COL 362/632 Assignment 1
SQL Queries

20 Jan, 2024

Deadline

Submission of the complete implementation of the SQL Queries is due on **5 Feb, 2024, 11:59 PM**. All submissions are to be made on Moodle.

General Instructions

Follow all instructions. Submissions not following these instructions will not be evaluated and will be given Zero marks.

1. Kindly ensure that this assignment is completed independently. Collaboration with any external entities, including individuals, AI agents, websites, discussion forums, etc., is strictly prohibited. You are free to discuss and post questions on the piazza to seek clarification.
2. You are expected to use **PostgreSQL 15.x** for this assignment.
3. There are a total of **26** queries in this assignment.
4. **Marking scheme:** Queries carry different weights based on the perceived difficulty level of the query. These weights are not revealed to you (intentionally). The weightage will be released after evaluation.
5. Note that we will evaluate your assignment automatically, so take care that your folder names, file names, and directory structure should strictly follow the instructions. Also, we will do binary binary grading for each SQL query; there will be no partial grading.
6. **Note that there will be no deadline extensions.** But, you can use your three skip days. Refer to lecture notes on course organization for skip day rules.
7. Download the cleaned-up data from this link. It is a **.sql** file that can be imported directly to PostgreSQL. Refer to lecture notes on importing a `.sql` file into PostgreSQL.
8. A single zip has to be submitted. The zip should be structured such that
    - Upon deflating, all submission files should be under a directory with the student's registration number. For, suppose a student's registration number is 22XXCSXX123. In that case, the zip submission should be named 22XXCSXX123.zip, and upon deflating **all contained files** should be under the directory named ./22XXCSXX123 only (names should be in uppercase) - your submission might be rejected and not be evaluated if you do not adhere to these specifications.
    - There are three sections (each of different difficulty levels) in the assignment, i.e., "Easy", "Intermediate", and "Hard." For this, you have to create three different folders named "EASY", "INTERMEDIATE", and "HARD". (names should be in uppercase)
    - In each designated section for every question, generate a new **.sql** file following the naming convention **`<folder_name>_<question_number>.sql`**. For instance, the solution to question "1" within the "Easy" section should be saved in a file named "**EASY_1.sql**". Similarly, the response to question "2" in the "Easy" section must be saved in a file named "**EASY_2.sql**". Please do not write any comment or other text except SQL query in the **.sql** files.

- The final directory structure will look like below:

```
22XXCSXX123
├── EASY
│   ├── EASY_1.sql
│   ├── EASY_2.sql
│   └── ...
├── INTERMEDIATE
│   ├── INTERMEDIATE_1.sql
│   ├── INTERMEDIATE_2.sql
│   └── ...
└── HARD
    ├── HARD_1.sql
    ├── HARD_2.sql
    └── ...
```

1 Dataset

This assignment will use a sizeable real-world medical dataset. The aim of curating this dataset was to improve patient care through knowledge discovery and algorithm development. It provided critical care data for over 40,000 patients admitted to one of the leading international medical centers. The dataset has been cleaned to an extent and organized specifically for this assignment. The ER diagram of this dataset can be found here.
Important! Please **do not** share the dataset on any forum (including public and private).

1.1 Description

The design follows these general principles. Each patient is assigned a unique ID (subject_id). All the information regarding that patient is tagged with his/her subject_id. The patient's gender, anchor_age, dod, etc., is stored in **patients** table. Further, the database is comprised of the following main table.

Table Name	Description
patients	Information that is consistent for the lifetime of a patient
admissions	Detailed information about hospital stays

Table 1: Main Tables

These main tables are supplemented by the additional tables:

Table Name	Description
drgcode	Billed diagnosis-related group (DRG) codes for hospitalizations
d_labitems	Dimension table for labevents provides a description of all lab items
labevents	Laboratory measurements sourced from patient-derived specimens
icustays	Tracking information for ICU stays including admission and discharge times
d_icd_diagnosis	Dimension table for diagnoses_icd; provides a description of ICD-9/ICD-10 billed diagnoses
d_icd_procedures	Dimension table for procedures_icd; provides a description of ICD-9/ICD-10 billed procedures
diagnoses_icd	Billed ICD-9/ICD-10 diagnoses for hospitalizations
prescriptions	It provides information about prescribed medications
procedures_icd	Billed procedures for patients during their hospital stay

Table 2: Additional Tables

The following is the description of the columns in each table:
1. patients table

Column Name	Data Type	Description
subject_id	INTEGER NOT NULL	unique identifier which specifies an individual patient
gender	VARCHAR(1) NOT NULL	genotypical sex of the patient
anchor_age	INTEGER NOT NULL	patient's age in the anchor_year
anchor_year	INTEGER NOT NULL	shifted year for the patient
anchor_year_group	VARCHAR(255) NOT NULL	range of years - the patient's anchor_year occurred during this range
dod	TIMESTAMP(0)	de-identified date of death for the patient

Table 3: patients.

Example: a patient has an anchor_year of 2153, anchor_year_group of 2008 - 2010, and an anchor_age of 60. The year 2153 for the patient corresponds to 2008, 2009, or 2010. The patient was 60 in the shifted year of 2153, i.e. they were 60 in 2008, 2009, or 2010. A patient admission in 2154 will occur in 2009-2011, an admission in 2155 will occur in 2010-2012, so on.

Note: If a patient's anchor_age is over 89 in the anchor_year then anchor_age is set to 91, regardless of true age.

2. admissions table

Column Name	Data Type	Description
subject_id	INTEGER NOT NULL	unique identifier which specifies an individual patient
hadm_id	INTEGER NOT NULL	unique id represents a single patient's admission to the hospital (ranges from 2000000 - 2999999)
admittime	TIMESTAMP NOT NULL	date and time the patient was admitted to the hospital
dischtime	TIMESTAMP	date and time the patient was discharged from the hospital
deathtime	TIMESTAMP	time of in-hospital death for the patient (if died)
admission_type	VARCHAR(40) NOT NULL	classifies the urgency of the admission, has 9 types
admit_provider_id	VARCHAR(10)	anonymous identifier for the provider who admitted the patient
admission_location	VARCHAR(60)	location of the patient prior to arriving at the hospital
discharge_location	VARCHAR(60)	disposition of the patient after discharged from the hospital
insurance	VARCHAR(255)	information about patient demographics
language	VARCHAR(10)	information about patient demographics
marital_status	VARCHAR(30)	information about patient demographics
race	VARCHAR(80)	information about patient demographics
edregtime	TIMESTAMP	date and time of registration at emergency dept
edouttime	TIMESTAMP	date and time of discharge at emergency dept
hospital_expire_flag	SMALLINT	1 if patient died in hospital else 0

Table 4: admissions.

3. drgcodes table

Column Name	Data Type	Description
subject_id	INTEGER NOT NULL	unique identifier which specifies an individual patient
hadm_id	INTEGER NOT NULL	refer to Table. 4
drg_type	VARCHAR(4)	specific DRG ontology used for the code
drg_code	VARCHAR(10)	correspond to the primary reason for a patient's stay
description	VARCHAR(195)	description for the given DRG code
drg_severity	SMALLINT	patient severity of illness
drg_mortality	SMALLINT	likelihood of mortality

Table 5: drgcodes.

4. d_labitems table

Column Name	Data Type	Description
itemid	INTEGER	unique identifier for a laboratory concept
label	VARCHAR(50)	describes the concept represented by itemid
fluid	VARCHAR(50)	substance on which the measurement was made
category	VARCHAR(50)	higher level information as to the type of measurement eg. 'ABG' indicates that the measurement is an arterial blood gas

Table 6: d_labitems.

5. labevents table

Column Name	Data Type	Description
labevent_id	INTEGER NOT NULL	unique for every row
subject_id	INTEGER NOT NULL	refer to Table. 3
hadm_id	INTEGER	refer to Table. 4
specimen_id	INTEGER NOT NULL	id of specimen used for lab measurement
itemid	INTEGER NOT NULL	refer to Table. 6
order_provider_id	VARCHAR(10)	anonymous id for provider who ordered measurement
charttime	TIMESTAMP(0)	time when the laboratory measurement was charted usually the time at which the specimen was acquired
storetime	TIMESTAMP(0)	time when the measurement was made available in the laboratory system
value	VARCHAR(200)	result of the laboratory measurement
valuenum	DOUBLE PRECISION	the value cast as a numeric data if numeric
valueuom	VARCHAR(20)	unit of measurement
ref_range_lower	DOUBLE PRECISION	lower ref range of measurement
ref_range_upper	DOUBLE PRECISION	upper ref range of measurement
flag	VARCHAR(10)	indicates if the measurement is abnormal
priority	VARCHAR(7)	either routine or stat (urgent)
comments	TEXT	Deidentified text comments associated with measurement

Table 7: labevents.

6. ICU Stays table

Column Name	Data Type	Description
subject_id	INT	refer to Table. 3
hadm_id	INT	refer to Table. 4
stay_id	INT	unique to a patient ward stay
first_careunit	VARCHAR(20)	first ICU Type in which the patient was cared for
last_careunit	VARCHAR(20)	last ICU Type in which the patient was cared for
intime	TIMESTAMP(0)	date and time patient was transferred into ICU
outtime	TIMESTAMP(0)	date and time patient was transferred outof ICU
los	DOUBLE PRECISION	length of stay which may include one or more ICU units

Table 8: icustays.

7. d_icd_diagnoses

Column Name	Data Type	Description
icd_code	CHAR(7) NOT NULL	International Coding Definitions code for diagnoses
icd_version	INTEGER NOT NULL	version 9 or 10
long_title	VARCHAR(255)	meaning of the ICD code

Table 9: d_icd_diagnoses.

8. d_icd_procedures

Column Name	Data Type	Description
icd_code	CHAR(7) NOT NULL	International Coding Definitions code for procedures
icd_version	INTEGER NOT NULL	version 9 or 10
long_title	VARCHAR(255)	brief definition for the given procedure

Table 10: d_icd_procedures.

9. diagnoses_icd

Column Name	Data Type	Description
subject_id	INTEGER NOT NULL	refer to Table. 3
hadm_id	INTEGER NOT NULL	refer to Table. 4
seq_num	INTEGER NOT NULL	priority assigned to the diagnoses
icd_code	CHAR(7) NOT NULL	refer to Table. 9
icd_version	INTEGER NOT NULL	refer to Table. 9

Table 11: diagnoses_icd.

10. prescriptions

Column Name	Data Type	Description
subject_id	INTEGER NOT NULL	refer to Table. 3
hadm_id	INTEGER NOT NULL	refer to Table. 4
pharmacy_id	INTEGER NOT NULL	identifier of pharmacy
poe_id	VARCHAR(25)	id of prescription
poe_seq	INTEGER	seq num of prescription
order_provider_id	VARCHAR(10)	anonymous id of provider who initiated order
starttime, stoptime	TIMESTAMP(3)	prescribed start, stop time for medication
drug_type	VARCHAR(20) NOT NULL	one of 'MAIN', 'BASE', or 'ADDITIVE'
drug	VARCHAR(255) NOT NULL	description of the medication
formulary_drug_cd	VARCHAR(50)	hospital specific ontology used to order drugs
gsn	VARCHAR(255)	Generic Sequence Number
ndc	VARCHAR(25)	National Drug Code
prod_strength	VARCHAR(255)	composition of the prescribed medication
form_rx	VARCHAR(25)	container in which the dose is delivered
dose_val_rx	VARCHAR(100)	prescribed dose for the patient
dose_unit_rx	VARCHAR(50)	unit of measurement for the dose
form_val_disp	VARCHAR(50)	amount of the medication contained in a single dose
form_unit_disp	VARCHAR(50)	unit of measurement used for the formulary dosage
doses_per_24_hrs	REAL	number of doses per 24 hours
route	VARCHAR(50)	route of administration for the medication

Table 12: prescriptions.

11. procedures_icd

Column Name	Data Type	Description
subject_id	INTEGER NOT NULL	refer to Table. 3
hadm_id	INTEGER NOT NULL	refer to Table. 4
seq_num	INTEGER NOT NULL	priority assigned to the diagnoses
chartdate	DATE NOT NULL	date of the associated procedures
icd_code	CHAR(7) NOT NULL	refer to Table. 10
icd_version	INTEGER NOT NULL	refer to Table. 10

Table 13: procedures_icd.

2 Queries

2.1 Easy Queries

Note (for this section): In case of a tie, output all the answers following the order. Use the strings (along with the case) given in the query for string comparison.

1. Number of female patients with 'anchor_age' between 18 and 30 (Use only 'patients' table).
   Output: count
2. Patient who was admitted the most number of times in the hospital.
   Output: subject_id, num_admissions
   Order: subject_id (Ascending)
3. Number of 'ROUTINE' lab-events which resulted in an 'abnormal' test result (Use 'priority', 'flag' columns).
   Output: count
4. Number of unique procedures the patient with subject_id = '10000117' has undergone (Two procedures are considered different if either their icd_version or icd_code are different).
   Output: count
5. Person whose first_careunit of ICUSTAY is 'Coronary Care Unit (CCU)' most number of times.
   Output: subject_id, anchor_age, count
   Order: anchor_age (Descending), subject_id(Descending)
6. Number of admissions which resulted in a diagnosis of 'Cholera due to vibrio cholerae'
   Output: count
7. 'pharmacy_id' of the pharmacy which was least visited by patients (Patient visits are only counted once, even if the same patient visited the pharmacy multiple times across various admissions).
   Output: pharmacy_id, num_patients_visited
   Order: pharmacy_id (Ascending)
8. Patients who were diagnosed with 'Typhoid fever' and admitted in the ICU during the same admission. (Even if a person satisfies the given criteria in multiple admission, output only once)
   Output: subject_id, anchor_age
   Order: subject_id (Acsending), anchor_age (Ascending)
9. Number of Admissions for which one of the lab-event resulted in a n 'abnormal' output and the patient died during the same admission.
   Output: count
10. Patients with anchor_age less than 50, who were admitted multiple times and underwent *atleast one* same procedure during two different admissions.
    Output: subject_id, anchor_age
    Order: subject_id (Acsending), anchor_age (Ascending)

2.2 Intermediate Queries

1. Unique patients who were admitted to the ICU atleast 5 times during their hospital stay, along with the count of their ICU stays(Number of times they were admitted to the ICU). Filter out the top 1000 patients who had the longest ICU stays.

   Output: subject_id, count
   Order: count (Descending), subject_id (Descending)

2. Top 1000 most prescribed medications during the first 12 hours of admission for patients.

   Output: drug, prescription_count
   Order: prescription_count (Descending), drug (Descending)

3. Patients with multiple admissions, and the number of times they were diagnosed with any ailment related to the term 'ALCOHOLIC' (case insensitive) from the drgcodes table's description column(The diagnosis description should have the term 'alcoholic' in it, remember the term is case insensitive). Use only the dgrcodes table.

   Output: subject_id, diagnoses_count
   Order: diagnoses_count (Descending), subject_id (Descending)

4. Patients with an admission type of "URGENT" (Case Sensitive) who died during their hospital stay. Also mention the long_titles of the ailments that they had D_ICD_DIAGNOSES.long_title). Only display the first 1000 such records.

   Output: subject_id, hadm_id, icd_code, long_title
   Order: subject_id (Descending), hadm_id (Descending), icd_code (Descending), long_title (Descending)

5. Average duration(days) of ICU stays for patients who had a particular laboratory test (e.g.Labevents ITEMID=50878) during their stay. Include patient's subject_id and average duration of stay in ICU. Only return the first 1000 records. (While grouping columns make sure that records with different subject_id and hadm_id are counted separately as 2 different records. Since the subject_id for 2 records may be same but hadm_id might vary). Further, only consider records where the LOS column of ICUSTAYS table is not NULL.

   Output: subject_id, avg_stay_duration
   Order: avg_stay_duration (Descending), subject_id (Descending)

6. Patients who had at least 1 admission with the diagnosis code '5723' (use ICD_CODE column in DIAGNOSES_ICD). Include the total number of distinct admissions for each patient(use column 'admissions.hadm_id' for this), along with the earliest and latest admit times (admissions.admittime). Additionally, from this result, give the count of distinct records where the patient was diagnosed with '5723'(column name 'diagnosis_count' in the resulting table. Use the ICD_CODE column of diagnoses_icd table). Ensure that the results only include patients who had at least one such admission. Also limit your results to the first 1000 records. Make sure that when you group columns use only the subject_id column.

   Output: subject_id, gender, total_admissions, last_admission, first_admission, diagnosis_count
   Order: total_admissions (Descending), diagnosis_count (Descending), last_admission (Descending), first_admission (Descending), gender (Descending), subject_id (Descending)

7. Patients who had at least 5 ICU stays(Distinct stay_ids in ICUs). Include the total number of ICU stays, and the average length of stay across all ICU admissions. Additionally, filter the results to only include patients who had an ICU stay in any kind of MICU(Medical Intensive Care Unit) (FIRST_CAREUNIT or LAST_CAREUNIT of ICUSTAYS table must have term 'MICU' in their name case sensitive), and limit the output to the top 500 patients. Make sure that whenever you group records, records with different subject_id must be considered as 2 seperate records.

   Output: subject_id, total_stays, avg_length_of_stay
   Order: avg_length_of_stay (Descending), total_stays (Descending), subject_id (Descending)

8. Patients with a history of heart-related diagnoses (DIAGNOSES_ICD.icd_code should start from 'V4' Case Sensitive) who were prescribed a specific medication (PRESCRIPTIONS.DRUG should have 'prochlorperazine' or 'bupropion' in its name, case insensitive. So any drug named 'BUPROpion' or 'buto bupropion amine' or 'prochlorperazine 60' should be included by your query) Finally, filter the results to only include patients with more than one distinct diagnoses count (use DISTINCT DIAGNOSES_ICD.icd_code). Make sure that when you group columns in the resulting table, records having different distinct_diagnoses_count, subject_id and drug must be treated as separate records

   Output: subject_id, hadm_id, distinct_diagnoses_count, drug
   Order: distinct_diagnoses_count (Descending), subject_id (Descending), hadm_id (Descending), drug (Ascending)

9. Patients who were diagnosed with a heart condition (DIAGNOSES_ICDİCD_CODE starts with 'I21' case sensitive) during their first admission and were readmitted afterwards so the second admission's admittime must be greater than the first admission's discharge time. Retrieve only the first 1000 rows.

   Output: subject_id
   Order: subject_id (Descending)

10. Patients who have been prescribed the same medication during multiple admissions, along with details of the drug(PRESCRIPTIONS.DRUG). Retrieve only the first 1000 rows. Whenever you group columns, remember that records with different subject_id and drug columns(Prescriptions table) are treated as separate.

    Output: subject_id, anchor_year, drug
    Order: subject_id (Descending), anchor_year (Descending), drug (Descending).

2.3 Hard Queries

Graph-1: The concept of graphical analysis can be applied to this dataset. The below constructed graph will be used in some of the queries in this section. Consider all patients admitted in 500 of the earliest admissions. These patients will be the nodes of the graph. Consider here only 500 of the earliest admissions. There exists an undirected edge between 2 patients if they were admitted in the hospital at least once in an overlapping period with at least one common diagnosis in that overlapping admission. This forms an undirected unweighted graph. Note that there are no self edges in the graph.

1. Considering latest admissions of every patient only, percentage of female and male patients who died in the hospital after being diagnosed with disorders related to 'Meningitis' (case sensitive, use long_title from d_icd_diagnoses table) in their latest admission
   Output: gender, mortality_rate
   Order: mortality_rate (Ascending), gender (Descending)

2. Find top 245 diagnoses with the highest mortality rate. Mortality rate of a diagnosis can be considered the percentage of admissions where a patient died when he was diagnosed with a diagnosis in that admission. Consider all those patients that didn't die in an admission that they were diagnosed with one of these. Output the average anchor_age of these patients for each of these diagnoses with the long_title of these diagnoses.
   Output: long_title, survived_avg_age
   Order: long_title (Ascending), survived_avg_age (Descending)

3. Find the average length of ICU stay required by patients for every procedure (consider total length of stay in ICU for an admission in which they underwent the procedure). Output all patients that required less than average ICU stay in any admission that they underwent the procedure along with the icd_code and icd_version of the procedure. Output every patient and procedure combination only once. Limit the output to first 1000 rows.

Output: subject_id, gender, icd_code, icd_version
Order: subject_id (Ascending), icd_code (Descending), icd_version (Descending), gender (Ascending)

4. Using Graph-1, check if there exists a path of length exactly 3 between patients with subject_ids 18237734 and 13401124. Output a boolean value: True for yes and False for no.
   Output: pathexists

5. Using Graph-1, check if there exists a path of length less than or equal to 5 between patients with subject_ids 10001725 and 19438360. Output a boolean value: True for yes and False for no.
   Output: pathexists

6. Using Graph-1, Find the shortest path between patients with subject_ids 10001725 and 14370607. Limit search to path lengths with 5 or less edges in your query. Output the path length. Output 0 if no such path exists.
   Output: pathlength