**CSC453 Homework 2 Due October 7th, 2018 at 11.00pm**

This HW is divided into 3 parts.

**Part A and B**

Please download the schema and dataset provided with Part A and B. Create the database within your Oracle installation, 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. 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 4 files for Part A and B.

**Part-A. Basic SQL**

Dataset: ComputerStore (Provided in D2L/Homeworks)

The schema is as follows:

|  |  |  |
| --- | --- | --- |
| ComputerStore | manufacturers | code, name |
| products | code, name, price, manufacturer |

The manufacturers table stores information about computer parts manufacturers as the manufacturer code and manufacturer name. The products table stores information about product as a product code, product name, product’s price and the code of the manufacturer who manufactured the product.

Answer all 7 questions.

1. [1pt] Select the names of the products with a price less than or equal to $200

2. [1pt] Select all the products with a price between $60 and $120

3. [1pt] Select the name and price in cents (i.e., the price is in dollars).

4. [2pt] Select the product name, price, and manufacturer name of all the products.

5. [3pt] Select all manufactures who currently do not have any listed products.

6. [3pt] Select the name of each manufacturer along with the name and price of its most expensive product.

7. [4pt] Select the names and average prices of manufacturer whose products

have an average price larger than or equal to $150.

**Part-B. Analysis SQL**

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.)

**Part C: Functional Dependencies**

**C-1 Transitive Dependency and Keys**

You have a relation R(L,M,N,O,P,Q) R(A,B,C,D,E,F) and a set of functional dependencies F = {LNO→M, MN→LOP, N→O, OP→LN}.

* [2pt] Can we infer NP → LM from F ?
* [3pt] Can we infer NQ → LO from F ?

**C-2 Keys**

(i)[5pt]Find **all** the candidate keys of the Relation R(ABCDE) with FD's:

D → C, CE → A, D → A, and AE → D

(ii) [5pt] Determine **all** the candidate and superkeys of the relation R(ABCDEF) with FD's:

AEF → C, BF → C, EF → D, and ACDE → F

**C-3** **Minimal Cover**

[5pt] Find a minimal cover for the following set F of functional dependencies.

A→BC

AB→D

C→AD

D→B

Show your working clearly. Points will be deducted if you do not show the extraneous attributes, and their elimination.

**C-4** **Equivalence (15 points)**

[10pt] Consider the following set of F.Ds. Determine if FD1 is equivalent to FD2 or to FD3:

FD1:

{BC->D,ACD->B,CG->B,CG->D,AB->C,C->A,D->E,BE->C,D->G,CE->A,CE->G}

FD2:

{AB->C,C->A,BC->D,CD->B,D->E,D->G,BE->C,CG->D}

FD3:

{AB->C,C->A,D->G,BE->C,CG->D,CE->G,BC->D,CD->B,D->E}