COSC265 Lab Test 2018

Time allowed: 90 min
Total marks: 100
Worth: 20%
Open book

There are 4 questions. It is highly recommended that you attempt question 4 only after you have finished all other questions.

Your answers should be clear and concise - material not relevant to the question will earn no marks whether or not it is correct.

- **Create a text file named with your usercode (e.g. *abc12.txt*), and include in it, by writing directly and/or copy and paste from your SQL environment:**
  - **all SQL statements that you execute to answer the questions, and**
  - **any query results or data showing the correctness of your SQL statements.**

- **Remember to keep saving your work as you go, as it will be impossible to retrieve your work from Oracle once you have logged out**

- **Make sure that you label each SQL statement submitted with the question number and letter (e.g. 1a).**

- **Include enough detail in your answer to show that what you have done is correct.**

- **You must submit your answer text file via Learn.**

- **Make sure that you submit all the work that you've done**

A bank database contains information necessary for the day-to-day operation of the bank and all of its branches.

An Oracle script bank-script.sql is available on your system, located in the Labtest2018 folder in your home directory, which contains statements to create and populate two tables for this database.

- The CUST relation stores the following information for each customer: the customer number (unique), first and last name, and a four-digit PIN code set by the customer.
- The ACCOUNT relation contains information about accounts within the bank. Each account has a unique number. There are only two types of accounts: savings or cheque. The account relation also stores the branch code, customer's number, the date of opening the account and the status of the account. An account is frozen if the balance is zero. When an account is closed, the date is stored in the same table. There is only one customer per account.

The SQL create table statements for these two relations are:

```
create table CUST
(Cust_No char(9) not null,
FName varchar(15) not null,
LName varchar(15) not null,
PIN char(4) not null constraint check_PIN check (PIN between '0000' and '9999'),
constraint pk_cust primary key (Cust_No));

create table ACCOUNT
(Acc_No integer not null check (ACC_No>0),
Type varchar(7) not null check (Type in ('savings','cheque')),
Branch varchar(7) not null,
Customer char(9) not null references CUST,
Open_Date date not null,
Status varchar(7) not null check (Status in 'active','closed','frozen')),
Closed_Date date,
constraint pk_account primary key (ACC_No),
constraint check_dates check (Closed_Date > Open_Date));
```

Question 1 (45 marks for the whole question)

a) (0 marks) Once you log on to Oracle, execute the command *set autocommit on;* to ensure that each command you execute is immediately applied to the database.

b) (0 marks) Execute the script *bank-script.sql*, which will create the two tables as above and populate those two tables with a number of rows.

c) (10 marks) Write an SQL statement to list names and numbers of customers whose accounts in the Central branch have been closed during last thirty (30) months.

d) (10 marks) Write an SQL statement to show how many accounts of each type (active, frozen, and closed) there are for each branch. Display the branches alphabetically.

e) (8 marks) Write an SQL statement to identify customers who have no accounts.

f) (7 marks) For any customers that have no accounts in question 1e) above, write an SQL statement to add a savings account in the Central branch for that customer, opened today.

g) (10 marks) Write an SQL statement to list the names of customers who have accounts in more than one branch. Show the total number of accounts per customer, as well as the number of branches the customer has accounts in.

Question 2 (15 marks for the whole question)

a) (10 marks) Write an SQL statement, and execute it, to create the TRANSACTION relation with the following schema:

TRANSACTION (Trans_No, Account, Amount, Date, Type, Location)

The following information is known for every transaction: unique number, the date, the account the transaction is related to, amount, the type of the transaction ('d' for deposit and 'w' for withdrawal), and location, which is the branch where the transaction was performed.

b) (5 marks) Populate the TRANSACTION relation you created with data provided in the *transaction-populate.sql* file. Write an SQL statement you can use to find the number of tuples in your table, and include this statement's result.

Question 3 (35 marks for the whole question)

a) (12 marks) The Location attribute of the Transaction table should only contain valid branches. However, it is not possible to specify this constraint in the CREATE TABLE statement. Why is that so? Write a trigger to enforce this constraint, and demonstrate that it works correctly.

b) (8 marks) Create a view that will show the total amount of all transactions performed in a branch on each day.

c) (10 marks) Use that view to find out the branch with the maximum total transaction amount on the 11th of March,.2018.

d) (5 marks) The bank decides that an account must remain open for at least 10 days before it can be closed. Modify the ACCOUNT table to reflect these requirements.

Question 4 (5 marks for the whole question)

a) (5 marks) Write an SQL statement to find each customer who has accounts with all status values.