PLSQL:

----------------

- Procedural Language with SQL

- programming stataments and SQL Queries.

- It is extention to SQL part/ It is advanced part to SQL part.

Points happened in SQL:

- SQL is collection of predefined object such as table, views, index, sequence, etc.

- At a time only one query is executed in SQL.

- Each qury hits db. If no of hits are increasing then db performance get reduces.

- Querys are not saved in DB/ Rewrite query again and again.

- No programming statements/ constructs like condiional Stmts,

loops and branching stmts.

[ Not Available ]

Points happened in PLSQL:

- Is used to automate the business activity.

- Database server is collection of programs, that programs are responded and they can be executed on database tables and getting imformation on frontend webpage

Def:

It is collection of user defined objects. According to client requirement the object will be defined and developed.

ADV:

----

- Executing multiple queries in the form of programms.

- Reduces no of hits to DB.

- Improve the DB performance and also network performance.

- Enhancibility: existing programs are enhancible for future requirement that is it.

- Reusabiliti:ATM checking Balance.

- Modularity:Dividing process is called as modularization criteria.

PLSQL Objects:

---------------

2 Categories:

1) Programs/ Anonymous blocks:

a) Static Programs: Doesn't Accepts runtime input values.

b) Dynamic Programs: Accepts runtime input values.

2) Procedures and functions

Triggers, Types, packages etc

Programs:

Collection of statements and queries to perform set of task/operation.

Nature of programs:

Not saved in DB

Structure:

Declare ---------- Optional

<statements>;

BEGIN ---------- Mandatory

<Assignments Stmts>;

<Data Processing Stmts/Queries>;

<Output Stmts>;

EXCEPTION ---------- Optional

<Error Handling Stmts>;

END; ---------- Mandatory

Declaring variable Dynamically and how to declare variable as record type:

declare variable:

v\_name varchar2(10);

if I get value from table which is graeter than 10 size then it is unable to process.

declare variable:

v\_accno number(12);

if I get value from table which is mix alphanumeric that character type then we cant store in this variable.

this is datatype incompatibility.

which is remove by giving

Variable:

A name which refers to buffer location.

Types of Variable:

1. Simple Variable/ Scalar Varia⁮ble

a) explicite variable

b) implicite variable(dynamic variable):

ex. vcity varchar2(10);

ex. vcity cust.city%type;

2. Composite Variable:

Under this variable you able to store more than one value. then it is composite variable.

a) Table Based Record Variable:

- Here we store the single record of table in one variable.

- Here %rowtype is used.

Syntax: var1 tablename%rowtype

Declaring table based record:

ex. emprec emp%rowtype;

Storing a record:

select \* into emprec from emp where empno=7566

Accessing values from emprec:

emprec.empno

emprec.ename

emprec.job.........

b) User Defined Record Variable

If you want to create our own record type structure with required column then used this type of variable.

For that Type structure is used.

How to define that record structure:

We know that record is collection of columns

(Step I): define that record structure

Syntax:

TYPE <name> IS RECORD

(

Col\_Name1 DataType(Size),

Col\_Name1 DataType(Size),

-----

-----

Col\_Name-n DataType(Size),

);

(Step II): Create a variable of above type

Syntax:

var1 Above\_Type\_name ;

Advantages:

a) It saves the memory reserved for variable.

b) It is also reducing number of variable declaration

c) It is also supporting columns from multiple tables.

(Step I): define that record structure

Example:

TYPE emptype IS RECORD

(

enm emp.ename%type,

esal emp.sal%type,

ejob emp.job%type

);

(Step II): Create a variable of above type

Example:

emprec emptype;

(Step III): How to access Values from record type

emprec.enm, emprec.esal, emprec.ejob.

3. Cursor variable

4. Array type variable