### Executing SQL ====>
========================

* From the body of a PL-SQL script we are only allowed to execute DML and TCL statements i.e INSERT , UPDATE , DELETE,COMMIT, ROLLBACK are the only valid SQL commands which can be executed from a PL-SQL script.

* Execution of SELECT is allowed but with the help of CURSOR.

## Executing INSERT ===>
=========================

# WAP to accept values for roll_no , name , per columns and insert the record in the table STUDENTS.

* For writing the above script we will require 4 variables declared to be of the same data type and size as the 3 columns of the table STUDENTS.

* This is done use %TYPE attribute.

* The %TYPE attribute allow you to declare a constant, variable, or parameter to be of the same data type as an existing database column.

* Syntax :-
# <var_name> <table_name>.<col_name>%TYPE;

* Example :-
# roll Student.roll_no%TYPE;

```
declare
roll students.roll_no%type;
name students.name%type;
per students.per%type;
begin
roll := &roll_no;
name := '&name';
per := '&per';
insert into students values(roll, name, per);
dbms_output.put_line('Insert Successfully');
commit;
end;
```

## Executing DELETE ===>
=========================

# WAP to accept value for roll_no from the user and delete the record of that student from the table STUDENTS.

```
declare
roll students.roll_no%type;
begin
roll := &roll_no;
delete from students where roll_no = roll;
```

```
dbms_output.put_line('Record Deleted');
commit;
end;
```

## Executing UPDATE ===>
=========================

# WAP to accept value for roll_no and per from the user and increase the per of the student with the given roll_no by adding the given per in the table STUDENTS.

```
declare
roll students.roll_no%type;
p students.per%type;
begin
roll := &roll_no;
p := &per;
    upadte students set per = per + p where roll_no = roll;
    dbms_output.put_line('Record Updated');
    commit;
end;
```

## PL-SQL Select Into ===>
===========================

* PL/SQL SELECT INTO statement is the simplest and fastest way to fetch a single row from a table into variables.

* The following illustrates the syntax of the PL/SQL SELECT INTO statement :-

```
SELECT column_list
INTO variable_list
FROM table_name
WHERE condition;
```

# WAP to accept a roll_no from the user and display the name of the student from the table STUDENTS.

```
declare
roll students.roll_no%type;
naam students.name%type;
begin
roll := &roll_no;
select name into naam
from students where roll_no = roll;
dbms_output.put_line('Name is ' || naam);
end;
```

## Selecting Complete Row ===>
==============================

# WAP to accept a roll_no from the user and display the complete record of the student from the table STUDENTS.

## Select Into Common Errors ===>
=================================

* If the number of columns and expression in the SELECT clause is greater than the number of variables in the INTO clause, Oracle issues this error:
# ORA-00947: not enough values

* Oracle issues the following error if the number of columns and expression in the SELECT clause is less than the number of variables in the INTO clause:
# ORA-00913: too many values