San José State University
Department of Applied Data Science

DATA 225
Database Systems for Analytics

Fall 2023
Section 21
Instructor: Ron Mak

Assignment #6

Assigned: Monday, October 2
Due: Monday, October 9 at 5:30 pm
Team assignment, 100 points max

SQL command practice #3

The purpose of this assignment is to give your team more practice using SQL commands on the data that you've already chosen or new data. Create a Python notebook to do the following.

1. `INSERT INTO SELECT` with `CASE`

Create and load a database table. Then write one or more **`INSERT INTO`** command with an embedded **`SELECT`** to create one or more tables from the first table. Use **`CASE`** with the **`SELECT`** to perform data transformation(s). Display the contents of the first table and the newly created table(s).

You may want to use the newly created table(s) for the following exercises.

2. Aggregate functions with `GROUP BY`

Use one or more aggregate functions with **`GROUP BY`**. Explain in a sentence or two what the nested query is supposed to do and display the result.

3. Aggregate functions with `GROUP BY HAVING`

Use one or more aggregate functions with **`GROUP BY HAVING`**. Explain in a sentence or two what the nested query is supposed to do and display the result.

4. Nested query

Write a **`SELECT`** query with a nested **`SELECT`**. Explain in a sentence or two what the nested query is supposed to do and display the result.

5. Left outer join

Perform a left outer join between two tables and display the result.

6. Right outer join

Perform a right outer join between two tables and display the result.

7. Full outer join

Perform a full outer join between two tables and display the result.

8. View

Create a view and display its contents. Use the view in a join with other table(s) and display the result.

What to submit

Your Python notebook.

Rubric

Criteria	Max points
1. **`INSERT INTO SELECT`** with **`CASE`**	1. 20
2. Aggregate functions with **`GROUP BY`**	2. 10
3. Aggregate functions with **`GROUP BY HAVING`**	3. 10
4. Nested query	4. 10
5. Left outer join	5. 10
6. Right outer join	6. 10
7. Full outer join	7. 10
8. View	8. 20