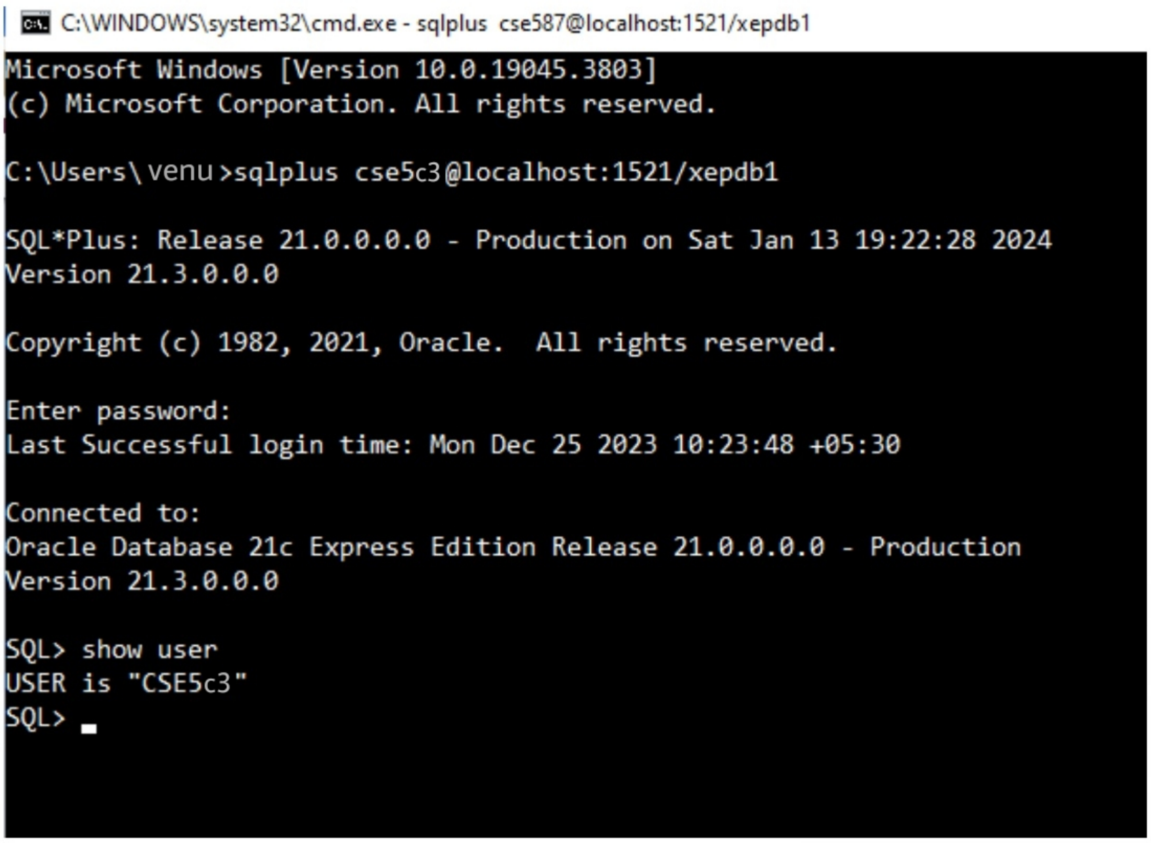
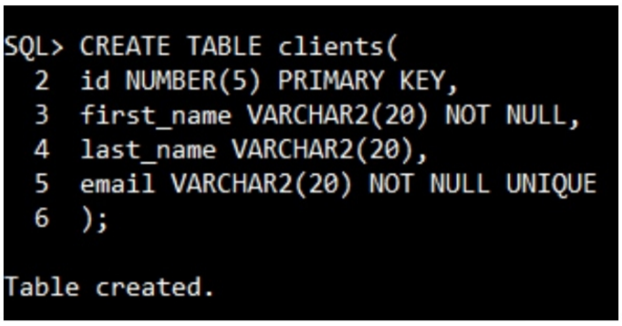


```
C:\WINDOWS\system32\cmd.exe - sqlplus  cse587@localhost:1521/xepdb1
Microsoft Windows [Version 10.0.19045.3803]
(c) Microsoft Corporation. All rights reserved.

C:\Users\venu>sqlplus cse5c3@localhost:1521/xepdb1

SQL*Plus: Release 21.0.0.0.0 - Production on Sat Jan 13 19:22:28 2024
Version 21.3.0.0.0

Copyright (c) 1982, 2021, Oracle.  All rights reserved.

Enter password:
Last Successful login time: Mon Dec 25 2023 10:23:48 +05:30

Connected to:
Oracle Database 21c Express Edition Release 21.0.0.0.0 - Production
Version 21.3.0.0.0

SQL> show user
USER is "CSE5c3"
SQL> _
```

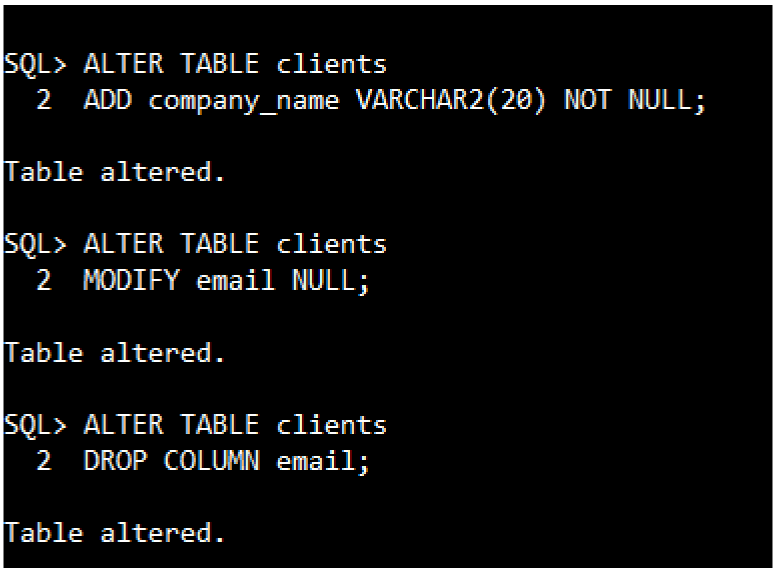
Exp 1-DDL COMMANDS

Create table:

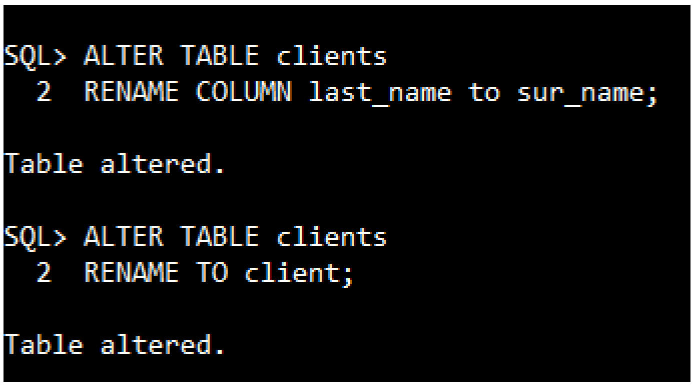


```
SQL> CREATE TABLE clients(
  2  id NUMBER(5) PRIMARY KEY,
  3  first_name VARCHAR2(20) NOT NULL,
  4  last_name VARCHAR2(20),
  5  email VARCHAR2(20) NOT NULL UNIQUE
  6  );

Table created.
```

Alter table:

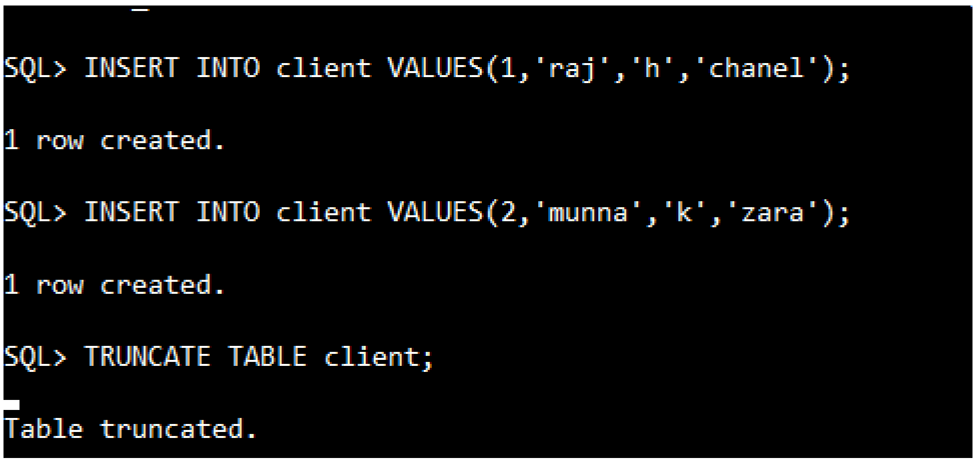


```
SQL> ALTER TABLE clients
  2  ADD company_name VARCHAR2(20) NOT NULL;

Table altered.

SQL> ALTER TABLE clients
  2  MODIFY email NULL;

Table altered.

SQL> ALTER TABLE clients
  2  DROP COLUMN email;

Table altered.
```

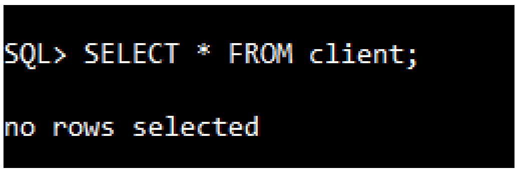


```
SQL> ALTER TABLE clients
  2  RENAME COLUMN last_name to sur_name;

Table altered.

SQL> ALTER TABLE clients
  2  RENAME TO client;

Table altered.
```

Truncate table:

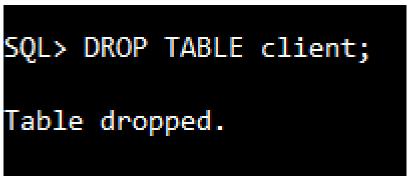


```
SQL> INSERT INTO client VALUES(1,'raj','h','chanel');

1 row created.

SQL> INSERT INTO client VALUES(2,'munna','k','zara');

1 row created.

SQL> TRUNCATE TABLE client;

Table truncated.
```

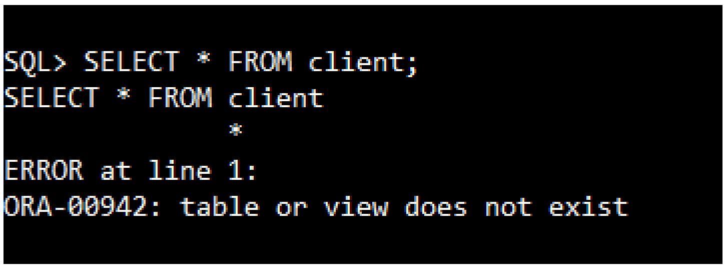


```
SQL> SELECT * FROM client;

no rows selected
```

Drop table:

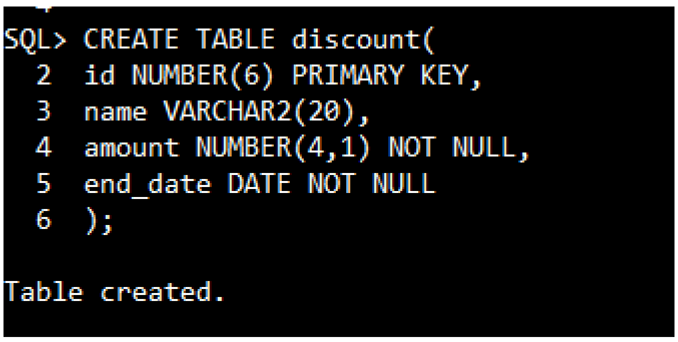


```
SQL> DROP TABLE client;

Table dropped.
```



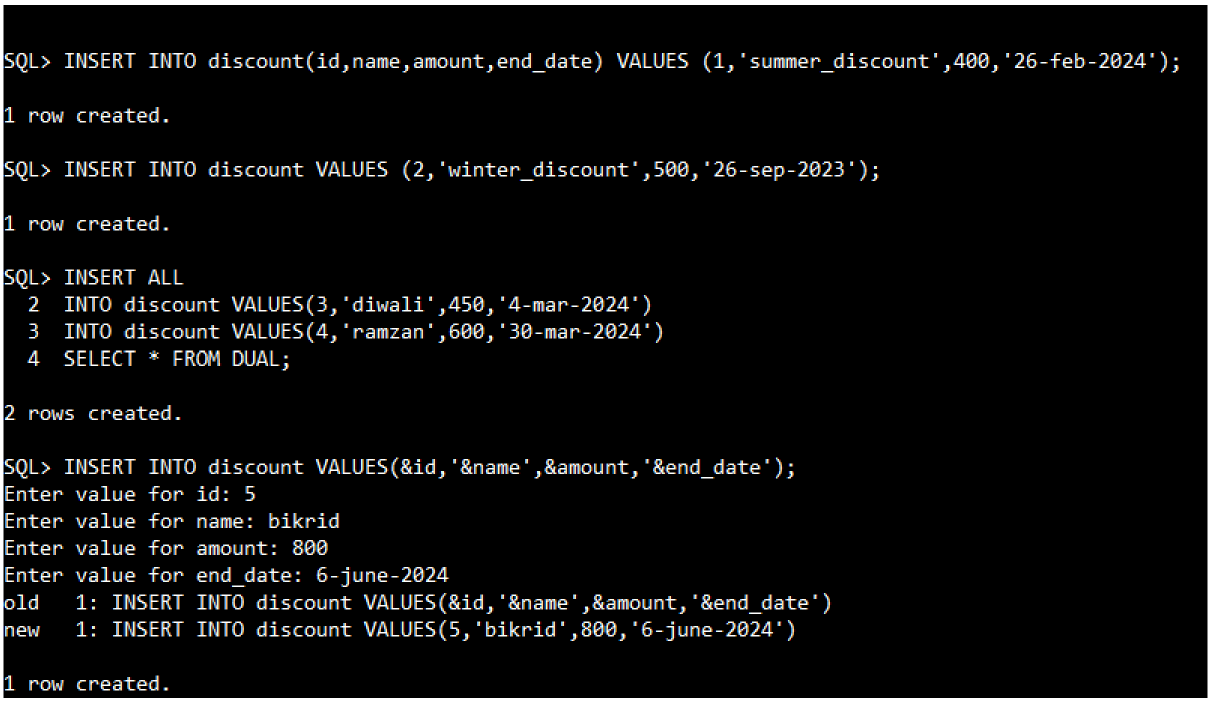
```
SQL> SELECT * FROM client;
SELECT * FROM client
              *
ERROR at line 1:
ORA-00942: table or view does not exist
```

EXP 2-DML COMMANDS



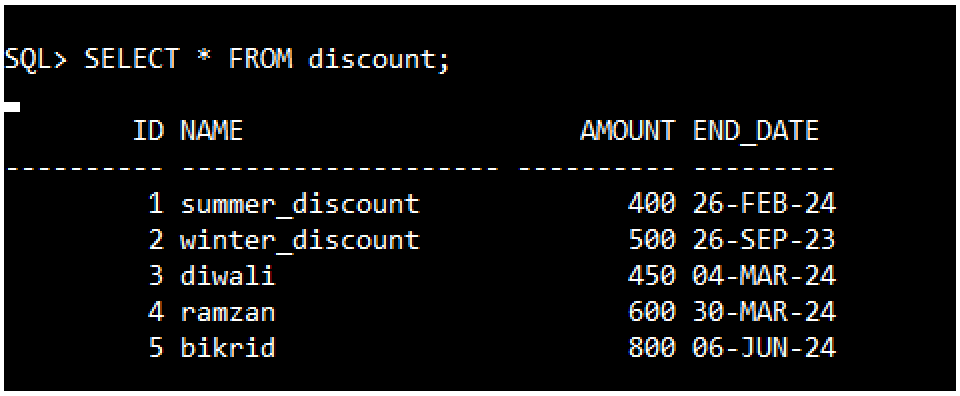
```
SQL> CREATE TABLE discount(
  2  id NUMBER(6) PRIMARY KEY,
  3  name VARCHAR2(20),
  4  amount NUMBER(4,1) NOT NULL,
  5  end_date DATE NOT NULL
  6  );

Table created.
```

Insert into table:



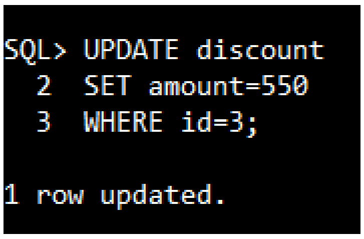
```
SQL> INSERT INTO discount(id,name,amount,end_date) VALUES (1,'summer_discount',400,'26-feb-2024');

1 row created.

SQL> INSERT INTO discount VALUES (2,'winter_discount',500,'26-sep-2023');

1 row created.

SQL> INSERT ALL
  2  INTO discount VALUES(3,'diwali',450,'4-mar-2024')
  3  INTO discount VALUES(4,'ramzan',600,'30-mar-2024')
  4  SELECT * FROM DUAL;

2 rows created.

SQL> INSERT INTO discount VALUES(&id,'&name',&amount,'&end_date');
Enter value for id: 5
Enter value for name: bikrid
Enter value for amount: 800
Enter value for end_date: 6-june-2024
old   1: INSERT INTO discount VALUES(&id,'&name',&amount,'&end_date')
new   1: INSERT INTO discount VALUES(5,'bikrid',800,'6-june-2024')

1 row created.
```

Select:

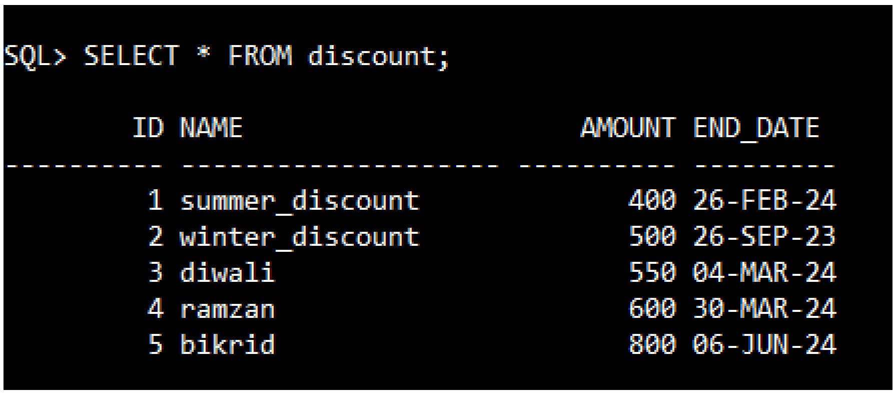


```
SQL> SELECT * FROM discount;

        ID NAME                     AMOUNT END_DATE
---------- -------------------- ---------- ---------
         1 summer_discount             400 26-FEB-24
         2 winter_discount             500 26-SEP-23
         3 diwali                      450 04-MAR-24
         4 ramzan                      600 30-MAR-24
         5 bikrid                      800 06-JUN-24
```

Update table:

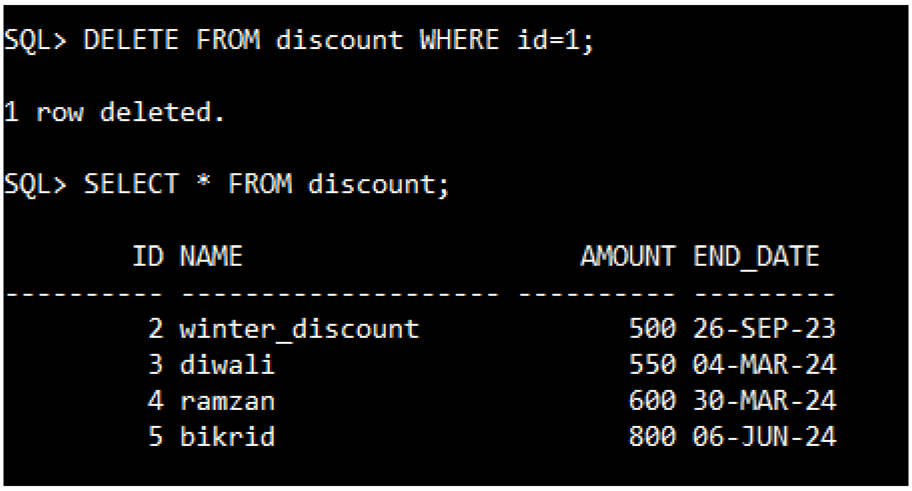


```
SQL> UPDATE discount
  2  SET amount=550
  3  WHERE id=3;

1 row updated.
```



```
SQL> SELECT * FROM discount;

        ID NAME                     AMOUNT END_DATE
---------- -------------------- ---------- ---------
         1 summer_discount             400 26-FEB-24
         2 winter_discount             500 26-SEP-23
         3 diwali                      550 04-MAR-24
         4 ramzan                      600 30-MAR-24
         5 bikrid                      800 06-JUN-24
```

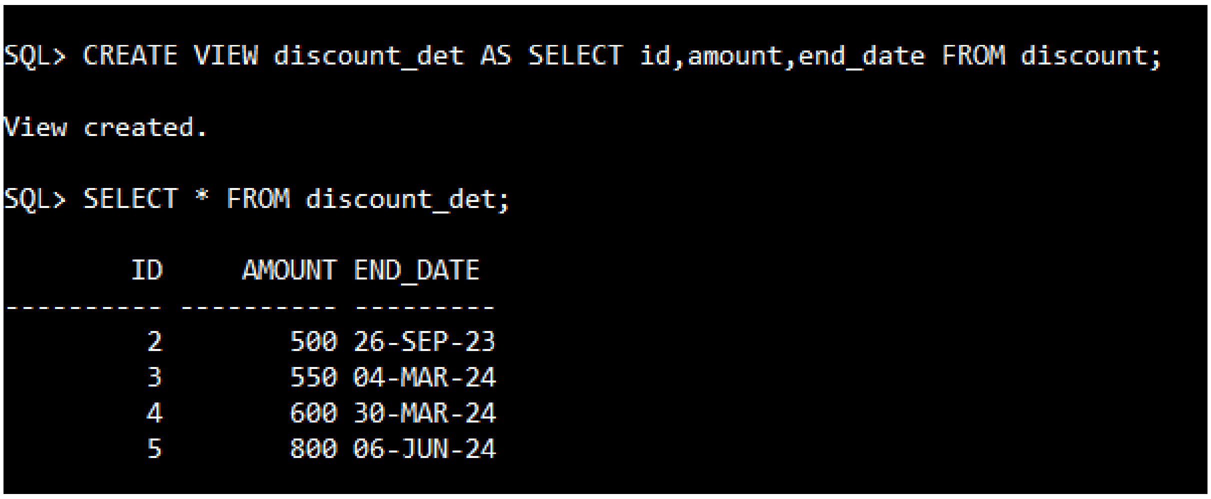
Delete:



```
SQL> DELETE FROM discount WHERE id=1;

1 row deleted.

SQL> SELECT * FROM discount;

        ID NAME                     AMOUNT END_DATE
---------- -------------------- ---------- ---------
         2 winter_discount             500 26-SEP-23
         3 diwali                      550 04-MAR-24
         4 ramzan                      600 30-MAR-24
         5 bikrid                      800 06-JUN-24
```

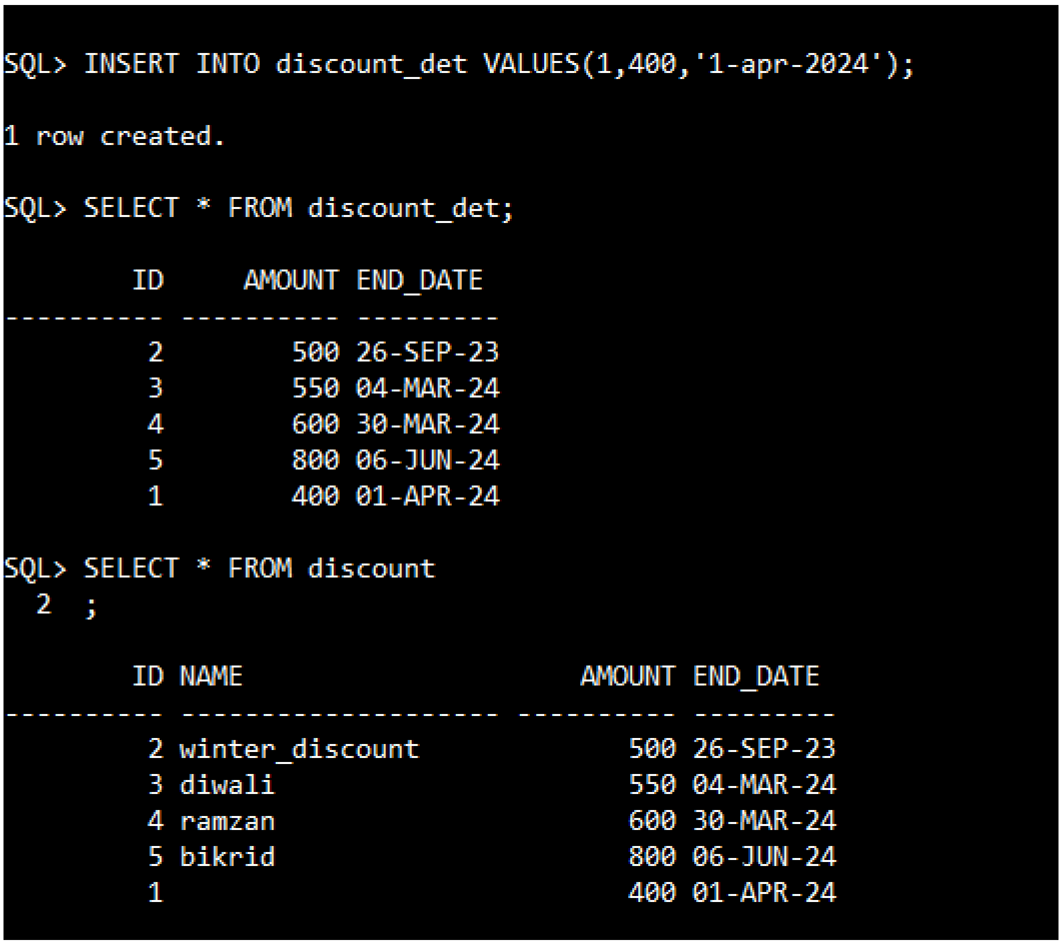
EXP 3-VIEWS

Create view:



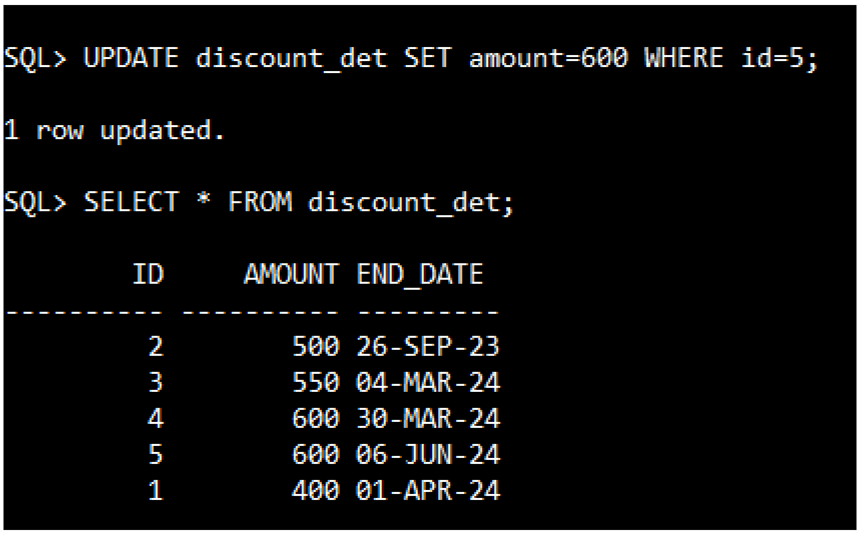
```
SQL> CREATE VIEW discount_det AS SELECT id,amount,end_date FROM discount;

View created.

SQL> SELECT * FROM discount_det;

        ID     AMOUNT END_DATE
---------- ---------- ---------
         2        500 26-SEP-23
         3        550 04-MAR-24
         4        600 30-MAR-24
         5        800 06-JUN-24
```

Insert into view:



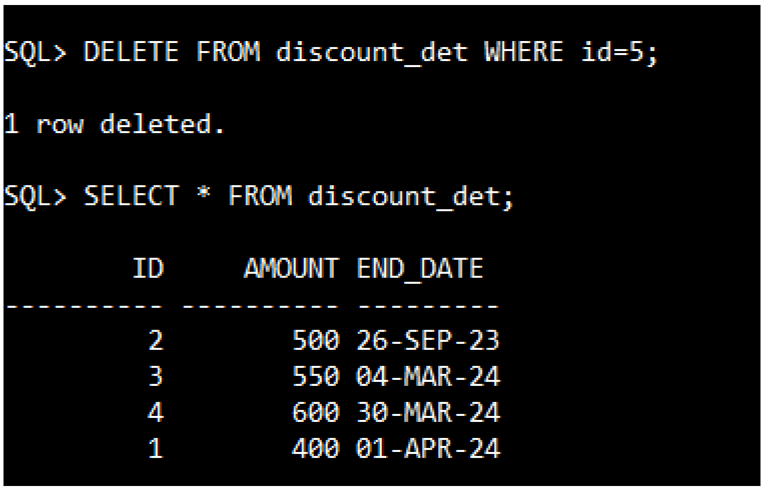
```
SQL> INSERT INTO discount_det VALUES(1,400,'1-apr-2024');

1 row created.

SQL> SELECT * FROM discount_det;

        ID     AMOUNT END_DATE
---------- ---------- ---------
         2        500 26-SEP-23
         3        550 04-MAR-24
         4        600 30-MAR-24
         5        800 06-JUN-24
         1        400 01-APR-24

SQL> SELECT * FROM discount
  2  ;

        ID NAME                     AMOUNT END_DATE
---------- -------------------- ---------- ---------
         2 winter_discount             500 26-SEP-23
         3 diwali                      550 04-MAR-24
         4 ramzan                      600 30-MAR-24
         5 bikrid                      800 06-JUN-24
         1                             400 01-APR-24
```

Update view:



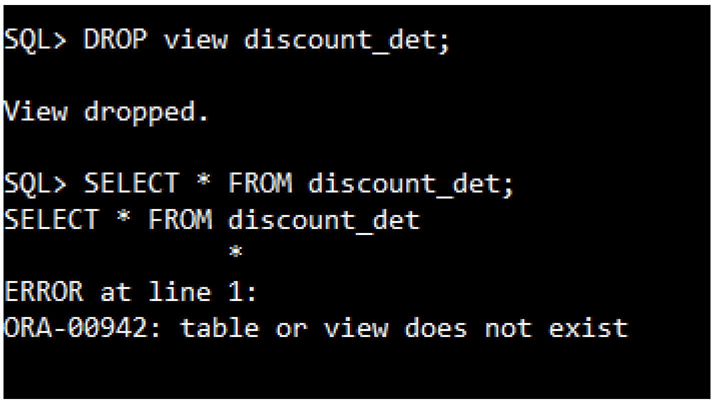
```
SQL> UPDATE discount_det SET amount=600 WHERE id=5;

1 row updated.

SQL> SELECT * FROM discount_det;

        ID     AMOUNT END_DATE
---------- ---------- ---------
         2        500 26-SEP-23
         3        550 04-MAR-24
         4        600 30-MAR-24
         5        600 06-JUN-24
         1        400 01-APR-24
```

Delete view:



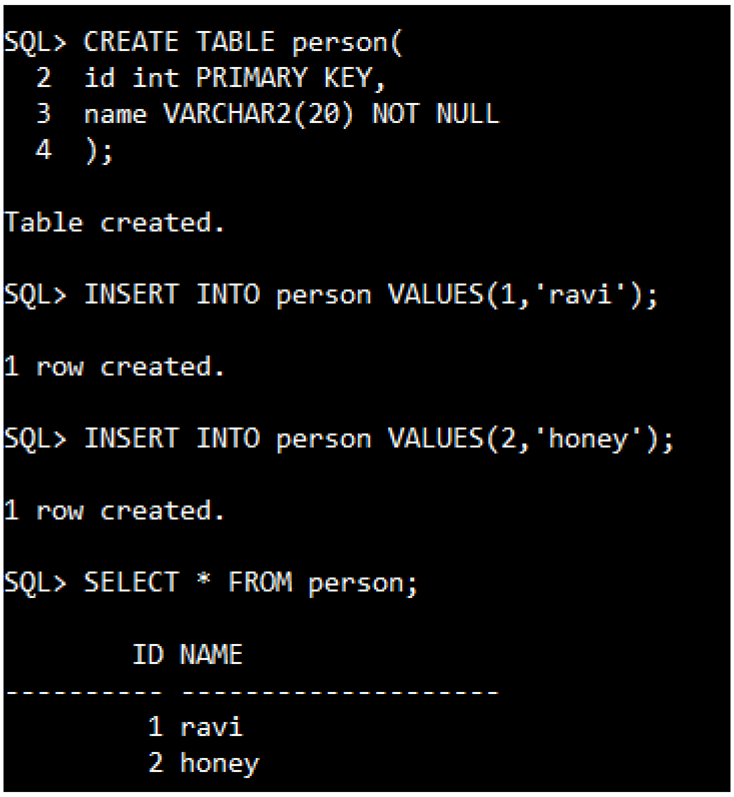
```
SQL> DELETE FROM discount_det WHERE id=5;

1 row deleted.

SQL> SELECT * FROM discount_det;

        ID     AMOUNT END_DATE
---------- ---------- ---------
         2        500 26-SEP-23
         3        550 04-MAR-24
         4        600 30-MAR-24
         1        400 01-APR-24
```

Drop view:

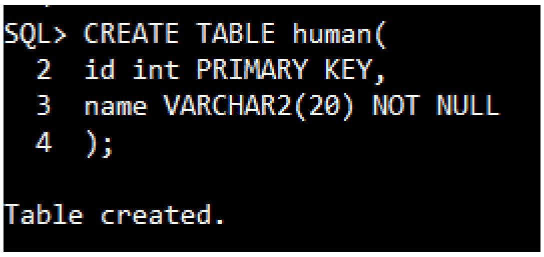


```
SQL> DROP view discount_det;

View dropped.

SQL> SELECT * FROM discount_det;
SELECT * FROM discount_det
              *
ERROR at line 1:
ORA-00942: table or view does not exist
```

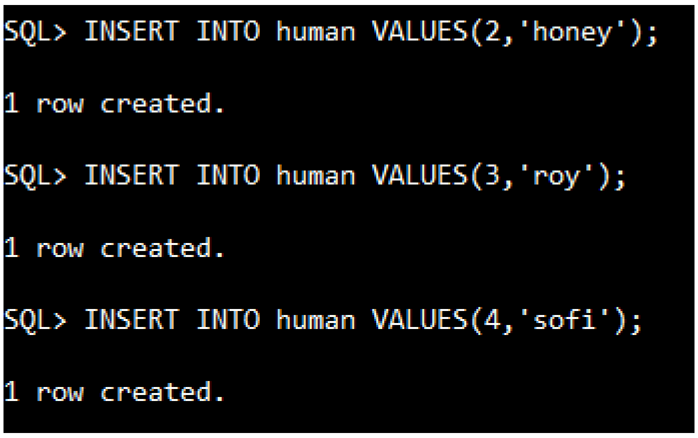
EXP4-RELATIONAL SET OPERATIONS



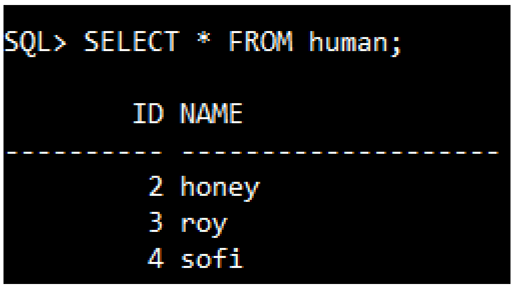
```
SQL> CREATE TABLE person(
  2  id int PRIMARY KEY,
  3  name VARCHAR2(20) NOT NULL
  4  );

Table created.

SQL> INSERT INTO person VALUES(1,'ravi');

1 row created.

SQL> INSERT INTO person VALUES(2,'honey');

1 row created.

SQL> SELECT * FROM person;

        ID NAME
---------- --------------------
         1 ravi
         2 honey
```



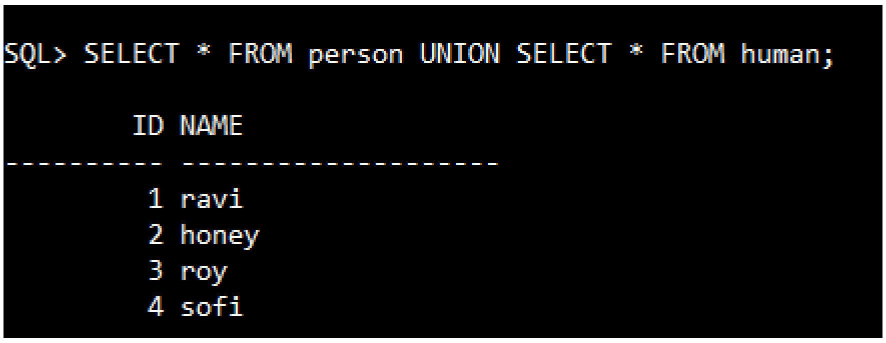
```
SQL> CREATE TABLE human(
  2  id int PRIMARY KEY,
  3  name VARCHAR2(20) NOT NULL
  4  );

Table created.
```



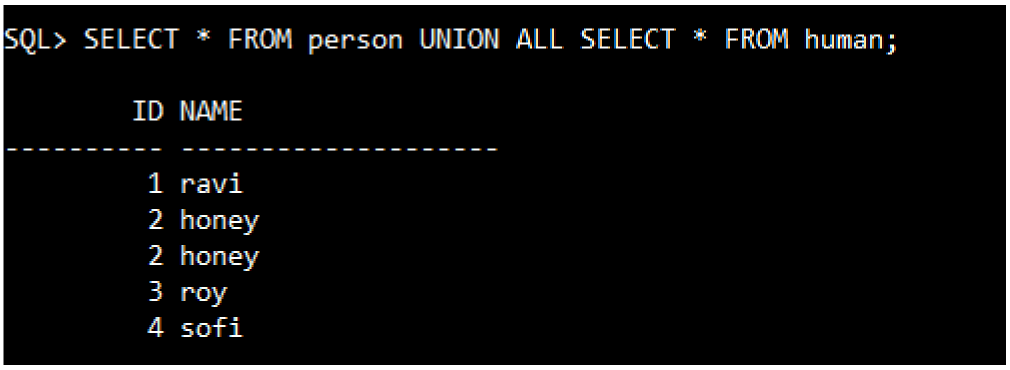
```
SQL> INSERT INTO human VALUES(2,'honey');

1 row created.

SQL> INSERT INTO human VALUES(3,'roy');

1 row created.

SQL> INSERT INTO human VALUES(4,'sofi');

1 row created.
```


```
SQL> SELECT * FROM human;

        ID NAME
---------- --------------------
         2 honey
         3 roy
         4 sofi
```

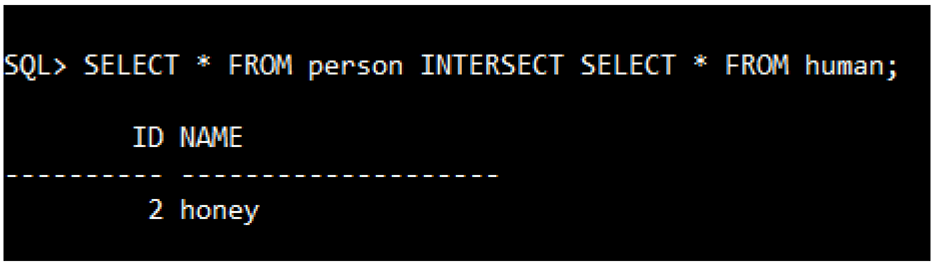
Union:


```
SQL> SELECT * FROM person UNION SELECT * FROM human;

        ID NAME
---------- --------------------
         1 ravi
         2 honey
         3 roy
         4 sofi
```

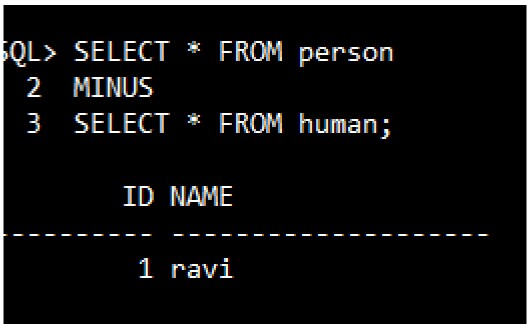
Union all:


```
SQL> SELECT * FROM person UNION ALL SELECT * FROM human;

        ID NAME
---------- --------------------
         1 ravi
         2 honey
         2 honey
         3 roy
         4 sofi
```

Intersect:

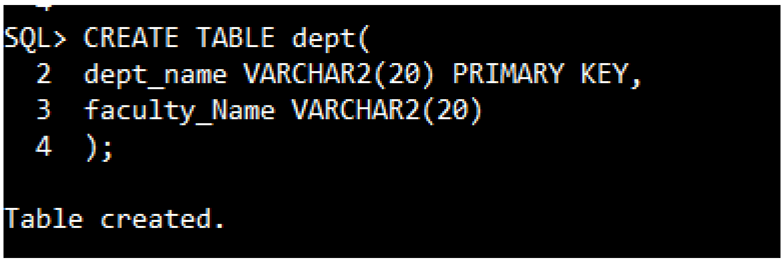

```
SQL> SELECT * FROM person INTERSECT SELECT * FROM human;

        ID NAME
---------- --------------------
         2 honey
```

Minus

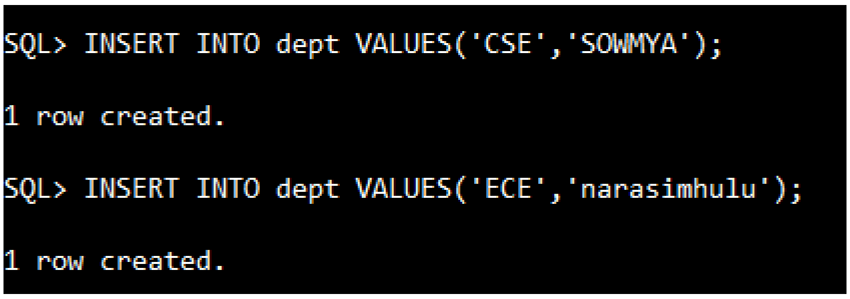


```
SQL> SELECT * FROM person
  2  MINUS
  3  SELECT * FROM human;

        ID NAME
---------- --------------------
         1 ravi
```

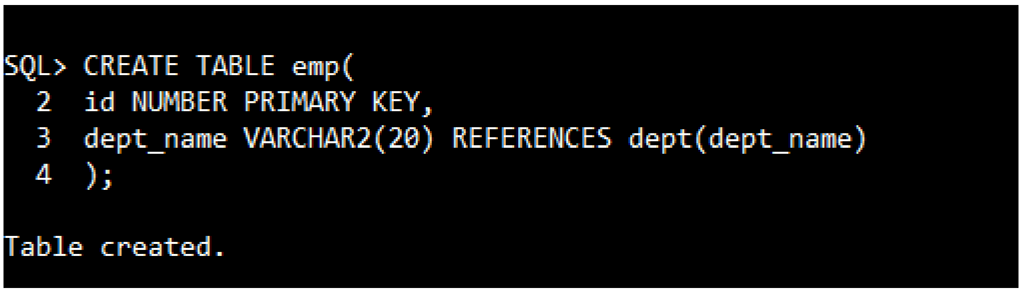
Natural join



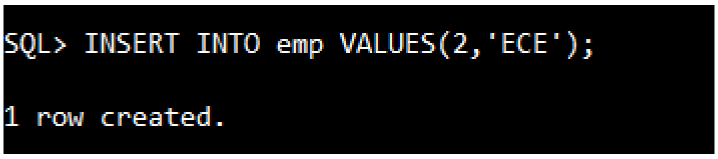
```
SQL> CREATE TABLE dept(
  2  dept_name VARCHAR2(20) PRIMARY KEY,
  3  faculty_Name VARCHAR2(20)
  4  );

Table created.
```



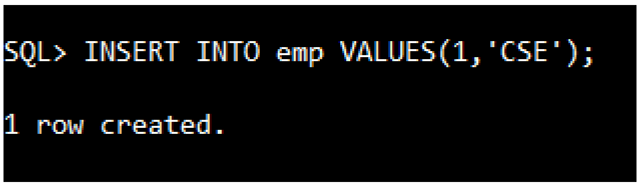
```
SQL> INSERT INTO dept VALUES('CSE','SOWMYA');

1 row created.

SQL> INSERT INTO dept VALUES('ECE','narasimhulu');

1 row created.
```



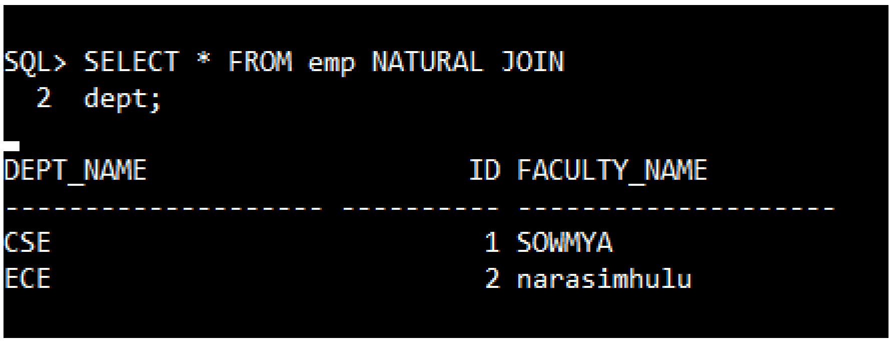
```
SQL> CREATE TABLE emp(
  2  id NUMBER PRIMARY KEY,
  3  dept_name VARCHAR2(20) REFERENCES dept(dept_name)
  4  );

Table created.
```



```
SQL> INSERT INTO emp VALUES(2,'ECE');

1 row created.
```

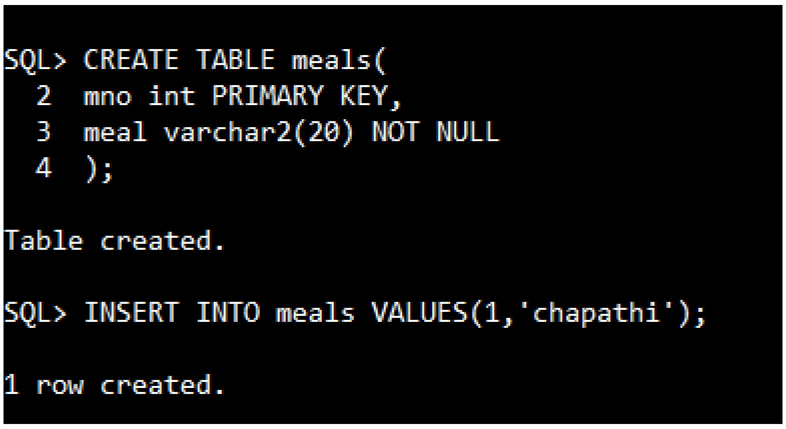


```
SQL> INSERT INTO emp VALUES(1,'CSE');

1 row created.
```

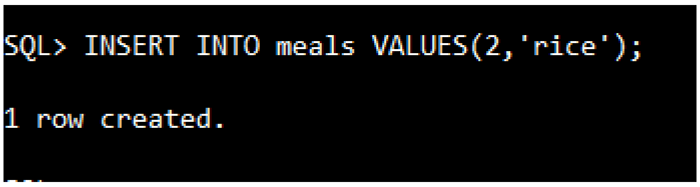


```
SQL> SELECT * FROM emp NATURAL JOIN
  2  dept;

DEPT_NAME                              ID FACULTY_NAME
-------------------- ---------- --------------------
CSE                                     1 SOWMYA
ECE                                     2 narasimhulu
```

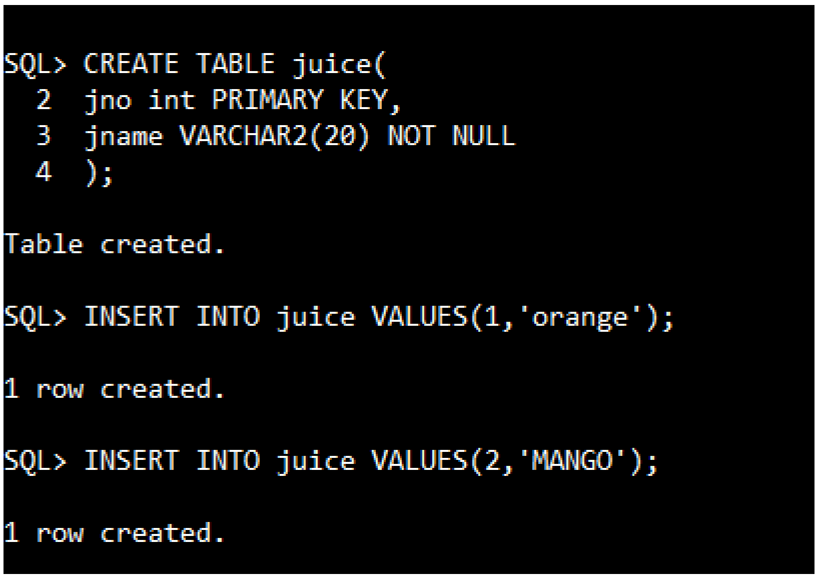
Cross join:



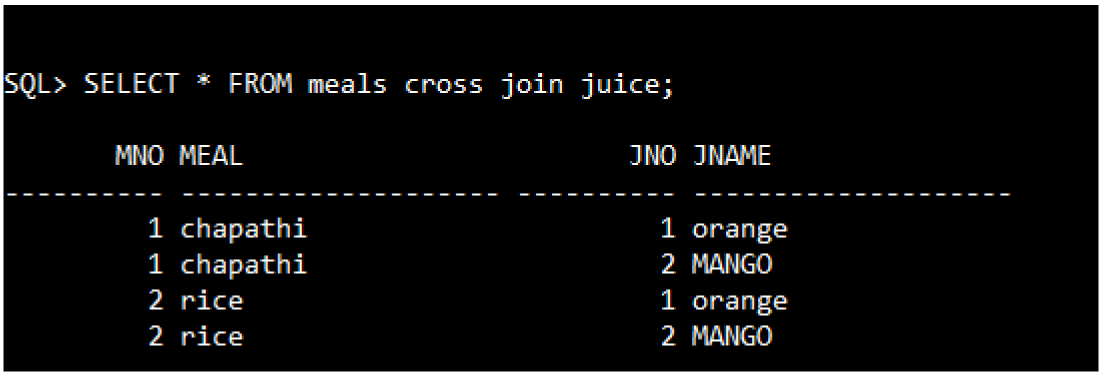
```
SQL> CREATE TABLE meals(
  2  mno int PRIMARY KEY,
  3  meal varchar2(20) NOT NULL
  4  );

Table created.

SQL> INSERT INTO meals VALUES(1,'chapathi');

1 row created.
```



```
SQL> INSERT INTO meals VALUES(2,'rice');

1 row created.
```

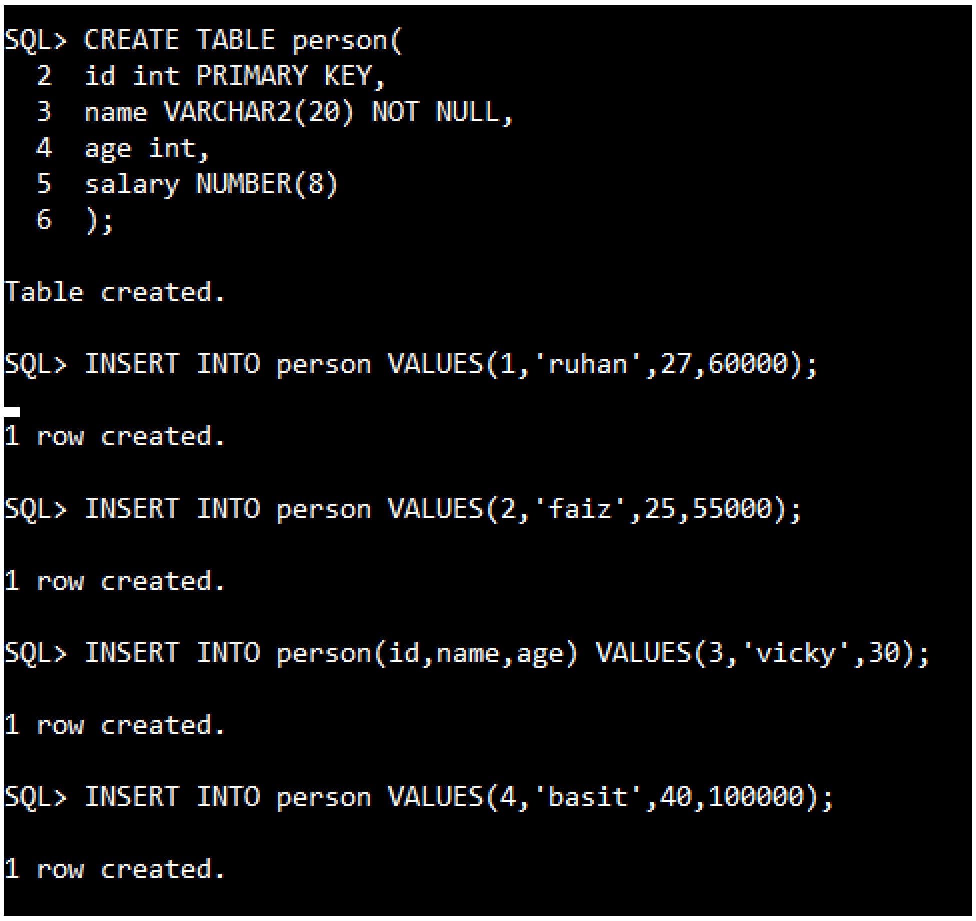


```
SQL> CREATE TABLE juice(
  2  jno int PRIMARY KEY,
  3  jname VARCHAR2(20) NOT NULL
  4  );

Table created.

SQL> INSERT INTO juice VALUES(1,'orange');

1 row created.

SQL> INSERT INTO juice VALUES(2,'MANGO');

1 row created.
```

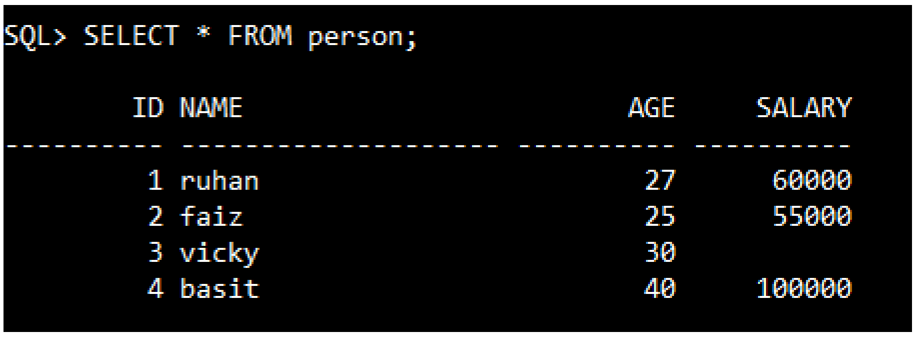


```
SQL> SELECT * FROM meals cross join juice;

       MNO MEAL                                  JNO JNAME
---------- -------------------- ---------- --------------------
         1 chapathi                                1 orange
         1 chapathi                                2 MANGO
         2 rice                                    1 orange
         2 rice                                    2 MANGO
```

EXP5-SPECIAL OPERATIONS

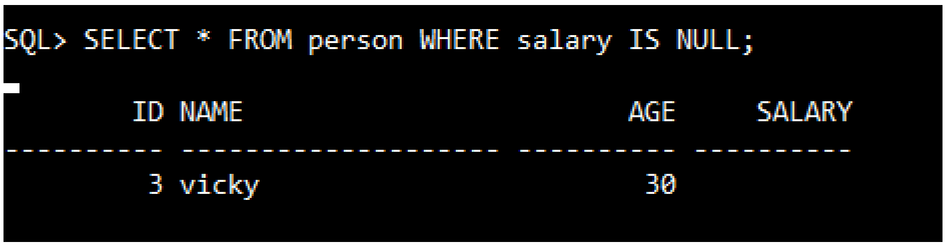


```
SQL> CREATE TABLE person(
  2  id int PRIMARY KEY,
  3  name VARCHAR2(20) NOT NULL,
  4  age int,
  5  salary NUMBER(8)
  6  );

Table created.

SQL> INSERT INTO person VALUES(1,'ruhan',27,60000);

1 row created.

SQL> INSERT INTO person VALUES(2,'faiz',25,55000);

1 row created.

SQL> INSERT INTO person(id,name,age) VALUES(3,'vicky',30);

1 row created.

SQL> INSERT INTO person VALUES(4,'basit',40,100000);

1 row created.
```



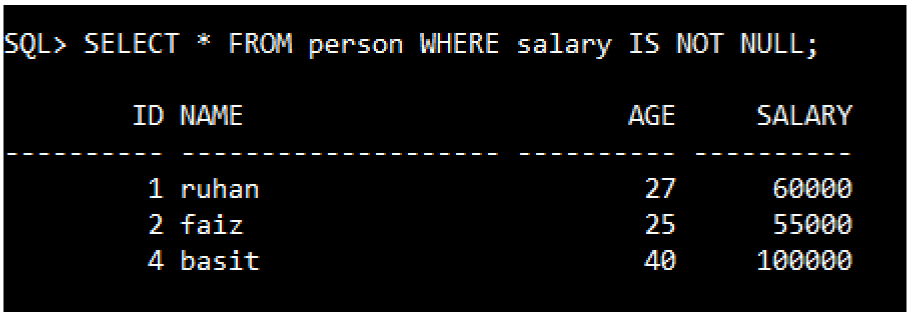
```
SQL> SELECT * FROM person;

        ID NAME                        AGE     SALARY
---------- -------------------- ---------- ----------
         1 ruhan                        27      60000
         2 faiz                         25      55000
         3 vicky                        30
         4 basit                        40     100000
```

IS NULL



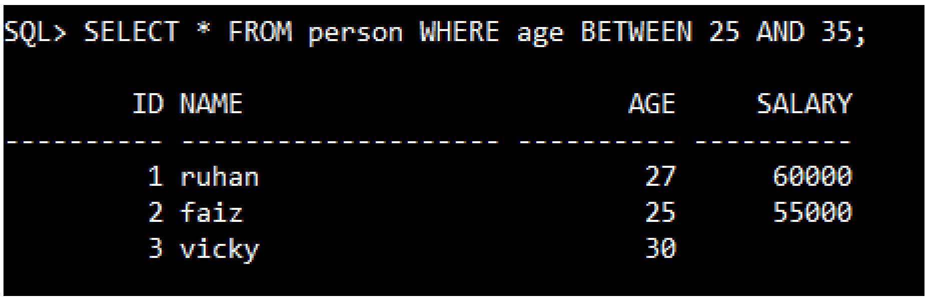
```
SQL> SELECT * FROM person WHERE salary IS NULL;

        ID NAME                        AGE     SALARY
---------- -------------------- ---------- ----------
         3 vicky                        30
```

IS NOT NULL



```
SQL> SELECT * FROM person WHERE salary IS NOT NULL;

        ID NAME                        AGE     SALARY
---------- -------------------- ---------- ----------
         1 ruhan                        27      60000
         2 faiz                         25      55000
         4 basit                        40     100000
```

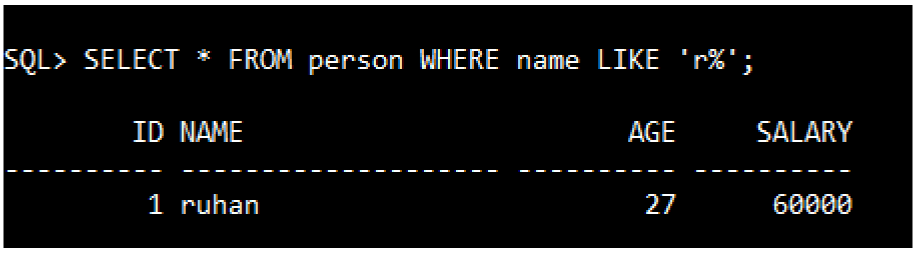
BETWEEN



```
SQL> SELECT * FROM person WHERE age BETWEEN 25 AND 35;

        ID NAME                        AGE     SALARY
---------- -------------------- ---------- ----------
         1 ruhan                        27      60000
         2 faiz                         25      55000
         3 vicky                        30
```

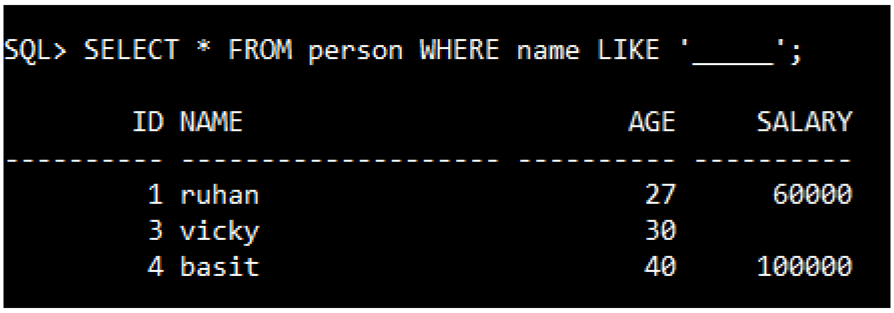
LIKE

%:



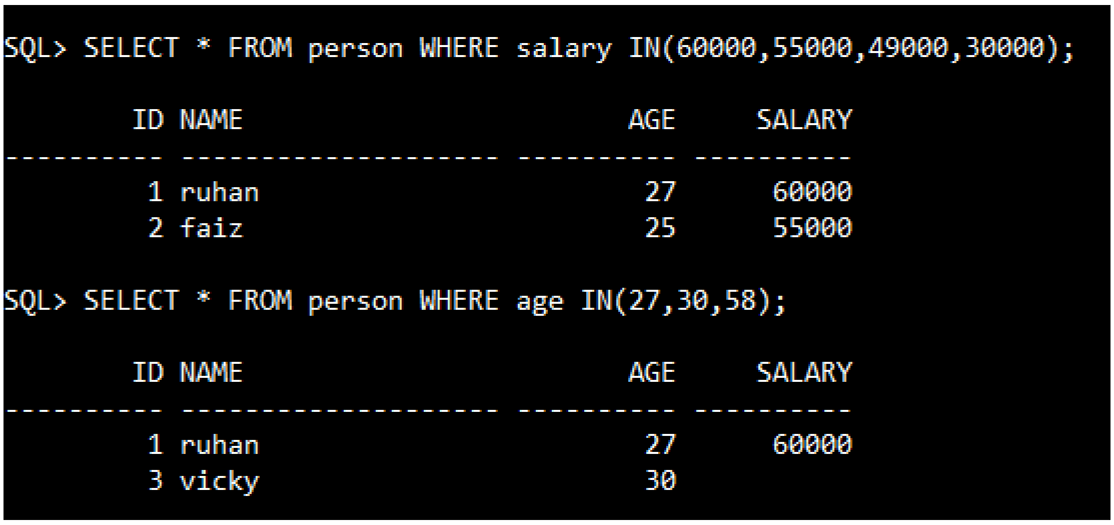
```
SQL> SELECT * FROM person WHERE name LIKE 'r%';

        ID NAME                        AGE     SALARY
---------- -------------------- ---------- ----------
         1 ruhan                        27      60000
```

_:

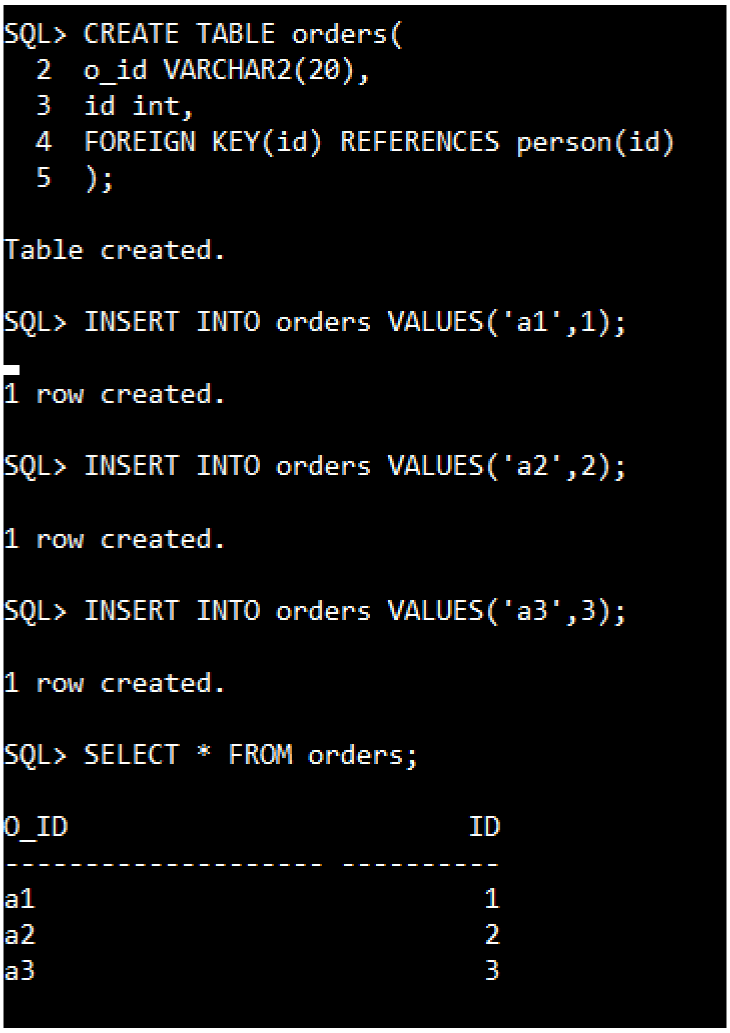


```
SQL> SELECT * FROM person WHERE name LIKE '_____';

        ID NAME                        AGE     SALARY
---------- -------------------- ---------- ----------
         1 ruhan                        27      60000
         3 vicky                        30
         4 basit                        40     100000
```

IN

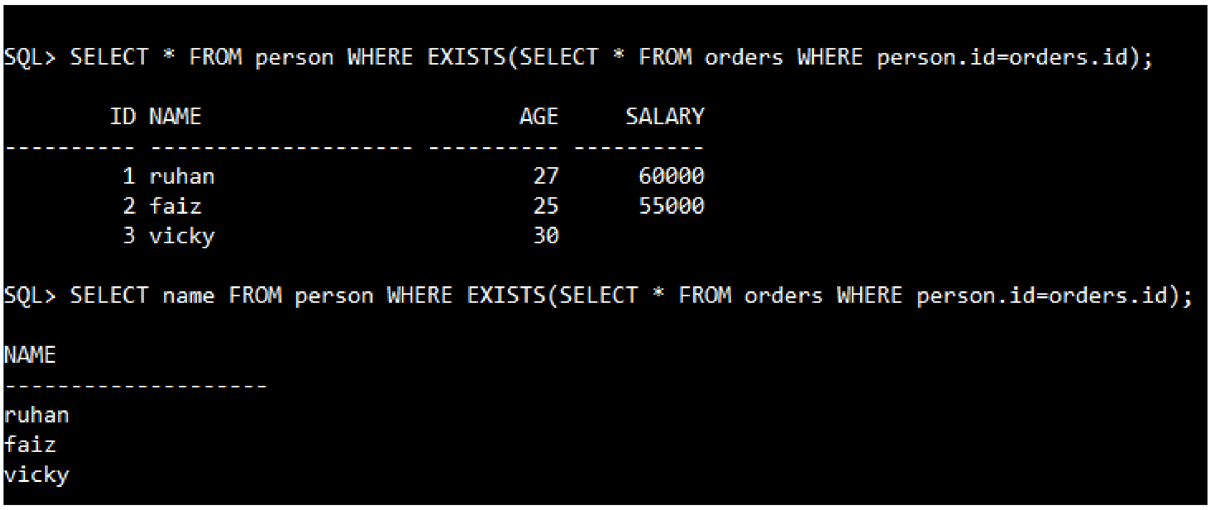


```
SQL> SELECT * FROM person WHERE salary IN(60000,55000,49000,30000);

        ID NAME                                  AGE     SALARY
---------- -------------------- ---------- ----------
         1 ruhan                                  27      60000
         2 faiz                                   25      55000

SQL> SELECT * FROM person WHERE age IN(27,30,58);

        ID NAME                                  AGE     SALARY
---------- -------------------- ---------- ----------
         1 ruhan                                  27      60000
         3 vicky                                  30
```



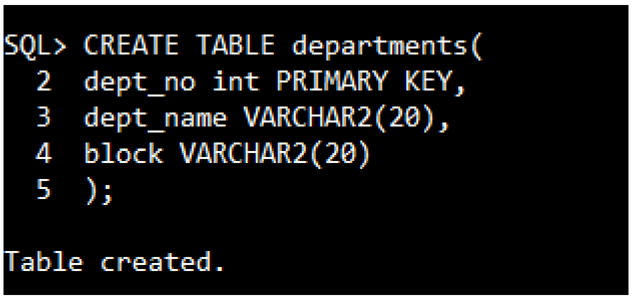
```
SQL> CREATE TABLE orders(
  2  o_id VARCHAR2(20),
  3  id int,
  4  FOREIGN KEY(id) REFERENCES person(id)
  5  );

Table created.

SQL> INSERT INTO orders VALUES('a1',1);

1 row created.

SQL> INSERT INTO orders VALUES('a2',2);

1 row created.

SQL> INSERT INTO orders VALUES('a3',3);

1 row created.

SQL> SELECT * FROM orders;

O_ID                         ID
-------------------- ----------
a1                            1
a2                            2
a3                            3
```

EXISTS

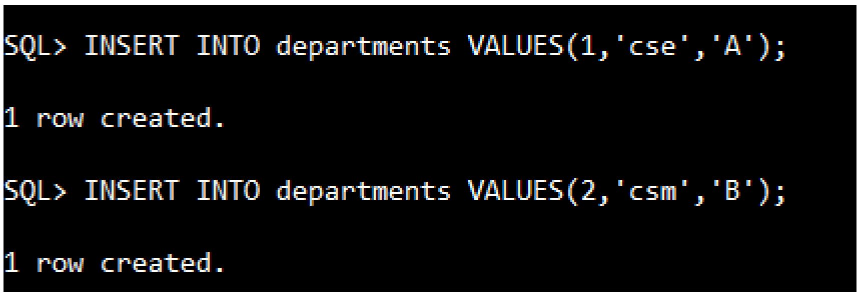


```
SQL> SELECT * FROM person WHERE EXISTS(SELECT * FROM orders WHERE person.id=orders.id);

        ID NAME                        AGE     SALARY
---------- -------------------- ---------- ----------
         1 ruhan                        27      60000
         2 faiz                         25      55000
         3 vicky                        30

SQL> SELECT name FROM person WHERE EXISTS(SELECT * FROM orders WHERE person.id=orders.id);

NAME
--------------------
ruhan
faiz
vicky
```

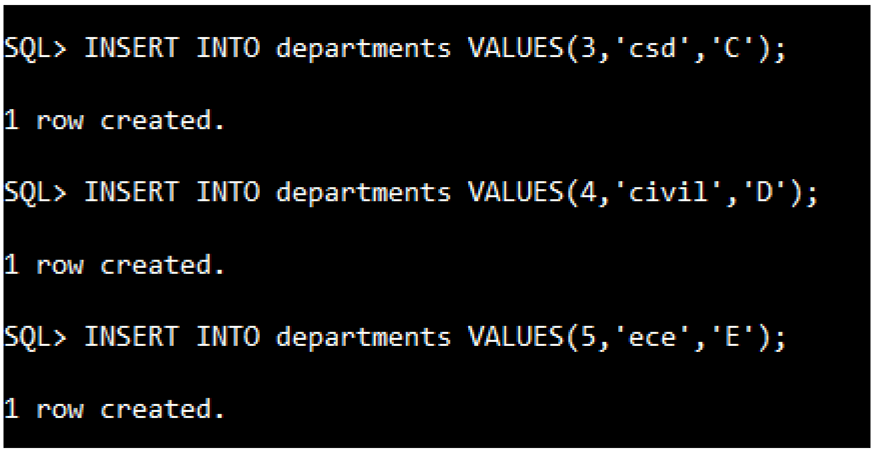
EXP6-JOIN OPERATIONS



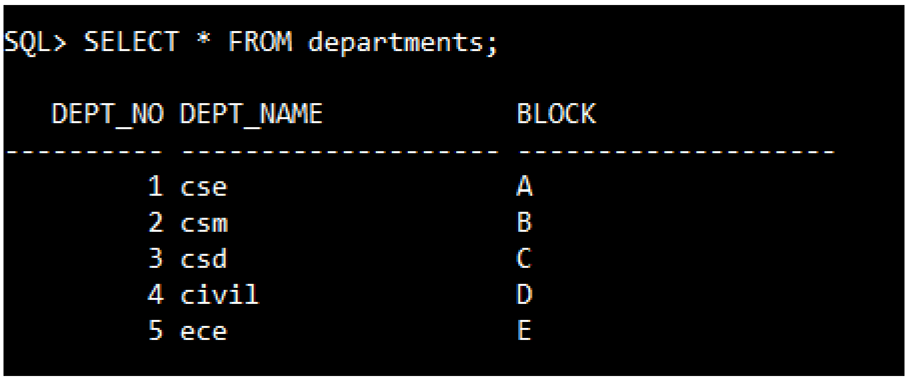
```
SQL> CREATE TABLE departments(
  2  dept_no int PRIMARY KEY,
  3  dept_name VARCHAR2(20),
  4  block VARCHAR2(20)
  5  );

Table created.
```



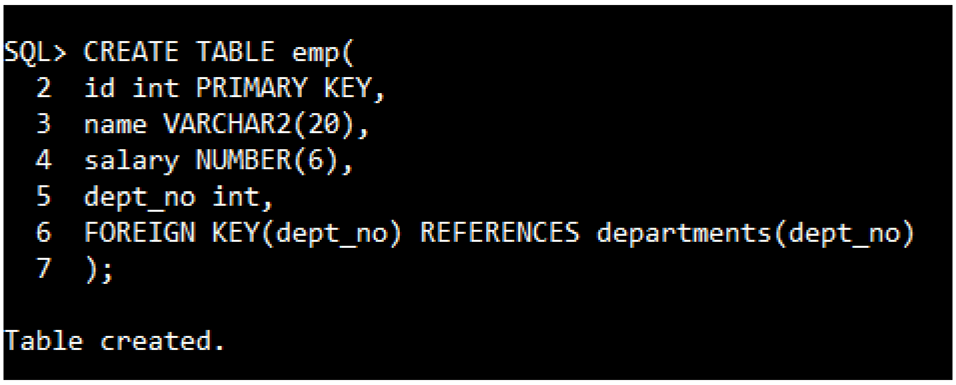
```
SQL> INSERT INTO departments VALUES(1,'cse','A');

1 row created.

SQL> INSERT INTO departments VALUES(2,'csm','B');

1 row created.
```



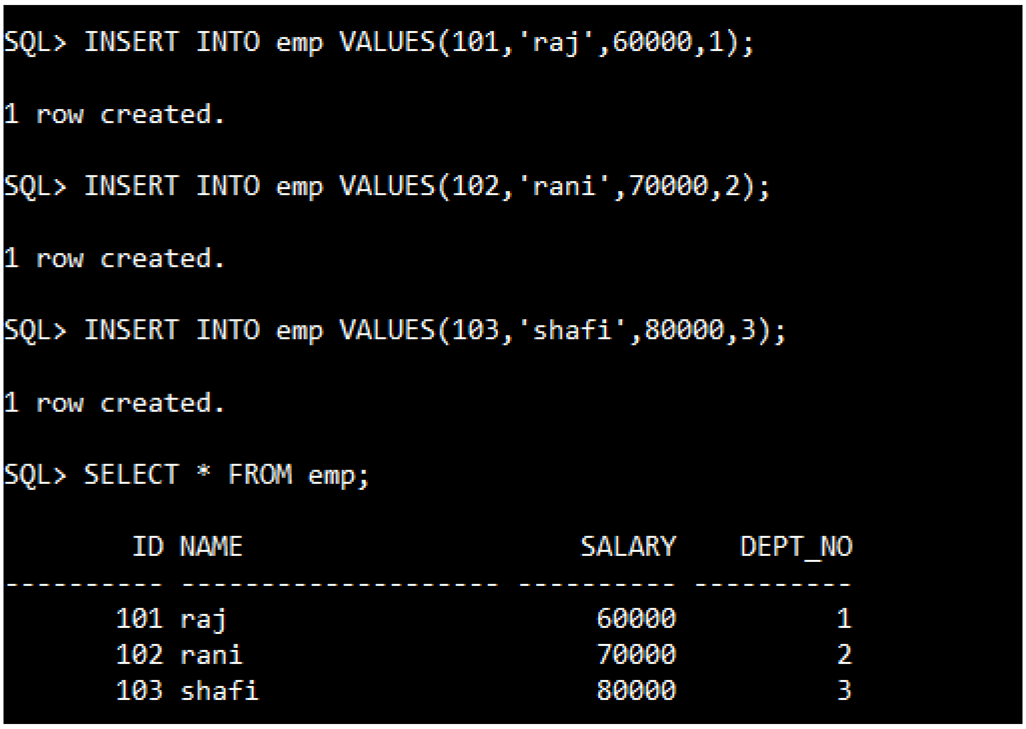
```
SQL> INSERT INTO departments VALUES(3,'csd','C');

1 row created.

SQL> INSERT INTO departments VALUES(4,'civil','D');

1 row created.

SQL> INSERT INTO departments VALUES(5,'ece','E');

1 row created.
```



```
SQL> SELECT * FROM departments;

   DEPT_NO DEPT_NAME            BLOCK
---------- -------------------- --------------------
         1 cse                  A
         2 csm                  B
         3 csd                  C
         4 civil                D
         5 ece                  E
```

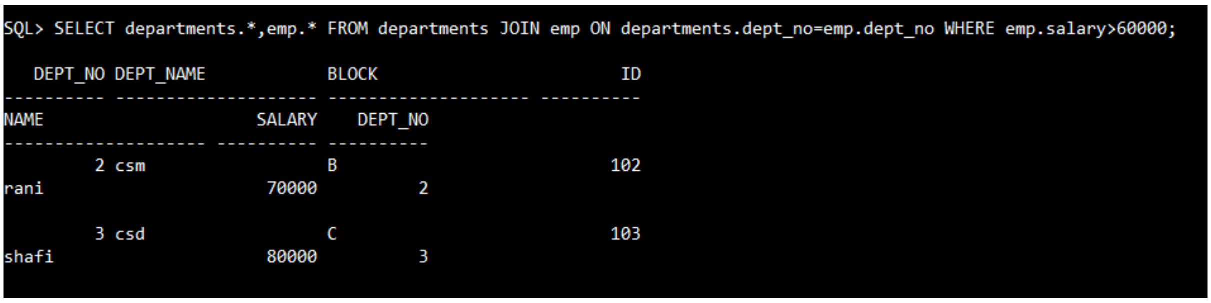


```
SQL> CREATE TABLE emp(
  2  id int PRIMARY KEY,
  3  name VARCHAR2(20),
  4  salary NUMBER(6),
  5  dept_no int,
  6  FOREIGN KEY(dept_no) REFERENCES departments(dept_no)
  7  );

Table created.
```



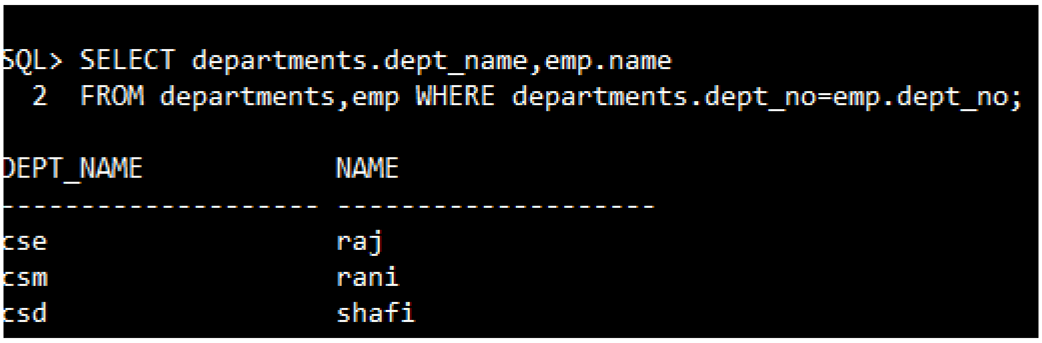
```
SQL> INSERT INTO emp VALUES(101,'raj',60000,1);

1 row created.

SQL> INSERT INTO emp VALUES(102,'rani',70000,2);

1 row created.

SQL> INSERT INTO emp VALUES(103,'shafi',80000,3);

1 row created.

SQL> SELECT * FROM emp;

        ID NAME                     SALARY    DEPT_NO
---------- -------------------- ---------- ----------
       101 raj                       60000          1
       102 rani                      70000          2
       103 shafi                     80000          3
```

Conditional join:

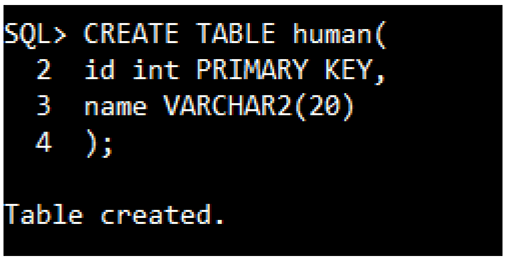


```
SQL> SELECT departments.*,emp.* FROM departments JOIN emp ON departments.dept_no=emp.dept_no WHERE emp.salary>60000;

   DEPT_NO DEPT_NAME            BLOCK                        ID
---------- -------------------- -------------------- ----------
NAME                     SALARY    DEPT_NO
-------------------- ---------- ----------
         2 csm                  B                           102
rani                      70000          2

         3 csd                  C                           103
shafi                     80000          3
```

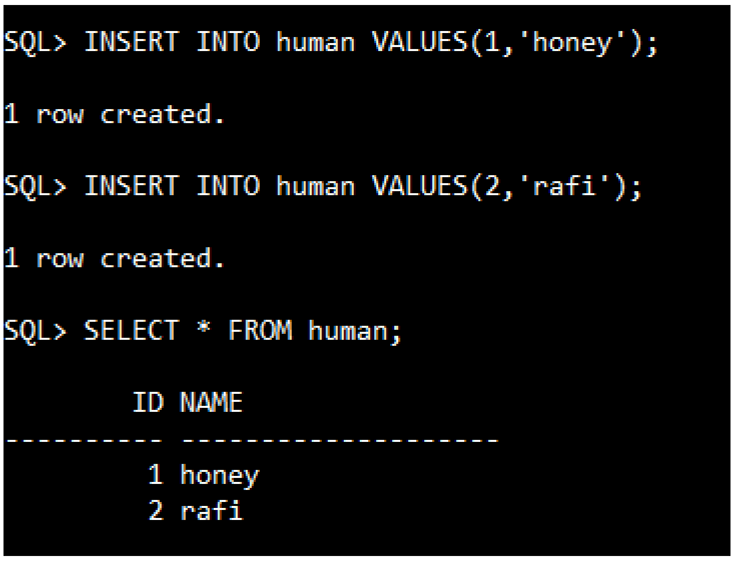
Equi join:



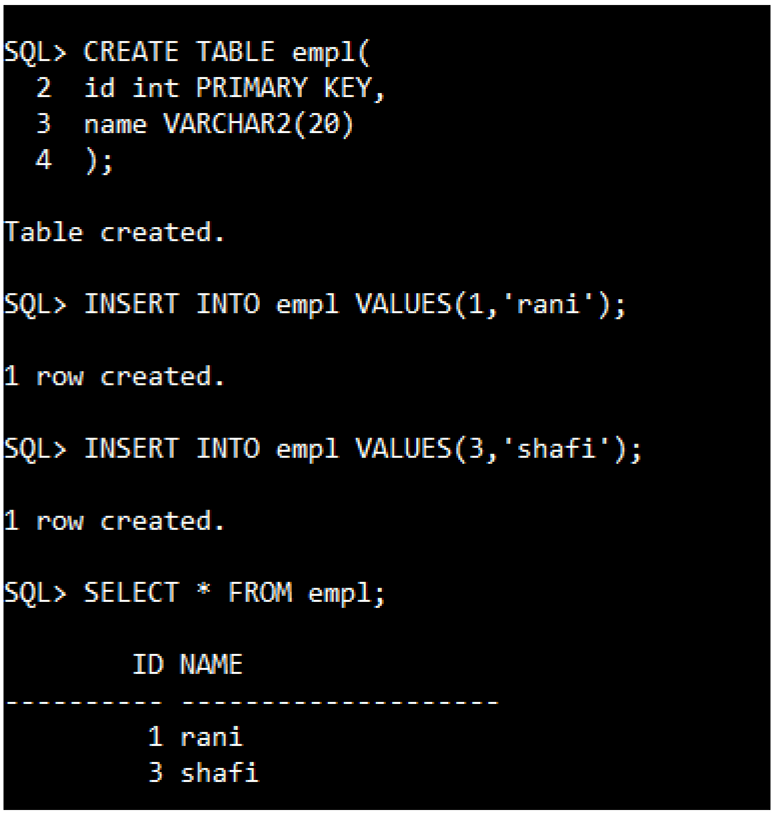
```
SQL> SELECT departments.dept_name,emp.name
  2  FROM departments,emp WHERE departments.dept_no=emp.dept_no;

DEPT_NAME            NAME
-------------------- --------------------
cse                  raj
csm                  rani
csd                  shafi
```



```
SQL> CREATE TABLE human(
  2  id int PRIMARY KEY,
  3  name VARCHAR2(20)
  4  );

Table created.
```

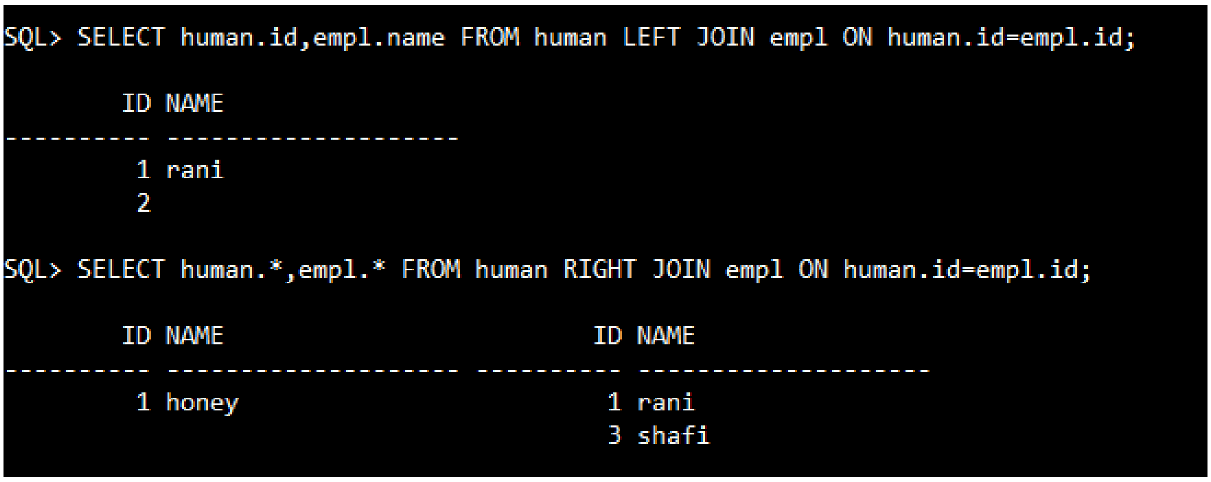


```
SQL> INSERT INTO human VALUES(1,'honey');

1 row created.

SQL> INSERT INTO human VALUES(2,'rafi');

1 row created.

SQL> SELECT * FROM human;

        ID NAME
---------- --------------------
         1 honey
         2 rafi
```



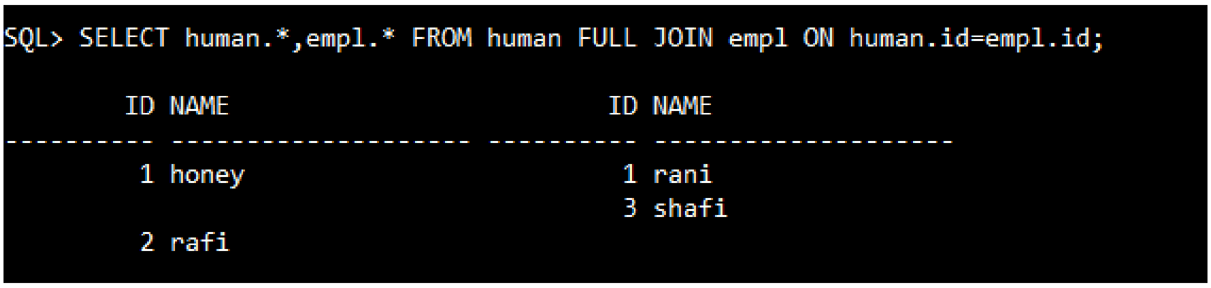
```
SQL> CREATE TABLE empl(
  2  id int PRIMARY KEY,
  3  name VARCHAR2(20)
  4  );

Table created.

SQL> INSERT INTO empl VALUES(1,'rani');

1 row created.

SQL> INSERT INTO empl VALUES(3,'shafi');

1 row created.

SQL> SELECT * FROM empl;

        ID NAME
---------- --------------------
         1 rani
         3 shafi
```

Left outer join and Right outer join:



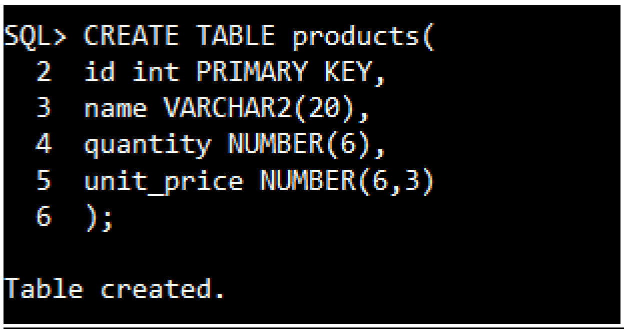
```
SQL> SELECT human.id,empl.name FROM human LEFT JOIN empl ON human.id=empl.id;

        ID NAME
---------- --------------------
         1 rani
         2

SQL> SELECT human.*,empl.* FROM human RIGHT JOIN empl ON human.id=empl.id;

        ID NAME                         ID NAME
---------- -------------------- ---------- --------------------
         1 honey                         1 rani
                                         3 shafi
```

Full outer join:

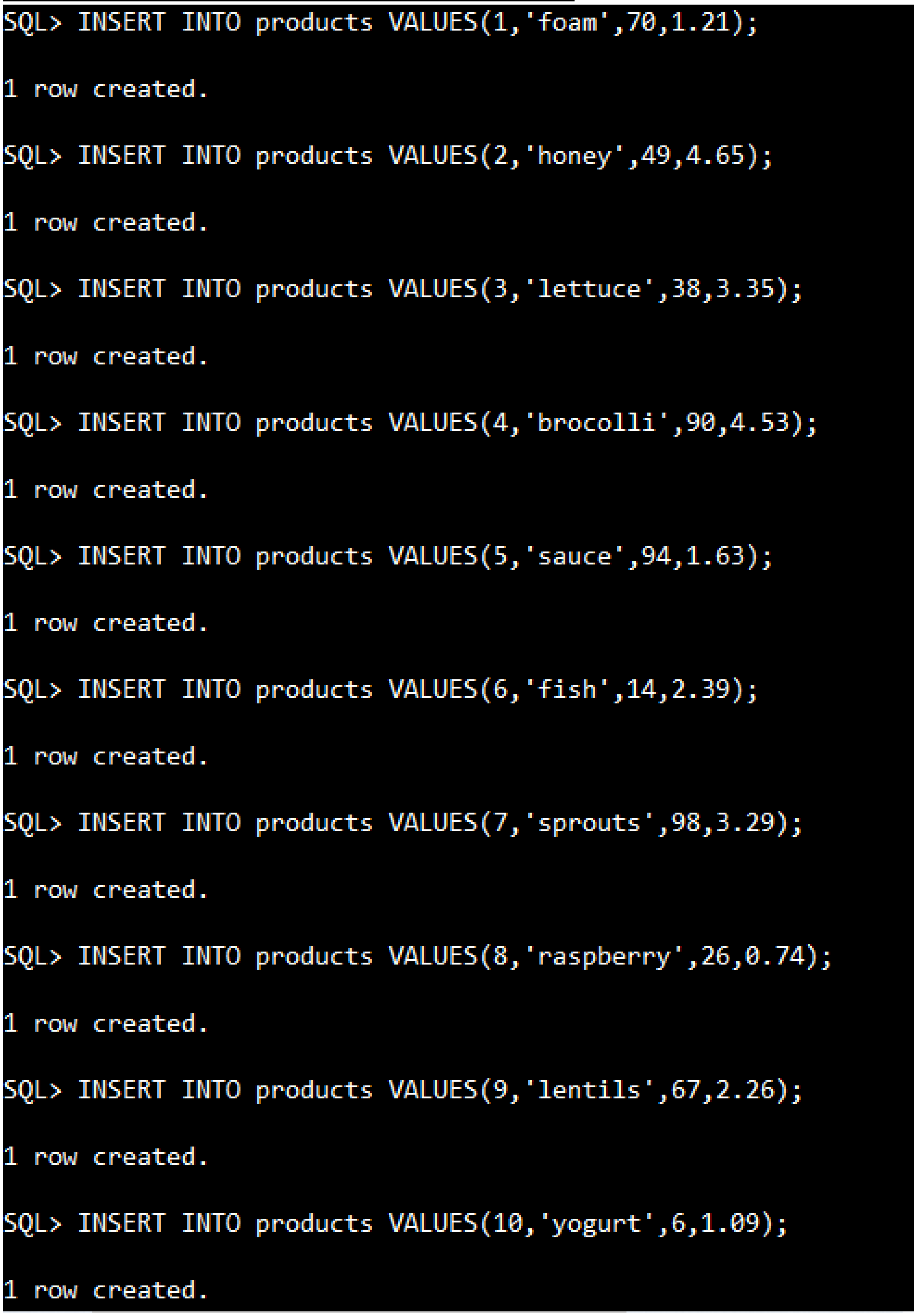


```
SQL> SELECT human.*,empl.* FROM human FULL JOIN empl ON human.id=empl.id;

        ID NAME                         ID NAME
---------- -------------------- ---------- --------------------
         1 honey                         1 rani
                                         3 shafi

         2 rafi
```

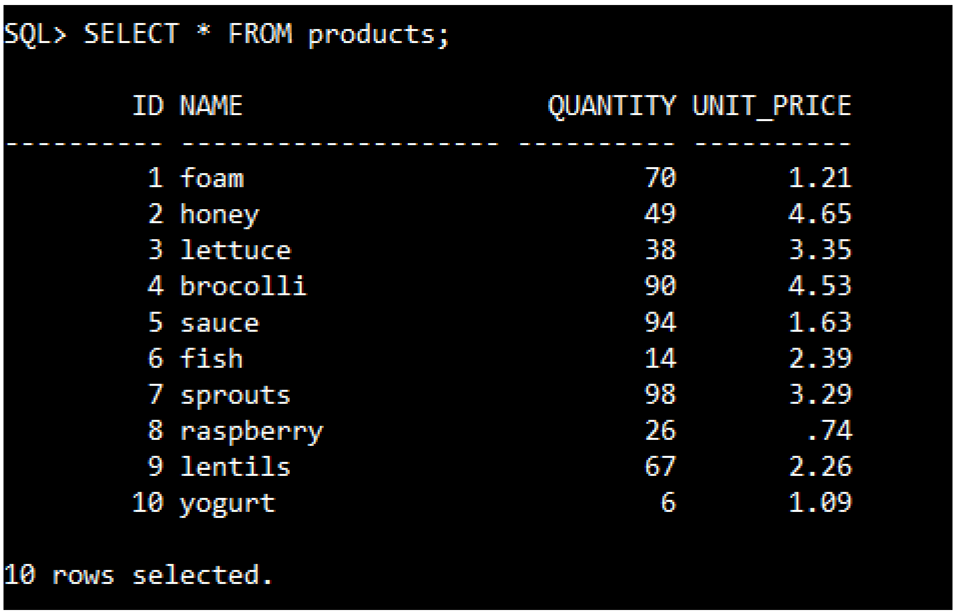
EXP7-AGGREGATE FUNCTIONS



```
SQL> CREATE TABLE products(
id int PRIMARY KEY,
name VARCHAR2(20),
quantity NUMBER(6),
unit_price NUMBER(6,3)
);

Table created.
```

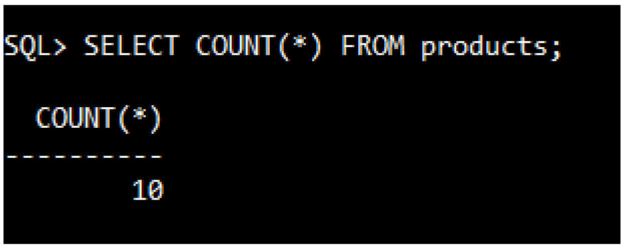


```
SQL> INSERT INTO products VALUES(1,'foam',70,1.21);

1 row created.

SQL> INSERT INTO products VALUES(2,'honey',49,4.65);

1 row created.

SQL> INSERT INTO products VALUES(3,'lettuce',38,3.35);

1 row created.

SQL> INSERT INTO products VALUES(4,'brocolli',90,4.53);

1 row created.

SQL> INSERT INTO products VALUES(5,'sauce',94,1.63);

1 row created.

SQL> INSERT INTO products VALUES(6,'fish',14,2.39);

1 row created.

SQL> INSERT INTO products VALUES(7,'sprouts',98,3.29);

1 row created.

SQL> INSERT INTO products VALUES(8,'raspberry',26,0.74);

1 row created.

SQL> INSERT INTO products VALUES(9,'lentils',67,2.26);

1 row created.

SQL> INSERT INTO products VALUES(10,'yogurt',6,1.09);

1 row created.
```

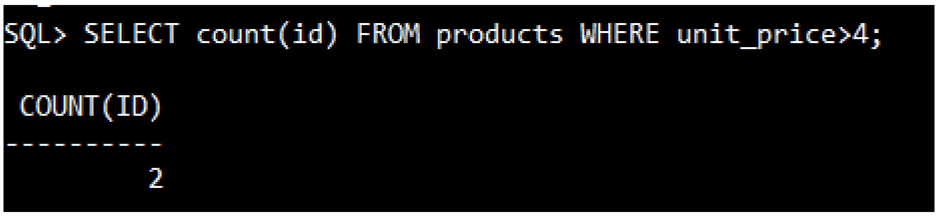


```
SQL> SELECT * FROM products;

        ID NAME                   QUANTITY UNIT_PRICE
---------- -------------------- ---------- ----------
         1 foam                         70       1.21
         2 honey                        49       4.65
         3 lettuce                      38       3.35
         4 brocolli                     90       4.53
         5 sauce                        94       1.63
         6 fish                         14       2.39
         7 sprouts                      98       3.29
         8 raspberry                    26        .74
         9 lentils                      67       2.26
        10 yogurt                        6       1.09

10 rows selected.
```

Count:

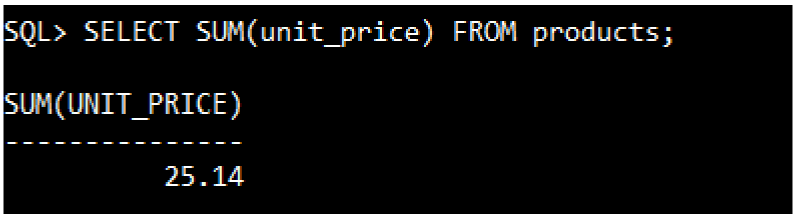


```
SQL> SELECT COUNT(*) FROM products;

  COUNT(*)
----------
        10
```

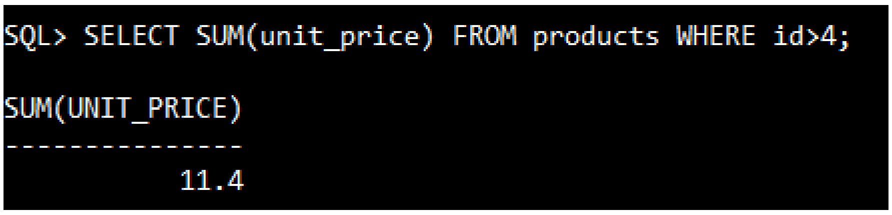


```
SQL> SELECT count(id) FROM products WHERE unit_price>4;

 COUNT(ID)
----------
         2
```

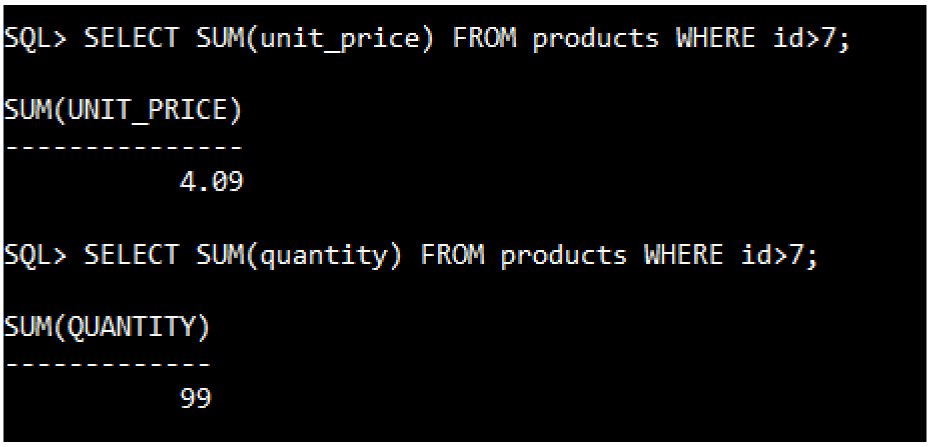
Sum:



```
SQL> SELECT SUM(unit_price) FROM products;

SUM(UNIT_PRICE)
---------------
          25.14
```

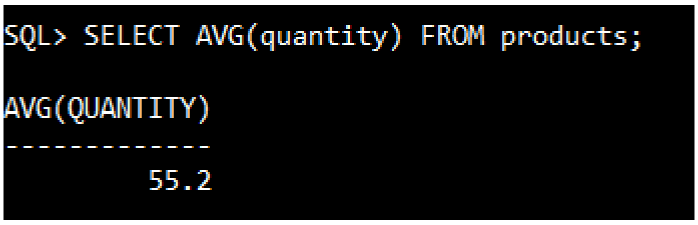


```
SQL> SELECT SUM(unit_price) FROM products WHERE id>4;

SUM(UNIT_PRICE)
---------------
           11.4
```

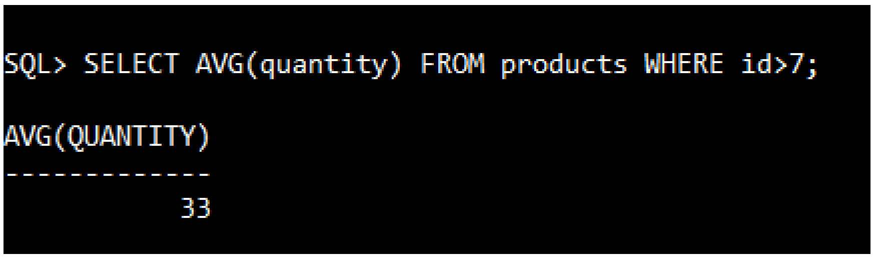


```
SQL> SELECT SUM(unit_price) FROM products WHERE id>7;

SUM(UNIT_PRICE)
---------------
           4.09

SQL> SELECT SUM(quantity) FROM products WHERE id>7;

SUM(QUANTITY)
-------------
           99
```

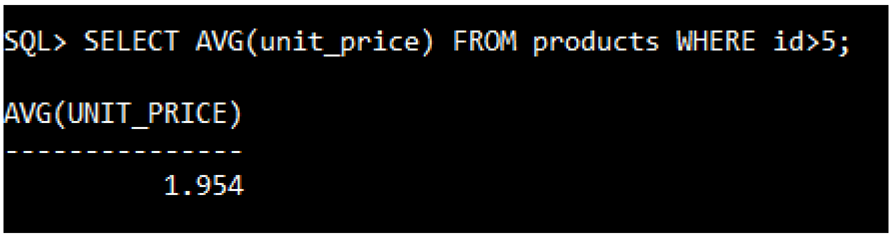
Average:



```
SQL> SELECT AVG(quantity) FROM products;

AVG(QUANTITY)
-------------
         55.2
```

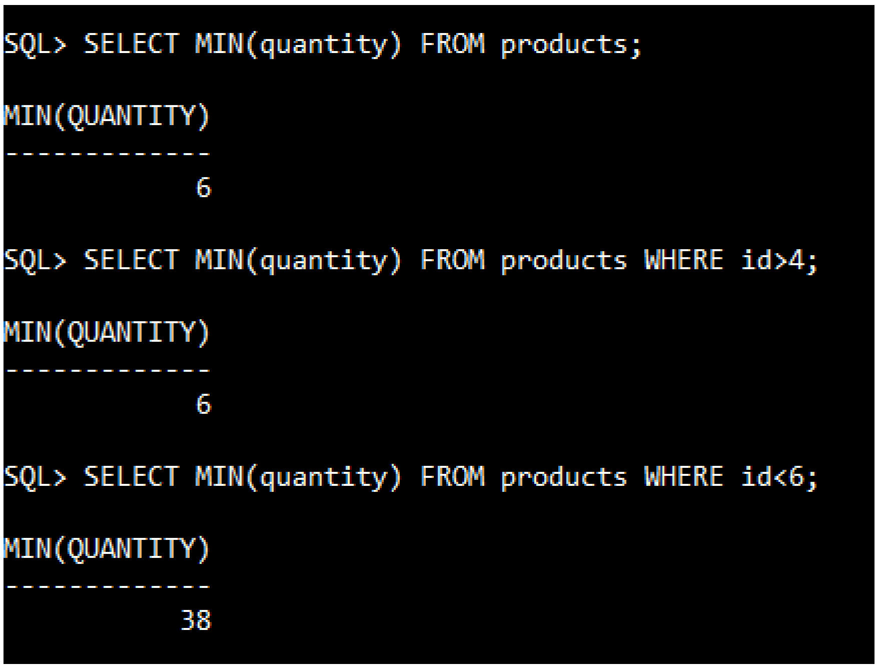


```
SQL> SELECT AVG(quantity) FROM products WHERE id>7;

AVG(QUANTITY)
-------------
           33
```



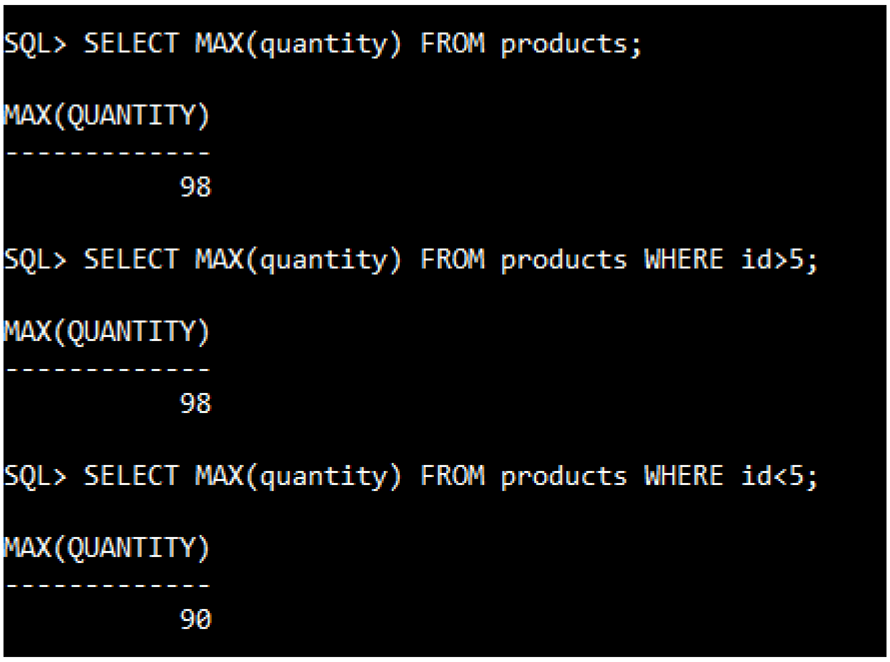
```
SQL> SELECT AVG(unit_price) FROM products WHERE id>5;

AVG(UNIT_PRICE)
---------------
          1.954
```

Minimum:

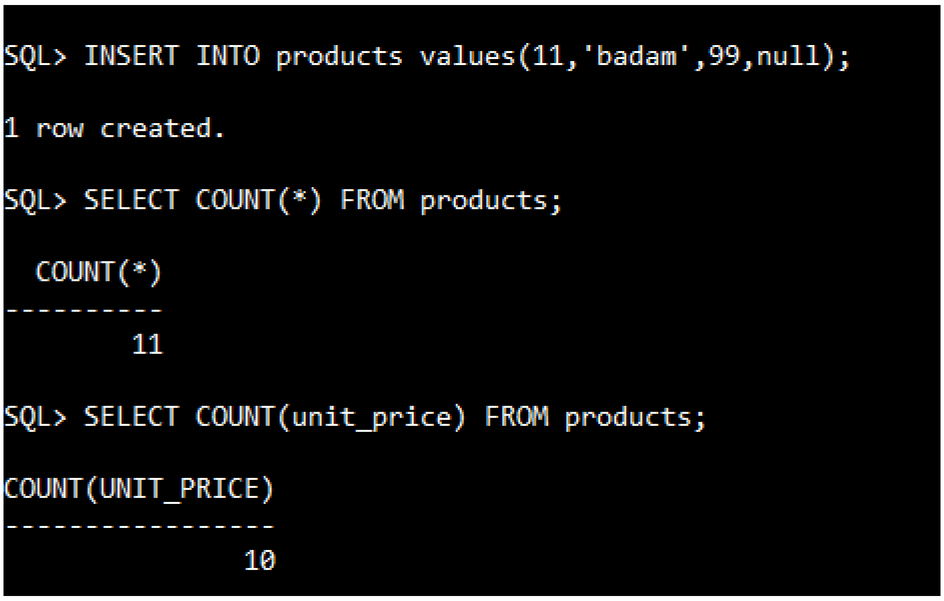


```
SQL> SELECT MIN(quantity) FROM products;

MIN(QUANTITY)
-------------
            6

SQL> SELECT MIN(quantity) FROM products WHERE id>4;

MIN(QUANTITY)
-------------
            6

SQL> SELECT MIN(quantity) FROM products WHERE id<6;

MIN(QUANTITY)
-------------
           38
```

Maximum:

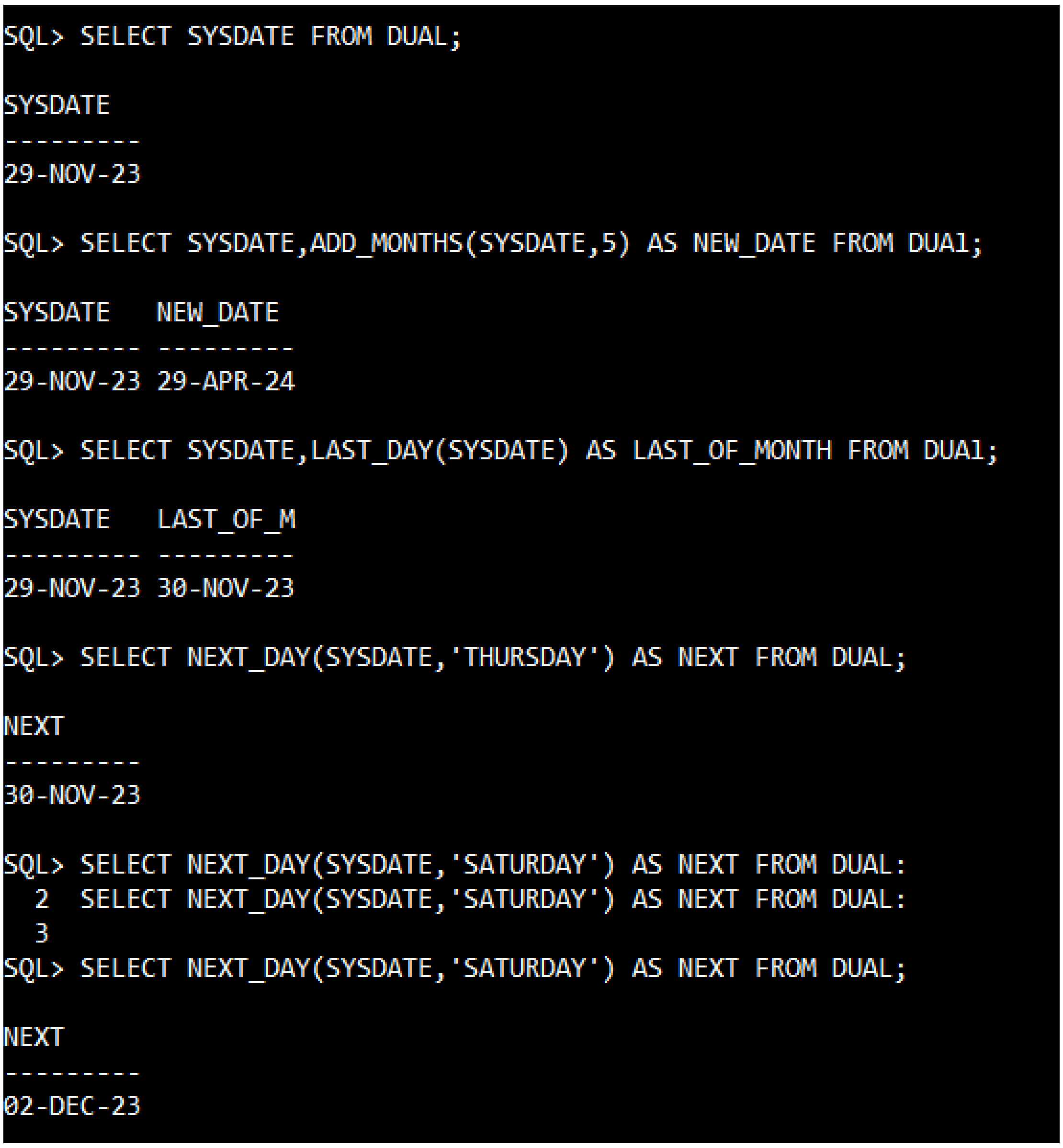


```
SQL> SELECT MAX(quantity) FROM products;

MAX(QUANTITY)
-------------
           98

SQL> SELECT MAX(quantity) FROM products WHERE id>5;

MAX(QUANTITY)
-------------
           98

SQL> SELECT MAX(quantity) FROM products WHERE id<5;

MAX(QUANTITY)
-------------
           90
```

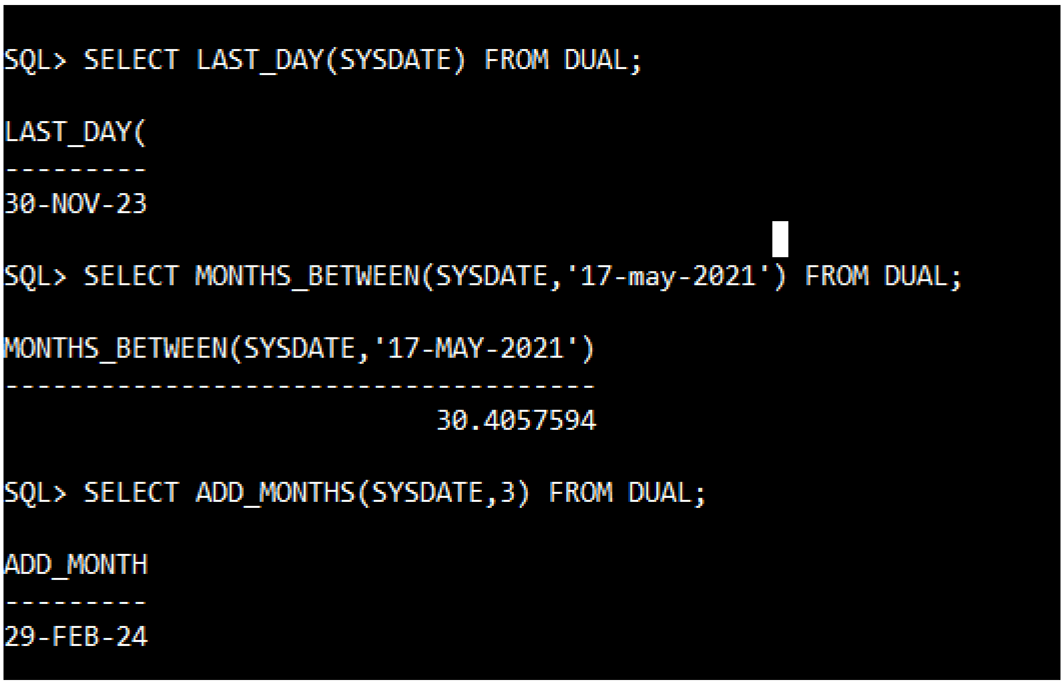


```
SQL> INSERT INTO products values(11,'badam',99,null);

1 row created.

SQL> SELECT COUNT(*) FROM products;

  COUNT(*)
----------
        11

SQL> SELECT COUNT(unit_price) FROM products;

COUNT(UNIT_PRICE)
-----------------
               10
```

EXP8-BUILTIN FUNCTIONS



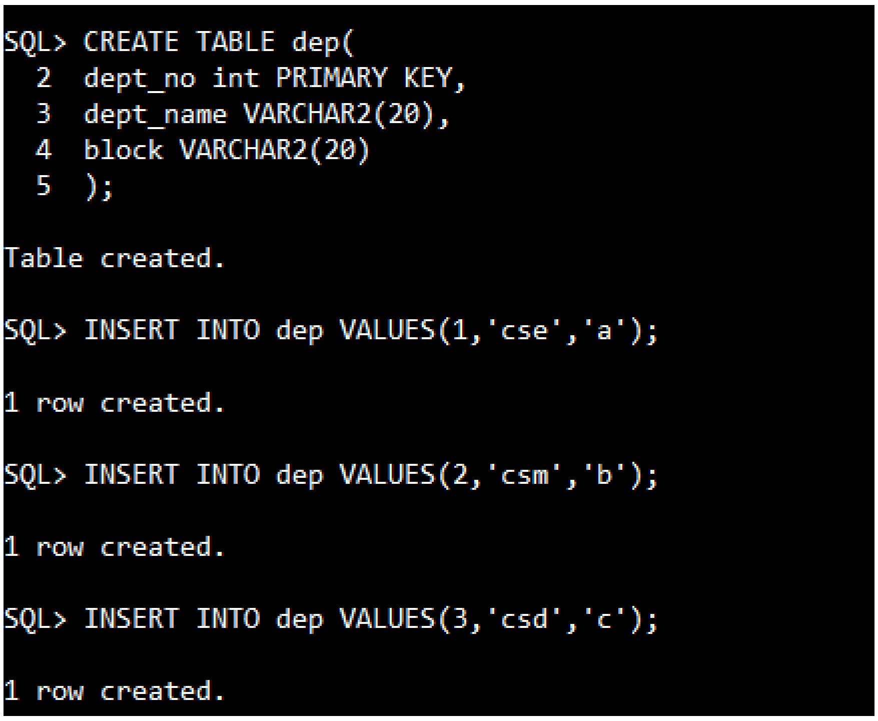
```
SQL> SELECT SYSDATE FROM DUAL;

SYSDATE
---------
29-NOV-23

SQL> SELECT SYSDATE,ADD_MONTHS(SYSDATE,5) AS NEW_DATE FROM DUA1;

SYSDATE   NEW_DATE
--------- ---------
29-NOV-23 29-APR-24

SQL> SELECT SYSDATE,LAST_DAY(SYSDATE) AS LAST_OF_MONTH FROM DUA1;

SYSDATE   LAST_OF_M
--------- ---------
29-NOV-23 30-NOV-23

SQL> SELECT NEXT_DAY(SYSDATE,'THURSDAY') AS NEXT FROM DUAL;

NEXT
---------
30-NOV-23

SQL> SELECT NEXT_DAY(SYSDATE,'SATURDAY') AS NEXT FROM DUAL:
  2  SELECT NEXT_DAY(SYSDATE,'SATURDAY') AS NEXT FROM DUAL:
  3
SQL> SELECT NEXT_DAY(SYSDATE,'SATURDAY') AS NEXT FROM DUAL;

NEXT
---------
02-DEC-23
```



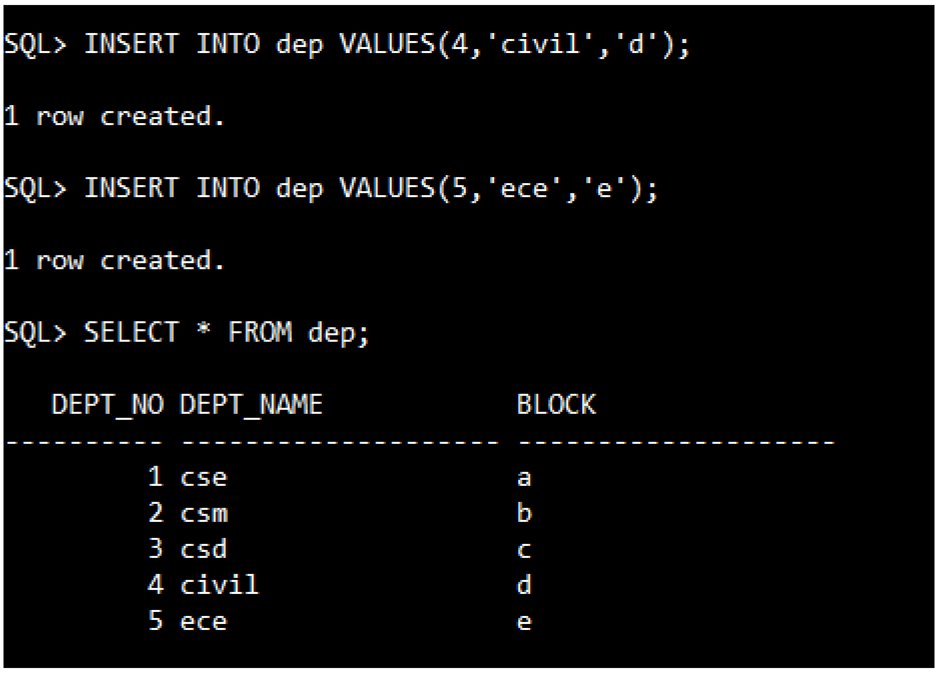
```
SQL> SELECT LAST_DAY(SYSDATE) FROM DUAL;

LAST_DAY(
---------
30-NOV-23

SQL> SELECT MONTHS_BETWEEN(SYSDATE,'17-may-2021') FROM DUAL;

MONTHS_BETWEEN(SYSDATE,'17-MAY-2021')
-------------------------------------
                           30.4057594

SQL> SELECT ADD_MONTHS(SYSDATE,3) FROM DUAL;

ADD_MONTH
---------
29-FEB-24
```

EXP9-KEY CONSTRAINTS

PRIMARY KEY:

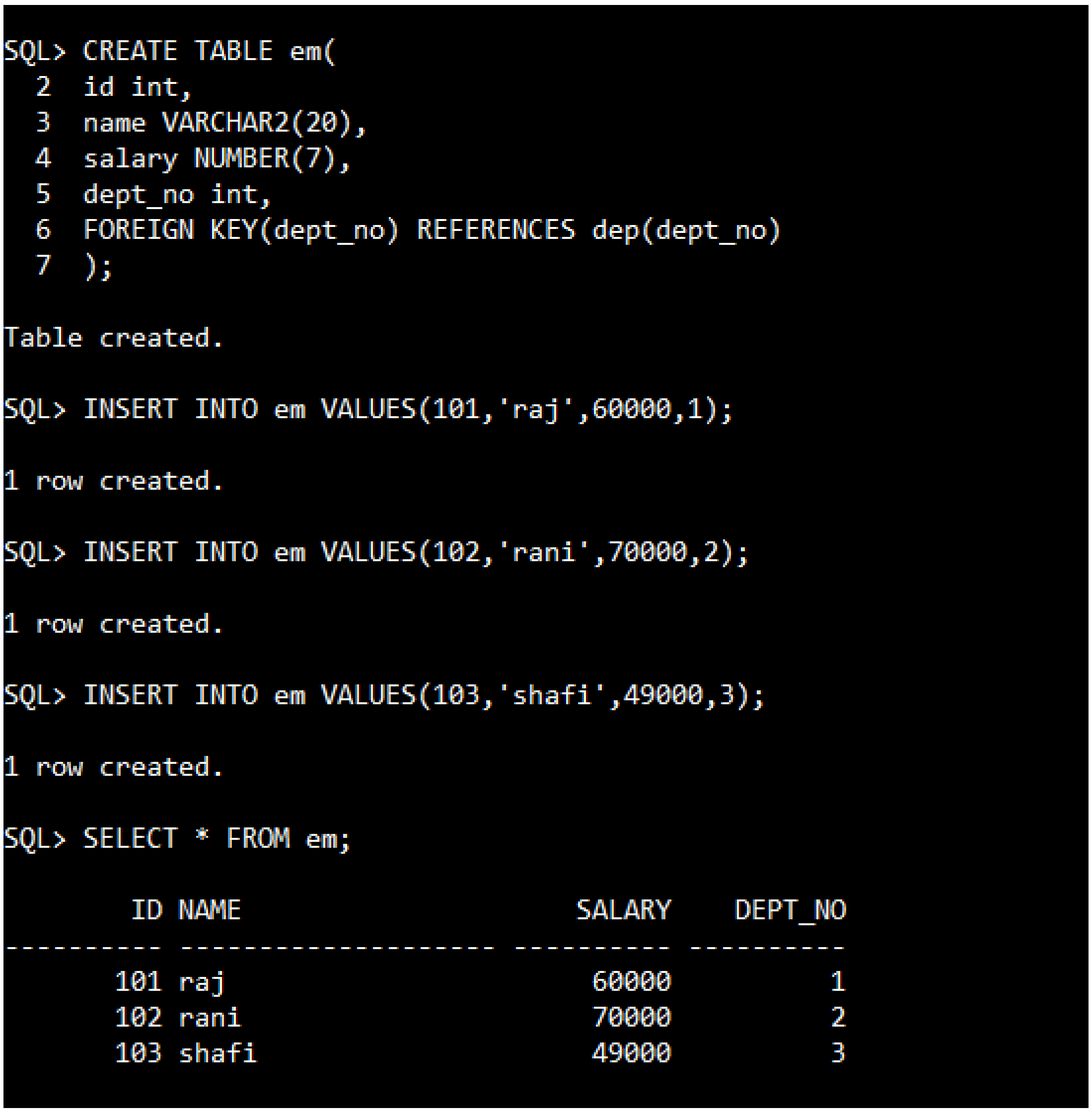


```
SQL> CREATE TABLE dep(
  2  dept_no int PRIMARY KEY,
  3  dept_name VARCHAR2(20),
  4  block VARCHAR2(20)
  5  );

Table created.

SQL> INSERT INTO dep VALUES(1,'cse','a');

1 row created.

SQL> INSERT INTO dep VALUES(2,'csm','b');

1 row created.

SQL> INSERT INTO dep VALUES(3,'csd','c');

1 row created.
```



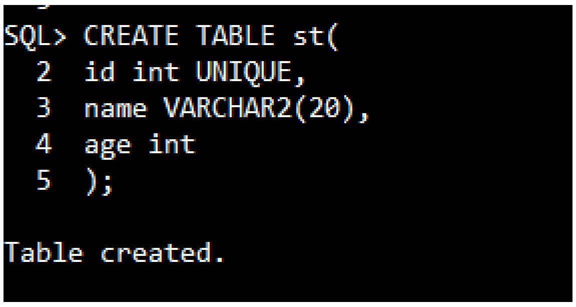
```
SQL> INSERT INTO dep VALUES(4,'civil','d');

1 row created.

SQL> INSERT INTO dep VALUES(5,'ece','e');

1 row created.

SQL> SELECT * FROM dep;

   DEPT_NO DEPT_NAME            BLOCK
---------- -------------------- --------------------
         1 cse                  a
         2 csm                  b
         3 csd                  c
         4 civil                d
         5 ece                  e
```

FOREIGN KEY:

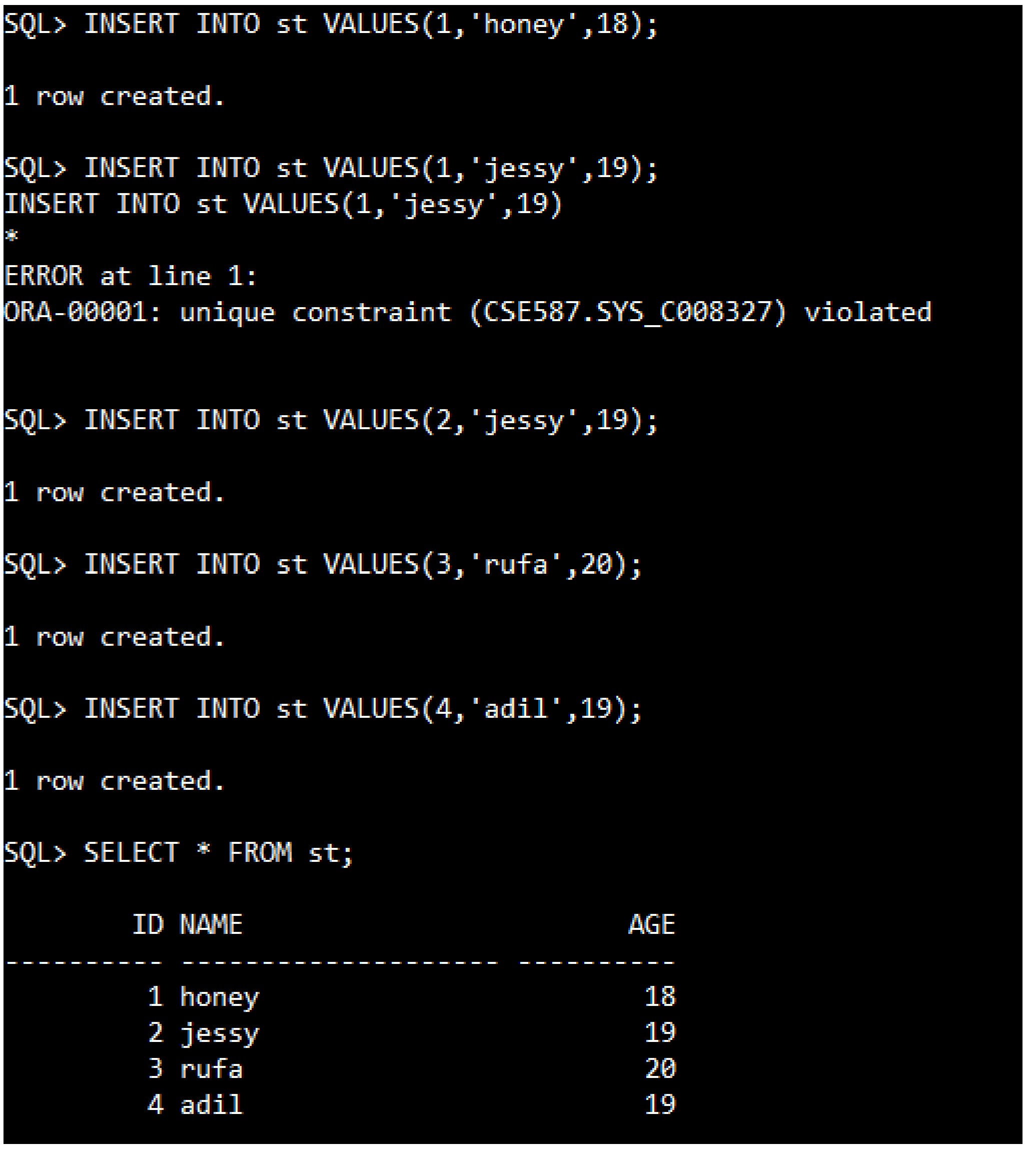


```
SQL> CREATE TABLE em(
  2  id int,
  3  name VARCHAR2(20),
  4  salary NUMBER(7),
  5  dept_no int,
  6  FOREIGN KEY(dept_no) REFERENCES dep(dept_no)
  7  );

Table created.

SQL> INSERT INTO em VALUES(101,'raj',60000,1);

1 row created.

SQL> INSERT INTO em VALUES(102,'rani',70000,2);

1 row created.

SQL> INSERT INTO em VALUES(103,'shafi',49000,3);

1 row created.

SQL> SELECT * FROM em;

        ID NAME                     SALARY    DEPT_NO
---------- -------------------- ---------- ----------
       101 raj                       60000          1
       102 rani                      70000          2
       103 shafi                     49000          3
```

UNIQUE KEY:

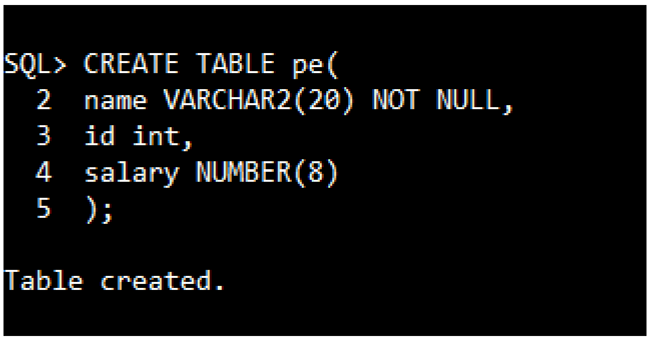


```
SQL> CREATE TABLE st(
  2  id int UNIQUE,
  3  name VARCHAR2(20),
  4  age int
  5  );

Table created.
```

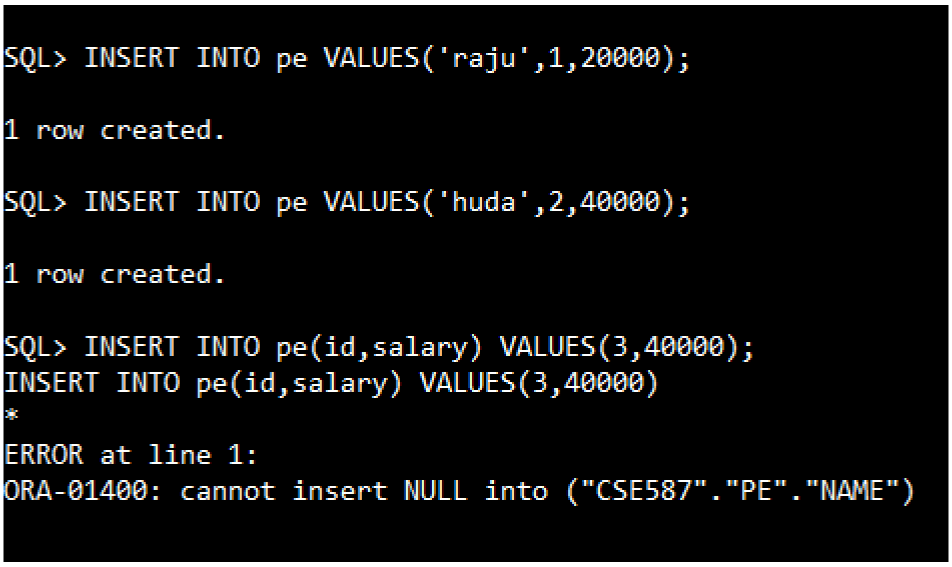


```
SQL> INSERT INTO st VALUES(1,'honey',18);

1 row created.

SQL> INSERT INTO st VALUES(1,'jessy',19);
INSERT INTO st VALUES(1,'jessy',19)
*
ERROR at line 1:
ORA-00001: unique constraint (CSE587.SYS_C008327) violated


SQL> INSERT INTO st VALUES(2,'jessy',19);

1 row created.

SQL> INSERT INTO st VALUES(3,'rufa',20);

1 row created.

SQL> INSERT INTO st VALUES(4,'adil',19);

1 row created.

SQL> SELECT * FROM st;

        ID NAME                        AGE
---------- -------------------- ----------
         1 honey                        18
         2 jessy                        19
         3 rufa                         20
         4 adil                         19
```

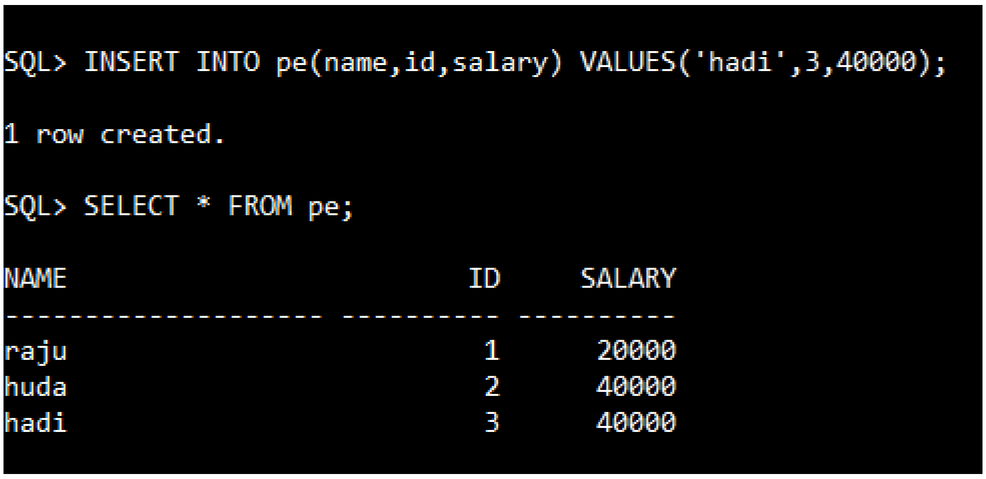
NOT NULL:



```
SQL> CREATE TABLE pe(
  2  name VARCHAR2(20) NOT NULL,
  3  id int,
  4  salary NUMBER(8)
  5  );

Table created.
```

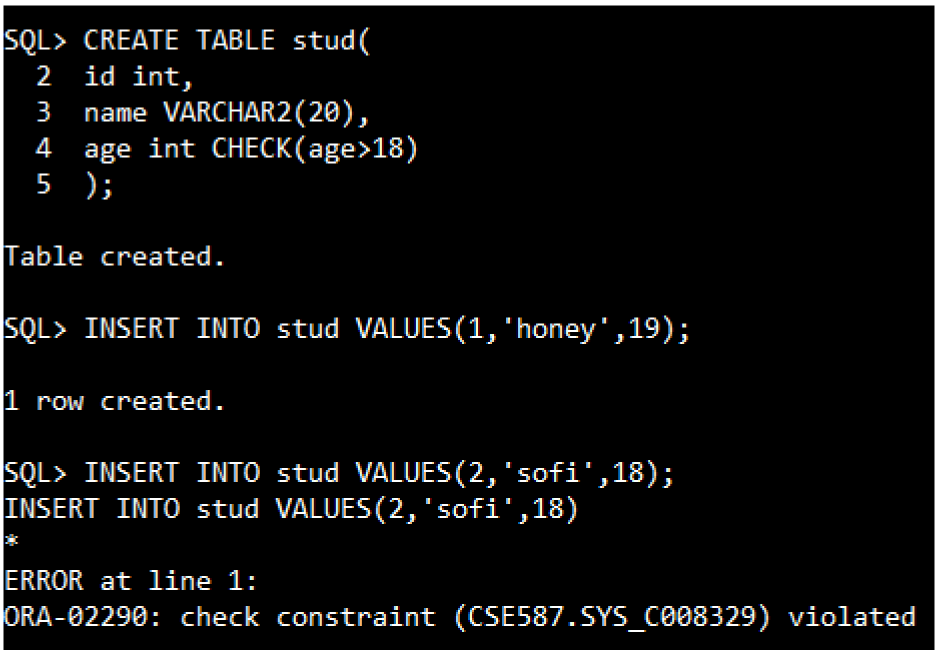


```
SQL> INSERT INTO pe VALUES('raju',1,20000);

1 row created.

SQL> INSERT INTO pe VALUES('huda',2,40000);

1 row created.

SQL> INSERT INTO pe(id,salary) VALUES(3,40000);
INSERT INTO pe(id,salary) VALUES(3,40000)
*
ERROR at line 1:
ORA-01400: cannot insert NULL into ("CSE587"."PE"."NAME")
```

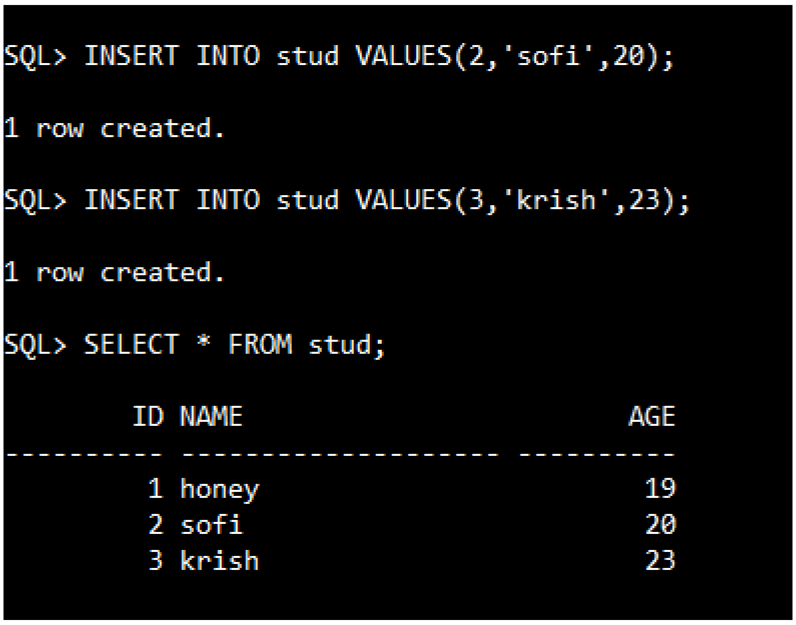


```
SQL> INSERT INTO pe(name,id,salary) VALUES('hadi',3,40000);

1 row created.

SQL> SELECT * FROM pe;

NAME                         ID     SALARY
-------------------- ---------- ----------
raju                          1      20000
huda                          2      40000
hadi                          3      40000
```

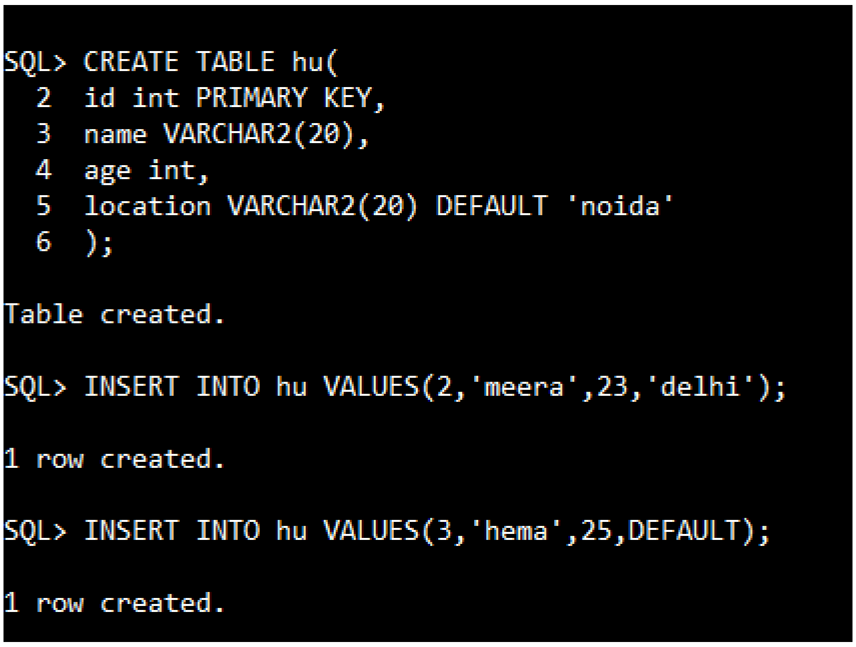
CHECK:

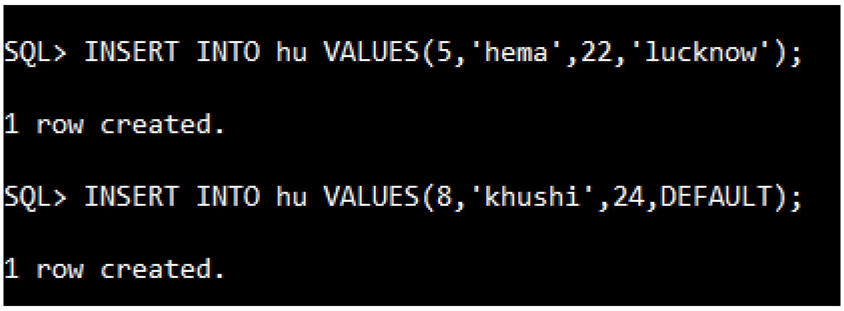


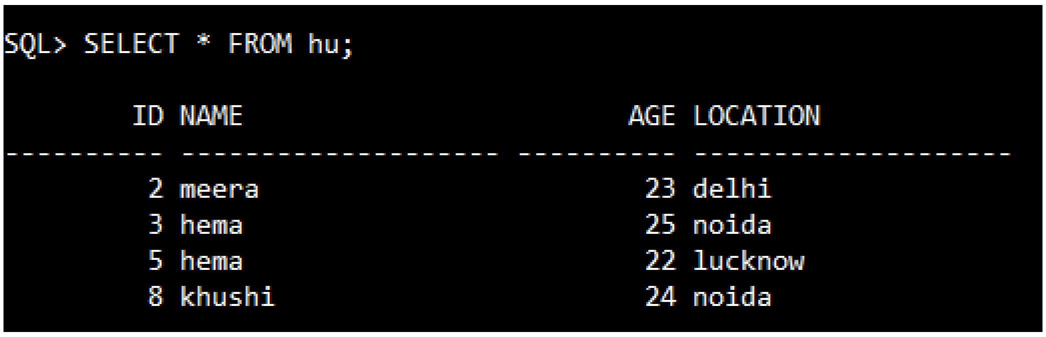
```
SQL> CREATE TABLE stud(
  2  id int,
  3  name VARCHAR2(20),
  4  age int CHECK(age>18)
  5  );

Table created.

SQL> INSERT INTO stud VALUES(1,'honey',19);

1 row created.

SQL> INSERT INTO stud VALUES(2,'sofi',18);
INSERT INTO stud VALUES(2,'sofi',18)
*
ERROR at line 1:
ORA-02290: check constraint (CSE587.SYS_C008329) violated
```


```
SQL> INSERT INTO stud VALUES(2,'sofi',20);

1 row created.

SQL> INSERT INTO stud VALUES(3,'krish',23);

1 row created.

SQL> SELECT * FROM stud;

        ID NAME                        AGE
---------- -------------------- ----------
         1 honey                        19
         2 sofi                         20
         3 krish                        23
```

DEFAULT:

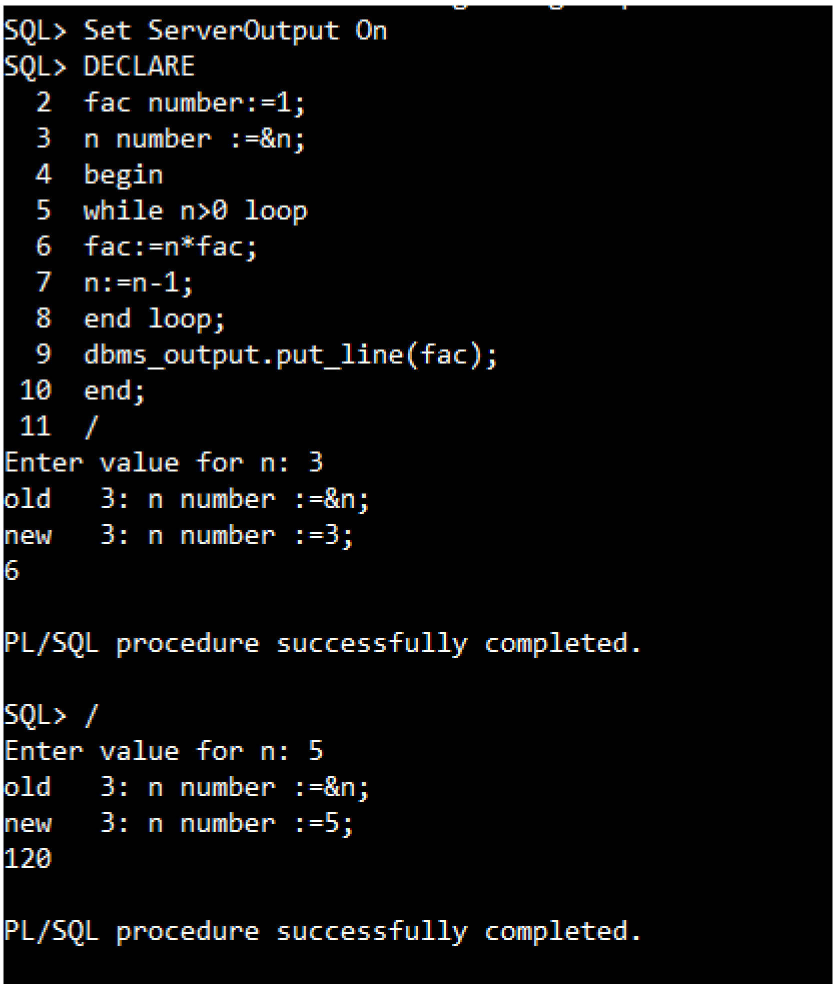

```
SQL> CREATE TABLE hu(
  2  id int PRIMARY KEY,
  3  name VARCHAR2(20),
  4  age int,
  5  location VARCHAR2(20) DEFAULT 'noida'
  6  );

Table created.

SQL> INSERT INTO hu VALUES(2,'meera',23,'delhi');

1 row created.

SQL> INSERT INTO hu VALUES(3,'hema',25,DEFAULT);

1 row created.
```


```
SQL> INSERT INTO hu VALUES(5,'hema',22,'lucknow');

1 row created.

SQL> INSERT INTO hu VALUES(8,'khushi',24,DEFAULT);

1 row created.
```



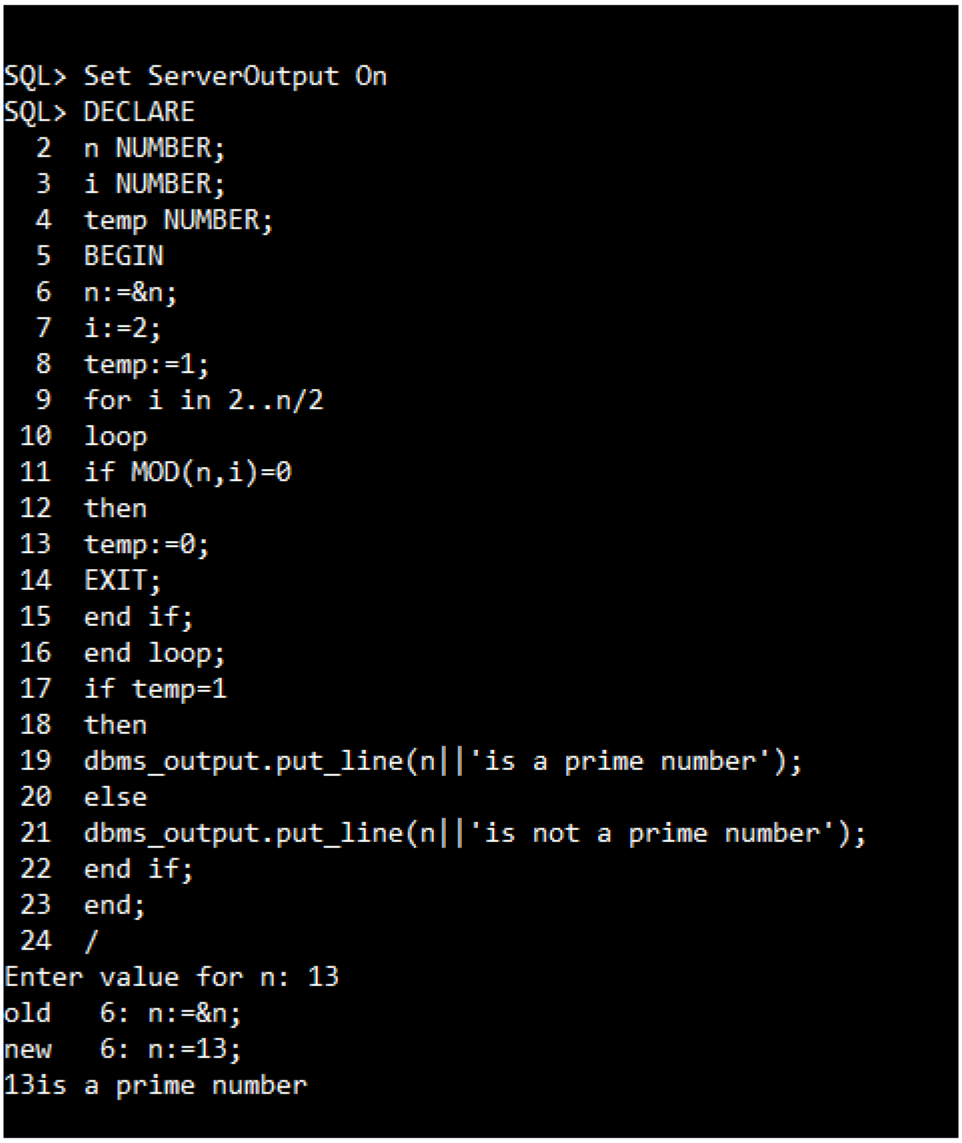
```
SQL> SELECT * FROM hu;

        ID NAME                                  AGE LOCATION
---------- -------------------- ---------- --------------------
         2 meera                                  23 delhi
         3 hema                                   25 noida
         5 hema                                   22 lucknow
         8 khushi                                 24 noida
```

EXP10-FACTORIAL



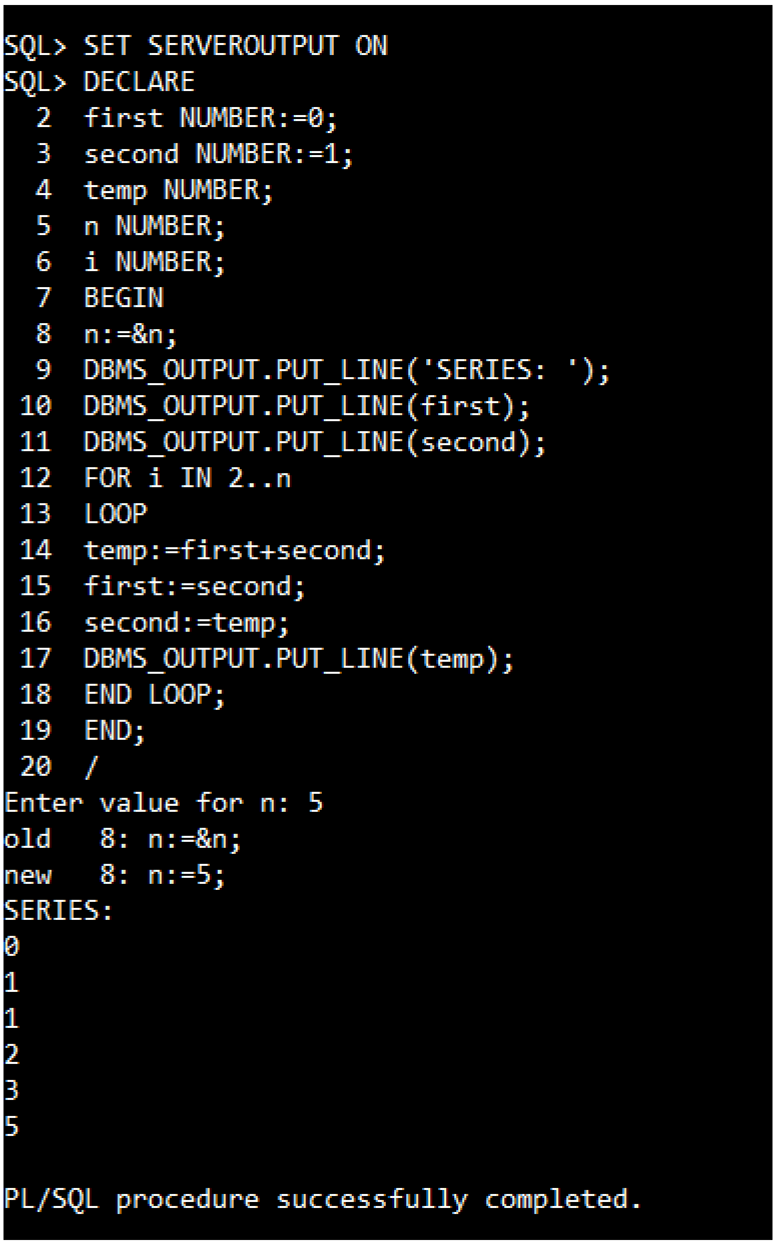
```
SQL> Set ServerOutput On
SQL> DECLARE
  2  fac number:=1;
  3  n number :=&n;
  4  begin
  5  while n>0 loop
  6  fac:=n*fac;
  7  n:=n-1;
  8  end loop;
  9  dbms_output.put_line(fac);
 10  end;
 11  /
Enter value for n: 3
old   3: n number :=&n;
new   3: n number :=3;
6

PL/SQL procedure successfully completed.

SQL> /
Enter value for n: 5
old   3: n number :=&n;
new   3: n number :=5;
120

PL/SQL procedure successfully completed.
```

EXP11-PRIME NUMBER



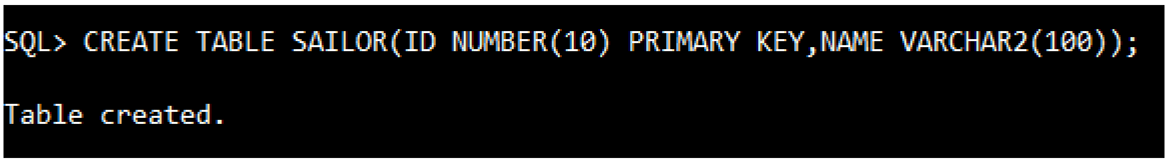
```
SQL> Set ServerOutput On
SQL> DECLARE
  2  n NUMBER;
  3  i NUMBER;
  4  temp NUMBER;
  5  BEGIN
  6  n:=&n;
  7  i:=2;
  8  temp:=1;
  9  for i in 2..n/2
 10  loop
 11  if MOD(n,i)=0
 12  then
 13  temp:=0;
 14  EXIT;
 15  end if;
 16  end loop;
 17  if temp=1
 18  then
 19  dbms_output.put_line(n||'is a prime number');
 20  else
 21  dbms_output.put_line(n||'is not a prime number');
 22  end if;
 23  end;
 24  /
Enter value for n: 13
old   6: n:=&n;
new   6: n:=13;
13is a prime number
```

EXP12-FIBONACCI SERIES



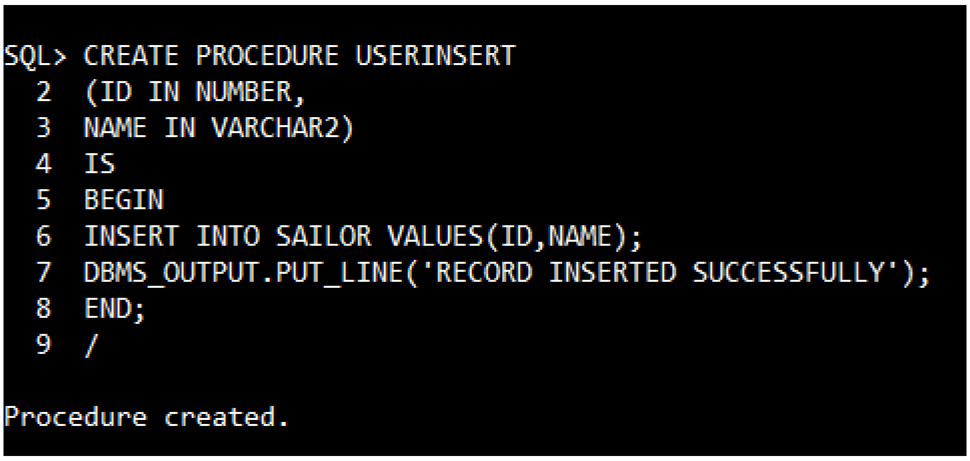
```
SQL> SET SERVEROUTPUT ON
SQL> DECLARE
  2  first NUMBER:=0;
  3  second NUMBER:=1;
  4  temp NUMBER;
  5  n NUMBER;
  6  i NUMBER;
  7  BEGIN
  8  n:=&n;
  9  DBMS_OUTPUT.PUT_LINE('SERIES: ');
 10  DBMS_OUTPUT.PUT_LINE(first);
 11  DBMS_OUTPUT.PUT_LINE(second);
 12  FOR i IN 2..n
 13  LOOP
 14  temp:=first+second;
 15  first:=second;
 16  second:=temp;
 17  DBMS_OUTPUT.PUT_LINE(temp);
 18  END LOOP;
 19  END;
 20  /
Enter value for n: 5
old   8: n:=&n;
new   8: n:=5;
SERIES:
0
1
1
2
3
5

PL/SQL procedure successfully completed.
```

EXP13-STORED PROCEDURE



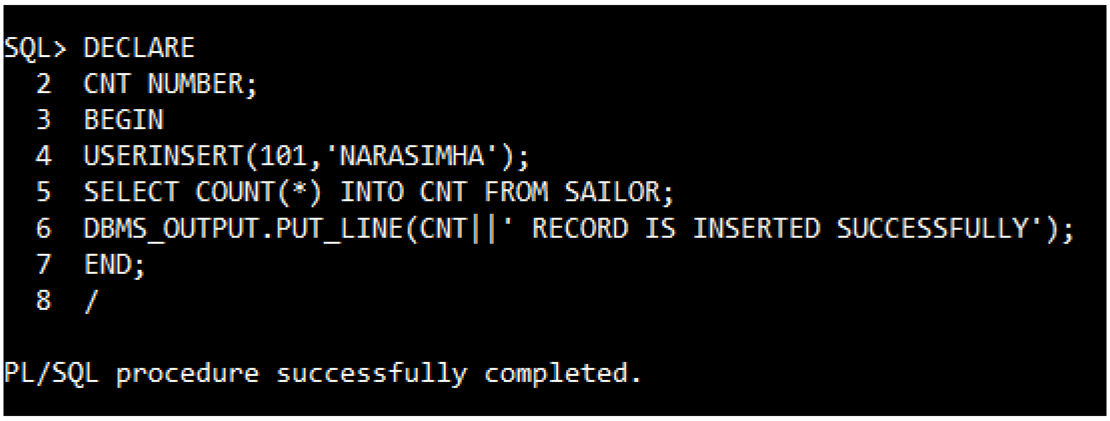
```
SQL> CREATE TABLE SAILOR(ID NUMBER(10) PRIMARY KEY,NAME VARCHAR2(100));

Table created.
```

Creating a procedure:



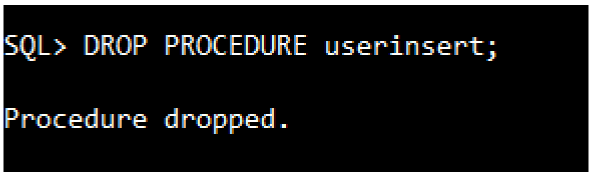
```
SQL> CREATE PROCEDURE USERINSERT
  2  (ID IN NUMBER,
  3  NAME IN VARCHAR2)
  4  IS
  5  BEGIN
  6  INSERT INTO SAILOR VALUES(ID,NAME);
  7  DBMS_OUTPUT.PUT_LINE('RECORD INSERTED SUCCESSFULLY');
  8  END;
  9  /

Procedure created.
```

Execution of procedure:



```
SQL> DECLARE
  2  CNT NUMBER;
  3  BEGIN
  4  USERINSERT(101,'NARASIMHA');
  5  SELECT COUNT(*) INTO CNT FROM SAILOR;
  6  DBMS_OUTPUT.PUT_LINE(CNT||' RECORD IS INSERTED SUCCESSFULLY');
  7  END;
  8  /

PL/SQL procedure successfully completed.
```

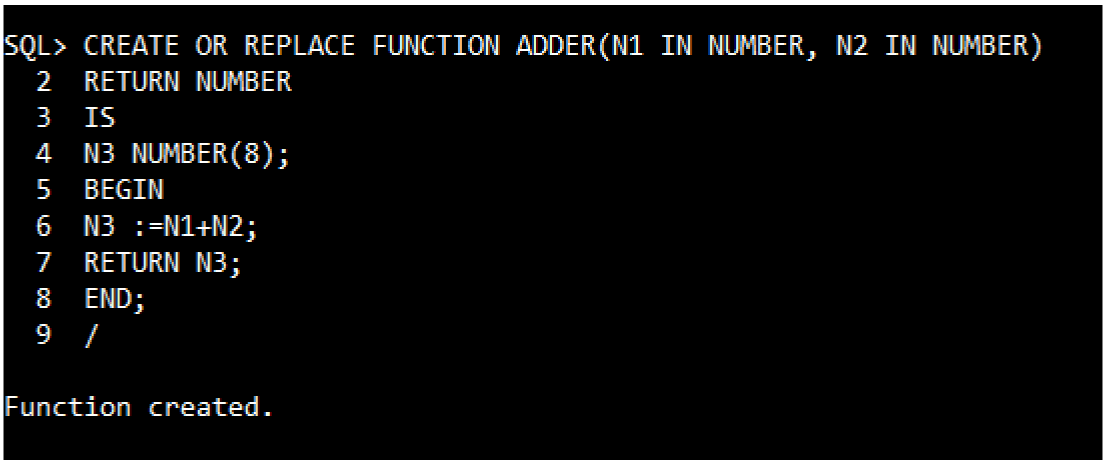
Drop procedure:



```
SQL> DROP PROCEDURE userinsert;

Procedure dropped.
```

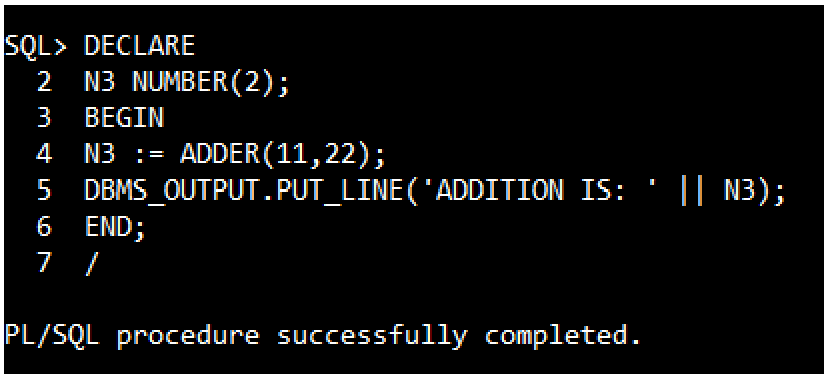
EXP14-STORED FUNCTION

Function creation:



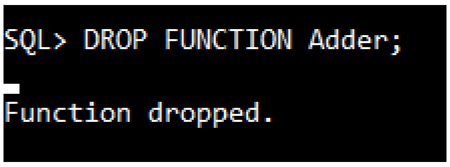
```
SQL> CREATE OR REPLACE FUNCTION ADDER(N1 IN NUMBER, N2 IN NUMBER)
RETURN NUMBER
IS
N3 NUMBER(8);
BEGIN
N3 :=N1+N2;
RETURN N3;
END;
/

Function created.
```

Executing a function:



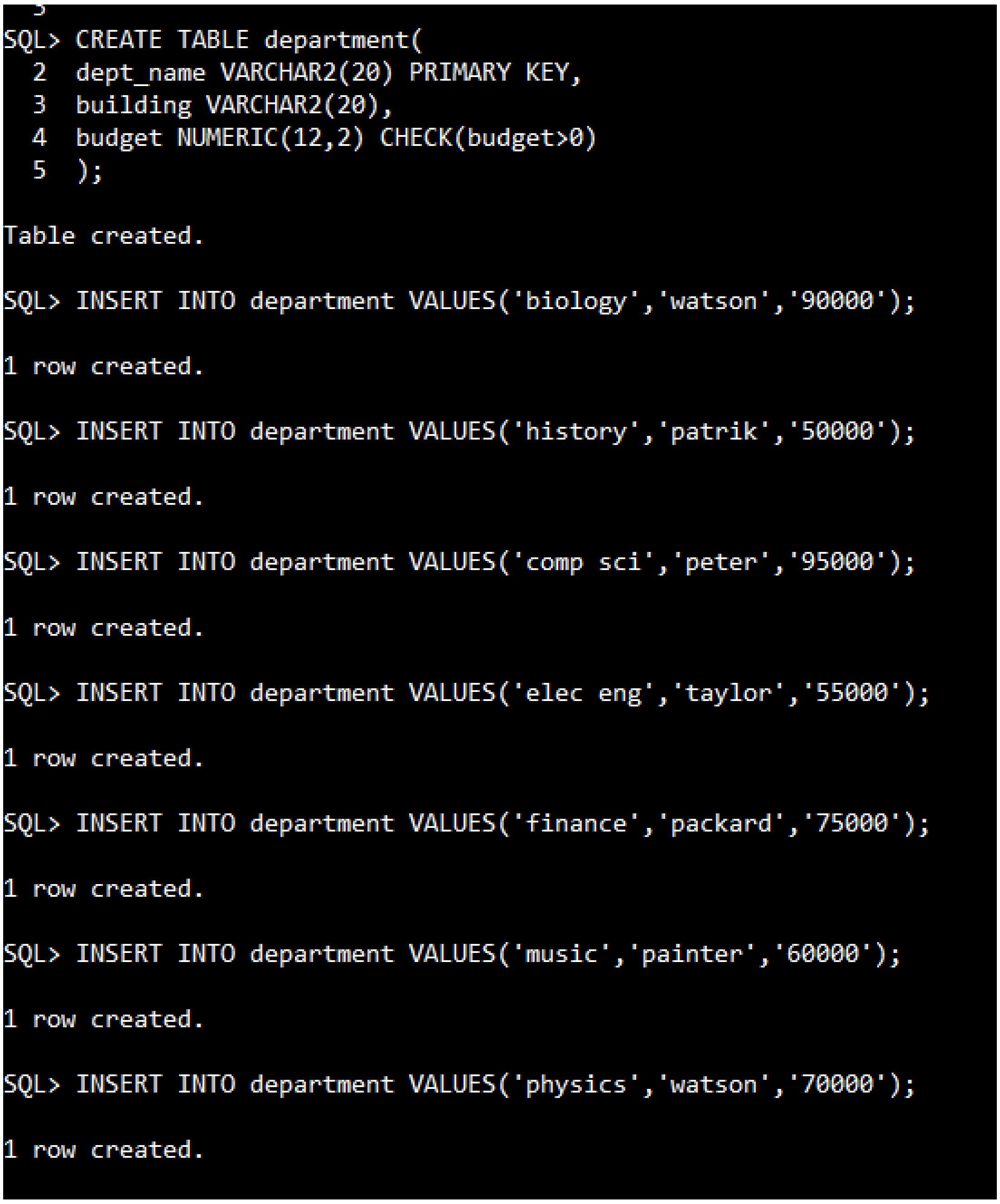
```
SQL> DECLARE
N3 NUMBER(2);
BEGIN
N3 := ADDER(11,22);
DBMS_OUTPUT.PUT_LINE('ADDITION IS: ' || N3);
END;
/

PL/SQL procedure successfully completed.
```

Drop function:

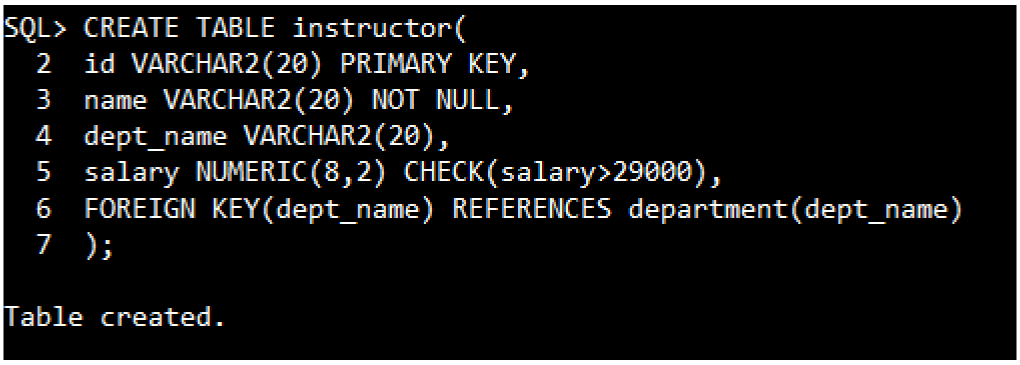


```
SQL> DROP FUNCTION Adder;

Function dropped.
```

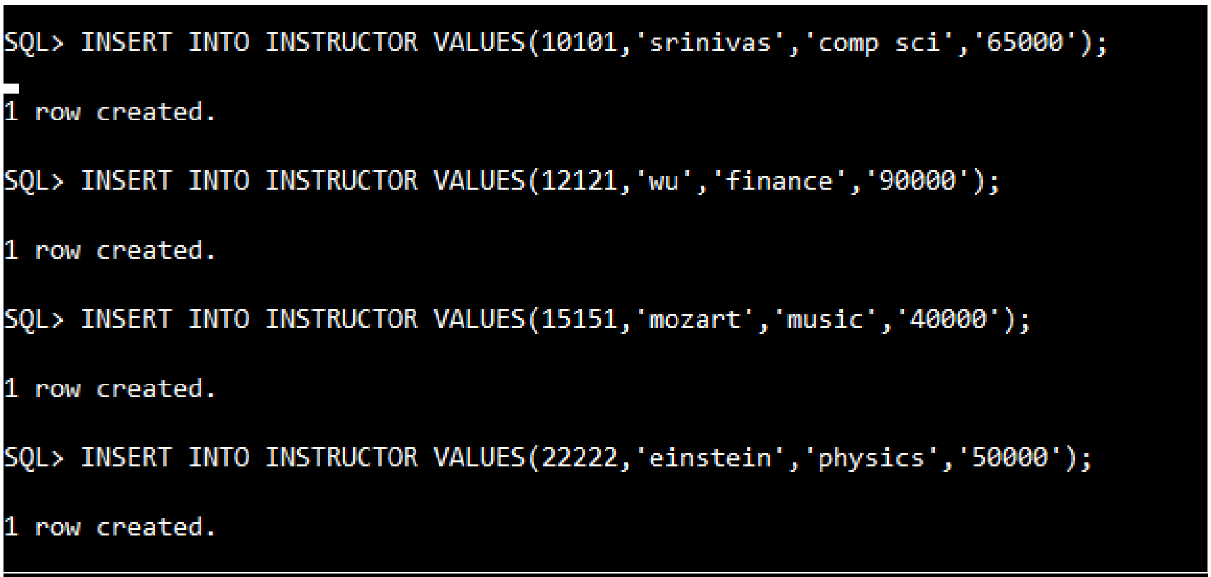
EXP15-TRIGGERS



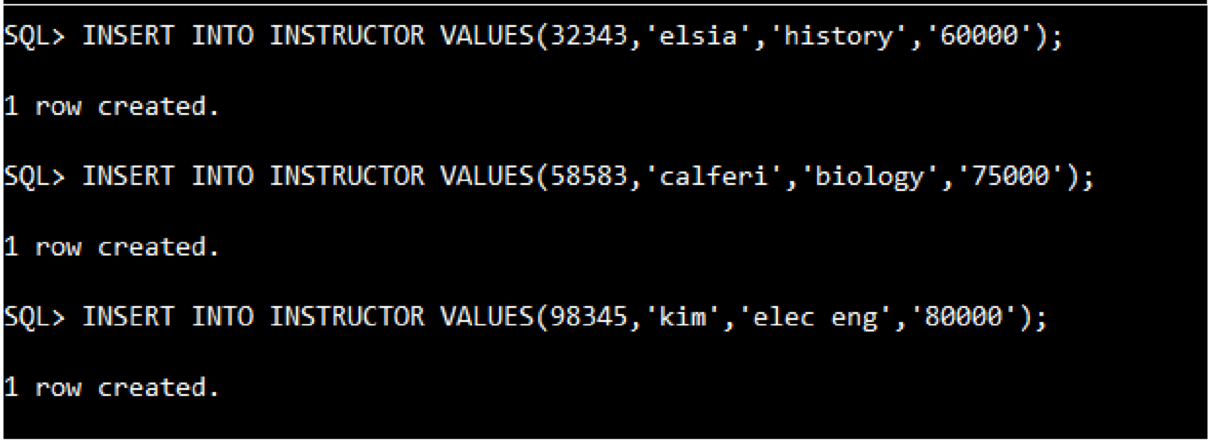
```
SQL> CREATE TABLE department(
dept_name VARCHAR2(20) PRIMARY KEY,
building VARCHAR2(20),
budget NUMERIC(12,2) CHECK(budget>0)
);

Table created.

SQL> INSERT INTO department VALUES('biology','watson','90000');

1 row created.

SQL> INSERT INTO department VALUES('history','patrik','50000');

1 row created.

SQL> INSERT INTO department VALUES('comp sci','peter','95000');

1 row created.

SQL> INSERT INTO department VALUES('elec eng','taylor','55000');

1 row created.

SQL> INSERT INTO department VALUES('finance','packard','75000');

1 row created.

SQL> INSERT INTO department VALUES('music','painter','60000');

1 row created.

SQL> INSERT INTO department VALUES('physics','watson','70000');

1 row created.
```



```
SQL> CREATE TABLE instructor(
id VARCHAR2(20) PRIMARY KEY,
name VARCHAR2(20) NOT NULL,
dept_name VARCHAR2(20),
salary NUMERIC(8,2) CHECK(salary>29000),
FOREIGN KEY(dept_name) REFERENCES department(dept_name)
);

Table created.
```

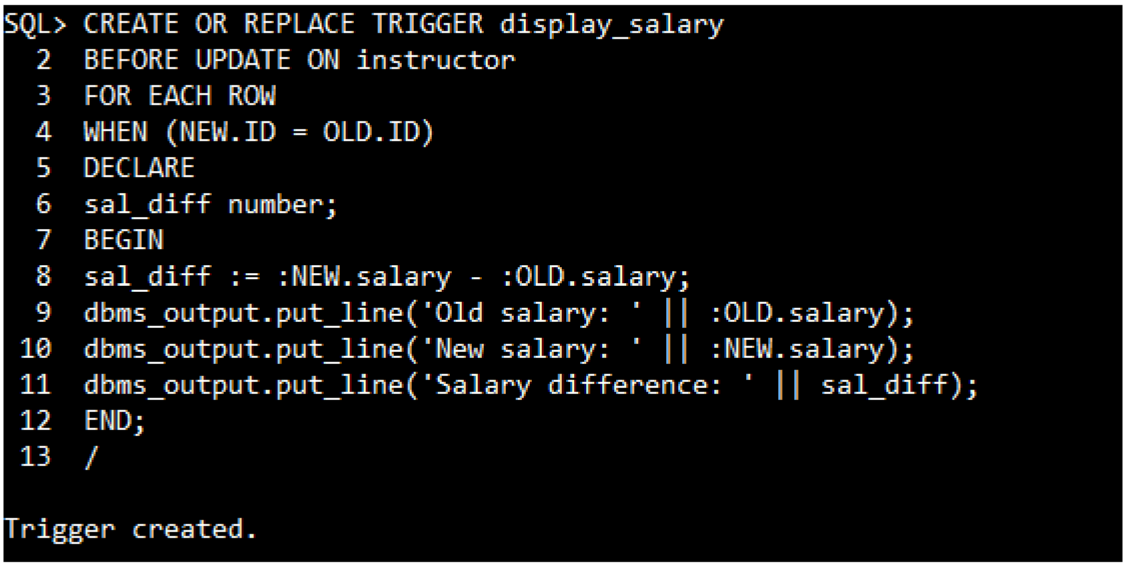


```
SQL> INSERT INTO INSTRUCTOR VALUES(10101,'srinivas','comp sci','65000');

1 row created.

SQL> INSERT INTO INSTRUCTOR VALUES(12121,'wu','finance','90000');

1 row created.

SQL> INSERT INTO INSTRUCTOR VALUES(15151,'mozart','music','40000');

1 row created.

SQL> INSERT INTO INSTRUCTOR VALUES(22222,'einstein','physics','50000');

1 row created.
```

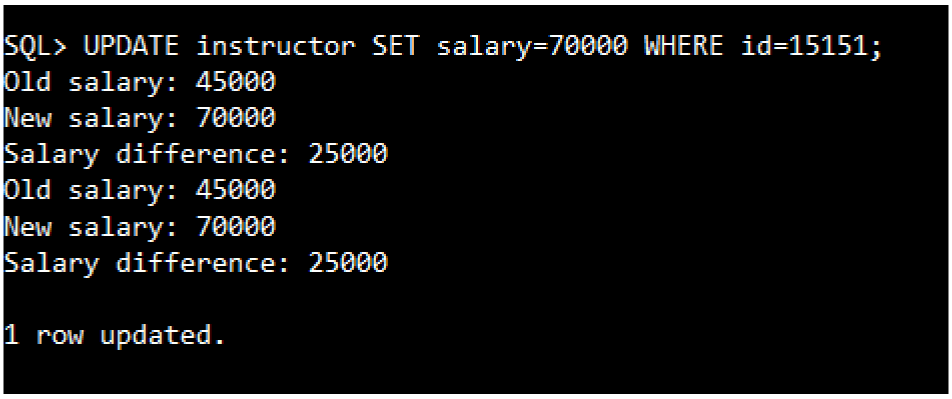


```
SQL> INSERT INTO INSTRUCTOR VALUES(32343,'elsia','history','60000');

1 row created.

SQL> INSERT INTO INSTRUCTOR VALUES(58583,'calferi','biology','75000');

1 row created.

SQL> INSERT INTO INSTRUCTOR VALUES(98345,'kim','elec eng','80000');

1 row created.
```

Creating trigger:

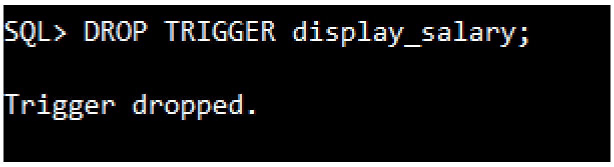


```
SQL> CREATE OR REPLACE TRIGGER display_salary
     BEFORE UPDATE ON instructor
     FOR EACH ROW
     WHEN (NEW.ID = OLD.ID)
     DECLARE
     sal_diff number;
     BEGIN
     sal_diff := :NEW.salary - :OLD.salary;
     dbms_output.put_line('Old salary: ' || :OLD.salary);
     dbms_output.put_line('New salary: ' || :NEW.salary);
     dbms_output.put_line('Salary difference: ' || sal_diff);
     END;
     /

Trigger created.
```



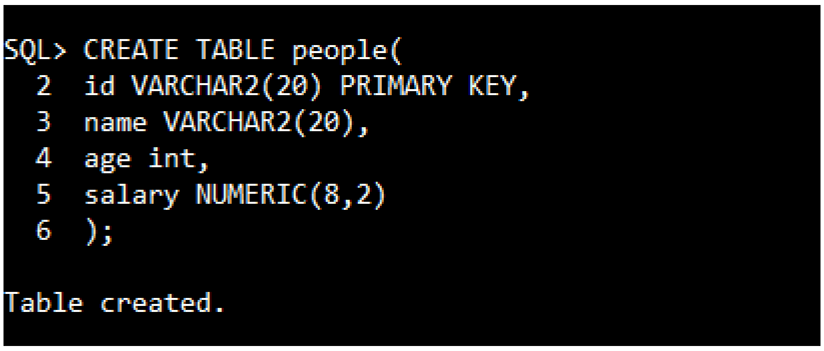
```
SQL> UPDATE instructor SET salary=70000 WHERE id=15151;
Old salary: 45000
New salary: 70000
Salary difference: 25000
Old salary: 45000
New salary: 70000
Salary difference: 25000

1 row updated.
```

Drop trigger:

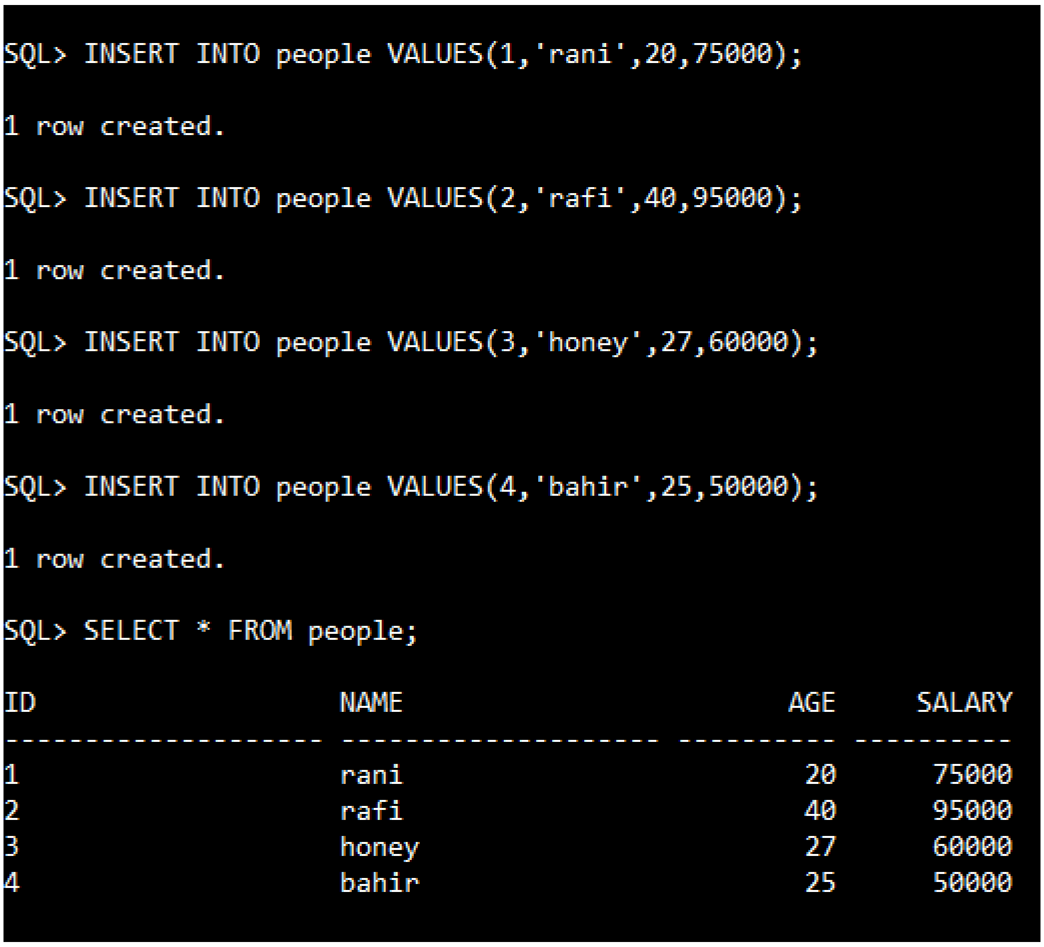


```
SQL> DROP TRIGGER display_salary;

Trigger dropped.
```

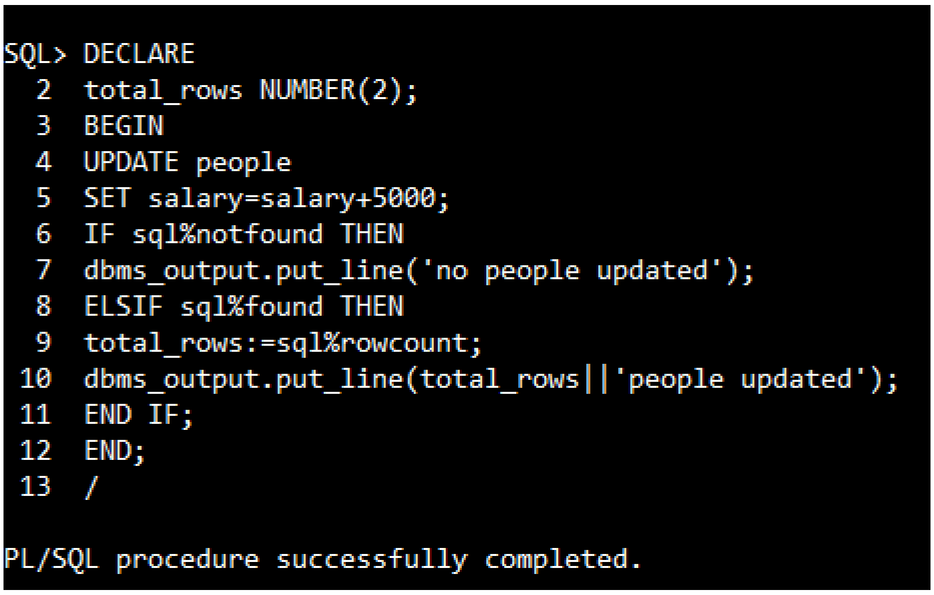
EXP16-CURSORS



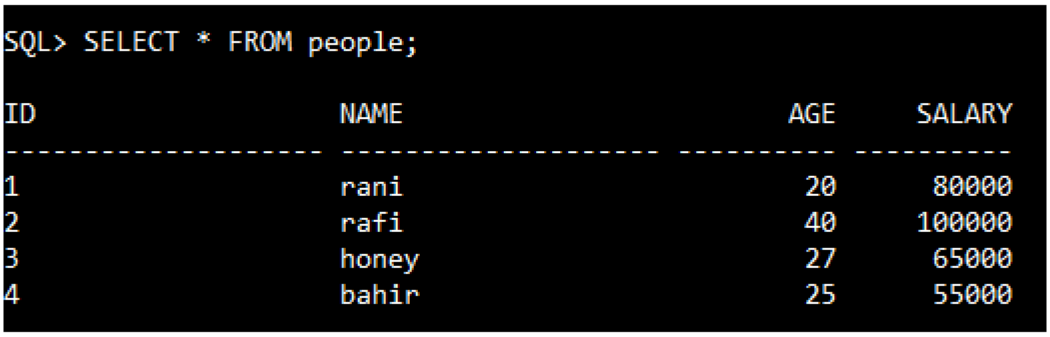
```
SQL> CREATE TABLE people(
  2  id VARCHAR2(20) PRIMARY KEY,
  3  name VARCHAR2(20),
  4  age int,
  5  salary NUMERIC(8,2)
  6  );

Table created.
```



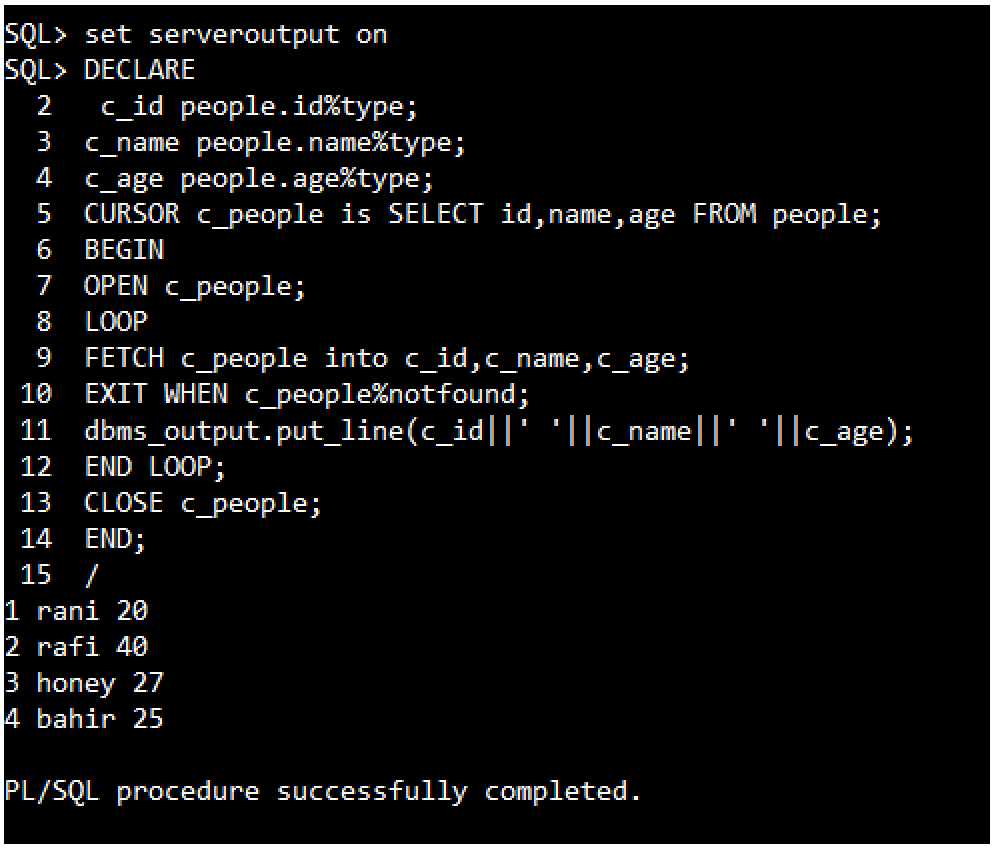
```
SQL> INSERT INTO people VALUES(1,'rani',20,75000);

1 row created.

SQL> INSERT INTO people VALUES(2,'rafi',40,95000);

1 row created.

SQL> INSERT INTO people VALUES(3,'honey',27,60000);

1 row created.

SQL> INSERT INTO people VALUES(4,'bahir',25,50000);

1 row created.

SQL> SELECT * FROM people;

ID                   NAME                        AGE     SALARY
-------------------- -------------------- ---------- ----------
1                    rani                         20      75000
2                    rafi                         40      95000
3                    honey                        27      60000
4                    bahir                        25      50000
```



```
SQL> DECLARE
total_rows NUMBER(2);
BEGIN
UPDATE people
SET salary=salary+5000;
IF sql%notfound THEN
dbms_output.put_line('no people updated');
ELSIF sql%found THEN
total_rows:=sql%rowcount;
dbms_output.put_line(total_rows||'people updated');
END IF;
END;
/

PL/SQL procedure successfully completed.
```



```
SQL> SELECT * FROM people;

ID                   NAME                        AGE     SALARY
-------------------- -------------------- ---------- ----------
1                    rani                         20      80000
2                    rafi                         40     100000
3                    honey                        27      65000
4                    bahir                        25      55000
```



```
SQL> set serveroutput on
SQL> DECLARE
 c_id people.id%type;
c_name people.name%type;
c_age people.age%type;
CURSOR c_people is SELECT id,name,age FROM people;
BEGIN
OPEN c_people;
LOOP
FETCH c_people into c_id,c_name,c_age;
EXIT WHEN c_people%notfound;
dbms_output.put_line(c_id||' '||c_name||' '||c_age);
END LOOP;
CLOSE c_people;
END;
/
1 rani 20
2 rafi 40
3 honey 27
4 bahir 25

PL/SQL procedure successfully completed.
```