CSE 4308: Database Management Systems Lab

Lab-02

Group-B

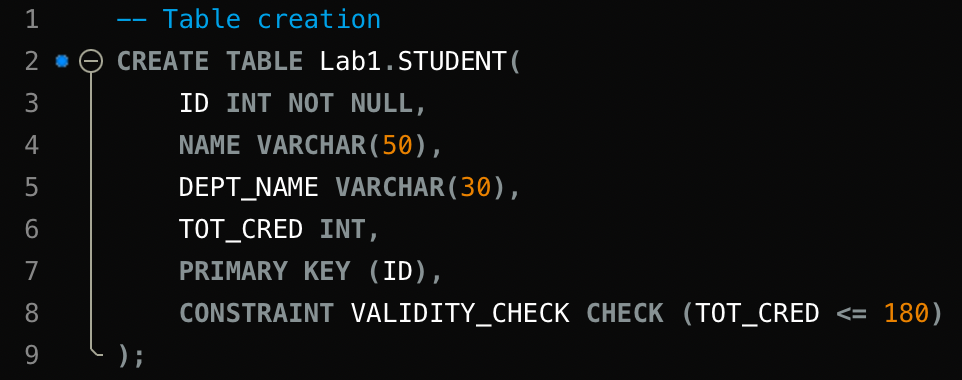
Mohammod Mahbub Ur Rahman

220042148

BSc.in Software Engineering

**Task-1**

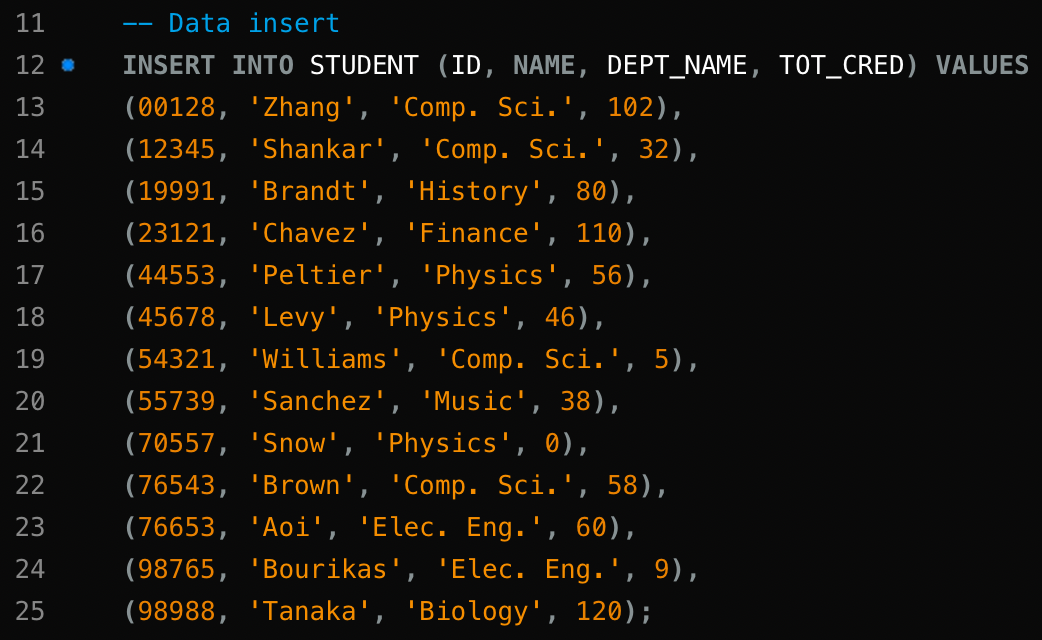
**1.**



creating the table using the ‘CREATE TABLE’.

Adding column names and their data type. Also declaring ID as not null. Setting ID as primary key and checking if an entry has credit greater than or = 180.

2.inserting data in the following format.



3.

a> “SELECT \* FROM STUDENT;”

it selects all the entry of the students in the student table and returns us all the entries with data.

b> “SELECT ID, NAME FROM STUDENT;”

it shows ID and name of the entries of the student table.

c> “SELECT NAME, DEPT\_NAME FROM STUDENT

WHERE (TOT\_CRED >=80 AND TOT\_CRED <= 120);”

It shows us name and department name from the student table where the entry’s credit is between 80 and 120.

d> “SELECT ID, NAME FROM STUDENT WHERE DEPT\_NAME = 'Comp. Sci.'”;

It shows us the ID and name from the student table which meet the condition where its department name is 'Comp. Sci.

e> “SELECT NAME, TOT\_CRED FROM STUDENT WHERE DEPT\_NAME = 'Physics';”

Same as the previous

f> “SELECT ID, NAME FROM STUDENT WHERE DEPT\_NAME = 'Comp. Sci.' OR TOT\_CRED < 10;”

Same as before but added an extra constraint where total credit is less than 10. Now it will show us the id and the name from the table where he is in 'Comp. Sci. or his credit is less than 10.

g> “SELECT DISTINCT DEPT\_NAME FROM STUDENT;”

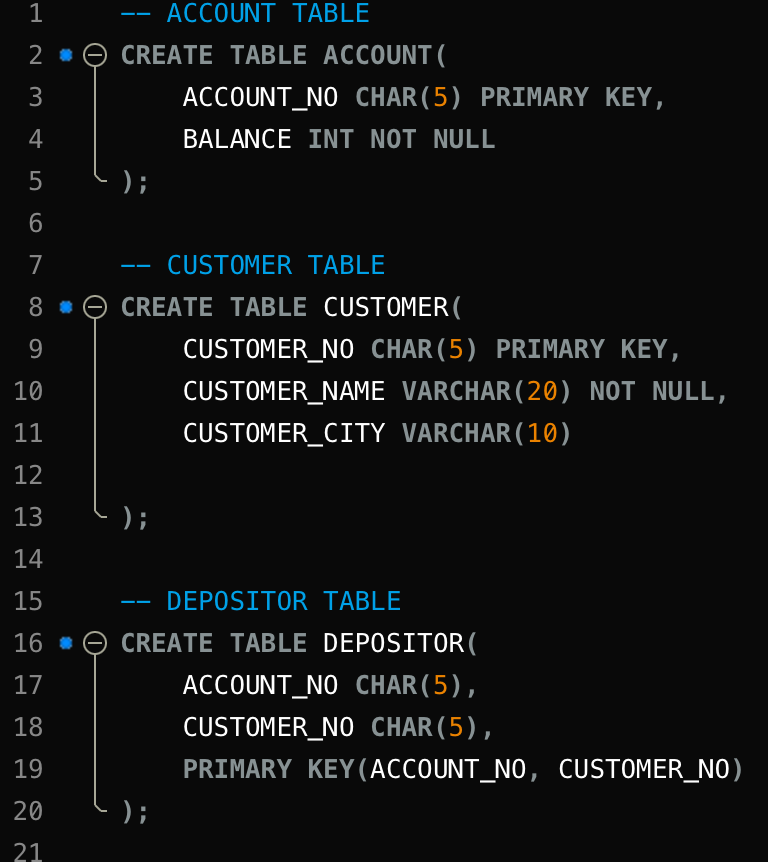
it gives us the distinct department name from the table. It will ignore duplicate data.

h> “DROP TABLE STUDENT CASCADE;”

it deletes the table with all its data. Cascade keyword deletes the data with constrains.

**Task-2**

**1**



Created 3 tables and selected primary key.

2.

a> “ALTER TABLE CUSTOMER ADD DATE\_OF\_BIRTH DATE;”

added date of birth column in the customer table.

b> “ALTER TABLE ACCOUNT MODIFY BALANCE DECIMAL (12, 2);”

modified the balance data type from int to decimal where it will be 12 digits and 2 digits after decimal point.

c> “ALTER TABLE DEPOSITOR RENAME COLUMN ACCOUNT\_NO TO A\_NO;

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

Renamed the columns to A\_NO and C\_NO

d> “ALTER TABLE DEPOSITOR RENAME TO DEPOSITOR\_INFO;”

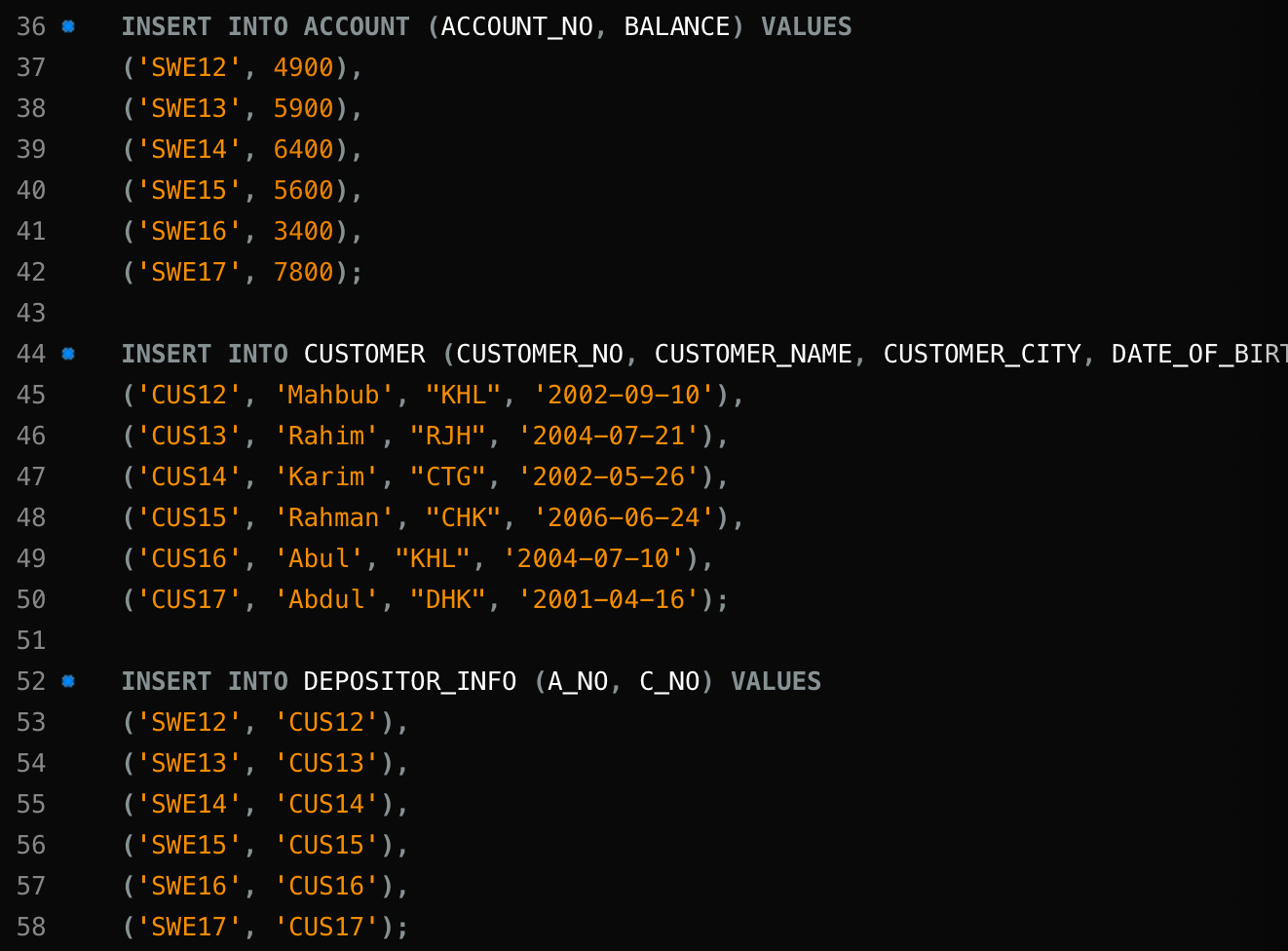
Changed the table name to DEPOSITOR\_INFO.

e> “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 constrains and naming them. Here C\_NO is foreign key and it is connected to CUSTOMER\_NO. “ON DELETE CASCADE ON UPDATE CASCADE” it is used so that if any data containing foreign key is changed it will automatically update or delete the connected data.

3



Inserted values into the tables.

4

a> “SELECT ACCOUNT\_NO FROM ACCOUNT WHERE BALANCE < 100000;”

Shows us the account number from the account table where the balance is below 100000.

b> “SELECT CUSTOMER\_NAME FROM CUSTOMER WHERE CUSTOMER\_CITY = 'KHL';”

Shows us the names of the customers whose city is KHL

c> “SELECT \* FROM ACCOUNT, DEPOSITOR\_INFO;”

Shows us all info in the account table and the depositor table.

d> “SELECT \* FROM CUSTOMER NATURAL JOIN DEPOSITOR\_INFO;”

this NATURAL JOIN keyword joins the tables data. Every single row of depositor\_info is added to the every customer rows.

e> “SELECT CUSTOMER\_NAME, CUSTOMER\_CITY FROM CUSTOMER, DEPOSITOR\_INFO WHERE CUSTOMER.CUSTOMER\_NO = DEPOSITOR\_INFO.C\_NO ;”

Here we are selecting name and city info from 2 tables where customer table’s customer\_no matches depositor\_info table’s c\_no.

f> “SELECT \* FROM CUSTOMER WHERE CUSTOMER\_NO IN (SELECT C\_NO FROM DEPOSITOR\_INFO WHERE A\_NO IN (SELECT ACCOUNT\_NO FROM ACCOUNT WHERE BALANCE > 1000));”

it is a nested condition.

we are selecting everything from customer table where customer\_no matches c\_no from DEPOSITOR\_INFO table where A\_NO matches from ACCOUNT table where balance is greater than 1000

5

a> “UPDATE CUSTOMER SET CUSTOMER\_CITY = 'KLN' WHERE CUSTOMER\_CITY = 'KHL';”

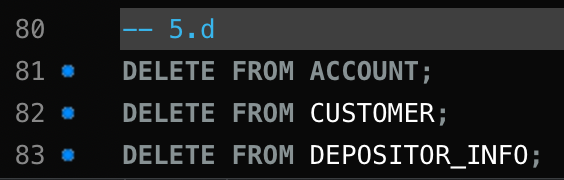
we are updating the table data of CUSTOMER\_CITY column where it is ’KHL’ we are updating it to ‘KLN’

b> “UPDATE CUSTOMER SET CUSTOMER\_NO = 'CUS32' WHERE CUSTOMER\_NO = 'CUS17';”

it will update the customer table data where the customer no is cus17 to cus32.

c> “DELETE FROM CUSTOMER WHERE CUSTOMER\_NO = 'CUS31';”

deletes the entry where customer no is cus31

d> 

deletes the table data without deleting the table itself.