# CSE 4308

## LAB 3 ASSIGNMENT

Task statement:

Our main goal for this task was to create three table with several attributes and complete various operations using SQL. The main focus of this task was to learn about primary keys, foreign keys and the relationship between them. We also had to alter and update the table.

Solution approach:

For the solution, we first had to create three separate tables. We also had to create primary keys accordingly. Afterwards in task 2, we had to add several attributes, rename some and create some foreign keys. The method of creating a foreign key was that we had to add it as a constraint, give the key a name and reference the table it’s a primary key of. The foreign key can also be a composite key of several other keys. In order to alter a primary key, we had to drop the previous primary key all together and start as a different one. Task 3 was relatively easier, as we only had to insert some records into the table. For task 4, we had to make some selection queries. A new topic of cartesian products was introduced here. We had to use queries such as cross join, natural join to merge tables in meaningful ways and perform queries on them. The final task was to update and delete some attributes of the table. We had to update the data and at the end, delete some data from the tables.

A snippet of the solution is provided below:

*CREATE TABLE ACCOUNT(*

*ACCOUNT\_NO CHAR(5),*

*BALANCE NUMBER NOT NULL,*

*CONSTRAINT AC PRIMARY KEY(ACCOUNT\_NO)*

*);*

*CREATE TABLE CUSTOMER(*

*CUSTOMER\_NO CHAR(5),*

*CUSTOMER\_NAME VARCHAR2(20) NOT NULL,*

*CUSTOMER\_BRANCH VARCHAR2(10),*

*CONSTRAINT C\_PK PRIMARY KEY(CUSTOMER\_NO)*

*);*

*CREATE TABLE DEPOSITOR(*

*ACCOUNT\_NO CHAR(5),*

*CUSTOMER\_NO CHAR(5),*

*CONSTRAINT ALL\_P PRIMARY KEY(ACCOUNT\_NO, CUSTOMER\_NO)*

*);*

*ALTER TABLE CUSTOMER ADD ROUTING\_NO NUMBER;*

*ALTER TABLE DEPOSITOR ADD ROUTING\_NO NUMBER;*

*ALTER TABLE DEPOSITOR ADD CUSTOMER\_BRANCH VARCHAR2(10);*

*ALTER TABLE DEPOSITOR ADD B\_NO NUMBER;*

*ALTER TABLE DEPOSITOR RENAME COLUMN ACCOUNT\_NO TO A\_NO ;*

*ALTER TABLE DEPOSITOR RENAME COLUMN CUSTOMER\_NO TO C\_NO ;*

*ALTER TABLE CUSTOMER DROP CONSTRAINT C\_PK;*

*ALTER TABLE CUSTOMER ADD PRIMARY KEY (CUSTOMER\_NO, ROUTING\_NO, CUSTOMER\_BRANCH,B\_NO);*

*ALTER TABLE DEPOSITOR ADD CONSTRAINT FK\_DEPOSITOR\_CUSTOMER FOREIGN KEY(C\_NO,CUSTOMER\_BRANCH, ROUTING\_NO) REFERENCES CUSTOMER(CUSTOMER\_NO,CUSTOMER\_BRANCH,ROUTING\_NO);*

*ALTER TABLE DEPOSITOR ADD CONSTRAINT FK\_DEPOSITOR\_ACCOUNT FOREIGN KEY(A\_NO,B\_NO) REFERENCES ACCOUNT(ACCOUNT\_NO,BALANCE);*

*INSERT INTO ACCOUNT (ACCOUNT\_NO, BALANCE) VALUES ('1',1);*

*INSERT INTO ACCOUNT (ACCOUNT\_NO, BALANCE) VALUES ('2',2);*

*INSERT INTO ACCOUNT (ACCOUNT\_NO, BALANCE) VALUES ('3',3);*

*INSERT INTO ACCOUNT (ACCOUNT\_NO, BALANCE) VALUES ('4566',10100000);*

*INSERT INTO CUSTOMER (CUSTOMER\_NO, CUSTOMER\_NAME, CUSTOMER\_BRANCH, ROUTING\_NO) VALUES ('1','1','1',1);*

*INSERT INTO CUSTOMER (CUSTOMER\_NO, CUSTOMER\_NAME, CUSTOMER\_BRANCH, ROUTING\_NO) VALUES ('2','2','2',2);*

*INSERT INTO CUSTOMER (CUSTOMER\_NO, CUSTOMER\_NAME, CUSTOMER\_BRANCH, ROUTING\_NO) VALUES ('3','3','3',3);*

*INSERT INTO CUSTOMER (CUSTOMER\_NO, CUSTOMER\_NAME, CUSTOMER\_BRANCH, ROUTING\_NO) VALUES ('5','MOUSHI','DHK',503);*

*INSERT INTO DEPOSITOR (A\_NO, C\_NO, ROUTING\_NO, CUSTOMER\_BRANCH) VALUES ('1','1',1,'1');*

*INSERT INTO DEPOSITOR (A\_NO, C\_NO, ROUTING\_NO, CUSTOMER\_BRANCH) VALUES ('2','2',2,'2');*

*INSERT INTO DEPOSITOR (A\_NO, C\_NO, ROUTING\_NO, CUSTOMER\_BRANCH) VALUES ('3','3',3,'3');*

*SELECT ACCOUNT\_NO FROM ACCOUNT WHERE BALANCE>=100000;*

*SELECT CUSTOMER\_NAME FROM CUSTOMER WHERE CUSTOMER\_BRANCH='DHK';*

*SELECT \* FROM CUSTOMER CROSS JOIN DEPOSITOR;*

*SELECT \* FROM CUSTOMER NATURAL JOIN DEPOSITOR;*

*SELECT A\_NO, B\_NO FROM DEPOSITOR WHERE C\_NO='Mr.X';*

*SELECT A\_NO,B\_NO FROM DEPOSITOR NATURAL JOIN ACCOUNT WHERE ACCOUNT.BALANCE>=5000 AND ACCOUNT.BALANCE<=10000 AND DEPOSITOR.C\_NO='Mr.X';*

*SELECT CUSTOMER\_NAME, CUSTOMER\_BRANCH FROM DEPOSITOR NATURAL JOIN CUSTOMER WHERE DEPOSITOR.A\_NO IS NOT NULL;*

*INSERT INTO CUSTOMER (CUSTOMER\_NO, CUSTOMER\_NAME, CUSTOMER\_BRANCH, ROUTING\_NO) VALUES ('C-101','IM','KHL',78);*

*UPDATE CUSTOMER SET CUSTOMER\_BRANCH = 'KLN' WHERE CUSTOMER\_BRANCH='KHL';*

*UPDATE CUSTOMER SET CUSTOMER\_NO = 'C-301' WHERE CUSTOMER\_NO='C-101';*

*DELETE FROM CUSTOMER WHERE CUSTOMER\_NO = ’C-301 ’;*

*DELETE FROM DEPOSITOR;*

*DELETE FROM CUSTOMER;*

*DELETE FROM ACCOUNT;*

The commands that were used are:

ALTER:

The ALTER statement in SQL is used to modify an existing database object's structure, such as a table. It allows us to add, modify, or delete columns in an existing table, among other things. For example, we can use ALTER TABLE to add a new column to an existing table.

ADD PRIMARY KEY CONSTRAINT:

Adding a primary key constraint to a table ensures that each row in the table has a unique identifier, and it enforces the uniqueness of values in the specified column(s). Primary keys are used to uniquely identify rows within a table. We can use the ALTER TABLE statement to add a primary key constraint to an existing table.

CROSS JOIN:

A CROSS JOIN in SQL is used to combine all rows from two or more tables, resulting in a Cartesian product. It essentially combines every row from one table with every row from another table, creating a new table with all possible combinations. It can generate a large number of rows.

NATURAL JOIN:

A NATURAL JOIN is a type of JOIN operation in SQL that combines two tables based on columns with matching names. It automatically matches columns with the same name in both tables, and it's often used when tables have related data with identical column names. However, it's important to be aware that using NATURAL JOIN can be risky if columns are added or renamed, as it relies solely on column names.

UPDATE:

The UPDATE statement in SQL is used to modify existing records in a table. We can specify which rows to update using a WHERE clause along with the UPDATE statement. This allows us to change the values of one or more columns in selected rows of the table.

DELETE:

The DELETE statement in SQL is used to remove one or more rows from a table. Like the UPDATE statement, we can use a WHERE clause with the DELETE statement to specify which rows should be deleted. It can delete all rows in the table.

Problems I faced:

The main problem I faced was that I had trouble using primary keys and connecting the tables. I also had trouble using the natural join and cross join commands and accurately showing the information between different tables. Other than that, the task went smoothly enough.

SUBMITTED BY

MEHREEN HOSSAIN CHOWDHURY

210041219