Oracle PL/SQL

Introduction to PL/SQL

Agenda

01 DBMS Fundamentals

02 RDBMS

03 Oracle Database

04 Oracle PL/SQL Installation

05 Oracle PL/SQL

06 Oracle PL/SQL Architecture

07 PL/SQL Block

08 PL/SQL Programming Environments

09 PL/SQL Variables

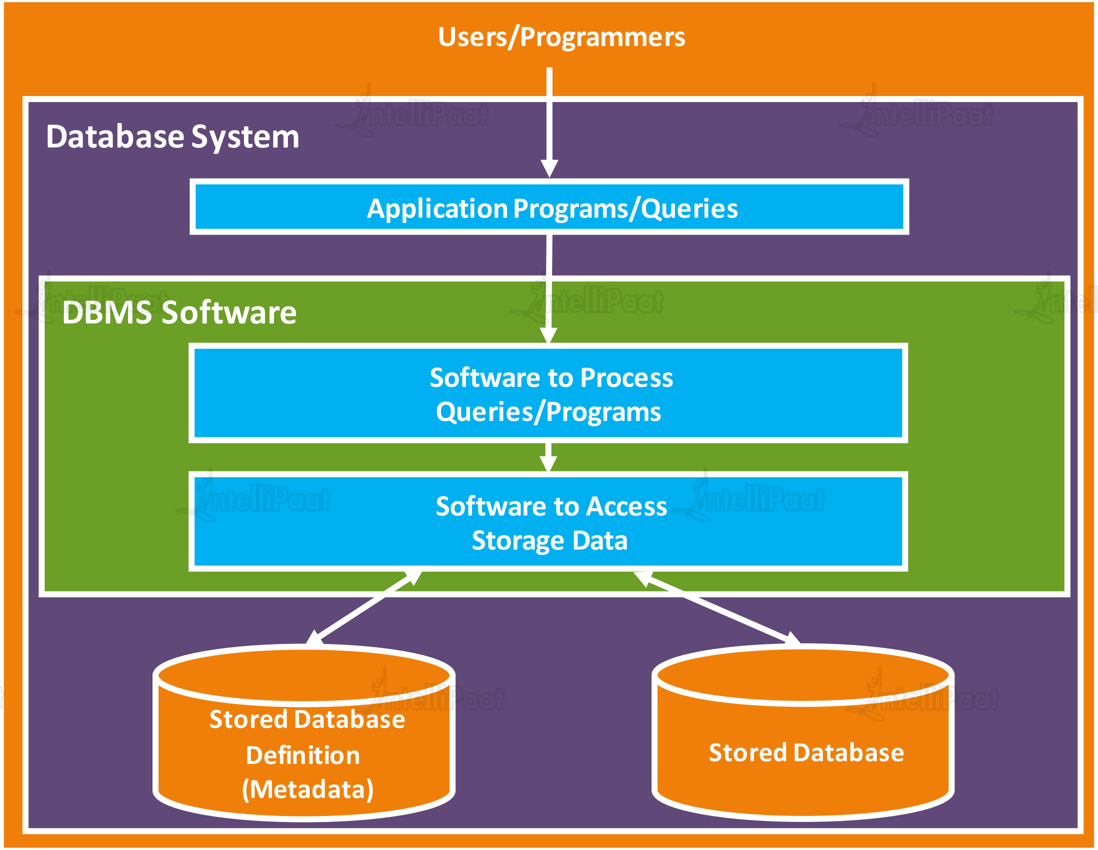
10 Basic PL/SQL Constructs

DBMS Fundamentals

Database Terminology

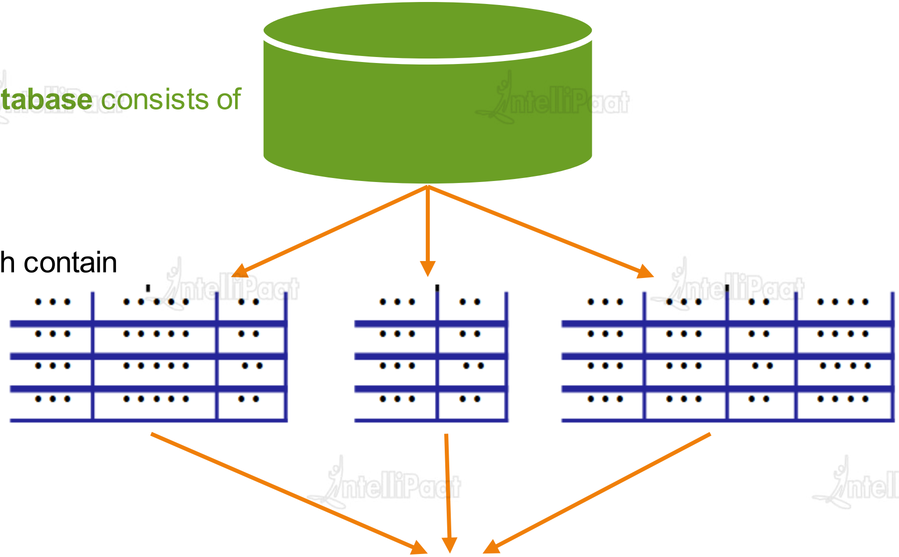
•Data	• Known facts that can be recorded and have implicit meaning
Database	• A collection of related data
Database System	• Composed of five major parts: Hardware, Software (DBMS), People, Procedures, and Data
Database Management System (DBMS)	• Collection of components that support data acquisition, dissemination, storage, maintenance, retrieval, and formatting

Database System Environment


Users/Programmers
Database System
Application Programs/Queries
DBMS Software
Software to Process Queries/Programs
Software to Access Storage Data
Stored Database Definition (Metadata)
Stored Database

RDBMS

Database Components



A **database** consists of

Files, which contain

Records that contain

Fields, which may consist of a variety of data types

Hernandez, Juan	123456789	72	2.42
Jones, Mary	234567890	102	3.87

Notice that there should always be a **Key** (Unique) Field

What Is RDBMS?

Relational Database Management System

It is a database management system based on a relational model, which is used to manage the relational database.

Relational model is the organization of data in tables which are interrelated. The relational database is an organized collection of tables.

Data is stored in tables. Tables are related to each other using one or more fields.

Oracle Database

What Is Oracle?

- A management system which uses the relational data model
- In the relational data model, data is seen by users in the form of ‘tables’ alone
- Produced and marketed by Oracle Corporation
- Also known as Oracle database, Oracle DB, or simply Oracle
- Oracle DB runs on most major platforms like Windows, UNIX, Linux, and Mac OS
- First database designed for Enterprise Grid Computing

ORACLE®

DATABASE

Oracle Versions

Year	Version	Features
1979	Version 1	Not commercially released
1980	Version 2	First commercial SQL database
1982	Version 3	First portable database and the first RDBMS to support SMP
1984	Version 4	Introduced read consistency
1986	Version 5	Supports client–server architecture and row-level locking
1988	Version 6	Financial applications built on Oracle
1992	Version 7	Varchar2 data type, stored procedures, functions, and triggers
1997	Version 8	First web database, object-oriented features, and table partitioning
1998	Version 8i	Java support, SQLJ, XML, and Oracle intermedia
2001	Version 9i	RAC, OLAP services, and native XML database
2004	Version 10g	Flashback query, data pump, automatic storage management, backup compression, and regular expressions
2007	Version 11g	Read-only table, virtual cols, pivot operator, follows clause, and compound triggers
2013	Version 12c	Identity columns, top *N* rows, multi-tenant architecture, in-memory database, etc.

ORACLE

DATABASE

Editions of Oracle Database

ORACLE
DATABASE

Enterprise Edition

- Most robust
- Superior performance and secure
- Offers all features

Standard Edition

- Provides base functionality for users

Express Edition (XE)

- Lightweight
- Free
- Limited edition

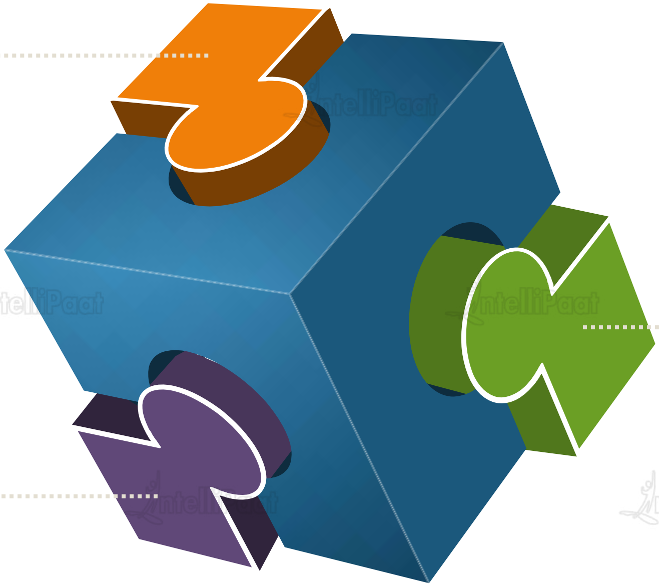
Oracle Lite

- Designed for mobile devices

Oracle Products

ORACLE®

DATABASE



Server

Connectivity Packages

Application Development Tools

Application Development Tools

SQL Developer (includes forms, reports, graphics, etc.)

Oracle Designer

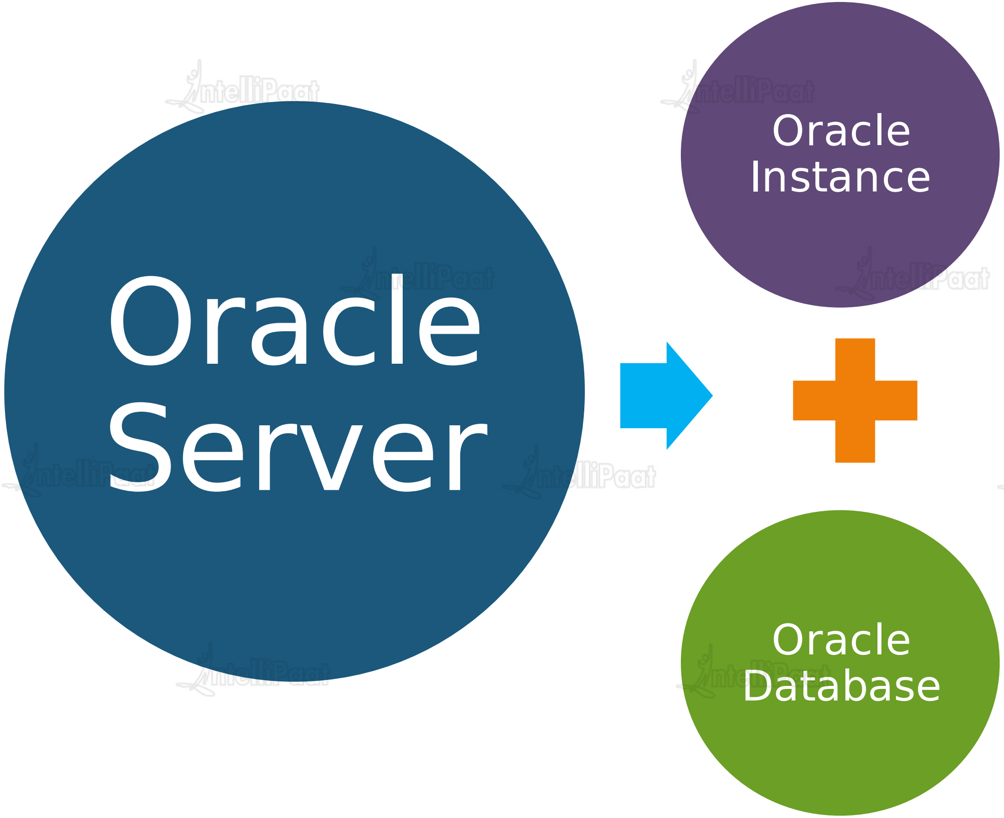
Oracle JDeveloper

SQL*Plus

Oracle Precompilers

PL/SQL

Oracle Server



Oracle Server

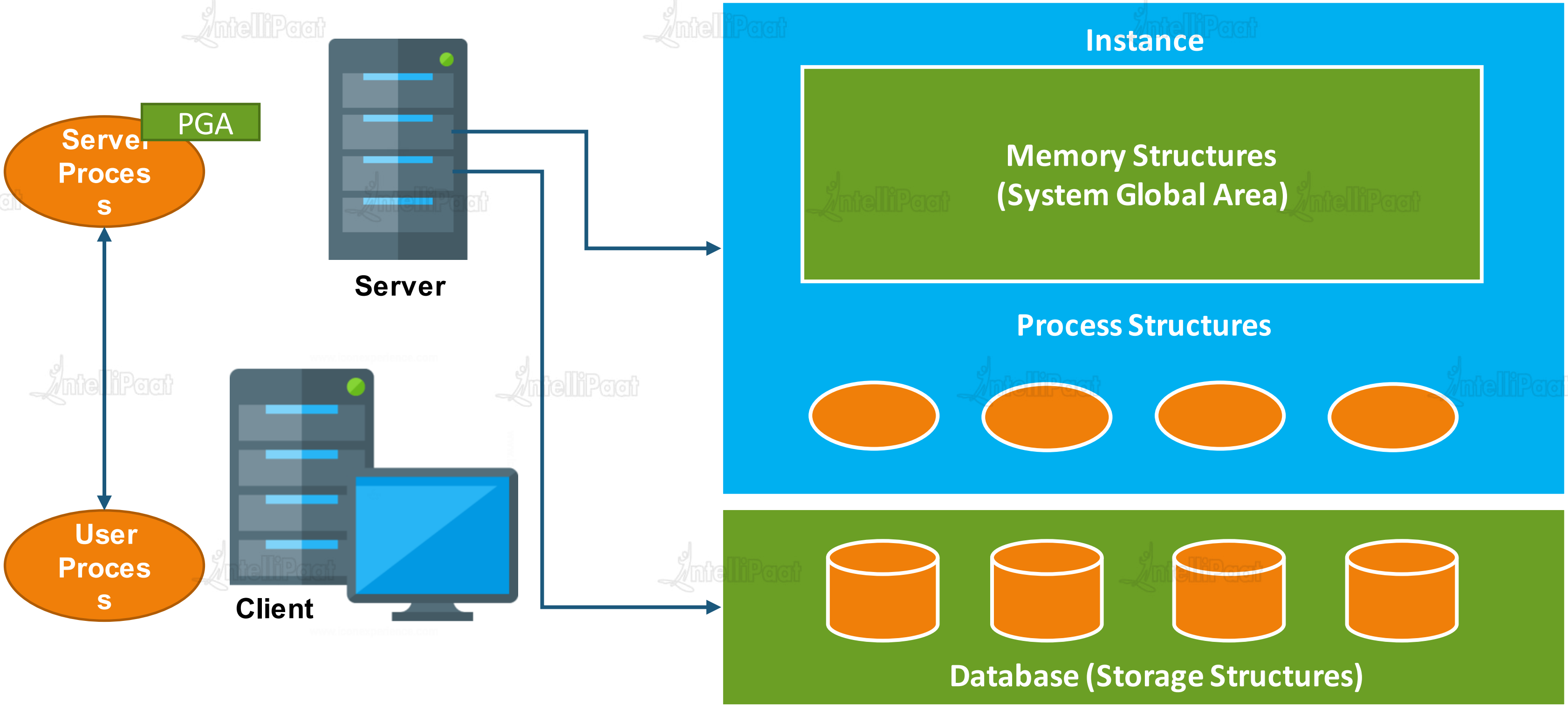
Oracle Instance

Oracle Database

- A means to access an Oracle database
- Always opens one and only one database
- Consists of memory and background process structures

- A collection of data that is treated as a unit
- Consists of three file types

Oracle Database Server Architecture



Oracle 12c Features

Multitenant Architecture

In-memory Database

Invisible Columns

- Mark one or more specific columns as invisible
- Columns will not be seen by the application; however, they can be explicitly referenced
- Make changes without affecting the application

Improved Defaults and Identity Columns

- Default values when using a sequence
- Default values when NULL is inserted

Oracle PL/SQL Installation

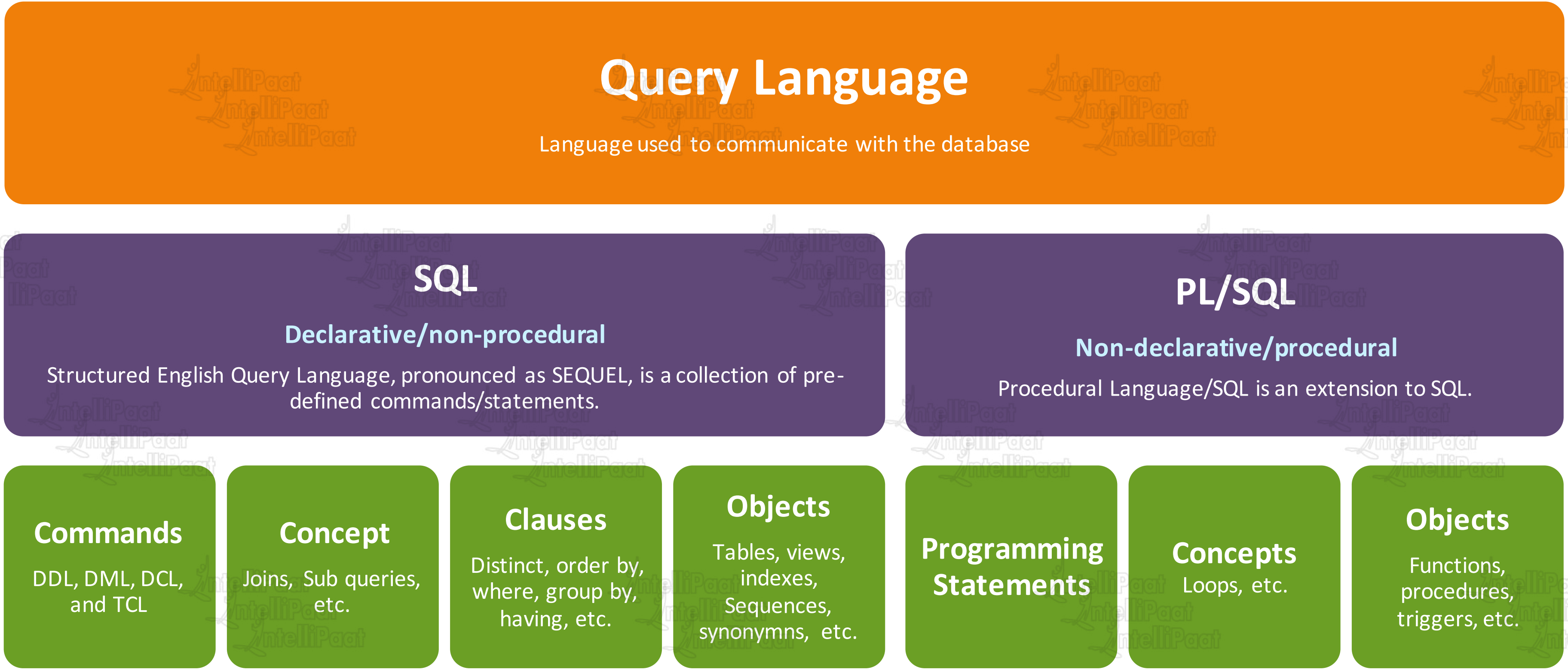
Getting the Oracle DB Software

Pre-installation

- Download Oracle 12c
- It comes in two editions
    - Enterprise edition
        - Most robust and secure edition
        - Offers all features, including superior performance and security
- Standard edition
        - Provides the base functionality for users that do not require Enterprise edition's robust package
- Download link: http://www.oracle.com/technetwork/database/enterprise-edition/downloads/index.html
- Create the installation folder
- Extract download to the installation folder
- Verify the contents of the installation folder

Role of SQL in RDBMS

Query Languages Used with Oracle



Query Language

Language used to communicate with the database

SQL

Declarative/non-procedural

Structured English Query Language, pronounced as SEQUEL, is a collection of pre-defined commands/statements.

PL/SQL

Non-declarative/procedural

Procedural Language/SQL is an extension to SQL.

Commands

DDL, DML, DCL, and TCL

Concept

Joins, Sub queries, etc.

Clauses

Distinct, order by, where, group by, having, etc.

Objects

Tables, views, indexes, Sequences, synonymns, etc.

Programming Statements

Concepts

Loops, etc.

Objects

Functions, procedures, triggers, etc.

What Is SQL?

- A collection of pre-defined statements/commands
- An interface between the user/client and the database
- Using this, all programs and users can access data in an Oracle database
- A client tool to interact with the Oracle database or any other databases
- Developed by IBM corporation, Inc., in the year 1972
- A standard database language for most of the RDBMSs like MS SQL Server, MySQL, DB2, etc.

SQL provides statements for a variety of tasks, including:

Querying data

Inserting, updating, and deleting rows in a table

Creating, replacing, altering, and dropping objects

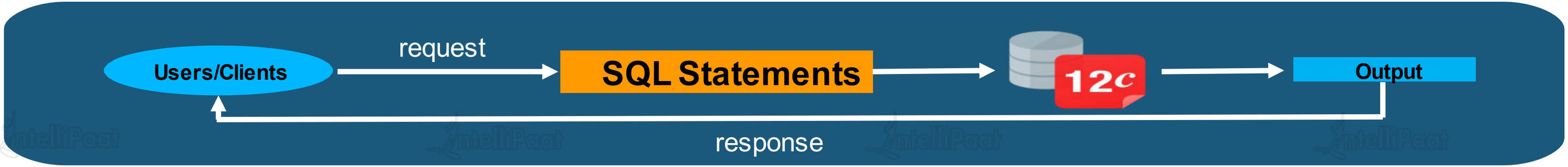
Controlling access to the database and its objects

Guaranteeing database consistency and integrity

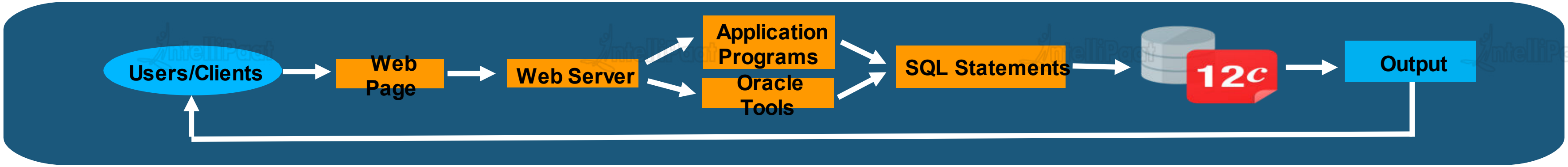
Note: SQL unifies all of the preceding tasks in one consistent language.

Role of SQL in RDBMS

- The purpose of SQL is to provide an interface to a relational database such as Oracle Database; all SQL statements are instructions to the database
- SQL is a data sublanguage
- SQL can be helpful for two kinds of users
- Technical users, who knows Oracle, such as developers and DBAs, can directly interact with the database



- Non-technical users, who don't know Oracle, can interact indirectly with database



Role of SQL in RDBMS

It is a command-based language, easy to learn and use

It is an efficient query language

Every command starts with a 'verb'

At a time, only one query is allowed to be executed. Every command should end with ';'

Both ANSI and ISO have accepted SQL as the standard language for relational databases

All programs written in SQL are portable

They can often be moved from one database to another with very little modification

Writing SQL Statements

SQL statements are not case sensitive

SQL statements can be entered on one or more lines

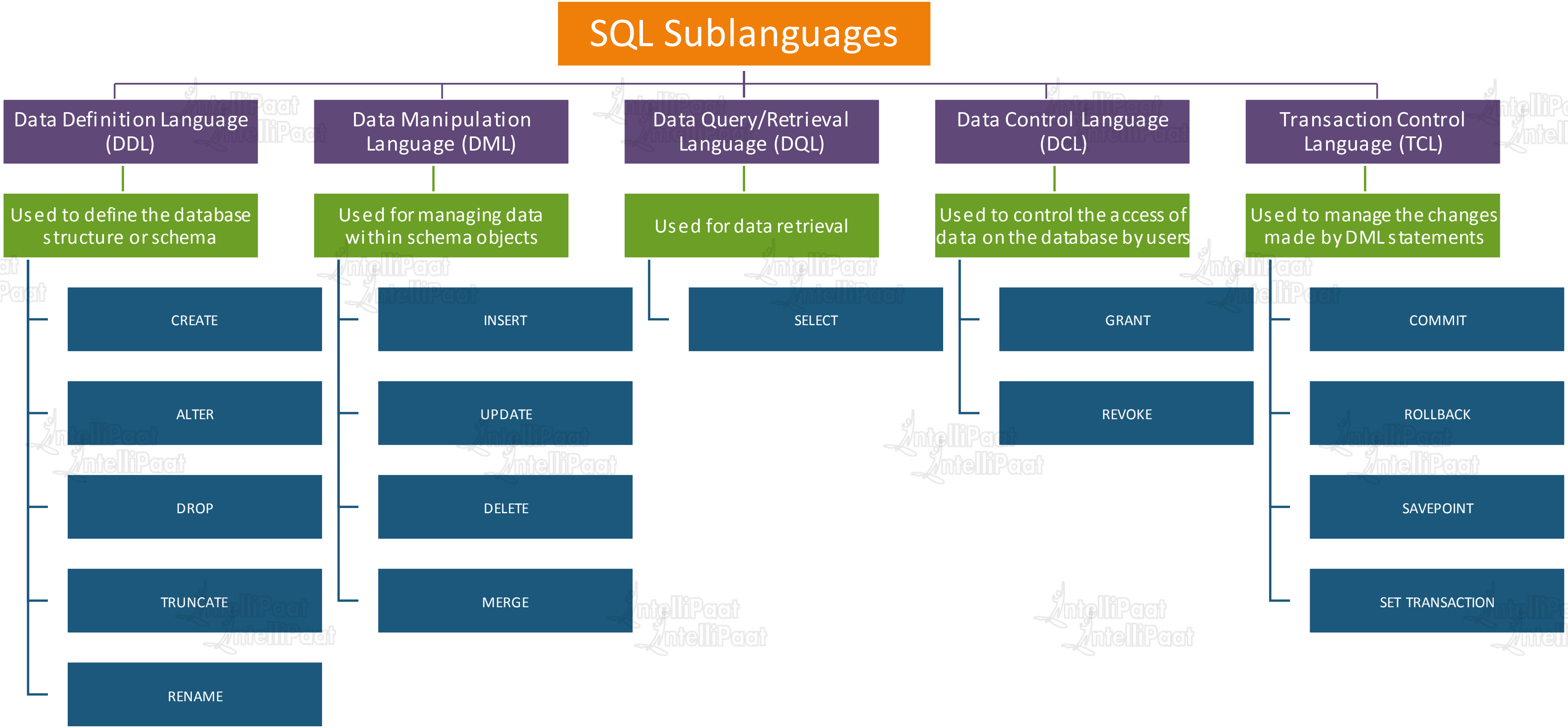
Keywords cannot be split across lines

Clauses are usually placed on separate lines for readability and ease of editing

Indents should be used to make the code more readable

Keywords are used in uppercase; all other words, like table names and column names will be in lowercase

SQL Statements



SQL Sublanguages

- **Data Definition Language (DDL)**
  - Used to define the database structure or schema
    - CREATE
    - ALTER
    - DROP
    - TRUNCATE
    - RENAME
- **Data Manipulation Language (DML)**
  - Used for managing data within schema objects
    - INSERT
    - UPDATE
    - DELETE
    - MERGE
- **Data Query/Retrieval Language (DQL)**
  - Used for data retrieval
    - SELECT
- **Data Control Language (DCL)**
  - Used to control the access of data on the database by users
    - GRANT
    - REVOKE
- **Transaction Control Language (TCL)**
  - Used to manage the changes made by DML statements
    - COMMIT
    - ROLLBACK
    - SAVEPOINT
    - SET TRANSACTION

SQL Tools

Oracle SQL Developer

- A graphical tool that lets you browse, create, edit, and delete (drop) database objects, edit and debug PL/SQL code, run SQL statements and scripts, etc.

SQL*Plus

- An interactive and batch query tool
- Web-based user interface called iSQL*Plus
- The only command-line tool in 11g
- Installed with every oracle database server or client installation

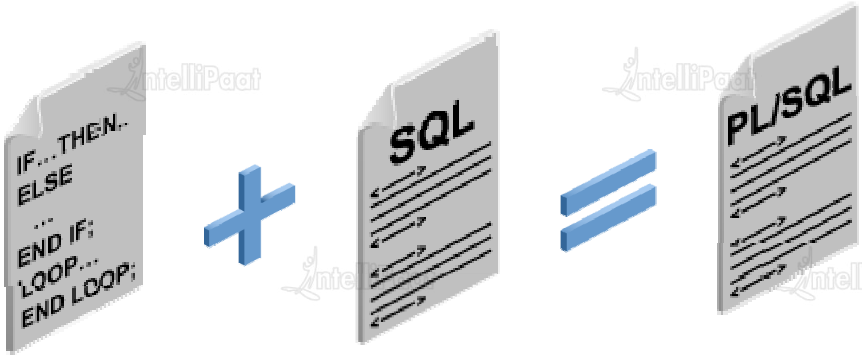
Oracle PL/SQL

PL/SQL: Introduction

Stands for Procedural Language extension to SQL

Oracle Corporation’s standard data access language for relational databases

Seamlessly integrates procedural constructs with SQL



PL/SQL: Introduction

A completely portable, high-performance transaction-processing language

Provides a built-in, interpreted, and OS-independent programming environment

Can also directly be called from the command-line SQL*Plus interface

A direct call can also be made from external programming language calls to database

Its general syntax is based on that of ADA and Pascal programming languages

Apart from Oracle, PL/SQL is available in TimesTen in-memory database and IBM DB2

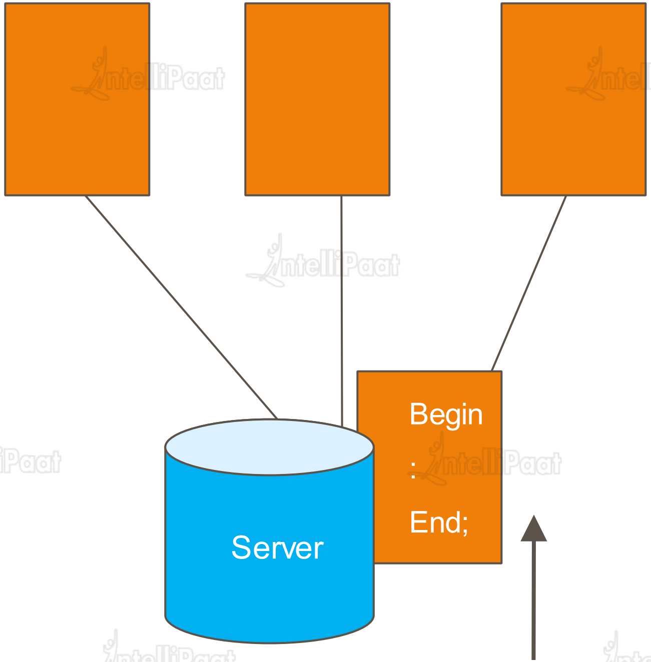
Features of PL/SQL

- ✓ Tightly integrated with SQL
- ✓ Offers extensive error checking
- ✓ Offers numerous data types
- ✓ Offers a variety of programming structures
- ✓ Supports structured programming through functions and procedures
- ✓ Supports object-oriented programming
- ✓ Supports the development of web applications and server pages

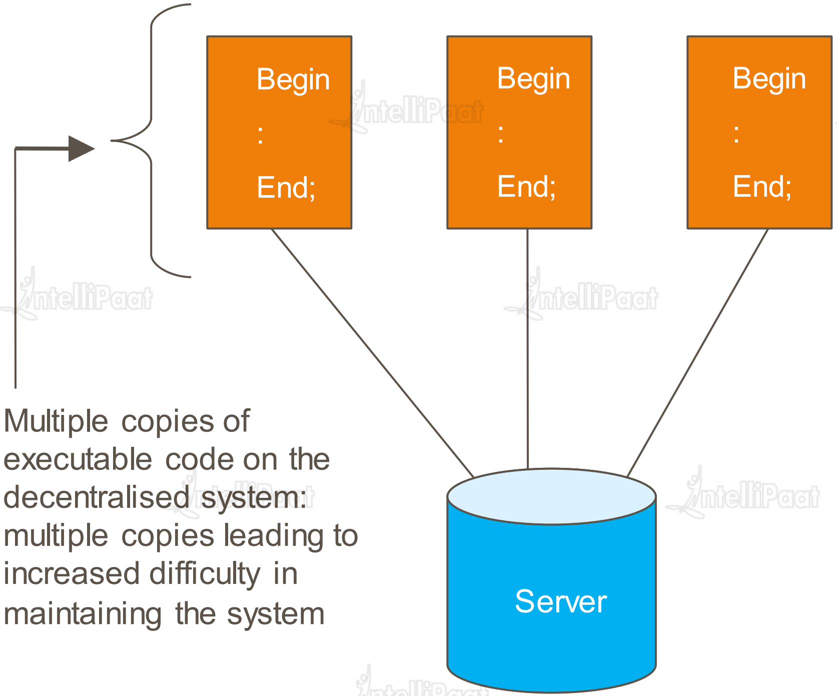
Why Use PL/SQL?

- To manage business rules, through a middle layer application logic
- To generate code for triggers
- To generate code for the user interface
- To enable database-centric client/server applications

Centralised Vs. Decentralised



Common copy of the executed code: it maintains only one common copy of executable code

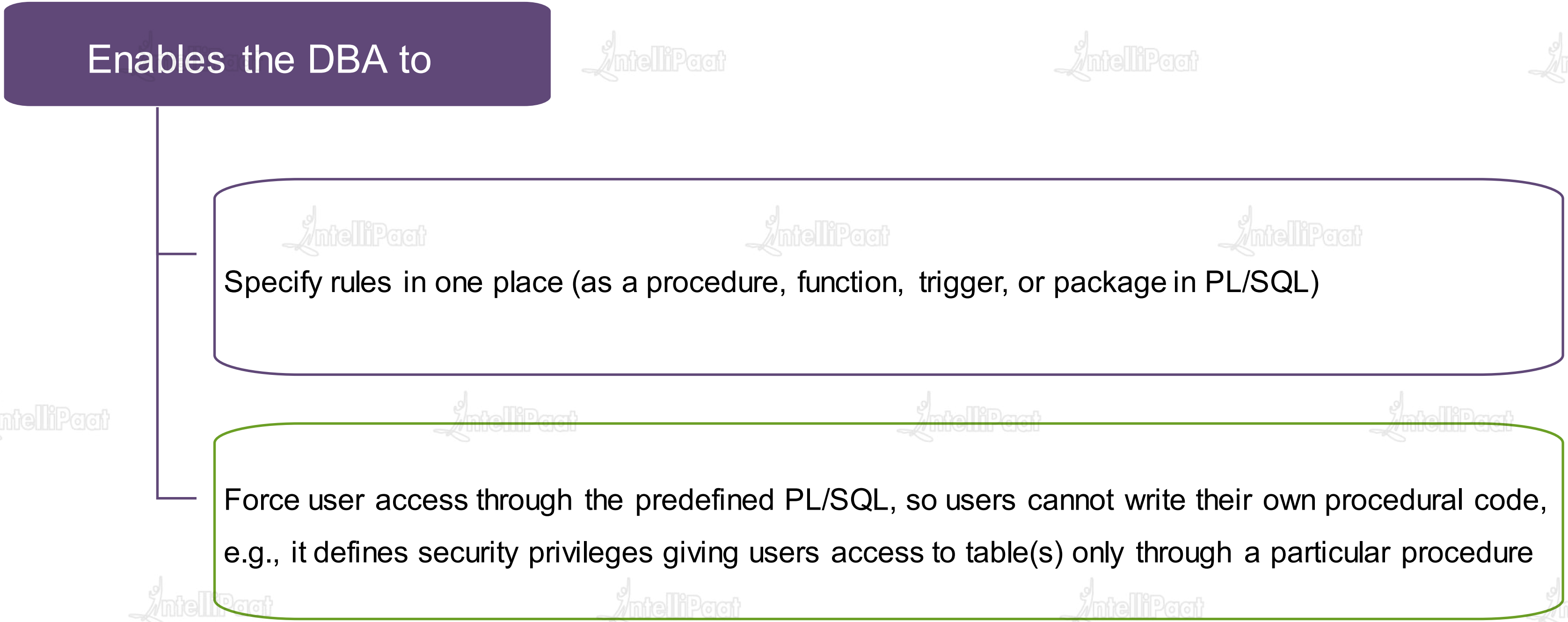


Multiple copies of executable code on the decentralised system: multiple copies leading to increased difficulty in maintaining the system

Advantages of Using PL/SQL to Access Oracle

- ✓ PL/SQL is managed centrally within the database
- ✓ Code is managed by the DBA, and execution privileges are managed in the same way as with other objects
- ✓ PL/SQL objects are first-class Oracle DB objects
- ✓ It is easy to read with modularity features and error handling

Centralised Control



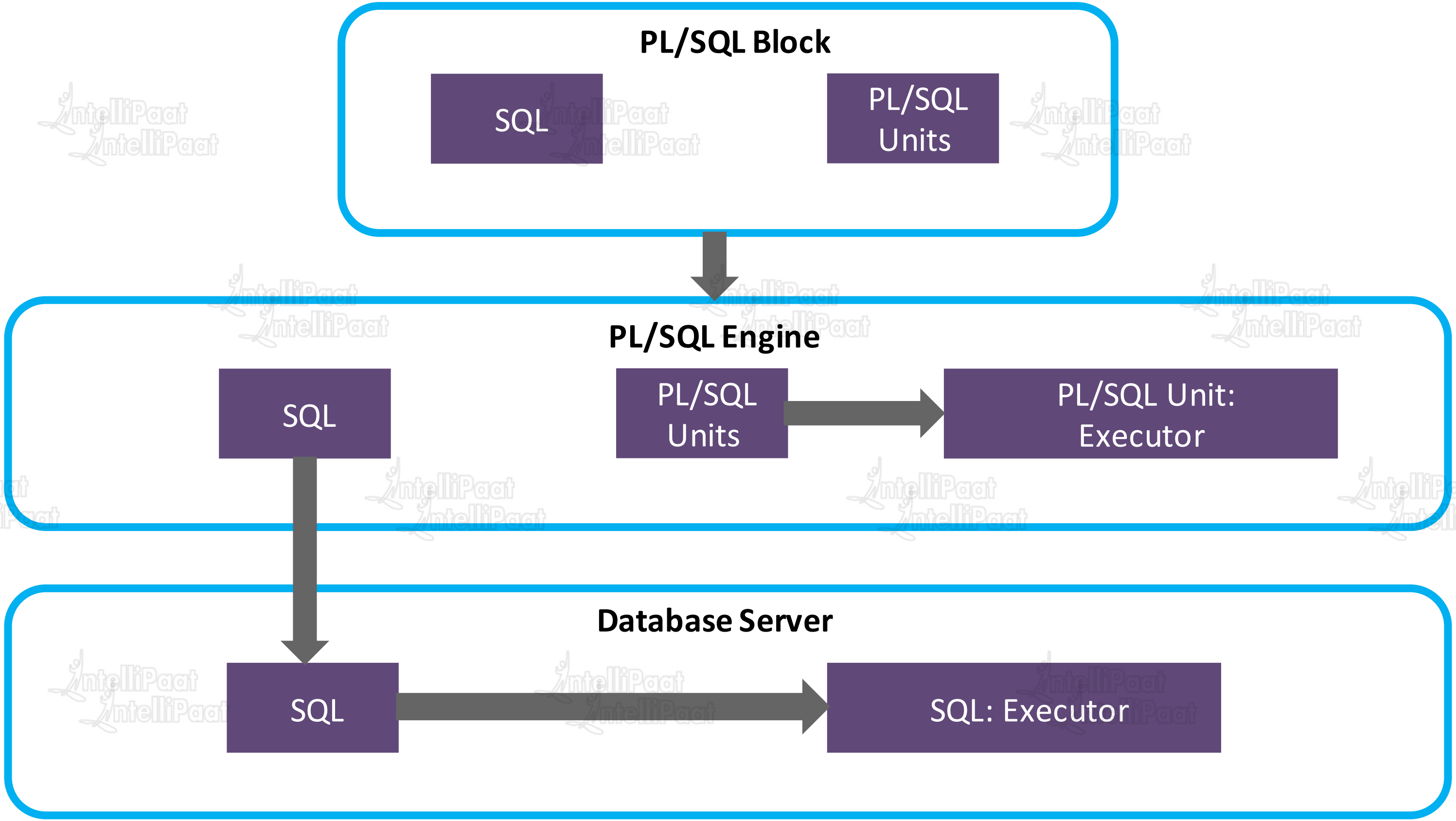
Enables the DBA to

Specify rules in one place (as a procedure, function, trigger, or package in PL/SQL)

Force user access through the predefined PL/SQL, so users cannot write their own procedural code, e.g., it defines security privileges giving users access to table(s) only through a particular procedure

Oracle PL/SQL Architecture

Oracle PL/SQL Architecture



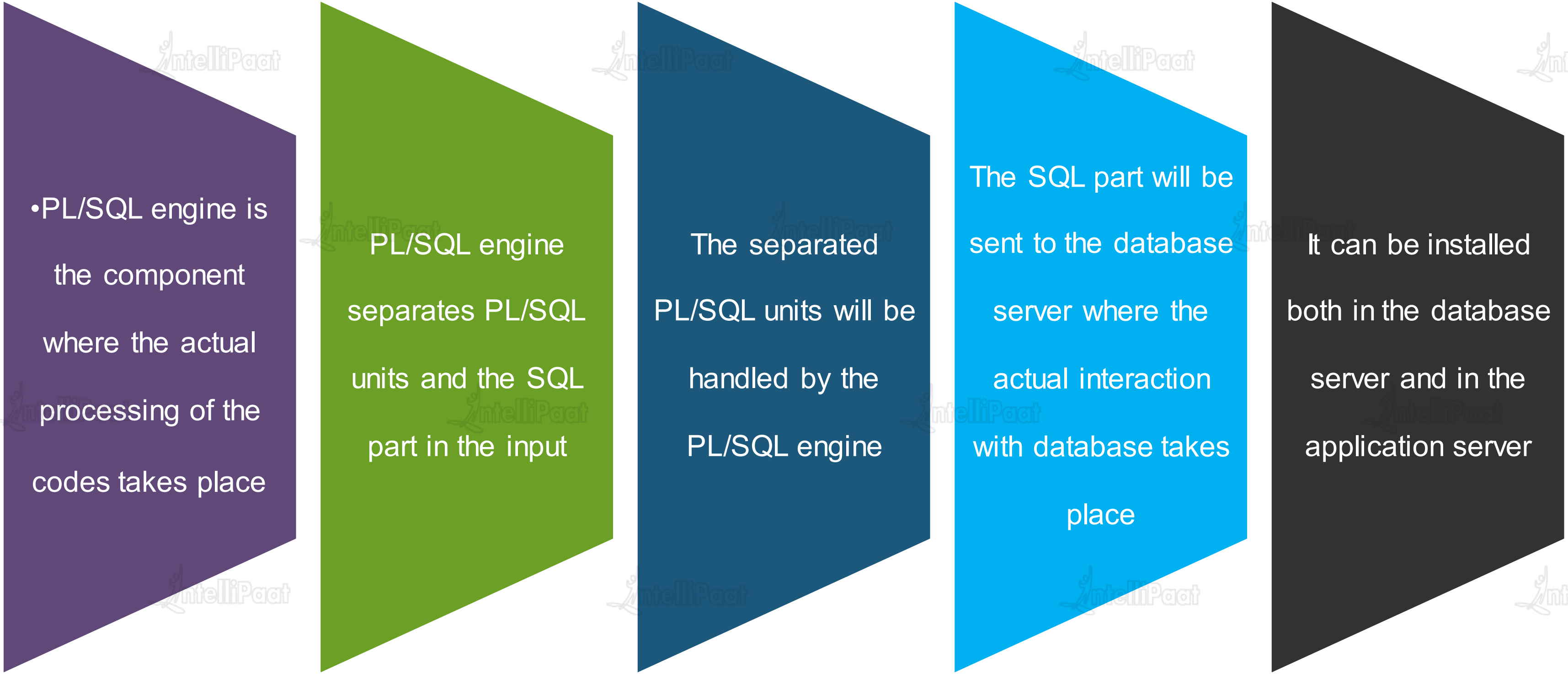
PL/SQL Block

- This is the component which has the actual PL/SQL code
- This consists of different sections to divide the code logically
- It also contains the SQL instruction that is used to interact with the database server
- All PL/SQL units are treated as PL/SQL blocks, and this is the starting stage of the architecture which serves as the primary input

Following are the different types of PL/SQL units:

- •Anonymous Block
- Function
- Library
- Procedure
- Package Body
- Package Specification
- Trigger
- Type
- Type Body

PL/SQL Engine



- PL/SQL engine is the component where the actual processing of the codes takes place

PL/SQL engine separates PL/SQL units and the SQL part in the input

The separated PL/SQL units will be handled by the PL/SQL engine

The SQL part will be sent to the database server where the actual interaction with database takes place

It can be installed both in the database server and in the application server

Database Server

•This is the most important component of Pl/SQL unit which stores data

The PL/SQL engine uses the SQL from PL/SQL units to interact with the database server

It consists of the SQL executor which parses the input SQL statements and executes the same

PL/SQL Block

PL/SQL Block Structure

DECLARE (optional)

Variables, cursors, and user-defined exceptions

BEGIN (mandatory)

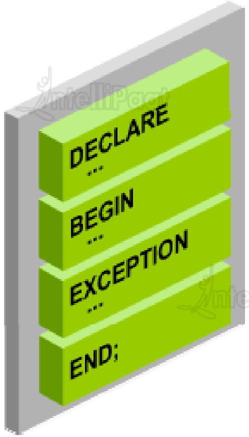
SQL statements

PL/SQL statements

EXCEPTION (optional)

Actions to perform when errors occur

END; (mandatory)



Block Types

Anonymous

```
[DECLARE]


BEGIN
   --statements
[EXCEPTION]
END;
```

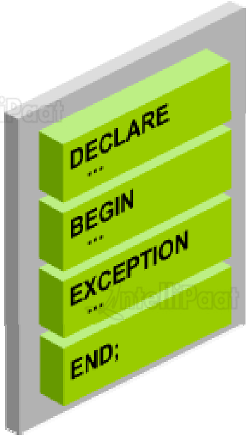
Procedure

```
PROCEDURE
name  IS

BEGIN
   --statements
[EXCEPTION]
END;
```

Function

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

END;
```

Program Constructs



Tools Constructs
Anonymous blocks
Application procedures or functions
Application packages
Application triggers
Object types

Database Server Constructs
Anonymous blocks
Stored procedures or functions
Stored packages
Database triggers
Object types

PL/SQL Programming Environments

PL/SQL Variables

Variables

Use of Variables

- Temporary storage of data
- Manipulation of stored values
- Reusability

Handling Variables

- Declaring and initializing variables in the declaration section
- Assigning new values to variables in the executable section
- Viewing results through output variables

Types of Variables

PL/SQL

- Scalar (holds a single value) - number, date, binary_integer, Boolean (true, false, null), timestamp, etc.
- Composite (group of values) - records, cursors, etc.
- Reference (other programs) - pointers
- LOB (large objects) - graphics, movies, etc.

Non-PL/SQL

- Binds and hosts variables - global values

Declaring Variables

Follow naming conventions:

- Declare one identifier per line
- Initialize identifiers by using the assignment operator (:=)

```
identifier [CONSTANT] datatype [NOT NULL]
[ := | DEFAULT expr];
```

```
DECLARE v_hiredate DATE;
v_deptno NUMBER(2) NOT NULL := 10;
v_location VARCHAR2(12) := ‘Atlanta’;
c_comm CONSTANT NUMBER := 1400;
v_count BINARY_INTEGER := 0;
v_total_sal NUMBER(9,2) := 0;
v_orderdate DATE := SYSDATE + 7;
v_valid BOOLEAN NOT NULL := TRUE;
```

%TYPE Attribute

The %TYPE attribute, used in PL/SQL variable and parameter declarations, is supported by the data server.

Declare variables according to:

- A database column definition
- Another previously declared variable

```
identifier table.column_name%TYPE ;
```

```
DECLARE
 v_name             employees.last_name%TYPE ;
 v_balance          NUMBER(7,2);
 v_min_balance      v_balance%TYPE := 10;
```

Boolean (TRUE/FALSE/NULL)

```
v_sal1 := 50000;
v_sal2 := 60000;
```

v_sal1 < v_sal2
This evaluates to TRUE

```
DECLARE
          v_flag BOOLEAN := FALSE;
BEGIN
          v_flag := TRUE;
END;
```

LOB

- Large Objects (4 GB)

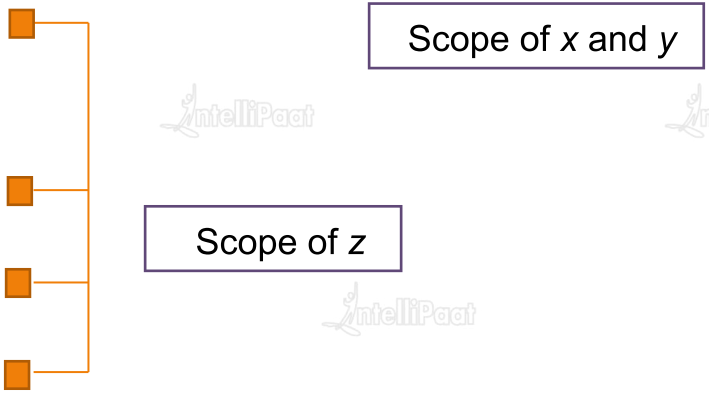
  - To store unstructured data (graphics, video, or soundwave)

• CLOB: **C**haracter **L**arge **Ob**ject	Lengthy text
• BLOB: **B**inary **L**arge **Ob**ject	Graphics, Photos, etc.
• BFILE: **B**inary **FILE**	Movies
• NCLOB: **N**ational **L**anguage **C**haracter **L**arge **Ob**ject	Other languages

Writing Executable Statements

Nested Block and Variable Scope

```
DECLARE
          x NUMBER;
          y NUMBER;
BEGIN
          .......
          DECLARE
                    z NUMBER;
          BEGIN
                    z := x ; ✓
          END;
          .......
          y := z ; ✗
END ;
```



A block can look up to the enclosing block.

A block cannot look down to the enclosed block.

Qualifying an Identifier

A qualifier can be the label of the enclosing block. Qualify an identifier using the block label prefix

```
<>
  DECLARE
                birthdate DATE;
        BEGIN
                DECLARE
                        birthdate DATE;
                        ..
                        abc.birthdate := TO_DATE(‘03-AUG-2004’,’DD-MON-YYYY’);
        END;
 ..
END;
```

Using Bind Variables

To reference a Bind Variable, prefix with colon (:)

```
VARIABLE g_monthly_sal  NUMBER
DEFINE p_annual_sal  = 5000
SET VERIFY OFF


DECLARE
        v_sal NUMBER(9,2) := &p_annual_sal;
BEGIN
         :g_monthly_sal  := v_sal / 12;
END;
/
PRINT g_monthly_sal
```

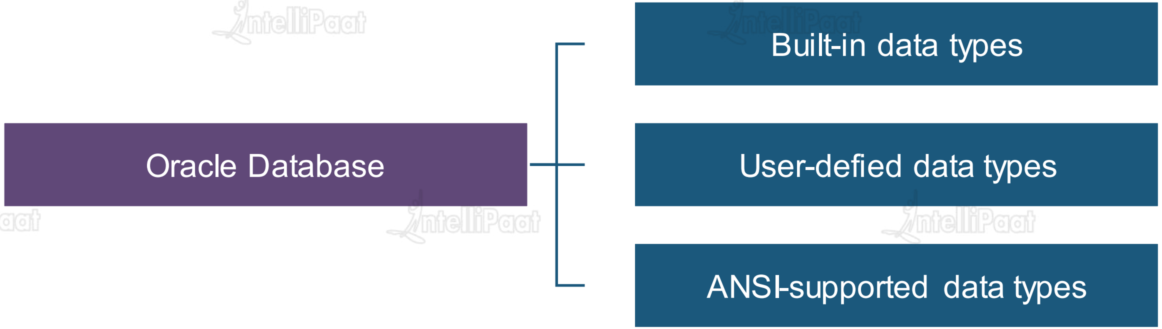
Basic SQL Constructs

Oracle SQL Data Types

- Each value manipulated by Oracle Database has a data type.
- The data type of a value associates a fixed set of properties with the value.
- These properties cause Oracle to treat values of one data type differently from values of another.

  Example: you can add values of a NUMBER data type but not values of a RAW data type.
- While creating a table or cluster, you must specify a data type for each of its columns.
- Data types define the domain of values for each column in a table.

  Example: DATE columns cannot accept the value ‘February 30’ or the values ‘2’ or ‘SHOE.’



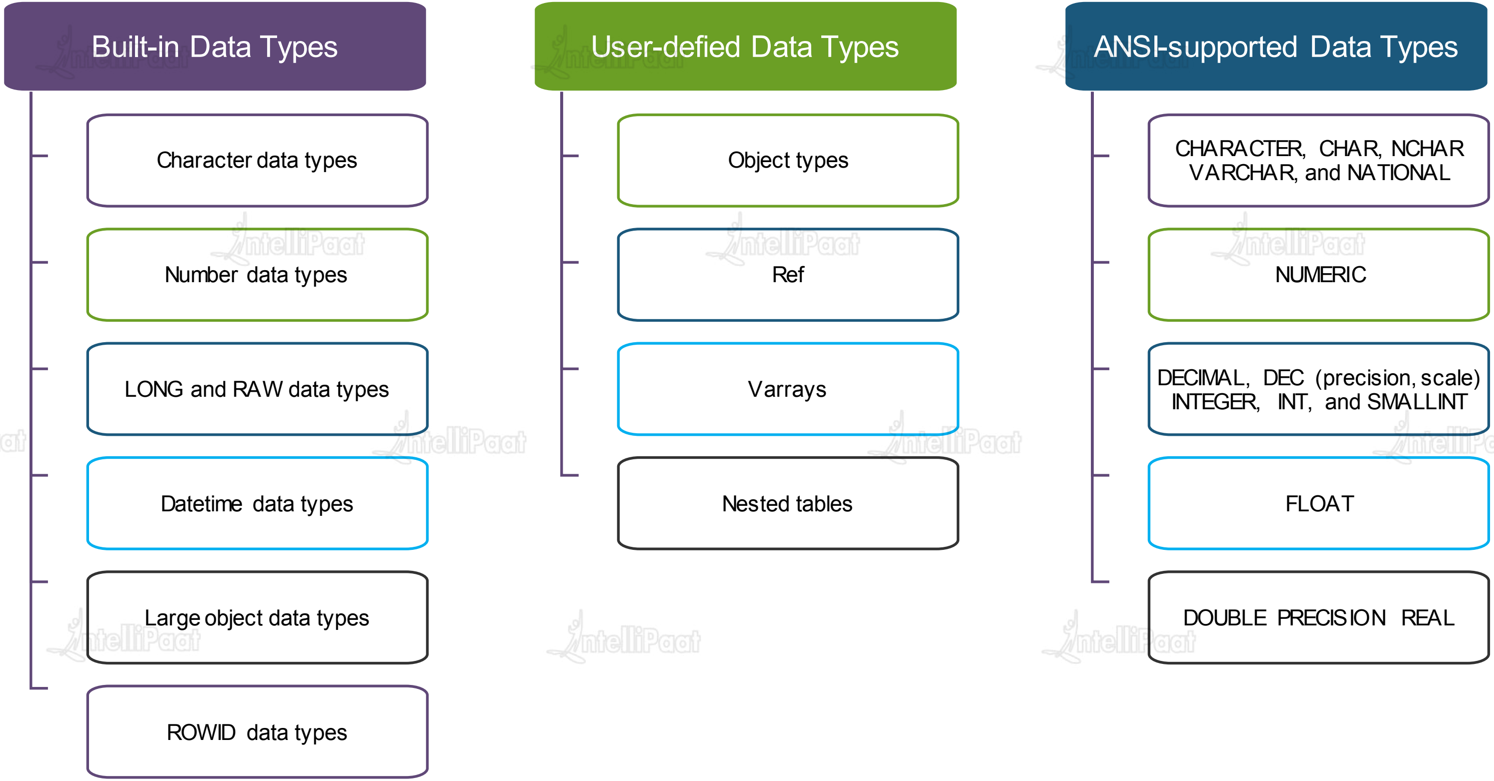
Oracle Database

Built-in data types

User-defied data types

ANSI-supported data types

Oracle SQL Data Types



Built-in Data Types

- Character data types
- Number data types
- LONG and RAW data types
- Datetime data types
- Large object data types
- ROWID data types

User-defied Data Types

- Object types
- Ref
- Varrays
- Nested tables

ANSI-supported Data Types

- CHARACTER, CHAR, NCHAR VARCHAR, and NATIONAL
- NUMERIC
- DECIMAL, DEC (precision, scale) INTEGER, INT, and SMALLINT
- FLOAT
- DOUBLE PRECISION REAL

Oracle SQL Data Types

Character Data types: Store character data

•CHAR

- Fixed-length character string; maximum allowed length is 2,000 bytes

NCHAR

- Fixed-length unicode character data; maximum allowed length is 2,000 bytes

VARCHAR2

- Variable-length character data; maximum allowed length is 1 to 4,000 bytes

NVARCHAR2

- Variable-length unicode character data only; maximum length is 1 to 4,000 bytes

Oracle SQL Data Types

Number Data types: Store numeric data

Number(p,s): Number having precision *p* and scale *s*

- Precision *p* can range from 1 to 38; scale *s* can range from −84 to 127.
- Both precision and scale are in decimal digits.
- A NUMBER value requires from 1 to 22 bytes.

FLOAT [(p)]

- A FLOAT value is represented internally as NUMBER.
- Precision *p* can range from 1 to 126 binary digits.
- A FLOAT value requires from 1 to 22 bytes.

Oracle SQL Data Types

Datetime Data Types

DATE	Timestamp	INTERVAL YEAR [(year_precision)] TO MONTH	INTERVAL DAY [(day_precision)] TO SECOND [(fractional_seconds_precision)]
This data type allows us to enter date values. The default date format is DD/MM/YYYY. Example: Jdate date	This data type allows to enter both date and time values. The default format of timestamp is DD/MM/YYYY HH/MM/SS. Example: Logintime timestamp	This data type stores a period of time in years and months, where year_precision is the number of digits in the YEAR. Accepted values are 0 to 9. The default value is 2. The size is fixed at 5 bytes.	This data type stores a period of time in days, hours, minutes, and seconds, where day_precision is the maximum number of digits in the DAY. Accepted values are 0 to 9. The default value is 2. The fractional_seconds_precision is the number of digits in the fractional part of the SECOND field. Accepted values are 0 to 9. The default value is 6. The size is fixed at 11 bytes.

Oracle SQL Data Types

LONG and RAW Data Types

•LONG

- Similar to VARCHAR2 data type. The maximum size is 2 GB.

RAW

- Stores images, logos, digital signatures, etc. The maximum size is 255 bytes.

LONG RAW

- Similar to RAW data type. The maximum size is 2 GB.

Oracle SQL Data Types

Large Object (LOB) Data Types: Store the data internally. LOB types can store large unstructured data like text, image, video, and spatial data. The maximum storage size is (4 gigabytes – 1) * (database block size).

CLOB (character large object)

- Used to store characters

BLOB (binary large object)

- Used to store binary data

NCLOB

- A character large object containing unicode characters
- Both fixed-width and variable-width character sets are supported

BFILE (binary file)

- Stores the data externally and allows us to enter BLOB, binary, XML, etc. types of values
- The length of the data type is 1 to 4 GB

Oracle SQL Data Types

ROWID Data Types

ROWID

- A string representing the unique address of a row in its table. This is for values returned by the ROWID pseudo-column

UROWID

- A string representing the logical address of a row of an index-organized table

User-defined Data Types

They use Oracle built-in data types and other user-defined data types as the building blocks of object types that model the structure and behavior of data in applications.

Object type is a schema object with three kinds of components

- Name, attributes, and methods
- For example, a data type to model an account in the bank, an address of the customer, etc.

A REF data type is a container for an object identifier. REF values are pointers to objects

- A collection is a group of values where all values are of the same type
- Oracle provides three types of collections: Indexed tables, Nested tables, and VARRAYs.

Table

- An object which is used to store data. It is a collection of rows and columns.
- Each vertical section is called a column, and the horizontal section is called a row/record.

Rules for Naming a Table or a Column:

- The table name should start with an alphabet, which contains 1 to (maximum) 30 characters.
- It should not contain spaces or any special characters such as except _# and 0 to 9.
- A table can have minimum 1 to maximum 1000 columns.
- A table can have 0 to maximum *n* number of records up to our hard disk capacity.
- Oracle reserved keywords and words should not be used as column names or table names.
- Rules which we are following for table names are applicable for column names as well.
- The numeric precision for column must be 1 to 38.

Creating a Table Command

```
Create table student (
SNO number (3),
SNAME varchar2 (20),
MARKS number (3)
);
```

This command is used to create a table. Syntax:

```
CREATE TABLE <TABLE_NAME> (
        COL_NAME1 DATATYPE(SIZE),
        COL_NAME2 DATATYPE(SIZE),
        COL_NAME3 DATATYPE(SIZE), … ,
        COL_NAMEn Datatype (size)
);
```

INSERT Command

Syntax:

```
INSERT INTO <TABLE_NAME> VALUES(VAL1, VAL2, VAL3,…………VALn);
```

Example: Insert into student values(101,Arun,60);

Error: **Arun**

Correct: **‘Arun’**

Insert into student values(101,’Arun’,60);

Note: The data for varchar2 field should be stored by enclosing it in ‘value.’

Insert into student values(102,’Anil’,86); where ‘Anil’ is of type varchar2

Insert into student values(103,’Raj’,50);

Insert into student values(104,’vijay’);//this statement is wrong

Insert into student values(105,’vijay’,null);

Null is a keyword used to represent an unavailable, undefined, or unknown value. It is neither space nor zero.

Inserting Null

Null can be inserted into a table in two different ways:

Explicit Insert

- Providing the value for the column as ‘null’ explicitly:
- Insert into student values (106,null,null);

Implicit Insert

- Implicitly inserting a null:
- Syntax: INSERT INTO <TABLE_NAME> (COL1,COL2,…..COLn) VALUES (VAL1,VAL2,…….VALn);

Example: Insert into student(rno) values (106);

This will store ‘null’ in the fields sname, marks

Insert values at runtime using ‘&’ operator

Example: INSERT INTO STUDENT VALUES (&SNO,’&SNAME’ , &MARKS);

&: Substitution variable, whose value will be accepted at runtime

Inserting Values with &&

Insert values at runtime using ‘&&’ operator

Example: INSERT INTO STUDENT VALUES (&SNO,’&SNAME’ , &&MARKS);

&&: Substitution variable, whose value will be accepted at runtime, and it applies for all instances as long as the SQL statement is executing

‘&’ Vs. ‘&&’ substitution variables: Both are used to store values into the database at runtime repeatedly.

&	&&
• Takes the input for each instance record that is getting inserted	• Takes the input only once and uses the same values for all records that are getting inserted.

Querying the Database: SELECT

DRL/DQL

- It is used to retrieve data from the database
- It has only one command

SELECT

- It retrieves data from one or more tables

Querying the Database: SELECT

Choosing all columns

- Syntax:
  - Select * from table-name;
- Example
  - Select * from student;

* refers to all columns and gets data from all columns of the student table

Choosing data from a specified column(s)

- Syntax:
  - select col-name1,co1-name1...... from tablename;
- Example
  - Select sname, marks from student;

Gets data from sname, marks columns of the student table

Note : See the table list using the following select command: SQL>select * from TAB;

Quiz

Quiz 1

What is the full form of SQL?

A Simple Query Language

B Structured Query Language

C Structured Query List

D Simple Query List

Answer 1

What is the full form of SQL?

A Simple Query Language

[✓ B Structured Query Language]

C Structured Query List

D Simple Query List

Quiz 2

Which is the subset of SQL commands used to manipulate Oracle Database structures, including tables?

A Data Definition Language(DDL)

B Data Manipulation Language(DML)

C Both of above

D None

Answer 2

Which is the subset of SQL commands used to manipulate Oracle Database structures, including tables?

- [✓ A Data Definition Language(DDL)]
- B Data Manipulation Language(DML)
- C Both of above
- D None

Quiz 3

Which of the following is true concerning systems information in an RDBMS?

A RDBMS store database definition information in system-created tables.

B This information can be accessed using SQL.

C This information often cannot be updated by a user.

D All of the above.

Answer 3

Which of the following is true concerning systems information in an RDBMS?

A RDBMS store database definition information in system-created tables.

B This information can be accessed using SQL.

C This information often cannot be updated by a user.

D All of the above.

Quiz 4

Which of the following is true about scalar data types in PL/SQL?

A They hold single values with no internal components.

B Examples of scalar data types are NUMBER, DATE, or BOOLEAN.

C PL/SQL provides subtypes of data types.

D All of the above.

Answer 4

Which of the following is true about scalar data types in PL/SQL?

A They hold single values with no internal components.

B Examples of scalar data types are NUMBER, DATE, or BOOLEAN.

C PL/SQL provides subtypes of data types.

D All of the above.

Quiz 5

Embedded SQL is which of the following?

A Hard-coded SQL statements in a program language such as Java.

B The process of making an application capable of generating specific SQL code on the fly.

C Hard-coded SQL statements in a procedure.

D Hard-coded SQL statements in a trigger.

Answer 5

Embedded SQL is which of the following?

A

Hard-coded SQL statements in a program language such as Java.

B

The process of making an application capable of generating specific SQL code on the fly.

C

Hard-coded SQL statements in a procedure.

D

Hard-coded SQL statements in a trigger.

IntelliPaat

☎ India: +91-7847955955

US: 1-800-216-8930 (TOLL FREE)

✉ sales@intellipaat.com

24/7 Chat with Our Course Advisor