University of Toronto
CSC343, Winter 2019

Assignment 2

Due: March 15, 11:59pm

Learning Goals

By the end of this assignment you should be able to:

- interpret specifications accurately
- read and interpret a new database schema
- write complex queries in SQL
- design datasets to test an SQL query
- embed SQL in a high-level language using JDBC
- recognize the limits of the expressive power of the standard SQL

General Instructions

We strongly encourage you to do your work for this assignment on the CS Teaching Labs. Your code must run on these machines in order to earn credit.

Download the data.zip file from the Quercus webpage includes:

- The database schema, ddl.sql
- A sample of data set in csv files

You can work with a partner for this assignment. You must declare your team (whether it is a team of one or of two students) and hand in your work electronically using MarkUs.

Schema

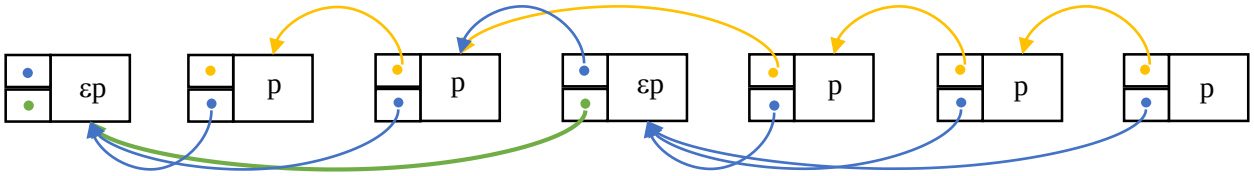
The data to be used in this assignment comes from ParlGov[1], a public database for political science. It contains information on political parties, elections and cabinets for most democracies that are part of the EU (European Union) or the OECD (Organization for Economic Co-operation and Development).

- The schema can record elections of two kinds:
  - a parliamentary election is an election within a country to choose a national government.
  - a European parliament election (or EP election) is an election, held across all European Union countries, to choose national representatives for the European parliament.

[1] http://www.parlgov.org

- The same schema is being used to represent election data from many countries, with numerous variations in their style of their form of governance, so some of the terminology may be used in an unfamiliar way.
- Each row of the election table records information about a single election. The row includes the ID of the previous Parliamentary election and the previous EP election (using attributes previous parliament election id and previous parliament election id). These two attributes essentially create two independent linked lists within the table. However, it's more complicated than that, because even a Parliamentary election has a reference to the previous EP election, and even an EP election has a reference to the previous Parliamentary election. This diagram may help you understand the structure embedded in the election table:

Time advances in this direction →



The orange arrows show the linked list of parliamentary elections going back through time, and the green arrow shows the linked list of EP elections going back through time. But we also store the references across election types that are also stored. When you look at the whole structure, you can see that it more than just two linked lists.

The election-result table records political alliances that form between different parties in an election. To represent that a set of parties formed an alliance in an election, the database singles one party out, called the head of the alliance, (it is arbitrary which party is the head) and has all the others refer to it in their alliance-id attribute. The other parties in the alliance refer to the head party by storing in alliance-id the id of the election result for the head party. The alliance-id value for the head party of the alliance is NULL. For example, if parties A, B, C, and D formed an alliance in election e1, and party A was chosen as the head of the alliance, then the table would include these rows:

id	election-id	party-id	alliance-id	seats	votes
id1	e1	A	NULL		
id2	e1	B	id1		
id3	e1	C	id1		
id4	e1	D	id1		

Note: although alliance-id sounds like a unique identifier for the alliance, it is not. It is a reference to one of the parties in the alliance.

Your code must work on any database instance that satisfies the schema, including instances with empty tables.

Part 1: SQL Statements

Write your SQL statement(s) for each question in separate files. You will submit two files for each ***question i*** as follows:

- In the qi.sql file you will define the table that is required for that query and insert data into the table you defined according to the given query specification
- In qi-order.sql, you simply order the table you created in qi.sql based on ordering criteria defined in the question.

In total, for this part, you should submit 12 files: q1.sql, q1-order.sql, q2.sql, q2-order.sql . . ., q6.sql, q6-order.sql

For example, suppose you were asked to find the sIDS of all students with a gpa greater than 3 and sort them by gpa. for the following table in q1 then you will take the following steps:

- In q1.sql:
  - create table q1(sID integer primary key, gpa integer);
  - insert into q1 select sID, gpa from student where gpa > 3;
- And then in q1-order.sql:
  - select * from q1 order by gpa;

You are encouraged to use views to make your queries more readable. However, each file should be entirely self-contained, and not depend on any other files; each will be run separately on a fresh database instance.

Each of your files must begin with the line **SET search-path TO** parlgov; Failure to do so will cause your query to raise an error.

The output from your queries must exactly match the specifications in the question, including attribute names, order and type of attributes, as well as the ordering of the tuples.

We will be testing your code in the **CS Teaching Labs environment** using PostgreSQL. It is your responsibility to make sure your code runs in this environment before the deadline. **Code which works on your machine but not on the CS Teaching Labs will not receive credit**.

IMPORTANT: We define the **winning party** to be the party that won the most votes. It is possible that several parties are tied for the most votes, in which case we say that there are multiple winning parties.

Write SQL queries for each of the following:

1. A political alliance is an agreement for cooperation between different political parties. We assume that each alliance is led by a party. Zero, one or more than one alliance might be formed in an election. In the *election-result* table, the row corresponding to the election result of a party that participates in an alliance links to leader party of the alliance by recording the election result id of the leader in *alliance-id* attribute. Note: the *alliance-id* attribute of the leader party of an alliance is NULL. Report the pair of parties that have been allies with each other in at least 30% of elections that have happened in a country.

Attribute	**Description**
countryId	id of a country
alliedPartyId1	id of an allied party
alliedPartyId2	id of an allied party
Order by	countryId descending, then alliedPartyId1 descending, then alliedPartyId2 descending
Everyone?	Every allied pair that satisfies the condition.
Duplicates?	No pair of parties hould be included more than once. Only include pairs that satisfy alliedPartyId1 <alliedPartyId2

2. A committed party is the one that has been a member of all cabinets in their country over the past 20 years. For each country, report the name of committed parties, their party families and their "regulation of the economy" value.

Attribute	**Description**
countryName	Name of a country
partyName	Name of a committed party
partyFamily	Name of a committed party's family if exists, otherwise, null.
stateMarket	Regulation of the economy property of the party if exists, otherwise, null.
Order by	countryName ascending, then partyName ascending, then stateMarket descending
Everyone?	Include only countries with committed parties
Duplicates?	There can be no duplicates.

3. Find parties that have won more than 3 times the average number of winning elections of parties of the same country. Report the country name, party name and the party's family name along with the total number of elections it has won. For each party included in the answer, report the id and year of the mostly recently won election.

Attribute	**Description**
countryName	Name of the country
partyName	Name of the party
partyFamily	Name of the family of a party
wonElections	Number of elections the party has won
mostRecentlyWonElectionId	The id of the election that was most recently won by this party
mostRecentlyWonElectionYear	The year of the election that was most recently won by this party
Order by	The name of the country ascending, then the number of won elections ascending, then the name of the party descending.
Everyone?	Include only countries and parties who meet the criteria of this question.
Duplicates?	Countries and party families can be included more than once with different party names.

4. For each of years between 1996 to 2016, both inclusive, for each country, and for each political party, report the name of country, the name of the party, and a description of the range into which the number of valid votes it received falls, in the following format: *(lb-ub]*, where *lb* is the lower bound of the range and *ub* is the upper bound of the range (for example, (20-30].) These are the range values to consider: non-zero and below 5 percent of valid votes inclusive, 5 to 10 percent of valid votes inclusive, 10 to 20 percent of valid votes inclusive, 20 to 30 percent of valid votes inclusive, 30 to 40 percent of valid votes inclusive, and above 40 percent of valid votes. If there is more than one election in the same country in the same year, use the average (across those elections) of the percent of valid votes that a party received. The range values are defined only for the parties and elections for which the number of votes is recorded. Where there were no parties in a given range, do not report that range. Where a country does not have any elections in a year, do not include it in the results.

Attribute	Description
year	year
countryName	name of a country
voteRange	the percentage range that the party falls into
partyName	short name of a party
Order by	year descending, countryName descending, voteRange descending and partyName descending
Everyone?	Every year where at least an election has happened should be included.
Duplicates?	No year-country-party combination occurs more than once.

5. The number of eligible voters and votes have been recorded for each election. The participation ratio of an election is the ratio of votes cast to the number of citizens who are eligible to vote. Note the participation ratio is a value between zero and one. Compute the participation ratio for each country, each year. If more than one election happens in a year in a country, compute the average participation ratio across those elections. Write a query to return the countries that had at least one election between 2001 to 2016, inclusive, and whose average election participation ratios during this period are *monotonically non-decreasing* (meaning that for Year Y and Year W, where at least one election has happened in each of them, if Y < W, then the average participation in Year Y is ≤ average participation in Year W). For such countries, report the name of the country and the average participation ratio per year between 2001 to 2016.

Attribute	Description
countryName	Name of the country
year	year
participationRatio	The average percentage ratio of citizens who cast votes in this year
Order by	The name of the country, descending, then the year, descending
Everyone?	Include only countries that meet the criteria of this question.
Duplicates?	No rows for a country, if there are no elections for a country between 2001 and 2016

6. The database also records the policy positions of political parties, including their "left-right dimension". Suppose the left-right range is divided into 5 intervals ([0,2), [2,4), [4,6), [6,8) and [8,10]). Create a table that is a histogram of parties and their left-right position. Note the values in party_position cannot be Null it must be Zero or greater than Zero.

Attribute	**Description**
countryName	Name of the country
r0-2	Number of parties whose left/right position is in [0,2).
r2-4	Number of parties whose left/right position is in [2,4).
r4-6	Number of parties whose left/right position is in [4,6).
r6-8	Number of parties whose left/right position is in [6,8).
r8-10	Number of parties whose left/right position is in [8,10].
Order by	countryName
Everyone?	Every country should be included, even if they have no parties with party position information.
Duplicates?	No country can be included more than once.

Part 2: Embedded SQL

Write a Java application that connects to a database containing election data and performs the functionality outlined below. The functionality is to be implemented as Java methods that act as wrappers around SQL queries.

General requirements

- **You may not use the standard input or output**.
- Write a method called connectDb() to connect to the database. It will receive the database URL, username, and password as parameters and it must call the getConnection() method passing it arguments that connectDb() received. **These values of these parameters must not be "hard-coded" in the methods.**
- You should not call connectDb() and disconnectDB() in the other methods you were asked to implement;
- **Do not change the interface** for any of the methods you were asked to implement.

- All your code must be written in **Assignment2.java**. This is the **only file** you have to submit for this part.
- JDBCSubmission is an abstract class that is provided to you. Do not make any changes in this file and do not submit this file.

- You will need to include the JDBC driver in your class path.

To Do

Open the starter code in Assignment2.java, and complete the following methods.

1. connectDB: Connect to a database with the supplied credentials.
2. disconnectDB: Disconnect from the database.
3. electionSequence: A method that, given a country, returns the list of elections in that country, in descending order of years, and the cabinets that have formed after that election and before the next election of the same type.
4. findSimilarPoliticians: A method that, given a president, returns other presidents that have similar comments and descriptions in the database. See section Similar Politicians below for details.

Similar Politicians Two politicians are considered similar if the textual information available about them is similar enough. You are provided with a helper method which computes the Jaccard similarity (see the note below) of two sets of strings. Use this similarity method to find the politicians whose similarity, calculated based on their description attributes, is above a given threshold.

NOTE: the "Jaccard" method provides for two given **sets** (e.g., sets of strings) a similarity score between 0 and 1. The Jaccard similarity for two sets is defined as the size of their intersection divided by the size of their union. For instance, the Jaccard similarity of S1 = {Ontario, Toronto} and S2 = {Alberta, Ontario, Manitoba} is 0.25.