COEN 178 Intro to Database Systems Spring 2022

Lab 2 (100 pts)

Objectives:

Learn writing SQL queries with

- Orderby

- aggregate functions

Requirements to complete the lab

1. Capture the session (queries and output) into a **commented** .log file for each part (only include correct runs!).
    - Comment before your queries! (ex: -- Part1 A or /* Part 1 A */) Either use "prompt" command or edit afterwards.
    - If you are running .sql files make sure you run "**SET ECHO ON**" so that the commands show as well as their output (You can see the status of the echo option by running "SHOW ECHO")
2. Submit your answers to questions, observations, and notes as .txt (or .doc/.docx/.pdf) file and upload to Camino.
3. Show the TA correct execution of the SQL programs and files

Oracle SQL Datatypes, Reference: http://www.ss64.com/orasyntax/datatypes.html

PART 1 (20 pts)

In this part, you will create a few tables and load them with given values and values of your choice.

Creating Tables

Assume that a customer can schedule a variety of home delivery services (grocery, movies, books etc).

Create the following tables based on the description given below **(primary keys are underlined**):

a) **Customer**: **custid** (a string of at most 5 characters), **firstname** (a string of at most 10 characters), **lastname** (a string of at most 15 characters), **city** (a string of at most 10 characters). Primary key: custid.

b) **DeliveryService**: **serviceid** (a string of at most 10 characters), **item** (a string of 15 characters), **location** (a string of at most 15 characters), **servicefee** (a number that has 5 digits with 2 after the decimal place). Primary key: serviced.

c) **Schedule**: **serviceid** (Foreign key referencing the table deliveryservice)**, custid** (Foreign key referencing the table customer)**, day** (a string of up to 2 char and cannot be null). We will define a rule/constraint for the day attribute, that the day has to be Mon('m') day through Fri ('f'). The rule is defined using a CHECK clause and **in** to indicate the set of values that are allowed. Use **'m','t','w','r','f'** to represent the day values.

See the example table below and use the same approach to define the rule for day in Schedule table.

Example table (this is an example to show the usage of CHECK. Do not create this table):

```
Create table Trip (id varchar(5) PRIMARY KEY,
source varchar(15), dest varchar(15)
CHECK (dest in ('NY','Paris','London')));
```

Inserting Tuples

Add the following tuples into the tables (I would recommend using a script file to add the data).

Customer:

'c1','John','Smith','SJ'

'c2', 'Mary', 'Jones','SFO'

'a1', 'Vincent','Chen','SJ';

'a12', 'Greg', 'King','SJ';

'c7', 'James','Bond','LA';

'x10', 'Susan','Blogg','SFO';

Add a few more tuples of your choice.

DeliveryService:

'dsg1','grocery','SJ',25.0

'dsb1','books','SJ',10.0

'dsm2','movies','LA',10.0

'dby3','babygoods','SFO',15.0

'dsg2','grocery','SFO',20.0

'dg5','greengoods','SFO',30.0

Add a few more tuples of your choice.

Schedule

'dsg1','c1','m'

'dsg1','a12','w'

'dby3','x10','f'

'dg5','c1','r'

'dg5','c1','t'

'dg5','c32','t'

Q1a) **Did you successfully insert all the above tuples? If not, explain the reason for error. (5 pts)**

Now, try to insert the following tuple into Schedule table.

'dsg2','c1','s'

Q1b) **Did you successfully insert the above tuple? If not, explain the reason for error. (5 pts)**

Add a few more tuples of your choice.

Part 2 (80 pts)

Write the SQL for the following queries (5 pts each) ***(You are free to refer to the class notes and examples posted).***

a) Show all the data in the **Customer** table

b) Show all the data in the **DeliveryService** table

c) Show all the data in the **Schedule** table

Show the data in the Customer table as follows:

d) custid, fullname, city, where fullname is the result of concatenating First name and last (‘**||**’ is the concatenation operator). Rename the column name as the fullname.

e**) Show data in the Customer, sorted by Customer last name (use *order by*).**

f) **Show data in the Schedule table, sorted by service id and then by customer id in descending order (use *order by*).**

g) **Show service ids of delivery services that are not in the schedule table (think set difference).**

h) The following query is given to show the names of customers who ordered a delivery service on Monday (‘M’). Will it work? If not, fix it and show the query and results.

Select firstname from customer, schedule where day = 'M';

i) **Show the last names of the customers that are scheduled delivery services. (What tables is the data coming from?)**

j) **Show the highest servicefee in Delivery Services (think of the aggregate function, max) renaming the result as highest_Servicefee.**

k) Show the number of delivery services scheduled by day (think of aggregate function, **count()** and **group by (day).**

The incomplete query below is given to show pairs of customer ids from the same city. Complete it.

```
Select A.custid,B.custid,A.city
from Customer A, Customer B
where A.city = B.city;
```

l) Write a query to show the customers (who scheduled delivery services) where the customer city and location of the delivery services are in the same city.

Do the following query against the staff table that you have created and loaded in lab1.

m) Write a query to show the minimum salary and maximum salary of staff members in the **staff** table.