PL/SQL

**Container Database(CDB):-** The CDB root, also called simply the root, is a collection of schemas, schema objects, and non-schema objects to which all PDBs belong.

Contain:

(control flies,data flies,undo,tempfiles,redo logs etc.). It also houses the data dictionary for those objects that are owned by the root container and those that are visible to all PDBs.

**Pluggable Database(PDB):** A **PDB**  is a user-created set of schemas, objects and related structures that appears logically to a client application as a separate database.

Contain: Data files and temp files to handle it’s own objects. This includes it’s own data dictionary, Containing information about those that are Specific to the PDB.

**What are possible with CDB –Container Database**

* CDB is the single entry point for Database operation.
* Create multiple PDBs, Replicate PDBs.
* Database can be tuning from one place.
* Backup can be scheduled for PDBs.
* PDBs are portable to can be relocated to another container present in another server or cloud.

**Create a PDB Syntax**

CREATE PLUGGABLE DATABASE PDB\_SALES

ADMIN USER demoadmin IDENTIFIED BY salesadm123

STORAGE (MAXSIZE 2G)

DEFAULT TABLESPACE users

DATAFILE 'E:\oracle\oradata\XE\PDB\_SALES\sales01.dbf' SIZE 250M

AUTOEXTEND ON

PATH\_PREFIX = 'E:\oracle\oradata\XE\PDB\_SALES\'

FILE\_NAME\_CONVERT = ('E:\oracle\oradata\XE\pdbseed\',

'E:\oracle\oradata\XE\PDB\_SALES\');

PDB\_SALES =Pluggable Database name

Salesadm =Admin user of the PDB i.e.-PDB\_SALES

Salesadm123=Admin user password

**To display the Pdbs in the SQL:-**

**Select \* from v$pdbs;**

**🡺**To see the database in the oracle

**Select tablespace\_name from dba\_tablespaces;**

***🡺****How to find all the PDBs inside containerDB*

**Select pdb\_name from dba\_pdbs order by pdb\_id;**

**Create a local user**

🡺Connect to SYS account

SQL> con SYS as SYSDBA

--Enter the Password

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

🡺Connect to PDB =PDB\_Sales

ALTER SESSION SET CONTAINER =PDB\_SALES;

Alter database open;

🡺Create User

CREATE USER guptadynasty Identified by gaurav45 Container= CURRENT;

GRANT CREATE SESSION,CONNECT ,RESOURCE TO guptadynasty CONTAINER=CURRENT;

🡺Connect to Database using the user

CONN guptadynasty/gaurav45 @localhost:1521/PDB\_SALES;

**🡺**To see table in the database;

**Select \* from user\_tables;**

**Select table\_name from user\_tables;**

**🡺**To grant permission for DML, DDL operation in table:-

**Grant unlimited tablespace to guptadynasty;**

Create a Common user

🡺 Connect to SYS acoount

**SQL> CONN SYS as SYSDBA**

**--Enter the Password**

🡺Create the common user using the container clause

**CREATE USER C##demouser IDENTIFIED BY demouser123**

**CONTAINER=ALL;**

**GRANT CREATE SESSION,CONNECT TO c##demouser CONTAINER=ALL;**

🡺Connect to PDB-PDB\_SALES USING demouser

**CONN C##demouser/demouser123@PDB\_SALES;**

**🡺**Grant access to common user from user guptadynasty

**Grant select on test to C##demouser;**

**🡺**All the common user can connect to any of the pdb available in the database.

Anonymous Block in PL-SQL

A PL-SQL is block structural language whose code is organized in the form of blocks. A block consist of three

Section:-

1. Declaration
2. Executable
3. Exception Handling

In a block Executable block in mandatory section where as Declaration and Exception Handling is option section.

A PL-SQL has a name, function and procedure is the example of named block. A name block is stored in oracle and can be used in future if required.

A Block without name is called **anonymous block**. An anonymous block is a block which don’t get saved in PL-SQL. So it is one time used block in the PL-SQL.

**Sample PL/SQL Block**

**Declare**

<Declaration Section>

**Begin**

<Executable Section>

**End;**

Any this which need to executed should be Inside the begin block.

**Sample Program of Anonymous Block:-**

set serveroutput on; --it will on the buffer memory which will print

**Declare**

**Begin**

DBMS\_output.put\_line(‘Welcome to PLSQL’); -- it print on the Screen

**End;**

**DBMS\_output.put\_line:-** The DBMS\_OUTPUT.PUT\_LINE procedure in Oracle PL/SQL is used to send messages from stored procedures, packages, and triggers. It is especially useful for displaying debugging information.

🡺Types of block in PL-SQL

* Anonymous Block
* Procedure
* Function

1.**Anonymous Block**:- A Block without name is called **anonymous block**. An anonymous block is a block which don’t get saved in PL-SQL. So it is one time used block in the PL-SQL.

Ex:-

set serveroutput on; --it will on the buffer memory which will print

**Declare**

**Begin**

DBMS\_output.put\_line(‘Welcome to PLSQL’); -- it print on the Screen

**Exception**

**End;**

2.**Procedure:-** PL-SQL procedure are reusable code which is same like anonymous block , The only Difference is it has name and can be use more than one time as it can be store in PL-SQL

**Syntax:-**

**Procedure<name>**

**IS**

--Declaration

**Begin**

--statement

**Exception**

**End;**

3.**Function:-** Function is same like procedure but here only difference is it has return type. It can return the value

**Syntax:-**

**Function** <Name> **Return** <Datatype>

**IS**

--Declaration

**Begin**

--statement

**Exception**

**End;**

**Generate Output from a PL/SQL Block**

**Into:-** Into clause generally used to put the value into certain variable like;

**||:-**To concat string with something like variable or string we use this ‘||’ operator

Ex:-

**Declare**

**max\_sal number(6);**

**Begin**

**select max(sal) into max\_sal from emp;**

**DBMS\_OUTPUT.PUT\_LINE('MAX SALARY IS '||max\_sal);**

**end;**

**Q:-Find the total no. employee as of today**

**set serveroutput on;**

**Declare**

**total\_emp number(6);**

**Begin**

**select count(\*) into total\_emp from emp where hiredate <= sysdate;**

**DBMS\_OUTPUT.PUT\_LINE('AS of '||sysdate||' the number of employee in emp table ='||total\_emp);**

**end;**

**output:**

As of 09-12-24 the number of employee in emp table =14;

Q:-Find the number employee from employee table department wise

Output should be like

Dept 10 the employee count =2

Dept 2- the employee count =5;

Dept 3 the employee count =6;

Ans

Declare

dept10 number(3);

dept20 number(3);

dept30 number(3);

Begin

select count(\*) into dept10 from emp where deptno=10;

select count(\*) into dept20 from emp where deptno=20;

select count(\*) into dept30 from emp where deptno=30;

DBMS\_OUTPUT.PUT\_LINE('Dept 10 the employee count =' || dept10);

DBMS\_OUTPUT.PUT\_LINE('Dept 20 the employee count =' || dept20);

DBMS\_OUTPUT.PUT\_LINE('Dept 30 the employee count =' || dept30);

end;