**Database Assignment 02**

**Deadline: Sunday October 27th (11:59pm)**

# **1. Introduction**

The Centers for Medicare and Medicaid Services is a U.S. agency that administers the Medicare program and works in partnership with state governments to administer Medicaid. CMS makes health related data available to the public through its website [data.cms.gov](https://data.cms.gov/). In this assignment you will use one of CMS’ datasets that contains hospital charge data, referred to as IPPS dataset[[1]](#footnote-0)and available [here](https://data.cms.gov/Medicare-Inpatient/Inpatient-Prospective-Payment-System-IPPS-Provider/97k6-zzx3). The IPPS dataset contains information about 2017 charges of top 100 groups of similar clinical conditions (diagnosis) by different health providers in the U.S. and the correspondent amounts covered by health insurance. The “IPPS dataset” shows how the same treatment for a clinical condition can result in very different costs for the patients depending on the health care provider.

# **2. Data Model**

The “IPPS dataset” has 163K rows and 12 columns. The goal of this assignment is to have you download this dataset in a CSV format so you can later load all of its data content into a MySQL database named ipps. The ipps database has to be designed so that all of its tables are normalized up to 3NF (third normal form). All Data Definition Language (DDL) SQL statements (CREATE DATABASE and CREATE TABLE statements) and DCL (Data Control Language) statements (CREATE USER, GRANT statements) should be submitted in a file named ipps.sql. In summary, all ipps’s tables of your database should be normalized up to 3NF, have primary keys, and appropriate foreign keys with referential integrity constraints in place. You should also create a user named ipps with full control of all tables in the ipps database.

# **3. Data Load**

In order to load the “IPPS dataset” CSV file into your MySQL database you will have to write a data load program (preferable languages are Python or Java). This program should be named ipps.[py|java] (extension depends on the programming language of your choice) and it is the second deliverable of this assignment. Use the examples described in the MySQL connector resources (available [here](https://sites.google.com/view/thyagomota/courses/19fcs3810/19fcs3810-resources)) to help you write the data load program. You may find the following csvSplit function (in Python) useful when splitting the lines from the CSV file by comma into an array of data fields. csvSplit ignores commas when they appear inside a string.

|  |
| --- |
| # split by comma function that avoids splitting when commas appear within a string  def csvSplit(line):  data = []  while True:  s = ''  inString = False  for c in line:  line = line[1:]  if c == '"':  inString = not inString  continue  if not inString and c == ',':  break  else:  s += c  if len(s) == 0:  break  data.append(s)  return data |

|  |
| --- |
| // csvSplit implementation in Java  **public static** String[] csvSplit(String line) {  ArrayList<String> data = **new** ArrayList<String>();  **char** chars[] = line.toCharArray();  **int** i = 0;  **while** (**true**) {  String s = **""**;  **boolean** inString = **false**;  **for** (; i < chars.**length**; i++) {  **char** c = chars[i];  **if** (c == **'"'**) {  inString = !inString;  **continue**;  }  **if** (!inString && c == **','**) {  i++;  **break**;  }  **else**  s += c;  }  **if** (s.length() == 0)  **break**;  data.add(s);  }  String template[] = **new** String[1];  **return** data.toArray(template);  } |

Please note that **you are not allowed to pre-process the CSV file** other than using your ipps data load program. For example, you cannot manually pre-process the CSV file using a spreadsheet application. I will test your data load program using the CSV file obtained directly from data.cms.gov’s download - CSV option. The CSV file should be renamed to ipps.csv. Because this file is large (26.8MB) and it can be downloaded from the internet, there is no need to submit it through blackboard.

# **3. Rubric**

+5 the two deliverables (source code and SQL file) use the naming format requested ;

+15 MySQL database tables are normalized to 3NF;

+15 All tables are created correctly, having appropriate use of data types, readable names, NOT NULL used when appropriate, primary key defined correctly, and foreign key defined whenever is needed

+5 the requested user is created with privileges;

+10 All 100 DRGs are loaded

+10 All 306 HRRs are loaded

+20 All 3,337 Providers are loaded

+20 All 163,065 charges and payments are loaded

1. “Inpatient Prospective Payment System (IPPS) Provider Summary for the Top 100 Diagnosis-Related Groups (DRG).” [↑](#footnote-ref-0)