Overview of PL/SQL

About PL/SQL

Belong to oracle

- PL/SQL is the procedural extension to SQL with design features of programming languages.
- Data manipulation and query statements of SQL are included within procedural units of code.

Why to use PL/SQL ?

Example: Updating salary according to department number

Dept 10 → raise $100
Dept 20 → raise $150
Dept 30 → raise $200
Dept 40 → raise $240

In SQL
You will do multiple SQL statements

```
Update emp
Set sal=sal+100
Where dept_id=10;

Update emp
Set sal=sal+150
Where dept_id=20;
……
…..
```

In PL/SQL
You can write procedure to do this

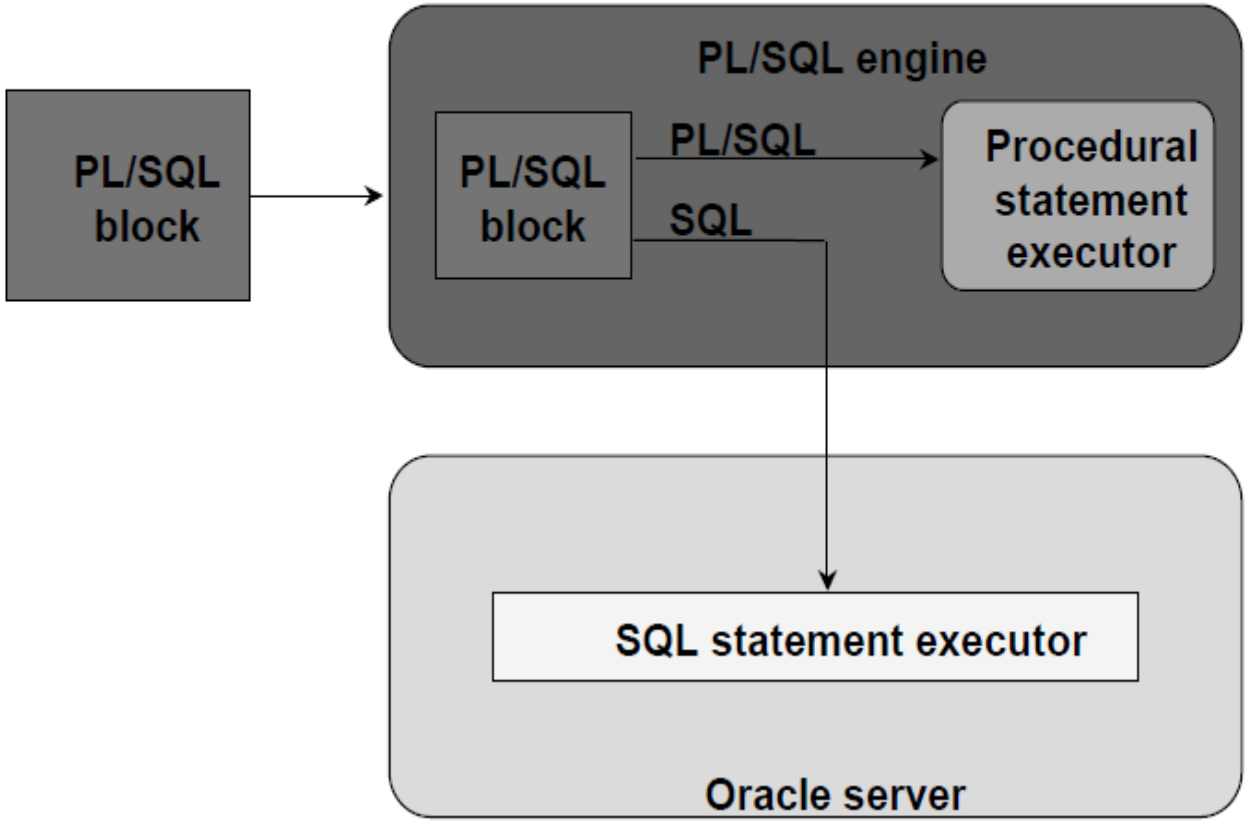
```
Create procedure update_sal
( p_dept_id number , p_amount number )
Is
Begin
…….
…….
…….
End;
```

About PL/SQL

PL/SQL:

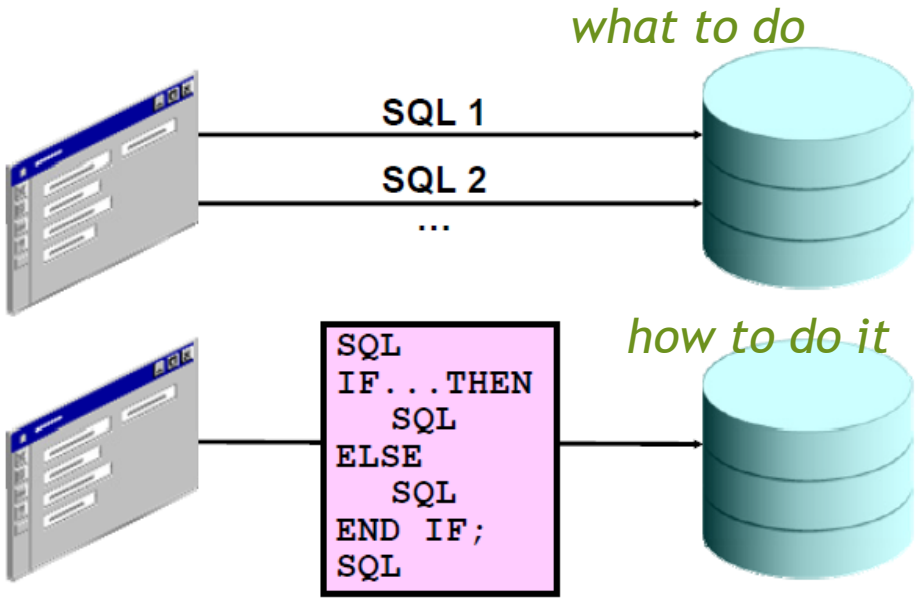
- Provides a block structure for executable units of code. Maintenance of code is made easier with such a well-defined structure.
- Provides procedural constructs such as:
  - Variables, constants, and data types
  - Control structures such as conditional statements and loops
  - Reusable program units that are written once and executed many times

PL/SQL Environment



Benefits of PL/SQL

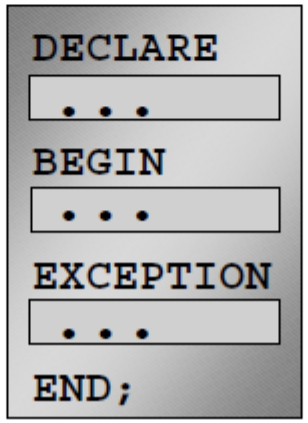
- Integration of procedural constructs with SQL
- Improved performance



- Modularized program development (You can create procedure, function, packages …)
- Integration with Oracle tools (oracle forms, oracle reports …)
- Portability PL/SQL programs can run anywhere an Oracle server runs
- Exception handling

PL/SQL Block Structure

```
DECLARE          – Optional
    Variables, cursors, user-defined exceptions
BEGIN            – Mandatory
    – SQL statements
    – PL/SQL statements
EXCEPTION        – Optional
    Actions to perform when errors occur
END;             – Mandatory
```



Block Types

Subprograms

Anonymous

```
[DECLARE]


BEGIN
   --statements

[EXCEPTION]

END;
```

Procedure

```
PROCEDURE name
IS

BEGIN
   --statements

[EXCEPTION]

END;
```

Function

```
FUNCTION name
RETURN datatype
IS
BEGIN
   --statements
   RETURN value;
[EXCEPTION]

END;
```

Differences Between Anonymous Blocks and Subprograms

Anonymous Blocks	Subprograms
Unnamed PL/SQL blocks	Named PL/SQL blocks
Compiled every time	Compiled only once
Not stored in the database	Stored in the database
Cannot be invoked by other applications	Named and, therefore, can be invoked by other applications
Do not return values	Subprograms called functions must return values.
Cannot take parameters	Can take parameters

Thank You