LAB REPORT

Lab 3
Data Definition and Data Manipulation

CSE 4308
DATABASE MANAGEMENT SYSTEMS LAB

NAME: CHOWDHURY ASHFAQ
STUDENT ID: 200042123
PROGRAM: SWE
GROUP: 1A
DATE: 11/09/22

Tasks:

3 Lab Task

You have to write all SQL statements in an editor first and save them with .sql extention. Then execute the SQL script.

1. Write SQL statements to create the following tables with the given specifications:

   (a) ACCOUNT

ACCOUNT_NO	CHAR(5)	(e.g.: A-101) Primary Key
BALANCE	NUMBER	Not Null

   (b) CUSTOMER

CUSTOMER_NO	CHAR(5)	(e.g.: C-101) Primary Key
CUSTOMER_NAME	VARCHAR2(20)	Not Null
CUSTOMER_CITY	VARCHAR2(10)	(e.g.: DHK, KHL, etc.)

   (c) DEPOSITOR

ACCOUNT_NO	CHAR(5)	(e.g.: A-101)
CUSTOMER_NO	CHAR(5)	(e.g.: C-101)
		Primary Key(ACCOUNT_NO, CUSTOMER_NO)

2. Write SQL statements to perform the following alteration operations:
   (a) Add a new attribute 'DATE_OF_BIRTH' (DATE type) in CUSTOMER table.
   (b) Modify the data type of BALANCE from NUMBER to NUMBER(12, 2).
   (c) Rename the attribute ACCOUNT_NO, CUSTOMER_NO from DEPOSITOR table to A_NO and C_NO, respectively.
   (d) Rename the table DEPOSITOR to DEPOSITOR_INFO.
   (e) Add two foreign key constraints FK_DEPOSITOR_ACCOUNT and FK_DEPOSITOR_CUSTOMER that identifies A_NO and C_NO as foreign keys.
3. Write SQL statements to answer the following queries:
   (a) Find all account number with balance less than 100000.
   (b) Find all customer names who live in 'KHL' city.
   (c) Find all customer number whose name contains 'A'.
   (d) Find distinct account numbers from DEPOSITOR_INFO table.
   (e) Show the result of Cartesian Product between ACCOUNT and DEPOSITOR_INFO table.
   (f) Show the result of Natural Join between CUSTOMER and DEPOSITOR_INFO table.
   (g) Find all customer names and their city who have an account.
   (h) Find all customer related information who have balance greater than 1000.
   (i) Find all accounts related information where balance is in between 5000 and 10000 or their depositor lives in 'DHK' city.

3

In this Lab Task we were given to perform various DDL (Data Definition Language) and DML (Data Modification Language) operations.

Analysis of the problem:

At first 5 subtasks where given which needed to be performed through DDL. These 5 subtasks include renaming a table and attribute, adding a attribute to a table, modifying the data type of an attribute and adding foreign key constraints to a table.

Next there was 9 tasks which where to be done by DML. The 9 tasks included selection of various attributes from different tables.

Solution:

```
CREATE TABLE ACCOUNT
(
    ACCOUNT_NO char(5),
    BALANCE number NOT NULL,
    CONSTRAINT PK_ACCOUNT_NO PRIMARY KEY (ACCOUNT_NO)
);

CREATE TABLE CUSTOMER
(
    CUSTOMER_NO char(5),
    CUSTOMER_NAME varchar2(20) NOT NULL,
    CUSTOMER_CITY varchar2(10),
    CONSTRAINT PK_CUSTOMER_NO PRIMARY KEY (CUSTOMER_NO)
);

CREATE TABLE DEPOSITOR
(
    ACCOUNT_NO char(5),
    CUSTOMER_NO char(5),
    CONSTRAINT PK_ACC_CUSTOMER PRIMARY KEY (ACCOUNT_NO,CUSTOMER_NO)
);


ALTER TABLE CUSTOMER ADD DATE_OF_BIRTH DATE;

ALTER TABLE ACCOUNT MODIFY BALANCE number(12,2);

ALTER TABLE DEPOSITOR RENAME COLUMN ACCOUNT_NO TO A_NO;
ALTER TABLE DEPOSITOR RENAME COLUMN CUSTOMER_NO TO C_NO;

ALTER TABLE DEPOSITOR RENAME TO DEPOSITOR_INFO;

ALTER TABLE DEPOSITOR_INFO ADD CONSTRAINT FK_DEPOSITOR_ACCOUNT FOREIGN KEY (A_NO) REFERENCES ACCOUNT(ACCOUNT_NO);
ALTER TABLE DEPOSITOR_INFO ADD CONSTRAINT FK_DEPOSITOR_CUSTOMER FOREIGN KEY (C_NO) REFERENCES CUSTOMER(CUSTOMER_NO);
```

```sql
INSERT INTO ACCOUNT VALUES('12345',70000);
INSERT INTO ACCOUNT VALUES('23456',120000);
INSERT INTO ACCOUNT VALUES('34567',100000);
INSERT INTO ACCOUNT VALUES('14567',7000);


INSERT INTO CUSTOMER VALUES('10000','ASHFAQ','KHL','30-JAN-2000');
INSERT INTO CUSTOMER VALUES('10001','SHOYEB','DHK','30-AUG-2001');
INSERT INTO CUSTOMER VALUES('20001','SHUVRO','CTG','09-JAN-1999');

INSERT INTO DEPOSITOR_INFO VALUES('23456','10000');
INSERT INTO DEPOSITOR_INFO VALUES('34567','20001');
INSERT INTO DEPOSITOR_INFO VALUES('12345','10001');
INSERT INTO DEPOSITOR_INFO VALUES('14567','10000');




SELECT ACCOUNT_NO
FROM ACCOUNT
WHERE BALANCE<100000;


SELECT CUSTOMER_NAME
 FROM CUSTOMER
WHERE CUSTOMER_CITY='KHL';


SELECT CUSTOMER_NO
 FROM CUSTOMER
WHERE (CUSTOMER_NAME like '%A%') or (CUSTOMER_NAME like '%a%');


SELECT DISTINCT A_NO
 FROM DEPOSITOR_INFO;
```

```sql
SELECT *
FROM ACCOUNT,DEPOSITOR_INFO;


SELECT *
FROM CUSTOMER NATURAL JOIN DEPOSITOR_INFO;


SELECT CUSTOMER_NAME, CUSTOMER_CITY
FROM CUSTOMER, DEPOSITOR_INFO
WHERE CUSTOMER.CUSTOMER_NO=DEPOSITOR_INFO.C_NO;


SELECT CUSTOMER_NO,CUSTOMER_NAME,CUSTOMER_CITY,DATE_OF_BIRTH
FROM CUSTOMER, ACCOUNT, DEPOSITOR_INFO
WHERE ACCOUNT.BALANCE>1000 AND DEPOSITOR_INFO.C_NO=CUSTOMER.CUSTOMER_NO AND ACCOUNT.ACCOUNT_NO=DEPOSITOR_INFO.A_NO;


SELECT ACCOUNT_NO,BALANCE
FROM CUSTOMER, ACCOUNT, DEPOSITOR_INFO
WHERE (DEPOSITOR_INFO.C_NO=CUSTOMER.CUSTOMER_NO AND ACCOUNT.ACCOUNT_NO=DEPOSITOR_INFO.A_NO)
 AND ((ACCOUNT.BALANCE>5000 AND ACCOUNT.BALANCE<10000) OR CUSTOMER.CUSTOMER_CITY='DHK') ;
```

OUTPUT:

```
ACCOU
-----
12345
14567


CUSTOMER_NAME
--------------------
ASHFAQ


CUSTO
-----
10000


A_NO
-----
23456
34567
12345
14567


ACCOU    BALANCE A_NO  C_NO
----- ---------- ----- -----
12345      70000 23456 10000
12345      70000 34567 20001
12345      70000 12345 10001
12345      70000 14567 10000
23456     120000 23456 10000
23456     120000 34567 20001
23456     120000 12345 10001
23456     120000 14567 10000
34567     100000 23456 10000
34567     100000 34567 20001
34567     100000 12345 10001

ACCOU    BALANCE A_NO  C_NO
----- ---------- ----- -----
34567     100000 14567 10000
14567       7000 23456 10000
14567       7000 34567 20001
14567       7000 12345 10001
14567       7000 14567 10000

16 rows selected.
```

```
CUSTO CUSTOMER_NAME        CUSTOMER_C DATE_OF_B A_NO  C_NO
----- -------------------- ---------- --------- ----- -----
10000 ASHFAQ               KHL        30-JAN-00 23456 10000
10000 ASHFAQ               KHL        30-JAN-00 34567 20001
10000 ASHFAQ               KHL        30-JAN-00 12345 10001
10000 ASHFAQ               KHL        30-JAN-00 14567 10000
10001 SHOYEB               DHK        30-AUG-01 23456 10000
10001 SHOYEB               DHK        30-AUG-01 34567 20001
10001 SHOYEB               DHK        30-AUG-01 12345 10001
10001 SHOYEB               DHK        30-AUG-01 14567 10000
20001 SHUVRO               CTG        09-JAN-99 23456 10000
20001 SHUVRO               CTG        09-JAN-99 34567 20001
20001 SHUVRO               CTG        09-JAN-99 12345 10001

CUSTO CUSTOMER_NAME        CUSTOMER_C DATE_OF_B A_NO  C_NO
----- -------------------- ---------- --------- ----- -----
20001 SHUVRO               CTG        09-JAN-99 14567 10000
```

```
CUSTOMER_NAME        CUSTOMER_C
-------------------- ----------
ASHFAQ               KHL
SHOYEB               DHK
SHUVRO               CTG

CUSTO CUSTOMER_NAME        CUSTOMER_C DATE_OF_B
----- -------------------- ---------- ---------
10001 SHOYEB               DHK        30-AUG-01
10000 ASHFAQ               KHL        30-JAN-00
20001 SHUVRO               CTG        09-JAN-99
```

```
ACCOU    BALANCE
----- ----------
12345      70000
14567       7000
```

Explanation:

- At first we create a new user named as ‘ash20042123’ with password ‘cse4308’. The command for the action is :

```
CREATE USER ash200042123 IDENTIFIED BY cse4308;
```

- Next we grant all privileges to the new user through the command:

```
GRANT ALL PRIVILEGES TO ash200042123;
```

- Then we are to create 3 tables named as ACCOUNT, CUSTOMER and DEPOSITOR.

ACCOUNT table has two columns ACCOUNT_NO and BALANCE whose domain are char(5) and number respectively. We also give a constraint PRIMARY KEY which is ACCOUNT_NO.

CUSTOMER table has three columns CUSTOMER_NO, CUSTOMER_NAME and CUSTOMER_CITY whose domain are char(5), varchar2(20) and varchar2(10) respectively. We also give a constraint PRIMARY KEY which is CUSTOMER_NO.

Next the DEPOSITOR table has ACCOUNT_NO and CUSTOMER_NO of domain char(5). Both of them are selected together as PRIMARY KEY.

```
CREATE TABLE ACCOUNT
(
    ACCOUNT_NO char(5),
    BALANCE number NOT NULL,
    CONSTRAINT PK_ACCOUNT_NO PRIMARY KEY (ACCOUNT_NO)
);

CREATE TABLE CUSTOMER
(
    CUSTOMER_NO char(5),
    CUSTOMER_NAME varchar2(20) NOT NULL,
    CUSTOMER_CITY varchar2(10),
    CONSTRAINT PK_CUSTOMER_NO PRIMARY KEY (CUSTOMER_NO)
);

CREATE TABLE DEPOSITOR
(
    ACCOUNT_NO char(5),
    CUSTOMER_NO char(5),
    CONSTRAINT PK_ACC_CUSTOMER PRIMARY KEY (ACCOUNT_NO,CUSTOMER_NO)
);
```

- Next we have to perform some DDL operations.

At first, we add a column DATE_OF_BIRTH to CUSTOMER table for which ALTER TABLE is used and with the help of ADD keyword.

Secondly, we modify the domain of BALANCE attribute of ACCOUNT table to number(12,2) from number using MODIFY keyword.

Thirdly, we rename the column name of ACCOUNT_NO and CUSTOMER_NO of DEPOSITOR table to A_NO and C_NO respectively with RENAME keyword.

After that we convert the name of DEPOSITOR table to DEPOSITOR_INFO with RENAME keyword.

Lastly, we are to add to FOREIGN KEY CONSTRAINTS to the table DEPOSITOR_INFO which are named as FK_DEPOSITOR_ACCOUNT and FK_DEPOSITOR_CUSTOMER and applied to A_NO and C_NO referencing ACCOUNT_NO of ACCOUNT table and CUSTOMER_NO of CUSTOMER table respectively.

```
ALTER TABLE CUSTOMER ADD DATE_OF_BIRTH DATE;

ALTER TABLE ACCOUNT MODIFY BALANCE number(12,2);

ALTER TABLE DEPOSITOR RENAME COLUMN ACCOUNT_NO TO A_NO;
ALTER TABLE DEPOSITOR RENAME COLUMN CUSTOMER_NO TO C_NO;

ALTER TABLE DEPOSITOR RENAME TO DEPOSITOR_INFO;

ALTER TABLE DEPOSITOR_INFO ADD CONSTRAINT FK_DEPOSITOR_ACCOUNT FOREIGN KEY (A_NO) REFERENCES ACCOUNT(ACCOUNT_NO);
ALTER TABLE DEPOSITOR_INFO ADD CONSTRAINT FK_DEPOSITOR_CUSTOMER FOREIGN KEY (C_NO) REFERENCES CUSTOMER(CUSTOMER_NO);
```

- Next we insert some values to the 3 tables to perform the DML operations later on.

```
INSERT INTO ACCOUNT VALUES('12345',70000);
INSERT INTO ACCOUNT VALUES('23456',120000);
INSERT INTO ACCOUNT VALUES('34567',100000);
INSERT INTO ACCOUNT VALUES('14567',7000);


INSERT INTO CUSTOMER VALUES('10000','ASHFAQ','KHL','30-JAN-2000');
INSERT INTO CUSTOMER VALUES('10001','SHOYEB','DHK','30-AUG-2001');
INSERT INTO CUSTOMER VALUES('20001','SHUVRO','CTG','09-JAN-1999');

INSERT INTO DEPOSITOR_INFO VALUES('23456','10000');
INSERT INTO DEPOSITOR_INFO VALUES('34567','20001');
INSERT INTO DEPOSITOR_INFO VALUES('12345','10001');
INSERT INTO DEPOSITOR_INFO VALUES('14567','10000');
```

- We then perform the DML operations which are as follows:

1. We select the ACCOUNT_NO of those who have BALANCE more than 1,00,000 from ACCOUNT table with SELECT, FROM, WHERE command.

2. In the same way as above we select CUSTOMER_NAME who live in ‘KHL’ city from CUSTOMER table.

3. Again from CUSTOMER table we select the customer number of those person who has ‘A’ in their name. The ‘like’ operator is a string matching operator and by ‘%A%’ we denote at any part of the string if A is present.

4. Then we select all the different A_NO from DEPOSITOR_INFO table. To select different A_NO we use DISTINCT keyword.

5. Then we perform a Cartesian Product between ACCOUNT and DEPOSITOR_INFO table. To do it we simply select all the elements from ACCOUNT, DEPOSITOR_INFO.

6. Then we perform NATURAL JOIN between the CUSTOMER table and DEPOSITOR_INFO table which is done with the NATURAL JOIN keyword.

```
SELECT ACCOUNT_NO
FROM ACCOUNT
WHERE BALANCE<100000;


SELECT CUSTOMER_NAME
 FROM CUSTOMER
WHERE CUSTOMER_CITY='KHL';


SELECT CUSTOMER_NO
 FROM CUSTOMER
WHERE (CUSTOMER_NAME like '%A%') or (CUSTOMER_NAME like '%a%');


SELECT DISTINCT A_NO
 FROM DEPOSITOR_INFO;


SELECT *
FROM ACCOUNT,DEPOSITOR_INFO;


SELECT *
FROM CUSTOMER NATURAL JOIN DEPOSITOR_INFO;
```

7. Then we find all the customer names and their city who have an account. Those who have an account will have an entry in DEPOSITOR_INFO table. So, we perform Cartesian Product

between CUSTOMER, DEPOSITOR_INFO table and take only the meaningful values. From there we select the customer name and customer city.

8. After that we find all customer related information who have balance greater than 1000. First we do Cartesian Product among the three tables and then we take only the meaningful tuples and those tuples where account balance is more than 1000. From there we select only the customer number, customer name, customer city and date of birth. As a customer may have two or more accounts so we select only the distinct customer numbers.

9. Next we select all account related information where balance is in between 5000 and 10000 or their depositor lives in 'DHK' city. For that at first we take only the meaningful tuples from the Cartesian product of the three tables. From there, we only take those tuples whose account balance is more than 5000 and less than 10000 or those customers who live in Dhaka city. We then select account number and balance from the result we've got.

```
SELECT DISTINCT CUSTOMER_NAME, CUSTOMER_CITY
FROM CUSTOMER, DEPOSITOR_INFO
WHERE CUSTOMER.CUSTOMER_NO=DEPOSITOR_INFO.C_NO;


SELECT DISTINCT CUSTOMER_NO,CUSTOMER_NAME,CUSTOMER_CITY,DATE_OF_BIRTH
FROM CUSTOMER, ACCOUNT, DEPOSITOR_INFO
WHERE ACCOUNT.BALANCE>1000 AND DEPOSITOR_INFO.C_NO=CUSTOMER.CUSTOMER_NO AND ACCOUNT.ACCOUNT_NO=DEPOSITOR_INFO.A_NO;


SELECT ACCOUNT_NO,BALANCE
FROM CUSTOMER, ACCOUNT, DEPOSITOR_INFO
WHERE (DEPOSITOR_INFO.C_NO=CUSTOMER.CUSTOMER_NO AND ACCOUNT.ACCOUNT_NO=DEPOSITOR_INFO.A_NO)
 AND ((ACCOUNT.BALANCE>5000 AND ACCOUNT.BALANCE<10000) OR CUSTOMER.CUSTOMER_CITY='DHK') ;
```

Interesting Findings:

- Learn about the problem of NATURAL JOIN.

Problems faced and solution:

- Didn’t know the format of DATE type in SQL. Had to google it.
- Faced difficulty to add constraint to a table through command prompt. Took help from YouTube to solve the problem.