FECTING VALUES IN A RECORD:

DECLARE

     CURSOR c_emp_cursor IS

      SELECT EMPLOYEE_ID, LAST_NAME FROM EMPL

      WHERE AGE = 69;

     v_emp_record c_emp_cursor%ROWTYPE;

    BEGIN

    OPEN c_emp_cursor;

    LOOP

     FETCH c_emp_cursor INTO v_emp_record;

    EXIT WHEN c_emp_cursor%NOTFOUND;

    DBMS_OUTPUT.PUT_LINE (v_emp_record.EMPLOYEE_ID || ' ' ||v_emp_record.LAST_NAME);

    END LOOP;

    CLOSE c_emp_cursor;

END;

```
DECLARE          CURSOR c_emp_cursor IS          SELECT EMPLOYEE_ID, LAST_NAME FROM EMPL          WHERE AGE = 69;          v_emp_record
c_emp_cursor%ROWTYPE;          BEGIN          OPEN c_emp_cursor;          LOOP          FETCH c_emp_cursor INTO v_emp_record;          EXIT WHEN
c_emp_cursor%NOTFOUND;          DBMS_OUTPUT.PUT_LINE (v_emp_record.EMPLOYEE_ID || ' ' ||v_emp_record.LAST_NAME);          END LOOP;          CLOSE
c_emp_cursor; END;

2 BABU
5 BUJJI
```

Using %ROWCOUNT and %NOTFOUND attributes:

DECLARE

     CURSOR c_emp_cursor IS

      SELECT EMPLOYEE_ID, LAST_NAME FROM EMPL

      WHERE AGE = 57;

     v_emp_record c_emp_cursor%ROWTYPE;

    BEGIN

    OPEN c_emp_cursor;

    LOOP

     FETCH c_emp_cursor INTO v_emp_record;

     EXIT WHEN c_emp_cursor%ROWCOUNT > 10 OR c_emp_cursor%NOTFOUND;

     DBMS_OUTPUT.PUT_LINE (v_emp_record.EMPLOYEE_ID || ' ' || v_emp_record.LAST_NAME);

END LOOP;

CLOSE c_emp_cursor;

END;

```
DECLARE          CURSOR c_emp_cursor IS          SELECT EMPLOYEE_ID, LAST_NAME FROM EMPL          WHERE AGE = 57;          v_emp_record
c_emp_cursor%ROWTYPE;          BEGIN          OPEN c_emp_cursor;          LOOP          FETCH c_emp_cursor INTO v_emp_record;          EXIT WHEN
c_emp_cursor%ROWCOUNT > 10 OR c_emp_cursor%NOTFOUND;          DBMS_OUTPUT.PUT_LINE (v_emp_record.EMPLOYEE_ID || ' ' || v_emp_record.LAST_NAME);          END
LOOP;          CLOSE c_emp_cursor;          END;

3 DON
1143 JHAN
```

Using a cursor forloop:

DECLARE

　　CURSOR c_emp_cursor IS

　　 SELECT EMPLOYEE_ID, LAST_NAME FROM EMPL

　　 WHERE AGE = 69;

　　BEGIN

　　 FOR emp_record IN c_emp_cursor

　　 LOOP

　　　DBMS_OUTPUT.PUT_LINE (emp_record.EMPLOYEE_ID || ' ' ||emp_record.LAST_NAME);

　　 END LOOP;

　　END;

```
DECLARE          CURSOR c_emp_cursor IS          SELECT EMPLOYEE_ID, LAST_NAME FROM EMPL          WHERE AGE = 69;          BEGIN          FOR emp_record
IN c_emp_cursor          LOOP          DBMS_OUTPUT.PUT_LINE (emp_record.EMPLOYEE_ID || ' ' ||emp_record.LAST_NAME);          END LOOP;          END;

2 BABU
5 BUJJI
```

Using a cursor for loop with subqueries:

BEGIN

　　FOR emp_record IN (SELECT EMPLOYEE_ID, LAST_NAME FROM EMPL

　　WHERE AGE = 69)

　　LOOP

　　 DBMS_OUTPUT.PUT_LINE (emp_record.EMPLOYEE_ID || ' ' ||

　　　emp_record.LAST_NAME);

　　 END LOOP;

　　END;

```
BEGIN          FOR emp_record IN (SELECT EMPLOYEE_ID, LAST_NAME FROM EMPL          WHERE AGE = 69)          LOOP          DBMS_OUTPUT.PUT_LINE
(emp_record.EMPLOYEE_ID || ' ' ||          emp_record.LAST_NAME);          END LOOP;          END;

2 BABU
5 BUJJI
```

Using cursor table in ascending order:

```
DECLARE

   CURSOR person_cursor IS

      SELECT * FROM PER ORDER BY AGE ASC;

   person_record PER %ROWTYPE;

BEGIN

   OPEN person_cursor;


   LOOP

      FETCH person_cursor INTO person_record;

      EXIT WHEN person_cursor%NOTFOUND;

      DBMS_OUTPUT.PUT_LINE(', Name: ' || person_record.FIRST_NAME || ' ' ||
person_record.LAST_NAME || ', Age: ' || person_record.AGE || ', CITY: ' || person_record.CITY);

   END LOOP;


   CLOSE person_cursor;

END;

/
```

```
, Name: AMMADUU NANNALUU, Age: 16, CITY: HEART
, Name: NANI KAMABATTULA, Age: 19, CITY: GUNTUR
, Name: PRABATH DOKUPARTHI, Age: 20, CITY: VIJAYAWADA
, Name: SAI BATTERY, Age: 21, CITY: CHIRALA
, Name: YESWANTH JHAN, Age: 22, CITY: NOWLURU
, Name: VENNU GOSIKA, Age: 25, CITY: MANGALAGIRI
, Name: GOPI SOLLU, Age: 26, CITY: CHENNAI
, Name: SUMAN SEELAM, Age: 30, CITY: ATMAKUR
, Name: RAMESH PAKODI, Age: 35, CITY: TENALI
, Name: SRINU BATTULA, Age: 46, CITY: PIDUGURALLA
```

Problem on GOTO Statement :

```
DECLARE

   a number(2) := 30;
```

```
BEGIN

  <<loopstart>>

  -- while loop execution

  WHILE a < 50 LOOP

    dbms_output.put_line ('value of a: ' || a);

    a := a + 1;

    IF a = 35 THEN

      a := a + 1;

      GOTO loopstart;

    END IF;

  END LOOP;

END;
```



```
DECLARE     a number(2) := 30;   BEGIN     <<loopstart>>     -- while loop execution    WHILE a < 50 LOOP      dbms_output.put_line ('value of a: ' || a);
a := a + 1;        IF a = 35 THEN       a := a + 1;       GOTO loopstart;      END IF;    END LOOP; END;

value of a: 30
value of a: 31
value of a: 32
value of a: 33
value of a: 34
value of a: 36
value of a: 37
value of a: 38
value of a: 39
value of a: 40
value of a: 41
value of a: 42
value of a: 43
value of a: 44
value of a: 45
value of a: 46
value of a: 47
value of a: 48
value of a: 49
```

To handle the exception when a division by zero occurs.

```
DECLARE

dividend   NUMBER := 10;

divisor    NUMBER := 2;

result     NUMBER;

BEGIN

  BEGIN

result := dividend / divisor;

    DBMS_OUTPUT.PUT_LINE('Result: ' || result);

  EXCEPTION

    WHEN ZERO_DIVIDE THEN

      DBMS_OUTPUT.PUT_LINE('Error: Division by zero');

    END;
```

END;

```
DECLARE dividend   NUMBER := 10; divisor    NUMBER := 2; result     NUMBER; BEGIN    BEGIN result := dividend / divisor;     DBMS_OUTPUT.PUT_LINE('Result: ' ||
result);    EXCEPTION      WHEN ZERO_DIVIDE THEN         DBMS_OUTPUT.PUT_LINE('Error: Division by zero');          END;          END;
Result: 5
```

(3.2)

INTO CLAUSE:

DECLARE

v_emp_lname EMPL.LAST_NAME%TYPE;

BEGIN

SELECT LAST_NAME

INTO v_emp_lname

FROM EMPL

WHERE EMPLOYEE_ID = 1143;

DBMS_OUTPUT.PUT_LINE('His last name is ' || v_emp_lname);

END;

```
DECLARE v_emp_lname EMPL.LAST_NAME%TYPE; BEGIN SELECT LAST_NAME INTO v_emp_lname FROM EMPL WHERE EMPLOYEE_ID = 1143; DBMS_OUTPUT.PUT_LINE('His last name is ' ||
v_emp_lname); END;
His last name is JHAN
```

RETRIEWING DATA:

DECLARE

v_emp_LAST_NAME EMPL.LAST_NAME%TYPE;

v_emp_AGE EMPL.AGE%TYPE;

BEGIN

SELECT LAST_NAME, AGE

INTO v_emp_LAST_NAME, v_emp_AGE

FROM EMPL

WHERE EMPLOYEE_ID = 7717;

DBMS_OUTPUT.PUT_LINE('LAST_NAME: ' || v_emp_LAST_NAME);

DBMS_OUTPUT.PUT_LINE('AGE: '|| v_emp_AGE);

END;

```
DECLARE v_emp_LAST_NAME EMPL.LAST_NAME%TYPE; v_emp_AGE EMPL.AGE%TYPE; BEGIN SELECT LAST_NAME, AGE INTO v_emp_LAST_NAME, v_emp_AGE FROM EMPL WHERE EMPLOYEE_ID = 7717;
DBMS_OUTPUT.PUT_LINE('LAST_NAME: ' || v_emp_LAST_NAME); DBMS_OUTPUT.PUT_LINE('AGE: '|| v_emp_AGE); END;
LAST_NAME: NANI
AGE: 54
```

CREATION OF PARTIAL TABLE AND INSERT COMMIT VALUES:

CREATE TABLE pairtable (

   column1 INT,

   column2 INT

);

BEGIN

INSERT INTO pairtable VALUES (1, 2);

COMMIT;

END;

```
CREATE TABLE pairtable (     column1 INT,     column2 INT )

Table created. 0.01 seconds

BEGIN INSERT INTO pairtable VALUES (1, 2); COMMIT; END;
```

ROLL BACK:

BEGIN INSERT INTO pairtable VALUES (3, 4); ROLLBACK; INSERT INTO pairtable VALUES (5, 6); COMMIT; END;

EDIT	COLUMN1	COLUMN2
✎	1	2
✎	5	6

SAVEPOINT:

BEGIN

INSERT INTO pairtable VALUES (7, 8);

SAVEPOINT my_sp_1;

INSERT INTO pairtable VALUES (9, 10);

SAVEPOINT my_sp_2;

INSERT INTO pairtable VALUES (11, 12);

ROLLBACK to my_sp_1;

INSERT INTO pairtable VALUES (13, 14);

COMMIT;

END;

EDIT	COLUMN1	COLUMN2
✎	1	2
✎	5	6
✎	7	8
✎	13	14