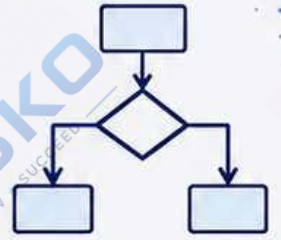


1010101010
0101010101
1010101010
0101010101



Notes





Unit III -

Any two out of 4 /
3 marks each / any one -

Q1.* → What is PL/SQL? Explain the block structure variable declaration & Exception handling.

PL/SQL -

PL/SQL stands for Procedural Language / Structure Query Language is a Oracle's programming language that combines SQL with procedural features like variables, loops, conditions and exception handling.

It allows execution of multiple SQL statements as a single unit called a block.

PL/SQL Block Structure :

A PL/SQL program is written in the form of blocks. A block has three main sections -

```

[ DECLARE ]
-- Variable declarations
BEGIN
-- Executable statements
EXCEPTION
-- Error handling statements
END %;
```


Eg-

DECLARE

num 1 NUMBER := 10;

num 2 NUMBER := 0;

result NUMBER;

BEGIN

result := num 1 / num 2;

-- Will rise exception

EXCEPTION

WHEN ZERO_DIVIDE THEN

DBMS_OUTPUT.PUT_LINE('Can't divided by 0');

END/;

1. Declaration Section (Optional) - related to

This section begins with the keyword **DECLARE**.

Variables, constants, cursor, and user defined types are declared here.

2. BEGIN / Execution Section (mandatory)

Begins with the key word **BEGIN**;

This part contains SQL statements, calculations, loops and conditions.

It must contain at least one executable statement.

3. Exception Section (Optional) -

Begins with the keyword **EXCEPTION**.
It handles runtime errors to prevent abnormal termination.

Variable Declaration

Variables are declared in **DECLARE** section. A variable must have a name, datatype and initial value.

Syntax -

variable-name datatype [:= value] ;

Salary NUMBER := 30000 ;

Features of PL/SQL.

1) Block Structured Language

PL/SQL programs are divided into blocks consisting of declaration, execution, and exception sections, which improves clarity and organization.

2) Tight Integration with SQL

PL/SQL allows SQL statements like SELECT, INSERT, UPDATE and DELETE to be used directly inside the program.

3) Procedural Features

It supports programming concepts such as variables, constants, loops, conditional statements and control structures.

4) Exception Handling-

PL/SQL provides a powerful exception handling to handle runtime errors gracefully.

5) Stored Programs

It supports procedures, functions and triggers which can be stored in a database for reuse.

6) Security-

Q2.* What is Cursor? Explain its types, attributes and life cycle.



Cursor -

A cursor is a pointer used in PL/SQL to handle and process the result of a query row by row.

When a SELECT statement returns multiple rows, PL/SQL cannot store them directly into variable.

A cursor allows you to fetch each row individually, process it and then move to the next row.

A cursor is a pointer used in PL/SQL to handle and process the result of a query row-by-row.

When a SELECT statement returns multiple rows, then the PL/SQL is not able to store them in variable.

A cursor is a pointer used in PL/SQL to handle and process the result row by row of the query.

classmate

Date _____

Page _____

Types of Cursor

PL/SQL supports two types of cursors.

1. Implicit Cursor
2. Explicit Cursor

Both are used to handle rows returned by SQL queries, but they are created and used.

1. Implicit Cursor

An implicit cursor is created automatically by PL/SQL whenever a SQL statement is executed.

You do not declare or control it manually.

PL/SQL creates implicit cursors for:

INSERT, UPDATE, DELETE, SELECT...INTO

Eg-

```
SELECT name INTO v_name FROM
employee WHERE id = 101;
```

PL/SQL automatically opens, fetches and closes the cursor.

2. Explicit Cursor-

An explicit cursor is created manually by the programmers for queries that return more than one row.

It gives full control over row-by-row processing.

Explicit cursor is use when you need to -

- i) Process multiple rows one at a time.
- ii) Perform operations inside loops.
- iii) Handle complex queries.

Eg-

```

DECLARE c1 IS
    SELECT name FROM
    student; -- cursor
    v_name student.name %TYPE;
BEGIN
    OPEN c1;
    -- Open
    FETCH c1 INTO v_name;
    -- Fetch one row
    DBMS_OUTPUT.PUT_LINE (v_name); -- display
    close c1; -- close    END;
```


classmate

Date _____
Page _____

Differentiate between Implicit & Explicit.

```

DECLARE CURSOR
c1 IS
SELECT name FROM students;
v_name student.name%TYPE;
BEGIN
    OPEN c1;
    LOOP
        FETCH c1 INTO v_name;
        DBMS_OUTPUT.PUT_LINE ('v_name');
    END LOOP;
    CLOSE c1;
END;

```


* Attributes of Cursor -

Cursor attributes are used to check the status of a cursor during execution. They help you understand whether a cursor has fetched rows, how many rows, and whether it is open or closed.

There are four main attributes.

1) %FOUND - It returns TRUE if the last

FETCH returned a row.

Returns FALSE if no row was fetched.

eg -

IF %FOUND THEN

DBMS_OUTPUT.PUT_LINE ('A row was fetched');

END IF;

2) %NOTFOUND -

It returns TRUE if the last FETCH did not return any row.

Used to stop loops when no more rows exist.

eg -

EXIT WHEN %NOTFOUND;

Commonly used inside cursor loops.

classmate

Date _____
Page _____

3) %ROWCOUNT-

Returns the number of rows fetched so far from the cursor.

eg-

```
DBMS_OUTPUT.PUT_LINE ('Row fetched: '  
                        || c1%ROWCOUNT);
```

4) %ISOPEN

Return TRUE if the cursor is open
Return FALSE if the cursor is closed

eg-

```
IF c1%ISOPEN THEN  
    CLOSE c1;  
END IF;
```