PL/SQL Stored Procedures

In this session, you will learn:

- What is PL/SQL
- Introducing PL/SQL block structure and anonymous block
- How to declare and use variables
- Introduction to PL/SQL Stored Procedures
- How to develop a simple stored procedure
- How to use conditional statements
- How to use various loop statements
- How to call a stored Procedure

What is PL/SQL

- Procedural Language(PL) that extends the Structured Query Language(SQL).

Why to use PL/SQL?

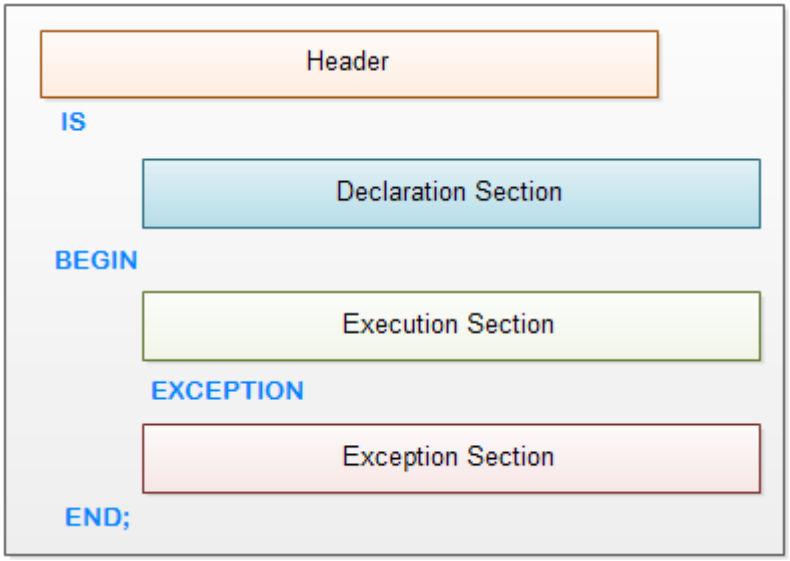
- ✓ high performance
- ✓ portability
- ✓ high productivity
- ✓ scalability
- ✓ manageability
- ✓ support for Object-Oriented Programming

Introducing PL/SQL block structure and anonymous block

- PL/SQL program units organize the code into blocks.
- A block without a name is known as an anonymous block.

Syntax

```
[DECLARE]
   Declaration statements;
BEGIN
   Execution statements;
  [EXCEPTION]
      Exception handling statements;
END;
/
```



PL/SQL block structure – Example

Example 1

```
BEGIN
  NULL;
END;
```

→ PL/SQL procedure successfully completed.

To display database's output on the screen, you need to:

- ✓ SET SERVEROUTPUT ON
- ✓ DBMS_OUTPUT.PUT_LINE

Example 2

```
SET SERVEROUTPUT ON
BEGIN
  DBMS_OUTPUT.PUT_LINE('Hello PL/SQL');
END;
/
```

→ Hello PL/SQL

Variable Declaration and Operation on variable

Example

```
SET SERVEROUTPUT ON;
DECLARE
   v_name Customer.FirstName%TYPE;
   v_phone Customer.Phone%TYPE;
BEGIN
   SELECT FirstName,Phone
   INTO v_name, v_phone
   FROM Customer
   WHERE Customer_Id = 100;
DBMS_OUTPUT.PUT_LINE(v_name);
DBMS_OUTPUT.PUT_LINE(v_phone);
END;
/
```

Variable Anchors

Arun
9852767818

PL/SQL procedure successfully completed.

Introduction to PL/SQL Stored Procedures

- PL/SQL procedure is a named block that does a specific task. It allows you to encapsulate complex business logic and reuse it in both database layer and application layer.

Why to use Stored Procedures?

- ✓ increase the performance of the applications
- ✓ promote reusability
- ✓ promote maintainability
- ✓ secure

How to create a Stored Procedure

The statement CREATE PROCEDURE creates a new procedure.

Syntax

```
CREATE [OR REPLACE] PROCEDURE procedure_name
   [ (parameter [,parameter]) ]
IS
   [declaration_section]
BEGIN
   executable_section
[EXCEPTION
   exception_section]
END [procedure_name];
```

Example for Stored Procedure

Create a procedure named ‘increasePrice’ which takes 2 input parameter

- n_id int
- inc_percent int

This procedure must update the price value of a particular product by increasing it by a percentage which is passed as parameter.

```
CREATE OR REPLACE PROCEDURE increasePrice
(n_id IN INT,
 inc_percent IN INT)
IS
BEGIN
UPDATE Product Set Price = Price + (Price * inc_percent / 100)
WHERE Product_Id = n_id;
END;
/
```

Example for Stored Procedure

How to call a stored procedures?

```
BEGIN
increasePrice(300,5);
END;
/
```

```
EXEC increasePrice(300,5);
```

```
EXECUTE increasePrice(300,5);
```

Example for Stored Procedure using IF-THEN-ELSE

Example

```
CREATE OR REPLACE PROCEDURE classifyProduct
(id IN INT, status OUT varchar2)
IS
amount int;
BEGIN
SELECT price INTO amount from Product where Product_Id = id;
 IF amount >= 8000 THEN
    status := 'High Price Product';
ELSIF amount between 3000 and 5000 THEN
    status := 'Medium Price Product';
ELSIF amount < 3000 THEN
    status := 'Low Price Product';
END IF;
END;
/
```

```
set serveroutput on
DECLARE
status varchar(20);
BEGIN
classifyProduct(1,status);
dbms_output.put_line(status);
END;
/
```

How to drop a Stored Procedure

The statement DROP PROCEDURE drops an existing procedure.

Syntax

```
DROP PROCEDURE  procedure_name;
```

Example

```
DROP PROCEDURE classifyProduct;
```

THANKS