COMP 3311 Assignment 3

(Oracle Databases and SQL Queries)

Spring 2020

Due date: 26th Mar 2020

(11:59 am according to Canvas clock HK time)

Please check the time and date carefully !

Assignment Rules:

1) This is an individual assignment. The assignment solution you submit must be solely your own work; copying or letting others to copy are both considered cheating.

2) **No late submission will be accepted !**

Assignment description

There are two parts in this assignment: part 1 and part 2.

Part 1. Creating tables using the SQL Data Definition Language (40%)

Create 8 tables using Oracle SQL Developer.

Use exactly the same table names and attribute names as listed below, otherwise marks will be deducted.

- The table order below is not necessarily the correct order for creating the tables; it is merely describing the column types and some constraints of the tables.
- Constraints are given except referential integrity constraints.
- You should add referential integrity constraints (if any) to the tables.
- If your scripts return error(s) or incorrect output, you will get zero mark for the corresponding tables.

- **product table**
  1. product_ID: varchar2(8), primary key.
  2. product_name: varchar2(80). NOT NULL
  3. price: number(3).
  4. product_date: number(8).

- **manu table**
  1. manu_ID: number(8), primary key.
  2. manu_first_name: varchar2(80).
  3. manu_last_name: varchar2(80).

- **manu_address table**
  1. manu_ID: number(8).
  2. address: varchar2(80).
  3. manu_phone: number(8).

  manu_ID foreign key referencing the Manu() table, primary key (manu_ID, manu_phone).

- **prerequisite table**
  1. main_product_ID: varchar2(8).
  2. prereq_product_ID: varchar2(8).

  main_product_ID foreign key referencing the product_ID column of the product() table, prereq_product_ID foreign key referencing the product_ID column of the product() table, primary key (main_product_ID, prereq_product_ID).

- manu_product table

1. manu_ID: number(8).
2. product_ID: varchar2(8).
3. offering_no: number(8).

manu_ID foreign key referencing the Manu() table,
(product_ID, offering_no) foreign key referencing the offering() table,
primary key (manu_ID, product_ID, offering_no).

- **retailor table**

1. retailor_ID: number(8), primary key.
2. retailor_first_name: varchar2 (80).
3. retailor_last_name: varchar2 (80).
4. region_code: number(3)
5. phone: number(8).
6. product_ID: varchar2(8), NOTNULL.
7. offering_no: number(8), NOTNULL.

(product_ID, offering_no) foreign key referencing the offering table.

- **manu_retailor table**

1. manu_ID: number(8).
2. retailor_ID: number(8).

manu_ID foreign key referencing the manu() table,
retailor_ID foreign key referencing the retailor() table,
primary key (manu_ID, retailor_ID).

- **offering table**

1. product_ID: varchar2(8).
2. offering_no: number(8).
3. store: number(5).
4. no_of_purchase: number(5).

product_ID foreign key referencing the product() table,
primary key (product_ID, offering_no).

Part 2. Write SQL Queries (60%)

Write the following SQL queries. If your scripts return error(s) or incorrect output, you will get zero mark for corresponding queries. You can write SQL scripts to insert some artificial data into tables for your own testing. The insert scripts are not graded and are not required to submit.

1) Find the product_ID for products manufactured before date "20200202" with the highest price.

2) Find the retailor_ID, retailor_fisrt_name, retailor_last_name of the retailors who have product ID "12345678" and region code "852".

3) Find the manu_ID, last_name, first_name of all manufacturers who has the offerings with store more than 10000 of "12345678".

4) Find the manu_ID, last_name, first_name of all the manufacturers who have NOT offered any of the prerequisites of "22345678".

5) Find the manu_ID, manu_first_name of all the manufacturers whose last name is "James" and have produced 'Model_A1' but not 'Model_B1'. ***(In this place, 'Model_A1' and 'Model_B1' are the sample product names.)***

Submission:

1. Write SQL statements and put them into two **.sql** files:

   - **create.sql** for creating the tables given in part 1,

   - **query.sql** for the queries in part 2.

2. You need to submit the **two .sql** files to Canvas. Please check out detailed instructions in the following link:

https://canvas.ust.hk/courses/29896/assignments/103485

3. Your submitted files will be tested directly on Oracle SQL Developer for grading.