CS 386/586 Winter 2012 Assignment 1

Assigned: Wednesday, January 9, 2013
Due: Wednesday, January 23, 2013 at midnight

You are strongly encouraged to finish your Assignment 1 by Wednesday, January 16, 2013 (one week after it is assigned). This schedule allows you to answer questions during class (on Wednesday, January 16) before it is due – the following Wednesday.

Submission: you **must** post a **note** to the "**instructors**" with an **attachment** containing your assignment as a pdf file in Piazza.
You **must** put your note with the attachment in the **turn-in-assign-here1** folder.

You can submit your assignment at any time. But, students are not allowed to delete questions or notes (even their own questions or notes) and they are not allowed to delete attachments. So, please submit your assignment when it's ready. (If you need to have us delete one of your questions or note, please notify the instructors – either through another post to instructors in piazza or by e-mail.

1. Demonstrate the difference between a primary key and a "unique" declaration for an attribute in a table in Postgresql with regard to how they handle null values.

   Your demonstration should show a table that you created with an attribute designated as a primary key where you are (or are not) able to use a null value for that attribute. Include screenshots to demonstrate your scenario and the success (or failure) to put a null value into a primary key.

   Your demonstration should show a table that you created with an attribute designated as unique where you are (or are not) able to use a null value for that attribute. Include screenshots as appropriate.

2. Demonstrate that you can (or cannot) have two primary keys for one table. Includes screenshots as appropriate.

3. Demonstrate that you can (or cannot) have two different attributes declared as unique for one table. Includes screenshots as appropriate.

4. Demonstrate that you can (or cannot) have one attribute as a primary key and another attribute as unique for one table. Includes screenshots as appropriate.

5. Write and execute an SQL statement that creates a table with a key that consists of two attributes. Show the SQL statement that you used. Show a screenshot that shows the result after you execute this SQL statement.

Using the Spy database available from the dbclass.cs.pdx.edu, write and execute queries in SQL to list the following information. Use only the "select", "from" and "where" clauses in

SQL (plus the union, intersect, or except) clause. (Do not use a JOIN clause. Students without prior experience with SQL will likely no know what the JOIN clause it. Don't worry; we'll talk about it next week.)

For each of these problems, show the SQL query, list the number of rows in the answer, and show the first 10 rows of the query answer. (You can copy and paste these into your assignment document or you can use screenshots.)

Also, write an equivalent query in relational algebra using only the select, project, cross product, union, intersection, and set difference operators. (You can use the words "select" or "project" or "cross" or "union" or "intersection" or "set-difference" if you don't want to type the Greek letters.

6. List the agent_id, agent first, middle, and last for agents with a salary greater than 52000.

7. List all attributes for agents with a first name of Jim who have a security clearance less than 5.

8. List all attributes for agents that do NOT appear in the answer to query 7 just above. (Hint: use the EXCEPT clause in SQL.)

9. List the two agent_ids, the two first and last names, and the security clearance for all pairs of agents where the two agents have the same first name, different last names, and the same security clearance.

   How can you check to make sure that the rows in the query answer meet the above criteria?

   How would you check (by issuing additional queries and examining the results) to see if there are any other agent pairs that meet the above criteria but did NOT appear in your query result?

10. List the mission name and the team name where the team is assigned to the mission. Hint: use a cross product and a select and project operator in relational algebra. (Do something similar in SQL.)

Other SQL queries:

11. Write a query against the Spy database that demonstrates that SQL does NOT eliminate duplicate rows from the query answer. Include screenshots that show this.

12. Write a similar query against the Spy database using the distinct clause that shows that the duplicate rows ARE eliminated.