Name   CWID

Homework Assignment 2

Due Date: October 12, 2019 (11:59PM CDT)

CS425 - Database Organization

Please leave this empty!

2.1		2.2	

2.3		2.4		2.5		2.6		2.7		2.8		2.9		2.10	

2.11		2.12		2.13		2.14		Sum	

Instructions

- Try to answer all the questions using what you have learned in class

- **When writing a query, write the query in a way that it would work over all possible database instances and not just for the given example instance!**

- After creating the relations, you can make up some data and insert them into the relations

- Please submit the homework (as a single sql file **your_name.sql**) electronically using blackboard

- In the **your_name.sql** file, make sure you have marked the answer for each question using comments

- All the sql scripts should be **executable** in Oracle 11g (e.g., on fourier) or recent versions

Consider the following database schema and example instance:

Employee

eid	ename	dob	salary
0011	Alice	07/03/1976	140,000
0012	Bob	09/14/1978	110,000
0013	Chris	06/22/1991	80,000
0014	David	05/17/1982	105,000
0015	Edward	04/15/1988	98,000
...	...	...	...

Customer

cid	cname	city	state
B101	Citi Bank	New York	NY
B102	JP Morgan	New York	NY
B103	Fidelity	Boston	MA
B104	Moto	Chicago	IL
B105	Disney	Orlando	FL
...	...	...	...

Product

pid	pname	unit_price
A17811	Desk	750
A17812	Chair	230
A23453	PC	900
A34451	Mac	300
...	...	...

Sales

eid	cid	pid	quantity
0011	B101	A17811	700
0011	B102	A17812	200
0012	B102	A23453	100
0013	B101	A34451	50
...	...	...	...

Manages

eid1	eid2
0011	0013
0013	0017
0012	0013
...	...

Hints:

- Underlined attribute(s) form the primary key of a relation.

- Relation *Sales* stores the sales information: the salesman (employee ID) who is in charge of the ordered product, the product ID, the customer ID, and the quantities of ordered products.

- The attributes *eid*, *cid* and *pid* of relation *Sales* are the foreign keys to relations *Employee* (salesman), *Customer* and *Product*, respectively.

- Relation *Manages* stores the "managing" information in the company (e.g., the employee with *eid1* is the direct manager of the employee with *eid2*).

- All the IDs' are strings.

Part 2.1 SQL DDL (Total: 20 Points)

Question 2.1.1   (15 Points)

Write the SQL statements for all the five relations: *Employee*, *Customer*, *Product*, *Sales*, and *Manages*. Note: (1) An appropriate data type should be used for each attribute, and (2) All the foreign keys should be created.

Question 2.1.2   (5 Points)

Write an SQL statement that adds a constraint to the *Employee* relation to make sure that the *salary* attribute cannot be NULL, and the value of this attribute has to be between 40,000 and 500,000. Furthermore, the default value for this attribute should be 80,000.

Part 2.2 SQL Queries (Total: 56 Points)

Question 2.2.1   (7 Points)

Write an SQL query that returns the products (pid and pname) which are sold by the direct manager(s) of employee '0013' and the quantity for any customer (not aggregated) is greater than 300. Hint: relation *Manages* stores the "managing" information in the company (e.g., the employee with *eid1* is the direct manager of the employee with *eid2*).

Question 2.2.2   (7 Points)

Write an SQL query that returns the IDs of all the products, with the total ordered quantity in 'Chicago' greater than the total ordered quantity in 'Boston'.

Question 2.2.3   (7 Points)

Write an SQL query that returns the name and unit price of all the products (pname and unit_price) which are never ordered by 'Disney'.

Question 2.2.4 (7 Points)

Write an SQL query that returns all the employees and their average sales amount (of all the customers) for each of the products (query result: eid, pid and avg_sales).
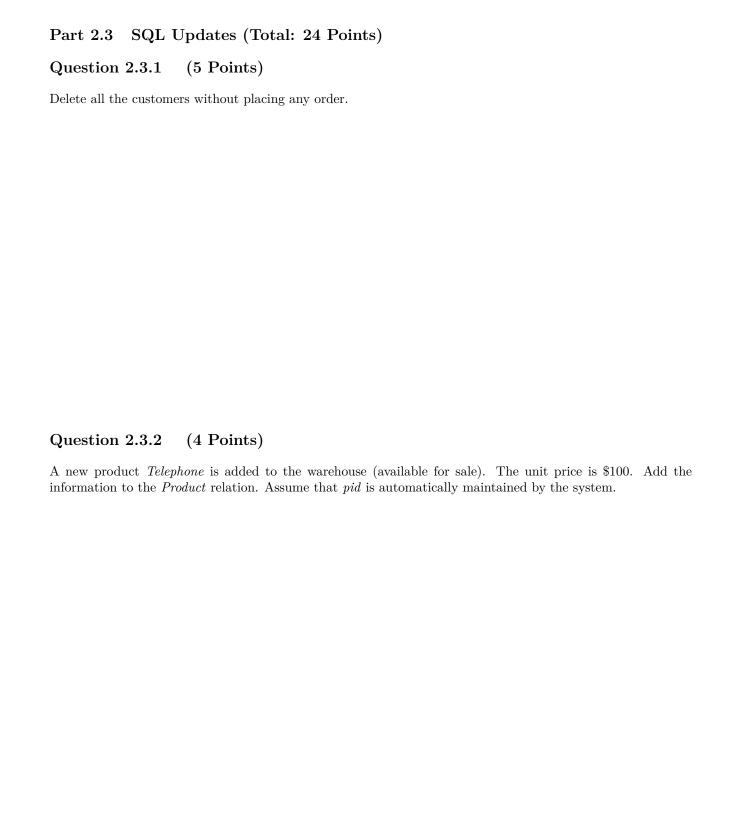
Question 2.2.5   (7 Points)

Write an SQL query that returns all the employees and the total number of employees directly managed by each employee (if no one is directly managed, then return 0). Hint: relation *Manages* stores the "managing" information in the company (e.g., the employee with *eid1* is the direct manager of the employee with *eid2*).

Question 2.2.6   (7 Points)

Write an SQL query that returns the number of products where the average ordered quantity (of all the employees and customers) is lower than 1000.

Question 2.2.7 (7 Points)

Write an SQL query that returns the product names and the name of the customer(s) who ordered the highest quantity (from all the employees). Query result: pname and cname.

Question 2.2.8   (7 Points)

Write an SQL query which returns all the employees' IDs and their managed employees' IDs. Hint: the result should include not only directly managed employees but also indirectly managed employees.

Part 2.3   SQL Updates (Total: 24 Points)

Question 2.3.1   (5 Points)

Delete all the customers without placing any order.

Question 2.3.2   (4 Points)

A new product *Telephone* is added to the warehouse (available for sale). The unit price is $100. Add the information to the *Product* relation. Assume that *pid* is automatically maintained by the system.

Question 2.3.3   (8 Points)

Update the *unit_price* in the *Product* relation according to this rule:

- if it is negative, set it to 0

- if it is larger than 10,000, then set it to 2,500

- if it is `NULL`, set it to 1,000

- if none of the above applies do not change the *unit_price*

Note that we expect you to write a single statement that implements this.

Question 2.3.4   (7 Points)

Update the salaries of employees as his/her current salary $+ 0.02 \cdot$ (his/her total sales amount).