# CSE 4308

## LAB 5 ASSIGNMENT

Task statement:

Our main goal for this task was to showcase various SQL commands by writing complex queries to show different information from given information.

Solution approach:

For the solution, we had to use functions such as LEFT JOIN, EXTRACT, UPDATE and we also had to use nested queries in order to match primary keys between separate tables and maintain data integrity.

A snippet of the solution is provided below:

1

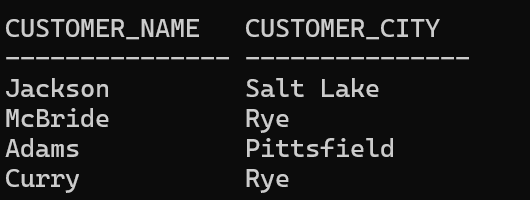
SELECT C.CUSTOMER\_NAME, C.CUSTOMER\_CITY

FROM CUSTOMER C

JOIN BORROWER B ON B.CUSTOMER\_NAME = C.CUSTOMER\_NAME

LEFT JOIN DEPOSITOR D ON D.CUSTOMER\_NAME = C.CUSTOMER\_NAME

WHERE D.ACCOUNT\_NUMBER IS NULL;

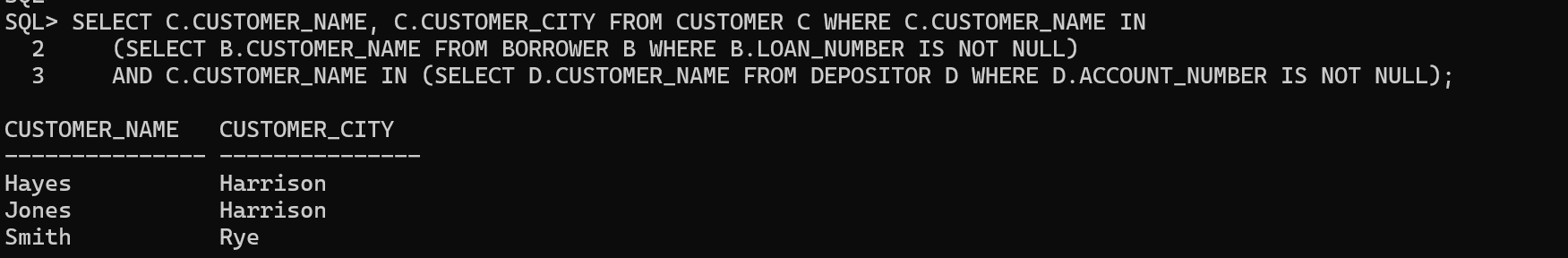


2

SELECT C.CUSTOMER\_NAME, C.CUSTOMER\_CITY FROM CUSTOMER C WHERE C.CUSTOMER\_NAME IN

(SELECT B.CUSTOMER\_NAME FROM BORROWER B WHERE B.LOAN\_NUMBER IS NOT NULL)

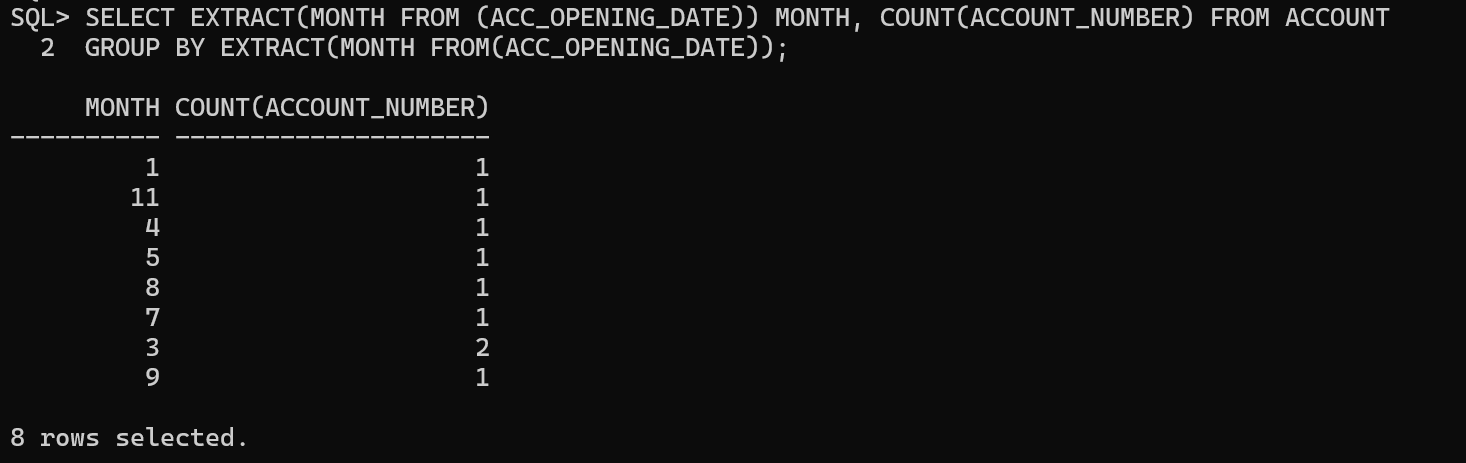
AND C.CUSTOMER\_NAME IN (SELECT D.CUSTOMER\_NAME FROM DEPOSITOR D WHERE D.ACCOUNT\_NUMBER IS NOT NULL);



3

SELECT EXTRACT(MONTH FROM (ACC\_OPENING\_DATE)) MONTH, COUNT(ACCOUNT\_NUMBER) FROM ACCOUNT

GROUP BY EXTRACT(MONTH FROM(ACC\_OPENING\_DATE));



4

SELECT MONTHS\_BETWEEN(MAX(L.LOAN\_DATE), MAX(AA.ACC\_OPENING\_DATE)) AS MONTHS\_BETWEEN FROM

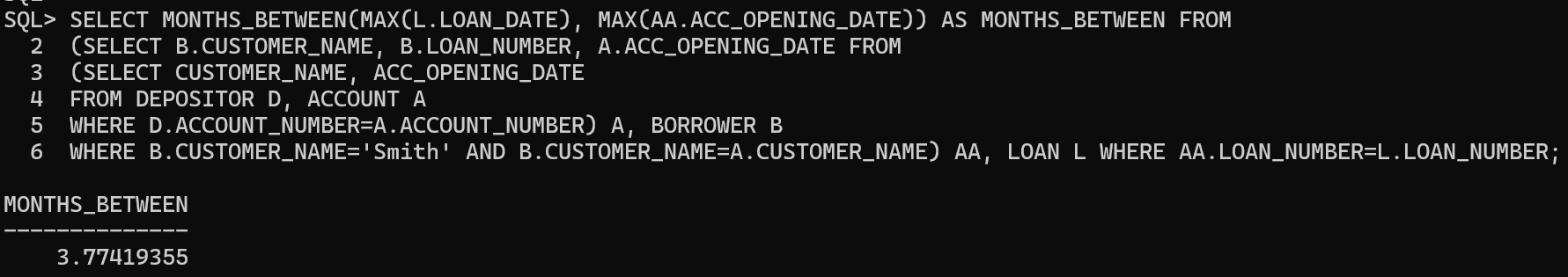
(SELECT B.CUSTOMER\_NAME, B.LOAN\_NUMBER, A.ACC\_OPENING\_DATE FROM

(SELECT CUSTOMER\_NAME, ACC\_OPENING\_DATE

FROM DEPOSITOR D, ACCOUNT A

WHERE D.ACCOUNT\_NUMBER=A.ACCOUNT\_NUMBER) A, BORROWER B

WHERE B.CUSTOMER\_NAME='Smith' AND B.CUSTOMER\_NAME=A.CUSTOMER\_NAME) AA, LOAN L WHERE AA.LOAN\_NUMBER=L.LOAN\_NUMBER;



5

SELECT A.CUSTOMER\_NAME,A.TOT\_BALANCE FROM

(SELECT CUSTOMER\_NAME, SUM(BALANCE) AS TOT\_BALANCE

FROM DEPOSITOR D, ACCOUNT A

WHERE D.ACCOUNT\_NUMBER=A.ACCOUNT\_NUMBER

GROUP BY CUSTOMER\_NAME) A

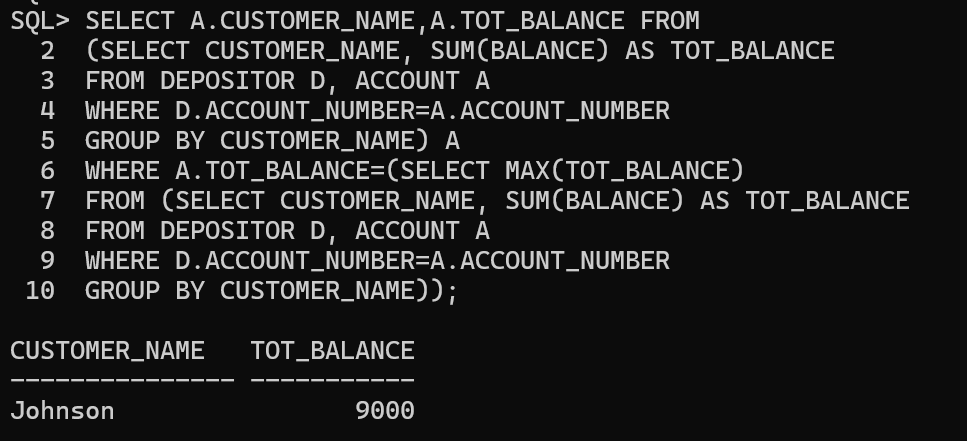
WHERE A.TOT\_BALANCE=(SELECT MAX(TOT\_BALANCE)

FROM (SELECT CUSTOMER\_NAME, SUM(BALANCE) AS TOT\_BALANCE

FROM DEPOSITOR D, ACCOUNT A

WHERE D.ACCOUNT\_NUMBER=A.ACCOUNT\_NUMBER

GROUP BY CUSTOMER\_NAME));



6

SELECT DISTINCT BR.CUSTOMER\_NAME || ' ELIGIBLE' FROM(

SELECT B.CUSTOMER\_NAME, L.AMOUNT AS LOAN\_VAL

FROM BORROWER B, LOAN L

WHERE L.LOAN\_NUMBER=B.LOAN\_NUMBER) TA,

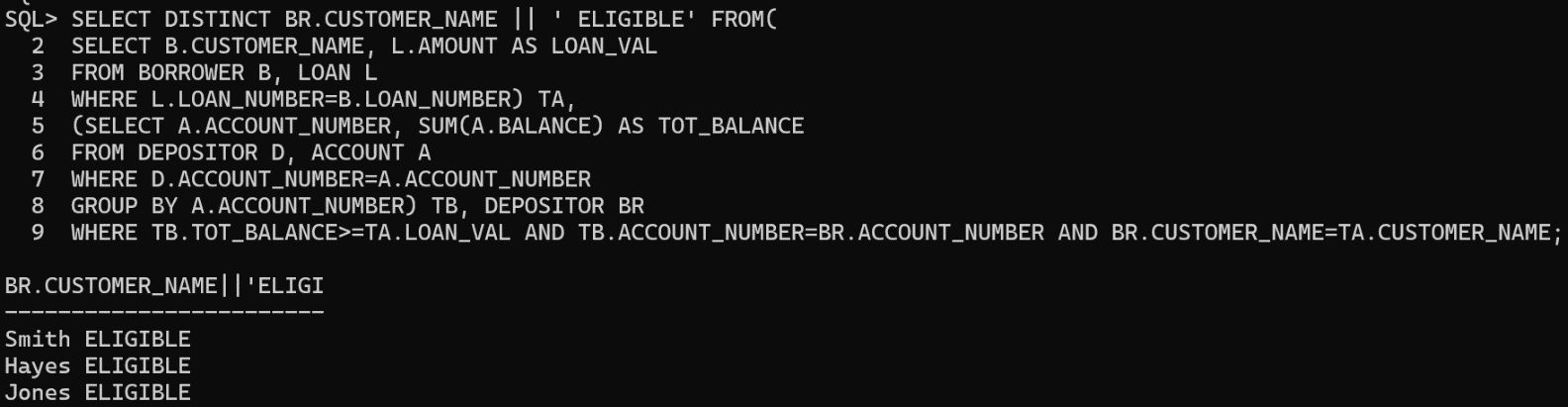
(SELECT A.ACCOUNT\_NUMBER, SUM(A.BALANCE) AS TOT\_BALANCE

FROM DEPOSITOR D, ACCOUNT A

WHERE D.ACCOUNT\_NUMBER=A.ACCOUNT\_NUMBER

GROUP BY A.ACCOUNT\_NUMBER) TB, DEPOSITOR BR

WHERE TB.TOT\_BALANCE>=TA.LOAN\_VAL AND TB.ACCOUNT\_NUMBER=BR.ACCOUNT\_NUMBER AND BR.CUSTOMER\_NAME=TA.CUSTOMER\_NAME;



7

SELECT CUSTOMER\_NAME ||' Elite' FROM(

SELECT CUSTOMER\_NAME, SUM(BALANCE) AS TOT\_BALANCE

FROM DEPOSITOR D, ACCOUNT A

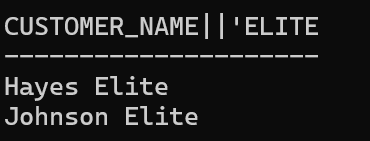
WHERE D.ACCOUNT\_NUMBER=A.ACCOUNT\_NUMBER

GROUP BY CUSTOMER\_NAME) A,

(SELECT AVG(BALANCE) AS AVG\_BALANCE

FROM ACCOUNT A) B

WHERE A.TOT\_BALANCE> B.AVG\_BALANCE+500;



SELECT CUSTOMER\_NAME ||' Moderate' FROM(

SELECT CUSTOMER\_NAME, SUM(BALANCE) AS TOT\_BALANCE

FROM DEPOSITOR D, ACCOUNT A

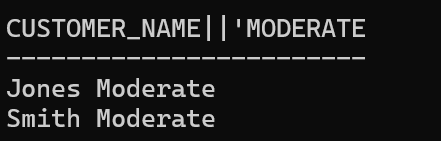
WHERE D.ACCOUNT\_NUMBER=A.ACCOUNT\_NUMBER

GROUP BY CUSTOMER\_NAME) A,

(SELECT AVG(BALANCE) AS AVG\_BALANCE

FROM ACCOUNT A) B

WHERE A.TOT\_BALANCE<= B.AVG\_BALANCE+500 AND A.TOT\_BALANCE>=B.AVG\_BALANCE-500;



SELECT CUSTOMER\_NAME ||' Poor' FROM(

SELECT CUSTOMER\_NAME, SUM(BALANCE) AS TOT\_BALANCE

FROM DEPOSITOR D, ACCOUNT A

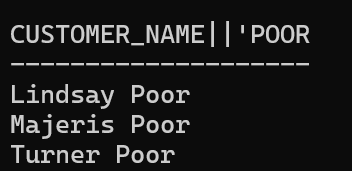
WHERE D.ACCOUNT\_NUMBER=A.ACCOUNT\_NUMBER

GROUP BY CUSTOMER\_NAME) A,

(SELECT AVG(BALANCE) AS AVG\_BALANCE

FROM ACCOUNT A) B

WHERE A.TOT\_BALANCE< B.AVG\_BALANCE-500;



8

SELECT DISTINCT B.BRANCH\_NAME, B.BRANCH\_CITY, B.ASSETS

FROM CUSTOMER C, BRANCH B

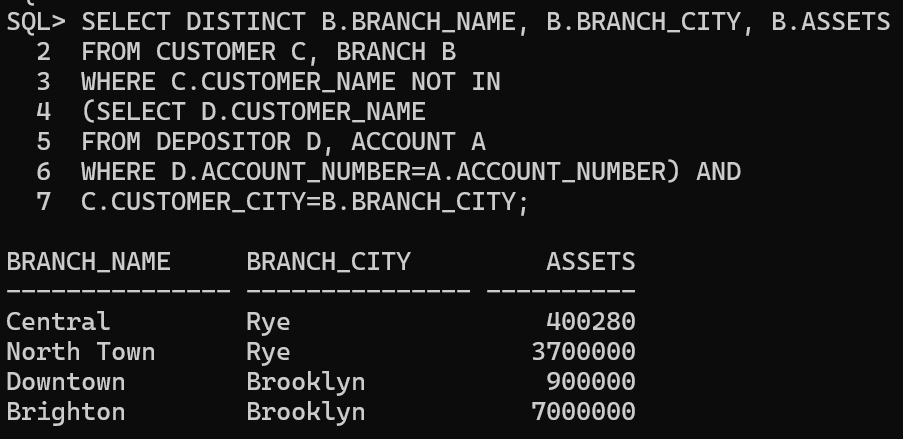
WHERE C.CUSTOMER\_NAME NOT IN

(SELECT D.CUSTOMER\_NAME

FROM DEPOSITOR D, ACCOUNT A

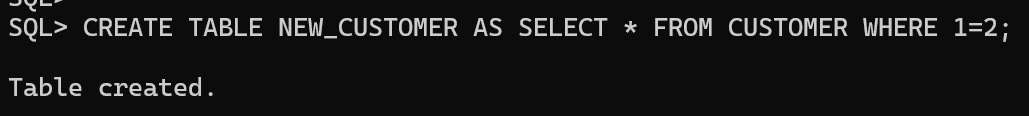
WHERE D.ACCOUNT\_NUMBER=A.ACCOUNT\_NUMBER) AND

C.CUSTOMER\_CITY=B.BRANCH\_CITY;



9

CREATE TABLE NEW\_CUSTOMER AS SELECT \* FROM CUSTOMER WHERE 1=2;



10

INSERT INTO NEW\_CUSTOMER

(SELECT DISTINCT C.CUSTOMER\_NAME, C.CUSTOMER\_STREET, C.CUSTOMER\_CITY

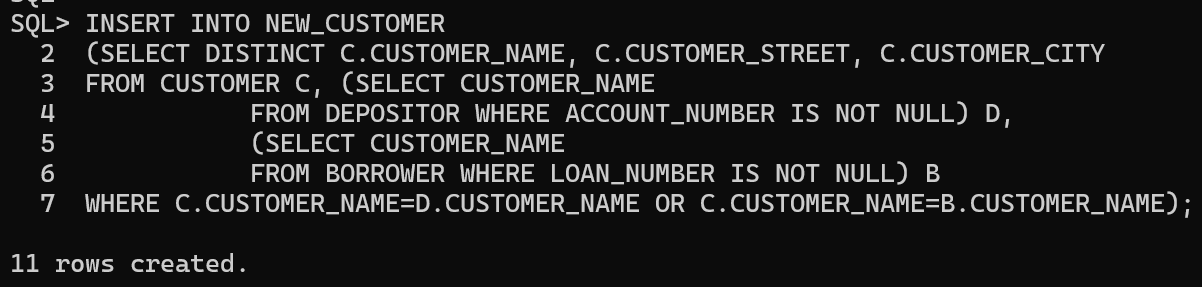
FROM CUSTOMER C, (SELECT CUSTOMER\_NAME

FROM DEPOSITOR WHERE ACCOUNT\_NUMBER IS NOT NULL) D,

(SELECT CUSTOMER\_NAME

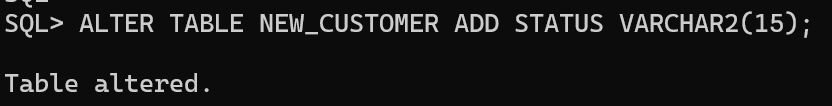
FROM BORROWER WHERE LOAN\_NUMBER IS NOT NULL) B

WHERE C.CUSTOMER\_NAME=D.CUSTOMER\_NAME OR C.CUSTOMER\_NAME=B.CUSTOMER\_NAME);



11

ALTER TABLE NEW\_CUSTOMER ADD STATUS VARCHAR2(15);



12

UPDATE NEW\_CUSTOMER

SET STATUS = CASE

WHEN

CUSTOMER\_NAME IN

(SELECT DISTINCT TA.CUSTOMER\_NAME

FROM

(SELECT SUM(A.BALANCE) AS BAL, D.CUSTOMER\_NAME

FROM DEPOSITOR D, ACCOUNT A

WHERE D.ACCOUNT\_NUMBER=A.ACCOUNT\_NUMBER

GROUP BY D.CUSTOMER\_NAME) TA,

(SELECT SUM(L.AMOUNT) AS LO, B.CUSTOMER\_NAME

FROM BORROWER B, LOAN L

WHERE B.LOAN\_NUMBER=L.LOAN\_NUMBER

GROUP BY B.CUSTOMER\_NAME) TB

WHERE TA.BAL>TB.LO AND TA.CUSTOMER\_NAME=TB.CUSTOMER\_NAME)

THEN 'In Savings'

WHEN

CUSTOMER\_NAME IN

(SELECT DISTINCT TA.CUSTOMER\_NAME

FROM

(SELECT SUM(A.BALANCE) AS BAL, D.CUSTOMER\_NAME

FROM DEPOSITOR D, ACCOUNT A

WHERE D.ACCOUNT\_NUMBER=A.ACCOUNT\_NUMBER

GROUP BY D.CUSTOMER\_NAME) TA,

(SELECT SUM(L.AMOUNT) AS LO, B.CUSTOMER\_NAME

FROM BORROWER B, LOAN L

WHERE B.LOAN\_NUMBER=L.LOAN\_NUMBER

GROUP BY B.CUSTOMER\_NAME) TB

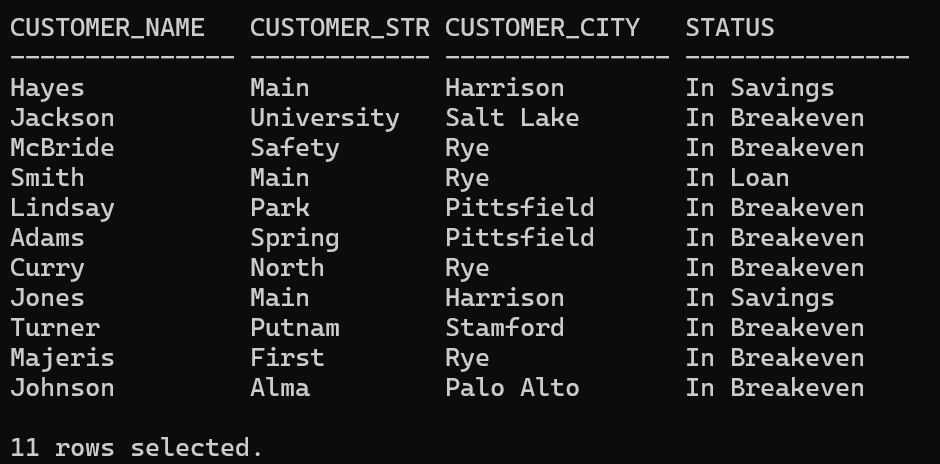
WHERE TA.BAL<TB.LO AND TA.CUSTOMER\_NAME=TB.CUSTOMER\_NAME)

THEN 'In Loan'

ELSE

'In Breakeven'

END;



Problems I faced:

I faced problems while joining the tables and also while dealing with null and not null values. I also faced some problems while using the EXTRACT function as they were something I’ve never worked with before.

SUBMITTED BY

MEHREEN HOSSAIN CHOWDHURY

210041219