Structured Query Language / ISO 9075

**RDBMS : Relational Database Management System**

1. MS SQL Server
2. Oracle Database
3. MySQL
4. Maria DB
5. IBM DB2

**SQL**

1. DML : Data Manipulation Language
   1. UPDATE
   2. INSERT
   3. DELETE
2. DDL : Data Definition Language
   1. CREATE
   2. ALTER
   3. DROP
   4. TRUNCATE
3. DQL : Data Query Language
   1. SELECT
4. TCL : Transaction Control Language
   1. COMMIT
   2. ROLLBACK
5. DCL : Data Control Language
   1. GRANT
   2. REVOKE

**CONSTRAINT:**

1. NOT NULL
2. UNIQUE
3. PRIMARY KEY : It is a combination of not null & unique constraint and used to identify each record.
4. FOREIGN KEY : Prevent action that would destroy relationship between tables.
5. CHECK : Ensure that the value in a column satisfies a specific condition

alter table employee add constraint chk\_salary check(salary >0);

1. DEFAULT : Sets a default value for a column

CREATE INDEX

SELECT \* FROM TABLE\_NAME [WHERE CONDITION ]

[ ORDER By COL1, COL2 , COL\_N]

[ GROUP BY COL [HAVING CONDITION ]]

**SUB QUERIES**

SELECT \* FROM EMPLOYEE WHERE SALARY = (SELECT MAX(SALARY) FROM EMPLOYEE);

SELECT \* FROM EMPLOYEE WHERE SALARY IN (SELECT MAX(SALARY) FROM EMPLOYEE);

SELECT \* FROM EMPLOYEE WHERE SALARY IN(34220,2300);

SELECT \* FROM EMPLOYEE WHERE SALARY =34220 OR SALARY = 2300

SELECT \* FROM EMPLOYEE WHERE SALARY IN(SELECT MAX(SALARY) FROM EMPLOYEE GROUP BY DEPT\_ID);

**JOIN QUERY**

**INNER JOIN :** Fetch all records from both tables that match.

SELECT EMP\_ID, EMP\_NAME, SALARY, EMPLOYEE.DEPT\_ID, DEPT\_NAME, DEPT\_LOCATION FROM EMPLOYEE INNER JOIN department ON EMPLOYEE.DEPT\_ID = DEPARTMENT.DEPT\_ID;

**With Alias**

SELECT EMP\_ID, EMP\_NAME, SALARY, e.DEPT\_ID, DEPT\_NAME, DEPT\_LOCATION FROM EMPLOYEE e JOIN department d ON e.DEPT\_ID = d.DEPT\_ID;

**LEFT OUTER JOIN :** Fetch All the records from Left Side Table and only matching records from right-side table

SELECT EMP\_ID, EMP\_NAME, SALARY, e.DEPT\_ID, DEPT\_NAME, DEPT\_LOCATION FROM EMPLOYEE e LEFT OUTER JOIN department d ON e.DEPT\_ID = d.DEPT\_ID;

**RIGHT OUTER JOIN :** Fetch All the records from Right Side Table

SELECT EMP\_ID, EMP\_NAME, SALARY, e.DEPT\_ID, DEPT\_NAME, DEPT\_LOCATION FROM EMPLOYEE e RIGHT OUTER JOIN department d ON e.DEPT\_ID = d.DEPT\_ID;

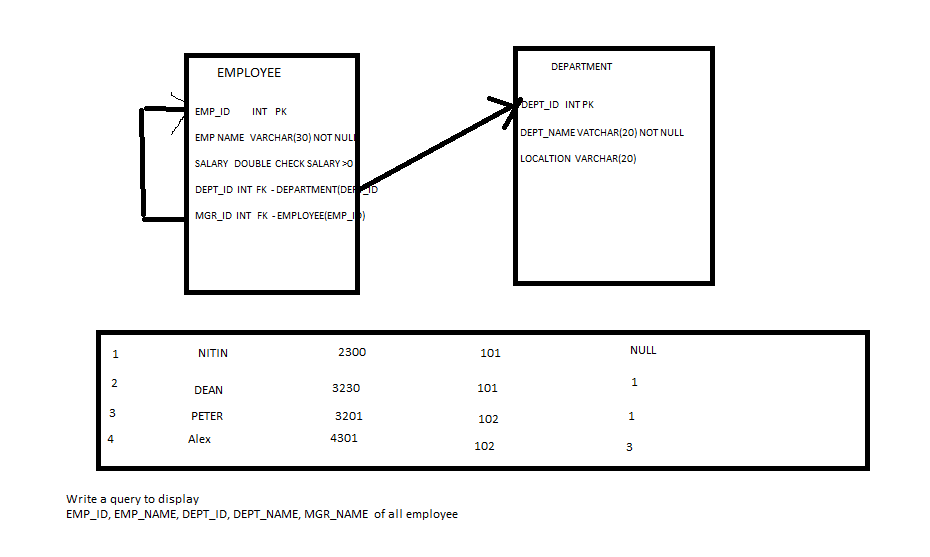
**SELECT EMP\_NAME , SALARY, SALARY \* 0.20 AS 'TAX\_ON\_SALARY' FROM practicedb.employee;**

**VIEW:** IT IS VIRTUAL TABLE THAT HOLDS THE RESULT-SET OF AN SQL QUERY.

CREATE VIEW EMP\_SALARY\_TAX AS

SELECT EMP\_NAME , SALARY, SALARY \* 0.20 AS 'TAX\_ON\_SALARY' FROM practicedb.employee;

SELECT \* FROM EMP\_SALARY\_TAX;

****

**FUNCTION**

**PROCEDURE**