* SQL statements :-

I] Data Defination Language :-

1] CREATE :-

Syntax :-

```
Create table table_name
(
COL-NAME      DATATYPE      NULL/NOT NULL,
CONSTRAINT    REFERENCE-NAME    CONSTRAINT (COL-NAME)
);
```

Ex.
```
Create table student
(
  STDID       CHAR(5)        NOT NULL,
  STDNAME     VARCHAR(20)    NOT NULL,
  SCONT       CHAR(10)       NOT NULL,
CONSTRAINT    PK1        PRIMARY KEY (STDID),
CONSTRAINT    U1         UNIQUE (SCONT),
CONSTRAINT    CK1        CHECK (LENGTH(SCONT) = 10)
);
```

e :- To save the queries : ① Go to spool files
② Save files in documents with the extension '.TXT'.
③ After solving all queries end it with spool off; .

2] RENAME :-

i] To rename the table

Syntax :- RENAME OLD TABLE TO NEW TABLE_NAME.

ii] To add new column

Syntax :- ALTER TABLE TABLE_NAME
ADD COL-NAME DATATYPE NULL/NOT NULL;

Ex. ALTER TABLE STUD
ADD SEMAIL VARCHAR(20) NOT NULL;

Note :- If we have data inside table and later we have to add new column then during adding ✱ take 'NULL'.

iii] To rename the column

Syntax :- ALTER TABLE TABLE_NAME
RENAME COLUMN OLD COL-NAME TO NEW COL-NAME;

Ex. ALTER TABLE STUD
RENAME COLUMN STDID TO SID;

iv] To Delete column

Syntax :- ALTER TABLE TABLE_NAME
DROP COLUMN COL-NAME ;

v] Modify Datatype

Syntax :- ALTER TABLE TABLE_NAME
MODIFY COL-NAME DATATYPE ;

vi] To change NULL TO NOT NULL & vice versa

Syntax :- ALTER TABLE TABLE_NAME
MODIFY COL-NAME DATATYPE NOT NULL ;

vii] To Add constraint

Syntax :- ALTER TABLE TABLE_NAME
ADD CONSTRAINT REFERENCE-NAME CONSTRAINT(COL-NAM

viii] To delete constraint

Syntax :- ALTER TABLE TABLE_NAME
DROP CONSTRAINT REFERENCE-NAME ;

DOMS | Page No.
Date / /

* Create Child Table :-

```
CREATE TABLE TABLE_NAME
(
COL-NAME        DATATYPE        NULL / NOT NULL
CONSTRAINT      REFERENCE_NAME  FOREIGN KEY (COL_NAME)
                (FK1)           REFERENCES TABLE_NAME
                                (COL-NAME)
);
```

Note: ① The datatype for Primary Key and foreign key should be same.

② We have to drop child table first then drop parent table.

Syntax :- `DROP TABLE TABLE_NAME;`

③ To restore dropped we use flashback

Syntax :-

```
FLASHBACK TABLE TABLE_NAME TO BEFORE DROP;
```

DOMS | Page No.
Date / /

II] Data Manipulation Language :-

1] **INSERT** :- It is use to insert data into the table.

Syntax :- 1] INSERT INTO TABLE_NAME VALUES ( );

Ex. INSERT INTO STUD VALUES ('S01', 'SUNDRA', 9876543210);

2] INSERT ALL
INTO TABLE_NAME (COL-NAME 1, COL-NAME2, ... COL_NAMEN)
VALUES ('COL1', 'COL2', ... 'COLN')
INTO TABLE_NAME (COL-NAME 1, COL_NAME 2, ..., COL-NAMEN)
VALUES ('COL1', 'COL2', ... , 'COLN')
SELECT * FROM DUAL;

Note :- The 2nd syntax is used to insert multiple rows at a same time.

2] **UPDATE** :-

Syntax :- UPDATE TABLE_NAME
SET COL-NAME = 'NEW VALUE'
WHERE <CONDITION>;

Ex. UPDATE STUD
SET SBRANCH = 'INFOTECH'
WHERE STDID = 'S04';

3] DELETE :-

Syntax :- DELETE
FROM TABLE_NAME
WHERE <CONDITION> ;

Ex. DELETE
FROM STUD
WHERE STDID = '503' ;

III] Transaction Control Language :-

1] **COMMIT** :- (i) It is a statement use to save all the transactions of data manipulation language.
(ii) Once we commit all the data get save permanently.

Syntax :- COMMIT;

Note : If we close the app or computer gets off before commiting then all the workdone gets erase. So always be carefull and commit the data.

2] **ROLLBACK** :- (i) It is a statement use to get back the transaction of data manipulation language before commit.

Syntax :- ROLLBACK;

Note : If, after the commit we have done some transactions and before commit if we use rollback all the transactions gets cancelled and we come back at 1st committed point / we will not have changes in data.

3] **SAVEPOINT** :- (i) It is a statement used to save perticular transaction without commit.
(ii) It also work as temporary barrier to prevent total rollback.

Syntax :- SAVEPOINT REFERENCE_NAME ;

Ex. SAVEPOINT S1 ;

(iii) The savepoint is given before we do transaction.
(iv) Every transaction have its own savepoint.
(v) If we have savepoints as S1, S2, S3, ... Sn and we use rollback to S3 then all the transactions after S3 get cancelled / rollbacked.
(vi) The rollback does not shift directly to S1 OR S2 without rolling S3 OR Sn-1. It will rollback all the transactions before that savepoint upto which we need rollback.

Ex.
```
SAVEPOINT S1;
UPDATE
SET SADD = 'SWARGATE, PUNE'
WHERE STDID = 'STD02';

SAVEPOINT S2;
DELETE
FROM STUD
WHERE STDID = 'STD03';

ROLLBACK TO S2;
```

Output ⟹ Here the rollback will be done till S2. i.e. The Delete transaction gets cancelled & Update transaction remains unchanged.

IV] Data Control Language :-

1] **GRANT** :- This statement is use to give permission.

Syntax :- To create new user and grand him/her permission to do work.

```
CONN
ENTER USER_NAME : SYSTEM
ENTER PASSWORD : ***

CREATE USER USER_NAME IDENTIFIED BY PASSWORD;
GRANT CREATE SESSION TO USER_NAME;
GRANT UNLIMITED TABLESPACE TO USER_NAME;
GRANT CREATE TABLE TO USER_NAME;
```

Ex. CREATE USER ABHI IDENTIFIED BY A3738;
op→ USER CREATED.

GRANT CREATE SESSION TO ABHI;
op→ GRANT SUCCEEDED.

GRANT UNLIMITED TABLESPACE TO ABHI;
op→ GRANT SUCCEEDED.

GRANT CREATE TABLE TO ABHI;
op→ GRANT SUCCEEDED.

Syntax:- GRANT SELECT ON TABLE_NAME TO USER_NAME;

Ex. GRANT SELECT ON EMP TO ABHI;

then,

```
CONN
ENTER USER_NAME = ABHI
ENTER PASSWORD = *****
CONNECTED
```

then,

```
SELECT * FROM SCOTT.EMP;
```

2] **REVOKE** :- This statement is use to take back the permission from user.

Syntax:- REVOKE SELECT ON TABLE_NAME FROM USER_NAME;

Ex. REVOKE SELECT ON EMP FROM ABHI;