

CSE 4308
Database Management Systems Lab

Lab-05

Group-B

Mohammad Mahbub Ur Rahman
220042148
BSc. in Software Engineering

Task 1

Creating tables and inserting values

```
-- creating table
CREATE TABLE CUSTOMER (
    CUSTOMER_NO CHAR (5) PRIMARY KEY ,
    CUSTOMER_NAME VARCHAR (20) NOT NULL ,
    CUSTOMER_CITY VARCHAR (10)
) ;
```

```
CREATE TABLE ACCOUNT (
    ACCOUNT_NO CHAR (5) PRIMARY KEY ,
    BALANCE DECIMAL (12 ,2) NOT NULL
) ;
```

```
CREATE TABLE DEPOSITOR (
    CUSTOMER_NO CHAR (5) ,
    ACCOUNT_NO CHAR (5) ,
    PRIMARY KEY ( CUSTOMER_NO , ACCOUNT_NO )
) ;
```

```
INSERT INTO CUSTOMER ( CUSTOMER_NO , CUSTOMER_NAME ,  
CUSTOMER_CITY ) VALUES  
("C-101", "John Doe", "DHK") ;  
INSERT INTO CUSTOMER ( CUSTOMER_NO , CUSTOMER_NAME ,  
CUSTOMER_CITY ) VALUES  
("C-102 ", "Jane Smith", "KHL") ;  
INSERT INTO CUSTOMER ( CUSTOMER_NO , CUSTOMER_NAME ,  
CUSTOMER_CITY ) VALUES  
("C-103", "Alice Brown", "CTG") ;  
INSERT INTO CUSTOMER ( CUSTOMER_NO , CUSTOMER_NAME ,  
CUSTOMER_CITY ) VALUES  
("C-104", "Bob Johnson", "DHK") ;
```

```
INSERT INTO ACCOUNT ( ACCOUNT_NO , BALANCE ) VALUES  
("A-101", 5000.00) ;  
INSERT INTO ACCOUNT ( ACCOUNT_NO , BALANCE ) VALUES  
("A-102", 15000.00) ;  
INSERT INTO ACCOUNT ( ACCOUNT_NO , BALANCE ) VALUES  
("A-103", 25000.00) ;  
INSERT INTO ACCOUNT ( ACCOUNT_NO , BALANCE ) VALUES  
("A-104", 8000.00) ;
```

```
INSERT INTO DEPOSITOR ( CUSTOMER_NO , ACCOUNT_NO ) VALUES  
("C-101", "A-101"),  
("C-102", "A-102"),  
("C-103", "A-103"),  
("C-104", "A-104") ;
```

1.

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

Adding DATE_OF_BIRTH column in customer table.

2.

```
RENAME TABLE DEPOSITOR TO DEPOSITOR_INFO;
```

Renaming table DEPOSITOR to DEPOSITOR_INFO

3.

```
ALTER TABLE DEPOSITOR_INFO  
CHANGE COLUMN ACCOUNT_NO A_NO CHAR (5) ,  
CHANGE COLUMN CUSTOMER_NO C_NO CHAR (5) ;
```

Changing the name of the columns in
DEPOSITOR_INFO TABLE.

4.

```
ALTER TABLE DEPOSITOR_INFO ADD CONSTRAINT  
FK_DEPOSITOR_ACCOUNT  
FOREIGN KEY (A_NO) REFERENCES ACCOUNT(ACCOUNT_NO)  
ON DELETE CASCADE  
ON UPDATE CASCADE;
```

```
ALTER TABLE DEPOSITOR_INFO ADD CONSTRAINT  
FK_DEPOSITOR_CUSTOMER  
FOREIGN KEY (C_NO) REFERENCES  
CUSTOMER(CUSTOMER_NO)  
ON DELETE CASCADE  
ON UPDATE CASCADE;
```

Adding foreign key constraint in the DEPOSITOR_INFO table. Adding on delete cascade and on update cascade ensures that the update in this table also sync across other table (Account).

5.

```
SELECT *  
FROM CUSTOMER;
```

Select all records from customer.

6.

```
SELECT C.CUSTOMER_NAME, A.BALANCE
FROM CUSTOMER C
INNER JOIN DEPOSITOR_INFO D ON C.CUSTOMER_NO =
D.C_NO
INNER JOIN ACCOUNT A ON D.A_NO = A.ACCOUNT_NO;
```

Selecting name and balance of customer by inner joining customer, depositor, account table. Here inner join is achieved by comparing column that have same value.

OUTPUT:

CUSTOMER_NAME	BALANCE
John Doe	7500.00
Jane Smith	15000.00
Alice Brown	25000.00

7.

```
SELECT AVG(balance)
FROM ACCOUNT;
```

Getting average balance from account table.

Output:

AVG(balance)
15833.333333

8.

```
SELECT CUSTOMER_NAME, A_NO  
FROM CUSTOMER  
NATURAL JOIN DEPOSITOR_INFO;
```

Selecting name and account number from customer and account table by natural joining.

CUSTOMER_NAME	A_NO	
Alice Brown	A-101	
Jane Smith	A-101	
John Doe	A-101	
Alice Brown	A-102	
Jane Smith	A-102	
John Doe	A-102	
Alice Brown	A-103	
Jane Smith	A-103	
John Doe	A-103	

9.

```
UPDATE CUSTOMER  
SET CUSTOMER_CITY = "KLN"  
WHERE CUSTOMER_NO = "C-102";
```

Updating the customer city where customer no is C-102.

10.

```
DELETE FROM ACCOUNT  
WHERE ACCOUNT_NO = "A-104";
```

Deleting the account entry whose account_no is A-104

11.

```
DELETE FROM CUSTOMER  
WHERE CUSTOMER_NO NOT IN (  
SELECT C_NO FROM DEPOSITOR_INFO);
```

Deleting the data from customer table who is not in the depositor_info table.

Task 2

1.

```
CREATE TABLE LOAN(  
    LOAN_NO CHAR(5) PRIMARY KEY,  
    AMOUNT DECIMAL(12, 2) NOT NULL  
);
```

```
INSERT INTO LOAN (LOAN_NO, AMOUNT) VALUES  
("L-201", 10000.00),  
("L-202", 20000.00),  
("L-203", 15000.00);
```

Creating loan table and inserting data.

2.

```
CREATE TABLE BORROWER (  
    CUSTOMER_NO CHAR(5),  
    LOAN_NO CHAR(5),  
    PRIMARY KEY (CUSTOMER_NO, LOAN_NO),  
    FOREIGN KEY (CUSTOMER_NO) REFERENCES  
CUSTOMER(CUSTOMER_NO),  
    FOREIGN KEY (LOAN_NO) REFERENCES LOAN(LOAN_NO)  
    ON DELETE CASCADE  
    ON UPDATE CASCADE  
);
```

```
INSERT INTO BORROWER (CUSTOMER_NO, LOAN_NO)
VALUES
("C-101", "L-201"),
("C-102", "L-202"),
("C-103", "L-203");
```

Creating borrower table, adding primary key and foreign key. Then adding data

3.

```
SELECT *
FROM CUSTOMER C
INNER JOIN BORROWER B ON C.CUSTOMER_NO =
B.CUSTOMER_NO;
```

Inner joining borrower and customer to find out who have both account and loan.

CUSTOMER_NO	CUSTOMER_NAME	CUSTOMER_CITY	DATE_OF_BIRTH	CUSTOMER_NO	LOAN_NO
C-101	John Doe	DHK	NULL	C-101	L-201
C-102	Jane Smith	KLN	NULL	C-102	L-202
C-103	Alice Brown	CTG	NULL	C-103	L-203

4.

```
SELECT *
FROM CUSTOMER
WHERE CUSTOMER_NO NOT IN
(SELECT CUSTOMER_NO FROM BORROWER);
```

Selecting customer who is not in borrower table/who doesn't have any loan.

5.

```
UPDATE ACCOUNT  
SET BALANCE = BALANCE + BALANCE*0.5  
WHERE BALANCE < 10000;
```

Updating account table and making balance = balance*1.5 where balance < 10000.