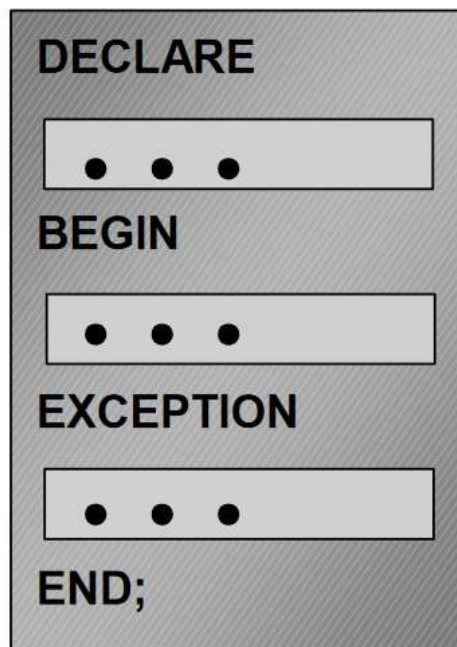


About PL/SQL

Procedural Language/SQL (PL/SQL) is Oracle Corporation's procedural language extension to SQL, the standard data access language for relational databases. PL/SQL offers modern software engineering features such as data encapsulation, exception handling, information hiding, object orientation, and brings state-of-the-art programming to the Oracle Server and toolset. PL/SQL incorporates many of the advanced features of programming languages that were designed during the 1970s and 1980s. It allows the data manipulation and query statements of SQL to be included in block-structured and procedural units of code, making PL/SQL a powerful transaction processing language. With PL/SQL, you can use SQL statements to finesse Oracle data, and PL/SQL control statements to process the data.

Modularize program development



Declaring Variables

1. DECLARE

```
v_variable VARCHAR2(5);
```

BEGIN

```
SELECT column_name
```

```
INTO v_variable
```

```
FROM table_name;
```

```
EXCEPTION WHEN exception_name TH END;
```

Base Scalar Data Types

- `CHAR [(maximum_length)]`
- `VARCHAR2 (maximum_length)`
- `LONG`
- `LONG RAW`
- `NUMBER [(precision, scale)]`
- `BINARY_INTEGER`
- `PLS_INTEGER`
- `BOOLEAN`

Example

2.DECLARE

```
v_job VARCHAR2(9);  
v_count BINARY_INTEGER := 0;  
v_total_sal NUMBER(9,2) := 0;  
v_orderdate DATE := SYSDATE + 7;  
c_tax_rate CONSTANT NUMBER(3,2) := 8.25;  
v_valid BOOLEAN NOT NULL := TRUE;
```

Declaring Variables with the %TYPE Attribute

Declare

```
3.v_name employees.last_name%TYPE;  
v_balance NUMBER(7,2);  
v_min_balance v_balance%TYPE := 10;
```

Using Bind Variables

```
VARIABLE g_salary NUMBER  
BEGIN  
SELECT salary  
INTO :g_salary  
FROM employees  
WHERE employee_id = 178;  
END;
```

DBMS_OUTPUT.PUT_LINE

6. SET SERVEROUTPUT ON

```
DEFINE p_annual_sal = 60000  
DECLARE  
v_sal NUMBER(9,2) := &p_annual_sal;  
BEGIN  
v_sal := v_sal/12;  
DBMS_OUTPUT.PUT_LINE ('The monthly salary is ' ||  
TO_CHAR(v_sal));  
END;
```