5A)Write PL/SQL Program to find biggest of three number using nested if.

DECLARE

num1 NUMBER;

num2 NUMBER;

num3 NUMBER;

max\_num NUMBER;

BEGIN

DBMS\_OUTPUT.PUT('Enter the first number: ');

num1 := &1;

DBMS\_OUTPUT.PUT('Enter the second number: ');

num2 := &2;

DBMS\_OUTPUT.PUT('Enter the third number: ');

num3 := &3;

IF num1 > num2 THEN

IF num1 > num3 THEN

max\_num := num1;

ELSE

max\_num := num3;

END IF;

ELSE

IF num2 > num3 THEN

max\_num := num2;

ELSE

max\_num := num3;

END IF;

END IF;

DBMS\_OUTPUT.PUT\_LINE('The biggest number is: ' || max\_num);

END;

Enter the first number: 10

Enter the second number: 20

Enter the third number: 15

The biggest number is: 20

5B)create table student(rollno,name,age,branch,percentage,address);  
i.Insert 6 fields & display the table.  
ii.update 4 data fields using update command

CREATE TABLE student (

rollno NUMBER,

name VARCHAR2(50),

age NUMBER,

branch VARCHAR2(50),

percentage NUMBER,

address VARCHAR2(100));

INSERT INTO student VALUES (1, 'John Doe', 20, 'Computer Science', 85.5, '123 Main St');

INSERT INTO student VALUES (2, 'Jane Smith', 22, 'Electrical Engineering', 78.2, '456 Elm St');

INSERT INTO student VALUES (3, 'Mike Johnson', 21, 'Mechanical Engineering', 92.7, '789 Oak St');

INSERT INTO student VALUES (4, 'Emily Davis', 19, 'Chemical Engineering', 81.9, '321 Pine St');

INSERT INTO student VALUES (5, 'David Wilson', 23, 'Civil Engineering', 77.6, '654 Maple St');

INSERT INTO student VALUES (6, 'Sarah Anderson', 20, 'Information Technology', 89.3, '987 Cedar St');

Display the table:

SELECT \* FROM student;

1. Update 4 data fields using the UPDATE command:

UPDATE student SET name = 'Mark Thompson', age = 24, percentage = 95.2, address = '555 Walnut St' WHERE rollno = 3; UPDATE student SET percentage = 84.7, address = '222 Oak St' WHERE rollno = 4; UPDATE student SET age = 21, address = '777 Elm St' WHERE rollno = 5; UPDATE student SET name = 'Jessica Taylor', branch = 'Computer Science' WHERE rollno = 6;

6A)Write PL/SQL CODE to demonstrate CASE.

DECLARE

grade CHAR(1);

marks NUMBER := 75;

BEGIN

CASE

WHEN marks >= 90 THEN grade := 'A'

WHEN marks >= 80 THEN grade := 'B'

WHEN marks >= 70 THEN grade := 'C'

WHEN marks >= 60 THEN grade := 'D'

ELSE grade := 'F'

END CASE;

DBMS\_OUTPUT.PUT\_LINE('Marks: ' || marks);

DBMS\_OUTPUT.PUT\_LINE('Grade: ' || grade);

END;

6B)Write a any 4 Queries using Conversion functions date functions (Sysdate, next\_day, add\_months, last\_day, months\_between, least, greatest, trunc, round, to\_char, to\_date)

Certainly! Here are four queries that utilize various conversion functions and date functions:

1. Query using `SYSDATE` and `TO\_CHAR` to display the current date in a specific format:

```sql

SELECT TO\_CHAR(SYSDATE, 'DD-MON-YYYY HH24:MI:SS') AS current\_date FROM dual;

```

This query retrieves the current date and time using `SYSDATE` and converts it to a specific format using `TO\_CHAR`.

2. Query using `NEXT\_DAY` to find the next occurrence of a specific day of the week:

```sql

SELECT NEXT\_DAY(SYSDATE, 'SATURDAY') AS next\_saturday FROM dual;

```

This query retrieves the next Saturday from the current date using `NEXT\_DAY`.

3. Query using `ADD\_MONTHS` and `TRUNC` to calculate a future date by adding a specific number of months:

```sql

SELECT TRUNC(ADD\_MONTHS(SYSDATE, 6), 'MM') AS future\_date FROM dual;

```

This query adds 6 months to the current date using `ADD\_MONTHS`, truncates the result to the beginning of the month using `TRUNC`, and retrieves the future date.

4. Query using `MONTHS\_BETWEEN`, `LEAST`, and `GREATEST` to calculate the age range in months between two dates:

```sql

SELECT LEAST(MONTHS\_BETWEEN(SYSDATE, birthdate), GREATEST(MONTHS\_BETWEEN(SYSDATE, birthdate) - 12, 0)) AS age\_in\_months FROM your\_table;

```

Replace `your\_table` with the appropriate table name, and `birthdate` with the column name that holds the birthdate information. This query calculates the age in months between the current date and the birthdate, and restricts the result to a minimum of 0 months and a maximum of 12 months (i.e., the first year).

Feel free to adjust these queries as per your specific requirements and column names in your database.