DML : Data Manipulation Language ====>
==========

* DML Commands stand for Data Manipulation Language. These commands deal with changing the content of the table i.e they deal with table data.

* The three popular commands in this category are:
- Insert
- Update
- Delete

INSERT Command ====>
==========

* The INSERT command is used to insert a new row into the table.

* There are multiple syntaxes of INSERT command and they are:
- Standard Insert
- Inserting into selected columns
- Interactive Insert

Standard Insert ====>

```
Syntax:   Insert into <table_name> values(<list_of_values.>);
Examples: insert into Students values(101, 'Prashant', '28-Aug-2002');
          insert into Students values(102, 'Shivam', '8-Sep-2001');
          insert into Students values(103, 'Santosh', '24-Jan-2001');
```

```
SQL> select * from Students;

    STD_ID S_NAME                         DOB
---------- ------------------------------ ---------
       101 Prashant                       28-AUG-02
       102 Shivam                         08-SEP-01
       103 Santosh                        24-JAN-01
```

Selective Insert ====>

* Some times , we may not want to insert values in all the columns of the table.

* In such cases , Oracle has provided us another syntax of INSERT command called as Selective Insert.

* In this we specify the names of columns after the table name in the INSERT command in which we want to insert the values followed by the list of values as before.

```
Syntax:   Insert into <table_name>(<list_of_columns>)
values(<list_of_values>);
Examples: insert into Students(s_name, std_id) values('Rishi', 107);
          insert into Students(s_name, dob) values('Vishal', '05-Aug-
1999');
```

```
SQL> select * from Students;

    STD_ID S_NAME                         DOB
---------- ------------------------------ ---------
       101 Prashant                       28-AUG-02
       102 Shivam                         08-SEP-01
       103 Santosh                        24-JAN-01
       107 Rishi                           (Null)
     (null)Vishal                         05-AUG-99
```

NULL or null ===>

* In the database world, NULL or null , is special.
* It is a marker for missing or unknown information.
* NULL is special in the sense that it is not a value like a number, character string, or date.

Interactive Insert ===>

* Oracle allows us to write the INSERT command in such a way that it demands input at run time.

* In simple words we can say that just like programming languages have functions for input , similarly, Oracle also has a provision to accept input at runtime.

* To do this we use the symbol of & in our INSERT command and this symbol is called as SUBSTITUTOR.

* This is because when the command is executed , the value is accepted from the user and it replaces the symbol of &.

Syntax: Insert into <table_name>values(&<some_text>,&<some_text>);

Example:

```
SQL> insert into Students values(&std_id, &s_name, &dob);
Enter value for std_id: 109
Enter value for s_name: 'Rohit'
Enter value for dob: '02-May-2002'
old   1: insert into Students values(&std_id, &s_name, &dob)
new   1: insert into Students values(109, 'Rohit', '02-May-2002')

SQL> select * from Students;
    STD_ID S_NAME                         DOB
---------- ------------------------------ ---------
       106 Vishal
       107 Rishi
           Vishal                         05-AUG-99
       109 Rohit                          02-MAY-02
```

UPDATE Command ====>
==========

* The UPDATE command is used to update columns of existing rows in the table.

* The UPDATE command has 2 versions:
- Update ……
- Update …… Where <test_cond>

* The first version updates the specified column of all the rows

* The second version is called SEARCHED UPDATE as it uses the WHERE clause containing a CONDITION and updates the columns of only those rows whose data matches the given
condition.

Syntax Of First Version ===>

* We can update multiple columns in a single update command.

- Syntax: Update <table_name> set <col_name>=<value>,<col_name>=<value>;
- Example: Suppose we want to increment s_id of every student by 100 , then our
command will be:

```
==> Update students set s_id = s_id + 100;
```

STD_ID	S_NAME	DOB
201	Prashant	28-AUG-02
202	Shivam	08-SEP-01
203	Santosh	24-JAN-01
204	Vikas	02-FEB-00
205	Pawan	18-JUL-02
206	Vishal	
207	Rishi Vishal	05-AUG-99
209	Rohit	02-MAY-02

Syntax Of Second Version ===> (SEARCHED UPDATE)

STD_ID	S_NAME	DOB
101	Prashant	28-AUG-02
102	Santosh	24-JAN-01
103	Shivam	08-MAR-02
104	Vikas	12-JUL-00
105	Vishal	

- Syntax: Update <table_name> set <col_name>=<value>,<col_name>=<value> where <test_cond>;

- Example: Suppose we want to change dob of student whose std_id is 104 to 20-Sep-1991 , then our command will be-
* update students set dob = '20-Sep-1991' where std_id = 104;

STD_ID	S_NAME	DOB
101	Prashant	28-AUG-02
102	Santosh	24-JAN-01
103	Shivam	08-MAR-02
104	Vikas	20-SEP-91
105	Vishal	

- Example: Suppose we want to change dob of student to 15-Aug-97 whose dob is null , then our command will be-
* Update students set dob = '15-Apr-97' where dob is null;

STD_ID	S_NAME	DOB
101	Prashant	28-AUG-02
102	Santosh	24-JAN-01
103	Shivam	08-MAR-02
104	Vikas	20-SEP-91
105	Vishal	15-Apr-97

DELETE Command ====>
==========

* The DELETE command is used to delete/remove one or more rows from the table.

* Remember , DELETE never deletes columns , it only deletes rows from the table.

* The DELETE command has 2 versions:
- Delete ……
- Delete …… Where <test_cond>

* The first version deletes all the rows.

* The second version is called SEARCHED DELETE as it uses the WHERE clause containing a CONDITION and deletes or removes only those rows whose data matches the given condition.

First Version ==>

- Syntax : Delete from <table_name> ;
- Example : Delete from Students;

Second Version ==>

- Syntax : Delete from <table_name> where <test_cond> ;
- Example: delete from students where std_id = 105;

STD_ID	S_NAME	DOB
101	Prashant	28-AUG-02
102	Santosh	24-JAN-01
103	Shivam	08-MAR-02
104	Vikas	20-SEP-91