DDL

(DATA DEFINITION LANGUAGE)

1. *CREATE*
2. *RENAME*
3. *ALTER*
4. *TRUNCATE*
5. *DROP*

ALTER

"IT IS USED TO MODIFY THE STRUCTURE OF THE TABLE."

1. **TO ADD A COLUMN.**
2. **TO REMOVE A COLUMN.**
3. **TO RENAME A COLUMN.**
4. **TO MODIFY THE DATATYPE.**
5. **TO MODIFY THE CONSTRAINTS.**
6. **TO ADD NEW CONSTRAINTS**
7. **TO ASSIGN A FOREIGN KEY.**

1. TO ADD A COLUMN.

```
ALTER TABLE TABLE_NAME
ADD COLUMN_NAME DATATYPE CONSTRAINTS;
```

2. TO REMOVE A COLUMN.

```
ALTER TABLE TABLE_NAME
DROP COLUMN COLUMN_NAME;
```

3. **TO RENAME A COLUMN.**

```
ALTER TABLE TABLE_NAME
RENAME COLUMN EXISTING_COLUMN_NAME TO NEW_COLUMN_NAME;
```

4. **TO MODIFY THE DATATYPE**

```
ALTER TABLE TABLE_NAME
MODIFY COLUMN_NAME NEW_DATATYPE;
```

5. **TO MODIFY THE CONSTRAINT**

```
ALTER TABLE TABLE_NAME
MODIFY COLUMN_NAME EXISTING DATATYPE NEW_CONSTRAINT (NULL/NOT NULL);
```

6. **TO ADD NEW CONSTRAINTS**

```
ALTER TABLE TABLE_NAME
ADD CONSTRAINT CONSTRAINT_REFERENCE_NAME CONSTRAINT_TYPE (COLUMN_NAME);
```

7. **TO ASSIGN NEW CONSTRAINTS**

STEP 1: ADD A COLUMN

SYNTAX:

```
ALTER TABLE TABLE_NAME
    ADD COLUMN_NAME DATATYPE;
```

STEP 2: ASSIGN FOREIGN KEY

```
ALTER TABLE TABLE_NAME
ADD CONSTRAINT CONSTRAINT_REFERENCES_NAME FOREIGN KEY(COLUMN_NAME);
```

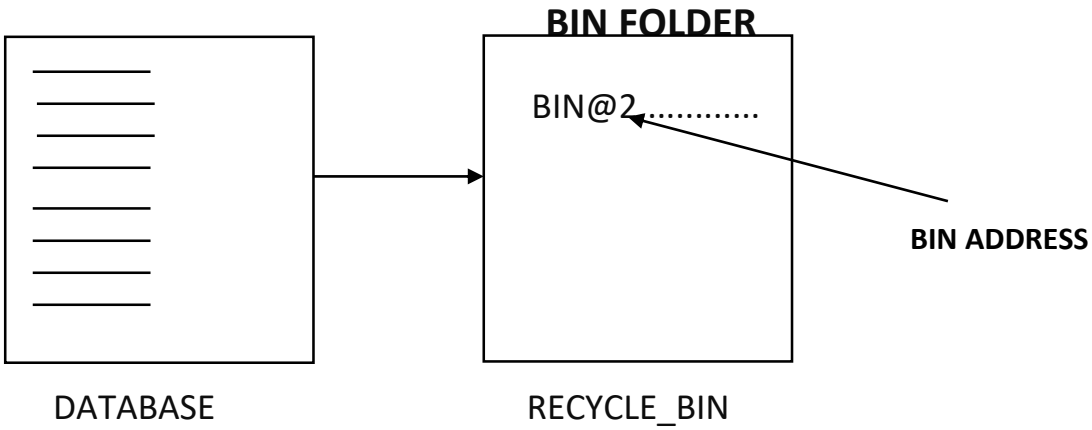
TO REMOVE CONSTRAINT

```
ALTER TABLE TABLE_NAME
DROP CONSTRAINT CONSTRAINT_REFERENCES_NAME;
```

DROP

“IT IS USED TO REMOVE THE TABLE FROM THE DATABASE.”

SYNTAX: DROP TABLE TABLE_NAME;



FLASHBACK: It is used to restore the table from the recycle bin.

SYNTAX:

FLASHBACK TABLE TABLE_NAME

TO BEFORE DROP;

PURGE: It is used to remove the table from the recycle bin.

SYNTAX:

PURGE TABLE TABLE_NAME;

```
SELECT *
FROM RECYCLE BIN;
```

DML

(DATA MANIPULATION LANGUAGE)

"IT IS USED TO MANIPULATE THE OBJECT/TABLE BY PERFORMING INSERTION, UPDATION, DELETION OF RECORDS."

1. **INSERT**
2. **UPDATE**
3. **DELETE**

INSERT: **It is used to create/insert the records in table.**

SYNTAX:

```
INSERT INTO TABLE_NAME VALUES (V1, V2, V3……);

INSERT INTO TABLE_NAME VALUES (&COL1, &COL2, &COL3….);
```

UPDATE: **It is used to modify/update the existing value in the table.**

SYNTAX:

```
UPDATE TABLE_NAME

SET COL_NAME=VALUE, COL_NAME=VALUE…….

[WHERE <FILTER_CONDITION>];
```

DELETE: **It is used to remove/delete a particular record from the table.**

SYNTAX:

```
DELETE FROM TABLE_NAME

[WHERE <FILTER_CONDITION>];
```

NOT COMMIT -----> DML

COMMIT------> DDL

TCL

(TRANSACTION CONTROL LANGUAGE)

1. **COMMIT: "This statement is used to SAVE the transactions into the database."**

Syntax: COMMIT;

2. **ROLLBACK: "This statement is used to obtain only the saved data from your database it will bring you to the point where you have committed last time.**

Syntax: ROLLBACK;

3. **SAVEPOINT: "This statement is used to mark the positions or restoration points.**

Syntax: SAVPOINT SAVEPOINT_NAME;

Syntax: ROLLBACK TO SAVEPOINT_NAME;

DCL

(DATA CONTROL LANGUAGE)

"This statement is used to control the flow of the data between the users."

1. **GRANT**
2. **REVOKE**

1. **GRANT: "This statement is used to give a permission to the user."**

```
SYNTAX: GRANT SQL_STATEMENT
        ON TABLE_NAME
        TO USER_NAME;
```

2. **REVOKE: "This statement is used to take back the permission from the user."**

```
SYNTAX: REVOKE SQL_STATEMENT
        ON TABLE_NAME
        FROM USER_NAME;
```

STEPS OF USING GRANT & REVOKE

SHOW USER;

USER IS 'SCOTT'

CONNECT

ENTER USER NAME: HR

ENTER PASSWORD: TIGER (*****)

SHOW USER;

USER IS 'HR'

SELECT *

FROM SCOTT.EMP; (ERROR: TABLE OR VIEW DOES NOT EXIST).

CONNECT

ENTER USER NAME: SCOTT

ENTER PASSWORD: TIGER (*****)

GRANT SELECT * ON EMP TO HR;

CONNECT

ENTER USER NAME: HR

ENTER PASSWORD: TIGER (*****)

SELECT *

FROM SCOTT.EMP;