

DBMS & SQL

Jagadish N Hegade

LEARNING OBJECTIVES

At the end of this lesson, you will be able to:

- Understand How to Use Coding Platform for SQL
- Understand DBMS
- Understand Types of SQL Commands(DML,DDL,DQL)
- Understand Aggregate Functions
- Understand Basics of Subqueries
- Understand Basics of Joins
- Understand Basics of PL/SQL
- Understand Usage of SQL Developer Tool



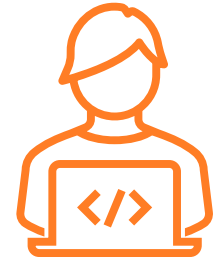
DBMS & RDBMS

DBMS is a software or group of programs designed to perform database operations.

Such as creating a new database, administering and managing the databases.

The DBMS internally might consist of a group of programs each performing different tasks related to database management.

Some DBMS examples, My SQL, Oracle, System 2000, MS Access, My SQL server, etc.



RDBMS

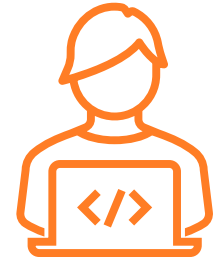
RDBMS is the collection of programs and capabilities that enables the user to interact with a relational database.

A relational database management system (RDBMS) is a type of DBMS with a row-based table structure.

The DBMS internally might consist of a group of programs each performing different tasks related to database management.

Most commercial RDBMS use SQL.

The most basic RDBMS functions are related to create, read, update and delete operations, collectively known as the CRUD cycle.



Features of RDBMS

An RDBMS is easily accessible using SQL commands.

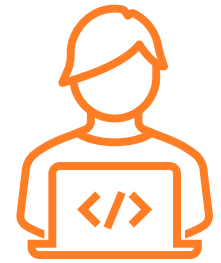
An RDBMS provides full data independence.

The basic unit of data storage in a relational database is called a table.

A table consists of tuples/rows/records and each record has one or more columns used to store values.

In RDBMS, we can use conditional operations such as joins and restrictions.

Also at the same time, you can ensure consistency of data across multiple tables by using integrity constraints.



Advantages of RDBMS

Support for a very large database

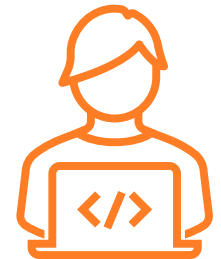
RDBMS has a simple view of the database that conforms to much of the data used in businesses.

RDBMS uses Structured Query Language.

RDBMS defines how the data is organized and how the relations among them are associated.

It defines the entities and relationships among them. It contains a descriptive detail of the database.

RDBMS has data security which is critical when data sharing is based on privacy.

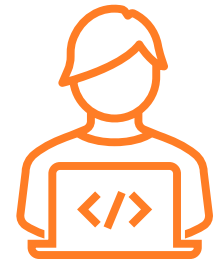


Disadvantages of RDBMS

No support for complex objects such as documents, video, images.

Often poor support for storage of complex objects.

Still no efficient and effective integrated support.



Features of SQL

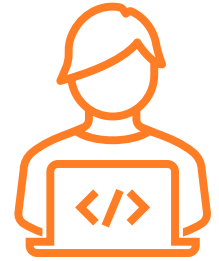
Easy to learn and use

High availability

High performance

Management ease

Comprehensive application development



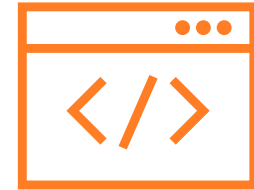
Difference between DBMS and RDBMS

In DBMS concept of a relationship is not there whereas RDBMS is based on the concept of relationship.

DBMS uses the concept of files whereas RDBMS uses the concept of tables.

DBMS hardware and software requirements are very low whereas RDBMS hardware and software requirements are very high.

DBMS speed of operation is slow whereas RDBMS speed of operation is fast



Data Types in Oracle

Numeric Datatypes

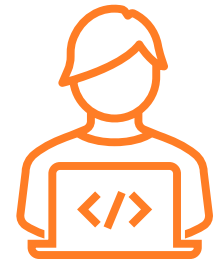
Character Datatypes/ String Datatypes

Long Datatype

Date Datatypes

Raw & Long Raw Datatypes

Lob Datatypes (Large Objects Datatypes)



SQL Commands-DDL

Data Definition Language (DDL) Commands in Oracle are used to define the database objects such as Tables, Views, Stored Procedures, Stored Functions, Triggers, etc

DDL statements in Oracle are used to ALTER or Modify a database or table structure and schema

DDL Commands in Oracle are working on the structure of a database object, not on the data of a table.

Create,Alter,Rename,Truncate,Drop

Latest feature in Oracle(Recyclebin, Flashback, Purge)



SQL Commands-DML

DML stands for Data Manipulation Language, are basically used to INSERT, UPDATE, and DELETE data in a database table

DML statements affect the records in a table.

Basic operations that we perform on data such as inserting new records, deleting unnecessary records, and updating/modifying existing records

Insert, Update, Delete

Latest Added (InsertAll, Merge)



SQL Commands-DQL

The DQL statements in Oracle or you can say the SELECT statements

The Select Statement in Oracle is basically used to return records in the form of a result set from one or more tables or views.

The SQL Select Query does not store any data itself, displays the data that is stored in database tables

```
SELECT * FROM Table_Name;
```

```
SELECT Column_List FROM Table_Name;
```



Constraints

Constraints define specific rules to the column(s) data in a database table.

Constraints in Oracle are defined while creating a new table

We can also alter the table and add new Constraints

Constraints in Oracle are basically used to restrict the insertion of unwanted data in the database

We can create the constraint on single or multiple columns of a table in Oracle



Types of Constraints

NOT NULL Constraint

UNIQUE KEY Constraint

CHECK KEY Constraint

PRIMARY KEY Constraint

FOREIGN KEY Constraint (REFERENCES Key).

DEFAULT Constraint



Functions in Oracle

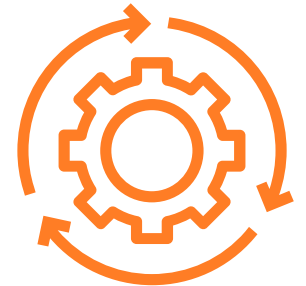
A function in Oracle is a subprogram that is used to perform an action such as complex calculations and returns the result of the action as a value.



Pre-define / Built in functions (use in SQL & PL/SQL)



User define functions (use in PL/SQL)



Single Row Functions



Numeric functions,String functions,Date functions,Conversion functions



Numeric Functions(`ABS()`,`CEIL()` ,`FLOOR()` ,`MOD()` ,`POWER()` ,`ROUND()`,`TRUNC()`)



String
Functions(`LENGTH()`,`LOWER()`,`UPPER()`,`INITCAP()`,`LTIRM()`,`TRIM()`,`LPAD()`,`RPAD()`,`CONCAT()`,`REPLACE()`,`TRANSLATE()`,`SUBSTR()` ,`INSTR()`)



Date Functions(`SYSDATE`,`ADD_MONTHS()` ,`LAST_DAY()` ,`NEXT_DAY()`
,`MONTHS_BETWEEN()`,`CURRENT_DATE()`)



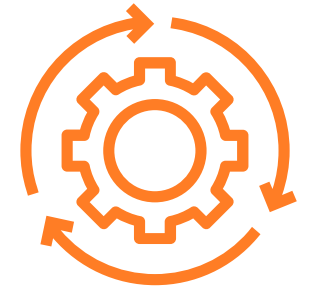
Conversion Functions(`TO_CHAR()` ,`TO_DATE()`)



Multiple Row or Aggregate Functions

The Multiple Row Functions in Oracle are used to return either group of values (or) a single value

These functions are basically operated on a set of rows and return one result or one result per group



SUM(),AVG(),MIN(),MAX(),COUNT()

Clauses in Oracle

Clauses provide the SQL Query with some additional functionalities such as filtering the records, sorting the records, fetching the records, and grouping the records.

Clauses are statements that are used to add to SQL pre-define query for providing additional facilities are like filtering rows, sorting values, grouping similar values

Clauses are used for database access and data extraction by applying a filter that describes the conditions.

Clauses helps us to access proper info and is manageable to deal with different tables in a database.



Types of Clauses

WHERE: Filtering rows (before grouping data)

ORDER BY: Sorting values

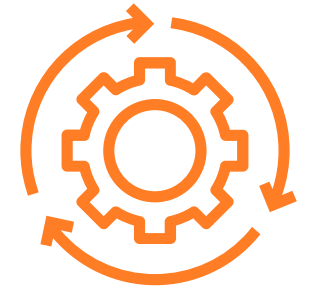
GROUP BY: Grouping similar data

HAVING: Filtering rows (after grouping data)

ROLLUP: Finding subtotal & grand total (single column)

CUBE: Finding subtotal & grand total (multiple columns)

FETCH FIRST: To Return top n Number of Records



Types of Operators

Arithmetic operators:(+, -, *, /, %)

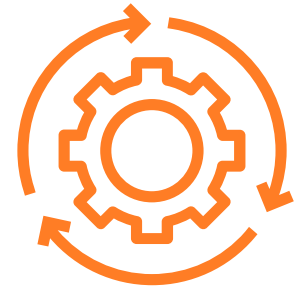
Assignment operator:(=)

Relational operators:(=, != or <>, >, <, >=, <=)

Logical operators:AND ,OR,NOT

Special Operators:IN,LIKE,BETWEEN

Set Operators:(Union,Union All,Intersect,Minus)



Subqueries

Subquery is a query within a query. We can create subqueries within our SQL statements.

Subqueries can reside in the WHERE clause, in the FROM clause, or in the SELECT clause.

Inner / Child / Sub query

Outer / Parent / Main query



Types of Subqueries

Non-Correlated Subqueries

In non-correlated subqueries, first, the inner query will execute and return value, and later outer query will execute.

Correlated Subqueries

In Correlated subqueries first outer query will execute and return value and later inner query will execute.



Non-Correlated Subqueries

Single Row Subquery:

Returns a single value is called a Single Row Subquery. In Single Row Subquery we can use operators such as = , < , > , <= , >=, !=.

Multiple Row Subquery:

Returns more than one value is called a Multiple Row Subquery in Oracle. In this Multiple Row Subquery, we can use the operators such as IN, ANY, ALL.

Multiple Column Subquery:

Multiple columns values of the inner query compared with multiple columns values of the outer query

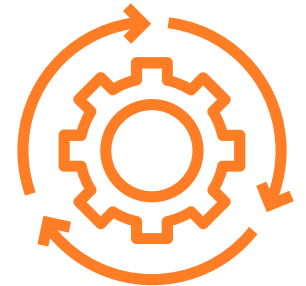


Advantages of Subqueries

Provide an alternative way to query data that would require complex joins and unions.

Make the complex queries more readable.

Allow a complex query to be structured in a way that it is possible to isolate each part.

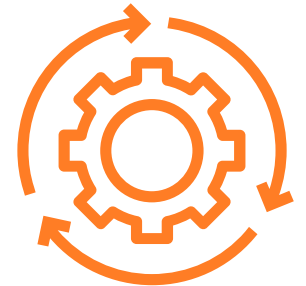


Advantages of Subqueries

Provide an alternative way to query data that would require complex joins and unions.

Make the complex queries more readable.

Allow a complex query to be structured in a way that it is possible to isolate each part.



Joins

Joins in Oracle are used to retrieve data from multiple tables at a time..

In relational databases, we are storing related data in multiple tables like employee details, department details, customer details, orders details, products details, etc

To combine data and retrieve data from those multiple tables then we need joins.



Types of Joins

NON – ANSI format joins: (oracle 8i joins)

When we are retrieving data from multiple tables based on the “WHERE” clause condition then we called a NON-ANSI format join. NON-ANSI joins are not portable.

NON – ANSI format joins(Equi Join,Non-Equi Join,Self Join)

ANSI format joins: (oracle 9i joins)

When we are retrieving data from multiple tables with “on” / “using” clause conditions then we called as the join as ANSI format join. ANSI joins are portability (move from one database to another database)

ANSI format joins(Inner Join,Outer Join(Left,Right,Full),Cross Join,Natural Join)



PL/SQL Basics

PL/SQL stands for “Procedural Language extensions to the Structured Query Language”.

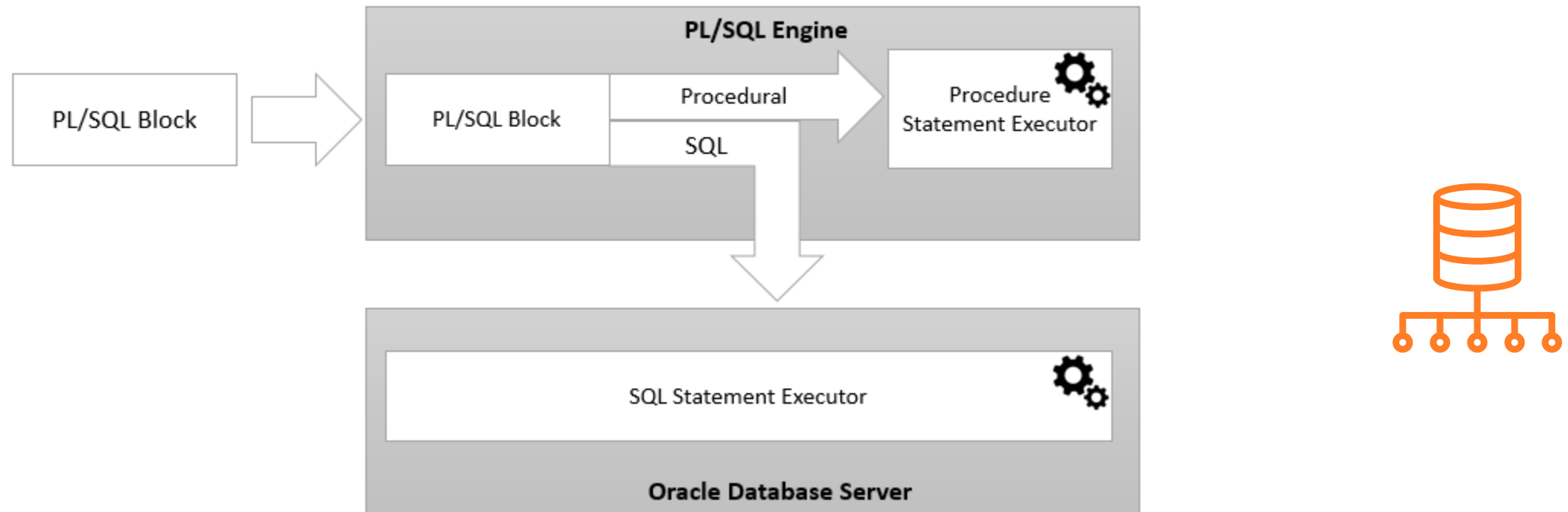
PL/SQL adds many procedural constructs to SQL language to overcome some limitations of SQL, provides a more comprehensive programming language solution for building mission-critical applications on Oracle Databases.

PL/SQL is a highly structured and readable language. PL/SQL is a standard and portable language for Oracle Database development.

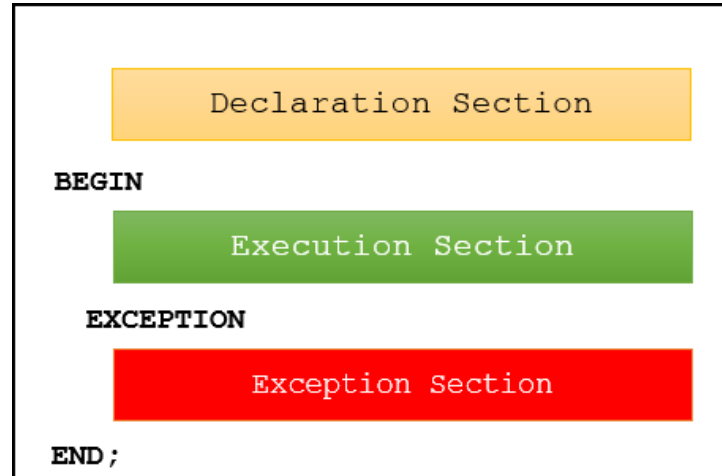
PL/SQL is an embedded language. PL/SQL only can execute in an Oracle Database. It was not designed to use as a standalone language like Java, C#, and C++.



PL/SQL Architecture



PL/SQL Block



Declaration section

A PL/SQL block has a declaration section where you declare variables, allocate memory for cursors, and define data types.

Executable section

A PL/SQL block has an executable section. An executable section starts with the keyword BEGIN and ends with the keyword END. The executable section must have a least one executable statement, even if it is the NULL statement which does nothing.

Exception-handling section

Exception-handling section

A PL/SQL block has an exception-handling section that starts with the keyword EXCEPTION. The exception-handling section is where you catch and handle exceptions raised by the code in the execution section.



How to execute a PL/SQL block using SQL*Plus:

```
SQL> SET SERVEROUTPUT ON;
SQL> BEGIN
  2      DBMS_OUTPUT.PUT_LINE('Hello World');
  3  END;
  4  /
Hello World

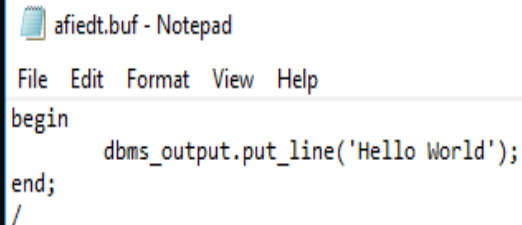
PL/SQL procedure successfully completed.
```

Note :

You must execute SET SERVEROUTPUT ON command in every session that you connect to the Oracle Database in order to show the message using the DBMS_OUTPUT.PUT_LINE procedure.

To edit the code block, use the edit command. SQL*Plus will write the code block to a file and open it in a text editor as shown in the following picture:

```
SQL> edit
Wrote file afiedt.buf
```

A screenshot of a Notepad window titled 'afiedt.buf - Notepad'. The window contains the following text: 'begin', 'dbms_output.put_line('Hello World');', 'end;', and '/'. The text is in a monospaced font, typical of a code editor.

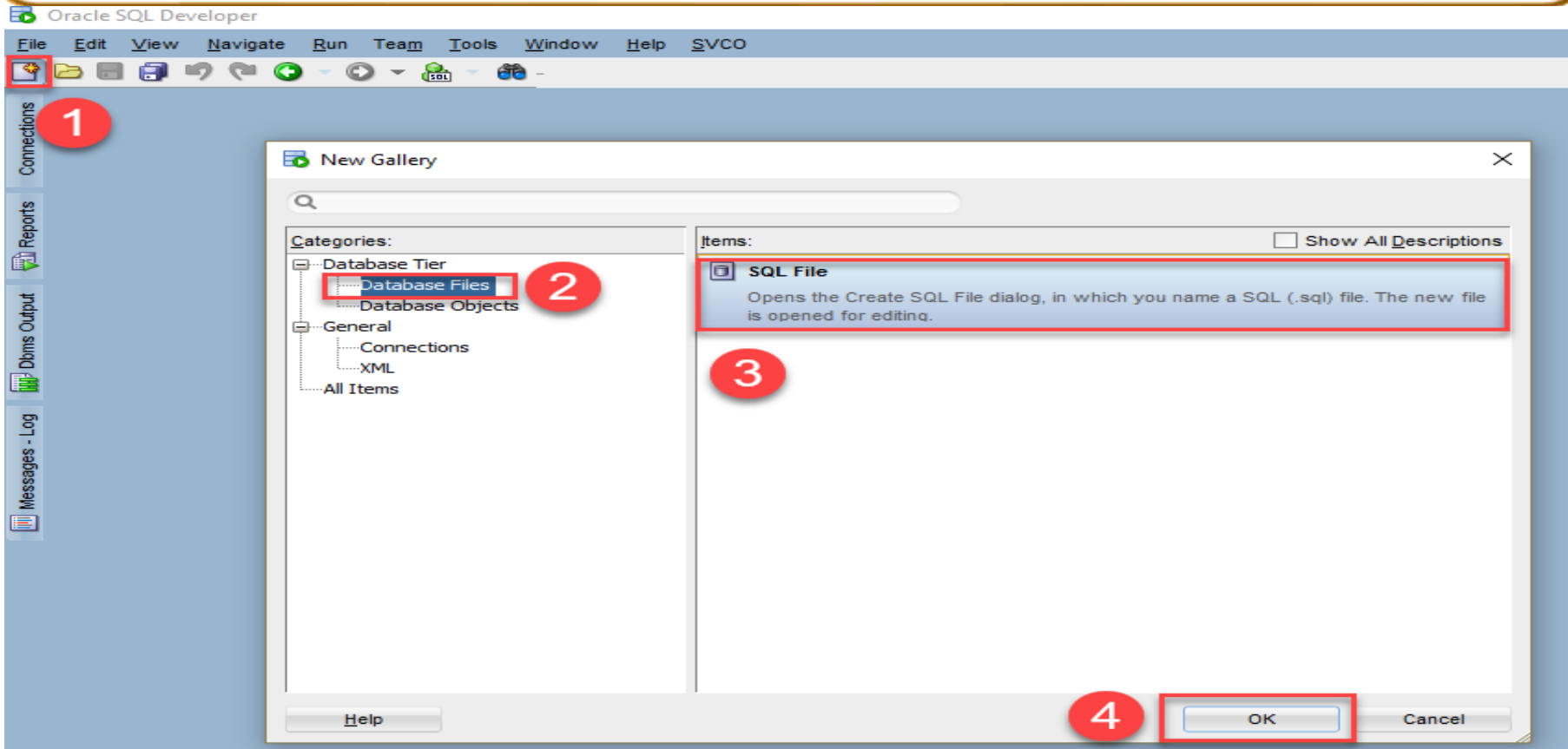
```
afiedt.buf - Notepad
File Edit Format View Help
begin
dbms_output.put_line('Hello World');
end;
/
```



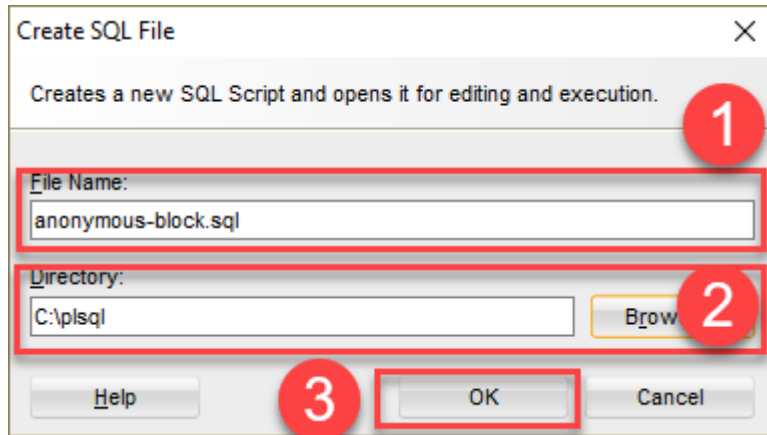
How to execute a PL/SQL block using SQL Developer

First, connect to the Oracle Database server using Oracle SQL Developer

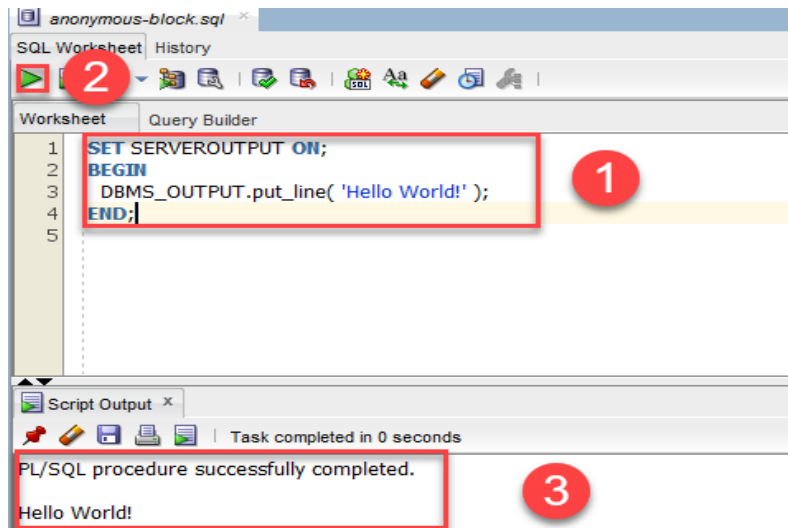
Second, create a new SQL file named anonymous-block.sql resided in the C:\plsqli directory that will store the PL/SQL code..



How to execute a PL/SQL block using SQL Developer



Third, enter the PL/SQL code and execute it by clicking the **Execute** button or pressing the **Ctrl-Enter** keyboard shortcut.



Anonymous block Example

```
DECLARE  
  l_message VARCHAR2( 255 ) := 'Hello World!';  
BEGIN  
  DBMS_OUTPUT.PUT_LINE(l_message );  
END;
```

```
DECLARE  
  v_result NUMBER;  
BEGIN  
  v_result := 1 / 0;  
EXCEPTION  
  WHEN ZERO_DIVIDE THEN  
    DBMS_OUTPUT.PUT_LINE( SQLERRM );  
END;
```

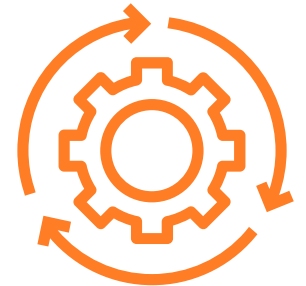
With Exception Handling



PL/SQL Datatypes

Scalar Data Types: Number, Boolean, Character, Datetime

A scalar data type may have subtypes. A subtype is a data type that is a subset of another data type, which is its base type. A subtype further defines a base type by restricting the value or size of the base data type.



PL/SQL Variables

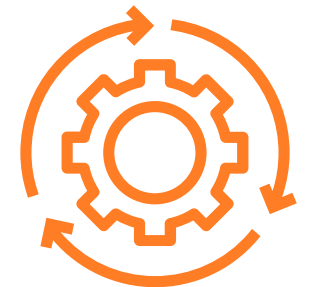
In PL/SQL, a variable is named storage location that stores a value of a particular data type. The value of the variable changes through the program. Before using a variable, you must declare it in the declaration section of a block.

```
variable_name datatype [NOT NULL] [:= initial_value];
```

First, specify the name of the variable. The name of the variable should be as descriptive as possible, e.g., l_total_sales, l_credit_limit, and l_sales_revenue.

Second, choose an appropriate data type for the variable, depending on the kind of value which you want to store, for example, number, character, Boolean, and datetime.

By convention, local variable names should start with l_ and global variable names should have a prefix of g_.

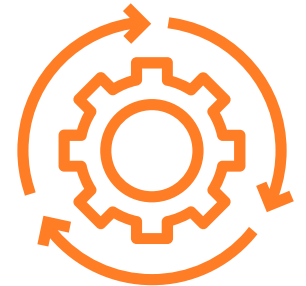


PL/SQL with Default Values

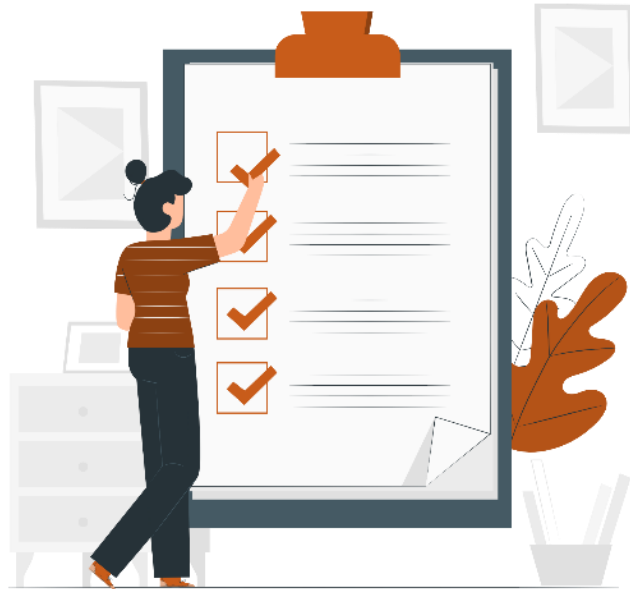
```
DECLARE  
  l_product_name VARCHAR2(100) DEFAULT  
  'Laptop';  
BEGIN  
  NULL;  
END;
```

OR

```
DECLARE  
  l_product_name VARCHAR2( 100 ) := 'Laptop';  
BEGIN  
  NULL;  
END;
```



SUMMARY



In this lesson, you've learned to:

- Differentiate between
 - DBMS vs RDBMS
 - SQL COMMANDS
 - Constraints, Data Types, Operators, Clauses
 - Subqueries vs Joins
 - Brief Understanding on PL/SQL

Thank You

No.308, 3rd Floor, 100ft Main Road, Indiranagar, Bangalore - 560038



+91 90192 17000



info@jigsawacademy.com