**CSC453 Homework 2**

This HW is divided into 3 parts. Please download the schema and dataset provided for each part. Create the database on Oracle server using SQLDeveloper, and then write the queries for each part. Your queries must work for arbitrary data, not just the provided data.

**Submission instructions:** For each Part submit a Part-X.sql file and a Part-X.pdf file, where X = A, B, and C. Part-X.sql contains all your SQL queries. In Part-X.pdf copy/paste screenshots of each query and its answer from SQLDeveloper. Screenshots instructions are [here](http://facsrv.cs.depaul.edu/~tmalik1/teaching/csc453/Resources/Screenshot.pdf). In total you shall be submitting 6 files.

**A.**  **Basic SQL Operators (20 points)**

Dataset: Trips

The schema is as follows:

|  |  |  |
| --- | --- | --- |
| Trips | Trips | TID, TripState, TravelMode, Fare |
| ByCar | TID, Rental Company, Mileage |
| ByTrain | TID, Type, Coach, TrainSpeed, NumberofStops |
| ByPlane | TID, Airline, Class, LayoverTime |

The Trips table stores information about trips taken by various modes of travel and their respective fares. Each trip, owing to its mode has some relevant information specific to the mode. For instance, a car trip has the rental company and the mileage, where as train trip is characterized by the trainspeed, type, coach, and number of stops. The TravelCompany is interested in answering the following queries.

1. [1pt] List car rental companies which have a mileage of at least 27 miles/gallon.
2. [1pt] List trip IDs taken on train costing strictly more than $150.
3. [1pt] Find trip IDs and their fare that are not taken in the US i.e., `Non-US` trips.
4. [1pt] Find the business class plane trip IDs that are greater than $1000.
5. [2pt] Find any car trip more expensive than a trip taken on a train in the same state or outside the country.
6. [2pt] List pairs of distinct trips that have exactly the same value of mileage.

**Note a pair of distinct trips is of the format: (TID1, TID2).**

**This distinct pair is not the same as the pair (TID2, TID1)**

1. [2pt] List pairs of distinct train trips that do not have the same speed.
2. [2pt] Find those pair of trips in the same state with the same mode of travel. List such pairs only once. In other words, given a pair (TID1,TID2) do NOT list (TID2,TID1).
3. [4pt] Find a state in which trips have been taken by all three modes of transportation:  train, plane, and car.
4. [4pt] Find the details of a) the most costly trip, b) the cheapest trip, regardless of the travel mode. Write two separate queries for (a) and (b). **Write both (a) and (b) as a self-join with basic SQL operators (Filter, Project, Rename, Join (cross-join, natural join), Union, Intersect, and Difference). Do not use ALL, ANY, DISTINCT, GROUP BY, HAVING, MAX, MIN, ORDER BY.**

Submit A.sql consisting of all SQL statements, and A.pdf consisting of screenshots of queries and results. Not following screenshot instructions will result in loss of points.

**Part-B. Advanced SQL (15 points)**

Dataset: ComputerStore (Provided in D2L/Homeworks)

The schema is as follows:

|  |  |  |
| --- | --- | --- |
| Company | Employee | Fname, Minit, Lname, Ssn, Bdate, Address, Sex, Salary, Super\_Ssn, Dno |
| Department | Dname, Dnumber, Mgr\_ssn, Mgr\_start\_date |
| Project | Pname, Pnumber, Plocation, Dnum |
| Works\_On | Essn, Pno, Hours |
| Dependent | Essn, Dependent\_name, Sex, Bdate, Relationship |

The Employee table stores information about employee. The Department table stores information about a department, and Project about the project. Works\_On and Dependent are relationship tables.

Answer all 6 questions.

1. [2pt] Retrieve the names of all employees who work on at least one of the projects. (In other words, look at the list of projects given in the PROJECT table, and retrieve the names of all employees who work on at least one of them.)

2. [2pt] For each department, retrieve the department number, department name, and the average salary of all employees working in that department. Order the output by department number in ascending order.

3. [3pt] List the last names of all department managers who have no dependents.

4. [3pt] Determine the department that has the employee with the lowest salary among all employees. For this department retrieve the names of all employees. Write one query for this question.

5. [3pt] Find the total number of employees and the total number of dependents for every department (the number of dependents for a department is the sum of the number of dependents for each employee working for that department). Return the result as department name, total number of employees, and total number of dependents.

6. [2pt] Retrieve the names of employees whose salary is within $20,000 of the salary of the employee who is paid the most in the company (e.g., if the highest salary in the company is $80,000, retrieve the names of all employees that make at least $60,000).

Submit B.sql consisting of all SQL statements, and B.pdf consisting of screenshots of queries and results. Not following screenshot instructions will result in loss of points.

**Part-C. Analysis SQL (25 points)**

Dataset: Restaurants (Provided in D2L/Homeworks)

The schema is as follows:

|  |  |  |
| --- | --- | --- |
| Restaurants | restaurant | rID, name, address, cuisine |
| reviewer | vID, name |
| rating | vID, rID, stars, ratingdate |

1. [1pt] Find the name of all restaurants offering Indian cuisine
2. [2pt] Find restaurant names that received a rating of 4 or 5, sort them in increasing order.
3. [2pt] Find the names of all restaurants that have no rating.
4. [2pt] Some reviewers didn't provide a date with their rating. Find the names of all reviewers who have ratings with a NULL value for the date.
5. [4pt] For all cases where the same reviewer rated the same restaurant twice and gave it a higher rating the second time, return the reviewer's name and the name of the restaurant.
6. [4pt] For each restaurant that has at least one rating, find the highest number of stars that a restaurant received. Return the restaurant name and number of stars. Sort by restaurant name.
7. [5pt] For each restaurant, return the name and the 'rating spread', that is, the difference between highest and lowest ratings given to that restaurant. Sort by rating spread from highest to lowest, then by restaurant name.
8. [5pt] Find the difference between the average rating of Indian restaurants and the average rating of Chinese restaurants. (Make sure to calculate the average rating for each restaurant, then the average of those averages for Indian and Chinese restaurants. Don't just calculate the overall average rating for Indian and Chinese restaurants.) Note: The difference can be negative.

Submit C.sql consisting of all SQL statements, and C.pdf consisting of screenshots of queries and results. Not following screenshot instructions will result in loss of points.