NAME

Shaikh Mohammad Shaad Shabbir Ali

REG. NO.

21BCE1542

VIT®
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

Lab Exercise 1 – Jan 2023

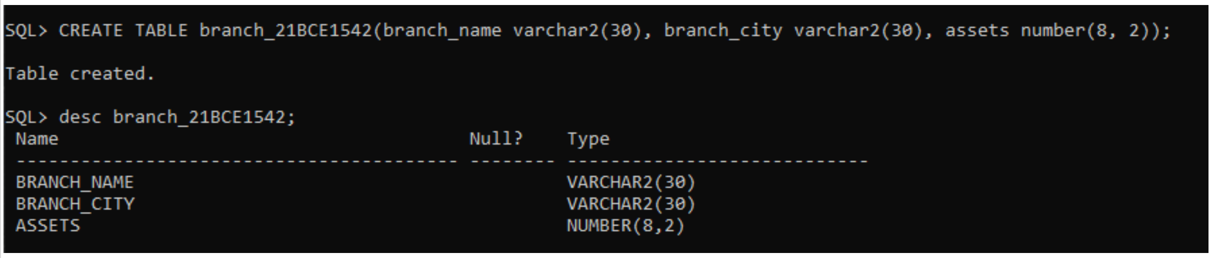
Name		**Mohammad Shaad**	**Reg. No.**		**21BCE1542**
Programme	:	BTech - BCE	Semester	:	WIN 2022-23
Course Title	:	DBMS Lab	Code	:	BCSE302L
			Class Nbr(s)	:	
Faculty(s)	:	Dr Leninisha Shanmugam	Slot		L13+L14+L43+L44
Date	:	04/01/2023			

DDL Commands and Constraints

Q.No.	Question Description	Marks
1	**Create and describe the following tables:**	10

A) **NAME**: branch

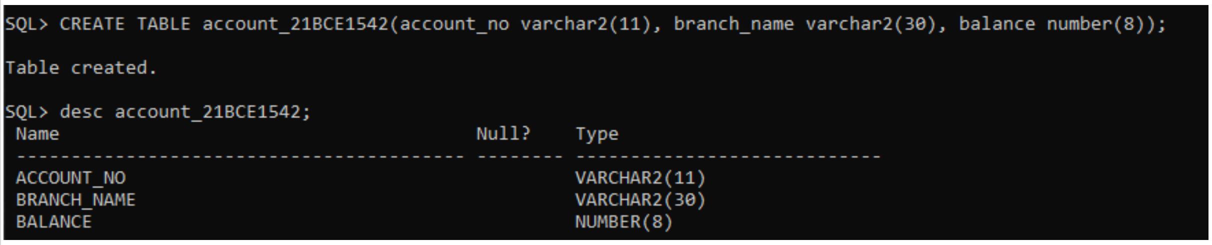
FIELDS	**DATATYPE**
branch_name	varchar2(30)
branch_city	varchar2(30)
assets	number(8,2)



```
SQL> CREATE TABLE branch_21BCE1542(branch_name varchar2(30), branch_city varchar2(30), assets number(8, 2));

Table created.

SQL> desc branch_21BCE1542;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 BRANCH_NAME                                        VARCHAR2(30)
 BRANCH_CITY                                        VARCHAR2(30)
 ASSETS                                             NUMBER(8,2)
```

B) **NAME**: account

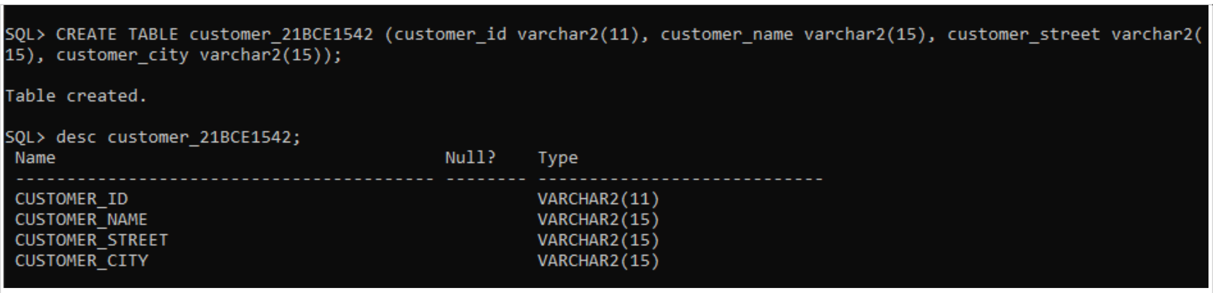
FIELDS	**DATATYPE**
account_no	varchar2(11)
branch_name	varchar2(30)
balance	number(8)



```
SQL> CREATE TABLE account_21BCE1542(account_no varchar2(11), branch_name varchar2(30), balance number(8));

Table created.

SQL> desc account_21BCE1542;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 ACCOUNT_NO                                         VARCHAR2(11)
 BRANCH_NAME                                        VARCHAR2(30)
 BALANCE                                            NUMBER(8)
```

C) **NAME**: customer

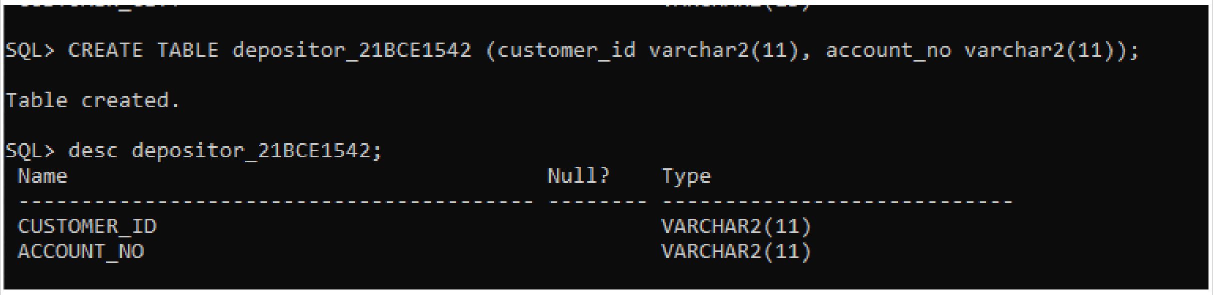
FIELD	**DATATYPE**
customer_id	varchar2(11)
customer_name	varchar2(20)
customer_street	varchar2(15)
customer_city	varchar2(15)



```
SQL> CREATE TABLE customer_21BCE1542 (customer_id varchar2(11), customer_name varchar2(15), customer_street varchar2(
15), customer_city varchar2(15));

Table created.

SQL> desc customer_21BCE1542;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 CUSTOMER_ID                                        VARCHAR2(11)
 CUSTOMER_NAME                                      VARCHAR2(15)
 CUSTOMER_STREET                                    VARCHAR2(15)
 CUSTOMER_CITY                                      VARCHAR2(15)
```

D) **NAME**: depositor

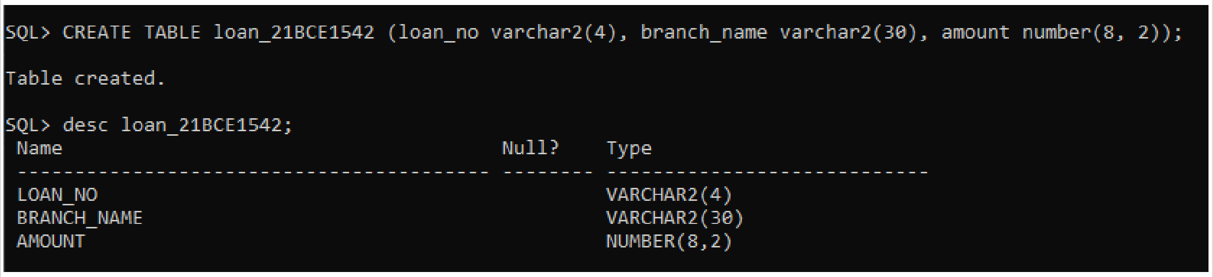
FIELD	**DATATYPE**
customer_id	varchar2(11)
account_no	varchar2(11)



```
SQL> CREATE TABLE depositor_21BCE1542 (customer_id varchar2(11), account_no varchar2(11));

Table created.

SQL> desc depositor_21BCE1542;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 CUSTOMER_ID                                        VARCHAR2(11)
 ACCOUNT_NO                                         VARCHAR2(11)
```

E) **NAME**: loan

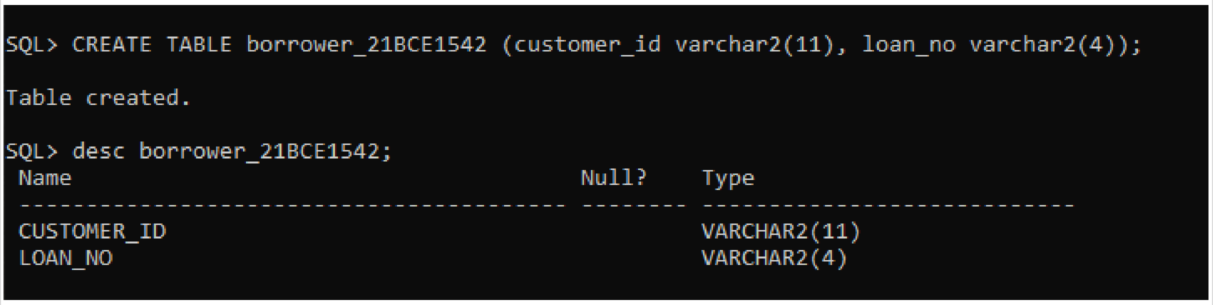
FIELDS	**DATATYPE**
loan_no	varchar2(4)
branch_name	varchar2(30)
amount	number(8,2)



```
SQL> CREATE TABLE loan_21BCE1542 (loan_no varchar2(4), branch_name varchar2(30), amount number(8, 2));

Table created.

SQL> desc loan_21BCE1542;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 LOAN_NO                                            VARCHAR2(4)
 BRANCH_NAME                                        VARCHAR2(30)
 AMOUNT                                             NUMBER(8,2)
```

F) **NAME**: borrower

FIELDS	**DATATYPE**
customer_id	varchar2(11)
loan_no	varcahr2(4)



```
SQL> CREATE TABLE borrower_21BCE1542 (customer_id varchar2(11), loan_no varchar2(4));

Table created.

SQL> desc borrower_21BCE1542;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 CUSTOMER_ID                                        VARCHAR2(11)
 LOAN_NO                                            VARCHAR2(4)
```

2 **Describe the structure of all database schemas.**

Table "branch" contains information about bank branches, including the branch name, city, and total assets. The branch name and city are stored as strings of up to 30 characters, while the assets are stored as a number with up to 8 digits and 2 decimal places.

Table "account" contains information about bank accounts, including the account number, branch name, and balance. The account number and branch name are stored as strings of up to 11 and 30 characters respectively, while the balance is stored as a number with up to 8 digits.

Table "customer" contains information about bank customers, including their customer ID, name, street, and city. The customer ID, name, street, and city are stored as strings of up to 11, 20, 15, and 15 characters respectively.

Table "depositor" connects customers to their accounts. It has two columns, "customer_id" and "account_no", both stored as strings of up to 11 characters, which reference the corresponding customer ID and account number in the "customer" and "account" tables respectively.

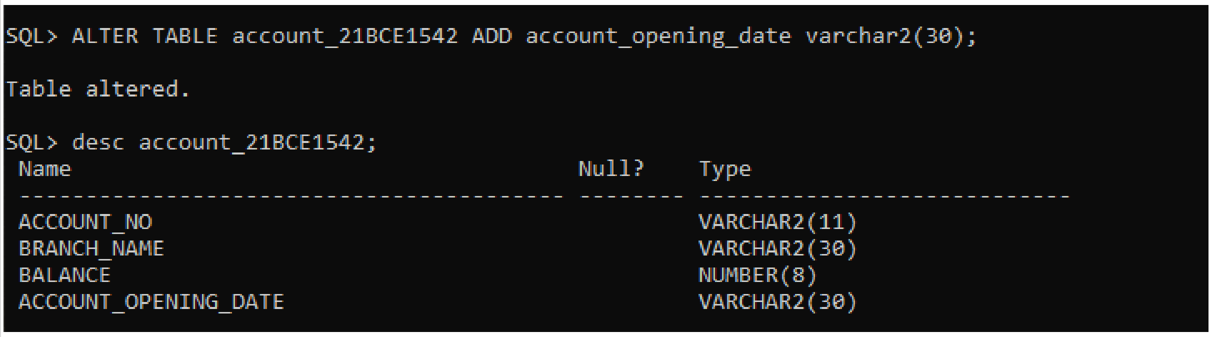
Table "loan" contains information about loans granted by the bank, including the loan number, branch name, and amount. The loan number and branch name are stored as strings of up to 4 and 30 characters respectively, while the amount is stored as a number with up to 8 digits and 2 decimal places.

Table "borrower" connects customers to their loans. It has two columns, "customer_id" and "loan_no", both stored as strings of up to 11 and 4 characters respectively, which reference the corresponding customer ID and loan number in the "customer" and "loan" tables respectively.

Together, these tables make up the database schema for a simple banking system.

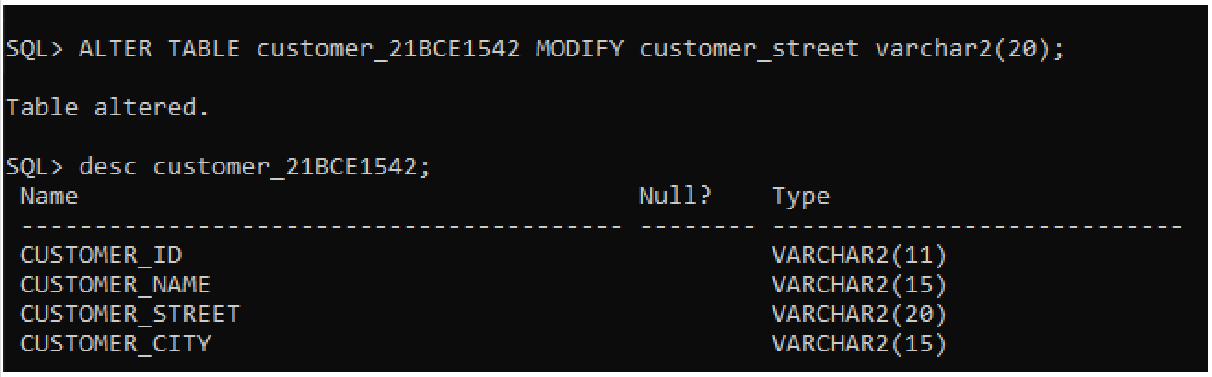
3 **Alter the structure of the Database**

a. Add a new column ‘account opening date’ in the account table.



```
SQL> ALTER TABLE account_21BCE1542 ADD account_opening_date varchar2(30);

Table altered.

SQL> desc account_21BCE1542;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 ACCOUNT_NO                                         VARCHAR2(11)
 BRANCH_NAME                                        VARCHAR2(30)
 BALANCE                                            NUMBER(8)
 ACCOUNT_OPENING_DATE                               VARCHAR2(30)
```

b. Increase the width of the column customer_street in table customer to 20.

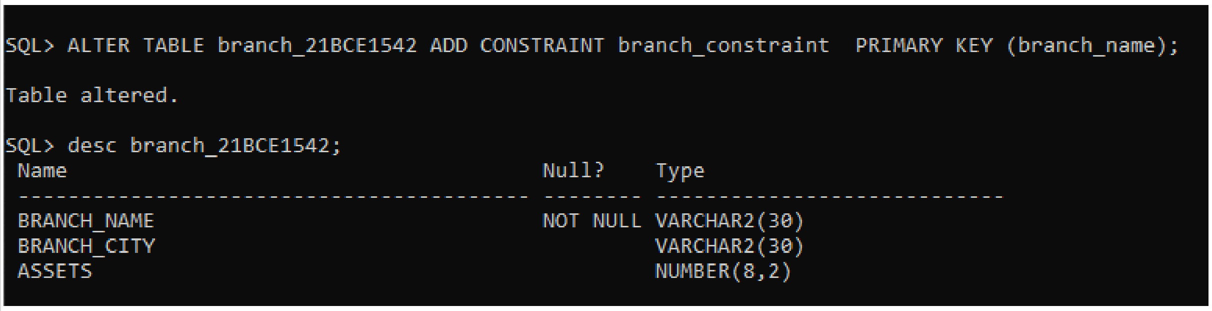


```
SQL> ALTER TABLE customer_21BCE1542 MODIFY customer_street varchar2(20);

Table altered.

SQL> desc customer_21BCE1542;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 CUSTOMER_ID                                        VARCHAR2(11)
 CUSTOMER_NAME                                      VARCHAR2(15)
 CUSTOMER_STREET                                    VARCHAR2(20)
 CUSTOMER_CITY                                      VARCHAR2(15)
```

4 **Add primary keys to all the tables for the specified attributes**

A) **NAME**: branch

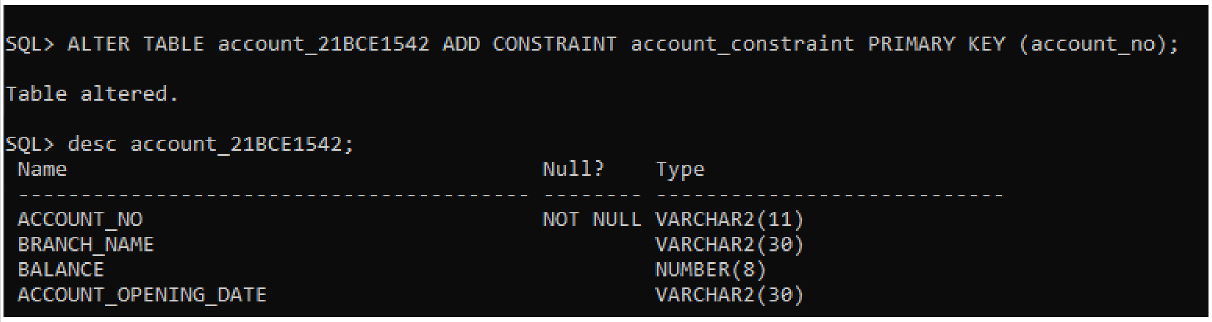
FIELDS	**DATATYPE**
branch_name	varchar2(30) primary key
branch_city	varchar2(30)
assets	number(8,2)



```
SQL> ALTER TABLE branch_21BCE1542 ADD CONSTRAINT branch_constraint  PRIMARY KEY (branch_name);

Table altered.

SQL> desc branch_21BCE1542;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 BRANCH_NAME                               NOT NULL VARCHAR2(30)
 BRANCH_CITY                                        VARCHAR2(30)
 ASSETS                                             NUMBER(8,2)
```

B) **NAME**: account

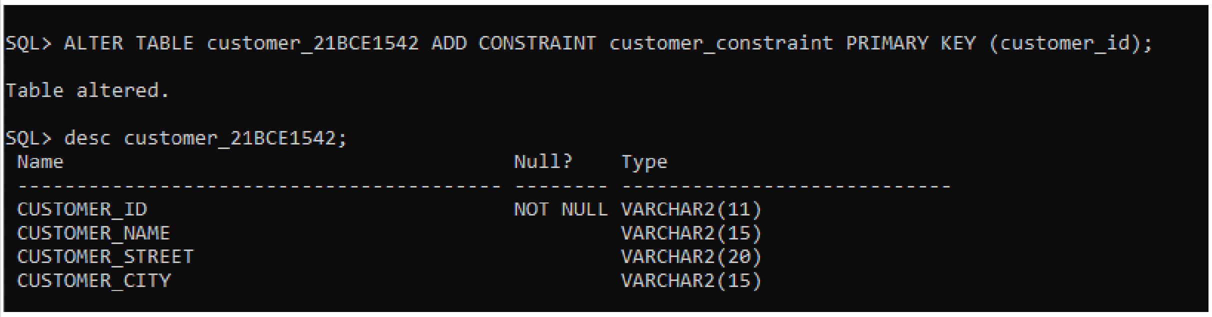
FIELDS	**DATATYPE**
account_no	varchar2(11) primary key
branch_name	varchar2(30)
balance	number(8)



```
SQL> ALTER TABLE account_21BCE1542 ADD CONSTRAINT account_constraint PRIMARY KEY (account_no);

Table altered.

SQL> desc account_21BCE1542;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 ACCOUNT_NO                                NOT NULL VARCHAR2(11)
 BRANCH_NAME                                        VARCHAR2(30)
 BALANCE                                            NUMBER(8)
 ACCOUNT_OPENING_DATE                               VARCHAR2(30)
```

C) **NAME**: customer

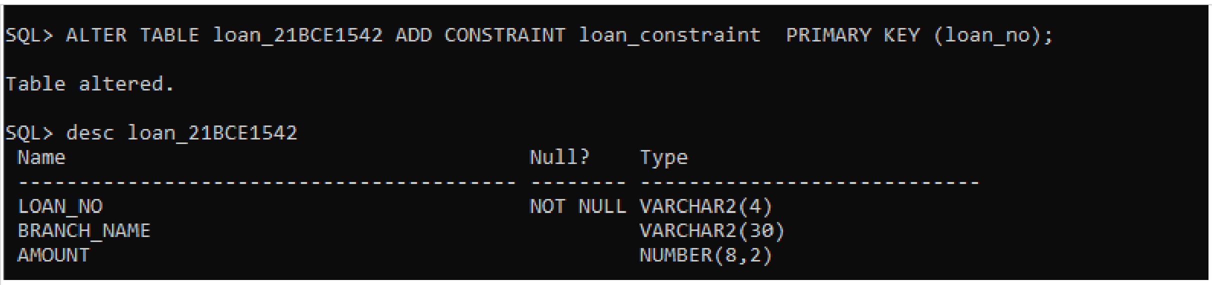
FIELD	**DATATYPE**
customer_id	varchar2(11) primary key
customer_name	varchar2(20)
customer_street	varchar2(15)
customer_city	varchar2(15)



```
SQL> ALTER TABLE customer_21BCE1542 ADD CONSTRAINT customer_constraint PRIMARY KEY (customer_id);

Table altered.

SQL> desc customer_21BCE1542;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 CUSTOMER_ID                               NOT NULL VARCHAR2(11)
 CUSTOMER_NAME                                      VARCHAR2(15)
 CUSTOMER_STREET                                    VARCHAR2(20)
 CUSTOMER_CITY                                      VARCHAR2(15)
```

D) **NAME**: loan

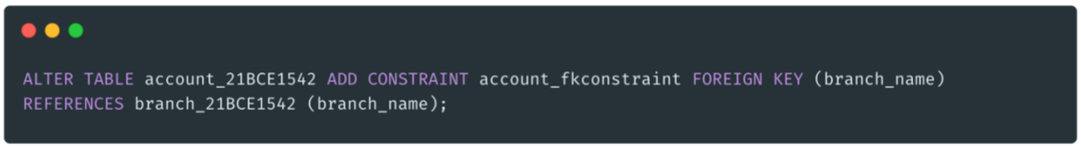
FIELDS	**DATATYPE**
loan_no	varchar2(4) primary key
branch_name	varchar2(30)
amount	number(8,2)


```
SQL> ALTER TABLE loan_21BCE1542 ADD CONSTRAINT loan_constraint  PRIMARY KEY (loan_no);

Table altered.

SQL> desc loan_21BCE1542
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 LOAN_NO                                   NOT NULL VARCHAR2(4)
 BRANCH_NAME                                        VARCHAR2(30)
 AMOUNT                                             NUMBER(8,2)
```

5 **Add foreign keys to the following tables for the specified attributes with mentioned reference table**

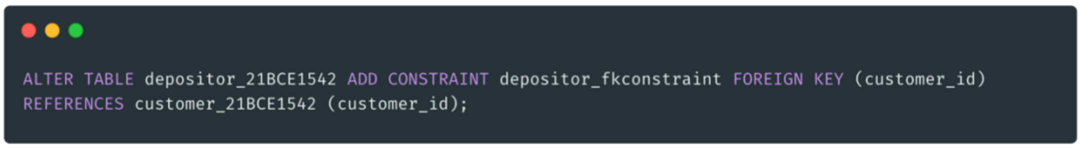
B) **NAME**: account

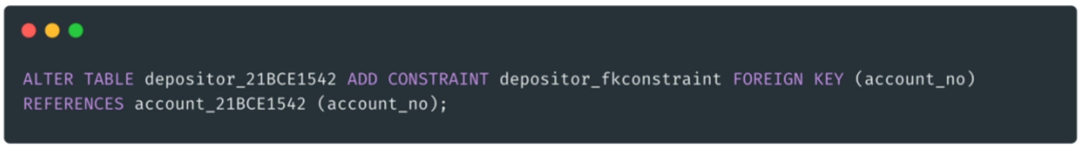
FIELDS	**DATATYPE**
account_no	varchar2(11) primary key
branch_name	varchar2(30) references branch(branch_name)
balance	number(8)


```
ALTER TABLE account_21BCE1542 ADD CONSTRAINT account_fkconstraint FOREIGN KEY (branch_name)
REFERENCES branch_21BCE1542 (branch_name);
```

C) **NAME**: depositor

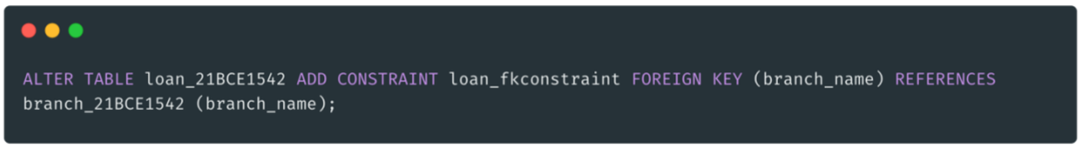
FIELD	**DATATYPE**
customer_id	varchar2(11)references customer (customer_id)
account_no	varchar2(11)references account (account_no)


```
ALTER TABLE depositor_21BCE1542 ADD CONSTRAINT depositor_fkconstraint FOREIGN KEY (customer_id)
REFERENCES customer_21BCE1542 (customer_id);
```


```
ALTER TABLE depositor_21BCE1542 ADD CONSTRAINT depositor_fkconstraint FOREIGN KEY (account_no)
REFERENCES account_21BCE1542 (account_no);
```

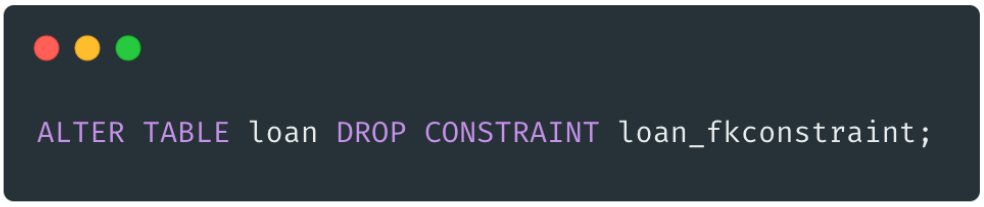
D) **NAME**: loan

FIELDS	**DATATYPE**
loan_no	varchar2(4) primary key
branch_name	varchar2(30) references branch(branch_name) (Create constraint with constraint name)
amount	number(8,2)



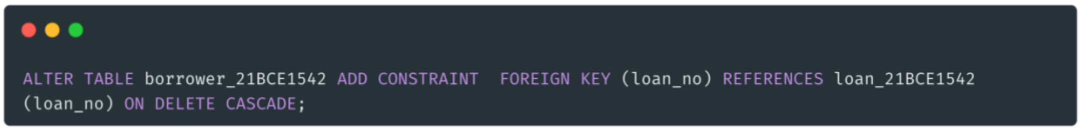
```
ALTER TABLE loan_21BCE1542 ADD CONSTRAINT loan_fkconstraint FOREIGN KEY (branch_name) REFERENCES
branch_21BCE1542 (branch_name);
```

6 Drop foreign key constraint from loan table



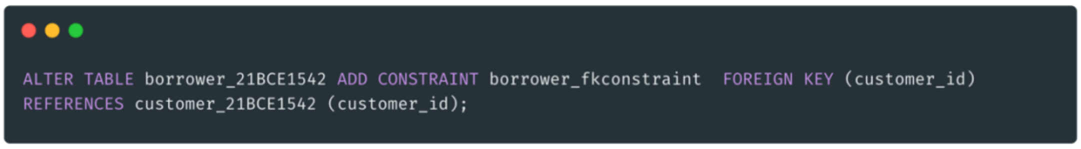
```
ALTER TABLE loan DROP CONSTRAINT loan_fkconstraint;
```

7 Set loan_no attribute of borrower table as foreign key with cascade deletion, which refers to loan table loan_no column.



```
ALTER TABLE borrower_21BCE1542 ADD CONSTRAINT  FOREIGN KEY (loan_no) REFERENCES loan_21BCE1542
(loan_no) ON DELETE CASCADE;
```

8 Add foreign key for the customer_id of borrower table which refers to customer table with constraint name.



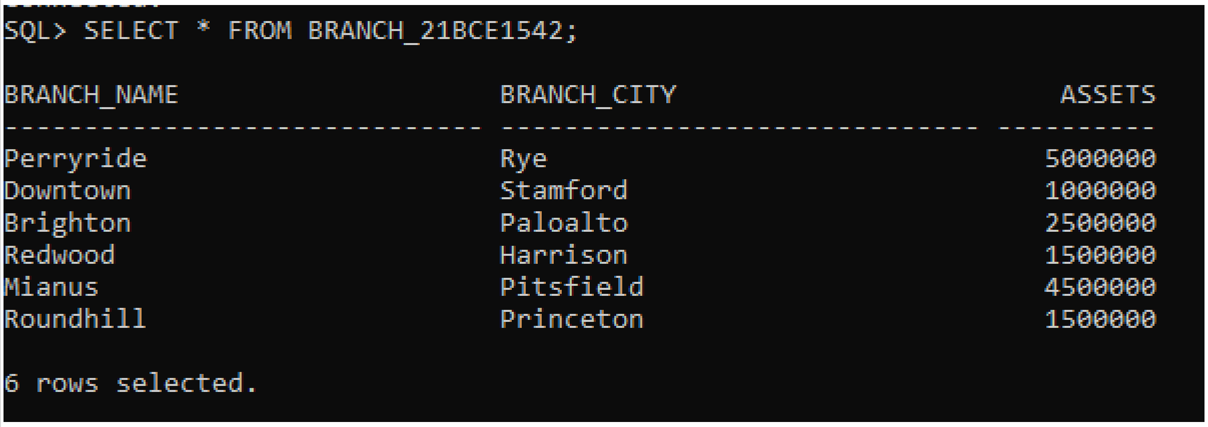
```
ALTER TABLE borrower_21BCE1542 ADD CONSTRAINT borrower_fkconstraint  FOREIGN KEY (customer_id)
REFERENCES customer_21BCE1542 (customer_id);
```

9

Insert the following values into the tables

1. branch :

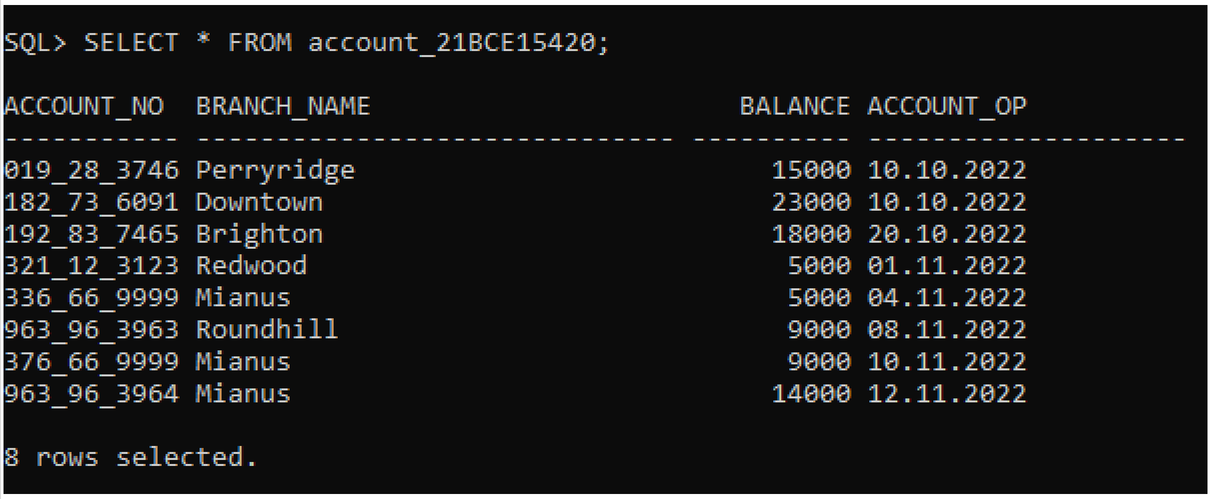
BRANCH_NAME	BRANCH_CITY	ASSETS
Perryridge	Rye	5000000
Downtown	Stamford	1000000
Brighton	Paloalto	2500000
Redwood	Harrison	1500000
Mianus	Pitsfield	4500000
Roundhill	Princeton	1500000



```
SQL> SELECT * FROM BRANCH_21BCE1542;

BRANCH_NAME                    BRANCH_CITY                        ASSETS
------------------------------ ------------------------------ ----------
Perryride                      Rye                               5000000
Downtown                       Stamford                          1000000
Brighton                       Paloalto                          2500000
Redwood                        Harrison                          1500000
Mianus                         Pitsfield                         4500000
Roundhill                      Princeton                         1500000

6 rows selected.
```

2. account :

ACCOUNT_NO	BRANCH_NAME	BALANCE
019_28_3746	Perryridge	15000
182_73_6091	Downtown	23000
192_83_7465	Brighton	18000
321_12_3123	Redwood	5000
336_66_9999	Mianus	5000
963_96_3963	Roundhill	5000
376_66_9999	Mianus	9000
963_96_3964	Mianus	13000

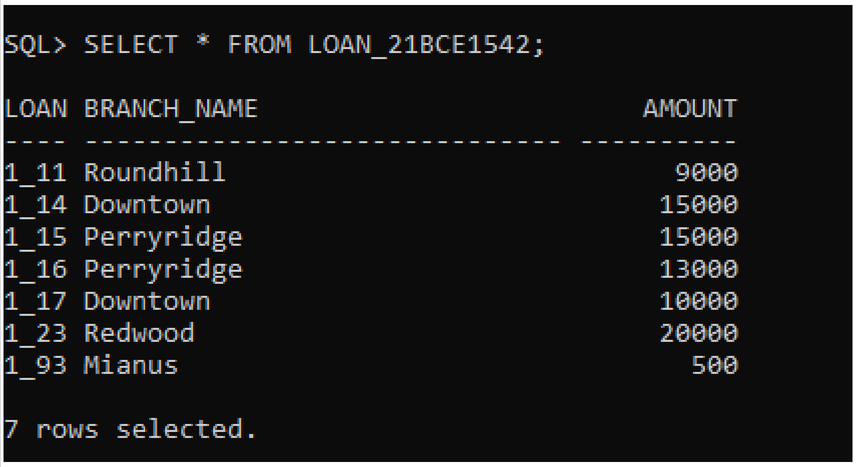


```
SQL> SELECT * FROM account_21BCE15420;

ACCOUNT_NO  BRANCH_NAME                       BALANCE ACCOUNT_OP
----------- ------------------------------ ---------- --------------------
019_28_3746 Perryridge                          15000 10.10.2022
182_73_6091 Downtown                            23000 10.10.2022
192_83_7465 Brighton                            18000 20.10.2022
321_12_3123 Redwood                              5000 01.11.2022
336_66_9999 Mianus                               5000 04.11.2022
963_96_3963 Roundhill                            9000 08.11.2022
376_66_9999 Mianus                               9000 10.11.2022
963_96_3964 Mianus                              14000 12.11.2022

8 rows selected.
```

3. loan :

LOAN	BRANCH_NAME	AMOUNT
1_11	Roundhill	9000
1_14	Downtown	15000
1_15	Perryridge	15000
1_16	Perryridge	13000
1_17	Downtown	10000

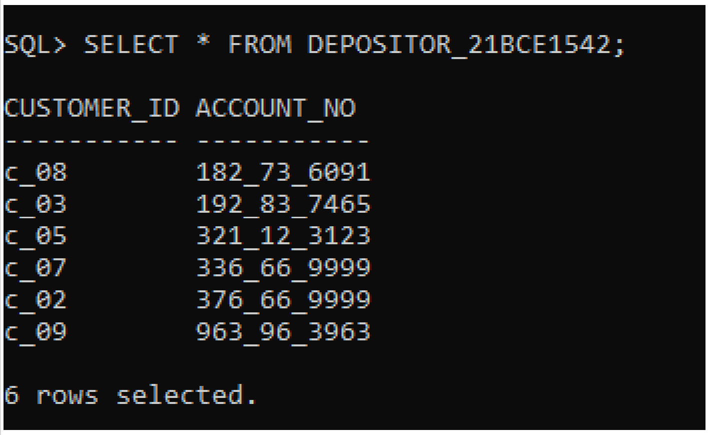
1_23	Redwood	20000
1_93	Mianus	500



```
SQL> SELECT * FROM LOAN_21BCE1542;

LOAN BRANCH_NAME                              AMOUNT
---- ------------------------------ ----------
1_11 Roundhill                                  9000
1_14 Downtown                                  15000
1_15 Perryridge                                15000
1_16 Perryridge                                13000
1_17 Downtown                                  10000
1_23 Redwood                                   20000
1_93 Mianus                                      500

7 rows selected.
```

4. depositor

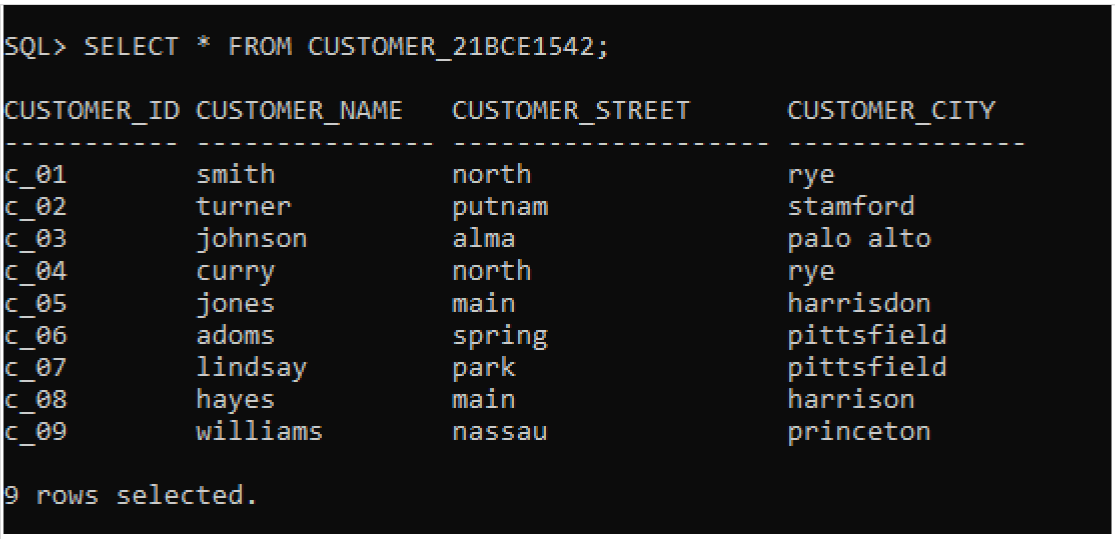
CUSTOMER_ID	ACCOUNT_NO
c_08	182_73_6091
c_03	192_83_7465
c_05	321_12_3123
c_07	336_66_9999
c_08	963_96_3963
c_02	376_66_9999



```
SQL> SELECT * FROM DEPOSITOR_21BCE1542;

CUSTOMER_ID ACCOUNT_NO
----------- -----------
c_08        182_73_6091
c_03        192_83_7465
c_05        321_12_3123
c_07        336_66_9999
c_02        376_66_9999
c_09        963_96_3963

6 rows selected.
```

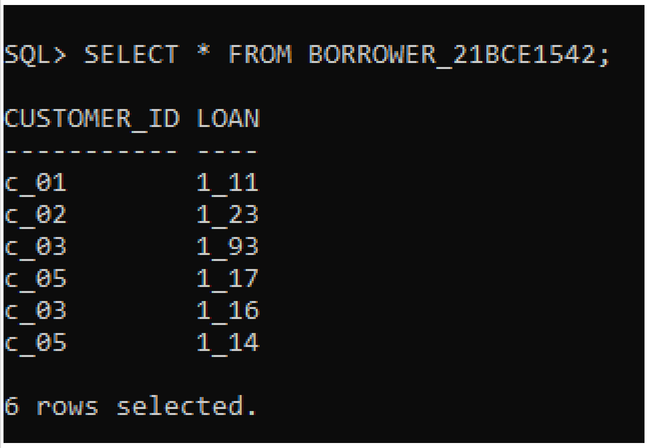
5. customer

CUSTOMER_ID	CUSTOMER_NAME	CUSTOMER_STREET	CUSTOMER_CIT
c_01	smith	north	rye
c_02	turner	putnam	stamford
c_03	johnson	alma	palo alto
c_04	curry	north	rye
c_05	jones	main	harrisdon
c_06	adoms	spring	pittsfield
c_07	lindsay	park	pittsfield
c_08	hayes	main	harrison
c_09	williams	nassau	Princeton


```
SQL> SELECT * FROM CUSTOMER_21BCE1542;

CUSTOMER_ID CUSTOMER_NAME   CUSTOMER_STREET      CUSTOMER_CITY
----------- --------------- -------------------- ---------------
c_01        smith           north                rye
c_02        turner          putnam               stamford
c_03        johnson         alma                 palo alto
c_04        curry           north                rye
c_05        jones           main                 harrisdon
c_06        adoms           spring               pittsfield
c_07        lindsay         park                 pittsfield
c_08        hayes           main                 harrison
c_09        williams        nassau               princeton

9 rows selected.
```

6. borrower

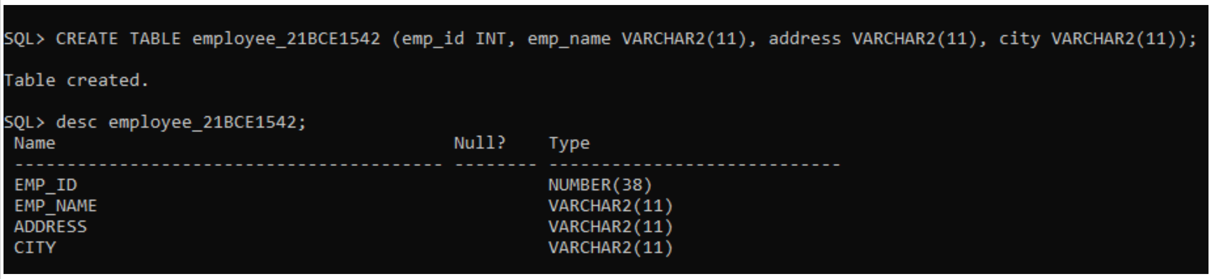
CUSTOMER_ID	LOAN_NO
c_01	1_11
c_01	1_23
c_03	1_93
c_05	1_17
c_03	1_16
c_05	1_14

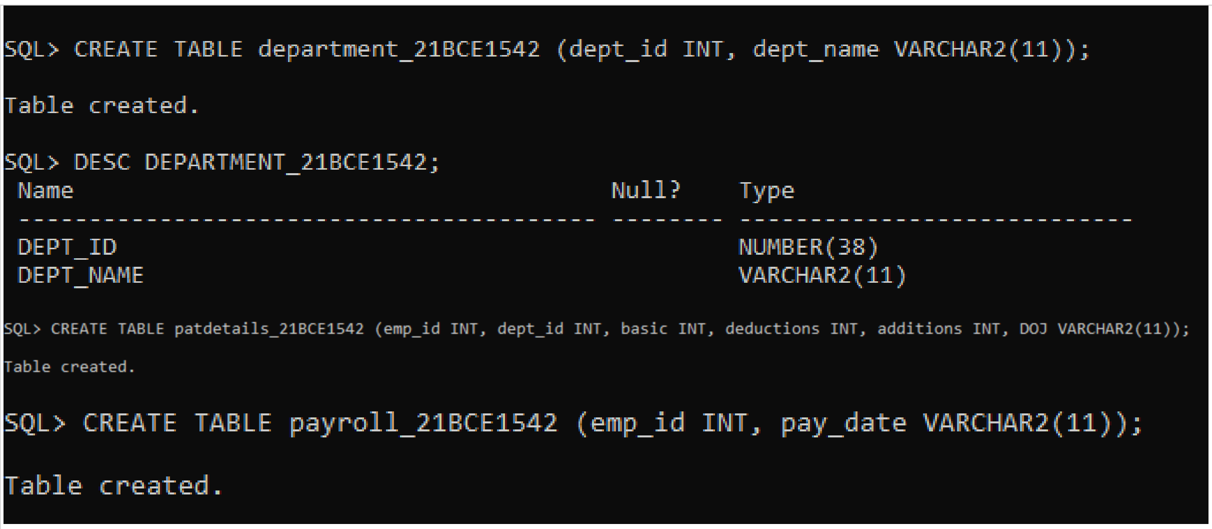

```
SQL> SELECT * FROM BORROWER_21BCE1542;

CUSTOMER_ID LOAN
----------- ----
c_01        1_11
c_02        1_23
c_03        1_93
c_05        1_17
c_03        1_16
c_05        1_14

6 rows selected.
```

10 **Create the Database Schema for a Employee-pay scenario**

a) employee(emp_id : integer, emp_name: string, address: string, city: string)
b) department(dept_id: integer, dept_name:string)
c) paydetails(emp_id : integer, dept_id: integer, basic: integer, deductions: integer, additions: integer, DOJ: date)
d) payroll(emp_id : integer, pay_date: date)

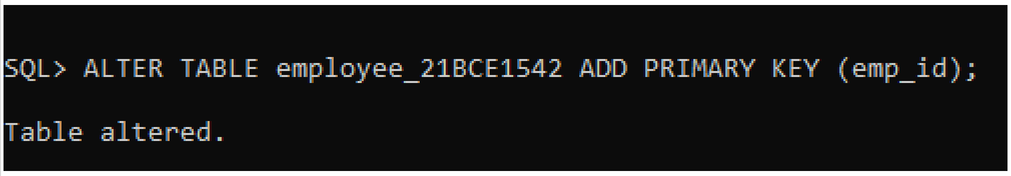
For the above schema, perform the following:


```
SQL> CREATE TABLE employee_21BCE1542 (emp_id INT, emp_name VARCHAR2(11), address VARCHAR2(11), city VARCHAR2(11));

Table created.

SQL> desc employee_21BCE1542;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 EMP_ID                                             NUMBER(38)
 EMP_NAME                                           VARCHAR2(11)
 ADDRESS                                            VARCHAR2(11)
 CITY                                               VARCHAR2(11)
```

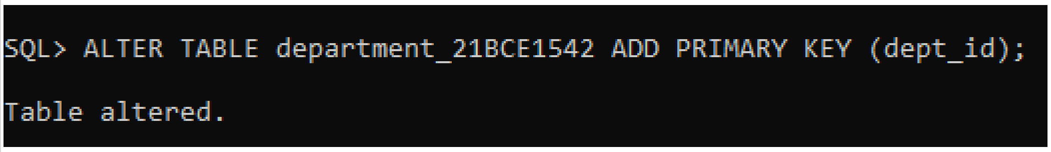


```
SQL> CREATE TABLE department_21BCE1542 (dept_id INT, dept_name VARCHAR2(11));

Table created.

SQL> DESC DEPARTMENT_21BCE1542;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 DEPT_ID                                            NUMBER(38)
 DEPT_NAME                                          VARCHAR2(11)

SQL> CREATE TABLE patdetails_21BCE1542 (emp_id INT, dept_id INT, basic INT, deductions INT, additions INT, DOJ VARCHAR2(11));

Table created.

SQL> CREATE TABLE payroll_21BCE1542 (emp_id INT, pay_date VARCHAR2(11));

Table created.
```

11 **Create PRIMARY KEY for employee(emp_id) and department(dept_id).**

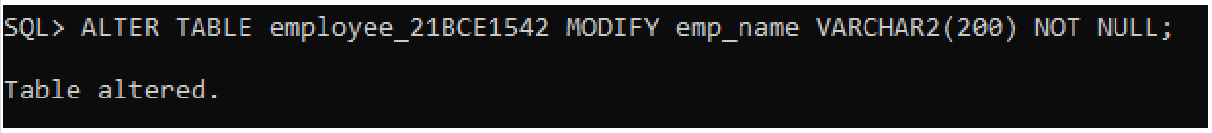


```
SQL> ALTER TABLE employee_21BCE1542 ADD PRIMARY KEY (emp_id);

Table altered.
```



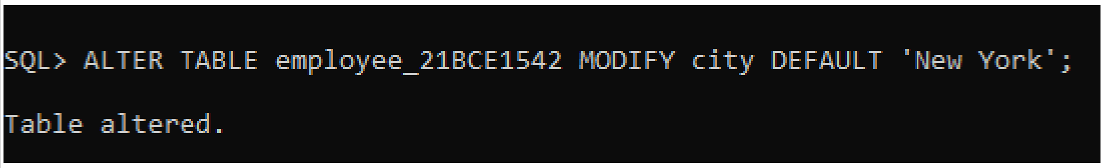
```
SQL> ALTER TABLE department_21BCE1542 ADD PRIMARY KEY (dept_id);

Table altered.
```

12 **Enforce NOT NULL constraint for emp_name.**



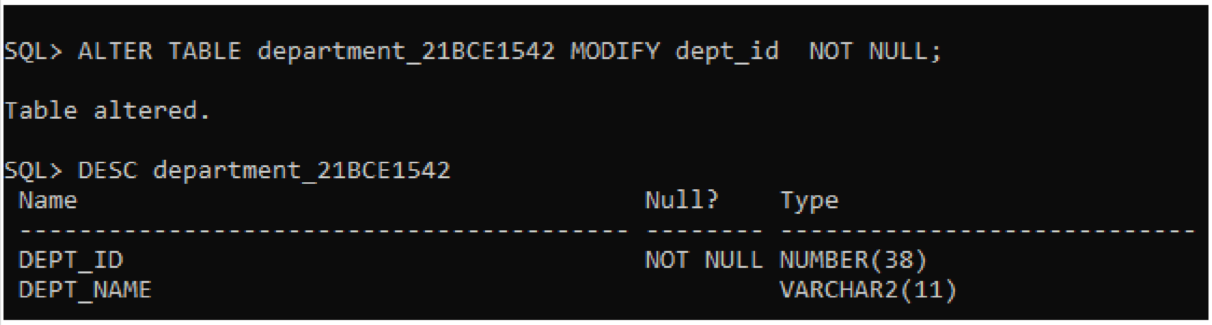
```
SQL> ALTER TABLE employee_21BCE1542 MODIFY emp_name VARCHAR2(200) NOT NULL;

Table altered.
```

13 **Creates a DEFAULT constraint on the "City" column of employee table.**



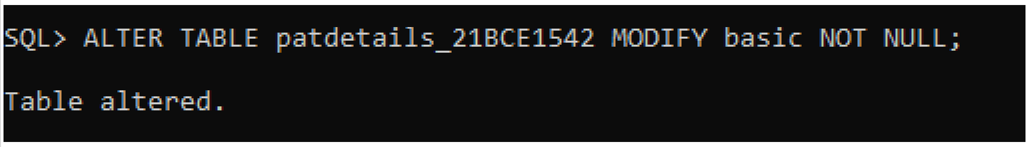
```
SQL> ALTER TABLE employee_21BCE1542 MODIFY city DEFAULT 'New York';

Table altered.
```

14 **Create NOT NULL for dept_id on department table.**



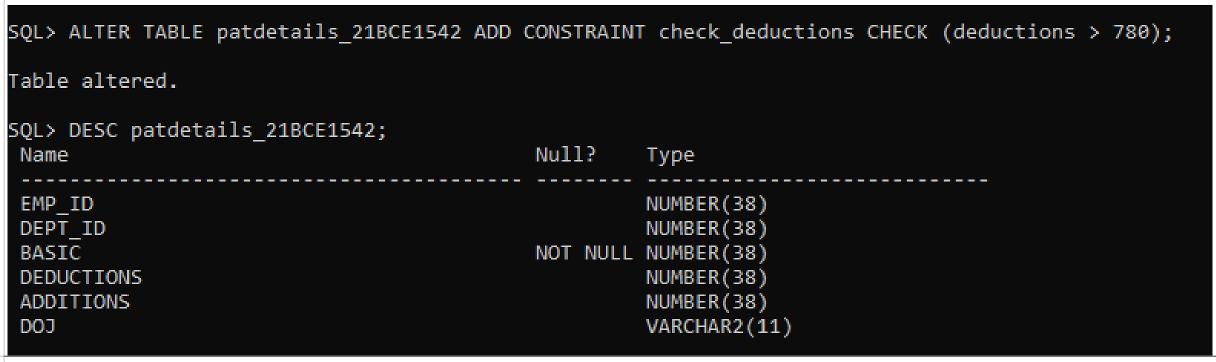
```
SQL> ALTER TABLE department_21BCE1542 MODIFY dept_id  NOT NULL;

Table altered.

SQL> DESC department_21BCE1542
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 DEPT_ID                                   NOT NULL NUMBER(38)
 DEPT_NAME                                          VARCHAR2(11)
```

15 **Create NOT NULL for basic in pay details.**



```
SQL> ALTER TABLE patdetails_21BCE1542 MODIFY basic NOT NULL;

Table altered.
```

16 **Enforce CHECK constraints for (deductions > 780) on pay details.**



```
SQL> ALTER TABLE patdetails_21BCE1542 ADD CONSTRAINT check_deductions CHECK (deductions > 780);

Table altered.

SQL> DESC patdetails_21BCE1542;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 EMP_ID                                             NUMBER(38)
 DEPT_ID                                            NUMBER(38)
 BASIC                                     NOT NULL NUMBER(38)
 DEDUCTIONS                                         NUMBER(38)
 ADDITIONS                                          NUMBER(38)
 DOJ                                                VARCHAR2(11)
```

10

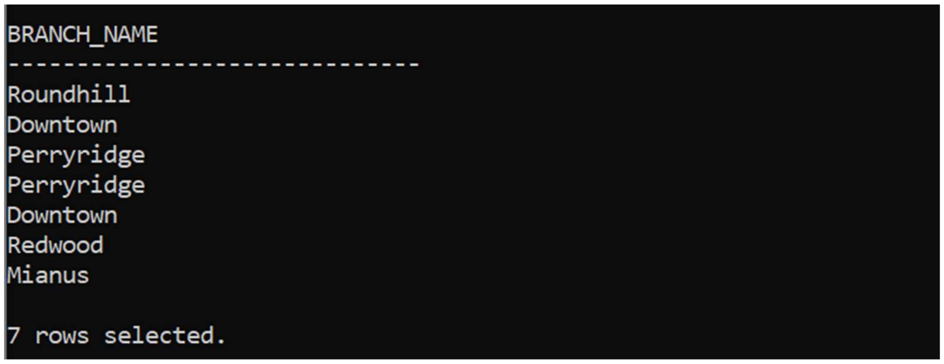
Ex. 2 Insert, Select Commands, Update & Delete Commands.

Aim : To perform the following queries using DML statements.

Retrieval operation-

Find the names of all branches in the loan relation with duplicates.

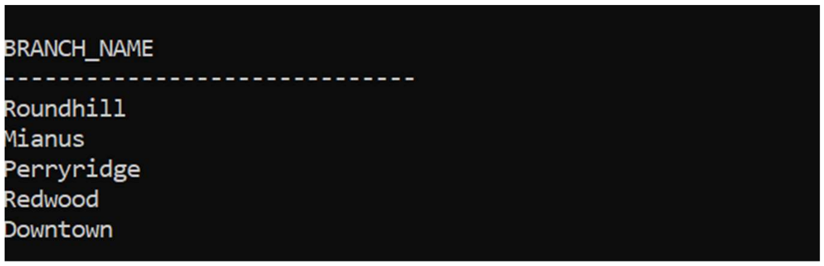
select branch_name from loan_21BCE1542;



```
BRANCH_NAME
------------------------------
Roundhill
Downtown
Perryridge
Perryridge
Downtown
Redwood
Mianus

7 rows selected.
```

Find branch names in the loan relation without duplicates.

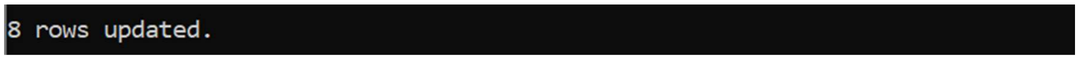
select distinct branch_name from loan_21BCE1542;



```
BRANCH_NAME
------------------------------
Roundhill
Mianus
Perryridge
Redwood
Downtown
```

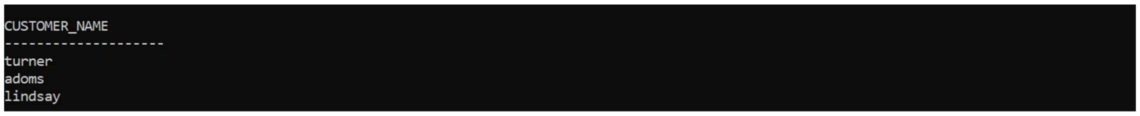
Modify the balance attribute alone such that it decreases the amount by 10% for the account table.

update account_21BCE1542 set balance = balance - (0.1 * balance);


```
8 rows updated.
```

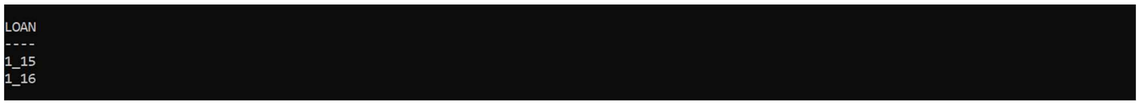
Display all the Customer names whose come from either pittsfield or stamford.

select customer_name from customer_21BCE1542 where customer_city = 'pittsfield' or customer_city = 'stamford';


```
CUSTOMER_NAME
--------------------
turner
adoms
lindsay
```

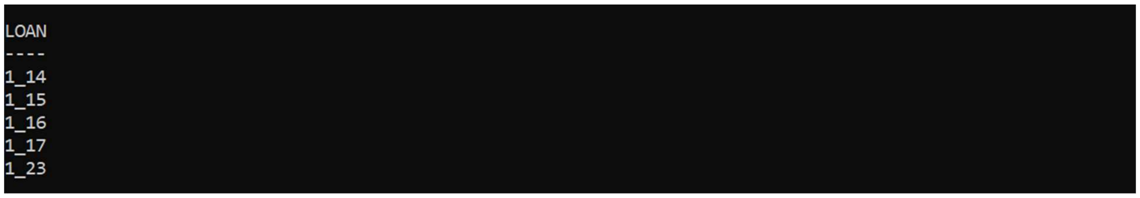
Find all loan numbers for loans made at the 'Perryridge' branch with loan amount greater than 1200.

select loan_no from loan_21BCE1542 where branch_name = 'Perryridge' and amount > 1200;


```
LOAN
----
1_15
1_16
```

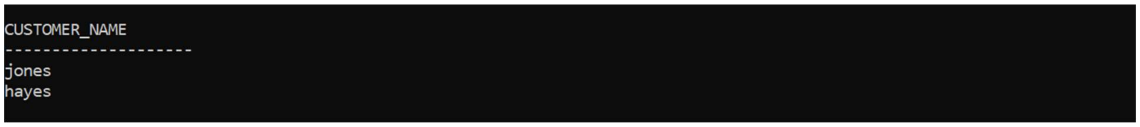
Find loan numbers of those loans with loan amount between 10000 and 20000.

select loan_no from loan_21BCE1542 where amount between 10000 and 20000;


```
LOAN
----
1_14
1_15
1_16
1_17
1_23
```

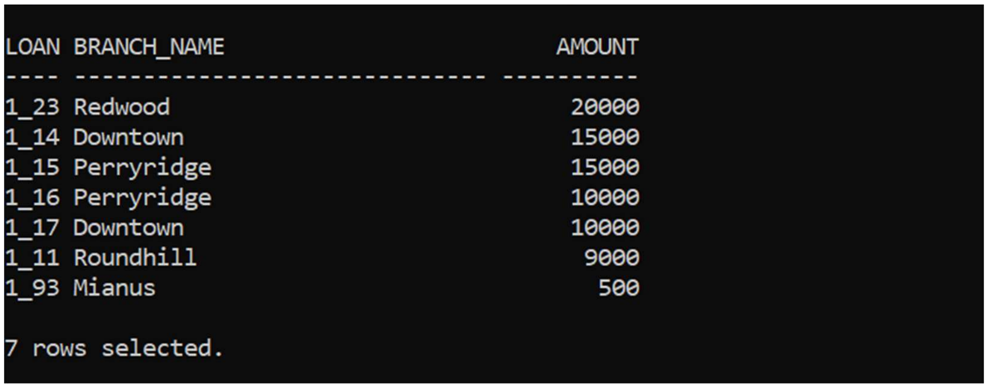
Find the names of all customers whose street address includes the substring 'Main'.

select customer_name from customer_21BCE1542 where customer_street like '%main%';


```
CUSTOMER_NAME
--------------------
jones
hayes
```

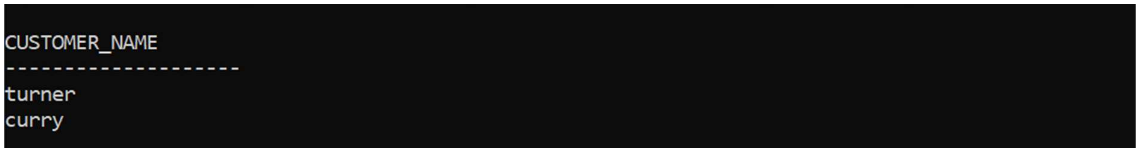
To list the entire loan relation in descending order of amount.

select * from loan_21BCE1542 order by amount desc;


```
LOAN BRANCH_NAME                         AMOUNT
---- ------------------------------ ----------
1_23 Redwood                             20000
1_14 Downtown                            15000
1_15 Perryridge                          15000
1_16 Perryridge                          10000
1_17 Downtown                            10000
1_11 Roundhill                            9000
1_93 Mianus                                500

7 rows selected.
```

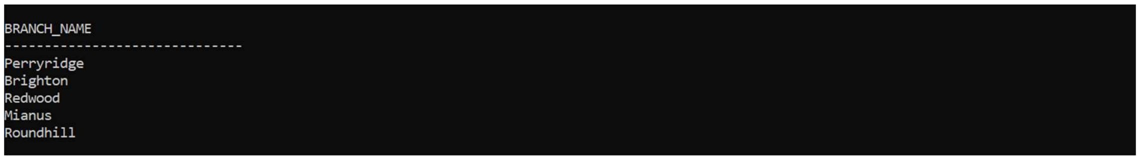
Find the names of the customer whose second letter with ‘u’.

select customer_name from customer_21BCE1542 where customer_name like '_u%';


```
CUSTOMER_NAME
--------------------
turner
curry
```

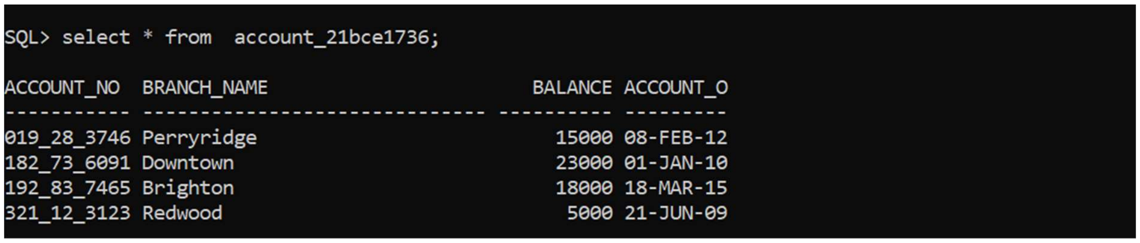
Find the names of all branches that have assets greater than at least one branch located in Stamford

select T.branch_name from branch_21BCE1542 T, branch_21BCE1542 S where T.assets > S.assets and S.branch_city = 'Stanmford';


```
BRANCH_NAME
------------------------------
Perryridge
Brighton
Redwood
Mianus
Roundhill
```

Display all the customer whose have account with more than10 yrs (add column account open date, if the column not present).

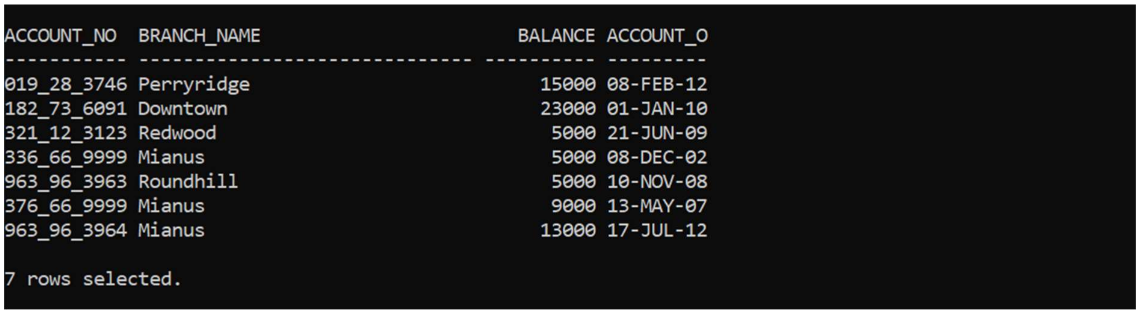
update account_21BCE1542 set account_opening_date='08-Feb-2012' where account_no='019_28_3746';
update account_21BCE1542 set account_opening_date='01-Jan-2010' where account_no='182_73_6091';
update account_21BCE1542 set account_opening_date='18-Mar-2015' where account_no='192_83_7465';
update account_21BCE1542 set account_opening_date='21-Jun-2009' where account_no='321_12_3123';
update account_21BCE1542 set account_opening_date='08-Dec-2002' where account_no='336_66_9999';
update account_21BCE1542 set account_opening_date='10-Nov-2008' where account_no='963_96_3963';
update account_21BCE1542 set account_opening_date='13-May-2007' where account_no='376_66_9999';
update account_21BCE1542 set account_opening_date='17-Jul-2012' where account_no='963_96_3964';



```
SQL> select * from  account_21bce1736;

ACCOUNT_NO  BRANCH_NAME                       BALANCE ACCOUNT_O
----------- ----------------------------- ---------- ---------
019_28_3746 Perryridge                          15000 08-FEB-12
182_73_6091 Downtown                            23000 01-JAN-10
192_83_7465 Brighton                            18000 18-MAR-15
321_12_3123 Redwood                              5000 21-JUN-09
```

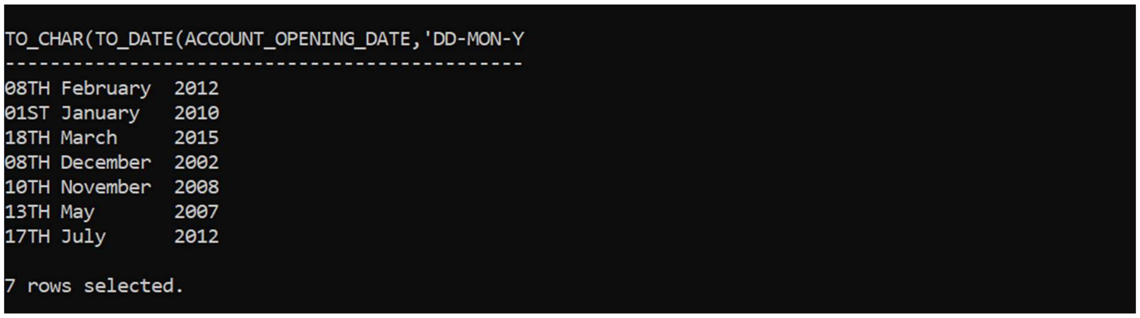
Display a date with following format" on 7th January 2001 at 5:30p.m".

select * from account_21BCE1542 where (sysdate-account_opening_date)/365.24>10;



```
ACCOUNT_NO  BRANCH_NAME                      BALANCE ACCOUNT_O
----------- ---------------------------- ---------- ---------
019_28_3746 Perryridge                        15000 08-FEB-12
182_73_6091 Downtown                          23000 01-JAN-10
321_12_3123 Redwood                            5000 21-JUN-09
336_66_9999 Mianus                             5000 08-DEC-02
963_96_3963 Roundhill                          5000 10-NOV-08
376_66_9999 Mianus                             9000 13-MAY-07
963_96_3964 Mianus                            13000 17-JUL-12

7 rows selected.
```

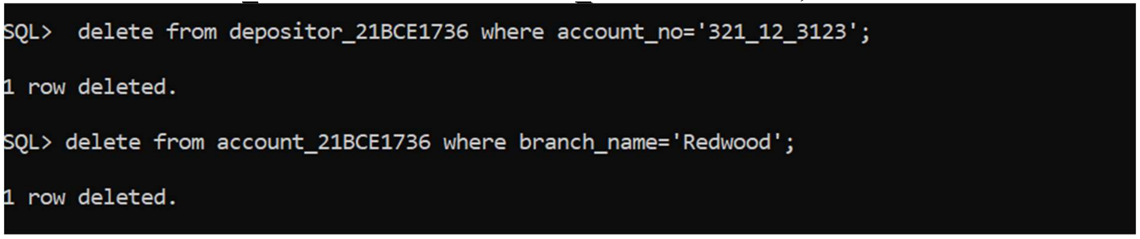
select to_char(to_date(account_opening_date, 'DD-MON-YY'), 'DDth Month YYYY') from
 2 account_21BCE1542;



```
TO_CHAR(TO_DATE(ACCOUNT_OPENING_DATE,'DD-MON-Y
----------------------------------------------
08TH February  2012
01ST January   2010
18TH March     2015
08TH December  2002
10TH November  2008
13TH May       2007
17TH July      2012

7 rows selected.
```

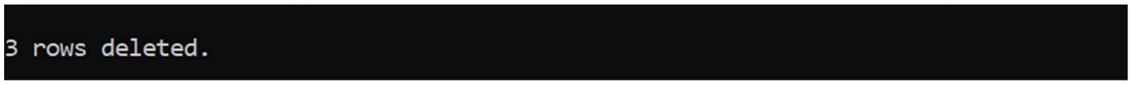
Delete all the account tuples in the ‘Redwood’ branch.

delete from depositor_21BCE1542 where account_no='321_12_3123';
delete from account_21BCE1542 where branch_name='Redwood';



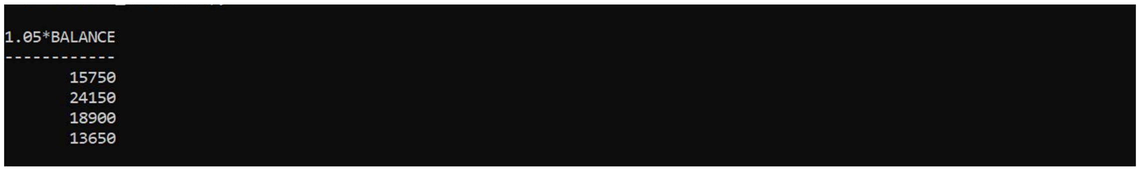
```
SQL>  delete from depositor_21BCE1736 where account_no='321_12_3123';

1 row deleted.

SQL> delete from account_21BCE1736 where branch_name='Redwood';

1 row deleted.
```

Delete all loans with loan amounts between 15000 to 20000.
delete from loan_21BCE1542 where amount between 15000 and 20000;



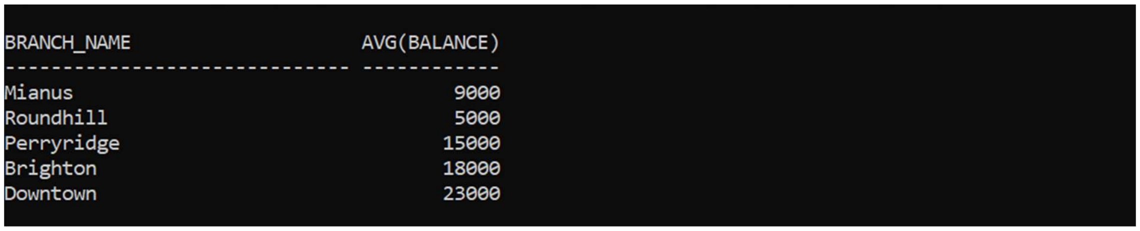
```
3 rows deleted.
```

select 1.05*balance from account_21BCE1542 where balance > (select avg(balance) from
  account_21BCE1542);

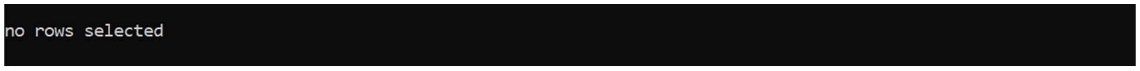

```
1.05*BALANCE
-----------
      15750
      24150
      18900
      13650
```

Built-In Functions

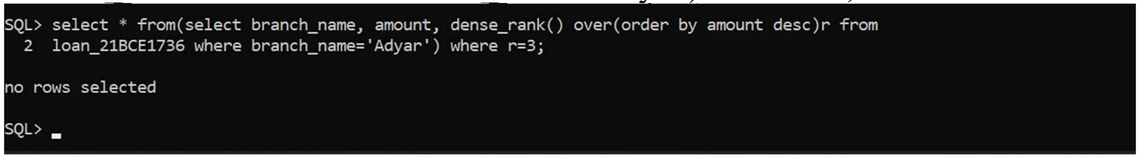
Find the average account balance at each branch
select branch_name, avg(balance) from account_21BCE1542 group
by branch_name;


```
BRANCH_NAME                   AVG(BALANCE)
----------------------------- ------------
Mianus                                9000
Roundhill                             5000
Perryridge                           15000
Brighton                             18000
Downtown                             23000
```

Select the customer city, which has more than 4 customers.
select customer_city, count(customer_city) from customer_21BCE1542 group by
customer_city
  2 having count(customer_city) > 4;

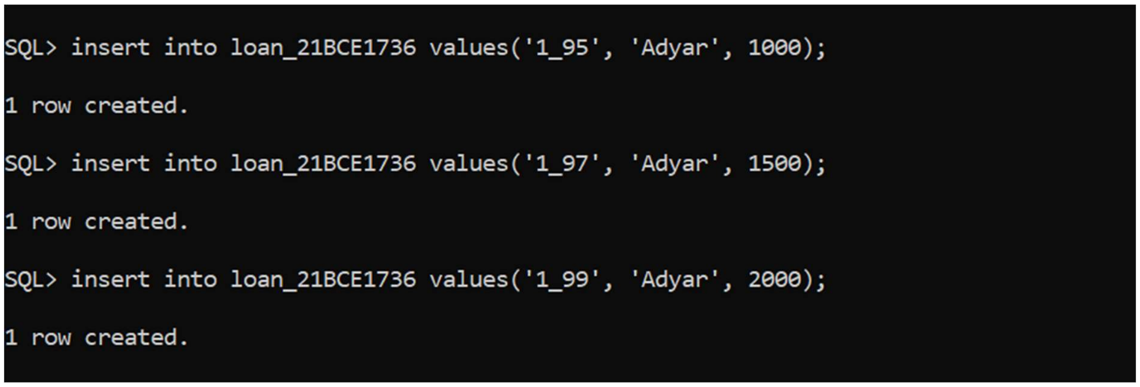

```
no rows selected
```

Retrieve the 3rd maximum amount in the Adyar branch.
select * from(select branch_name, amount, dense_rank() over(order by amount
desc)r from
 2 loan_21BCE1542 where branch_name='Adyar') where r=3;


```
SQL> select * from(select branch_name, amount, dense_rank() over(order by amount desc)r from
  2  loan_21BCE1736 where branch_name='Adyar') where r=3;

no rows selected

SQL> _
```

insert into loan_21BCE1542 values('1_95', 'Adyar', 1000);

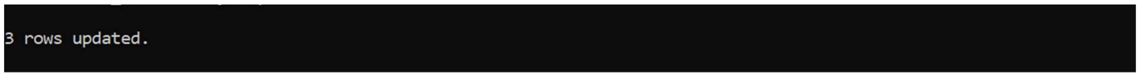
insert into loan_21BCE1542 values('1_97', 'Adyar', 1500);
insert into loan_21BCE1542 values('1_99', 'Adyar', 2000);



```
SQL> insert into loan_21BCE1736 values('1_95', 'Adyar', 1000);

1 row created.

SQL> insert into loan_21BCE1736 values('1_97', 'Adyar', 1500);

1 row created.

SQL> insert into loan_21BCE1736 values('1_99', 'Adyar', 2000);

1 row created.
```

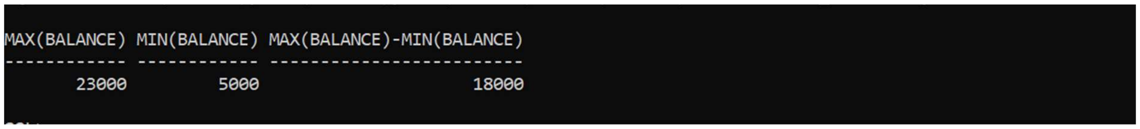
Use replace function and change the word adyar to Vadapalani.

update loan_21BCE1542 set branch_name = replace(branch_name, 'Adyar', 'Vadapalani') where
  2 branch_name = 'Adyar';



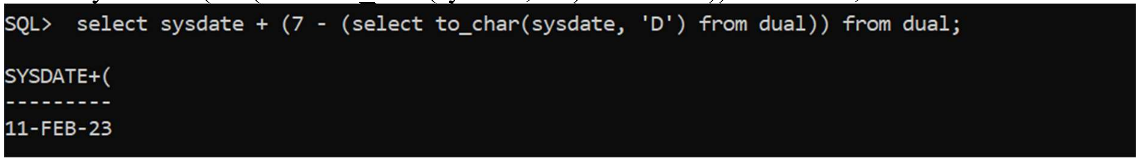
```
3 rows updated.
```

Find the highest balance, lowest balance and difference between both.
select max(balance), min(balance), max(balance) - min(balance) from account_21BCE1542;



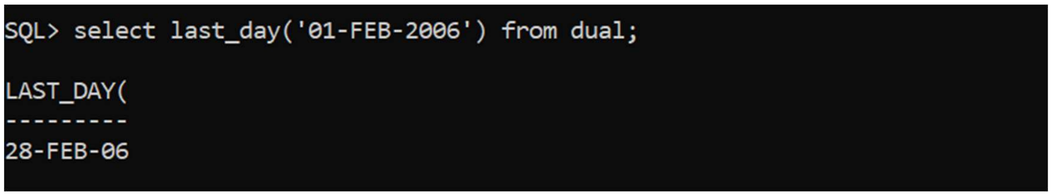
```
MAX(BALANCE) MIN(BALANCE) MAX(BALANCE)-MIN(BALANCE)
----------- ----------- ------------------------
      23000        5000                   18000
```

Display the day of next Saturday.
select sysdate + (7 - (select to_char(sysdate, 'D') from dual)) from dual;



```
SQL>  select sysdate + (7 - (select to_char(sysdate, 'D') from dual)) from dual;

SYSDATE+(
---------
11-FEB-23
```

Display the last date of Feb 2006.
select last_day('01-FEB-2006') from dual;



```
SQL> select last_day('01-FEB-2006') from dual;

LAST_DAY(
---------
28-FEB-06
```

Display the word as " ****welcome".

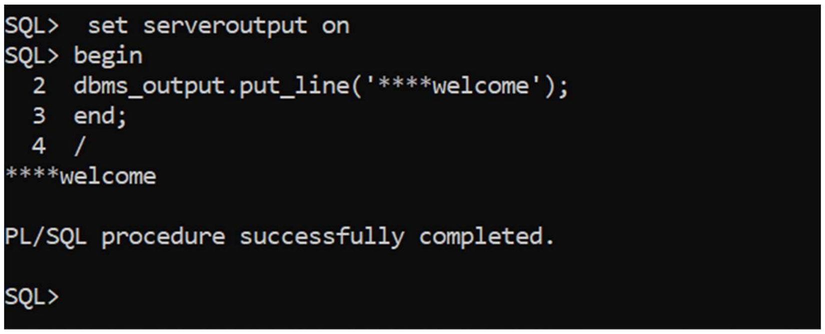
set serveroutput on

begin

dbms_output.put_line('****welcome');

 end;

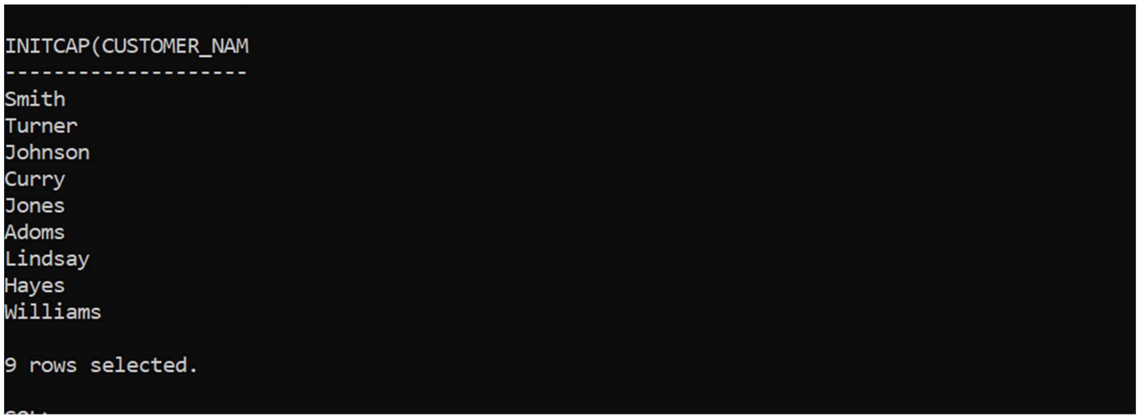
/



```
SQL>  set serveroutput on
SQL> begin
  2  dbms_output.put_line('****welcome');
  3  end;
  4  /
****welcome

PL/SQL procedure successfully completed.

SQL>
```

Display the customer names with first letter in capital. select

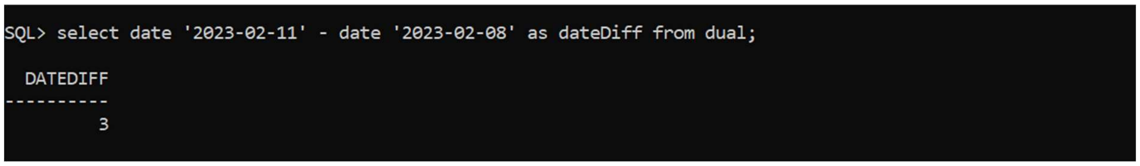
initcap(customer_name) from customer_21BCE1542;



```
INITCAP(CUSTOMER_NAM
--------------------
Smith
Turner
Johnson
Curry
Jones
Adoms
Lindsay
Hayes
Williams

9 rows selected.
```

Count the number of days present between today and Sunday.

select date '2023-02-11' - date '2023-02-08' as dateDiff from dual;



```
SQL> select date '2023-02-11' - date '2023-02-08' as dateDiff from dual;

  DATEDIFF
----------
         3
```

Change the name of employee “AAA” to “BBB”.

alter table employee_21BCE1542 add DOB date;

insert into employee_21BCE1542 values(1, 'John', 'Palk Street', 'Mumbai', '25-Feb-2006');

insert into employee_21BCE1542 values(2, 'Brendon', 'Church Street', 'Bangalore', '22-Jan-1996');

insert into employee_21BCE1542 values(3, 'Blenda', 'Connaught Palace', 'Delhi', '28-Aug-1987');

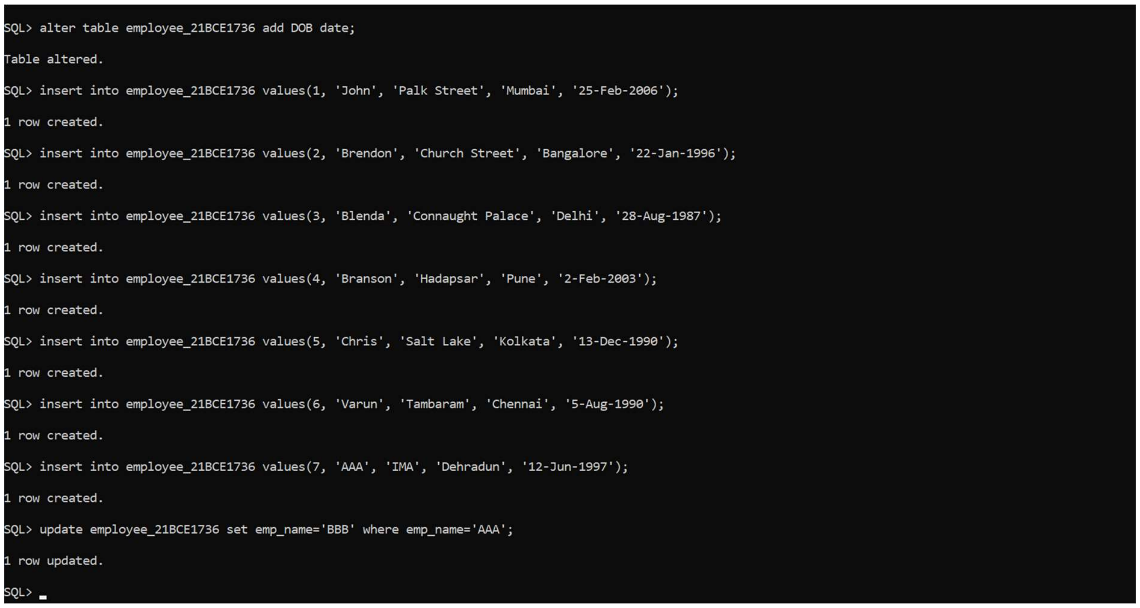
insert into employee_21BCE1542 values(4, 'Branson', 'Hadapsar', 'Pune', '2-Feb-2003');

insert into employee_21BCE1542 values(5, 'Chris', 'Salt Lake', 'Kolkata', '13-Dec-1990');

insert into employee_21BCE1542 values(6, 'Varun', 'Tambaram', 'Chennai', '5-Aug-1990');

insert into employee_21BCE1542 values(7, 'AAA', 'IMA', 'Dehradun', '12-Jun-1997');

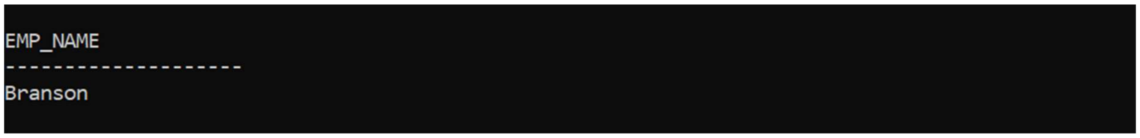
update employee_21BCE1542 set emp_name='BBB' where emp_name='AAA';



```
SQL> alter table employee_21BCE1736 add DOB date;

Table altered.

SQL> insert into employee_21BCE1736 values(1, 'John', 'Palk Street', 'Mumbai', '25-Feb-2006');

1 row created.

SQL> insert into employee_21BCE1736 values(2, 'Brendon', 'Church Street', 'Bangalore', '22-Jan-1996');

1 row created.

SQL> insert into employee_21BCE1736 values(3, 'Blenda', 'Connaught Palace', 'Delhi', '28-Aug-1987');

1 row created.

SQL> insert into employee_21BCE1736 values(4, 'Branson', 'Hadapsar', 'Pune', '2-Feb-2003');

1 row created.

SQL> insert into employee_21BCE1736 values(5, 'Chris', 'Salt Lake', 'Kolkata', '13-Dec-1990');

1 row created.

SQL> insert into employee_21BCE1736 values(6, 'Varun', 'Tambaram', 'Chennai', '5-Aug-1990');

1 row created.

SQL> insert into employee_21BCE1736 values(7, 'AAA', 'IMA', 'Dehradun', '12-Jun-1997');

1 row created.

SQL> update employee_21BCE1736 set emp_name='BBB' where emp_name='AAA';

1 row updated.

SQL>
```

Using LIKE operator list the enames starting with B and third character with A.

select emp_name from employee_21BCE1542 where emp_name like 'B_a%';

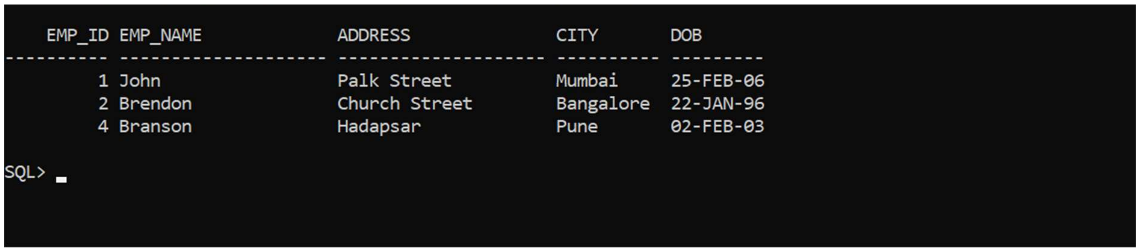


```
EMP_NAME
--------------------
Branson
```

List the customer who are 24 years and above and those who are born in the month of jan or feb or mar.

select * from employee_21BCE1542 where (sysdate-dob)/365.24 > 24 and to_char(dob,'mon')='jan'

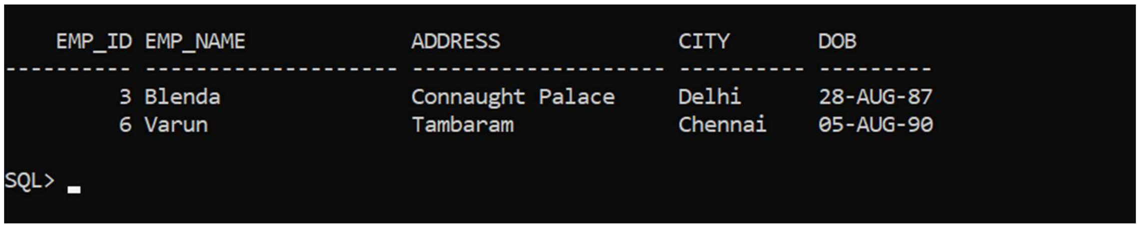
or to_char(dob,'mon')='feb' or to_char(dob,'mon')='mar';



```
    EMP_ID EMP_NAME             ADDRESS              CITY       DOB
---------- -------------------- -------------------- ---------- ---------
         1 John                 Palk Street          Mumbai     25-FEB-06
         2 Brendon              Church Street        Bangalore  22-JAN-96
         4 Branson              Hadapsar             Pune       02-FEB-03

SQL>
```

Display all the employees who are born in the month of August.

select * from employee_21BCE1542 where to_char(dob,'mon') = 'aug';

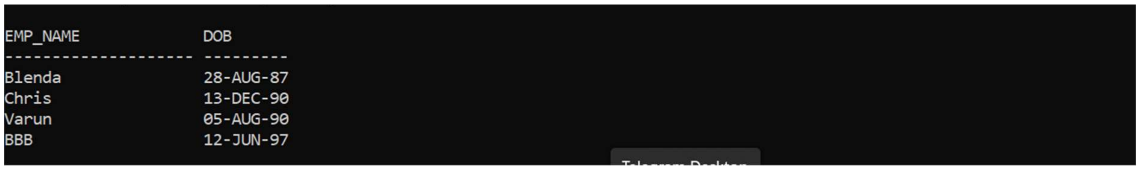


```
   EMP_ID EMP_NAME             ADDRESS              CITY       DOB
---------- -------------------- -------------------- ---------- ---------
         3 Blenda               Connaught Palace     Delhi      28-AUG-87
         6 Varun                Tambaram             Chennai    05-AUG-90

SQL> _
```

List all employees who are born between 1st May and 31st Dec of any year.

select emp_name, dob from employee_21BCE1542 where extract(month from dob) > 5 and

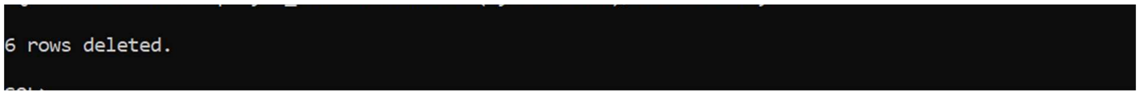
extract(month from dob) <= 12;



```
EMP_NAME             DOB
-------------------- ---------
Blenda               28-AUG-87
Chris                13-DEC-90
Varun                05-AUG-90
BBB                  12-JUN-97
```

Delete all the employees whose age is above 20 yrs.

delete from employee_21BCE1542 where (sysdate-dob)/365.24 > 20;

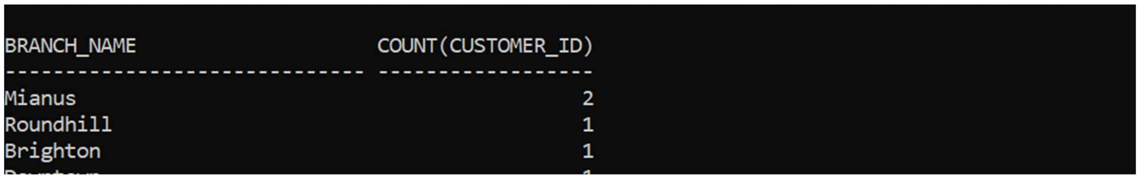


```
6 rows deleted.
```

Group By clause

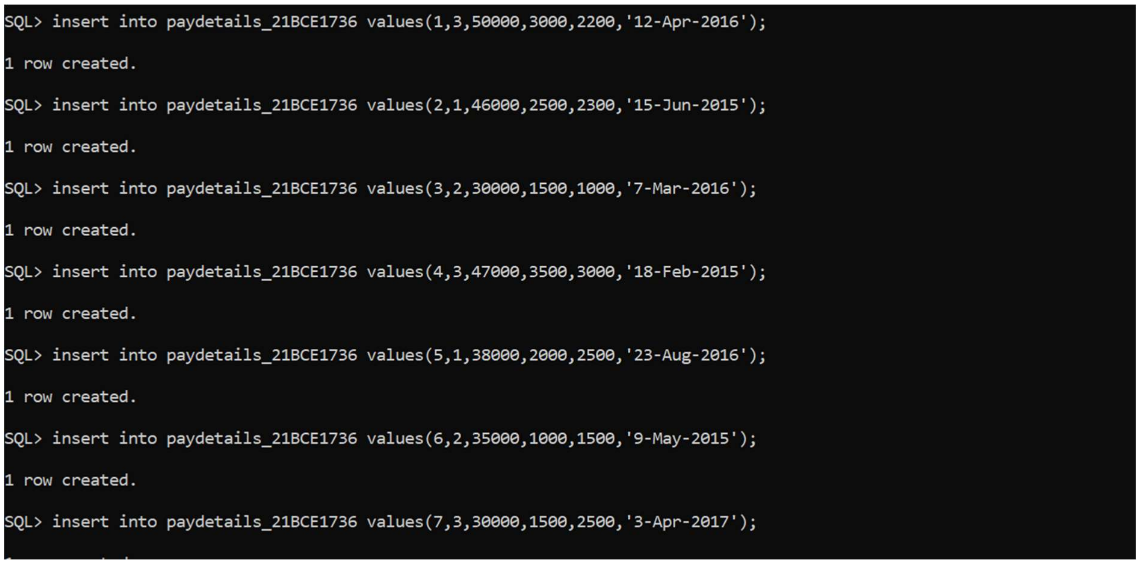
Find the number of the depositors for each branch.

select branch_name, count(customer_id) from depositor_21BCE1542, account_21BCE1542 where
 depositor_21BCE1542.account_no = account_21BCE1542.account_no group by branch_name;


```
BRANCH_NAME                   COUNT(CUSTOMER_ID)
----------------------------- ------------------
Mianus                                         2
Roundhill                                      1
Brighton                                       1
```

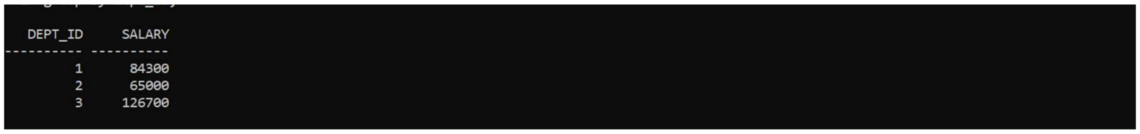
Find the total salary of a department.

insert into paydetails_21BCE1542 values(1,3,50000,3000,2200,'12-Apr-2016'); insert into paydetails_21BCE1542 values(2,1,46000,2500,2300,'15-Jun-2015'); insert into paydetails_21BCE1542 values(3,2,30000,1500,1000,'7-Mar-2016'); insert into paydetails_21BCE1542 values(4,3,47000,3500,3000,'18-Feb-2015'); insert into paydetails_21BCE1542 values(5,1,38000,2000,2500,'23-Aug-2016'); insert into paydetails_21BCE1542 values(6,2,35000,1000,1500,'9-May-2015'); insert into paydetails_21BCE1542 values(7,3,30000,1500,2500,'3-Apr-2017');


```
SQL> insert into paydetails_21BCE1736 values(1,3,50000,3000,2200,'12-Apr-2016');

1 row created.

SQL> insert into paydetails_21BCE1736 values(2,1,46000,2500,2300,'15-Jun-2015');

1 row created.

SQL> insert into paydetails_21BCE1736 values(3,2,30000,1500,1000,'7-Mar-2016');

1 row created.

SQL> insert into paydetails_21BCE1736 values(4,3,47000,3500,3000,'18-Feb-2015');

1 row created.

SQL> insert into paydetails_21BCE1736 values(5,1,38000,2000,2500,'23-Aug-2016');

1 row created.

SQL> insert into paydetails_21BCE1736 values(6,2,35000,1000,1500,'9-May-2015');

1 row created.

SQL> insert into paydetails_21BCE1736 values(7,3,30000,1500,2500,'3-Apr-2017');
```

select dept_id, sum(basic) + sum(additions) - sum(deductions) as salary
from paydetails_21BCE1542

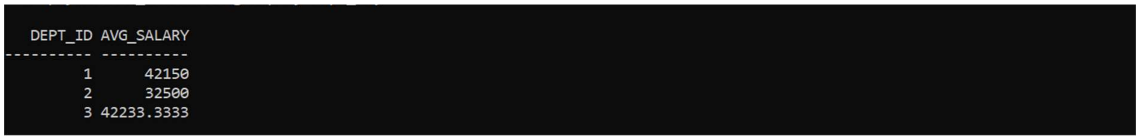
group by dept_id;


```
  DEPT_ID     SALARY
---------- ----------
         1      84300
         2      65000
         3     126700
```

Find the average salary of an employee for each department numberwise.

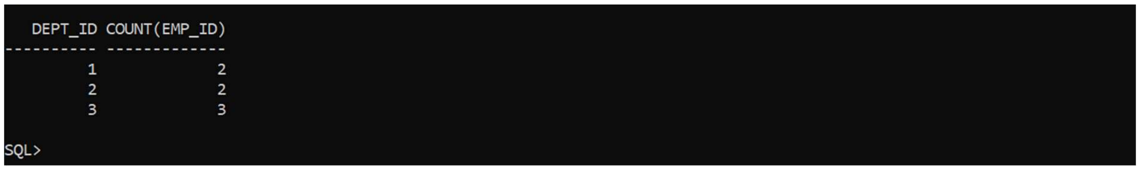
select dept_id, avg(basic) + avg(additions) - avg(deductions) as avg_salary

from paydetails_21BCE1542 group by dept_id;


```
   DEPT_ID AVG_SALARY
---------- ----------
         1      42150
         2      32500
         3 42233.3333
```

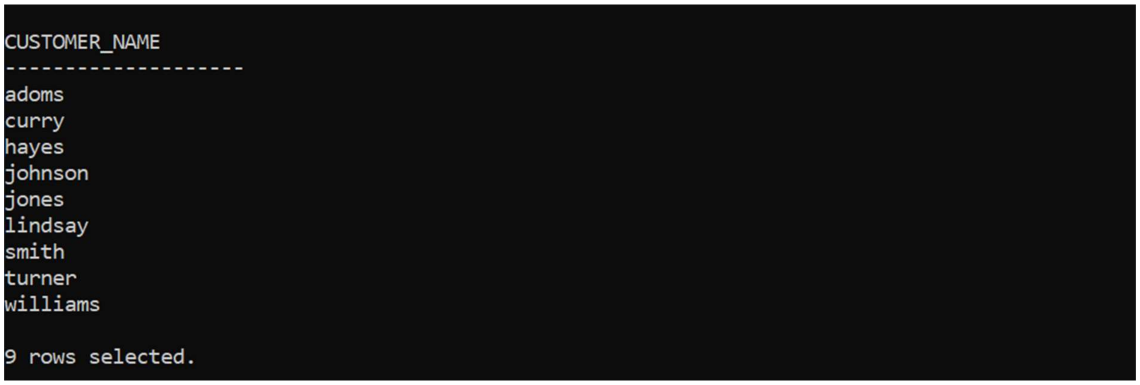
Find the total number of persons working from the employee table and also group by deptnumberwise

select dept_id, count(emp_id) from paydetails_21BCE1542 group by dept_id;


```
   DEPT_ID COUNT(EMP_ID)
---------- ------------
         1            2
         2            2
         3            3

SQL>
```

Order By clause

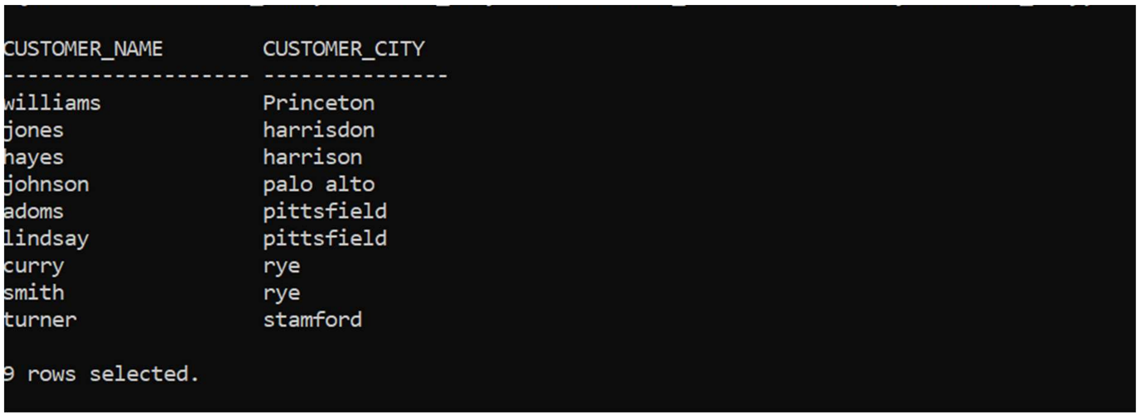
Display the customer name in alphabetical order.
select customer_name from customer_21BCE1542 order by customer_name;



```
CUSTOMER_NAME
--------------------
adoms
curry
hayes
johnson
jones
lindsay
smith
turner
williams

9 rows selected.
```

Display all the customer names ordered by customer city

select customer_name, customer_city from customer_21BCE1542 order by customer_city;

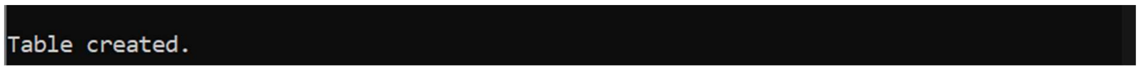


```
CUSTOMER_NAME        CUSTOMER_CITY
-------------------- ---------------
williams             Princeton
jones                harrisdon
hayes                harrison
johnson              palo alto
adoms                pittsfield
lindsay              pittsfield
curry                rye
smith                rye
turner               stamford

9 rows selected.
```

LAB 3

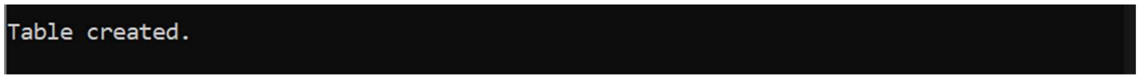
create table flight_21BCE1542(no INT primary key, frm VARCHAR(20), too varchar(20), distance INT, departs VARCHAR(20), arrives VARCHAR(20), price int);

alter table flight_21BCE1542 add (too varchar(20));



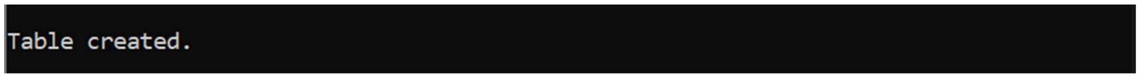
```
Table created.
```

create table aircraft_21bce1542(aid INT primary key, aname VARCHAR(20), cruisingrange INT);



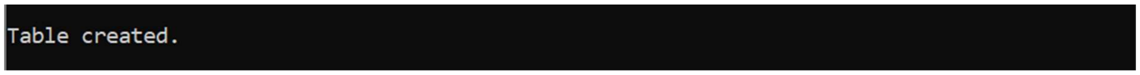
```
Table created.
```

create table employees_21bce1542(eid int primary key, ename varchar(20), salary int);



```
Table created.
```

create table certified_21bce1542(eid int, aid int, foreign key(eid) references employees_21BCE1542(eid), foreign key(aid) references aircraft_21BCE1542(aid));



```
Table created.
```

INSERT INTO flight_21bce1542(no,frm,too,distance,departs,arrives,price)

VALUES(1,'Bangalore','Mangalore',360,'10:45:00','12:00:00',10000);

```
1 row created.

SQL> _
```

INSERT INTO flight_21bce1542 (no,frm,too,distance,departs,arrives,price)
VALUES(2,'Bangalore','Delhi',5000,'12:15:00','04:30:00',25000);

```
1 row created.

SQL> _
```

INSERT INTO flight_21bce1542 (no,frm,too,distance,departs,arrives,price)

VALUES(3,'Bangalore','Mumbai',3500,'02:15:00','05:25:00',30000);

```
1 row created.

SQL> _
```

INSERT INTO flight_21BCE1542 (no,frm,too,distance,departs,arrives,price)

VALUES(4,'Delhi','Mumbai',4500,'10:15:00','12:05:00',35000);

INSERT INTO flight_21BCE1542 (no,frm,too,distance,departs,arrives,price)

VALUES(4,'Delhi','Mumbai',4500,'10:15:00','12:05:00',35000);

INSERT INTO flight_21BCE1542 (no,frm,too,distance,departs,arrives,price)

VALUES(6,'Bangalore','Frankfurt',19500,'10:00:00','07:45:00',95000);

INSERT INTO flight_21BCE1542 (no,frm,too,distance,departs,arrives,price)

VALUES(7,'Bangalore','Frankfurt',17000,'12:00:00','06:30:00',99000);

INSERT INTO aircraft_21BCE1542(aid,aname,cruisingrange) values(123,'Airbus',1000);

INSERT INTO aircraft_21BCE1542(aid,aname,cruisingrange) values(302,'Boeing',5000);

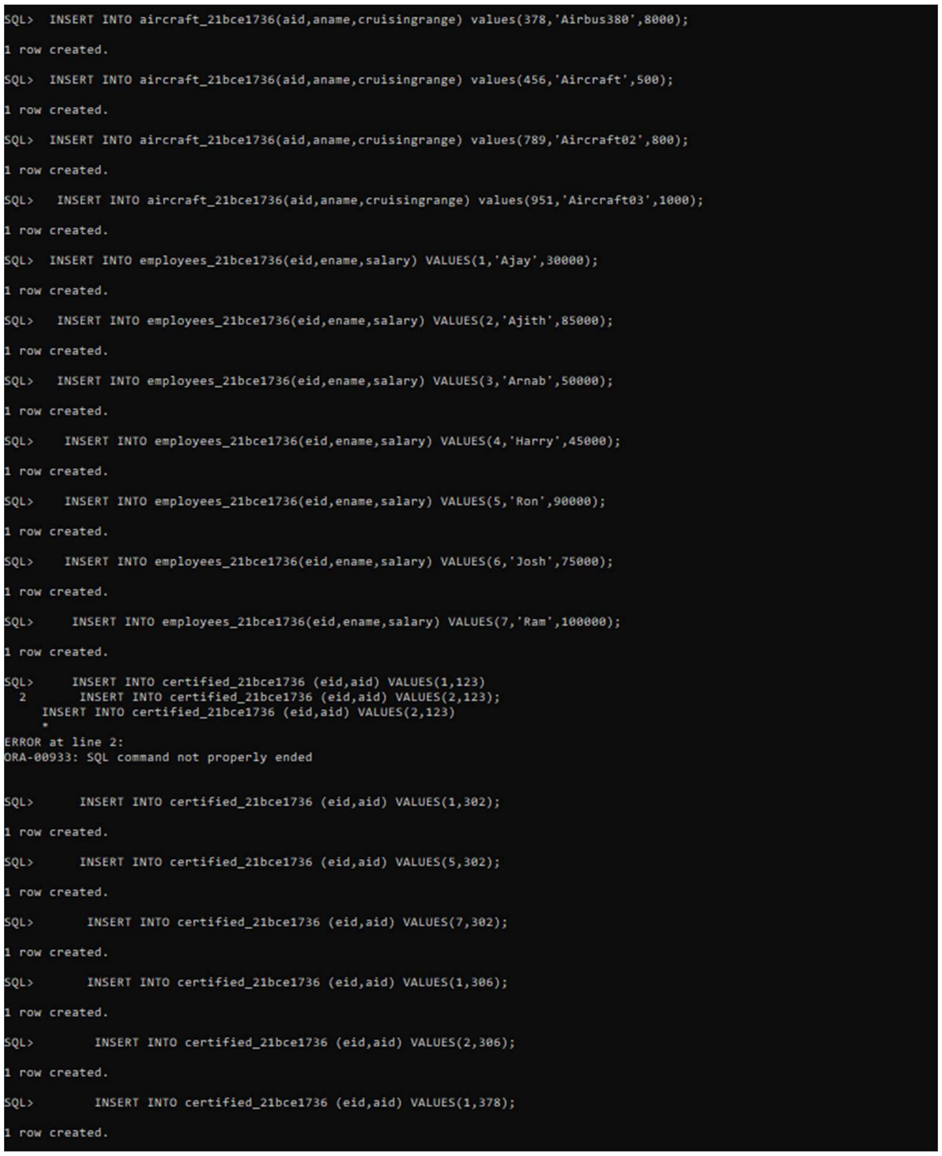
INSERT INTO aircraft_21BCE1542(aid,aname,cruisingrange) values(306,'Jet01',5000);

INSERT INTO aircraft_21BCE1542(aid,aname,cruisingrange)

values(378,'Airbus380',8000); INSERT INTO

aircraft_21BCE1542(aid,aname,cruisingrange) values(456,'Aircraft',500);

```
INSERT INTO aircraft_21BCE1542(aid,aname,cruisingrange) values(789,'Aircraft02',800);   INSERT INTO
aircraft_21BCE1542(aid,aname,cruisingrange) values(951,'Aircraft03',1000);  INSERT INTO
employees_21BCE1542(eid,ename,salary) VALUES(1,'Ajay',30000);
INSERT INTO employees_21BCE1542(eid,ename,salary) VALUES(2,'Ajith',85000);  INSERT INTO
employees_21BCE1542(eid,ename,salary) VALUES(3,'Arnab',50000);
INSERT INTO employees_21BCE1542(eid,ename,salary) VALUES(4,'Harry',45000);  INSERT INTO
employees_21BCE1542(eid,ename,salary) VALUES(5,'Ron',90000);  INSERT INTO
employees_21BCE1542(eid,ename,salary) VALUES(6,'Josh',75000);  INSERT INTO
employees_21BCE1542(eid,ename,salary) VALUES(7,'Ram',100000);  INSERT INTO certified_21BCE1542 (eid,aid)
VALUES(1,123);
INSERT INTO certified_21BCE1542 (eid,aid) VALUES(2,123);  INSERT INTO certified_21BCE1542 (eid,aid)
VALUES(1,302);  INSERT INTO certified_21BCE1542 (eid,aid) VALUES(5,302);  INSERT INTO
certified_21BCE1542 (eid,aid) VALUES(7,302);  INSERT INTO certified_21BCE1542 (eid,aid) VALUES(1,306);


INSERT INTO certified_21BCE1542 (eid,aid) VALUES(2,306);
INSERT INTO certified_21BCE1542 (eid,aid) VALUES(1,378);
INSERT INTO certified_21BCE1542 (eid,aid) VALUES(2,378);
INSERT INTO certified_21BCE1542 (eid,aid) VALUES(4,378);
INSERT INTO certified_21BCE1542 (eid,aid) VALUES(6,456);
INSERT INTO certified_21BCE1542 (eid,aid) VALUES(3,456);
INSERT INTO certified_21BCE1542 (eid,aid) VALUES(5,789);
INSERT  INTO  certified_21BCE1542  (eid,aid)  VALUES(6,789);    INSERT  INTO
certified_21BCE1542      (eid,aid)      VALUES(3,951);           INSERT      INTO
certified_21BCE1542      (eid,aid)      VALUES(1,951);           INSERT      INTO
certified_21BCE1542 (eid,aid) VALUES(1,789);
```



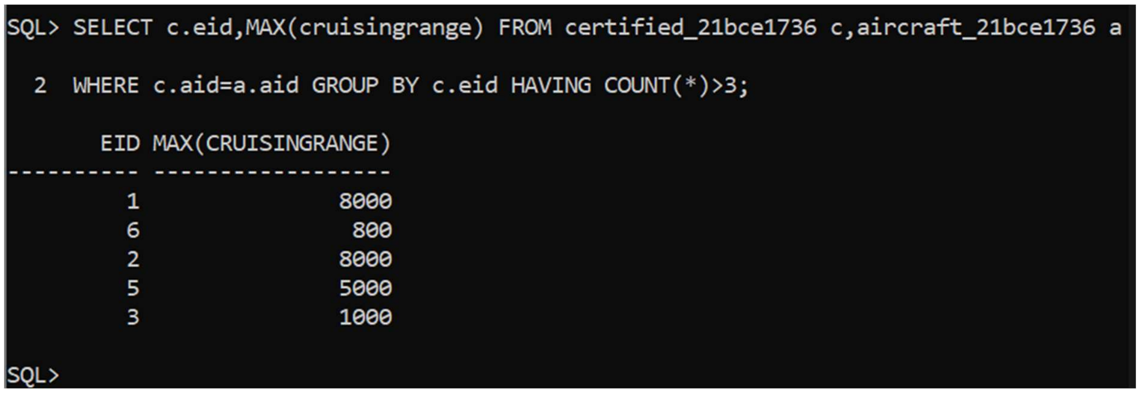
```
SQL>  INSERT INTO aircraft_21bce1736(aid,aname,cruisingrange) values(378,'Airbus380',8000);

1 row created.

SQL>  INSERT INTO aircraft_21bce1736(aid,aname,cruisingrange) values(456,'Aircraft',500);

1 row created.

SQL>  INSERT INTO aircraft_21bce1736(aid,aname,cruisingrange) values(789,'Aircraft02',800);

1 row created.

SQL>   INSERT INTO aircraft_21bce1736(aid,aname,cruisingrange) values(951,'Aircraft03',1000);

1 row created.

SQL>  INSERT INTO employees_21bce1736(eid,ename,salary) VALUES(1,'Ajay',30000);

1 row created.

SQL>   INSERT INTO employees_21bce1736(eid,ename,salary) VALUES(2,'Ajith',85000);

1 row created.

SQL>   INSERT INTO employees_21bce1736(eid,ename,salary) VALUES(3,'Arnab',50000);

1 row created.

SQL>    INSERT INTO employees_21bce1736(eid,ename,salary) VALUES(4,'Harry',45000);

1 row created.

SQL>    INSERT INTO employees_21bce1736(eid,ename,salary) VALUES(5,'Ron',90000);

1 row created.

SQL>    INSERT INTO employees_21bce1736(eid,ename,salary) VALUES(6,'Josh',75000);

1 row created.

SQL>     INSERT INTO employees_21bce1736(eid,ename,salary) VALUES(7,'Ram',100000);

1 row created.

SQL>     INSERT INTO certified_21bce1736 (eid,aid) VALUES(1,123)
  2       INSERT INTO certified_21bce1736 (eid,aid) VALUES(2,123);
    INSERT INTO certified_21bce1736 (eid,aid) VALUES(2,123)
    *
ERROR at line 2:
ORA-00933: SQL command not properly ended


SQL>      INSERT INTO certified_21bce1736 (eid,aid) VALUES(1,302);

1 row created.

SQL>      INSERT INTO certified_21bce1736 (eid,aid) VALUES(5,302);

1 row created.

SQL>       INSERT INTO certified_21bce1736 (eid,aid) VALUES(7,302);

1 row created.

SQL>       INSERT INTO certified_21bce1736 (eid,aid) VALUES(1,306);

1 row created.

SQL>        INSERT INTO certified_21bce1736 (eid,aid) VALUES(2,306);

1 row created.

SQL>        INSERT INTO certified_21bce1736 (eid,aid) VALUES(1,378);

1 row created.
```

Write each of the following queries in SQL.

1. For each pilot who is certified for more than three aircraft, find the eid and the maximum cruising range of the aircraft for which she or he is certified.

SELECT c.eid,MAX(cruisingrange) FROM certified_21BCE1542 c,aircraft_21BCE1542

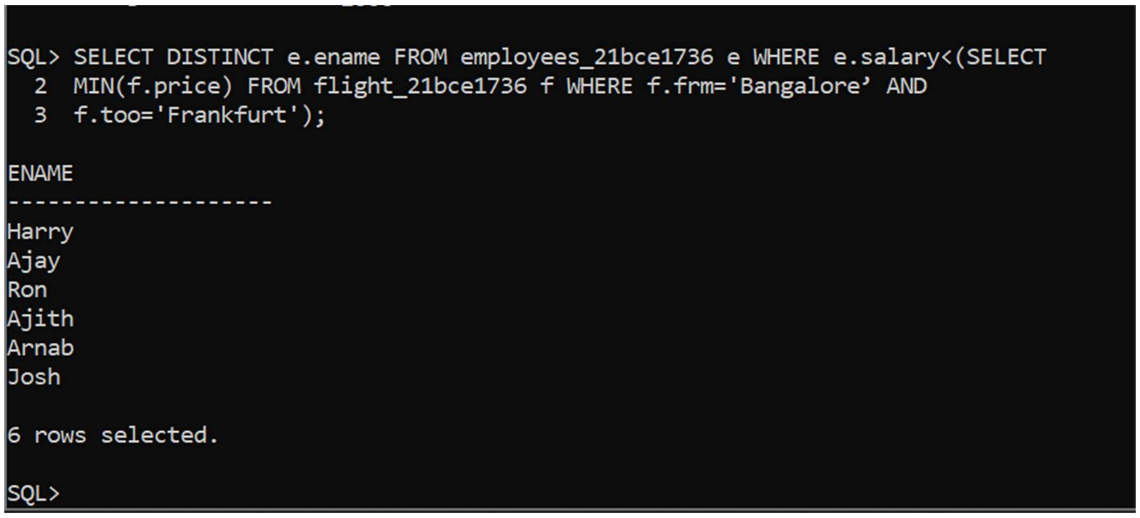
a WHERE c.aid=a.aid GROUP BY c.eid HAVING COUNT(*)>3;



```
SQL> SELECT c.eid,MAX(cruisingrange) FROM certified_21bce1736 c,aircraft_21bce1736 a

  2  WHERE c.aid=a.aid GROUP BY c.eid HAVING COUNT(*)>3;

       EID MAX(CRUISINGRANGE)
---------- ------------------
         1               8000
         6                800
         2               8000
         5               5000
         3               1000

SQL>
```

2. Find the names of pilots whose salary is less than the price of the cheapest route from BANGLORE to FRANKFURT.

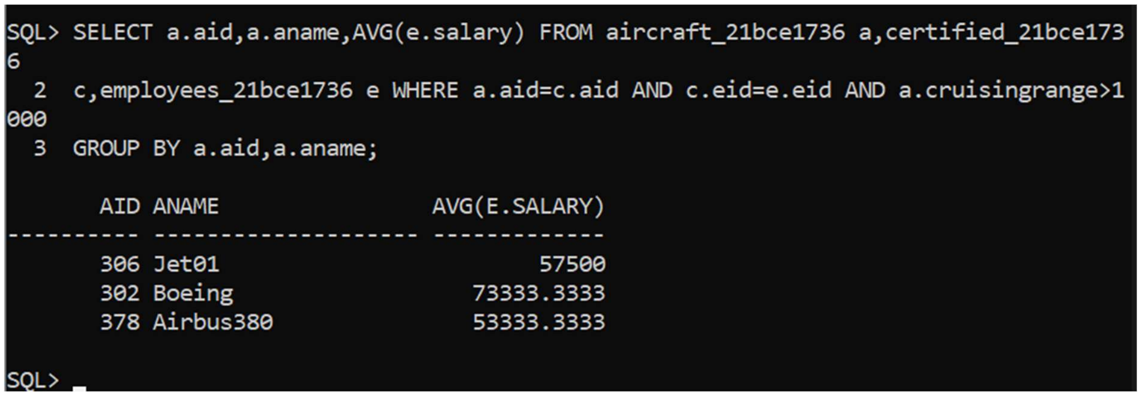
   SELECT c.eid,MAX(cruisingrange) FROM certified_21BCE1542 c,aircraft_21BCE1542 a WHERE c.aid=a.aid GROUP BY c.eid HAVING COUNT(*)>3;



```
SQL> SELECT DISTINCT e.ename FROM employees_21bce1736 e WHERE e.salary<(SELECT
  2  MIN(f.price) FROM flight_21bce1736 f WHERE f.frm='Bangalore’ AND
  3  f.too='Frankfurt');

ENAME
--------------------
Harry
Ajay
Ron
Ajith
Arnab
Josh

6 rows selected.

SQL>
```

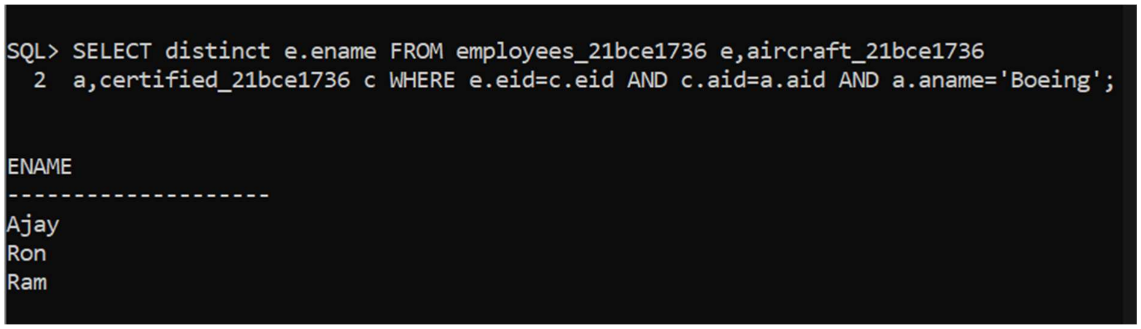
3. For all aircraft with cruising range over 1000 miles, find the name of the aircraft and the average salary of all pilots certified for this aircraft.

   SELECT c.eid,MAX(cruisingrange) FROM certified_21BCE1542 c,aircraft_21BCE1542 a WHERE c.aid=a.aid GROUP BY c.eid HAVING COUNT(*)>3;



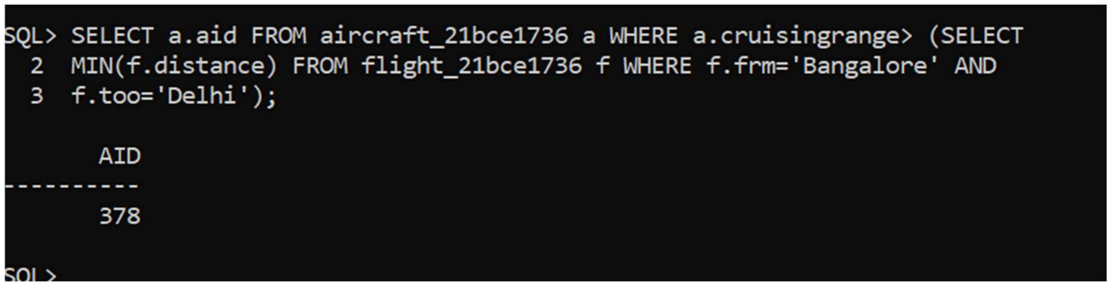
```
SQL> SELECT a.aid,a.aname,AVG(e.salary) FROM aircraft_21bce1736 a,certified_21bce173
6
  2  c,employees_21bce1736 e WHERE a.aid=c.aid AND c.eid=e.eid AND a.cruisingrange>1
000
  3  GROUP BY a.aid,a.aname;

       AID ANAME                AVG(E.SALARY)
---------- -------------------- -------------
       306 Jet01                        57500
       302 Boeing                  73333.3333
       378 Airbus380               53333.3333

SQL> _
```

4. Find the names of pilots certified for some Boeing aircraft.
SELECT distinct e.ename FROM employees_21BCE1542 e,aircraft_21BCE1542 a,certified_21BCE1542 c WHERE e.eid=c.eid AND c.aid=a.aid AND



```
SQL> SELECT distinct e.ename FROM employees_21bce1736 e,aircraft_21bce1736
  2  a,certified_21bce1736 c WHERE e.eid=c.eid AND c.aid=a.aid AND a.aname='Boeing';


ENAME
--------------------
Ajay
Ron
Ram
```

5. Find the aids of all aircraft that can be used on routes from Los Angeles to Chicago.
SELECT a.aid FROM aircraft_21BCE1542 a WHERE a.cruisingrange>
(SELECT MIN(f.distance) FROM flight_21BCE1542 f WHERE
f.frm='Bangalore' AND f.too='Delhi');

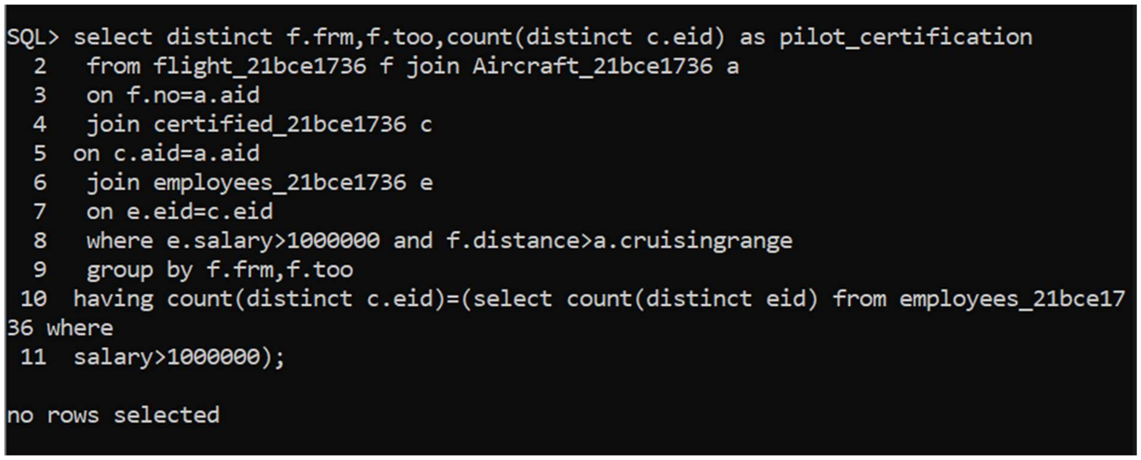


```
SQL> SELECT a.aid FROM aircraft_21bce1736 a WHERE a.cruisingrange> (SELECT
  2  MIN(f.distance) FROM flight_21bce1736 f WHERE f.frm='Bangalore' AND
  3  f.too='Delhi');

       AID
----------
       378

SQL>
```

6. Identify the routes that can be piloted by every pilot who makes more than $100,000

select distinct f.frm,f.too,count(distinct c.eid) as
pilot_certification from flight_21BCE1542 f join
Aircraft_21BCE1542 a
on f.no=a.aid
join certified_21BCE1542 c
on c.aid=a.aid
join employees_21BCE1542 e
on e.eid=c.eid
where e.salary>1000000 and f.distance>a.cruisingrange
group by f.frm,f.too

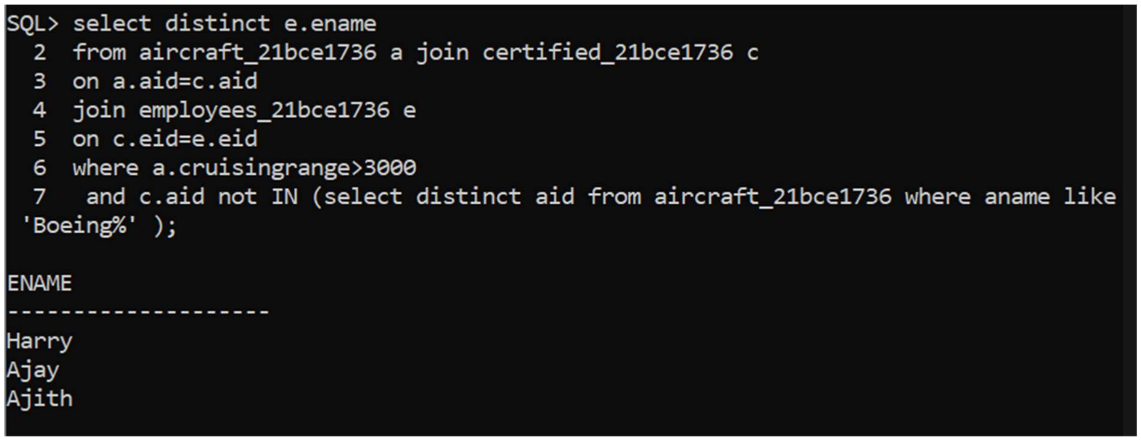
having count(distinct c.eid)=(select count(distinct eid) from employees_21BCE1542
where salary>1000000);



```
SQL> select distinct f.frm,f.too,count(distinct c.eid) as pilot_certification
  2   from flight_21bce1736 f join Aircraft_21bce1736 a
  3   on f.no=a.aid
  4   join certified_21bce1736 c
  5  on c.aid=a.aid
  6   join employees_21bce1736 e
  7   on e.eid=c.eid
  8   where e.salary>1000000 and f.distance>a.cruisingrange
  9   group by f.frm,f.too
 10  having count(distinct c.eid)=(select count(distinct eid) from employees_21bce17
36 where
 11  salary>1000000);

no rows selected
```

7. Print the enames of pilots who can operate planes with cruising range greater than 3000 miles but are not certified on any Boeing aircraft.

select distinct e.ename
from aircraft_21BCE1542 a join certified_21BCE1542 c
on a.aid=c.aid
join employees_21BCE1542 e
on c.eid=e.eid
where a.cruisingrange>3000
and c.aid not IN (select distinct aid from aircraft_21BCE1542 where aname like 'Boeing
%');



```
SQL> select distinct e.ename
  2  from aircraft_21bce1736 a join certified_21bce1736 c
  3  on a.aid=c.aid
  4  join employees_21bce1736 e
  5  on c.eid=e.eid
  6  where a.cruisingrange>3000
  7   and c.aid not IN (select distinct aid from aircraft_21bce1736 where aname like
 'Boeing%' );

ENAME
--------------------
Harry
Ajay
Ajith
```

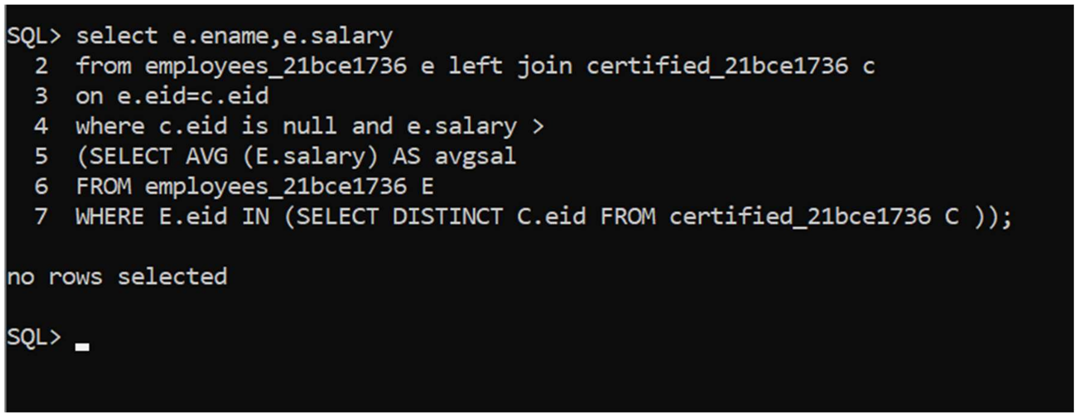
8. Compute the difference between the average salary of a pilot and the average salary of all employees (including pilots).

SELECT Temp1.avgsal - Temp2.avgsal
FROM (SELECT AVG (E.salary) AS avgsal
FROM employees_21BCE1542 E
WHERE E.eid IN (SELECT DISTINCT C.eid
FROM certified_21BCE1542 C)) as
Temp1,

(SELECT AVG (E1.salary) AS avgsal
FROM employees_21BCE1542 E1) as
Temp2;

9. Print the name and salary of every nonpilot whose salary is more than the average salary for pilots

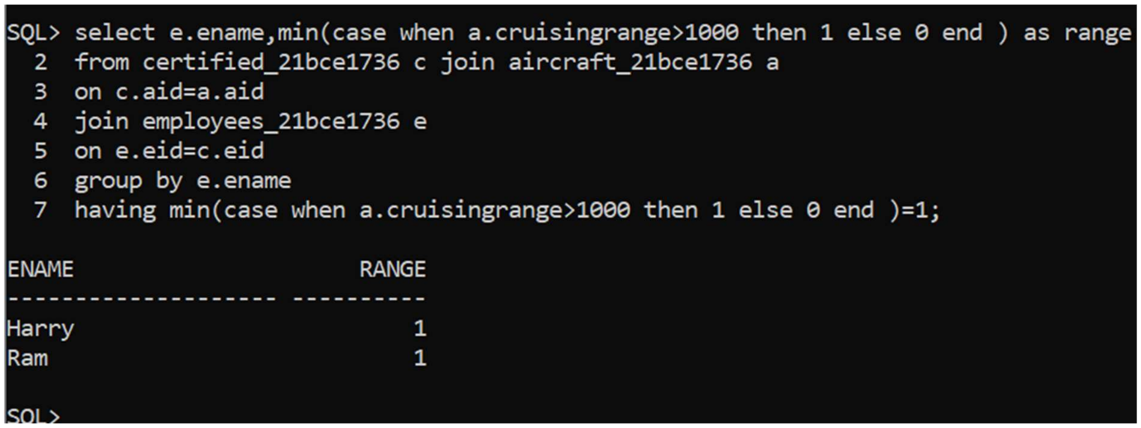
SELECT Temp1.avgsal - Temp2.avgsal
FROM (SELECT AVG (E.salary) AS avgsal
FROM employees_21BCE1542 E
WHERE E.eid IN (SELECT DISTINCT C.eid
FROM certified_21BCE1542 C)) as Temp1,
(SELECT AVG (E1.salary) AS avgsal
FROM employees_21BCE1542 E1) as



```
SQL> select e.ename,e.salary
  2  from employees_21bce1736 e left join certified_21bce1736 c
  3  on e.eid=c.eid
  4  where c.eid is null and e.salary >
  5  (SELECT AVG (E.salary) AS avgsal
  6  FROM employees_21bce1736 E
  7  WHERE E.eid IN (SELECT DISTINCT C.eid FROM certified_21bce1736 C ));

no rows selected

SQL>
```

10. Print the names of employees who are certified only on aircrafts with cruising range longer than 1000 miles.

select e.ename,min(case when a.cruisingrange>1000 then 1 else 0 end) as range
from certified_21BCE1542 c join aircraft_21BCE1542 a
on c.aid=a.aid
join employees_21BCE1542 e
on e.eid=c.eid
group by e.ename
having min(case when a.cruisingrange>1000 then 1 else 0 end)=1;



```
SQL> select e.ename,min(case when a.cruisingrange>1000 then 1 else 0 end ) as range
  2  from certified_21bce1736 c join aircraft_21bce1736 a
  3  on c.aid=a.aid
  4  join employees_21bce1736 e
  5  on e.eid=c.eid
  6  group by e.ename
  7  having min(case when a.cruisingrange>1000 then 1 else 0 end )=1;

ENAME                     RANGE
-------------------- ----------
Harry                         1
Ram                           1

SQL>
```

LAB 4 Aggregate Functions & Views

Exercise Questions on Aggregate Functions

1. To display the average and maximum cgpa for each school, we can use the GROUP BY clause with the AVG() and MAX() aggregate functions as follows:

```
SELECT school, AVG(cgpa) AS avg_cgpa, MAX(cgpa) AS max_cgpa
FROM student
GROUP BY school;
```

This will give the following result:

```
+--------+----------+----------+
| school | avg_cgpa | max_cgpa |
+--------+----------+----------+
| cse    | 9.15     | 9.3      |
| ece    | 7.3      | 7.8      |
| ecm    | 8.3      | 8.3      |
| it     | 8.9      | 8.9      |
+--------+----------+----------+
4 rows in set (0.00 sec)
```

2. To display the number of students whose cgpa is >9, we can use the COUNT() function with a WHERE clause as follows:

```
SELECT COUNT(*) AS num_students
FROM student
WHERE cgpa > 9;
```

This will give the following result:

```
+--------------+
| num_students |
+--------------+
| 2            |
+--------------+
1 row in set (0.00 sec)
```

3. To display the student name who secured more marks in each school, we can use a subquery to find the maximum cgpa for each school and join it with the original table to get the corresponding student name as follows:

```
SELECT s.school, s.sname
FROM student s
JOIN (SELECT school, MAX(cgpa) AS max_cgpa
      FROM student
      GROUP BY school) s2
ON s.school = s2.school AND s.cgpa = s2.max_cgpa;
```

This will give the following result:

```
+--------+-------+
| school | sname |
+--------+-------+
| cse    | alex  |
| ece    | virat |
| ecm    | anmol |
| it     | abhi  |
+--------+-------+
4 rows in set (0.00 sec)
```

4. To display the number of students registered under each school, we can use the GROUP BY clause with the COUNT() function as follows:

```
SELECT school, COUNT(*) AS num_students
FROM student
GROUP BY school;
```

This will give the following result:

```
+--------+--------------+
| school | num_students |
+--------+--------------+
| cse    | 2            |
| ece    | 2            |
| ecm    | 1            |
| it     | 1            |
+--------+--------------+
4 rows in set (0.00 sec)
```

Exercise Questions on views

1. View to find salesmen of New York with commission > 13%

```
CREATE VIEW ny_high_commission_salesmen AS
SELECT salesman_id, name, city, commission
FROM salesman
WHERE city = 'New York' AND commission > 0.13;
```

2. View to find order number, amount, salesman name, and customer name for each order

```
CREATE VIEW order_details AS
SELECT o.ord_no, o.purch_amt, s.name as salesman_name, c.cust_name
FROM orders o
INNER JOIN salesman s ON o.salesman_id = s.salesman_id
INNER JOIN customers c ON o.customer_id = c.customer_id;
```

3. View to find the number of salesmen in each city

```
CREATE VIEW salesmen_count_by_city AS
SELECT city, COUNT(salesman_id) AS num_salesmen
FROM salesman
GROUP BY city;
```

4. View to track the number of customers ordering, number of salesmen attached, average amount of orders, and total amount of orders in a day

```
CREATE VIEW daily_order_summary AS
SELECT
    o.ord_date,
    COUNT(DISTINCT o.customer_id) AS num_customers,
    COUNT(DISTINCT o.salesman_id) AS num_salesmen,
    AVG(o.purch_amt) AS avg_order_amount,
    SUM(o.purch_amt) AS total_order_amount
FROM orders o
GROUP BY o.ord_date;
```

NAME

MOHAMMAD SHAAD

REG. NO.

21BCE1542

SUBJECT

DATABASE MANAGEMENT SYSTEM

EXERCISE

FIVE

1. PL/SQL program to find small and large value

```
DECLARE
  num1 INTEGER := &Enter_first_integer;
  num2 INTEGER := &Enter_second_integer;
  small INTEGER;
  big INTEGER;
BEGIN
  IF num1 < num2 THEN
    small := num1;
    big := num2;
  ELSE
    small := num2;
    big := num1;
  END IF;
  DBMS_OUTPUT.PUT_LINE('The smallest value is ' ||
small);
  DBMS_OUTPUT.PUT_LINE('The largest value is ' ||
big);
END;
/
```

2. PL/SQL program to count employees and check vacancies

```
DECLARE
  dept_count INTEGER;
  emp_count INTEGER;
BEGIN
  FOR dept IN (SELECT deptid FROM department) LOOP
    SELECT COUNT(*) INTO emp_count FROM employee
WHERE deptid = dept.deptid;
    IF emp_count < 45 THEN
      DBMS_OUTPUT.PUT_LINE('Department ' ||
dept.deptid || ' has ' || emp_count || ' employees
and has vacancies.');
    ELSE
      DBMS_OUTPUT.PUT_LINE('Department ' ||
dept.deptid || ' has ' || emp_count || ' employees
and is full.');
    END IF;
  END LOOP;
END;
/
```

3. PL/SQL procedure to calculate incentive amount

```
CREATE OR REPLACE PROCEDURE
calculate_incentive(emp_id IN INTEGER) AS
  salary NUMBER;
  incentive NUMBER;
BEGIN
  SELECT salary INTO salary FROM employee WHERE empid
= emp_id;
  incentive := salary * 0.1;
  DBMS_OUTPUT.PUT_LINE('Employee ' || emp_id || ' has
a salary of ' || salary || ' and is eligible for an
incentive of ' || incentive);
END;
/
```

4.Stored procedure to seek and delete/update stock

```
CREATE OR REPLACE PROCEDURE update_stock(item_code IN
VARCHAR2) AS
  last_purchased DATE;
  current_stock INTEGER;
BEGIN
  SELECT last_purchase, current_stock INTO
last_purchased, current_stock FROM stock WHERE
item_code = item_code;
  IF last_purchased < ADD_MONTHS(SYSDATE, -12) THEN
    DELETE FROM stock WHERE item_code = item_code;
  ELSE
    UPDATE stock SET current_stock = current_stock +
1 WHERE item_code = item_code;
  END IF;
END;
/
```

5. Function to search for address using phone number

```
CREATE OR REPLACE FUNCTION get_address(phone_num IN
VARCHAR2) RETURN VARCHAR2 AS
  address VARCHAR2(100);
BEGIN
  SELECT address INTO address FROM phone_user WHERE
phone_number = phone_num;
  RETURN address;
END;
/
```

NAME

MOHAMMAD SHAAD

REG. NO.

21BCE1542

SUBJECT

DATABASE MANAGEMENT SYSTEM

EXERCISE

SIX

```
CREATE TABLE EMP21BCE1542 (EMPNO NUMBER(4), ENAME
VARCHAR(30), JOB VARCHAR2(15), SAL NUMBER(8), DEPTNO
NUMBER(2), COMMISSION NUMBER(7) );
```

```
INSERT INTO EMP21BCE1542 VALUES (1005,'SHAAD','FULL
STACK DEV',60000,10,526);
```

I. (a)

```
DECLARE
  CURSOR emp_cur IS
    SELECT Empno, Ename, Job
    FROM EMP21BCE1542
    WHERE DeptNo = 10;
BEGIN
  FOR emp_rec IN emp_cur LOOP
    DBMS_OUTPUT.PUT_LINE(emp_rec.Empno || ' ' ||
emp_rec.Ename || ' ' || emp_rec.Job);
  END LOOP;
END;
/
```

(b)

```
DECLARE
  CURSOR sal_cur IS
    SELECT Empno, Sal, DeptNo
    FROM EMP21BCE1542;
  v_empno emp21bce1542.empno%TYPE;
  v_old_sal emp21bce1542.sal%TYPE;
  v_new_sal emp21bce1542.sal%TYPE;
BEGIN
  FOR sal_rec IN sal_cur LOOP
    v_empno := sal_rec.Empno;
    v_old_sal := sal_rec.Sal;
    IF sal_rec.DeptNo = 10 THEN
      v_new_sal := v_old_sal + 1000;
    ELSIF sal_rec.DeptNo = 20 THEN
      v_new_sal := v_old_sal + 500;
    ELSIF sal_rec.DeptNo = 30 THEN
      v_new_sal := v_old_sal + 800;
    END IF;
    UPDATE EMP21BCE1542
    SET Sal = v_new_sal
    WHERE Empno = v_empno;
    INSERT INTO TEMP (Empid, Old, New)
    VALUES (v_empno, v_old_sal, v_new_sal);
  END LOOP;
END;
/
```

(c)

```
DECLARE
  v_avg_sal emp21bce1542.sal%TYPE;
CURSOR emp_cur IS
  SELECT Ename, Sal
  FROM EMP21BCE1542
  WHERE Sal < v_avg_sal;
BEGIN
  SELECT AVG(Sal) INTO v_avg_sal
  FROM EMP21BCE1542;
  FOR emp_rec IN emp_cur LOOP
    DBMS_OUTPUT.PUT_LINE(emp_rec.Ename || ' ' ||
emp_rec.Sal);
  END LOOP;
END;
/
```

II. (a)

```
CREATE OR REPLACE TRIGGER sal_update_trigger
BEFORE UPDATE ON EMP21BCE1542
FOR EACH ROW
BEGIN
  IF :new.Sal <= :old.Sal THEN
    RAISE_APPLICATION_ERROR(-20001, 'Salary cannot be
decreased');
  END IF;
END;
/
```

(b)

```
CREATE OR REPLACE TRIGGER dept_emp_limit_trigger
BEFORE INSERT ON EMP21BCE1542
FOR EACH ROW
DECLARE
  v_emp_count NUMBER;
BEGIN
  SELECT COUNT(*) INTO v_emp_count
  FROM EMP21BCE1542
  WHERE DeptNo = 2;
  IF v_emp_count >= 5 THEN
    RAISE_APPLICATION_ERROR(-20002, 'Cannot add
employee, department employee limit exceeded');
  END IF;
END;
/
```

(c)

```
CREATE OR REPLACE TRIGGER negative_balance_trigger
AFTER UPDATE ON Account
FOR EACH ROW
WHEN (NEW.balance < 0)
DECLARE
    loan_no Account.acctno%TYPE := :NEW.acctno;
    br_name Account.br_name%TYPE := :NEW.br_name;
    loan_amount Loan.amount%TYPE := -1 *
:NEW.balance;
BEGIN
    -- Insert new tuple into Loan table
    INSERT INTO Loan (loan_no, br_name, amount)
VALUES (loan_no, br_name, loan_amount);

    -- Insert new tuple into Borrower table
    INSERT INTO Borrower (custname, loan_no) VALUES
('Jones', loan_no);

    -- Update the balance in Account table
    UPDATE Account SET balance = 0 WHERE acctno =
loan_no;

    DBMS_OUTPUT.PUT_LINE('Trigger executed
successfully!');
END;
/
```