CSIT 5300 - Spring 2017 - HKUST
Assignment 2

Deadline: Wednesday, March 08, 2017, 10:20pm
Submission: Please bring a hard copy of your solution during the lecture

<table>
<thead>
<tr>
<th>Student Name</th>
<th>ANSWER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Student ID</td>
<td></td>
</tr>
<tr>
<td>Email Address</td>
<td></td>
</tr>
<tr>
<td>Section</td>
<td></td>
</tr>
</tbody>
</table>

For T.A. use only

<table>
<thead>
<tr>
<th>Problem</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>/ 25</td>
</tr>
<tr>
<td>2</td>
<td>/ 25</td>
</tr>
<tr>
<td>3</td>
<td>/ 25</td>
</tr>
<tr>
<td>4</td>
<td>/ 25</td>
</tr>
<tr>
<td>Total</td>
<td>/ 100</td>
</tr>
</tbody>
</table>
A Simple e-Shop Database

Consider an e-Shop, which allows merchants to sell their products to customers via its website, potentially offering discounts. You are required to design a simple database for this e-Shop, which has the following specifications:

- A user in the system is either a merchant, or a customer, but not both.
- For every customer, the database records a unique id, name, city, credit card number, and date of birth.
- For each merchant, the database records a unique id, name, city, account number, and a phone number list with arbitrary size.
- The e-Shop can distribute only up to 10 different products from each merchant.
- The same product cannot be sold by two different merchants via the e-Shop.
- Each product has a unique id, a description, and a price.
- A merchant may offer a discount for a product, which lasts for a single day. Each discount is associated with exactly one product, whereas many discounts can be associated with the same product. Each discount has an id, a discount percentage, and the validity date.
- A customer can purchase an arbitrary number of products through the e-Shop. The database records the date of each purchase.

We also make the following assumptions:

- Every company in the database sells at least one product.
- Every product must be sold by a company.
- Every customer in the database has purchased at least one product.
- A customer does not purchase more than one product on the same date.
- For simplicity, assume that the date is only comprised of month and day (the year is omitted)
Problem 1 [25 points] ER-Diagram

Create the ER-diagram for the e-Shop database specified above.

Answer:
Problem 2 [25 points] Relational Tables

Construct the relational tables for the ER-diagram you created in Problem 1, underlining the primary keys.

Answer:

(*The foreign keys appear in *italics*)

Customer(\textit{uid}, name, city, credit\_card, birth\_date)
Merchant(\textit{uid}, name, city, account)
Merchant-Phone(\textit{uid}, phone)
Discount(\textit{pid}, did, percentage, validity\_date)
Purchases(\textit{uid}, \textit{pid}, date)
Product(\textit{pid}, description, price, \textit{uid})
Problem 3 [25 points] Relational Algebra

Give an expression in relational algebra to formulate each of the following queries:

(a) [5 points] Find the ids of the products whose price is greater than 100.

Answer:

$$\pi_{\text{pid}}(\sigma_{\text{price}>100}(\text{Product}))$$

(b) [5 points] Find the description and original price (i.e., ignoring any potential discounts) of the products purchased by customer with the id “stavros”.

Answer:

$$\pi_{\text{price}, \text{description}}(\sigma_{\text{Purchases.uid} = \text{"stavros"}}(\text{Purchases} \bowtie \text{pid Product}))$$

(c) [5 points] Find the ids of the customers who did not purchase a product (at any date) that had a discount on their birthday.

Answer:

Ids of all customers:

$$\rho(A, \pi_{\text{uid}}(\text{Customer}))$$

Ids of customers that did purchase a product (at any date) with a discount on their birth-date:

$$\rho(B, \pi_{\text{Customer.uid}}(\sigma_{\text{validity.date}=\text{birth.date}}(\text{Customer} \bowtie \text{uid Purchases} \bowtie \text{pid Discount})))$$

Final result: $A - B$
(d) [5 points] Find the ids of the customers who purchased all the products of the merchant with the id “autotech”.

Answer:

$$\pi_{uid,pid}(\text{Purchases})/\pi_{pid}(\sigma_{uid=\text{autotech}}(\text{Product}))$$

(e) [5 points] Find the id of the most expensive product sold by the merchant with the id “autotech”.

Answer:

All product ids of merchant “autotech”:

$$\rho(\text{A}, \pi_{\text{pid}}(\sigma_{uid=\text{autotech}}(\text{Product})))$$

Two copies of all “autotech” products:

$$\rho(\text{P1}, \sigma_{uid=\text{autotech}}(\text{Product}))$$
$$\rho(\text{P2}, \sigma_{uid=\text{autotech}}(\text{Product}))$$

Ids of “autotech” products whose price is smaller than the price of some other “autotech” product:

$$\rho(\text{B}, \pi_{\text{P1.pid}}(\sigma_{\text{p1.price}<\text{p2.price}}(\text{P1} \times \text{P2})))$$

Final result: A − B
Problem 4 [25 points] SQL

Give an expression in SQL to formulate each of the following queries:

(a) [5 points] Find the ids of the products whose price is greater than 100.

Answer:

```
select pid
from Product
where price > 100
```

(b) [5 points] Find the names of the merchants who offered a discount on “01/01”.

Answer:

```
select distinct name
from Merchant as M, Product as P, Discount as D
where M.uid = P.uid and P.pid = D.pid and validity_date = "01/01"
```

(c) [5 points] For each merchant, return the id and total amount of money that its products cost (disregarding discounts), under the condition that the total amount is greater than 1000. The result should be ordered in ascending order of the total amount.

Answer:

```
select uid, sum(price) as total
from Product
group by uid
having total > 1000
order by total asc
```

(d) [5 points] Find the ids of the customers who have not purchased any product with a discount (i.e., none of the products they have bought had a discount on the date of purchase).

Answer:

```
select C.uid
from Customer as C
where not exists ( select R.uid
from Purchases as R, Discount as D
where C.uid = R.uid and R.pid = D.pid and R.date = D.validity_date )
```
(e) [5 points] Find the id of the merchant with the largest sum of product prices.

Answer:

```sql
select uid
from ( select uid, sum(price) as total
        from Product
        group by uid ) as T
where total = ( select max(total)
                from T )
```