DBMS Assignment 1

Dharmit Shah SYIT 330

Q1 Explain INTO clause with syntax and example.

- Into clause used to fetch a single row of data from a table into variables in the simplest and the fastest way.

  Syntax:

  SELECT select_list **INTO** variable_list **FROM** table_name **WHERE** condition;

```
declare
totemp int;
begin
select count(*) into totemp from emp_dharmit where deptno=10 and job='MANAGER';
dbms_output.put_line(totemp);
end
/
```

Results Explain Describe Saved SQL History

```
1

Statement processed.

0.37 seconds
```

Q2 Explain %type with syntax and example.

- %TYPE attribute provides the data type of a database column to a variable. It is generally used when variables need to hold the value of the same type as declared in the database.
  Syntax:
  variablename tablename.column %type [:=value];

```
declare
d emp_dharmit.sal%type;
begin
select sum(sal) into d from emp_dharmit where deptno=20;
dbms_output.put_line('total salary ' ||d);
end;
/
```

Results Explain Describe Saved SQL History

```
total salary 10875

Statement processed.

0.00 seconds
```

Q3 Write a PL/SQL block to print the total number of employees working as MANAGER in deptno 20.

```
declare
temp int;
begin
select count(*) into temp from emp_dharmit where deptno=20 and job='MANAGER';
dbms_output.put_line('the total number of employees working as MANAGER in deptno 20 are '||temp);
end;
/
```

Results Explain Describe Saved SQL History

```
the total number of employees working as MANAGER in deptno 20 are 1

Statement processed.
```

Q4 What are default variables? How can they be initialized in PL/SQL blocks? Can the value of default variables be changed during execution of programs, explain with examples?

- When a variable is declared in pl/sql it assigns a default value of **NULL** or a default value can optionally be given to it but in order to change the value, you have to re-assign in between the begin and end block. For example:

```
SQL> declare
  2  message varchar(30) default 'HELLO DHARMIT';
  3  begin
  4  dbms_output.put_line(message);
  5  end;
  6  /
HELLO DHARMIT

PL/SQL procedure successfully completed.

SQL> declare
  2  message varchar(30) default 'HELLO DHARMIT';
  3  begin
  4  message := 'NAMASTE';
  5  dbms_output.put_line(message);
  6  end;
  7  /
NAMASTE

PL/SQL procedure successfully completed.
```

Q5 What are datatypes? Briefly describes its types. Also give an example.

- Datatypes are used to define how the data will be stored, handled, and treated by Oracle during the data storage and processing.
- The data types are as follows:
    - Char: This data types stores the string value, and the size of the string is fixed at the time of declaring the variable.
      Syntax
      Variable char(10) := ‘assign vale’;
      Example:

```
SQL> declare
  2  a char(8) := 'Dharmit';
  3  begin
  4   dbms_output.put_line(a);
  5  end;
  6  /
Dharmit

PL/SQL procedure successfully completed.
```

    - Varchar: this data type stores the string, but the length of the string is not fixed.
      Syntax
      Variable varchar(20) := ‘Dharmit is cool’;
      Example:

```
SQL> declare
  2  b varchar(30) := 'Dharmit is cool';
  3  begin
  4   dbms_output.put_line(b);
  5  end;
  6  /
Dharmit is cool

PL/SQL procedure successfully completed.
```

    - Number: This data type stores fixed or floating-point numbers up to 38 digits of precision. It is used to work with fields that will contain only numberic data.
      Syntax:
      Variable number(8,2);
      Variable int;
      Variable float;

Example:

```
SQL> declare
  2  a number(6,2) := 300.72;
  3  b int := 4;
  4  c float := 43.6;
  5  begin
  6   dbms_output.put_line(a || ','||b||','||c);
  7  end;
  8  /
300.72,4,43.6

PL/SQL procedure successfully completed.
```

- Data time: This data type stores the value in date format, as of date, month, and year. Whenever a variable is defined with DATE data type along with the date it can hold time information is set to 12:00:00 if not specified.

  Syntax:

  Current_date DATE := SYSDATE;

  Example:

```
SQL> declare
  2  current_date date := sysdate;
  3  begin
  4  dbms_output.put_line(current_date);
  5  end;
  6  /
30-JUL-21

PL/SQL procedure successfully completed.
```