SQL STATEMENTS

Rohan Singh R

I.DATA DEFINITION LANGUAGE:

1.CREATE :

SYNTAX:

```
CREATE TABLE   table_name
(
COLUMN_NAME_1  DATATYPE  NOT NULL / [NULL],
COLUMN_NAME_2  DATATYPE  NOT NULL / [NULL],
.
.
COLUMN_NAME_n  DATATYPE  NOT NULL /[ NULL],
CONSTRAINT constraint_ref_name UNIQUE(COLUMN_NAME),
CONSTRAINT constraint_ref_name CHECK(CONDITION),
CONSTRAINT constraint_ref_name PRIMARY KEY(COLUMN_NAME),
CONSTRAINT constraint_ref_name FOREIGN KEY(COLUMN_NAME)
REFERENCES parent_table_name (COLUMN_NAME)
);
```

2.RENAME:

SYNTAX:

```
    RENAME  current_table_name  TO  New_table_name;
```

3.ALTER:

SYNTAX:

1.TO ADD A COL :

ALTER TABLE table_name

ADD COLUMN_NAME DATATYPE[NULL/NOT NULL];

2.TO DROP A COL :

ALTER TABLE table_name

DROP COLUMN COLUMN_NAME ;

3.TO CHANGE THE DATATYPE:

ALTER TABLE table_name

MODIFY COLUMN_NAME new_datatype;

4.TO CHANGE THE NOT NULL CONSTRAINT:

ALTER TABLE table_name

MODIFY COLUMN_NAME existing_datatype NULL/NOTNULL;

5.TO RENAME THE COLUMN:

ALTER TABLE table_name

RENAME COLUMN current_name TO new_name;

6.TO MODIFY CONSTRAINTS:

a) ALTER TABLE table_name

ADD CONSTRAINT constraint_ref_name UNIQUE(column_name);

b) ALTER TABLE table_name

ADD CONSTRAINT constraint_ref_name CHECK(condition);

c) ALTER TABLE table_name

ADD CONSTRAINT constraint_ref_name PRIMARY KEY(column_name);

d) ALTER TABLE table_name

ADD CONSTRAINT constraint_ref_name FOREIGN KEY(column_name) REFERENCES parent_table_name (column_name);

7.TO DROP/DISABLE/ENABLE A CONSTRAINT:

ALTER TABLE table_name

DROP/DISABLE/ENABLE CONSTRAINT constraint_ref_name;

4. TRUNCATE:

SYNTAX: TRUNCATE TABLE table_name;

5. DROP:

SYNTAX: DROP TABLE table_name;

TO RECOVER THE TABLE:(only in oracle)

SYNTAX: FLASHBACK TABLE table_name
TO BEFORE DROP
[RENAME TO new_name] ;

TO DROP THE TABLE FROM RECYCLE BIN

SYNTAX: PURGE TABLE table_name;

II.DATA MANIPULATION LANGUAGE

1.INSERT:

SYNTAX 1: INSERT INTO table_name VALUES (V1,V2,....,Vn);

2: INSERT INTO table_name (COL1,COL2,....COLn)
VALUES(V1,V2,....,Vn);

Or

INSERT INTO table_name (COL1,COL2,....COLn)
VALUES(&COL1,&COL2,...&COLn)

3. INSERT INTO table_name
SELECT statement;

2.UPDATE:

SYNTAX: UPDATE table_name
SET COL1=V1,COL2=V2,.....,COLn=Vn
[WHERE <filter_condition>];

3.DELETE:

SYNTAX : DELETE
FROM table_name
[WHERE <filter_condition>];

III.TRANSACTION CONTROL LANGUAGE

1.COMMIT:

SYNTAX: COMMIT;

2.SAVEPOINT:

SYNTAX: SAVEPOINT savepoint_name;

3.ROLLBACK:

SYNTAX: ROLLBACK;

ROLLBACK TO SAVEPOINT

SYNTAX: ROLLBACK TO savepoint_name;

IV.DATA CONTROL LANGUAGE:

1.GRANT:

SYNTAX: GRANT sql_statement ON table_name
TO user_name;

2.REVOKE: :

SYNTAX: REVOKE sql_statement ON table_name
FROM user_name;

V.DATA QUERY LANGUAGE:

1.SELECT:

SELECT */[DISTINCT] column_name/Expression [ALIAS]

2.PROJECTION:

SYNTAX: SELECT */[DISTINCT] column_name/Expression [ALIAS]
FROM table_name ;

3.SELECTION:

SYNTAX: SELECT */[DISTINCT] column_name/Expression [ALIAS]
FROM table_name
WHERE <filter_condition> ;

4.JOIN

1.CARTESIAN JOIN/CROSS JOIN

SYNTAX:ANSI->

```
SELECT col_name
FROM table_name1 CROSS JOIN table_name2;
```

SYNTAX:ORACLE->

```
SELECT col_name
FROM table_name1, table_name2;
```

2.INNER JOIN/EQUI JOIN

SYNTAX:ANSI->

```
SELECT col_name
FROM table_name1 INNER JOIN table_name2
ON table_name1.col_name=table_name2.col_name;
```

SYNTAX:ORACLE->

```
SELECT col_name
FROM table_name1, table_name2
WHERE table_name1.col_name=table_name2.col_name;
```

3.OUTER JOIN

I. LEFT OUTER JOIN

SYNTAX:ANSI->

```
SELECT col_name
FROM table_name1 LEFT [OUTER] JOIN table_name2
ON table_name1.col_name=table_name2.col_name;
```

SYNTAX:ORACLE->

```
SELECT col_name
FROM table_name1, table_name2
WHERE table_name1.col_name=table_name2.col_name(+);
```

II. RIGHT OUTER JOIN

SYNTAX:ANSI->

```
SELECT col_name
FROM table_name1 RIGHT [OUTER] JOIN table_name2
ON table_name1.col_name=table_name2.col_name;
```

SYNTAX:ORACLE->

```
SELECT col_name
FROM table_name1, table_name2
WHERE table_name1.col_name(+)=table_name2.col_name;
```

III. FULL OUTER JOIN

SYNTAX:ANSI->

```
SELECT col_name
FROM table_name1 FULL  [OUTER] JOIN table_name2
ON table_name1.col_name=table_name2.col_name;
```

[NOTE: NO ORACLE SYNTAX FOR FULL OUTER JOIN]

4.SELF JOIN

SYNTAX:ANSI->

```
SELECT col_name
FROM table_name1  T1   JOIN table_name1  T2
ON T1.col_name=T2.col_name;
```

SYNTAX:ORACLE->

```
SELECT col_name
FROM table_name1  T1, table_name1  T2
WHERE T1.col_name=T2.col_name;
```

5.NATURAL JOIN

SYNTAX:ANSI->

```
SELECT col_name
FROM table_name1  NATURAL JOIN table_name2 ;
```

NOTE: NO ORACLE SYNTAX FOR NATURAL JOIN