



DBMS ASSIGNMENT FOR TASK 3

NAME: NUSRAT SIDDIQUE TULI

DEPT: CSE

PROGRAM:SWE

ID: 210042131

COURSE: DAT

COURSE: DATABASE MANAGEMENT SYSTEM

I am writing a report explaining task 03, my solution approach, also the SQL command, snippet of my solution and explaining the problems.

1.

I have to write SQL statements to create the following tables with the given specifications.

To create a table in SQL, use the following syntax:

```
CREATE TABLE table_name (
    column1 datatype,
    column2 datatype,
    column3 datatype,
    ...
);
```

For example, to create a table named "employees" with columns for name, age, and job title, you could use the following SQL statement:

```
CREATE TABLE employees (
    name VARCHAR(50),
    age INT,
    job_title VARCHAR(50)
);
```

SQL code for creating account , customer and depositor table is-

(a)

```
CREATE TABLE ACCOUNT
(
    ACCOUNT_NO VARCHAR2(5),
    BALANCE NUMBER NOT NULL,
    CONSTRAINT ACCOUNT_TABLE_PK PRIMARY KEY (ACCOUNT_NO)
);
```

(b)

```
CREATE TABLE CUSTOMER
(
    CUSTOMER_NO VARCHAR2(5),
    CUSTOMER_NAME VARCHAR(20) NOT NULL,
    CUSTOMER_CITY VARCHAR(10),
    CONSTRAINT CUSTOMER_TABLE_PK PRIMARY KEY (CUSTOMER_NO)
);
```

(c)

```
CREATE TABLE DEPOSITOR
(
    ACCOUNT_NO VARCHAR2(5),
    CUSTOMER_NO VARCHAR2(5),
    CONSTRAINT DEPOSITOR_TABLE_PK PRIMARY KEY (ACCOUNT_NO,CUSTOMER_NO)
);
```

2. I have to write the following SQL commands-

(a) Add a new attribute 'DATE_OF_BIRTH' (DATE type) in CUSTOMER table.

```
ALTER TABLE CUSTOMER ADD DATE_OF_BIRTH DATE;
```

(b) Modify the data type of BALANCE from NUMBER to NUMBER(12, 2).

```
ALTER TABLE ACCOUNT MODIFY BALANCE NUMBER (12 , 1);
```

(c) Rename the attribute ACCOUNT_NO, CUSTOMER_NO from DEPOSITOR table to A_NO and C_NO, respectively.

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

(d) Rename the table DEPOSITOR to DEPOSITOR_INFO.

```
ALTER TABLE DEPOSITOR RENAME 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.

```
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);
```

3.

now, I have to Insert 3 records in each table following the example.

SQL for inserting values is-

```
INSERT INTO ACCOUNT VALUES(101,10000);  
INSERT INTO ACCOUNT VALUES(102,20000);  
INSERT INTO ACCOUNT VALUES(103,30000);  
  
INSERT INTO CUSTOMER VALUES(1, 'ABC', 'DHK', '20-JAN-22');  
INSERT INTO CUSTOMER VALUES(2, 'EFG', 'BAR', '20-JAN-22');  
INSERT INTO CUSTOMER VALUES(3, 'HIJ', 'KHL', '20-JAN-22');  
  
INSERT INTO DEPOSITOR_INFO VALUES(101,1);  
INSERT INTO DEPOSITOR_INFO VALUES(102,2);  
INSERT INTO DEPOSITOR_INFO VALUES(103,3);
```

4 .

SQL statements to answer the following queries:

(a) Find all account numbers with balances less than 100000.

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

(b) Find all customer names who live in 'KHL' city.

```
SELECT * FROM CUSTOMER WHERE CUSTOMER_CITY='KHL' ;
```

(c) Show the result of Cartesian Product between ACCOUNT and DEPOSITOR_INFO table.

```
SELECT * FROM ACCOUNT, DEPOSITOR_INFO;
```

(d) Show the result of Natural Join between CUSTOMER and DEPOSITOR_INFO table.

```
SELECT * FROM CUSTOMER NATURAL JOIN DEPOSITOR_INFO ;
```

(e) Find all customer names and their city who have an account.

```
SELECT CUSTOMER_NAME, CUSTOMER_CITY FROM CUSTOMER, ACCOUNT, DEPOSITOR_INFO WHERE ACCOUNT.ACCOUNT_NO=DEPOSITOR_INFO.A_NO AND CUSTOMER.CUSTOMER_NO=DEPOSITOR_INFO.C_NO ;
```

(f) Find all customer-related information who have balance greater than 1000.

```
SELECT * FROM CUSTOMER,ACCOUNT,DEPOSITOR_INFO WHERE ACCOUNT.ACCOUNT_NO=DEPOSITOR_INFO.A_NO AND CUSTOMER.CUSTOMER_NO=DEPOSITOR_INFO.C_NO AND ACCOUNT.BALANCE>1000;
```

- (g) Find all accounts-related information where balance is in between 5000 and 10000 or
their depositor lives in 'DHK' city.

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

5.

DML statements:

- (a) Update the CUSTOMER_CITY 'KHL' as 'KLN'.

```
UPDATE CUSTOMER SET CUSTOMER_CITY ='KHN' WHERE CUSTOMER_CITY ='KHL';
```

- (b) Update the CUSTOMER_ID 101 as 301.

```
UPDATE CUSTOMER SET CUSTOMER_NO =301 WHERE CUSTOMER_NO =101;
```

- (c) Delete Customer with CUSTOMER_ID 301.

```
DELETE FROM CUSTOMER WHERE CUSTOMER_NO =301;
```

- (d) Delete all the information without deleting the table structure.

```
DELETE FROM ACCOUNT;  
DELETE FROM CUSTOMER;  
DELETE FROM DEPOSITOR_INFO;
```