

Introduction

Mag. Thomas Griesmayer

SQL vs. Java

SQL (Structure Query Language)

- WHAT

```
SELECT *  
FROM STUDENTS  
WHERE AGE < 30;
```



Java (Object Oriented Language)

- HOW

```
for (Student s: students) {  
    if (s.getAge() < 30)  
        System.out.println(s);  
}
```

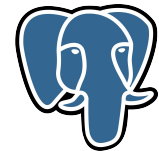


PL/SQL (Procedural Language/Structured Query Language)

- It is the Oracle's procedural extension to SQL.
- It allows users to write code that includes loops, conditions, and variables, making it more powerful and flexible than standard SQL.
- PL/SQL is used for writing complex queries, stored procedures, functions, and triggers, allowing for more control and reusability in database operations.

Procedural Languages

- Oracle Database – PL/SQL (Procedural Language/SQL). Contains loops, conditions, exception handling, and modular programming
- MySQL – SQL/PSM (Procedural Language). Contains loops and conditionals, but it's less feature-rich than PL/SQL or T-SQL
- Microsoft SQL Server – T-SQL (Transact-SQL). Contains variables, loops, conditionals, and error handling, similar to PL/SQL
- PostgreSQL – PL/pgSQL (Procedural Language/PostgreSQL). Contains control structures and is similar to PL/SQL, but it also supports multiple other procedural languages like PL/Python and PL/Perl
- IBM Db2 – SQL PL (SQL Procedural Language). Contains control flow, variable declaration, and exception handling, similar to PL/SQL and T-SQL

The Oracle logo, consisting of the word "ORACLE" in a bold, red, sans-serif font.The MySQL logo, featuring the word "MySQL" in a blue and orange font with a blue silhouette of a dolphin above the "Y".The Microsoft SQL Server logo, featuring a red wireframe globe icon above the text "Microsoft SQL Server" in a black font.The IBM logo, consisting of the letters "IBM" in a blue, striped, sans-serif font.

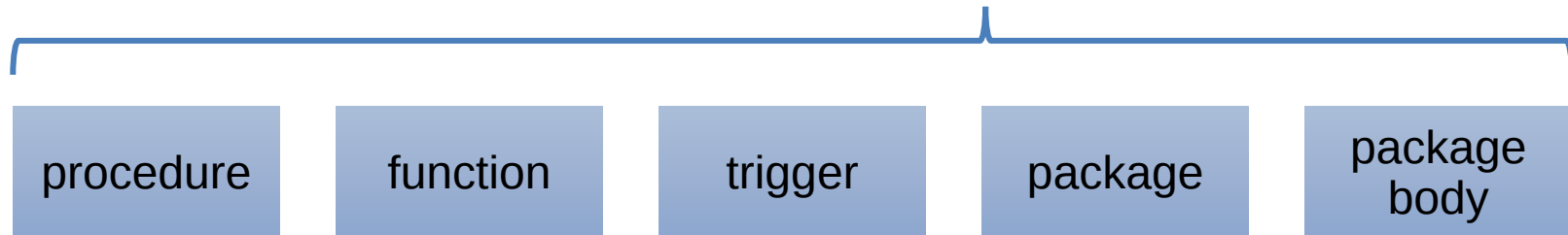
unnamed vs. named

- An unnamed block (anonymous block) in PL/SQL does not have an explicit name.
- These blocks are not stored in the database and are typically used for quick, one-time execution of PL/SQL logic.
- Once executed, they do not persist and cannot be invoked again unless written and executed again.

```
BEGIN
    DBMS_OUTPUT.PUT_LINE('...');
END;
```

- A named PL/SQL block is a reusable PL/SQL code structure that has a name.
- These named blocks are typically used for modularizing code in the form of procedures, functions, packages, and triggers.

```
CREATE OR REPLACE PROCEDURE
    procedure_name IS
BEGIN
    DBMS_OUTPUT.PUT_LINE('Hello
World!');
END;
```



Structure

```
CREATE OR REPLACE PROCEDURE procedure_name  
                                (param_name IN datatype) IS
```

```
/*
```

The procedure header includes the procedure's name and the parameters it accepts. You can define parameters as IN (input), OUT (output), or IN OUT (both input and output).

```
*/
```

```
DECLARE
```

```
    v_variable datatype;
```

```
/*
```

Declaration section is an optional part to declare local variables, constants, cursors, or types.

```
*/
```

Structure

```
BEGIN
    SELECT column_name INTO v_variable
    FROM    table_name
    WHERE   condition;

    IF v_variable IS NOT NULL THEN
        DBMS_OUTPUT.PUT_LINE(v_variable);
    END IF;
```

/*

The execution section is mandatory and contains the main logic of the procedure. It includes SQL statements (e.g., SELECT, INSERT, UPDATE, DELETE) and control structures (loops, conditions).

*/

Structure

```
BEGIN
    SELECT column_name INTO v_variable
    FROM   table_name
    WHERE  condition;

    IF v_variable IS NOT NULL THEN
        DBMS_OUTPUT.PUT_LINE(v_variable);
    END IF;
```

/*

The execution section is mandatory and contains the main logic of the procedure. It includes SQL statements (e.g., SELECT, INSERT, UPDATE, DELETE) and control structures (loops, conditions).

*/

Structure

```
EXCEPTION
```

```
    WHEN NO_DATA_FOUND THEN
```

```
        DBMS_OUTPUT.PUT_LINE('No data found!');
```

```
    WHEN OTHERS THEN
```

```
        DBMS_OUTPUT.PUT_LINE('An unexpected error occurred.');
```

```
/*
```

The exception handling section is optional and used for handling errors or exceptions that may occur during the execution of the procedure.

You can define specific actions to take when particular exceptions are raised.

```
*/
```

```
END procedure_name;
```

```
/*
```

Every procedure must be concluded with the keyword END;, followed optionally by the procedure's name for clarity.

```
*/
```