



Introduction to PL/SQL

Objective: Understand PL/SQL, how it differs from SQL, and its applications in Oracle databases

Introduction to PL/SQL

- ❖ **Hook:** Master PL/SQL in 2025: Build smarter, faster Oracle database solutions!
- ❖ **Objective:** Learn what PL/SQL is, how it differs from SQL, and its role in Oracle databases.
- ❖ **Agenda:** Definition, structure, comparison, applications, and hands-on example.

What is PL/SQL? Where It Runs

- ❖ **Definition:** PL/SQL (Procedural Language/Structured Query Language) is Oracle's procedural extension to SQL.
- ❖ **Execution:** Runs on Oracle Database servers for high-performance, server-side processing.
- ❖ **Purpose:** Combines SQL's data manipulation with procedural logic (e.g., loops, conditionals).

Ultra-simple example:

```
BEGIN  
  DBMS_OUTPUT.PUT_LINE('Welcome to PL/SQL!');  
END;
```

PL/SQL vs SQL (Quick Contrast)

- ❖ **SQL:** Declarative, for querying/manipulating data (e.g., SELECT, INSERT).
- ❖ **PL/SQL:** Procedural, supports programming constructs (e.g., loops, error handling).

Aspect	SQL	PL/SQL
Paradigm	Declarative (what)	Procedural (how)
Execution	Single statements	Blocks, procedures, functions
Use Case	Queries & DML	Business logic, automation, triggers
Error Handling	Limited (codes)	Structured (EXCEPTION block)

Table 1: Comparison table

PL/SQL Block Structure

Core structure: DECLARE (variables), BEGIN (logic), EXCEPTION (error handling), END.

Example: Simple block to check inventory levels:

```
DECLARE
    v_stock NUMBER := 100;
BEGIN
    IF v_stock < 50 THEN
        DBMS_OUTPUT.PUT_LINE('Low stock!');
    END IF;
EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Error occurred');
END;
```

Variables, Constants, and Data Types

- ❖ **Variables:** Declared in DECLARE, e.g., v_salary NUMBER;.
- ❖ **Constants:** Fixed values, e.g., c_tax_rate CONSTANT NUMBER := 0.05;.
- ❖ **Common Data Types:** NUMBER, VARCHAR2, DATE, BOOLEAN, TIMESTAMP, %TYPE, %Rowtype.

Example:

```
DECLARE
    v_emp_count NUMBER := 10;
    c_max_count CONSTANT NUMBER := 100;
BEGIN
    DBMS_OUTPUT.PUT_LINE('Employees: ' || v_emp_count);
END;
```

Applications of PL/SQL

- ❖ Automating tasks (e.g., batch updates).
- ❖ Generating complex reports.
- ❖ Managing transactions and error handling.
- ❖ Triggers for event-driven actions.
- ❖ Reduce network round trips by running logic close to data (server-side).
- ❖ Real-world example: “A bank uses PL/SQL triggers to log suspicious transactions automatically.”

Sample PL/SQL Code

```
BEGIN
    v_tax := p_salary * 0.1;
    DBMS_OUTPUT.PUT_LINE('Salary: ' || p_salary || ', Tax: ' || v_tax);
EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Error in calculation');
END;
```


Conclusion and Next Steps

- ❖ Summary: PL/SQL is a powerful tool for Oracle database programming.
- ❖ Next steps: Try PL/SQL on Oracle LiveSQL (link: livesql.oracle.com).
- ❖ Call to action: Subscribe, comment, or ask questions on YouTube.
- ❖ Contact info or YouTube channel link.