(DML) Data Manipulation Language

- **DML defines a set of commands that are used to query and modify data within existing schema objects.**
- **COMMIT is not implicit, i.e. changes are not permanent till explicitly committed.**
- **DML statements consist of SELECT, INSERT, UPDATE and DELETE.**

SELECT Statement

1. Using arithmetic operators

```
SELECT ENAME, SAL, SAL+300
   FROM EMP;
```

2. Operator precedence

```
SELECT ENAME, SAL, 12*SAL+100
   FROM EMP;
```

```
SELECT ENAME, SAL, 12*(SAL+100)
   FROM EMP;
```

The basic operators in sql are * / + - from left to right

3. Using column aliases

```
SELECT ENAME "NAME", SAL*12 "ANNUAL SALARY"
   FROM EMP;
```

4. Concatenation operator

Printing name and job as one string as column name "Employees":

```
SELECT ENAME ||JOB "Employees"
   FROM EMP;
```

5. Using literal character string.

To print <name> IS A <job> as one string with column name "Employees"

```
SELECT ENAME || 'IS A' || JOB AS "Employee"
      FROM EMP;
```

6. To eliminate duplicate rows (DISTINCT operator)

```
SELECT DISTINCT DEPTNO
   FROM EMP;
```

7. Special comparison operator used in WHERE Clause

(a) **Between .... And** → **gives a range between two values, both inclusive.**

```
SELECT ENAME, SAL
FROM EMP
WHERE SAL BETWEEN 1000 AND 5000;
```

(b) In (list) – matches any of a list of values

```
SELECT EMPNO, ENAME, SAL, DEPTNO
FROM EMP
WHERE DEPTNO IN (D1,D4,D5);
```

(C) LIKE – MATCH A CHARACTER PATTERN

- **LIKE operator is used only with char and varchar to match a pattern**
- **% denotes zero or many characters**
- **_ (underscore) denotes one character**
- **Combination of % and _ can also be used**

(i) List the names of emps whose name starts with 'S'

(ii) List the names ending with 'S'

(iii) List names having 'o' as the second character.

(i)

```
SELECT ENAME
FROM EMP
WHERE ENAME LIKE 'S%';
```

(ii)

```
SELECT ENAME
FROM EMP
WHERE ENAME LIKE '%S';
```

(iii)

```
SELECT ENAME
FROM EMP
WHERE ENAME LIKE '_o%';
```

(d) IS NULL operator

To find the employees who does not belong to any department

```
SELECT ENAME, DEPTNO
FROM EMP
WHERE DEPTNO IS NULL;
```

8. Logical operators

Rules of precedence

1. **All comparison operators (>,>=,=,<,<=)**
2. **NOT**
3. **AND**
4. **OR**

To print those records of salesman or president who are having salary above 15,000/-

```
SELECT ENAME, JOB, SAL
FROM EMP
WHERE (JOB="Salesman" OR JOB="President")
      AND SAL>15000;
```

9. ORDER BY Clause

- **Used in the last portion of SELECT statement**
- **Rows will be sorted**
- **Default will be ascending**
- **Descending order sort can be done by using DESC**
- **Sort by column alias is also possible.**

```
SELECT EMPNO, ENAME, SAL*12 “ANNUAL SAL”
FROM EMP
ORDER BY “ANNUAL SAL”;

SELECT ENAME, DEPTNO, SAL
FROM EMP
ORDER  BY DEPTNO, SAL DESC;
```

10. Aggregate Function

- Some of these functions are COUNT, MIN, MAX, AVG
- These functions help in getting consolidated information from a group of tuples.

To find the total number of employees.

```
SELECT COUNT(*)
      FROM EMP;
```

To find the min, max and average salary of employees in dept D4.

```
SELECT MIN(SAL), MAX(SAL), AVG(SAL)
FROM EMP
WHERE DEPTNO = “D4”;
```

11. GROUP BY Clause

- Used to group database tuples on the basis of certain common attribute values eg. Employee of a department.
- We can use WHERE clause

To find the department number and number of employees working in the dept.

```
SELECT DEPTNO, COUNT(EMPNO)
FROM EMP
GROUP BY DEPTNO;
```

Note that while using GROUP BY and aggregate functions the only attributes that can be put in SELECT clause are the aggregate functions and the attributes that have been used for grouping the information.

12. HAVING Clause

- **This clause is used for creating conditions on grouped information**

Find Department number and max salary of those departments where max salary is more than Rs.10,000/-

```
SELECT DEPTNO, MAX(SAL)
FROM EMP
GROUP BY DEPTNO
HAVING MAX(SAL)>10000;
```

INSERT INTO Command

- **Values can be inserted for all columns or for selected columns**
- **Values can be given through sub-queries**
- **In place of values, parameter substitution can also be used with INSERT.**

Insert a new tuple in the EMP table (10,Sekhar, ,7000,D1)

```
INSERT INTO EMP
VALUES(10, Sekhar, , 7000, D1);
```

Insert a new tuple in the EMP table using parameter substitution.

```
INSERT INTO EMP
VALUES(&1, &2, NULL, &3, &4);
```

Note that these values need to be supplied at runtime.

Insert the EMPNO, an increment of 500, ENAME into a table INCR, for those employees of Dept "D3" FROM EMP table.

```
INSERT INTO INCR
    SELECT EMPNO, 500, ENAME, DEPTNO
    FROM EMP
    WHERE DEPTNO = "D3";
```

UPDATE command

Syntax:

```
UPDATE <table name>
SET <col name> = <value>
WHERE <condition>
```

Double the salary of the employees working in dept "D1"

```
UPDATE EMP
SET SAL = SAL*2
WHERE DEPTNO = "D1";
```

DELETE command

Delete the records of the employees whose salary is less than 4000.

```
DELETE FROM EMP
WHERE SAL<=4000;
```

Summary of the SELECT command

1. SELECT <Field Names>
2. FROM <Table name>
3. WHERE
    a. SAL BETWEEN 1000 AND 5000
    b. DEPT IN (D1,D4,D6)
    c. ENAME LIKE 'S%' ('%S' , '_I%, IS NOT NULL)
4. ORDER BY DEPTNO, SAL DESC
5. GROUP BY DEPTNO
6. HAVING MAX(SAL)>10000 → creates conditions on group functions
7. SELECT DISTINCT DEPTNO

Exercise

Consider the Supplier relations

S

SNO	SNAME	STATUS	CITY
S1	Prentice Hall	30	Calcutta
S2	McGraw Hill	30	Mumbai
S3	Willey	20	Chennai
S4	Pearson	40	Delhi
S5	Galgothia	10	Delhi

SNO → Supplier No.

PNO → Part No.

SP

SNO	PNO	QTY
S1	P1	300
S1	P2	200
S2	P1	100
S2	P2	400
S3	P2	200
S4	P2	200

Q1. Get supplier number for suppliers with satus>20 and City Delhi

Q2. Get Supplier name, supp number and status for suppliers in Delhi in Descending order of Status

Q3. Get all pairs of supp_no such that the two suppliers are located in the same city.

Q4. Get unique supplier names for suppliers who supply part P2 without using IN operator/using IN operator

Q5. Get supplier numbers for suppliers who are located in the same city as supplier S1

Q6. Suppose for the supplier S5, the value for status is NULL instead of 10. Get supplier numbers for suppliers greater than 25

Q7. Get unique supplier numbers supplying parts.

Q8. For each part supplied, get the part number and the total quantity supplied for that. (use GROUP, SUM(QTY))

Q9. Get part numbers for all parts supplied by more than one supplier (Hint: use GROUP with HAVING)

Q10. Double the status of all suppliers in Delhi (Hint: use UPDATE)