PL/SQL - an extension to SQL

Book1 - PL/SQL Fundamentals and BOOK2 - PL/SQL Program Units

PL/SQL is compatible with: Oracle SQL Developer, Oracle J Developer, Oracle DB Express, Oracle EM Cloud Control, APEX

What makes PL/SQL different from SQL:(Why PL/SQL)

1. Robust
2. Dynamic
3. Integrated (Control + Conditional Statements with SQL) → More control over SQL Statements
4. Compatible
5. Portability → Can run anywhere irrespective of OS and Platform
6. Secured
7. Reusability → one of the critical components
8. Exception Handling → Handles it very efficiently and separate blocks can be defined for that

Study Topics using PL/SQL:

a. Control Structures
b. Iterations and Loops (are same as SQL)
c. DB Objects like functions, blocks, Procedures, triggers, Cursors, exceptions.

PL/SQL Applications:

1. Database Designs (Schema – group of logically related objects) creations
    a. DDL Statements (Data Definition Language) like CREATE, ALTER, DROP which affects database schema.
    (DROP Generally not given to normal user)
    b. DML (Data Manipulation Language) like INSERT, UPDATE, DELETE which affects the data of database.
    c. DCL (Data Control Language) like GRANT and REVOKE gives permissions to user.
2. Integration with other applications

3. Database Security (One of the widely used application of PL/SQL)
    a. It provides **Privilege analysis (Assigned by Admin)**
        i. USER1 → Developer → SELECT, CREATE, INSERT
        ii. USER2 → Tester → SELECT
        iii. USER3 → System Admin → DROP, ALTER, MODIFY

Examples of Privileges:

iv. DBMS_SYSTEM_USED_PRIVS → Used for whole schema
v. DBMS_SYSTEM_UNUSED_PRIVS → Used for whole Schema
vi. DBMS_OBJECT_USED_PRIVS → Used for only one particular Object
vii. DBMS_OBJECT_UNUSED_PRIVS → Used for only one particular Object

b. **Data Reduction**: Masking the data
c. **SQL Injection**: Attackers can use SQL Injections to find the credentials of other users in the database. Generally used to check how secure database is. Might destroy Database.

4. Performance/Tuning Enhancement: Optimization of quarries
5. Import/Export of Data: Cross platform migrations

Why we use SQL? – To communicate with the Data (User uses SQL and PL/SQL in order to use data from database)

- SELECT * FROM hr.employee; → for multiple schema with the same columns
- Dynamic Performance View: V$ - MVS (Data Warehousing)

PRIMARY KEY → Unique and not null values -pk
FOREIGN KEY → To display relations between tables -fk
UNIQUE →
CHECK
DEFAULT
NULL
NOT NULL

Code Maintenance:

1. Refactoring – modify code or edit code - restructuring existing computer code—changing the factoring—without changing its external behaviour.
2. Overloading - used to avoid redundant code where the same method name is used multiple times but with a different set of parameters
3. C.I (Continues Integration) – Generally used for servers (Build, Test and Deploy). Automated CI provides all the functionalities after building code, Test and deployment gets done.

   CI = practice of merging all developer working copies to a shared mainline several times a day.

4. Patching – Upgradation, released in form of .exe files. Basically Updates.

PL/SQL Anonymous Block Structures:

1. DECLARE (optional)
2. BEGIN (Mandatory) →needs to have at least one statement in that
3. EXECEPTION (optional) → aka executable → Anything that comes between Begin and End
4. END; (Mandatory) → it's a part of BEGIN…..END

Block Type: Procedure, Function, Anonymous

Enabling Output of a PL/SQL Block

1. To enable output in SQL Developer, execute the following command before running the PL/SQL block:
   SET SERVEROUTPUT ON
2. Use a predefined Oracle package and its procedure in the anonymous block:
   DBMS_OUTPUT.PUT_LINE

Block Types:

1. Procedures → are named objects
2. Functions → are named objects but returns a value of specified data types
3. Anonymous → unnamed blocks, if you execute it first time and want to execute again you have to rewrite it because this block cannot be invoked or called

Practice:

```
SET SERVEROUTPUT ON
DECLARE
  v_firstname VARCHAR(20);
BEGIN
  SELECT first_name INTO v_firstname FROM HR.employees
  WHERE employee_id=100;
  DBMS_OUTPUT.PUT_LINE('The first Name of the Employee is ' || v_firstname);
END;
```

```
SET SERVEROUTPUT ON
DECLARE
  v_fname VARCHAR(20);
  v_lastname VARCHAR(20);
BEGIN
  SELECT first_name, last_name INTO v_fname, v_lastname FROM hr.employees
  WHERE salary = 7400;
  DBMS_OUTPUT.PUT_LINE('The Name of the Employee is ' || v_fname || ', ' ||
v_lastname);
END;
```