_4 (key,year,dqtr, age, female, race, weekend_adm,
  elec_adm,pr1,pr2,pr3,dx1, o_pr1,o_pr2,o_pr3,o_dx1, died, LOS, dispo)
  values(pat_pr.key,'2007', pat_pr.dqtr, pat_pr.age_num, pat_pr.female, pat_pr.race,
  pat_pr.aweekend,
  pat_pr.elective,pat_pr.pr1,pat_pr.pr2,pat_pr.pr3,pat_pr.dx1,pat_pr.pr1,pat_pr.pr2,pat_pr.pr3,pat_pr.dx1,pat_pr.died,decode(pat_pr.LOS,null,null,'A',null,'B',null,'C',null,to_number(pat_pr.LOS)),pat_pr.dispuniform);
```

```
  insert into pat_como_4
```



```
(key,CM_AIDS,CM_ALCOHOL,CM_ANEMDEF,CM_ARTH,CM_BLDLOSS,CM_CHF,CM_CHRNLUNG,CM_COAG,CM_DEPRESS,CM_DM,
```

```
CM_DMCX,CM_DRUG,CM_HTN_C,CM_HYPOTHY,CM_LIVER,CM_LYMPH,CM_LYTES,CM_METS,CM_NEURO,CM_OBESE,CM_PARA,CM_PERIVASC,
```

```
CM_PSYCH,CM_PULMCIRC,CM_RENLFAIL,CM_TUMOR,CM_ULCER,CM_VALVE,CM_WGHTLOSS)
```

```
VALUES
```

```
(pat_pr.key,pat_pr.CM_AIDS,pat_pr.CM_ALCOHOL,pat_pr.CM_ANEMDEF,pat_pr.CM_ARTH,pat_pr.CM_BLDLOSS,pat_pr.CM_CHF,pat_pr.CM_CHRNLUNG,pat_pr.CM_COAG,pat_pr.CM_DEPRESS,pat_pr.CM_DM,
```

```
pat_pr.CM_DMCX,pat_pr.CM_DRUG,pat_pr.CM_HTN_C,pat_pr.CM_HYPOTHY,pat_pr.CM_LIVER,pat_pr.CM_LYMPH,pat_pr.CM_LYTES,pat_pr.CM_METS,pat_pr.CM_NEURO,pat_pr.CM_OBESE,pat_pr.CM_PARA,
```

```
pat_pr.CM_PERIVASC,pat_pr.CM_PSYCH,pat_pr.CM_PULMCIRC,pat_pr.CM_RENLFAIL,pat_pr.CM_TUMOR,pat_pr.CM_ULCER,pat_pr.CM_VALVE,pat_pr.CM_WGHTLOSS);
```

```
elsif substr(pat_pr.pr1,1,1) = '5' then
```

```
insert into pat_pr_5 (key,year,dqtr, age, female, race, weekend_adm, elec_adm,pr1,pr2,pr3,dx1, o_pr1,o_pr2,o_pr3,o_dx1, died, LOS, dispo)
```

```
values(pat_pr.key,'2007', pat_pr.dqtr, pat_pr.age_num, pat_pr.female, pat_pr.race, pat_pr.aweekend,
```

```
pat_pr.elective,pat_pr.pr1,pat_pr.pr2,pat_pr.pr3,pat_pr.dx1,pat_pr.pr1,pat_pr.pr2,pat_pr.pr3,pat_pr.dx1,pat_pr.died,decode(pat_pr.LOS,null,null,'A',null,'B',null,'C',null,to_number(pat_pr.LOS)),pat_pr.dispuniform);
```

```
insert into pat_como_5
```

```
(key,CM_AIDS,CM_ALCOHOL,CM_ANEMDEF,CM_ARTH,CM_BLDLOSS,CM_CHF,CM_CHRNLUNG,CM_COAG,CM_DEPRESS,CM_DM,
```

```
CM_DMCX,CM_DRUG,CM_HTN_C,CM_HYPOTHY,CM_LIVER,CM_LYMPH,CM_LYTES,CM_METS,CM_NEURO,CM_OBESE,CM_PARA,CM_PERIVASC,
```

```
CM_PSYCH,CM_PULMCIRC,CM_RENLFAIL,CM_TUMOR,CM_ULCER,CM_VALVE,CM_WGHTLOSS)
```

```
VALUES
```

```
(pat_pr.key,pat_pr.CM_AIDS,pat_pr.CM_ALCOHOL,pat_pr.CM_ANEMDEF,pat_pr.CM_ARTH,pat_pr.CM_BLDLOSS,pat_pr.CM_CHF,pat_pr.CM_CHRNLUNG,pat_pr.CM_COAG,pat_pr.CM_DEPRESS,pat_pr.CM_DM,
```

```
pat_pr.CM_DMCX,pat_pr.CM_DRUG,pat_pr.CM_HTN_C,pat_pr.CM_HYPOTHY,pat_pr.CM
```

```
M_LIVER,pat_pr.CM_LYMPH,pat_pr.CM_LYTES,pat_pr.CM_METS,pat_pr.CM_NEURO,pat_pr.CM_OBESE,pat_pr.CM_PARA,
```

```
pat_pr.CM_PERIVASC,pat_pr.CM_PSYCH,pat_pr.CM_PULMCIRC,pat_pr.CM_RENLFAIL,pat_pr.CM_TUMOR,pat_pr.CM_ULCER,pat_pr.CM_VALVE,pat_pr.CM_WGHTLOSS);
```

```
elseif substr(pat_pr.pr1,1,1) = '6' then
```

```
  insert into pat_pr_6 (key,year,dqtr, age, female, race, weekend_adm,
  elec_adm,pr1,pr2,pr3,dx1, o_pr1,o_pr2,o_pr3,o_dx1, died, LOS, dispo)
  values(pat_pr.key,'2007', pat_pr.dqtr, pat_pr.age_num, pat_pr.female, pat_pr.race,
  pat_pr.aweekend,
  pat_pr.elective,pat_pr.pr1,pat_pr.pr2,pat_pr.pr3,pat_pr.dx1,pat_pr.pr1,pat_pr.pr2,pat_pr.pr3,pat_pr.dx1,pat_pr.died,decode(pat_pr.LOS,null,null,'A',null,'B',null,'C',null,to_number(pat_pr.LOS)),pat_pr.dispuniform);
```

```
  insert into pat_como_6
```

```
(key,CM_AIDS,CM_ALCOHOL,CM_ANEMDEF,CM_ARTH,CM_BLDLOSS,CM_CHF,CM_CHRNLUNG,CM_COAG,CM_DEPRESS,CM_DM,
```

```
CM_DMCX,CM_DRUG,CM_HTN_C,CM_HYPOTHY,CM_LIVER,CM_LYMPH,CM_LYTES,CM_METS,CM_NEURO,CM_OBESE,CM_PARA,CM_PERIVASC,
```

```
CM_PSYCH,CM_PULMCIRC,CM_RENLFAIL,CM_TUMOR,CM_ULCER,CM_VALVE,CM_WGHTLOSS)
```

```
  VALUES
```

```
(pat_pr.key,pat_pr.CM_AIDS,pat_pr.CM_ALCOHOL,pat_pr.CM_ANEMDEF,pat_pr.CM_ARTH,pat_pr.CM_BLDLOSS,pat_pr.CM_CHF,pat_pr.CM_CHRNLUNG,pat_pr.CM_COAG,pat_pr.CM_DEPRESS,pat_pr.CM_DM,
```

```
pat_pr.CM_DMCX,pat_pr.CM_DRUG,pat_pr.CM_HTN_C,pat_pr.CM_HYPOTHY,pat_pr.CM_LIVER,pat_pr.CM_LYMPH,pat_pr.CM_LYTES,pat_pr.CM_METS,pat_pr.CM_NEURO,pat_pr.CM_OBESE,pat_pr.CM_PARA,
```

```
pat_pr.CM_PERIVASC,pat_pr.CM_PSYCH,pat_pr.CM_PULMCIRC,pat_pr.CM_RENLFAIL,pat_pr.CM_TUMOR,pat_pr.CM_ULCER,pat_pr.CM_VALVE,pat_pr.CM_WGHTLOSS);
```

```
elseif substr(pat_pr.pr1,1,1) = '7' then
```

```
  insert into pat_pr_7 (key,year,dqtr, age, female, race, weekend_adm,
  elec_adm,pr1,pr2,pr3,dx1, o_pr1,o_pr2,o_pr3,o_dx1, died, LOS, dispo)
  values(pat_pr.key,'2007', pat_pr.dqtr, pat_pr.age_num, pat_pr.female, pat_pr.race,
  pat_pr.aweekend,
  pat_pr.elective,pat_pr.pr1,pat_pr.pr2,pat_pr.pr3,pat_pr.dx1,pat_pr.pr1,pat_pr.pr2,pat_pr.pr3,pat_pr.dx1,pat_pr.died,decode(pat_pr.LOS,null,null,'A',null,'B',null,'C',null,to_number(pat_pr.LOS)),pat_pr.dispuniform);
```

```
  insert into pat_como_7
```

```
(key,CM_AIDS,CM_ALCOHOL,CM_ANEMDEF,CM_ARTH,CM_BLDLOSS,CM_CHF,CM_CHRNLUNG,CM_COAG,CM_DEPRESS,CM_DM,
```

```
CM_DMCX,CM_DRUG,CM_HTN_C,CM_HYPOTHY,CM_LIVER,CM_LYMPH,CM_LYTES,CM_METS,CM_NEURO,CM_OBESE,CM_PARA,CM_PERIVASC,
```

```
CM_PSYCH,CM_PULMCIRC,CM_RENLFAIL,CM_TUMOR,CM_ULCER,CM_VALVE,CM_WGHTLOSS)
```

```
VALUES
```

```
(pat_pr.key,pat_pr.CM_AIDS,pat_pr.CM_ALCOHOL,pat_pr.CM_ANEMDEF,pat_pr.CM_ARTH,pat_pr.CM_BLDLOSS,pat_pr.CM_CHF,pat_pr.CM_CHRNLUNG,pat_pr.CM_COAG,pat_pr.CM_DEPRESS,pat_pr.CM_DM,
```

```
pat_pr.CM_DMCX,pat_pr.CM_DRUG,pat_pr.CM_HTN_C,pat_pr.CM_HYPOTHY,pat_pr.CM_LIVER,pat_pr.CM_LYMPH,pat_pr.CM_LYTES,pat_pr.CM_METS,pat_pr.CM_NEURO,pat_pr.CM_OBESE,pat_pr.CM_PARA,
```

```
pat_pr.CM_PERIVASC,pat_pr.CM_PSYCH,pat_pr.CM_PULMCIRC,pat_pr.CM_RENLFAIL,pat_pr.CM_TUMOR,pat_pr.CM_ULCER,pat_pr.CM_VALVE,pat_pr.CM_WGHTLOSS);
```

```
elsif substr(pat_pr.pr1,1,1) = '8' then
```

```
insert into pat_pr_8 (key,year,dqtr, age, female, race, weekend_adm, elec_adm,pr1,pr2,pr3,dx1, o_pr1,o_pr2,o_pr3,o_dx1, died, LOS, dispo)
```

```
values(pat_pr.key,'2007', pat_pr.dqtr, pat_pr.age_num, pat_pr.female, pat_pr.race, pat_pr.aweekend,
```

```
pat_pr.elective,pat_pr.pr1,pat_pr.pr2,pat_pr.pr3,pat_pr.dx1,pat_pr.pr1,pat_pr.pr2,pat_pr.pr3,pat_pr.dx1,pat_pr.died,decode(pat_pr.LOS,null,null,'A',null,'B',null,'C',null,to_number(pat_pr.LOS)),pat_pr.dispuniform);
```

```
insert into pat_como_8
```

```
(key,CM_AIDS,CM_ALCOHOL,CM_ANEMDEF,CM_ARTH,CM_BLDLOSS,CM_CHF,CM_CHRNLUNG,CM_COAG,CM_DEPRESS,CM_DM,
```

```
CM_DMCX,CM_DRUG,CM_HTN_C,CM_HYPOTHY,CM_LIVER,CM_LYMPH,CM_LYTES,CM_METS,CM_NEURO,CM_OBESE,CM_PARA,CM_PERIVASC,
```

```
CM_PSYCH,CM_PULMCIRC,CM_RENLFAIL,CM_TUMOR,CM_ULCER,CM_VALVE,CM_WGHTLOSS)
```

```
VALUES
```

```
(pat_pr.key,pat_pr.CM_AIDS,pat_pr.CM_ALCOHOL,pat_pr.CM_ANEMDEF,pat_pr.CM_ARTH,pat_pr.CM_BLDLOSS,pat_pr.CM_CHF,pat_pr.CM_CHRNLUNG,pat_pr.CM_COAG,pat_pr.CM_DEPRESS,pat_pr.CM_DM,
```

```
pat_pr.CM_DMCX,pat_pr.CM_DRUG,pat_pr.CM_HTN_C,pat_pr.CM_HYPOTHY,pat_pr.CM
```

```
M_LIVER,pat_pr.CM_LYMPH,pat_pr.CM_LYTES,pat_pr.CM_METS,pat_pr.CM_NEURO,pat_pr.CM_OBESE,pat_pr.CM_PARA,
```

```
pat_pr.CM_PERIVASC,pat_pr.CM_PSYCH,pat_pr.CM_PULMCIRC,pat_pr.CM_RENLFAIL,pat_pr.CM_TUMOR,pat_pr.CM_ULCER,pat_pr.CM_VALVE,pat_pr.CM_WGHTLOSS);
```

```
elseif substr(pat_pr.pr1,1,1) = '9' then
```

```
  insert into pat_pr_9 (key,year,dqtr, age, female, race, weekend_adm,
  elec_adm,pr1,pr2,pr3,dx1, o_pr1,o_pr2,o_pr3,o_dx1, died, LOS, dispo)
  values(pat_pr.key,'2007', pat_pr.dqtr, pat_pr.age_num, pat_pr.female, pat_pr.race,
  pat_pr.aweekend,
  pat_pr.elective,pat_pr.pr1,pat_pr.pr2,pat_pr.pr3,pat_pr.dx1,pat_pr.pr1,pat_pr.pr2,pat_pr.pr3,pat_pr.dx1,pat_pr.died,decode(pat_pr.LOS,null,null,'A',null,'B',null,'C',null,to_number(pat_pr.LOS)),pat_pr.dispuniform);
```

```
  insert into pat_como_9
```

```
(key,CM_AIDS,CM_ALCOHOL,CM_ANEMDEF,CM_ARTH,CM_BLDLOSS,CM_CHF,CM_CHRNLUNG,CM_COAG,CM_DEPRESS,CM_DM,
```

```
CM_DMCX,CM_DRUG,CM_HTN_C,CM_HYPOTHY,CM_LIVER,CM_LYMPH,CM_LYTES,CM_METS,CM_NEURO,CM_OBESE,CM_PARA,CM_PERIVASC,
```

```
CM_PSYCH,CM_PULMCIRC,CM_RENLFAIL,CM_TUMOR,CM_ULCER,CM_VALVE,CM_WGHTLOSS)
```

```
  VALUES
```

```
(pat_pr.key,pat_pr.CM_AIDS,pat_pr.CM_ALCOHOL,pat_pr.CM_ANEMDEF,pat_pr.CM_ARTH,pat_pr.CM_BLDLOSS,pat_pr.CM_CHF,pat_pr.CM_CHRNLUNG,pat_pr.CM_COAG,pat_pr.CM_DEPRESS,pat_pr.CM_DM,
```

```
pat_pr.CM_DMCX,pat_pr.CM_DRUG,pat_pr.CM_HTN_C,pat_pr.CM_HYPOTHY,pat_pr.CM_LIVER,pat_pr.CM_LYMPH,pat_pr.CM_LYTES,pat_pr.CM_METS,pat_pr.CM_NEURO,pat_pr.CM_OBESE,pat_pr.CM_PARA,
```

```
pat_pr.CM_PERIVASC,pat_pr.CM_PSYCH,pat_pr.CM_PULMCIRC,pat_pr.CM_RENLFAIL,pat_pr.CM_TUMOR,pat_pr.CM_ULCER,pat_pr.CM_VALVE,pat_pr.CM_WGHTLOSS);
```

```
end if;
```

```
  end if; --dqtr = 1 or 2 or 3
end loop;
```

```
commit;
END;
```

```
/
```

```

*/

/*
-- populate 2006
SET serveroutput on

declare
equiv_pr1 v_array:= v_array();
equiv_pr1_1 v_array:= v_array();
orig_pr1 v_array:=v_array();

equiv_pr2 v_array:= v_array();
equiv_pr2_1 v_array:= v_array();
orig_pr2 v_array:=v_array();

equiv_pr3 v_array:= v_array();
equiv_pr3_1 v_array:= v_array();
orig_pr3 v_array:=v_array();

equiv_dx1 v_array:= v_array();
equiv_dx1_1 v_array:= v_array();
orig_dx1 v_array:=v_array();

v_pr2_cnt NUMBER(10):=0;
v_pr3_cnt NUMBER(10):=0;
v_dx1_cnt NUMBER(10):=0;
i number(10):=0;

BEGIN

--populate pat_pr
for pat_pr in (select nc_6.key, dqtr, age_num, female, race, aweekend,
elective,pr1,pr2,pr3,dx1,died,LOS,dispuniform,

CM_AIDS,CM_ALCOHOL,CM_ANEMDEF,CM_ARTH,CM_BLDLOSS,CM_CHF,CM_CHRNLU
NG,CM_COAG,CM_DEPRESS,CM_DM,

CM_DMCX,CM_DRUG,CM_HTN_C,CM_HYPOTHY,CM_LIVER,CM_LYMPH,CM_LYTES,CM_
METS,CM_NEURO,CM_OBESE,CM_PARA,CM_PERIVASC,

CM_PSYCH,CM_PULMCIRC,CM_RENLFAIL,CM_TUMOR,CM_ULCER,CM_VALVE,CM_WG
HTLOSS
from nis_2006_core nc_6, nis_2006_severity ns_6 where nc_6.key = ns_6.key
and pr1 is not null and pr1 !='A' and pr1 !='B' and pr1 !='C' and dqtr in ('1','2','3','4') )
loop
i:= i+1;

if i = 100 then

```

```

        insert into debug_script values('2006 data','100 records',SYSTIMESTAMP);
        i:= 0;
        commit;
    end if;

    if (pat_pr.dqtr ='1') OR (pat_pr.dqtr ='2') OR (pat_pr.dqtr ='3') -- need to upgrade to
    2007 and then 2008
    then

        orig_pr1.delete;
        equiv_pr1.delete;
        equiv_pr1_1.delete;
        orig_pr1.extend;
        orig_pr1(1):= pat_pr.pr1;
        equiv_pr1 := get_equiv('2006',orig_pr1,'p');
        equiv_pr1_1 := get_equiv('2007',equiv_pr1,'p');

-- for elem in 1 .. equiv_pr1.count loop
--     dbms_output.put_line('pr1' || ':' || pat_pr.pr1||', equiv_pr1' || ':' ||
trim(equiv_pr1(elem)));
-- end loop;

-- dbms_output.put_line('pr2' || ':' || pat_pr.pr2);
-- get pr2 equiv
if(pat_pr.pr2 is not null) and (pat_pr.pr2 !='A') and (pat_pr.pr2 !='B') and (pat_pr.pr2
!='C')
THEN
    orig_pr2.delete;
    equiv_pr2.delete;
    equiv_pr2_1.delete;
    orig_pr2.extend;
    orig_pr2(1):= pat_pr.pr2;
    equiv_pr2 := get_equiv('2006',orig_pr2,'p');
    equiv_pr2_1 := get_equiv('2007',equiv_pr2,'p');

    else
    orig_pr2.delete;
    equiv_pr2.delete;
    equiv_pr2_1.delete;
    equiv_pr2_1.extend;
    equiv_pr2_1(1):= 'AAA';

end if; -- pr1 is null

-- for elem in 1 .. equiv_pr2.count loop
--     dbms_output.put_line('pr2' || ':' || pat_pr.pr2||', equiv_pr2' || ':' ||
trim(equiv_pr2(elem)));
-- end loop;

```

```

-- dbms_output.put_line('pr3' || ':' || pat_pr.pr3);
-- get pr3 equiv
if(pat_pr.pr3 is not null) and (pat_pr.pr3 !='A') and (pat_pr.pr3 !='B') and (pat_pr.pr3
!='C')
THEN
  orig_pr3.delete;
  equiv_pr3.delete;
  equiv_pr3_1.delete;
  orig_pr3.extend;
  orig_pr3(1):= pat_pr.pr3;
  equiv_pr3 := get_equiv('2006',orig_pr3,'p');
  equiv_pr3_1 := get_equiv('2007',equiv_pr3,'p');

else
  orig_pr3.delete;
  equiv_pr3.delete;
  equiv_pr3_1.delete;
  equiv_pr3_1.extend;
  equiv_pr3_1(1):= 'AAA';

end if; -- pr3 is null

-- for elem in 1 .. equiv_pr3.count loop
--   dbms_output.put_line('pr3' || ':' || pat_pr.pr3||', equiv_pr3' || ':' ||
trim(equiv_pr3(elem)));
-- end loop;

-- dbms_output.put_line('dx1' || ':' || pat_pr.dx1);
-- get dx1 equiv
if(pat_pr.dx1 is not null) and (pat_pr.dx1 !='A') and (pat_pr.dx1 !='B') and (pat_pr.dx1
!='C')
THEN
  orig_dx1.delete;
  equiv_dx1.delete;
  equiv_dx1_1.delete;
  orig_dx1.extend;
  orig_dx1(1):= pat_pr.dx1;
  equiv_dx1 := get_equiv('2006',orig_dx1,'d');
  equiv_dx1_1 := get_equiv('2007',equiv_dx1,'d');

else
  orig_dx1.delete;
  equiv_dx1.delete;
  equiv_dx1_1.delete;
  equiv_dx1_1.extend;
  equiv_dx1_1(1):= 'AAA';

```

```

end if; -- dx1 is null

-----
elsif pat_pr.dqtr = '4' then

-- get pr1 equiv
orig_pr1.delete;
equiv_pr1_1.delete;
orig_pr1.extend;
orig_pr1(1):= pat_pr.pr1;
equiv_pr1_1 := get_equiv('2007',orig_pr1,'p');

-- for elem in 1 .. equiv_pr1.count loop
--   dbms_output.put_line('pr1' || ':' || pat_pr.pr1||', equiv_pr1' || ':' ||
trim(equiv_pr1(elem)));
-- end loop;

-- dbms_output.put_line('pr2' || ':' || pat_pr.pr2);
-- get pr2 equiv
if(pat_pr.pr2 is not null) and (pat_pr.pr2 !='A') and (pat_pr.pr2 !='B') and (pat_pr.pr2
!='C')
THEN
orig_pr2.delete;
equiv_pr2_1.delete;
orig_pr2.extend;
orig_pr2(1):= pat_pr.pr2;
equiv_pr2_1 := get_equiv('2007',orig_pr2,'p');

else
orig_pr2.delete;
equiv_pr2_1.delete;
equiv_pr2_1.extend;
equiv_pr2_1(1):= 'AAA';

end if; -- pr1 is null

-- for elem in 1 .. equiv_pr2.count loop
--   dbms_output.put_line('pr2' || ':' || pat_pr.pr2||', equiv_pr2' || ':' ||
trim(equiv_pr2(elem)));
-- end loop;

-- dbms_output.put_line('pr3' || ':' || pat_pr.pr3);
-- get pr3 equiv
if(pat_pr.pr3 is not null) and (pat_pr.pr3 !='A') and (pat_pr.pr3 !='B') and (pat_pr.pr3
!='C')
THEN
orig_pr3.delete;
equiv_pr3_1.delete;

```



```

orig_pr3.extend;
orig_pr3(1):= pat_pr.pr3;
equiv_pr3_1 := get_equiv('2007',orig_pr3,'p');

else
orig_pr3.delete;
equiv_pr3_1.delete;
equiv_pr3_1.extend;
equiv_pr3_1(1):= 'AAA';

end if; -- pr3 is null

-- for elem in 1 .. equiv_pr3.count loop
--   dbms_output.put_line('pr3' || ':' || pat_pr.pr3||', equiv_pr3' || ':' ||
trim(equiv_pr3(elem)));
-- end loop;

-- dbms_output.put_line('dx1' || ':' || pat_pr.dx1);
-- get dx1 equiv
if(pat_pr.dx1 is not null) and (pat_pr.dx1 !='A') and (pat_pr.dx1 !='B') and (pat_pr.dx1
!='C')
THEN
orig_dx1.delete;
equiv_dx1_1.delete;
orig_dx1.extend;
orig_dx1(1):= pat_pr.dx1;
equiv_dx1_1 := get_equiv('2007',orig_dx1,'d');

else
orig_dx1.delete;
equiv_dx1_1.delete;
equiv_dx1_1.extend;
equiv_dx1_1(1):= 'AAA';

end if; -- dx1 is null
--- for elem in 1 .. equiv_dx1.count loop
--   dbms_output.put_line('$$$ equiv_elem_dx' || ':' || equiv_dx1(elem));-
-- end loop;

end if; -- dqtr = 1,2,3,4

for elem1 in 1 .. equiv_pr1_1.count loop
  for elem2 in 1 .. equiv_pr2_1.count loop
    for elem3 in 1 .. equiv_pr3_1.count loop
      for elemdx in 1 .. equiv_dx1_1.count loop

```

```

if substr(equiv_pr1_1(elem1),1,1) IN ('0','1','2') then
  insert into pat_pr_0_2 (key,year,dqtr, age, female, race, weekend_adm,
elec_adm,pr1,pr2,pr3,dx1, o_pr1,o_pr2,o_pr3,o_dx1, died, LOS, dispo)
  values(pat_pr.key,'2006', pat_pr.dqtr, pat_pr.age_num, pat_pr.female, pat_pr.race,
pat_pr.aweekend,
pat_pr.elective,trim(equiv_pr1_1(elem1)),trim(replace(equiv_pr2_1(elem2),'AAA','')),tri
m(replace(equiv_pr3_1(elem3),'AAA','')),trim(replace(equiv_dx1_1(elemdx),'AAA','')),pa
t_pr.pr1,pat_pr.pr2,pat_pr.pr3,pat_pr.dx1,pat_pr.died,decode(pat_pr.LOS,null,null,'A',n
ull,'B',null,'C',null,to_number(pat_pr.LOS)),pat_pr.dispuniform);

```

```

insert into pat_como_0_2

```

```

(key,CM_AIDS,CM_ALCOHOL,CM_ANEMDEF,CM_ARTH,CM_BLDLOSS,CM_CHF,CM_CHR
NLUNG,CM_COAG,CM_DEPRESS,CM_DM,

```

```

CM_DMCX,CM_DRUG,CM_HTN_C,CM_HYPOTHY,CM_LIVER,CM_LYMPH,CM_LYTES,CM_
METS,CM_NEURO,CM_OBESE,CM_PARA,CM_PERIVASC,

```

```

CM_PSYCH,CM_PULMCIRC,CM_RENLFAIL,CM_TUMOR,CM_ULCER,CM_VALVE,CM_WG
HTLOSS)

```

```

VALUES

```

```

(pat_pr.key,pat_pr.CM_AIDS,pat_pr.CM_ALCOHOL,pat_pr.CM_ANEMDEF,pat_pr.CM_A
RTH,pat_pr.CM_BLDLOSS,pat_pr.CM_CHF,pat_pr.CM_CHRNLUNG,pat_pr.CM_COAG,pat
_pr.CM_DEPRESS,pat_pr.CM_DM,

```

```

pat_pr.CM_DMCX,pat_pr.CM_DRUG,pat_pr.CM_HTN_C,pat_pr.CM_HYPOTHY,pat_pr.C
M_LIVER,pat_pr.CM_LYMPH,pat_pr.CM_LYTES,pat_pr.CM_METS,pat_pr.CM_NEURO,pa
t_pr.CM_OBESE,pat_pr.CM_PARA,

```

```

pat_pr.CM_PERIVASC,pat_pr.CM_PSYCH,pat_pr.CM_PULMCIRC,pat_pr.CM_RENLFAIL,p
at_pr.CM_TUMOR,pat_pr.CM_ULCER,pat_pr.CM_VALVE,pat_pr.CM_WGHTLOSS);

```

```

elseif substr(equiv_pr1_1(elem1),1,1) = '3' then

```

```

  insert into pat_pr_3 (key,year,dqtr, age, female, race, weekend_adm,
elec_adm,pr1,pr2,pr3,dx1, o_pr1,o_pr2,o_pr3,o_dx1, died, LOS, dispo)
  values(pat_pr.key,'2006', pat_pr.dqtr, pat_pr.age_num, pat_pr.female, pat_pr.race,
pat_pr.aweekend,
pat_pr.elective,trim(equiv_pr1_1(elem1)),trim(replace(equiv_pr2_1(elem2),'AAA','')),tri
m(replace(equiv_pr3_1(elem3),'AAA','')),trim(replace(equiv_dx1_1(elemdx),'AAA','')),pa
t_pr.pr1,pat_pr.pr2,pat_pr.pr3,pat_pr.dx1,pat_pr.died,decode(pat_pr.LOS,null,null,'A',n
ull,'B',null,'C',null,to_number(pat_pr.LOS)),pat_pr.dispuniform);

```

```

insert into pat_como_3

```

```

(key,CM_AIDS,CM_ALCOHOL,CM_ANEMDEF,CM_ARTH,CM_BLDLOSS,CM_CHF,CM_CHR
NLUNG,CM_COAG,CM_DEPRESS,CM_DM,

CM_DMCX,CM_DRUG,CM_HTN_C,CM_HYPOTHY,CM_LIVER,CM_LYMPH,CM_LYTES,CM_
METS,CM_NEURO,CM_OBESE,CM_PARA,CM_PERIVASC,

CM_PSYCH,CM_PULMCIRC,CM_RENLFAIL,CM_TUMOR,CM_ULCER,CM_VALVE,CM_WG
HTLOSS)
VALUES

(pat_pr.key,pat_pr.CM_AIDS,pat_pr.CM_ALCOHOL,pat_pr.CM_ANEMDEF,pat_pr.CM_A
RTH,pat_pr.CM_BLDLOSS,pat_pr.CM_CHF,pat_pr.CM_CHRNLUNG,pat_pr.CM_COAG,pat
_pr.CM_DEPRESS,pat_pr.CM_DM,

pat_pr.CM_DMCX,pat_pr.CM_DRUG,pat_pr.CM_HTN_C,pat_pr.CM_HYPOTHY,pat_pr.C
M_LIVER,pat_pr.CM_LYMPH,pat_pr.CM_LYTES,pat_pr.CM_METS,pat_pr.CM_NEURO,pa
t_pr.CM_OBESE,pat_pr.CM_PARA,

pat_pr.CM_PERIVASC,pat_pr.CM_PSYCH,pat_pr.CM_PULMCIRC,pat_pr.CM_RENLFAIL,p
at_pr.CM_TUMOR,pat_pr.CM_ULCER,pat_pr.CM_VALVE,pat_pr.CM_WGHTLOSS);

elsif substr(equiv_pr1_1(elem1),1,1) = '4' then
insert into pat_pr_4 (key,year,dqtr, age, female, race, weekend_adm,
elec_adm,pr1,pr2,pr3,dx1, o_pr1,o_pr2,o_pr3,o_dx1, died, LOS, dispo)
values(pat_pr.key,'2006', pat_pr.dqtr, pat_pr.age_num, pat_pr.female, pat_pr.race,
pat_pr.aweekend,
pat_pr.elective,trim(equiv_pr1_1(elem1)),trim(replace(equiv_pr2_1(elem2),'AAA','')),tri
m(replace(equiv_pr3_1(elem3),'AAA','')),trim(replace(equiv_dx1_1(elemdx),'AAA','')),pa
t_pr.pr1,pat_pr.pr2,pat_pr.pr3,pat_pr.dx1,pat_pr.died,decode(pat_pr.LOS,null,null,'A',n
ull,'B',null,'C',null,to_number(pat_pr.LOS)),pat_pr.dispuniform);

insert into pat_como_4

(key,CM_AIDS,CM_ALCOHOL,CM_ANEMDEF,CM_ARTH,CM_BLDLOSS,CM_CHF,CM_CHR
NLUNG,CM_COAG,CM_DEPRESS,CM_DM,

CM_DMCX,CM_DRUG,CM_HTN_C,CM_HYPOTHY,CM_LIVER,CM_LYMPH,CM_LYTES,CM_
METS,CM_NEURO,CM_OBESE,CM_PARA,CM_PERIVASC,

CM_PSYCH,CM_PULMCIRC,CM_RENLFAIL,CM_TUMOR,CM_ULCER,CM_VALVE,CM_WG
HTLOSS)
VALUES

(pat_pr.key,pat_pr.CM_AIDS,pat_pr.CM_ALCOHOL,pat_pr.CM_ANEMDEF,pat_pr.CM_A
RTH,pat_pr.CM_BLDLOSS,pat_pr.CM_CHF,pat_pr.CM_CHRNLUNG,pat_pr.CM_COAG,pat
_pr.CM_DEPRESS,pat_pr.CM_DM,

```

```
pat_pr.CM_DMCX,pat_pr.CM_DRUG,pat_pr.CM_HTN_C,pat_pr.CM_HYPOTHY,pat_pr.CM_LIVER,pat_pr.CM_LYMPH,pat_pr.CM_LYTES,pat_pr.CM_METS,pat_pr.CM_NEURO,pat_pr.CM_OBESE,pat_pr.CM_PARA,
```

```
pat_pr.CM_PERIVASC,pat_pr.CM_PSYCH,pat_pr.CM_PULMCIRC,pat_pr.CM_RENLFAIL,pat_pr.CM_TUMOR,pat_pr.CM_ULCER,pat_pr.CM_VALVE,pat_pr.CM_WGHTLOSS);
```

```
elseif substr(equiv_pr1_1(elem1),1,1) = '5' then
  insert into pat_pr_5 (key,year,dqtr, age, female, race, weekend_adm,
  elec_adm,pr1,pr2,pr3,dx1, o_pr1,o_pr2,o_pr3,o_dx1, died, LOS, dispo)
  values(pat_pr.key,'2006', pat_pr.dqtr, pat_pr.age_num, pat_pr.female, pat_pr.race,
  pat_pr.aweekend,
  pat_pr.elective,trim(equiv_pr1_1(elem1)),trim(replace(equiv_pr2_1(elem2),'AAA','')),trim(
  replace(equiv_pr3_1(elem3),'AAA','')),trim(replace(equiv_dx1_1(elemdx),'AAA','')),pat
  _pr.pr1,pat_pr.pr2,pat_pr.pr3,pat_pr.dx1,pat_pr.died,decode(pat_pr.LOS,null,null,'A',n
  ull,'B',null,'C',null,to_number(pat_pr.LOS)),pat_pr.dispuniform);
```

```
insert into pat_como_5
```

```
(key,CM_AIDS,CM_ALCOHOL,CM_ANEMDEF,CM_ARTH,CM_BLDLOSS,CM_CHF,CM_CHRNLUNG,CM_COAG,CM_DEPRESS,CM_DM,
```

```
CM_DMCX,CM_DRUG,CM_HTN_C,CM_HYPOTHY,CM_LIVER,CM_LYMPH,CM_LYTES,CM_METS,CM_NEURO,CM_OBESE,CM_PARA,CM_PERIVASC,
```

```
CM_PSYCH,CM_PULMCIRC,CM_RENLFAIL,CM_TUMOR,CM_ULCER,CM_VALVE,CM_WGHTLOSS)
```

```
VALUES
```

```
(pat_pr.key,pat_pr.CM_AIDS,pat_pr.CM_ALCOHOL,pat_pr.CM_ANEMDEF,pat_pr.CM_ARTH,pat_pr.CM_BLDLOSS,pat_pr.CM_CHF,pat_pr.CM_CHRNLUNG,pat_pr.CM_COAG,pat_pr.CM_DEPRESS,pat_pr.CM_DM,
```

```
pat_pr.CM_DMCX,pat_pr.CM_DRUG,pat_pr.CM_HTN_C,pat_pr.CM_HYPOTHY,pat_pr.CM_LIVER,pat_pr.CM_LYMPH,pat_pr.CM_LYTES,pat_pr.CM_METS,pat_pr.CM_NEURO,pat_pr.CM_OBESE,pat_pr.CM_PARA,
```

```
pat_pr.CM_PERIVASC,pat_pr.CM_PSYCH,pat_pr.CM_PULMCIRC,pat_pr.CM_RENLFAIL,pat_pr.CM_TUMOR,pat_pr.CM_ULCER,pat_pr.CM_VALVE,pat_pr.CM_WGHTLOSS);
```

```
elseif substr(equiv_pr1_1(elem1),1,1) = '6' then
  insert into pat_pr_6 (key,year,dqtr, age, female, race, weekend_adm,
  elec_adm,pr1,pr2,pr3,dx1, o_pr1,o_pr2,o_pr3,o_dx1, died, LOS, dispo)
  values(pat_pr.key,'2006', pat_pr.dqtr, pat_pr.age_num, pat_pr.female, pat_pr.race,
  pat_pr.aweekend,
  pat_pr.elective,trim(equiv_pr1_1(elem1)),trim(replace(equiv_pr2_1(elem2),'AAA','')),trim(
  replace(equiv_pr3_1(elem3),'AAA','')),trim(replace(equiv_dx1_1(elemdx),'AAA','')),pat
```

```
t_pr.pr1,pat_pr.pr2,pat_pr.pr3,pat_pr.dx1,pat_pr.died,decode(pat_pr.LOS,null,null,'A',null,'B',null,'C',null,to_number(pat_pr.LOS)),pat_pr.dispuniform);
```

```
insert into pat_como_6
```

```
(key,CM_AIDS,CM_ALCOHOL,CM_ANEMDEF,CM_ARTH,CM_BLDLOSS,CM_CHF,CM_CHRNLUNG,CM_COAG,CM_DEPRESS,CM_DM,
```

```
CM_DMCX,CM_DRUG,CM_HTN_C,CM_HYPOTHY,CM_LIVER,CM_LYMPH,CM_LYTES,CM_METS,CM_NEURO,CM_OBESE,CM_PARA,CM_PERIVASC,
```

```
CM_PSYCH,CM_PULMCIRC,CM_RENLFAIL,CM_TUMOR,CM_ULCER,CM_VALVE,CM_WGHTLOSS)
```

```
VALUES
```

```
(pat_pr.key,pat_pr.CM_AIDS,pat_pr.CM_ALCOHOL,pat_pr.CM_ANEMDEF,pat_pr.CM_ARTH,pat_pr.CM_BLDLOSS,pat_pr.CM_CHF,pat_pr.CM_CHRNLUNG,pat_pr.CM_COAG,pat_pr.CM_DEPRESS,pat_pr.CM_DM,
```

```
pat_pr.CM_DMCX,pat_pr.CM_DRUG,pat_pr.CM_HTN_C,pat_pr.CM_HYPOTHY,pat_pr.CM_LIVER,pat_pr.CM_LYMPH,pat_pr.CM_LYTES,pat_pr.CM_METS,pat_pr.CM_NEURO,pat_pr.CM_OBESE,pat_pr.CM_PARA,
```

```
pat_pr.CM_PERIVASC,pat_pr.CM_PSYCH,pat_pr.CM_PULMCIRC,pat_pr.CM_RENLFAIL,pat_pr.CM_TUMOR,pat_pr.CM_ULCER,pat_pr.CM_VALVE,pat_pr.CM_WGHTLOSS);
```

```
elsif substr(equiv_pr1_1(elem1),1,1) = '7' then
```

```
insert into pat_pr_7 (key,year,dqtr, age, female, race, weekend_adm, elec_adm,pr1,pr2,pr3,dx1, o_pr1,o_pr2,o_pr3,o_dx1, died, LOS, dispo)
```

```
values(pat_pr.key,'2006', pat_pr.dqtr, pat_pr.age_num, pat_pr.female, pat_pr.race, pat_pr.aweekend,
```

```
pat_pr.elective,trim(equiv_pr1_1(elem1)),trim(replace(equiv_pr2_1(elem2),'AAA','')),trim(replace(equiv_pr3_1(elem3),'AAA','')),trim(replace(equiv_dx1_1(elemdx),'AAA','')),pat_pr.pr1,pat_pr.pr2,pat_pr.pr3,pat_pr.dx1,pat_pr.died,decode(pat_pr.LOS,null,null,'A',null,'B',null,'C',null,to_number(pat_pr.LOS)),pat_pr.dispuniform);
```

```
insert into pat_como_7
```

```
(key,CM_AIDS,CM_ALCOHOL,CM_ANEMDEF,CM_ARTH,CM_BLDLOSS,CM_CHF,CM_CHRNLUNG,CM_COAG,CM_DEPRESS,CM_DM,
```

```
CM_DMCX,CM_DRUG,CM_HTN_C,CM_HYPOTHY,CM_LIVER,CM_LYMPH,CM_LYTES,CM_METS,CM_NEURO,CM_OBESE,CM_PARA,CM_PERIVASC,
```

```
CM_PSYCH,CM_PULMCIRC,CM_RENLFAIL,CM_TUMOR,CM_ULCER,CM_VALVE,CM_WGHTLOSS)
```

```
VALUES
```

```
(pat_pr.key,pat_pr.CM_AIDS,pat_pr.CM_ALCOHOL,pat_pr.CM_ANEMDEF,pat_pr.CM_
ARTH,pat_pr.CM_BLDLOSS,pat_pr.CM_CHF,pat_pr.CM_CHRNLUNG,pat_pr.CM_COAG,pat
_pr.CM_DEPRESS,pat_pr.CM_DM,
```

```
pat_pr.CM_DMCX,pat_pr.CM_DRUG,pat_pr.CM_HTN_C,pat_pr.CM_HYPOTHY,pat_pr.C
M_LIVER,pat_pr.CM_LYMPH,pat_pr.CM_LYTES,pat_pr.CM_METS,pat_pr.CM_NEURO,pa
t_pr.CM_OBESE,pat_pr.CM_PARA,
```

```
pat_pr.CM_PERIVASC,pat_pr.CM_PSYCH,pat_pr.CM_PULMCIRC,pat_pr.CM_RENLFAIL,p
at_pr.CM_TUMOR,pat_pr.CM_ULCER,pat_pr.CM_VALVE,pat_pr.CM_WGHTLOSS);
```

```
elsif substr(equiv_pr1_1(elem1),1,1) = '8' then
```

```
insert into pat_pr_8 (key,year,dqtr, age, female, race, weekend_adm,
elec_adm,pr1,pr2,pr3,dx1, o_pr1,o_pr2,o_pr3,o_dx1, died, LOS, dispo)
values(pat_pr.key,'2006', pat_pr.dqtr, pat_pr.age_num, pat_pr.female, pat_pr.race,
pat_pr.aweekend,
pat_pr.elective,trim(equiv_pr1_1(elem1)),trim(replace(equiv_pr2_1(elem2),'AAA','')),tri
m(replace(equiv_pr3_1(elem3),'AAA','')),trim(replace(equiv_dx1_1(elemdx),'AAA','')),pa
t_pr.pr1,pat_pr.pr2,pat_pr.pr3,pat_pr.dx1,pat_pr.died,decode(pat_pr.LOS,null,null,'A',n
ull,'B',null,'C',null,to_number(pat_pr.LOS)),pat_pr.dispuniform);
```

```
insert into pat_como_8
```

```
(key,CM_AIDS,CM_ALCOHOL,CM_ANEMDEF,CM_ARTH,CM_BLDLOSS,CM_CHF,CM_CHR
NLUNG,CM_COAG,CM_DEPRESS,CM_DM,
```

```
CM_DMCX,CM_DRUG,CM_HTN_C,CM_HYPOTHY,CM_LIVER,CM_LYMPH,CM_LYTES,CM_
METS,CM_NEURO,CM_OBESE,CM_PARA,CM_PERIVASC,
```

```
CM_PSYCH,CM_PULMCIRC,CM_RENLFAIL,CM_TUMOR,CM_ULCER,CM_VALVE,CM_WG
HTLOSS)
```

```
VALUES
```

```
(pat_pr.key,pat_pr.CM_AIDS,pat_pr.CM_ALCOHOL,pat_pr.CM_ANEMDEF,pat_pr.CM_A
RTH,pat_pr.CM_BLDLOSS,pat_pr.CM_CHF,pat_pr.CM_CHRNLUNG,pat_pr.CM_COAG,pat
_pr.CM_DEPRESS,pat_pr.CM_DM,
```

```
pat_pr.CM_DMCX,pat_pr.CM_DRUG,pat_pr.CM_HTN_C,pat_pr.CM_HYPOTHY,pat_pr.C
M_LIVER,pat_pr.CM_LYMPH,pat_pr.CM_LYTES,pat_pr.CM_METS,pat_pr.CM_NEURO,pa
t_pr.CM_OBESE,pat_pr.CM_PARA,
```

```
pat_pr.CM_PERIVASC,pat_pr.CM_PSYCH,pat_pr.CM_PULMCIRC,pat_pr.CM_RENLFAIL,p
at_pr.CM_TUMOR,pat_pr.CM_ULCER,pat_pr.CM_VALVE,pat_pr.CM_WGHTLOSS);
```

```
elsif substr(equiv_pr1_1(elem1),1,1) = '9' then
```

```
insert into pat_pr_9 (key,year,dqtr, age, female, race, weekend_adm,
elec_adm,pr1,pr2,pr3,dx1, o_pr1,o_pr2,o_pr3,o_dx1, died, LOS, dispo)
```

```

values(pat_pr.key,'2006', pat_pr.dqtr, pat_pr.age_num, pat_pr.female, pat_pr.race,
pat_pr.aweekend,
pat_pr.elective,trim(equiv_pr1_1(elem1)),trim(replace(equiv_pr2_1(elem2),'AAA','')),trim
(replace(equiv_pr3_1(elem3),'AAA','')),trim(replace(equiv_dx1_1(elemdx),'AAA','')),pa
t_pr.pr1,pat_pr.pr2,pat_pr.pr3,pat_pr.dx1,pat_pr.died,decode(pat_pr.LOS,null,null,'A',n
ull,'B',null,'C',null,to_number(pat_pr.LOS)),pat_pr.dispuniform);

```

```

insert into pat_como_9

```

```

(key,CM_AIDS,CM_ALCOHOL,CM_ANEMDEF,CM_ARTH,CM_BLDLOSS,CM_CHF,CM_CHR
NLUNG,CM_COAG,CM_DEPRESS,CM_DM,

```

```

CM_DMCX,CM_DRUG,CM_HTN_C,CM_HYPOTHY,CM_LIVER,CM_LYMPH,CM_LYTES,CM_
METS,CM_NEURO,CM_OBESE,CM_PARA,CM_PERIVASC,

```

```

CM_PSYCH,CM_PULMCIRC,CM_RENLFAIL,CM_TUMOR,CM_ULCER,CM_VALVE,CM_WG
HTLOSS)

```

```

VALUES

```

```

(pat_pr.key,pat_pr.CM_AIDS,pat_pr.CM_ALCOHOL,pat_pr.CM_ANEMDEF,pat_pr.CM_A
RTH,pat_pr.CM_BLDLOSS,pat_pr.CM_CHF,pat_pr.CM_CHRNLUNG,pat_pr.CM_COAG,pat
_pr.CM_DEPRESS,pat_pr.CM_DM,

```

```

pat_pr.CM_DMCX,pat_pr.CM_DRUG,pat_pr.CM_HTN_C,pat_pr.CM_HYPOTHY,pat_pr.C
M_LIVER,pat_pr.CM_LYMPH,pat_pr.CM_LYTES,pat_pr.CM_METS,pat_pr.CM_NEURO,pa
t_pr.CM_OBESE,pat_pr.CM_PARA,

```

```

pat_pr.CM_PERIVASC,pat_pr.CM_PSYCH,pat_pr.CM_PULMCIRC,pat_pr.CM_RENLFAIL,p
at_pr.CM_TUMOR,pat_pr.CM_ULCER,pat_pr.CM_VALVE,pat_pr.CM_WGHTLOSS);

```

```

end if;

```

```

    end loop; -- dx1

```

```

    end loop; -- pr3

```

```

    end loop; -- pr2

```

```

    end loop; -- pr1

```

```

end loop;

```

```

commit;

```

```

END;

```

```

/

```

```

*/

```

```

/*

```

```
select count(*) from pat_pr_0_2;
select count(*) from pat_como_0_2;
select count(*) from pat_pr_3;
select count(*) from pat_como_3;
select count(*) from pat_pr_4;
select count(*) from pat_como_4;
select count(*) from pat_pr_5;
select count(*) from pat_como_5;
select count(*) from pat_pr_6;
select count(*) from pat_como_6;
select count(*) from pat_pr_7;
select count(*) from pat_como_7;
select count(*) from pat_pr_8;
select count(*) from pat_como_8;
select count(*) from pat_pr_9;
select count(*) from pat_como_9;

*/

/*
-- populate pr_counts
set serveroutput on

declare

tot_cnt Number(10):=0;
tot_died Number(10):=0;
avg_los Number(10):=0;
tot_dispo_home Number(10):=0;
tot_dispo_facility Number(10):=0;
tot_dispo_hhc Number(10):=0;

BEGIN

for proc in (select distinct pr1 from pat_pr_0_2 )
loop

select count(*) into tot_cnt from pat_pr_0_2 where pr1 = proc.pr1;

select count(*) into tot_died from pat_pr_0_2 where pr1 = proc.pr1 and died ='1';

select AVG(LOS) into avg_los from pat_pr_0_2 where pr1 = proc.pr1 and LOS is not
null;

select count(*) into tot_dispo_home from pat_pr_0_2 where pr1 = proc.pr1 and dispo
='1';
```



```
select count(*) into tot_dispo_facility from pat_pr_0_2 where pr1 = proc.pr1 and dispo
IN ('2','5');
```

```
select count(*) into tot_dispo_hhc from pat_pr_0_2 where pr1 = proc.pr1 and dispo
='6';
```

```
insert into pr_counts
values(proc.pr1,tot_cnt,tot_died,round(avg_los,1),tot_dispo_home,tot_dispo_facility,to
t_dispo_hhc);
```

```
end loop;
```

```
insert into debug_script values('pat_pr_0_2 counts','done',SYSTIMESTAMP);
commit;
```

```
for proc in (select distinct pr1 from pat_pr_3)
loop
```

```
select count(*) into tot_cnt from pat_pr_3 where pr1 = proc.pr1;
```

```
select count(*) into tot_died from pat_pr_3 where pr1 = proc.pr1 and died ='1';
```

```
select AVG(LOS) into avg_los from pat_pr_3 where pr1 = proc.pr1 and LOS is not null;
```

```
select count(*) into tot_dispo_home from pat_pr_3 where pr1 = proc.pr1 and dispo
='1';
```

```
select count(*) into tot_dispo_facility from pat_pr_3 where pr1 = proc.pr1 and dispo IN
('2','5');
```

```
select count(*) into tot_dispo_hhc from pat_pr_3 where pr1 = proc.pr1 and dispo ='6';
```

```
insert into pr_counts
values(proc.pr1,tot_cnt,tot_died,round(avg_los,1),tot_dispo_home,tot_dispo_facility,to
t_dispo_hhc);
```

```
end loop;
```

```
insert into debug_script values('pat_pr_3 counts','done',SYSTIMESTAMP);
commit;
```

```
for proc in (select distinct pr1 from pat_pr_4)
loop

  select count(*) into tot_cnt from pat_pr_4 where pr1 = proc.pr1;

  select count(*) into tot_died from pat_pr_4 where pr1 = proc.pr1 and died = '1';

  select AVG(LOS) into avg_los from pat_pr_4 where pr1 = proc.pr1 and LOS is not null;

  select count(*) into tot_dispo_home from pat_pr_4 where pr1 = proc.pr1 and dispo
= '1';

  select count(*) into tot_dispo_facility from pat_pr_4 where pr1 = proc.pr1 and dispo IN
('2','5');

  select count(*) into tot_dispo_hhc from pat_pr_4 where pr1 = proc.pr1 and dispo = '6';

insert into pr_counts
values(proc.pr1,tot_cnt,tot_died,round(avg_los,1),tot_dispo_home,tot_dispo_facility,tot
_dispo_hhc);

end loop;

insert into debug_script values('pat_pr_4 counts','done',SYSTIMESTAMP);
commit;

for proc in (select distinct pr1 from pat_pr_5)
loop

  select count(*) into tot_cnt from pat_pr_5 where pr1 = proc.pr1;

  select count(*) into tot_died from pat_pr_5 where pr1 = proc.pr1 and died = '1';

  select AVG(LOS) into avg_los from pat_pr_5 where pr1 = proc.pr1 and LOS is not null;

  select count(*) into tot_dispo_home from pat_pr_5 where pr1 = proc.pr1 and dispo
= '1';

  select count(*) into tot_dispo_facility from pat_pr_5 where pr1 = proc.pr1 and dispo IN
('2','5');

  select count(*) into tot_dispo_hhc from pat_pr_5 where pr1 = proc.pr1 and dispo = '6';

insert into pr_counts
values(proc.pr1,tot_cnt,tot_died,round(avg_los,1),tot_dispo_home,tot_dispo_facility,tot
_dispo_hhc);
```

```
end loop;

insert into debug_script values('pat_pr_5 counts','done',SYSTIMESTAMP);
commit;

for proc in (select distinct pr1 from pat_pr_6)
loop

select count(*) into tot_cnt from pat_pr_6 where pr1 = proc.pr1;

select count(*) into tot_died from pat_pr_6 where pr1 = proc.pr1 and died = '1';

select AVG(LOS) into avg_los from pat_pr_6 where pr1 = proc.pr1 and LOS is not null;

select count(*) into tot_dispo_home from pat_pr_6 where pr1 = proc.pr1 and dispo
='1';

select count(*) into tot_dispo_facility from pat_pr_6 where pr1 = proc.pr1 and dispo IN
('2','5');

select count(*) into tot_dispo_hhc from pat_pr_6 where pr1 = proc.pr1 and dispo = '6';

insert into pr_counts
values(proc.pr1,tot_cnt,tot_died,round(avg_los,1),tot_dispo_home,tot_dispo_facility,tot
_dispo_hhc);

end loop;

insert into debug_script values('pat_pr_6 counts','done',SYSTIMESTAMP);
commit;

for proc in (select distinct pr1 from pat_pr_7)
loop

select count(*) into tot_cnt from pat_pr_7 where pr1 = proc.pr1;

select count(*) into tot_died from pat_pr_7 where pr1 = proc.pr1 and died = '1';

select AVG(LOS) into avg_los from pat_pr_7 where pr1 = proc.pr1 and LOS is not null;

select count(*) into tot_dispo_home from pat_pr_7 where pr1 = proc.pr1 and dispo
='1';
```

```
select count(*) into tot_dispo_facility from pat_pr_7 where pr1 = proc.pr1 and dispo IN ('2','5');
```

```
select count(*) into tot_dispo_hhc from pat_pr_7 where pr1 = proc.pr1 and dispo ='6';
```

```
insert into pr_counts  
values(proc.pr1,tot_cnt,tot_died,round(avg_los,1),tot_dispo_home,tot_dispo_facility,tot_dispo_hhc);
```

```
end loop;
```

```
insert into debug_script values('pat_pr_7 counts','done',SYSTIMESTAMP);  
commit;
```

```
for proc in (select distinct pr1 from pat_pr_8)  
loop
```

```
select count(*) into tot_cnt from pat_pr_8 where pr1 = proc.pr1;
```

```
select count(*) into tot_died from pat_pr_8 where pr1 = proc.pr1 and died ='1';
```

```
select AVG(LOS) into avg_los from pat_pr_8 where pr1 = proc.pr1 and LOS is not null;
```

```
select count(*) into tot_dispo_home from pat_pr_8 where pr1 = proc.pr1 and dispo ='1';
```

```
select count(*) into tot_dispo_facility from pat_pr_8 where pr1 = proc.pr1 and dispo IN ('2','5');
```

```
select count(*) into tot_dispo_hhc from pat_pr_8 where pr1 = proc.pr1 and dispo ='6';
```

```
insert into pr_counts  
values(proc.pr1,tot_cnt,tot_died,round(avg_los,1),tot_dispo_home,tot_dispo_facility,tot_dispo_hhc);
```

```
end loop;
```

```
insert into debug_script values('pat_pr_8 counts','done',SYSTIMESTAMP);  
commit;
```

```
for proc in (select distinct pr1 from pat_pr_9)  
loop
```

```
select count(*) into tot_cnt from pat_pr_9 where pr1 = proc.pr1;

select count(*) into tot_died from pat_pr_9 where pr1 = proc.pr1 and died ='1';

select AVG(LOS) into avg_los from pat_pr_9 where pr1 = proc.pr1 and LOS is not null;

select count(*) into tot_dispo_home from pat_pr_9 where pr1 = proc.pr1 and dispo
='1';

select count(*) into tot_dispo_facility from pat_pr_9 where pr1 = proc.pr1 and dispo IN
('2','5');

select count(*) into tot_dispo_hhc from pat_pr_9 where pr1 = proc.pr1 and dispo ='6';

insert into pr_counts
values(proc.pr1,tot_cnt,tot_died,round(avg_los,1),tot_dispo_home,tot_dispo_facility,to
t_dispo_hhc);

end loop;

insert into debug_script values('pat_pr_9 counts','done',SYSTIMESTAMP);
commit;

end;
*/

/*
-- create age_category

update pat_pr_0_2
set age_category = case
    when age between 0 and 10 then '0 to 10'
    when age between 11 and 20 then '11 to 20'
    when age between 21 and 30 then '21 to 30'
    when age between 31 and 40 then '31 to 40'
    when age between 41 and 50 then '41 to 50'
    when age between 51 and 60 then '51 to 60'
    when age between 61 and 70 then '61 to 70'
    when age between 71 and 80 then '71 to 80'
    when age between 81 and 90 then '81 to 90'
    when age between 91 and 140 then 'Above 90'
    else 'Other'
end;

update pat_pr_3
```

```
set age_category = case
  when age between 0 and 10 then '0 to 10'
  when age between 11 and 20 then '11 to 20'
  when age between 21 and 30 then '21 to 30'
  when age between 31 and 40 then '31 to 40'
  when age between 41 and 50 then '41 to 50'
  when age between 51 and 60 then '51 to 60'
  when age between 61 and 70 then '61 to 70'
  when age between 71 and 80 then '71 to 80'
  when age between 81 and 90 then '81 to 90'
  when age between 91 and 140 then 'Above 90'
  else 'Other'
end;
```

```
update pat_pr_4
set age_category = case
  when age between 0 and 10 then '0 to 10'
  when age between 11 and 20 then '11 to 20'
  when age between 21 and 30 then '21 to 30'
  when age between 31 and 40 then '31 to 40'
  when age between 41 and 50 then '41 to 50'
  when age between 51 and 60 then '51 to 60'
  when age between 61 and 70 then '61 to 70'
  when age between 71 and 80 then '71 to 80'
  when age between 81 and 90 then '81 to 90'
  when age between 91 and 140 then 'Above 90'
  else 'Other'
end;
```

```
update pat_pr_5
set age_category = case
  when age between 0 and 10 then '0 to 10'
  when age between 11 and 20 then '11 to 20'
  when age between 21 and 30 then '21 to 30'
  when age between 31 and 40 then '31 to 40'
  when age between 41 and 50 then '41 to 50'
  when age between 51 and 60 then '51 to 60'
  when age between 61 and 70 then '61 to 70'
  when age between 71 and 80 then '71 to 80'
  when age between 81 and 90 then '81 to 90'
  when age between 91 and 140 then 'Above 90'
  else 'Other'
end;
```

```
update pat_pr_6
set age_category = case
  when age between 0 and 10 then '0 to 10'
  when age between 11 and 20 then '11 to 20'
```

```
when age between 21 and 30 then '21 to 30'  
when age between 31 and 40 then '31 to 40'  
when age between 41 and 50 then '41 to 50'  
when age between 51 and 60 then '51 to 60'  
when age between 61 and 70 then '61 to 70'  
when age between 71 and 80 then '71 to 80'  
when age between 81 and 90 then '81 to 90'  
when age between 91 and 140 then 'Above 90'  
else 'Other'  
end;
```

```
update pat_pr_7  
set age_category = case  
  when age between 0 and 10 then '0 to 10'  
  when age between 11 and 20 then '11 to 20'  
  when age between 21 and 30 then '21 to 30'  
  when age between 31 and 40 then '31 to 40'  
  when age between 41 and 50 then '41 to 50'  
  when age between 51 and 60 then '51 to 60'  
  when age between 61 and 70 then '61 to 70'  
  when age between 71 and 80 then '71 to 80'  
  when age between 81 and 90 then '81 to 90'  
  when age between 91 and 140 then 'Above 90'  
  else 'Other'  
end;
```

```
update pat_pr_8  
set age_category = case  
  when age between 0 and 10 then '0 to 10'  
  when age between 11 and 20 then '11 to 20'  
  when age between 21 and 30 then '21 to 30'  
  when age between 31 and 40 then '31 to 40'  
  when age between 41 and 50 then '41 to 50'  
  when age between 51 and 60 then '51 to 60'  
  when age between 61 and 70 then '61 to 70'  
  when age between 71 and 80 then '71 to 80'  
  when age between 81 and 90 then '81 to 90'  
  when age between 91 and 140 then 'Above 90'  
  else 'Other'  
end;
```

```
update pat_pr_9  
set age_category = case  
  when age between 0 and 10 then '0 to 10'  
  when age between 11 and 20 then '11 to 20'  
  when age between 21 and 30 then '21 to 30'  
  when age between 31 and 40 then '31 to 40'  
  when age between 41 and 50 then '41 to 50'
```

```

    when age between 51 and 60 then '51 to 60'
    when age between 61 and 70 then '61 to 70'
    when age between 71 and 80 then '71 to 80'
    when age between 81 and 90 then '81 to 90'
    when age between 91 and 140 then 'Above 90'
    else 'Other'
end;

COMMIT;
*/

alter table pat_pr_0_2 add hospid varchar2(100);
alter table pat_pr_3 add hospid varchar2(100);
alter table pat_pr_4 add hospid varchar2(100);
alter table pat_pr_5 add hospid varchar2(100);
alter table pat_pr_6 add hospid varchar2(100);
alter table pat_pr_7 add hospid varchar2(100);
alter table pat_pr_8 add hospid varchar2(100);
alter table pat_pr_9 add hospid varchar2(100);

update pat_pr_0_2 pr set pr.hospid = (select hospid from nis_2008_core where key =
pr.key) where year = '2008';
update pat_pr_0_2 pr set pr.hospid = (select hospid from nis_2007_core where key =
pr.key) where year = '2007';
update pat_pr_0_2 pr set pr.hospid = (select hospid from nis_2006_core where key =
pr.key) where year = '2006';
commit;
update pat_pr_3 pr set pr.hospid = (select hospid from nis_2008_core where key =
pr.key) where year = '2008';
update pat_pr_3 pr set pr.hospid = (select hospid from nis_2007_core where key =
pr.key) where year = '2007';
update pat_pr_3 pr set pr.hospid = (select hospid from nis_2006_core where key =
pr.key) where year = '2006';
commit;
update pat_pr_4 pr set pr.hospid = (select hospid from nis_2008_core where key =
pr.key) where year = '2008';
update pat_pr_4 pr set pr.hospid = (select hospid from nis_2007_core where key =
pr.key) where year = '2007';
update pat_pr_4 pr set pr.hospid = (select hospid from nis_2006_core where key =
pr.key) where year = '2006';
commit;
update pat_pr_5 pr set pr.hospid = (select hospid from nis_2008_core where key =
pr.key) where year = '2008';
update pat_pr_5 pr set pr.hospid = (select hospid from nis_2007_core where key =
pr.key) where year = '2007';
update pat_pr_5 pr set pr.hospid = (select hospid from nis_2006_core where key =
pr.key) where year = '2006';

```



```
commit;
update pat_pr_6 pr set pr.hospid = (select hospid from nis_2008_core where key =
pr.key) where year = '2008';
update pat_pr_6 pr set pr.hospid = (select hospid from nis_2007_core where key =
pr.key) where year = '2007';
update pat_pr_6 pr set pr.hospid = (select hospid from nis_2006_core where key =
pr.key) where year = '2006';
commit;
update pat_pr_7 pr set pr.hospid = (select hospid from nis_2008_core where key =
pr.key) where year = '2008';
update pat_pr_7 pr set pr.hospid = (select hospid from nis_2007_core where key =
pr.key) where year = '2007';
update pat_pr_7 pr set pr.hospid = (select hospid from nis_2006_core where key =
pr.key) where year = '2006';
commit;
update pat_pr_8 pr set pr.hospid = (select hospid from nis_2008_core where key =
pr.key) where year = '2008';
update pat_pr_8 pr set pr.hospid = (select hospid from nis_2007_core where key =
pr.key) where year = '2007';
update pat_pr_8 pr set pr.hospid = (select hospid from nis_2006_core where key =
pr.key) where year = '2006';
commit;
update pat_pr_9 pr set pr.hospid = (select hospid from nis_2008_core where key =
pr.key) where year = '2008';
update pat_pr_9 pr set pr.hospid = (select hospid from nis_2007_core where key =
pr.key) where year = '2007';
update pat_pr_9 pr set pr.hospid = (select hospid from nis_2006_core where key =
pr.key) where year = '2006';
commit;
```