**ASSIGNMENT 12**

Hotel (Hotel\_No, Name, Address)

Room (Room\_No, Hotel\_No, Type, Price)

Booking (Hotel\_No, Guest\_No, Date\_From, Date\_To, Room\_No)

Guest (Guest\_No, Name, Address)

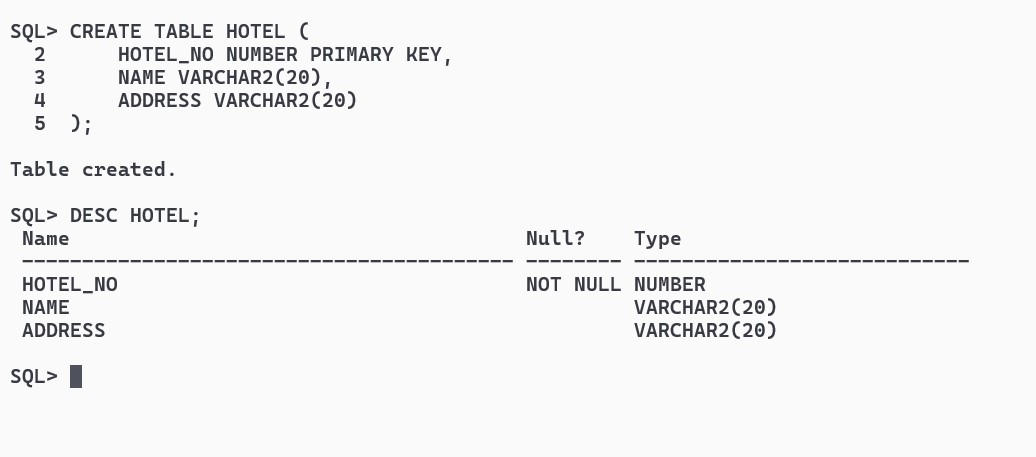
CREATE TABLE HOTEL (

HOTEL\_NO NUMBER PRIMARY KEY,

NAME VARCHAR2(20),

ADDRESS VARCHAR2(20)

);



CREATE TABLE ROOM (

ROOM\_NO NUMBER PRIMARY KEY,

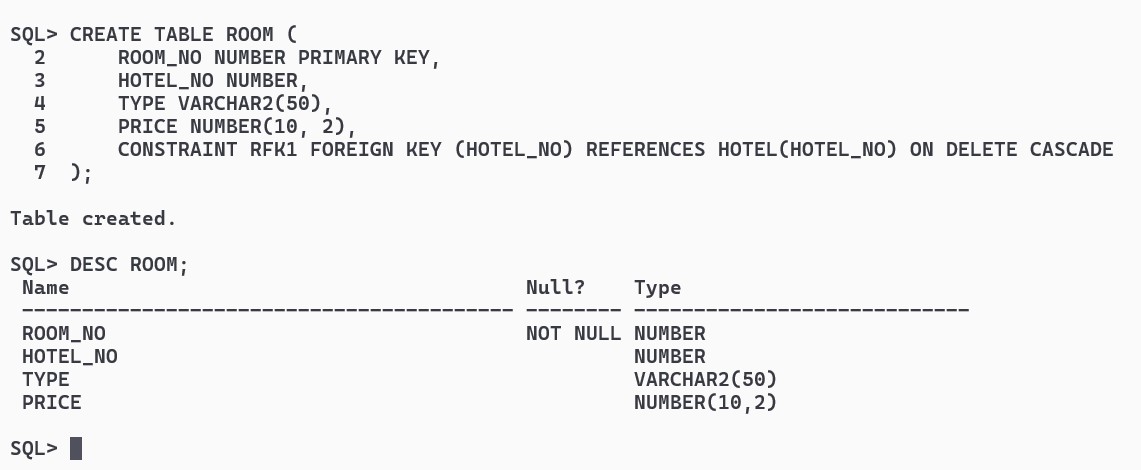
HOTEL\_NO NUMBER,

TYPE VARCHAR2(50),

PRICE NUMBER(10, 2),

CONSTRAINT RFK1 FOREIGN KEY (HOTEL\_NO) REFERENCES HOTEL(HOTEL\_NO) ON DELETE CASCADE

);



CREATE TABLE BOOKING (

HOTEL\_NO NUMBER,

GUEST\_NO NUMBER,

DATE\_FROM DATE,

DATE\_TO DATE,

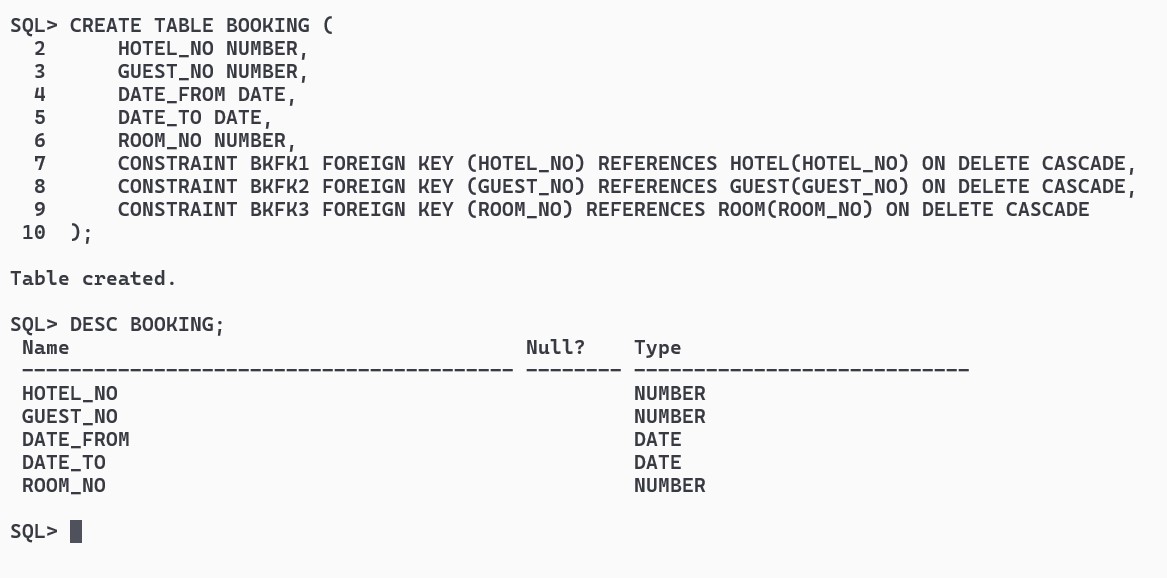
ROOM\_NO NUMBER,

CONSTRAINT BKFK1 FOREIGN KEY (HOTEL\_NO) REFERENCES HOTEL(HOTEL\_NO) ON DELETE CASCADE,

CONSTRAINT BKFK2 FOREIGN KEY (GUEST\_NO) REFERENCES GUEST(GUEST\_NO) ON DELETE CASCADE,

CONSTRAINT BKFK3 FOREIGN KEY (ROOM\_NO) REFERENCES ROOM(ROOM\_NO) ON DELETE CASCADE

);



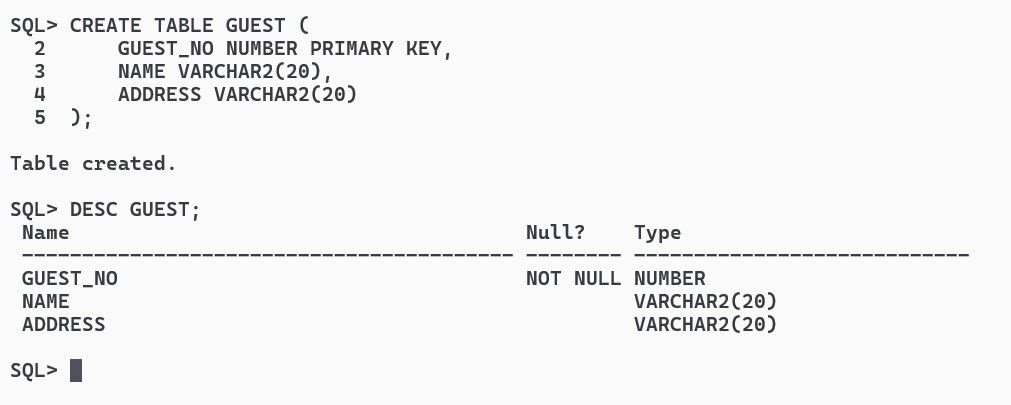
CREATE TABLE GUEST (

GUEST\_NO NUMBER PRIMARY KEY,

NAME VARCHAR2(20),

ADDRESS VARCHAR2(20)

);



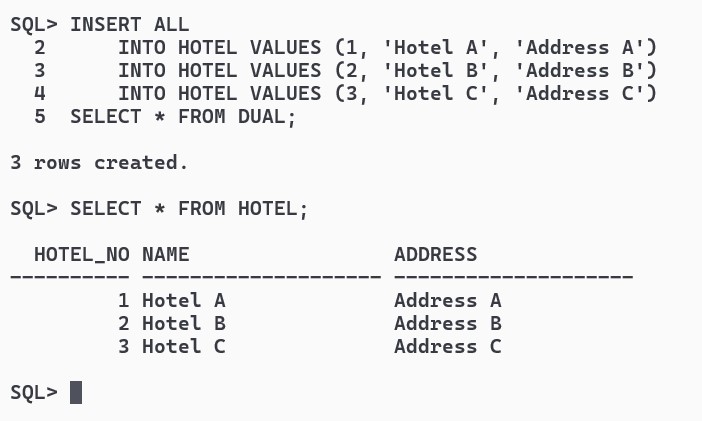
INSERT ALL

INTO HOTEL VALUES (1, 'Hotel A', 'Address A')

INTO HOTEL VALUES (2, 'Hotel B', 'Address B')

INTO HOTEL VALUES (3, 'Hotel C', 'Address C')

SELECT \* FROM DUAL;



INSERT ALL

INTO ROOM VALUES (101, 1, 'Single', 100.00)

INTO ROOM VALUES (102, 1, 'Double', 150.00)

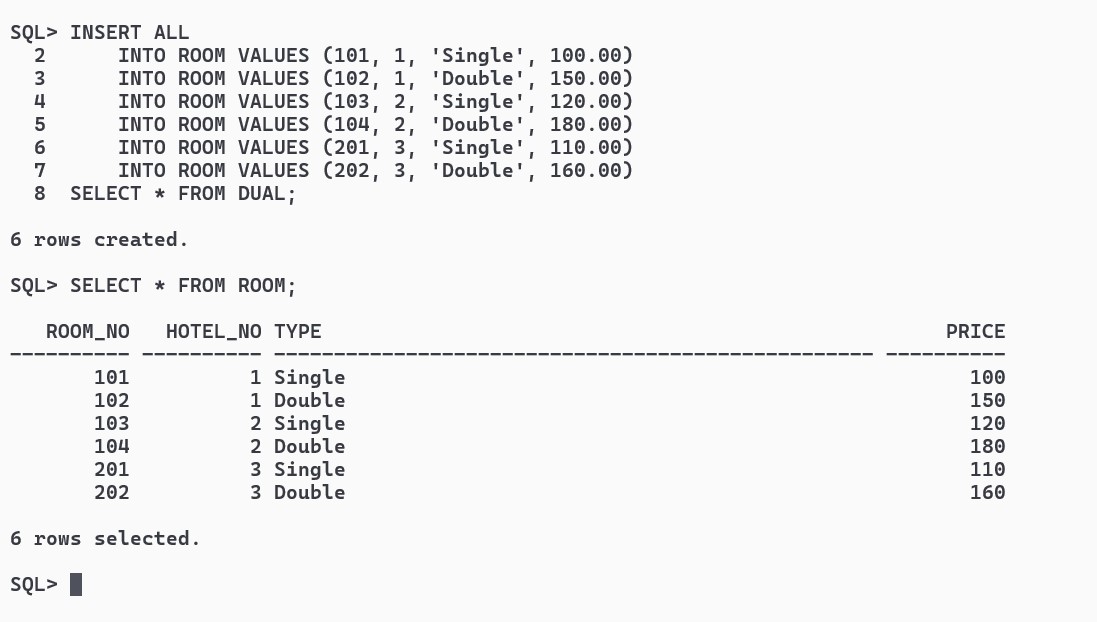
INTO ROOM VALUES (103, 2, 'Single', 120.00)

INTO ROOM VALUES (104, 2, 'Double', 180.00)

INTO ROOM VALUES (201, 3, 'Single', 110.00)

INTO ROOM VALUES (202, 3, 'Double', 160.00)

SELECT \* FROM DUAL;



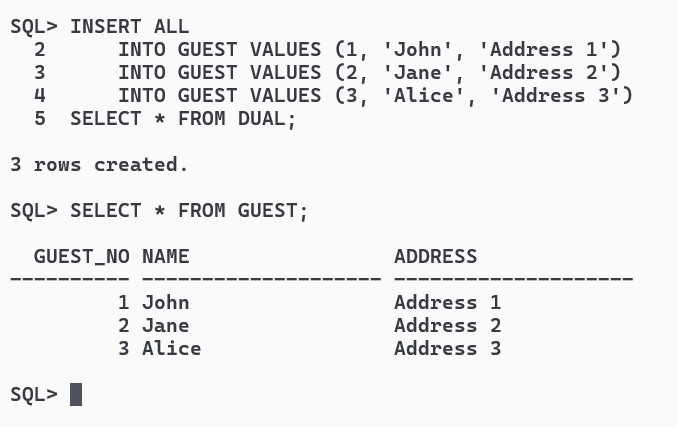
INSERT ALL

INTO GUEST VALUES (1, 'John', 'Address 1')

INTO GUEST VALUES (2, 'Jane', 'Address 2')

INTO GUEST VALUES (3, 'Alice', 'Address 3')

SELECT \* FROM DUAL;



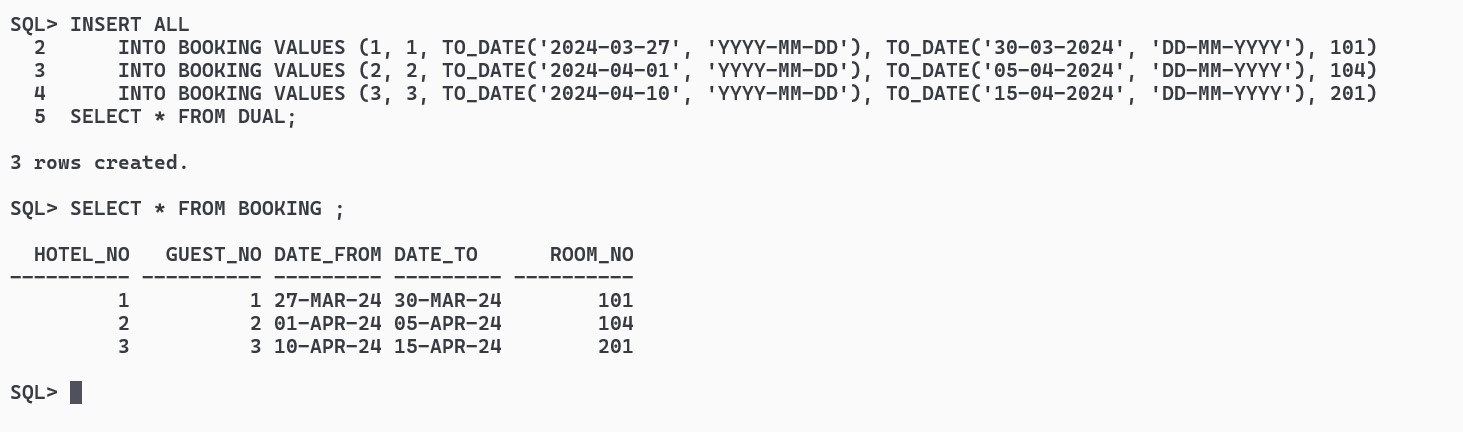
INSERT ALL

INTO BOOKING VALUES (1, 1, TO\_DATE('2024-03-27', 'YYYY-MM-DD'), TO\_DATE('30-03-2024', 'DD-MM-YYYY'), 101)

INTO BOOKING VALUES (2, 2, TO\_DATE('2024-04-01', 'YYYY-MM-DD'), TO\_DATE('05-04-2024', 'DD-MM-YYYY'), 104)

INTO BOOKING VALUES (3, 3, TO\_DATE('2024-04-10', 'YYYY-MM-DD'), TO\_DATE('15-04-2024', 'DD-MM-YYYY'), 201)

SELECT \* FROM DUAL;



Populate the tables Answer the following query using SQL.

1.List the names and addresses of all guests in London, alphabetically ordered by name

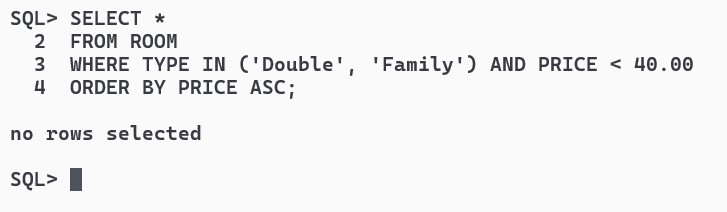
SELECT NAME, ADDRESS FROM GUEST WHERE ADDRESS LIKE '%LONDON%' ORDER BY NAME;



2.List all double or family rooms with a price below £40.00 per night, in ascending order of price.

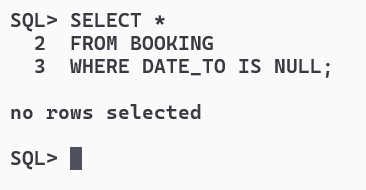
SELECT \* FROM ROOM WHERE TYPE IN ('Double', 'Family') AND PRICE < 40.00

ORDER BY PRICE ASC;



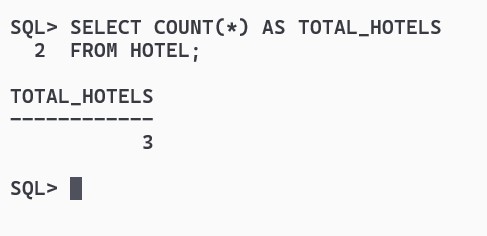
3.List the bookings for which no date\_to has been specified.

SELECT \* FROM BOOKING WHERE DATE\_TO IS NULL;



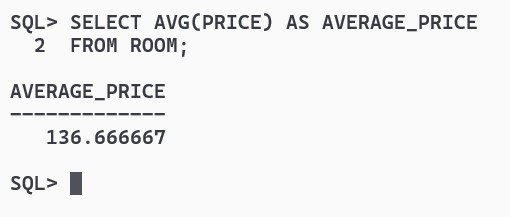
4.How many hotels are there?

SELECT COUNT(\*) AS TOTAL\_HOTELS FROM HOTEL;



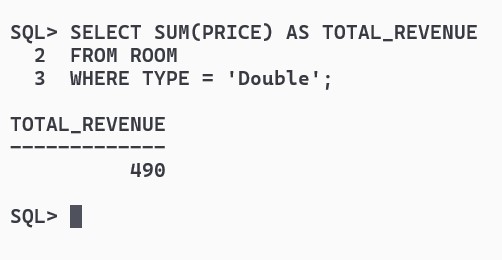
5.What is the average price of a room?

SELECT AVG(PRICE) AS AVERAGE\_PRICE FROM ROOM;



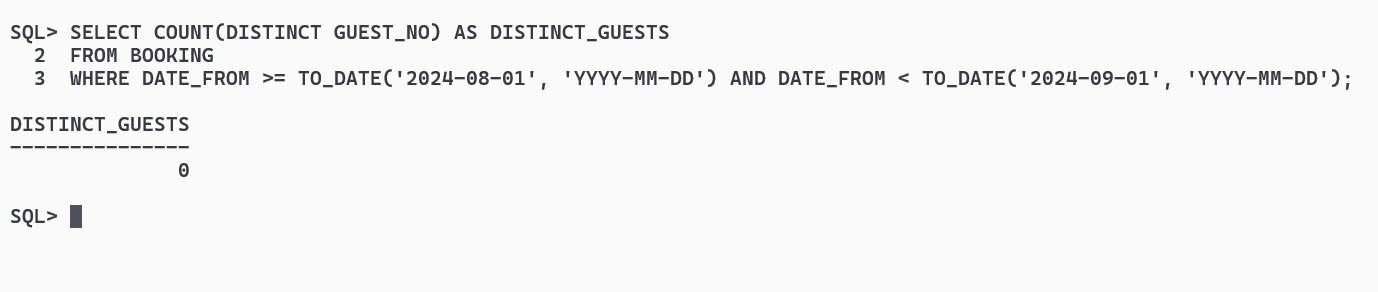
6.What is the total revenue per night from all double rooms?

SELECT SUM(PRICE) AS TOTAL\_REVENUE FROM ROOM WHERE TYPE = 'Double';



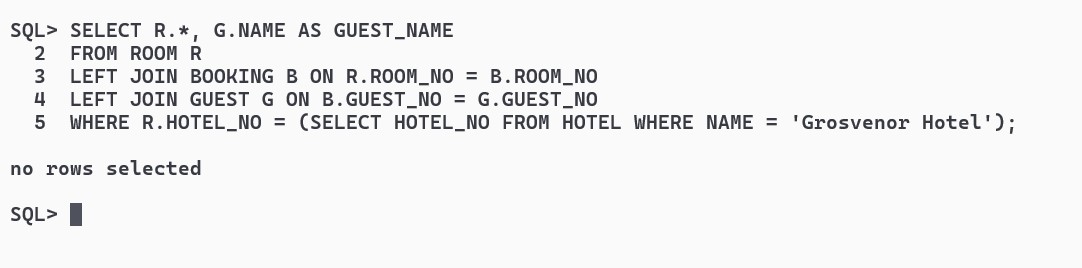
7.How many different guests have made bookings for August?

SELECT COUNT(DISTINCT GUEST\_NO) AS DISTINCT\_GUESTS FROM BOOKING WHERE DATE\_FROM >= TO\_DATE('2024-08-01', 'YYYY-MM-DD') AND DATE\_FROM < TO\_DATE('2024-09-01', 'YYYY-MM-DD');



8.List the details of all rooms at the Grosvenor Hotel, including the name of the guest staying in the room, if the room is occupied.

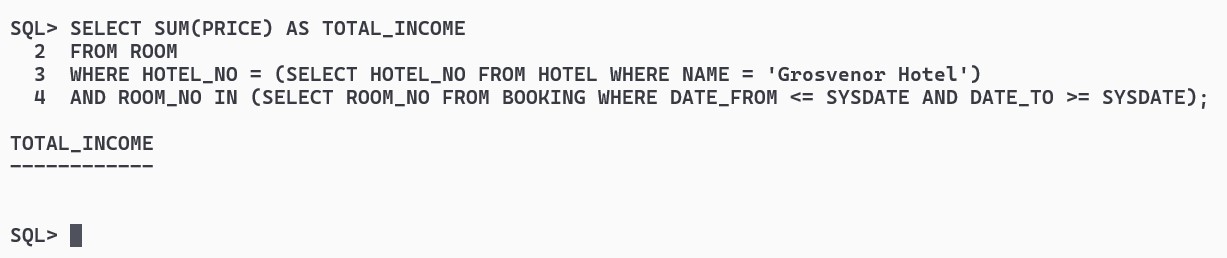
SELECT R.\*, G.NAME AS GUEST\_NAME FROM ROOM R LEFT JOIN BOOKING B ON R.ROOM\_NO = B.ROOM\_NO LEFT JOIN GUEST G ON B.GUEST\_NO = G.GUEST\_NO WHERE R.HOTEL\_NO = (SELECT HOTEL\_NO FROM HOTEL WHERE NAME = 'Grosvenor Hotel');



9.What is the total income from bookings for the Grosvenor Hotel today?

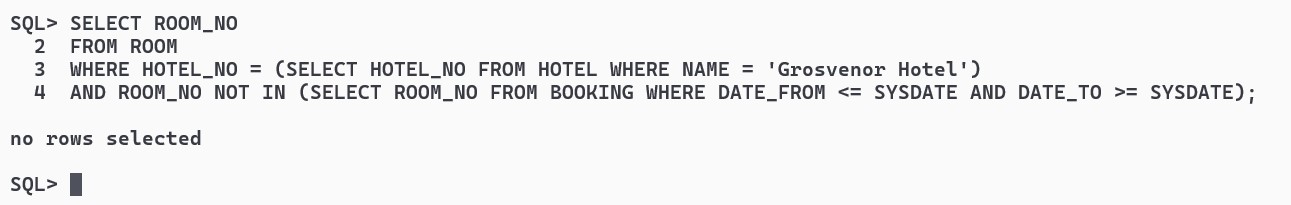
SELECT SUM(PRICE) AS TOTAL\_INCOME FROM ROOM

WHERE HOTEL\_NO = (SELECT HOTEL\_NO FROM HOTEL WHERE NAME = 'Grosvenor Hotel') AND ROOM\_NO IN (SELECT ROOM\_NO FROM BOOKING WHERE DATE\_FROM <= SYSDATE AND DATE\_TO >= SYSDATE);

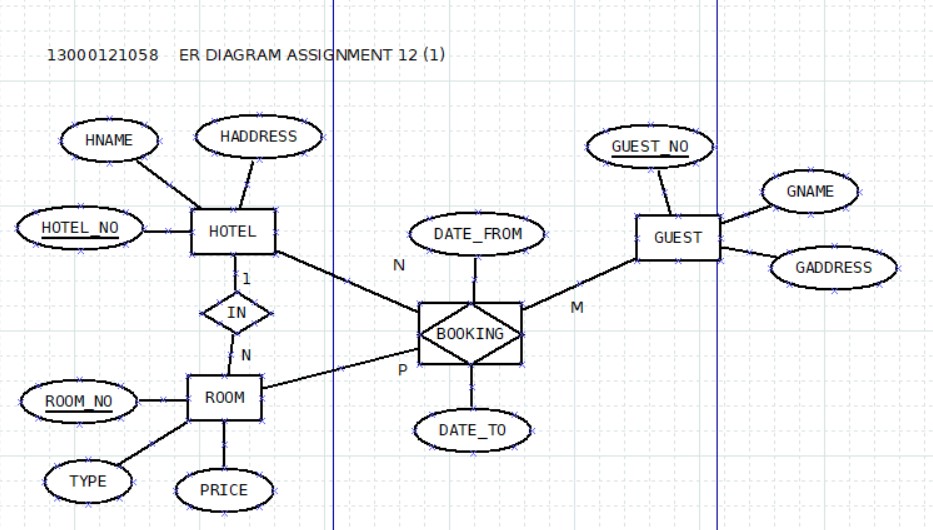


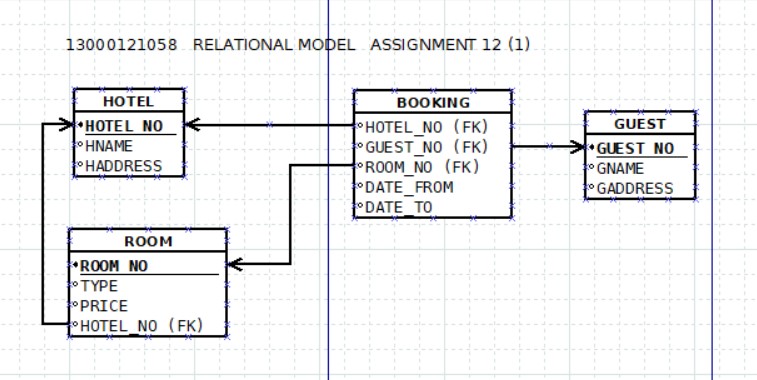
10.List the rooms that are currently unoccupied at the Grosvenor Hotel.

SELECT ROOM\_NO FROM ROOM WHERE HOTEL\_NO = (SELECT HOTEL\_NO FROM HOTEL WHERE NAME = 'Grosvenor Hotel') AND ROOM\_NO NOT IN (SELECT ROOM\_NO FROM BOOKING WHERE DATE\_FROM <= SYSDATE AND DATE\_TO >= SYSDATE);



Design an ER Model for an application where hotels are booked by guests wanting to go on aholiday in India or abroad. Your design should meet all requirements. Map into a relationalmodel.





EMPLOYEE (SSN, Name, Address, Sex, Salary, SuperSSN, DNo)

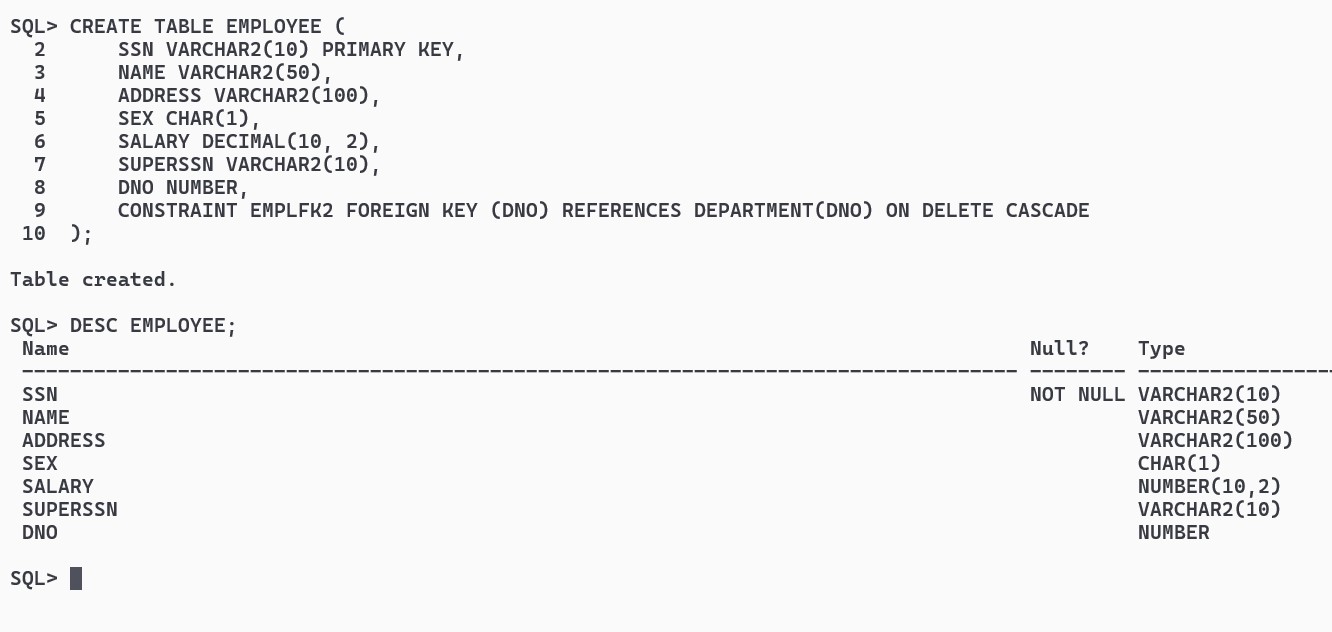
DEPARTMENT (DNo, DName, MgrSSN, MgrStartDate)

DLOCATION (DNo,DLoc)

PROJECT (PNo, PName, PLocation, DNo)

WORKS\_ON (SSN, PNo, Hours)

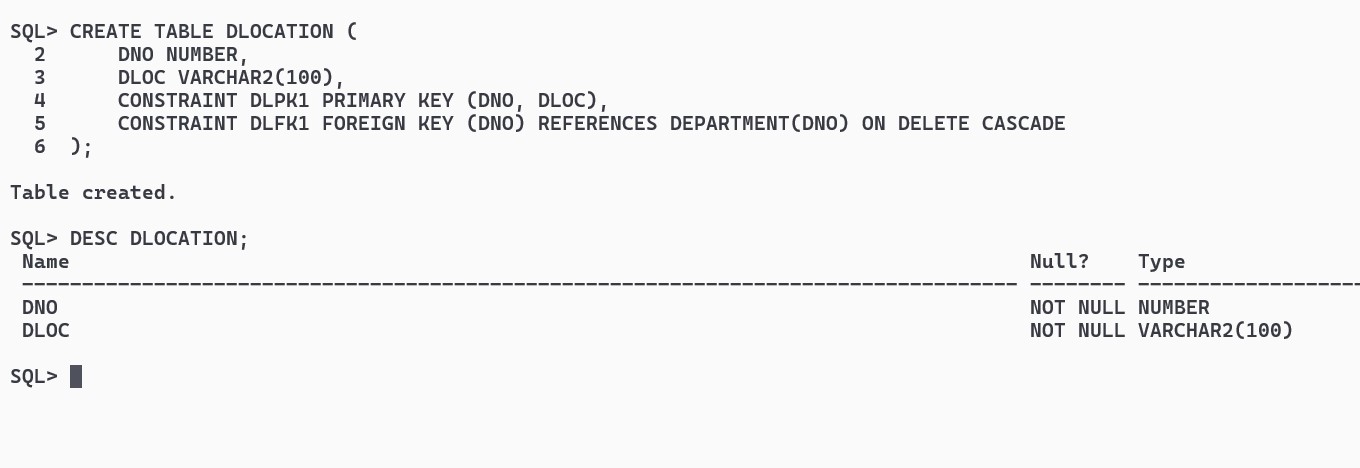
CREATE TABLE EMPLOYEE (SSN VARCHAR2(10) PRIMARY KEY, NAME VARCHAR2(50), ADDRESS VARCHAR2(100), SEX CHAR(1), SALARY DECIMAL(10, 2), SUPERSSN VARCHAR2(10), DNO NUMBER, CONSTRAINT EMPLFK2 FOREIGN KEY (DNO) REFERENCES DEPARTMENT(DNO) ON DELETE CASCADE);



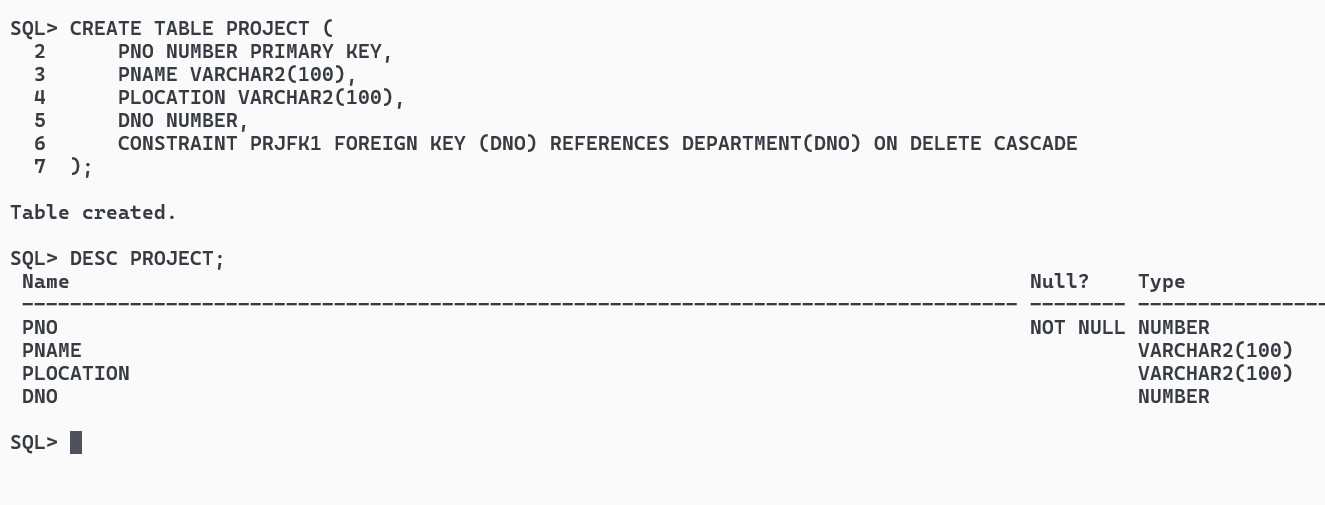
CREATE TABLE DEPARTMENT (DNO NUMBER PRIMARY KEY, DNAME VARCHAR2(50), MGRSSN VARCHAR2(10), MGRSTARTDATE DATE);



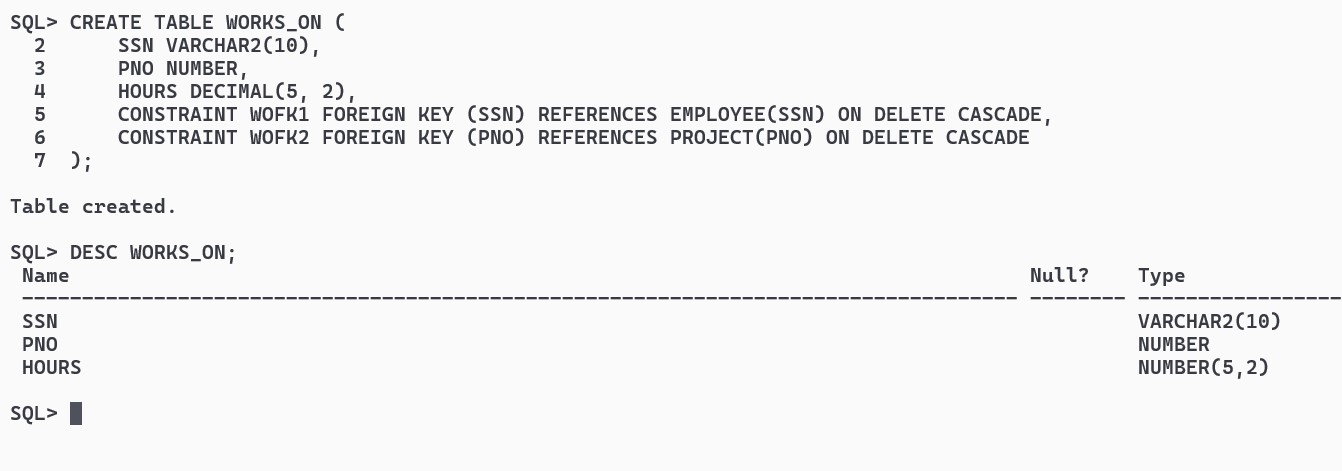
CREATE TABLE DLOCATION (DNO NUMBER, DLOC VARCHAR2(100), CONSTRAINT DLPK1 PRIMARY KEY (DNO, DLOC), CONSTRAINT DLFK1 FOREIGN KEY (DNO) REFERENCES DEPARTMENT(DNO) ON DELETE CASCADE);



CREATE TABLE PROJECT ( PNO NUMBER PRIMARY KEY, PNAME VARCHAR2(100), PLOCATION VARCHAR2(100), DNO NUMBER, CONSTRAINT PRJFK1 FOREIGN KEY (DNO) REFERENCES DEPARTMENT(DNO) ON DELETE CASCADE);



CREATE TABLE WORKS\_ON (SSN VARCHAR2(10), PNO NUMBER, HOURS DECIMAL(5, 2), CONSTRAINT WOFK1 FOREIGN KEY (SSN) REFERENCES EMPLOYEE(SSN) ON DELETE CASCADE, CONSTRAINT WOFK2 FOREIGN KEY (PNO) REFERENCES PROJECT(PNO) ON DELETE CASCADE);



INSERT ALL

INTO EMPLOYEE VALUES ('1111111111', 'John Doe', '123 Main St', 'M', 50000, NULL, 1)

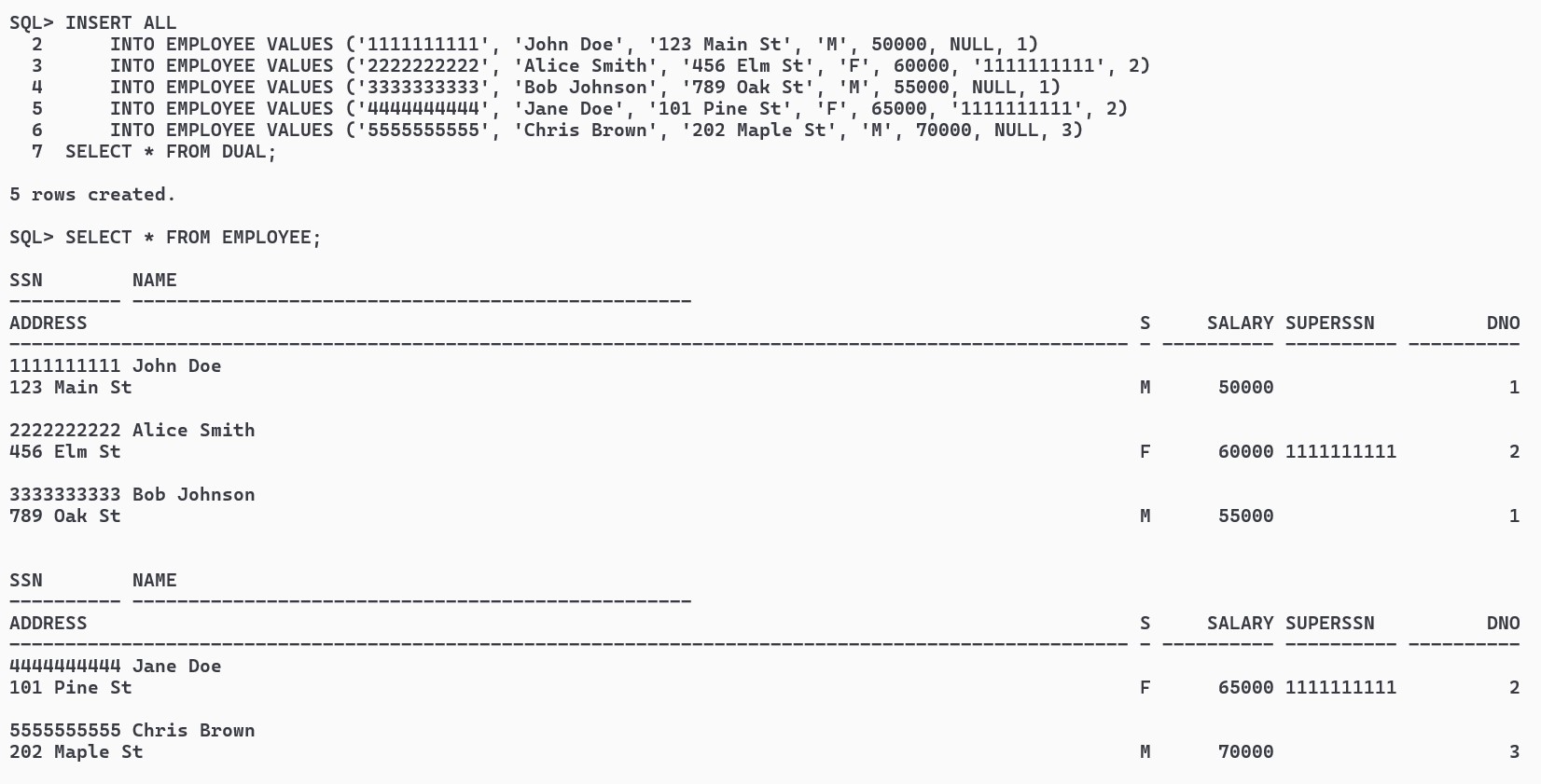
INTO EMPLOYEE VALUES ('2222222222', 'Alice Smith', '456 Elm St', 'F', 60000, '1111111111', 2)

INTO EMPLOYEE VALUES ('3333333333', 'Bob Johnson', '789 Oak St', 'M', 55000, NULL, 1)

INTO EMPLOYEE VALUES ('4444444444', 'Jane Doe', '101 Pine St', 'F', 65000, '1111111111', 2)

INTO EMPLOYEE VALUES ('5555555555', 'Chris Brown', '202 Maple St', 'M', 70000, NULL, 3)

SELECT \* FROM DUAL;



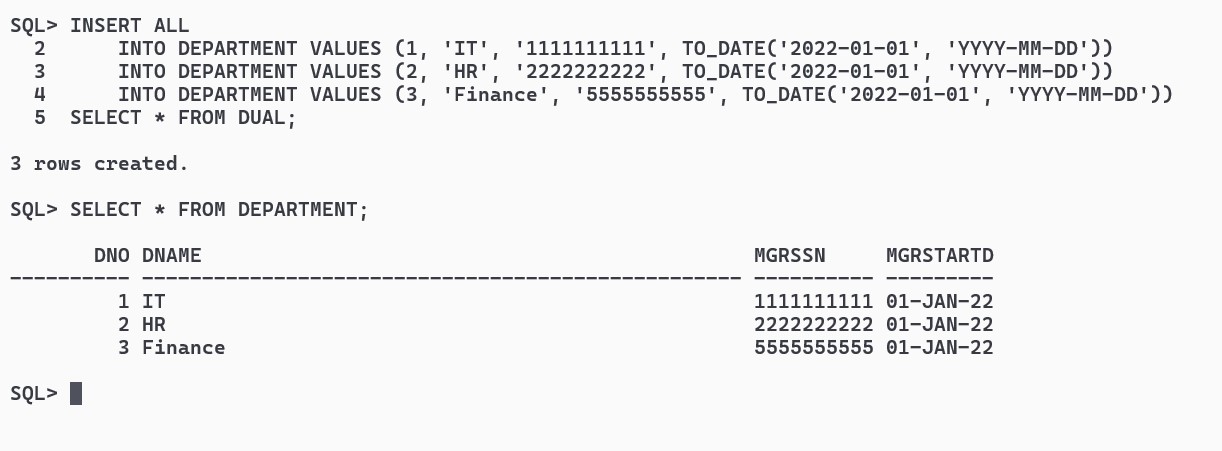
INSERT ALL

INTO DEPARTMENT VALUES (1, 'IT', '1111111111', TO\_DATE('2022-01-01', 'YYYY-MM-DD'))

INTO DEPARTMENT VALUES (2, 'HR', '2222222222', TO\_DATE('2022-01-01', 'YYYY-MM-DD'))

INTO DEPARTMENT VALUES (3, 'Finance', '5555555555', TO\_DATE('2022-01-01', 'YYYY-MM-DD'))

SELECT \* FROM DUAL;



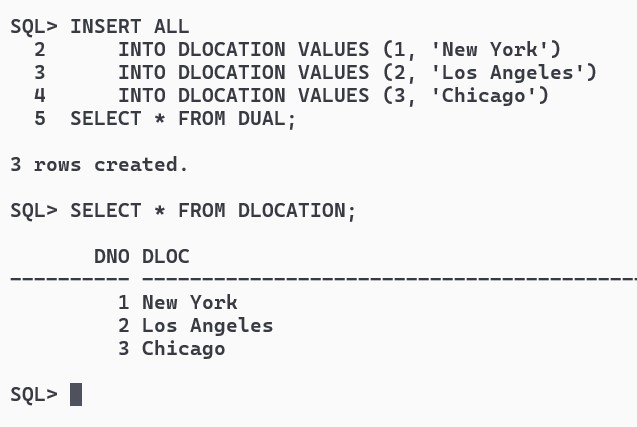
INSERT ALL

INTO DLOCATION VALUES (1, 'New York')

INTO DLOCATION VALUES (2, 'Los Angeles')

INTO DLOCATION VALUES (3, 'Chicago')

SELECT \* FROM DUAL;



INSERT ALL

INTO PROJECT VALUES (101, 'Project X', 'New York', 1)

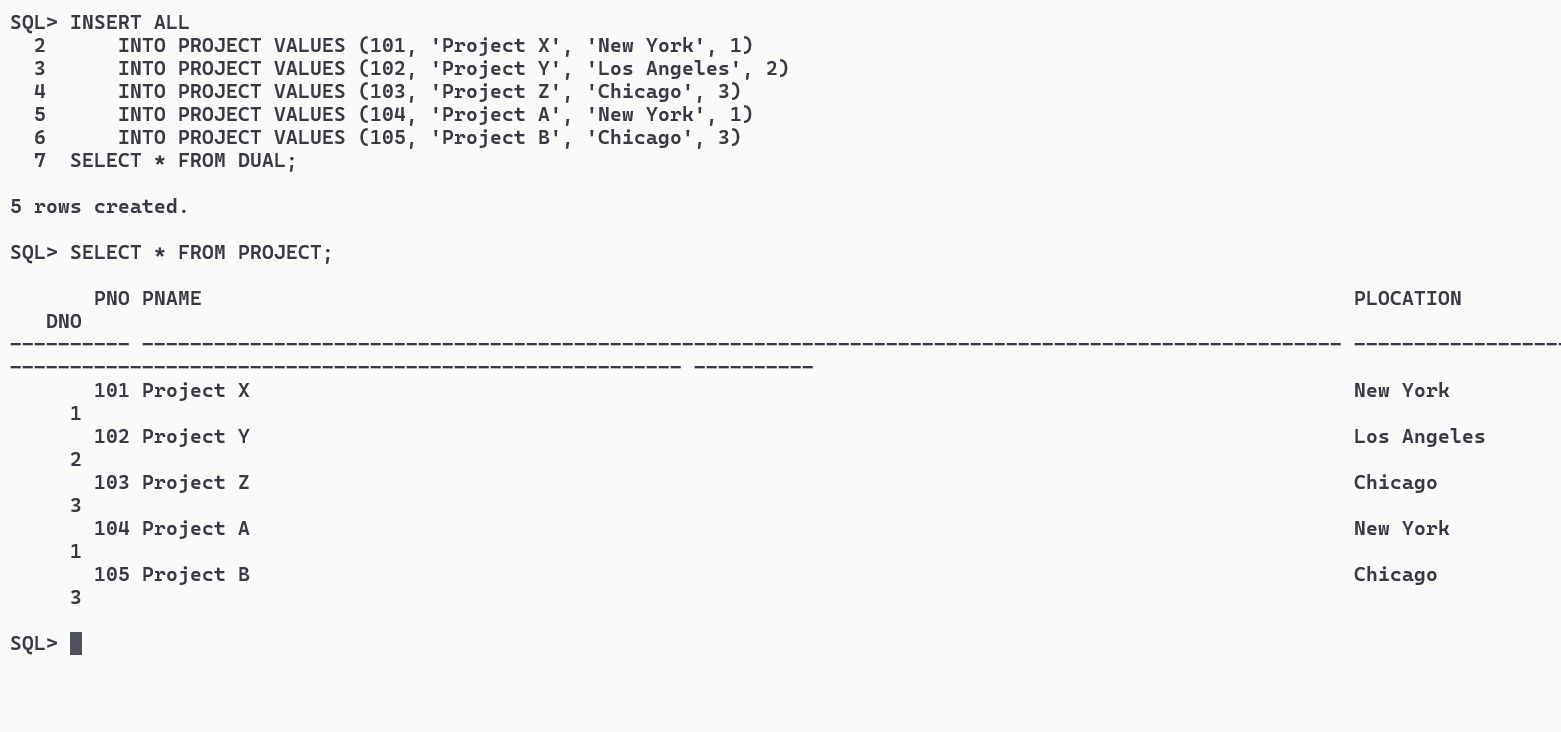
INTO PROJECT VALUES (102, 'Project Y', 'Los Angeles', 2)

INTO PROJECT VALUES (103, 'Project Z', 'Chicago', 3)

INTO PROJECT VALUES (104, 'Project A', 'New York', 1)

INTO PROJECT VALUES (105, 'Project B', 'Chicago', 3)

SELECT \* FROM DUAL;



INSERT ALL

INTO WORKS\_ON VALUES ('1111111111', 101, 40)

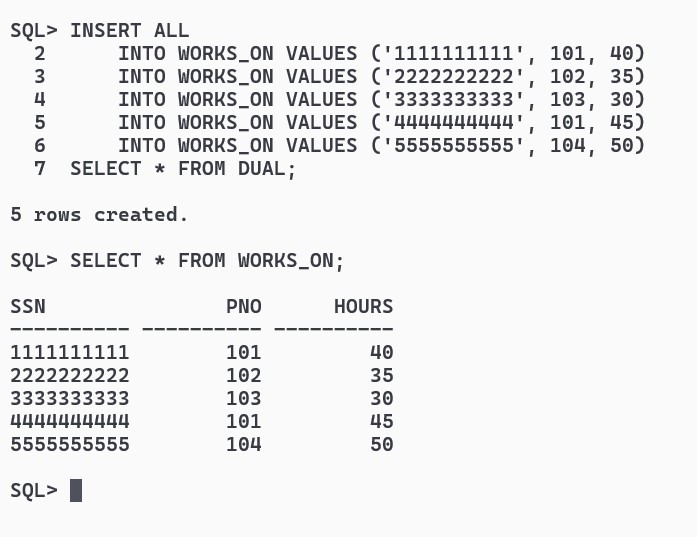
INTO WORKS\_ON VALUES ('2222222222', 102, 35)

INTO WORKS\_ON VALUES ('3333333333', 103, 30)

INTO WORKS\_ON VALUES ('4444444444', 101, 45)

INTO WORKS\_ON VALUES ('5555555555', 104, 50)

SELECT \* FROM DUAL;



Write SQL queries to ............

1.Make a list of all project numbers for projects that involve an employee whose last name is ‘Scott’, either as a worker or as a manager of the department that controls the project.

SELECT DISTINCT P.PNO

FROM PROJECT P

JOIN WORKS\_ON W ON P.PNO = W.PNO

JOIN EMPLOYEE E ON W.SSN = E.SSN

WHERE E.NAME LIKE '%Scott%'

UNION

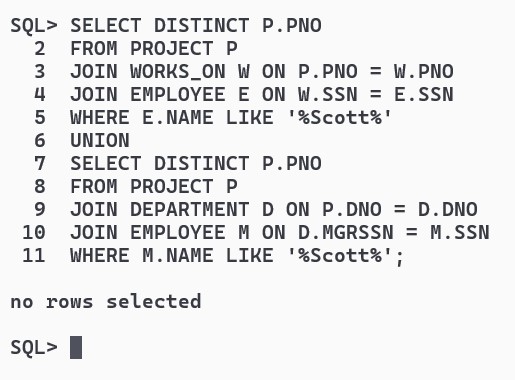
SELECT DISTINCT P.PNO

FROM PROJECT P

JOIN DEPARTMENT D ON P.DNO = D.DNO

JOIN EMPLOYEE M ON D.MGRSSN = M.SSN

WHERE M.NAME LIKE '%Scott%';



2.Show the resulting salaries if every employee working on the ‘IoT’ project is given a 10 percent raise.

UPDATE PROJECT SET PNAME = 'IoT' WHERE PNO = 104;

UPDATE EMPLOYEE

SET SALARY = SALARY \* 1.10

WHERE SSN IN (

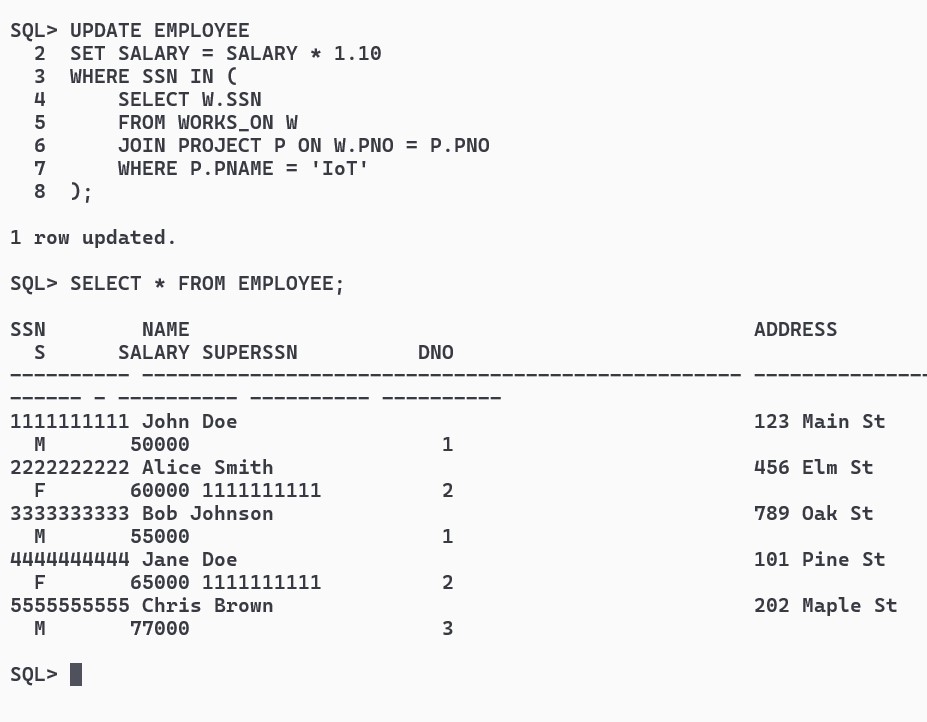
SELECT W.SSN

FROM WORKS\_ON W

JOIN PROJECT P ON W.PNO = P.PNO

WHERE P.PNAME = 'IoT'

);



3.Find the sum of the salaries of all employees of the ‘Accounts’ department, as well as the maximum salary, the minimum salary, and the average salary in this department

UPDATE DEPARTMENT SET DNAME = 'Accounts' WHERE DNO = 3;

SELECT SUM(SALARY) AS Total\_Salary,

MAX(SALARY) AS Max\_Salary,

MIN(SALARY) AS Min\_Salary,

AVG(SALARY) AS Avg\_Salary

FROM EMPLOYEE

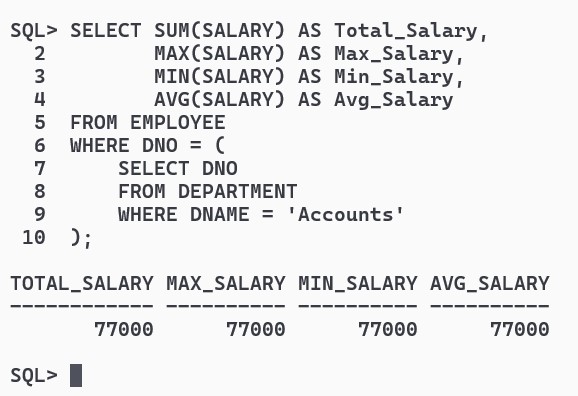
WHERE DNO = (

SELECT DNO

FROM DEPARTMENT

WHERE DNAME = 'Accounts'

);



4.Retrieve the name of each employee who works on all the projects controlled by department number

SELECT DISTINCT E.NAME

FROM EMPLOYEE E

WHERE NOT EXISTS (

SELECT P.PNO

FROM PROJECT P

WHERE NOT EXISTS (

SELECT \*

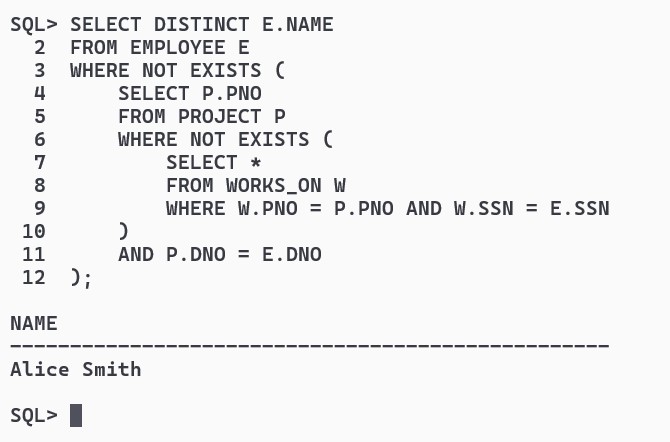
FROM WORKS\_ON W

WHERE W.PNO = P.PNO AND W.SSN = E.SSN

)

AND P.DNO = E.DNO

);



5 (use NOT EXISTS operator).

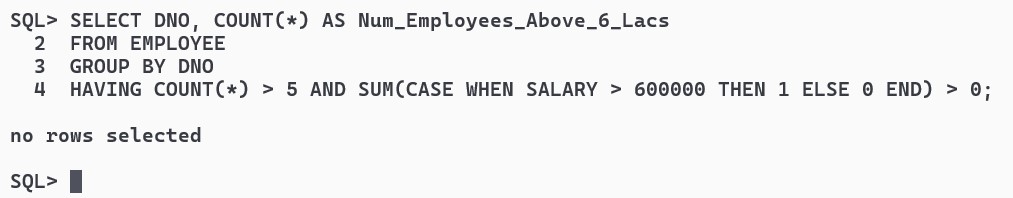
5.For each department that has more than five employees, retrieve the department number and the number of its employees who are making more than Rs. 6,00,000.

SELECT DNO, COUNT(\*) AS Num\_Employees\_Above\_6\_Lacs

FROM EMPLOYEE

GROUP BY DNO

HAVING COUNT(\*) > 5 AND SUM(CASE WHEN SALARY > 600000 THEN 1 ELSE 0 END) > 0;



B. Write a program in PL/SQL to create a procedure to displays the GCD of nos.

CREATE OR REPLACE PROCEDURE Calculate\_GCD(x IN NUMBER, y IN NUMBER) AS

num1 NUMBER := x;

num2 NUMBER := y;

gcd NUMBER;

BEGIN

WHILE num2 != 0 LOOP

gcd := num1;

num1 := num2;

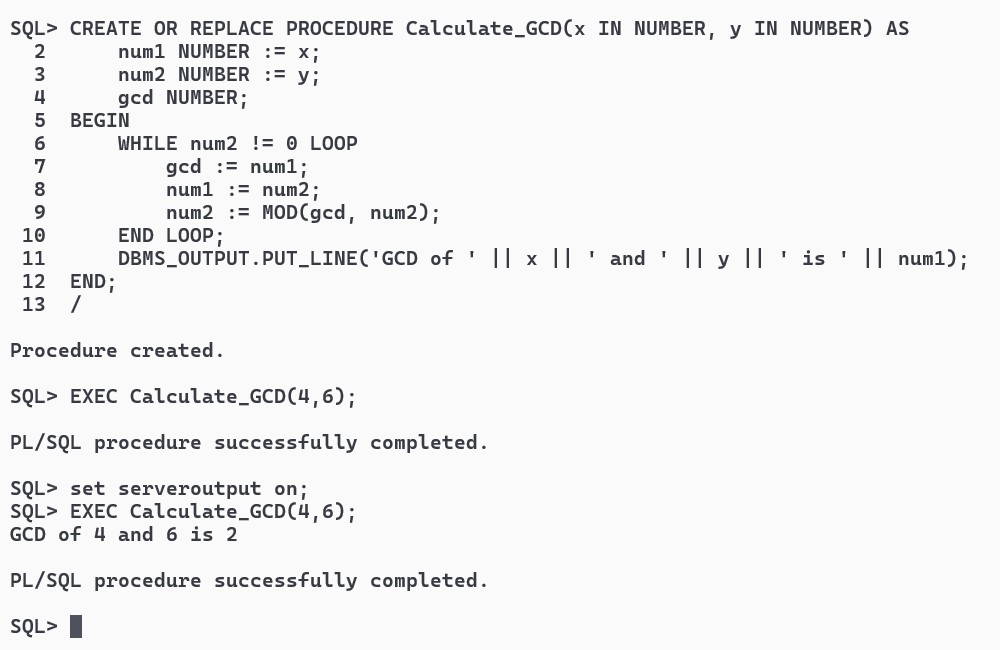
num2 := MOD(gcd, num2);

END LOOP;

DBMS\_OUTPUT.PUT\_LINE('GCD of ' || x || ' and ' || y || ' is ' || num1);

END;

/



C.Write a program in PL/SQL to create a cursor displays the name and salary of each employee in the EMPLOYEES table whose salary is less than that specified by a passed-in parameter value.

CREATE OR REPLACE PROCEDURE DISSAL(salary\_limit IN NUMBER) AS

CURSOR Employee\_Cur IS

SELECT NAME, SALARY

FROM EMPLOYEE

WHERE SALARY < salary\_limit;

emp\_name EMPLOYEE.NAME%TYPE;

emp\_salary EMPLOYEE.SALARY%TYPE;

BEGIN

OPEN Employee\_Cur;

LOOP

FETCH Employee\_Cur INTO emp\_name, emp\_salary;

EXIT WHEN Employee\_Cur%NOTFOUND;

DBMS\_OUTPUT.PUT\_LINE('Name: ' || emp\_name || ', Salary: ' || emp\_salary);

END LOOP;

CLOSE Employee\_Cur;

END;

/

