

1. CREATE AND MANIPULATE VARIOUS DB OBJECTS FOR A TABLE

DATA DEFINITION LANGUAGE

1. CREATE TABLE COMMAND

- A table is a unit of storage which holds data in the form of rows and columns.
- In a table,
 - A unique column name should be specified.
 - Proper data type along with its width should be specified.
 - 'Not null' condition can be included when needed, by default it is 'Null'.

Syntax: create table <table name> (column definition1, column definition 2, ...);

Example: create table student (regno number (11), name varchar2 (25), address varchar2(30), dept varchar2 (3));

Output:

Table created.

2. ALTER TABLE COMMAND

- This command is used to add a new column, modify the existing column definition, include or drop integrity constraints.

Syntax: alter table <table name> modify (column definition ...);
alter table <table name> add (column definition ...);

Example:

I. alter table student modify (name varchar2 (30));

Output:

Table altered.

II. alter table student add (comments long);

Output:

Table altered.

3. TRUNCATE TABLE COMMAND

- The truncate command deletes all rows from the table. Only the structure of the table remains.

Syntax: Truncate table <table name>;

Example: Truncate table student;

Output:

Table truncated.

4. DESC COMMAND

- This command will display the structure of the table.

Syntax: Desc <table name>;

Example: Desc student;

Output:

Name	Null?	Type

REGNO		NUMBER(11)
NAME		VARCHAR2(30)
DEPT		VARCHAR2(3)
COMMENTS		LONG

5. DROP TABLE COMMAND

- The drop table command is used to delete the table permanently from the database.

Syntax: Drop table <table name>;

Example: Drop table student;

Output:

Table dropped.

DATA MANIPULATION COMMANDS

1. INSERT COMMAND

- Used to add one or more rows to a table.
- While using this command the values are separated by commas and the data types char and date are enclosed in apostrophes.
- The values must be entered in the same order as they are defined in the table.

Syntax:

- To insert all column values for each row
SQL> Insert into <table_name> values (a list of data values);
- To insert specific column values for each row
SQL> Insert into <table_name> (col_names) values (list of values);

Examples:

1. The example given below inserts a record in Student table. The structure of Student table is:

Name	Null?	Type

REGNO		NUMBER(11)
NAME		VARCHAR2(25)
DOB		DATE
ADDR		VARCHAR2(25)
DEPT		VARCHAR2(3)

```
SQL> Insert into student values (42207621001, 'Ananthi', to_date ('20/04/1987',
'dd/mm/yyyy'), 'Kanchipuram', 'MCA');
```

Output:

1 row created.

2. To insert more than one record using a single insert command.

```
SQL> Insert into student values (&RegNo, '&Name', '&DOB', '&addr', '&Dept');
```

Output:

Enter value for regno: 42207621035

Enter value for name: Prince

Enter value for DOB: 20-APR-87

Enter value for addr: Arakkonam

Enter value for dept: MCA

old 1: insert into student values (&RegNo, '&Name', '&DOB', '&addr'

new 1: insert into student values (42207621035, 'Prince', '20-APR-87',
'kancheepuram', '

1 row created.

- The above command however inserts only one row. To insert another row, type a / in the SQL prompt.

3. To insert a record with values for specific columns.

```
SQL> insert into student (RegNo, Name, Dept) values (42207621050, 'Subramani',  
'MCA');
```

Output:

1 row created.

4. The following example copies all the rows from Student1 table to Student table, provided a table named Student exists having the same structure as Student1.

```
SQL> insert into Student (select * from Student1);
```

Output:

3 rows created.

2. SELECT COMMAND

- Used to perform a query. The query is a request for information.

Syntax: Select column_name ... from table_name ... where conditions [order by column_name ...];

Examples:

1. The following example selects all the columns from the table Student for which RegNo less than or equal to 42207621035 ordered by RegNo in descending fashion. The default ordering is ascending.

SQL> Select * from Student where RegNo <= 42207621035 Order By RegNo Desc;

Output:

```
REGNO NAME
-----
ADDR
-----
DEP
---
4.2208E+10 Prince
'Arakkonam', MCA

4.2208E+10 Ananthi
'Kanchipuram', MCA

REGNO NAME
-----
ADDR
-----
DEP
---
```

- To select only specific columns, specify the column names instead of * in the select command.
2. The example given below creates a new table Student1 from the existing table Student along with its records.

SQL> Create table Student1 as select * from Student;

Output:

Table created.

3. The following command displays the records from department table whose dcode begins with 'M'.

```
SQL> select * from department where dcode like 'M%';
```

Output:

DCO	DNAME	HODNAME
MBA	Management Studies	Dr. Padmini
MCA	Computer Applications	Dr. Pandurangan

4. The command given below lists all the columns from department table whose dcode is three letters long and the middle letter is 'C'.

```
SQL> select * from department where dcode like '_C_';
```

Output:

DCO	DNAME	HODNAME
MCA	Computer Applications	Dr. Pandurangan

3. UPDATE COMMAND

- Used to alter the column values in a table.
- Specific rows could be updated based on a specific condition.

Syntax: Update table_name set field = value, ... where condition;

- The 'where' clause and the 'set' clause can also include queries.

Examples:

1. The following command updates the values for Name and Dept columns in the Student table for RegNo 42207621050.

```
SQL> update Student set Name = 'Abbas', Dept = 'MBA' where RegNo = 42207621050;
```

Output:

1 row updated.

- The result of the above command can be verified by selecting rows from the Student table.

4. DELETE COMMAND

- Used to delete one or more rows to a table.

Syntax: delete from <table_name> where conditions;

Examples:

1. The following example will delete a row from Student table whose RegNo is 42207621050.

SQL> delete from Student where RegNo = 42207621050;

Output:

1 row deleted.