EXP 10

BRANCHING IN SQL

SIMPLE IF

1.WRITE A PL/SQL PROGRAM TO FIND A SQUARE OF GIVEN NUMBER

```
DECLARE
   a NUMBER;
BEGIN
   a := 2;

   IF a > 0 THEN
      a := a * a;
      dbms_output.put_line('square of number ' || a);
   END IF;
END;/
```

```
square of number 4
```

IF THEN ELSE

2.BIGGEST OF TWO NYMBERS

```
DECLARE
   x NUMBER := 10;
BEGIN
   dbms_output.put_line('Before entering loop, x is: ' || x);

   LOOP
      dbms_output.put_line(x);
      x := x + 10;

      IF x > 50 THEN
         EXIT;
      END IF;
   END LOOP;

   dbms_output.put_line('After exiting loop, x is: ' || x);
END;/
```

```
8 is Greater than 5
```

ELSE IF STATEMENT

TO FIND LARGEST OF THREE NUMBERS

```
DECLARE
   a NUMBER := 5;
   b NUMBER := 10;
   c NUMBER := 2;
BEGIN
   IF a > b AND a > c THEN
      dbms_output.put_line('Greater value is ' || a);
   ELSIF b > c THEN
      dbms_output.put_line('Greater value is ' || b);
   ELSE
      dbms_output.put_line('Greater value is ' || c);
   END IF;
END;/
```

```
Greater value is 10
```

NESTED IF

FIND TOTAL AND AVERAGE OF 6 SUBJECTS AND DISPLAY GRADE

```
DECLARE
   java NUMBER := 30;
   dbms NUMBER := 30;
   co NUMBER := 30;
   se NUMBER := 30;
   es NUMBER := 30;
   ppl NUMBER := 30;
   total NUMBER;
   avg NUMBER;
   per NUMBER;
BEGIN
   total := java + dbms + co + se + es + ppl;
   avg := total / 6;
   per := (total / 180) * 100;

   IF java >= 50 AND dbms >= 50 AND co >= 50 AND se >= 50 AND es >= 50 AND ppl >= 50
THEN
      IF per > 75 THEN
         dbms_output.put_line('Percentage is ' || per);
         dbms_output.put_line('Grade A');
      ELSIF per > 65 THEN
         dbms_output.put_line('Percentage is ' || per);
         dbms_output.put_line('Grade B');
      ELSIF per > 50 THEN
         dbms_output.put_line('Percentage is ' || per);
         dbms_output.put_line('Grade C');
```

```
      ELSE
         dbms_output.put_line('Invalid Input');
      END IF;
   ELSE
      dbms_output.put_line('Fail');
      dbms_output.put_line('No grade');
   END IF;

   dbms_output.put_line('Total is ' || total);
dbms_output.put_line('Average is ' || avg);
END;/
```

```
Fail
No grade
Total is 180
Average is 30
```

ITERATION IN SQL

SIMPLE LOOP

program to implement nested loop

```
DECLARE
   i NUMBER := 5;
   j NUMBER := 0;
BEGIN
   LOOP
      LOOP
         DBMS_OUTPUT.PUT_LINE('*');
         j := j + 1;
         EXIT WHEN j > i;
      END LOOP;

      i := i - 1;

      EXIT WHEN i < 0;
   END LOOP;
END;/
```

==========

implement simple loop

```
DECLARE
   x NUMBER := 10;
BEGIN
   dbms_output.put_line('Before entering loop, x is: ' || x);

   LOOP
```

```
      dbms_output.put_line(x);
      x := x + 10;

      IF x > 50 THEN
         EXIT;
      END IF;
   END LOOP;

   dbms_output.put_line('After exiting loop, x is: ' || x);
END;/

efore entering loop, x is: 10
10
20
30
40
50
After exiting loop, x is: 60


================================================================
FOR LOOP


check prime or not

DECLARE
   n NUMBER;
   i NUMBER;
   temp NUMBER;
BEGIN
   n := &n; -- Input the number
   temp := 1;

   FOR i IN 2..n/2 LOOP
      IF MOD(n, i) = 0 THEN
         temp := 0;
         EXIT;
      END IF;
   END LOOP;

   IF temp = 1 THEN
      DBMS_OUTPUT.PUT_LINE('YES');
   ELSE
      DBMS_OUTPUT.PUT_LINE('NO');
   END IF;
END;/

13
YES
```

4
NO

==========

WHILE LOOP

FIND THE SUM OF DIGITS IN A GIVEN NUMBER

```
DECLARE
   n NUMBER;
   sums NUMBER := 0;
   temp NUMBER;
BEGIN
   n := 789; -- Input the number 789

   WHILE n <> 0 LOOP
      temp := MOD(n, 10); -- Calculate the remainder when dividing by 10, temp = 9
for the first iteration
      sums := sums + temp; -- Add the remainder to the running total, sums = 0 + 9 =
9
      n := TRUNC(n / 10); -- Remove the last digit, n = 78 after the first iteration
   END LOOP;

   DBMS_OUTPUT.PUT_LINE('sum=' || sums); -- Print the final sum of digits, 'sum=24'
END;/
```

sum=24

==========

SELECTION INPL/SQL

SIMPLE CASE

program to display the appreciate for given grade

```
DECLARE
   grade CHAR(1) := 'A';
BEGIN
   CASE grade
      WHEN 'A' THEN
         dbms_output.put_line('EXCELLENT');
      WHEN 'B' THEN
         dbms_output.put_line('VERY GOOD');
      WHEN 'C' THEN
         dbms_output.put_line('GOOD');
      WHEN 'D' THEN
```

```
         dbms_output.put_line('AVERAGE');
      ELSE
         dbms_output.put_line('UNKNOWN GRADE');
   END CASE;
END;
/
```

```
EXCELLENT
```

```
============================================================
```

```
SEARCHED CASE

 TO IMPLEMENT SEARCHED CASE
```

```
DECLARE
   a NUMBER;
BEGIN
   a := &a;

   CASE
      WHEN a = 0 THEN
         dbms_output.put_line('sunday');
      WHEN a = 1 THEN
         dbms_output.put_line('monday');
      WHEN a = 2 THEN
         dbms_output.put_line('tuesday');
      WHEN a = 3 THEN
         dbms_output.put_line('wednesday');
      WHEN a = 4 THEN
         dbms_output.put_line('thursday');
      WHEN a = 5 THEN
         dbms_output.put_line('friday');
      WHEN a = 6 THEN
         dbms_output.put_line('saturday');
      ELSE
         dbms_output.put_line('invalid');
   END CASE;
END;
/
```

```
a as 1

monday

==================================================
```