Data

A Data is a Rawfact which describes the Attributes of an Entity

Rawfact : Unchanged fact / Original Truth

Attributes : Properties / Fields

Entity : Object

Example :

Entity/Object

Properties	Data / values
Name :	Stark
Age :	12
Gender :	Male
Blood group:	A +ve

Database

A Database is a place / medium which is used to store the Data in a Systematic and Organized Manner

Example :

Existence of Database in Real-Time Usage - Users

Online Shopping: Amazon, flipkart , ...

Banking : Transactions, Balance, ...

Mobile Applications

Food Delivery

Existence of Database in Real-Time Usage - Business Domains

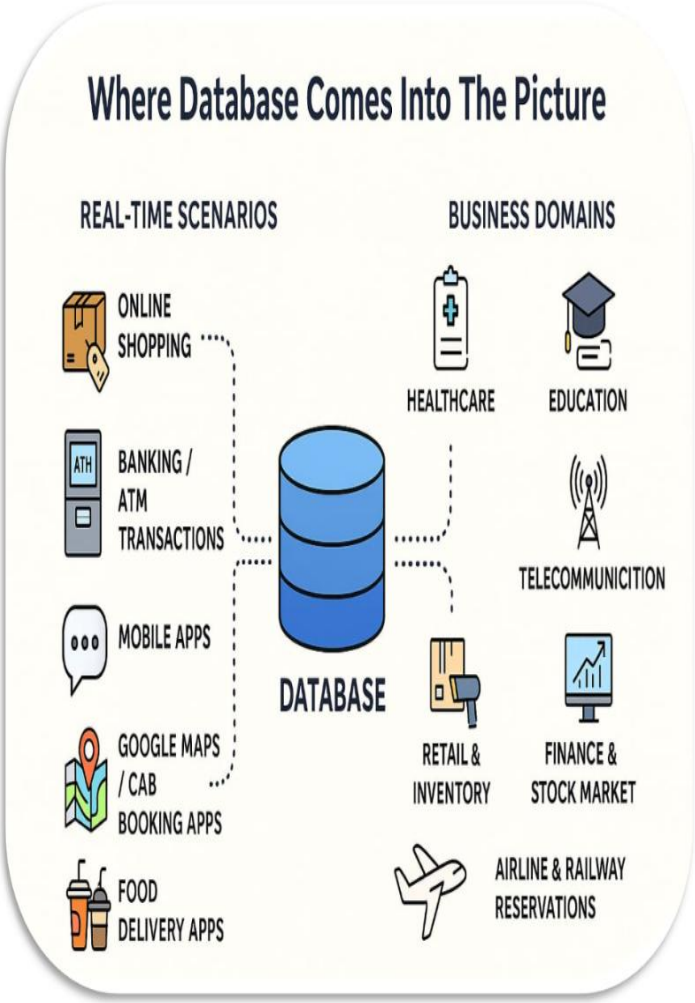
Healthcare

Education

Telecommunication

Retail & Inventory

Finance



Database → Data → Store, Fetch / Retrieve, Update, ...

CRUD Operation

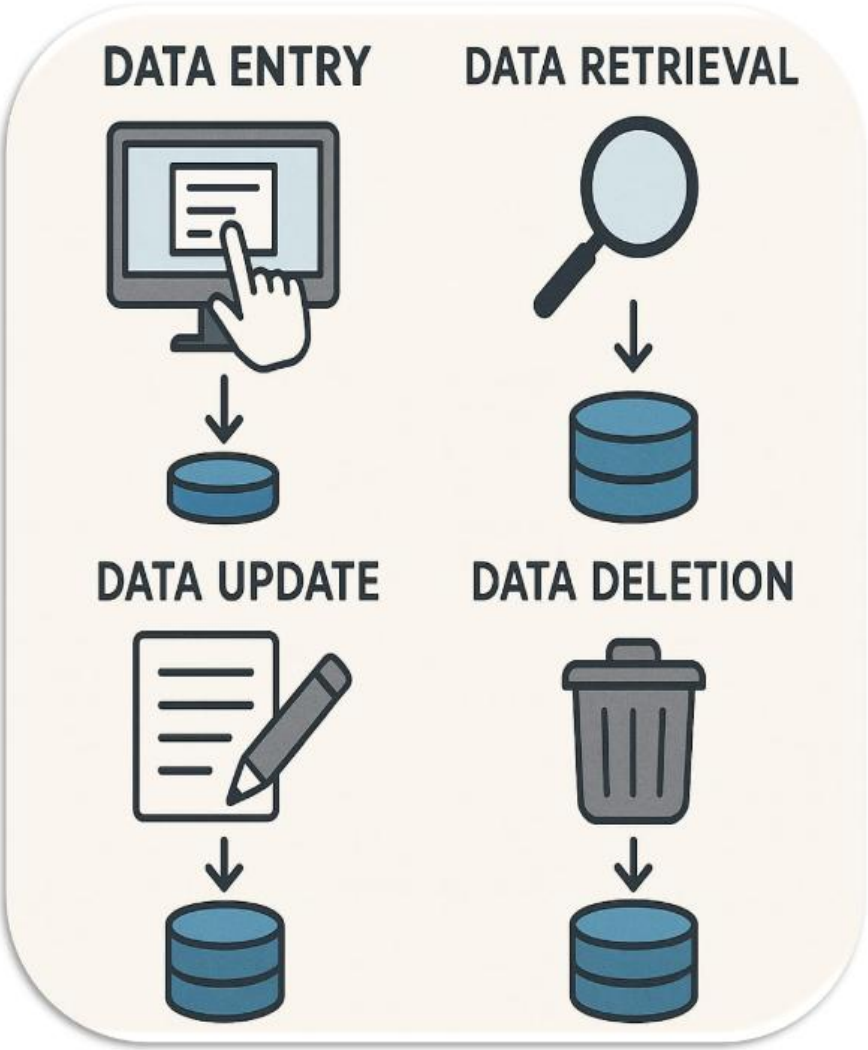
The basic operation performed on the Database is known as "CRUD" Operation

C - Create / Insert

R - Read / Retrieve

U - Update / Modify

D - Delete / Drop



DBMS (Database Management System)

It is a Software which is used to Maintain and Manage the Database.

Characteristics of DBMS :

> DBMS provides two Main / Key Features:

1. Security

Here, Security --> Keeping data safe from unauthorized people.

- Security means protecting the data from people who should not see or change it.
- It keeps the information safe — just like you keep your school bag or phone password safe
- Example: -
  - Imagine your school has a computer where all student marks are stored.
  - Only the Principal and teachers can open it.
  - If someone else (like another student) tries to see it — they can't because of Security.

2. Authorization

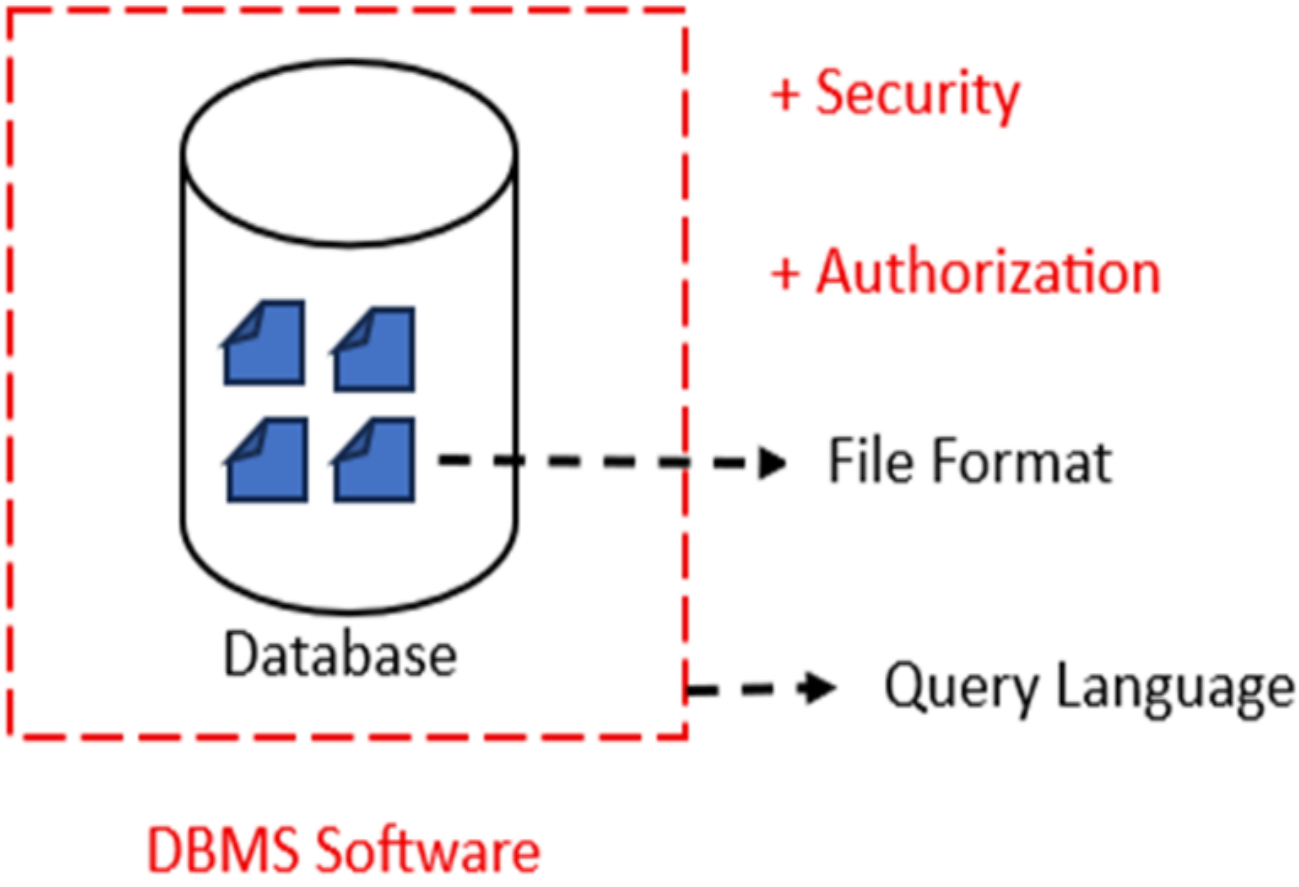
Here, Authorization --> Giving permission to the right person to use data.

- Authorization means deciding who can do what with the data.
- It gives permission to people to use the data in different ways.
- Example: - In the same school computer:
  - The Principal can see and change all marks.
  - The Teacher can only change marks for their own class.
  - The Student can only see their own marks.

So, everyone has different permissions — this is called Authorization.

> In DBMS Data is Stored in File Format

> We have Query Language To communicate with our DBMS



Types of DBMS:

1. Hierarchical DBMS
2. Network DBMS
3. Object-Oriented DBMS
4. RDBMS

→ Hierarchical DBMS:

Data is Organized in a Tree-like Structure

Positive : Fast Data Retrieval

Negative : Modifying is Complex

→ Network DBMS:

Data is organized in a link where it is connected to too many complex structures

Positive : For complex relationships

Negative : Too much Complex structures

→ Object- Oriented DBMS:

Data is organized in a manner which is related to Object oriented Programmings

Positive : Object oriented programs

Negative : other than Object oriented programs it is too complex

RDBMS : (Relational Database Management System)

It is a type of DBMS software which is used to maintain and manage the Database

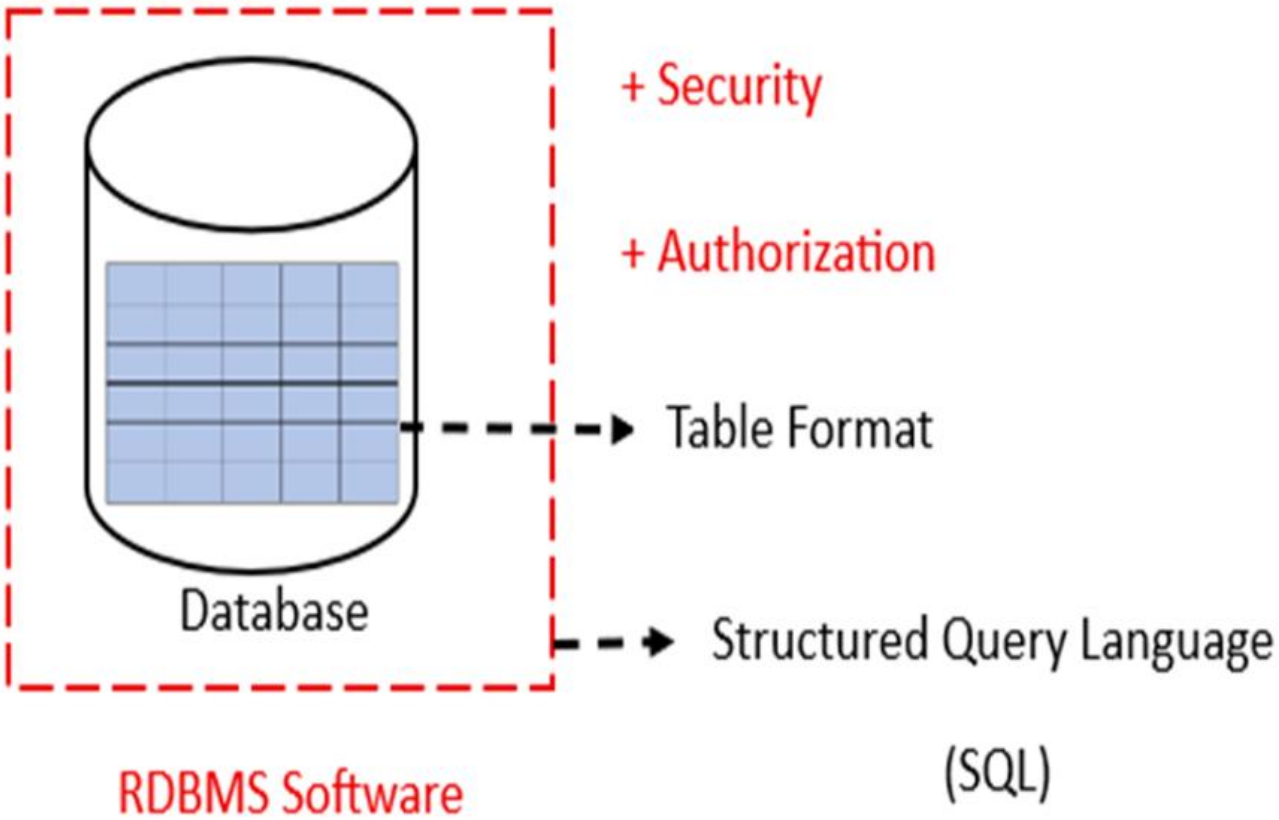
Characteristics of RDBMS:

> RDBMS provides two main / key features:

1. Security
2. Authorization

> In RDBMS, data is stored in the form of Table Format

> We use Structured Query Language to communicate with database through RDBMS



Relational Model

> It is a model where Data is stored in the form of Rows and Columns --> Tables

> It is a Model Discovered by a Data Scientist Edgar Frank Codd (E.F.Codd)

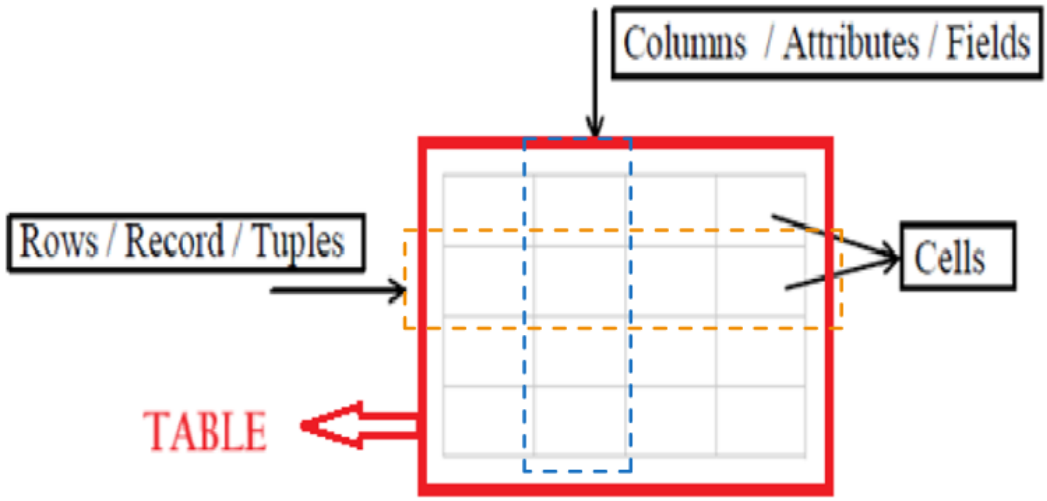
> Any DBMS which follows the Relational Model it will become RDBMS

DBMS —Relational model→ RDBMS

Tables -> Combination of Rows and Columns (or)
The Logical Organization of Rows and Columns

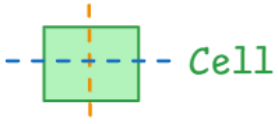
Rows -> The Horizontal entities in a Table are called as "Rows" or Records or Tuples

Columns -> The Vertical entities in a Table are called as "Columns" or Fields or Attributes or Properties



Cell :

It is the smallest unit (or)
The Intersection of Rows and Columns creates a space and that space is called a Cell



DBMS vs RDBMS

DBMS	RDBMS
> In DBMS, Data is stored in File Format	> In RDBMS, Data is stored in Table Format
> We use Query Language to communicate with DBMS	> We use Structured Query Language to communicate with RDBMS
> Compared to RDBMS, it is difficult to perform CRUD operation in DBMS	> Compared to DBMS, it is more easy to perform CRUD operation in RDBMS
Example :	Example :
> Operation for searching Stark's Phone no	> Operation for searching Stark's Phone no
Open Student details file ↓ Search Each file by Stark's name till you get ↓ Open Stark's file ↓ Search Phone no ↓ output	Open Student details table ↓ Search Stark by s.no ↓ Output

Rules of E.F.Codd :

Rule - 1: A Data entered into a Cell must be a Single Valued Data

SID	SName	Ph_No
1	Stark	1234567890
2	Steve	0123654987
3	Natasha	9876543210, 9988776655
4	Bruce	8899774422

Not Suitable to Perform CRUD operation

SID	SName	Ph_No	Alt_Ph_NO
1	Stark	1234567890	
2	Steve	0123654987	
3	Natasha	9876543210	9988776655
4	Bruce	8899774422	

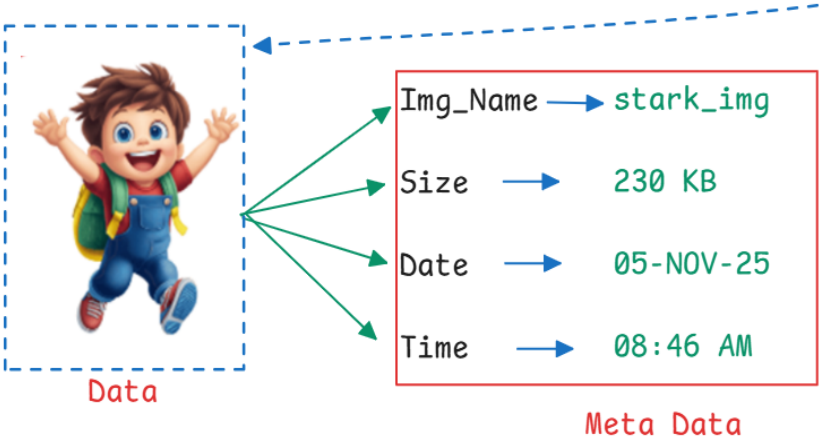
Suitable to Perform CRUD operation

Rule - 2: In RDBMS, we can store everything in the form of Tables, including Meta Data

Everything (any format) → Char, Num, audios, Videos, .txt,.pdf, .jpg, ...

Meta Data → Further Information about a Data

SID	SName	Ph_No	Image
1	Stark	1234567890	.jpg
2	Steve	0123654987	.jpg
3	Natasha	9876543210	.jpg
4	Bruce	8899774422	.jpg



Meta Table → The table where we store the Meta Data

Meta Table

img_Name	Size	Date	Time
stark_ img	230 KB	05-NOV-25	08:46 AM

Auto - Generated

Compiler

Rule - 3: According to E.F.Codd, we can store Data in Multiple Tables and if necessary we can establish a connection between the Tables with the help of Key Attributes

---

Students Table

SID	SName	Class
1	Stark	09th
2	Steve	10th
3	Natasha	10th
4	Bruce	12th

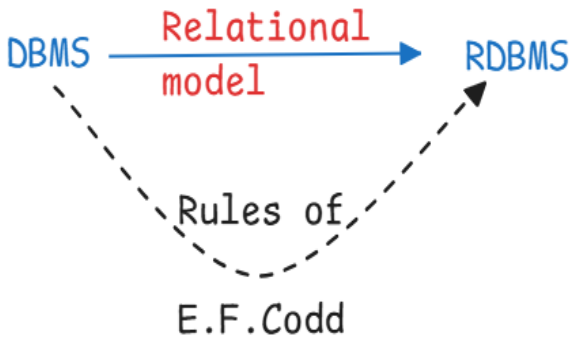
Connection ?

Teachers Table

TID	TName
T1	Berlin
T2	Walter
T3	Charles
T4	Walter

Rule - 4: Data Entered into a Table must be validated in two steps:

1. By assigning Datatypes
2. By assigning Constraints



Datatypes:

Datatype is used to Specify What type of Data or Which type of Data to be stored in a particular Memory Location or in a Particular Column

Types of Datatypes :

1. CHAR
2. VARCHAR / VARCHAR2
3. DATE
4. NUMBER
5. LARGE OBJECTS

CHAR Datatype :

> CHAR Datatype is used to store characters such as 'A-Z'; 'a-z'; '0-9'; Special Characters like '@,!,#,$,%,^,&,*,...'; Alphanumeric Characters

Characteristics :

> Always Characters should be enclosed within Single Quotes (' ')

> Whenever we are using CHAR Datatype we need to mention the Size of it

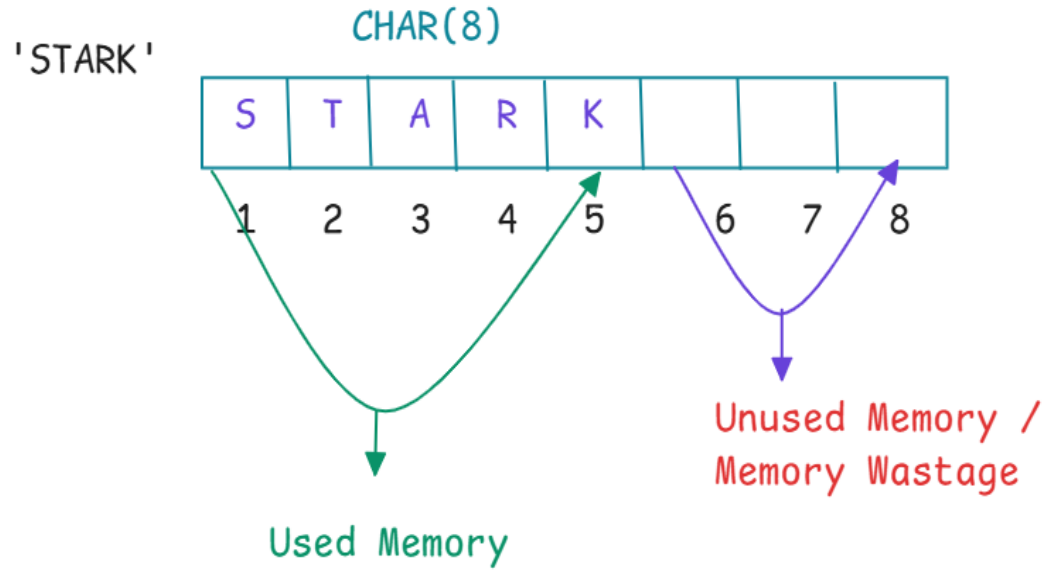
> Syntax :

```
CHAR(Size)
```

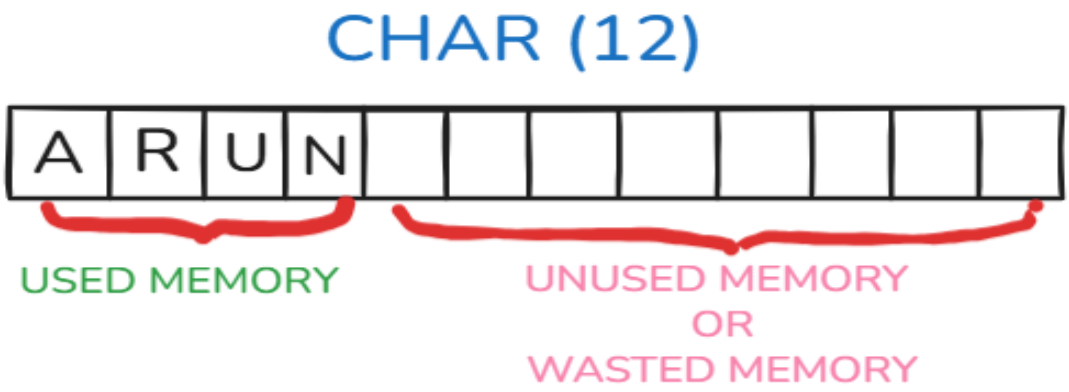
> The Maximum Number of Characters that can be stored is up to 2000 Characters

> CHAR Datatype always follows "FIXED LENGTH MEMORY ALLOCATION"

Example :



- ***Example – 1 for CHAR Datatype*: -**



- ***Example – 2 for CHAR Datatype*: -**

CHAR(7) CHAR(10)

Eld Number	PAN card
Eld_001	ABGPS1234L
Eld_002	ABIPS4321K
Eld_003	ABESD1234S
Eld_004	ABEPS4321H
Eld_005	ABGST1234N

VARCHAR Datatype :

> VARCHAR Datatype is used to store characters such as 'A-Z'; 'a-z'; '0-9'; Special Characters like '@,!,#,$,%,^,&,*,...'; Alphanumeric Characters

Characteristics :

> Always Characters should be enclosed within Single Quotes (' ')

> Whenever we are using VARCHAR Datatype we need to mention the Size of it

> Syntax :

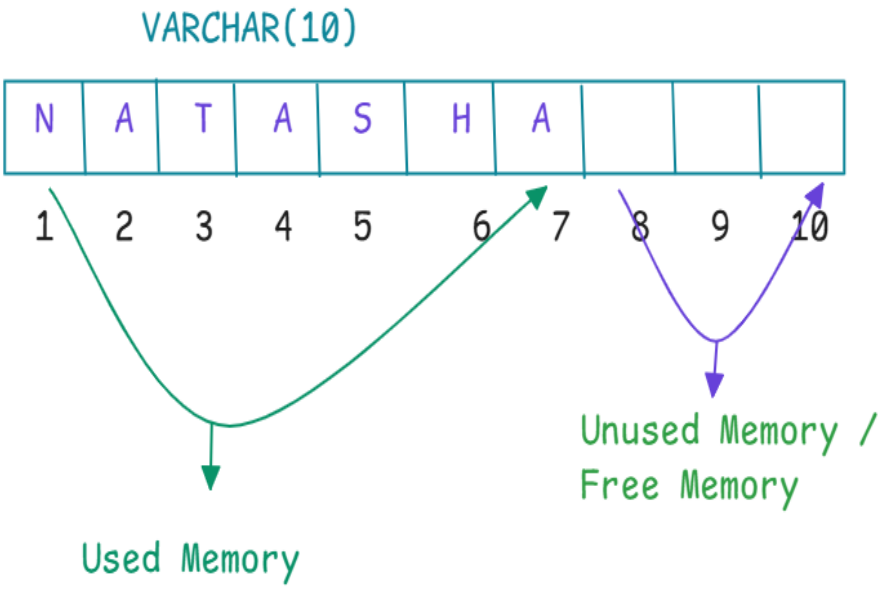
```
VARCHAR(Size)
```

> The Maximum Number of Characters that can be stored is up to 2000 Characters

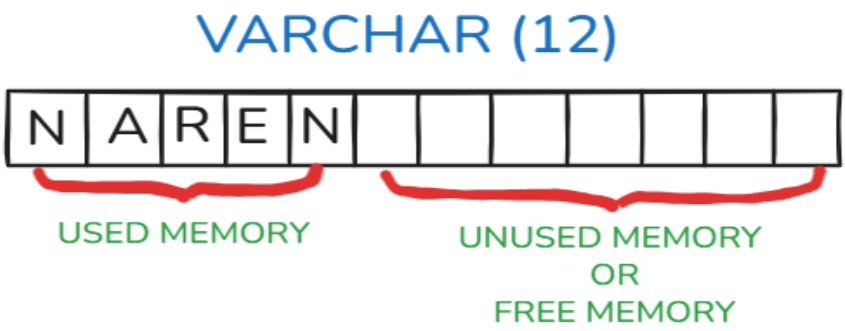
> VARCHAR Datatype always follows "VARIABLE LENGTH MEMORY ALLOCATION"

Example :

'NATASHA'



- ***Example – 1 for VARCHAR Datatype*: -**



- ***Example – 2 for VARCHAR Datatype*: -**

VARCHAR(10) VARCHAR(30) VARCHAR(15)

EMP NAME	MAIL ID	GENDER
Raja	raja_king@gmail.com	Male
Rani	rani_queen@gmail.com	Female
Mini_1	mini_minister.edu@gmail.com	Female
Mini_2	mini_minister.home@gmail.com	Male
Mini_3	mini_minister.agri@gmail.com	Male

VARCHAR2 Datatype :

> It is the Updated Version of VARCHAR Datatype
> It is automatically considered as VARCHAR2 even if you are mentioning VARCHAR

Characteristics :

> The Maximum Number of Characters that can be stored is up to 4000 Characters

CHAR Vs VARCHAR :

CHAR Datatype	VARCHAR Datatype
> CHAR Datatype always Follows "Fixed Length Memory allocation" >Here the Unused Memory gets Wasted and it will not return Back to Memory Location > Example : Aadhar Card, PAN, Roll Number, ...	> VARCHAR Datatype always Follows "Variable Length Memory allocation" >Here the Unused Memory gets Wasted and it will return Back to Memory Location > Example : Name, EMail, Job Role, Address, Password, ...

DATE Datatype :

> DATE Datatype is used to store the Dates in Specific Format called as "ORACLE FORMAT"

>ORACLE FORMAT : 'DD-MON-YYYY' OR 'DD-MON-YY'

> EX:- '07-11-2025' '07-11-25' ✗ '7-FEB-1992' ✗

'07-NOV-2025' '07-NOV-25' '07-FEB-1992' ✓

> SYNTAX : DATE

> EX:- DOB DATE

Date_of_Birth	Date_of_Joining	Transaction_Date
DATE	DATE	DATE
DD-MON-YY	DD-MON-YYYY	DD-MON-YYYY
10-Jan-87	04-Aug-2016	03-Dec-2019
06-Feb-92	20-Nov-2020	26-Jun-2021
21-Jul-98	05-May-2023	15-Oct-2024

NUMBER Datatype :

> NUMBER Datatype is used to store the Numeric values or Numbers

> syntax : NUMBER(PRECISION[,SCALE])

PRECISION:

> Precision is used to determine the number of digits to store integer values

> The range of Precision is 1 to 38

SCALE:

> Scale is used to determine the number of digits to store the decimal values within Precision

> The range of Scale is -84 to 127

> The default value of Scale is 0

Examples :

case 1: P > S

NUMBER(7,2) ⟶ +/- 9 9 9 9 9.9 9

P S

NUMBER(6) ⟶ +/- 9 9 9 9 9 9

NUMBER(8,4) ⟶ +/- 9 9 9 9.9 9 9 9 ⟶ 1,1.1111,1.1112,....9999.9999

NUMBER(-1) ✗ --> ERROR

NUMBER(39) ✗ --> ERROR

case 2: P = S

NUMBER(4,4) ⟶ +/- .9 9 9 9

NUMBER(7,7) ⟶ +/- .9 9 9 9 9 9 9

case 3: P < S

NUMBER(2,4) ⟶ +/- .0 0 9 9

NUMBER(1,4) ⟶ +/- .0 0 0 9 ✓ +/- .0 0 9 0 ✗

NUMBER(0,4) ⟶ +/- .0 0 0 0 ✗

NUMBER(3,7) ⟶ +/- .0 0 0 0 9 9 9

EX: 1. SALARY

2. PHONE NUMBER

3. PRICE

4. BANK ACCOUNT NUMBER

5. AND SO ON,...

Precision

Example : Number (4)	+/- 9 9 9 9
Example : Number (6)	+/- 9 9 9 9 9 9
Example : Number (2 , 0) P S	+/- 9 9

SCALE

Example : Number (4 ,2)	+/- 99.99
Example : Number (7 ,2)	+/- 99999.99
Example : Number (3 ,3)	+/- .999
Example : Number (2 ,0)	+/- 99
Example : Number (3 ,5)	+/- .00999
Example : Number (4 ,8) P S	+/- .00009999

LARGE OBJECTS Datatype :

> LARGE OBJECTS Datatypes is categorized as
1. CHARACTER LARGE OBJECT
2. BINARY LARGE OBJECT

CHARACTER LARGE OBJECT (CLOB) :

2000 BYTES

4000 BYTES

VARCHAR(4000) = '4000CHARACTERS' ✓

VARCHAR(4001) = ---ERROR--- ✗

> Character Large Objects Datatype is used to store a large / huge amount of Characters up to 4GB

> Example: Files stored as .txt, .html, .xml,...

> Syntax : CLOB

BINARY LARGE OBJECT (BLOB) :

> Binary Large Objects Datatype is used to store Binary values of Images, Documents, videos, audios,etc... up to 4GB

> Example: Files stored as .jpg, .pdf, .gif, .mp3, .mp4,...

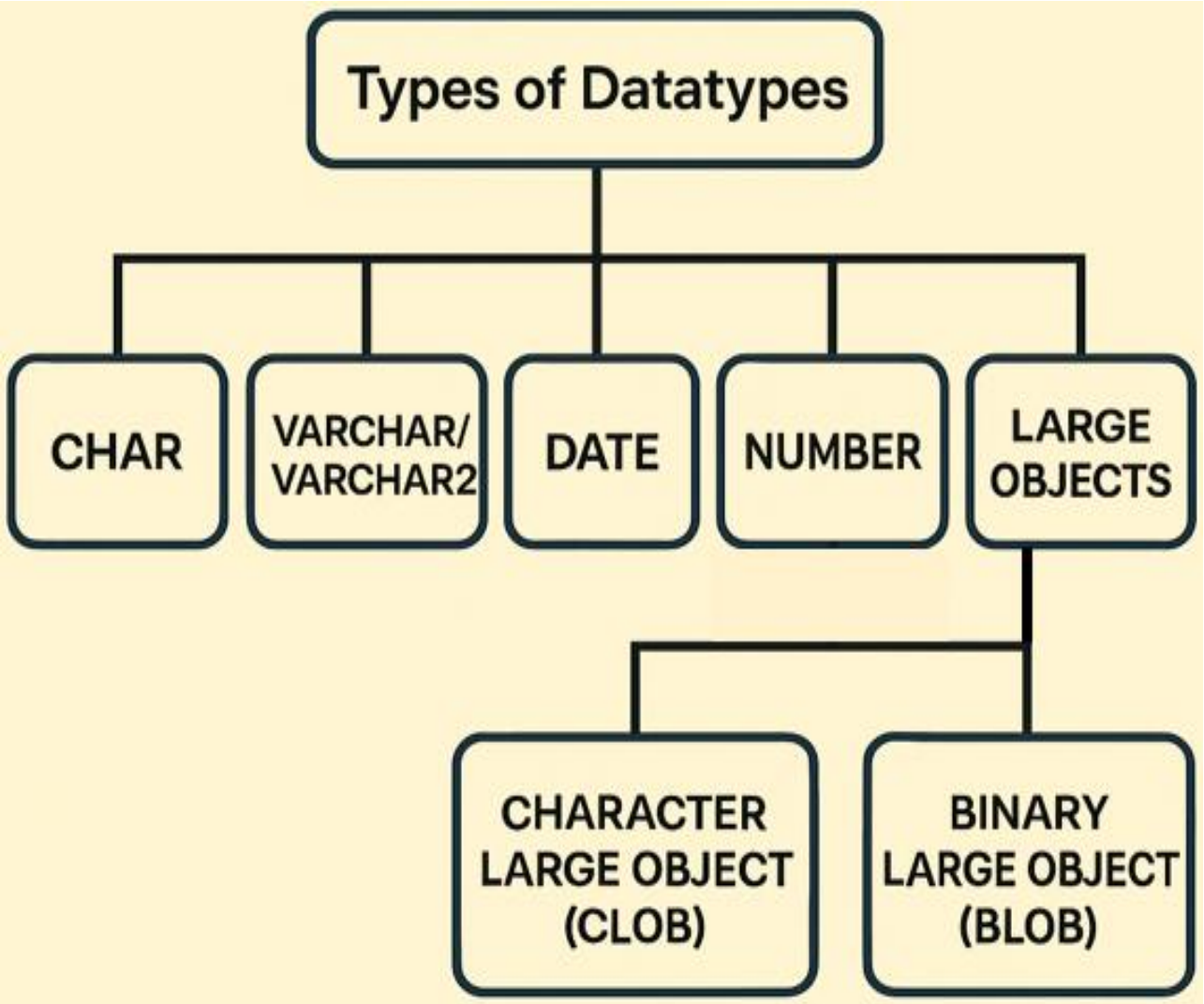
> Syntax : BLOB

NOTES :

*** DATATYPES are Mandatory if we create any table

--> CONSTRAINTS are Optional

****** SQL is CASE INSENSITIVE or Not CASE SENSITIVE
BUT LITERALS / VALUES / DATA are CASE SENSITIVE ******



EID	EName	Salary	Ph_No
NUMBER(3)	VARCHAR(15)	NUMBER(6,2)	NUMBER(10)
1	Stark	2000.12	1234567890
1	Steve		21369
3123	Natasha	10000	-3123023698
-42	Bruce	-6000	-4123652

Constraints are the Rules given to a Column to validate the Data

Types of Constraints:

1. UNIQUE
2. NOT NULL
3. CHECK
4. PRIMARY KEY
5. FOREIGN KEY

UNIQUE :

> UNIQUE Constraint is assigned to a Column which should not accept Duplicated or Repeated Values

> Syntax : UNIQUE

EID	EName	Salary	Ph_No
NUMBER(3)	VARCHAR(15)	NUMBER(6,2)	NUMBER(10)
UNIQUE			UNIQUE
1	Stark	2000.12	1234567890
2	Steve		1234567891
3123	Natasha	10000	-3123023698

NOT NULL :

> NOT NULL Constraint is assigned to a Column which should not accept NULL or BLANK Values or it avoids Null in a Column

> Syntax : NOT NULL

EID	EName	Salary	Ph_No
NUMBER(3)	VARCHAR(15)	NUMBER(6,2)	NUMBER(10)
NOT NULL	NOT NULL	NOT NULL	NOT NULL
1	Stark	2000.12	3210654987
2	Steve	2000.12	123456789
3123	Natasha	0	-3123023698

CHECK :

> CHECK is a Constraint assigned to a Column which provides extra validation with a condition, if the condition is satisfied then the value is accepted else it will be rejected

> Syntax : CHECK(CONDITION)

Examples > CASE-1 (EID) :

CHECK(LENGTH(COL_NAME)=SIZE)

> CASE-2 (Salary) :

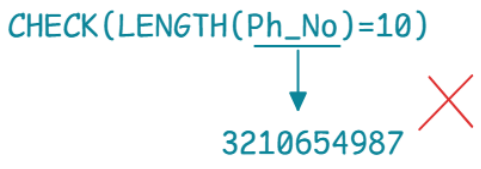
CHECK(COL_NAME > 0)

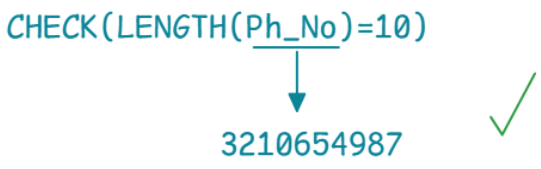
> CASE-3 (Ph_No) :

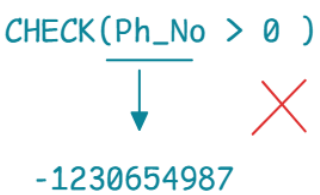
CHECK(LENGTH(COL_NAME)=SIZE)

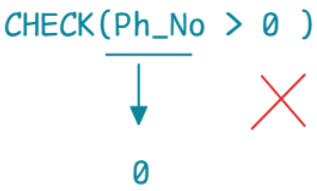
CHECK(COL_NAME > 0)

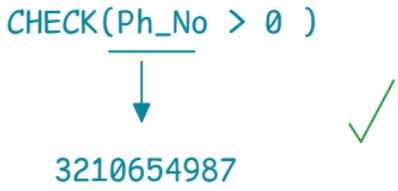
EID	EName	Salary	Ph_No
NUMBER(3)	VARCHAR(15)	NUMBER(6,2)	NUMBER(10)
CHECK(LENGTH (EID)=3)		CHECK(Salary>0)	CHECK(LENGTH(Ph_No)=10) CHECK(Ph_No>0)
123	Stark	2000.12	3210654987
210	Steve	2000.12	1234567895
312	Natasha	5000.25	3123023698











PRIMARY KEY :

> PRIMARY KEY is a Constraint which is used to identify the records Uniquely from the Table

> Syntax : PRIMARY KEY

Characteristics of Primary Key :

> Primary Key Cannot accept Duplicated or Repeated Values

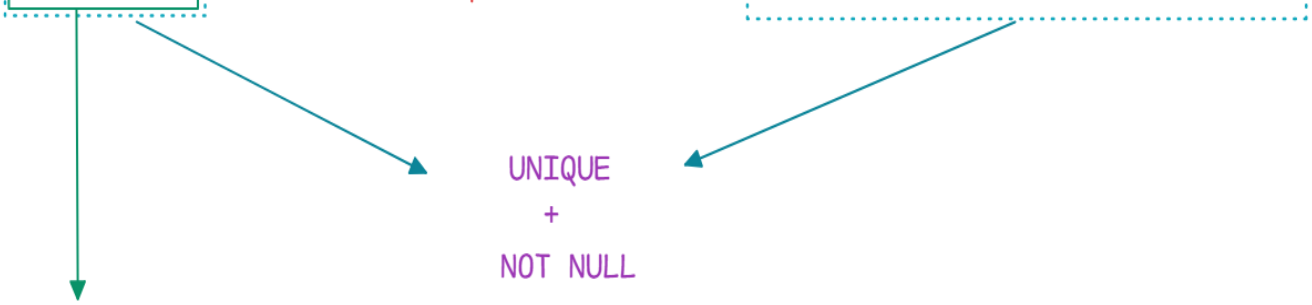
> Primary Key Cannot accept Null values

> Primary Key is always a Combination of Unique and Not Null Constraints

> We can have only one Primary Key in a Table

EMPLOYEES

EID	EName	Salary	Ph_No	PAN_No
NUMBER(3)	VARCHAR(15)	NUMBER(6,2)	NUMBER(10)	VARCHAR(15)
UNIQUE			UNIQUE	UNIQUE
NOT NULL	NOT NULL	NOT NULL	NOT NULL	NOT NULL
123	Stark	2000.12	3210654987	ABCD32154T
210	Steve	2000.12	1234567895	BERL03121G
312	Natasha	5000.25	3123023698	BLAC34527W



AVENGERS

AVENGER_ID	NAME	GF_NAME	PH_NO	ACC_NO
5821	TONY STARK	PEPPER POTTS	9988771122	9990006001
1945	STEVE ROGERS	AGENT CARTER	1245637890	9970000001
3221	NATASHA ROMANOFF	CLINT BARTON	1234567890	9980000007
1998	PETER PARKER		9874563210	9990107001
2008	BRUCE BANNER	BETTY ROSS	3523697410	9992006041
2012	PETER PARKER	GWENN STACY	7896541230	9997004032

UNIQUE NOT NULL — NOT NULL — UNIQUE — UNIQUE NOT NULL — UNIQUE NOT NULL

PRIMARY KEY

→ ALL THREE COLUMNS (HIGLIGHTED IN GREEN BOX) ARE ELIGIBLE FOR PRIMARY KEY

→ BUT AS PER PRIMARY KEY CHARACTERISTIC ONLY 1 PK CAN BE GIVEN

→ WITH ID WE CAN EASILY IDENTIFY ALL THE RECORDS UNIQUELY EVEN IF PH_NO IS CHANGED OR BANK ACCOUNT IS CLOSED

FOREIGN KEY :

> FOREIGN KEY is a Constraint which is used to Establish a Connection between the Tables

> Syntax : FOREIGN KEY

Characteristics of Foreign Key :

> Foreign Key Can accept Duplicated or Repeated Values

> Foreign Key Can accept Null values

> Foreign Key is not a Combination of Unique and Not Null Constraints

> We can have more than one Foreign Key in a Table

> Foreign Key is also known as "Referential Integrity Constraint"

> For an Attribute(Column) to become a Foreign Key in a Table(CHILD) it must be a Primary Key in its own Table(PARENT)

EMPLOYEES

EID	EName	Salary	Ph_No	Dept_No
NUMBER(3)	VARCHAR(15)	NUMBER(6,2)	NUMBER(10)	NUMBER(2)
			UNIQUE	
PRIMARY KEY	NOT NULL	NOT NULL	NOT NULL	
123	Stark	2000.12	3210654987	10
210	Steve	2000.12	1234567895	20
312	Natasha	5000.25	3123023698	30
187	Peter	2000.12	7788994455	
230	Bruce	2000.12	9988665544	20
364	Thor	5000.25	3215478961	30

FOREIGN KEY

CHILD TABLE

DEPARTMENT

Dept_No	Team_Lead_Name	Dept_Name	LOC
10	Howard	Research	New York
20	Nick Fury	Analyst	Germany
30	Barton	HR	Texas

PARENT TABLE

PRIMARY KEY

PRIMARY KEY

AVENGERS

AVENGER_ID	NAME	GF_NAME
5821	TONY STARK	PEPPER POTTS
1945	STEVE ROGERS	AGENT CARTER
3221	NATASHA ROMANOFF	CLINT BARTON
1998	PETER PARKER	MARY JANE
2008	BRUCE BANNER	BETTY ROSS
2011	THOR	JANE FOSTER

FOREIGN KEY

POWERS

AVENGER_ID	TAG_NAME	WEAPON
1998	SPIDER MAN	WEB SHOOTERS
3221	BLACK WIDOW	MIXED MARTIAL ARTS
2011	ASGARDIAN	STORM BREAKER
1945	CAPTAIN AMERICA	MJOLNIR
1945	CAPTAIN AMERICA	SHIELD
2011	GOD OF THUNDER	MJOLNIR

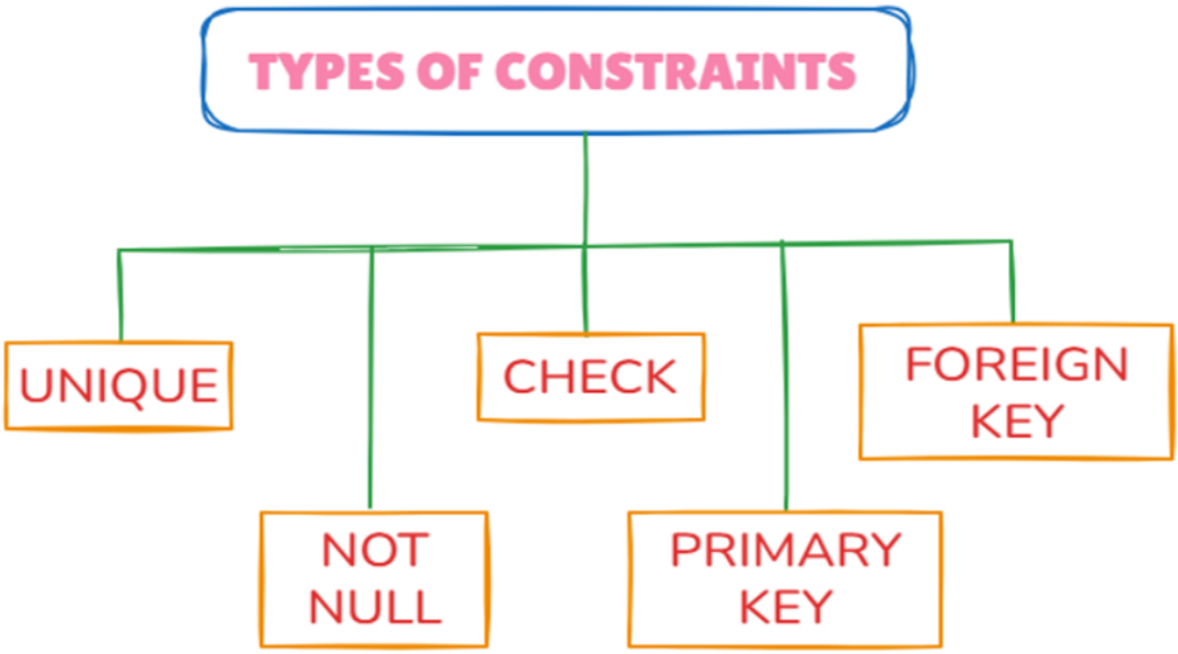
PRIMARY KEY Vs FOREIGN KEY :

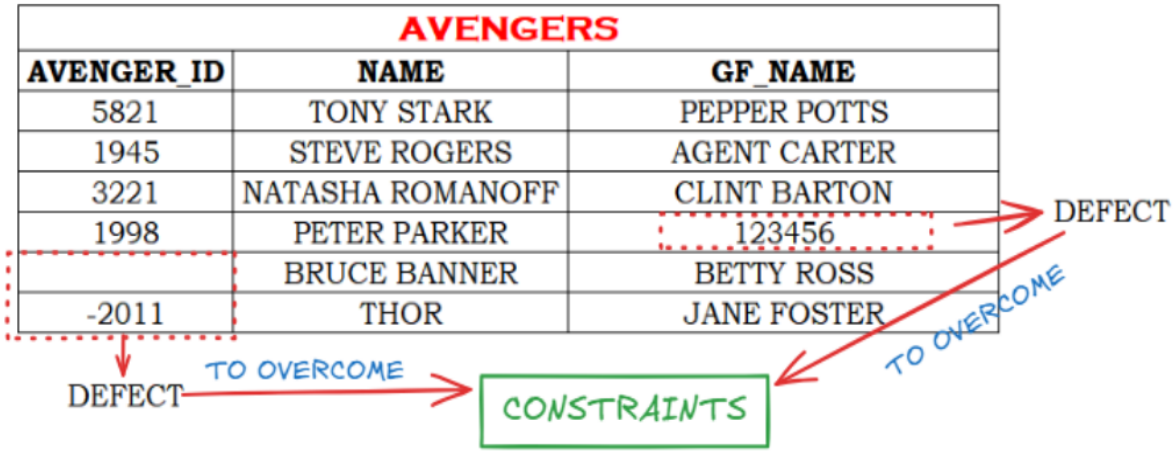
PRIMARY KEY	FOREIGN KEY
> Primary Key is constraint which is used to identify the records uniquely in a Table	> Foreign Key is constraint which is used to establish a connection between the Tables
> It cannot accept Duplicated or Repeated Values	> It can accept Duplicated or Repeated Values
> It cannot accept Null Values	> It can accept Null Values
> It is always a Combination of Unique and Not Null Constraints	> It is not a Combination of Unique and Not Null Constraints
> We can have only one Primary Key in a Table	> We can have more than one Foreign Key in a Table

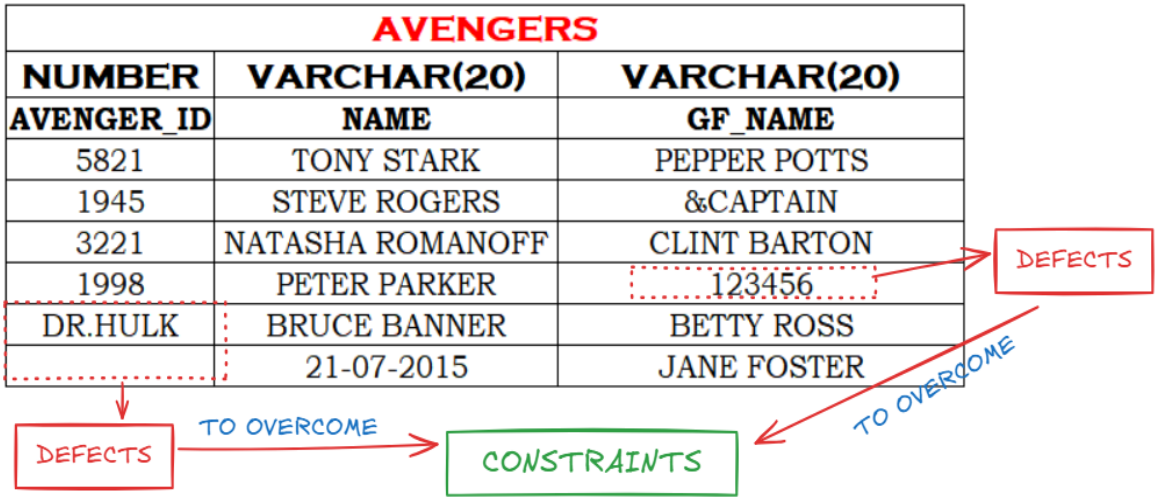
PRIMARY KEY Vs UNIQUE CONSTRAINT :

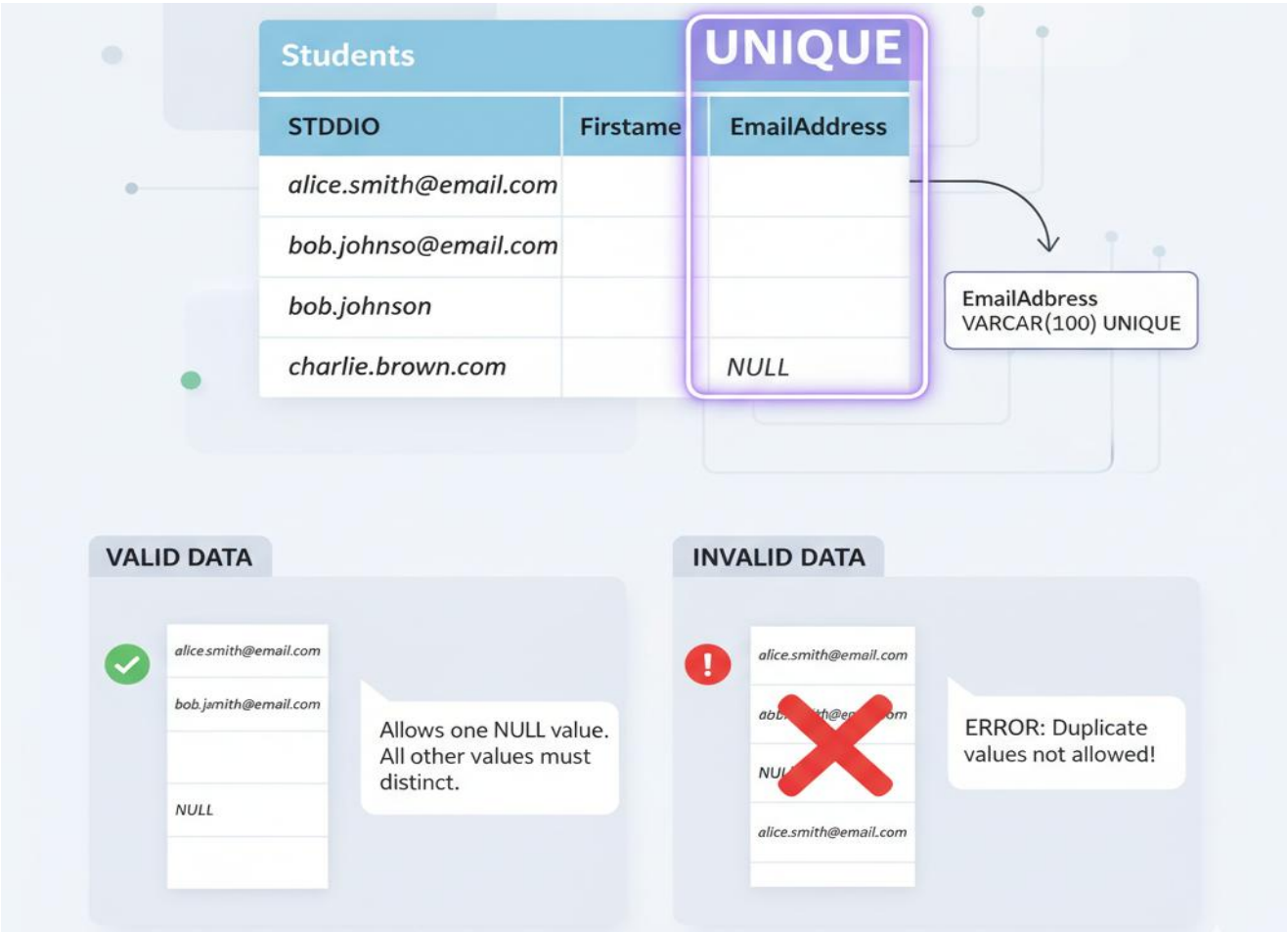
PRIMARY KEY	UNIQUE CONSTRAINT
> Primary Key is constraint which is used to identify the records uniquely in a Table	> UNIQUE Constraint is assigned to a Column which should not accept Duplicated or Repeated Values
> It cannot accept Null Values	> It can accept atleast one Null Value
> We can have only one column as a Primary Key in a Table	> It will accept more than one column as an Unique Constraint

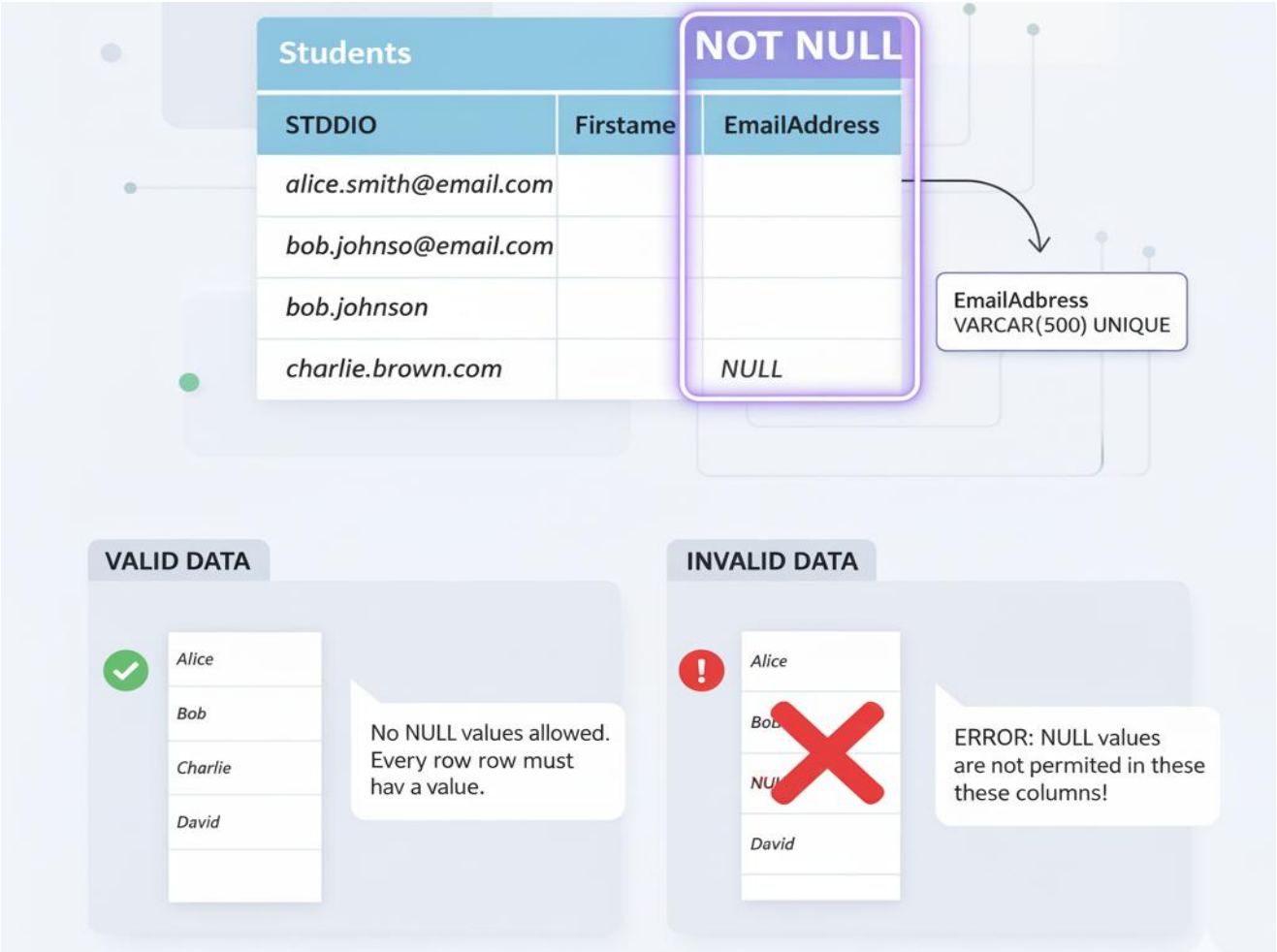
SUMMARY & OVERVIEW :

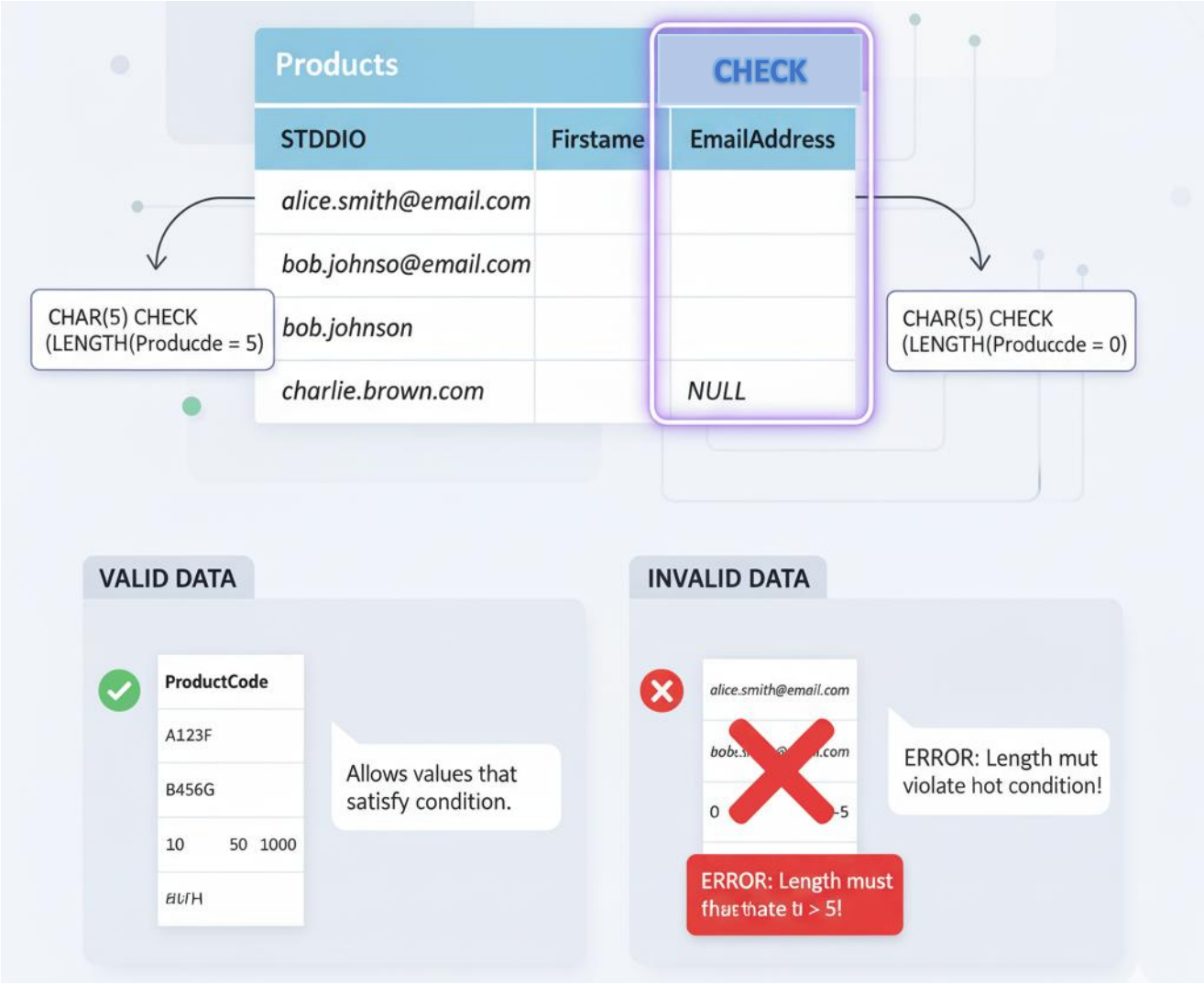







Students
UNIQUE
STDDIO
Firstame
EmailAddress
alice.smith@email.com
bob.johnso@email.com
bob.johnson
charlie.brown.com
NULL
EmailAdbress
VARCAR(100) UNIQUE
VALID DATA
alice.smith@email.com
bob.jamith@email.com
NULL
Allows one NULL value. All other values must distinct.
INVALID DATA
alice.smith@email.com
alice.smith@email.com
ERROR: Duplicate values not allowed!


Students
NOT NULL
STDDIO
Firstame
EmailAddress
alice.smith@email.com
bob.johnso@email.com
bob.johnson
charlie.brown.com
NULL
EmailAdbress
VARCAR(500) UNIQUE
VALID DATA
Alice
Bob
Charlie
David
No NULL values allowed. Every row row must hav a value.
INVALID DATA
Alice
David
ERROR: NULL values are not permited in these these columns!


Products
CHECK
STDDIO
Firstame
EmailAddress
alice.smith@email.com
bob.johnso@email.com
bob.johnson
charlie.brown.com
NULL
CHAR(5) CHECK (LENGTH(Producde = 5)
CHAR(5) CHECK (LENGTH(Produccde = 0)
VALID DATA
ProductCode
A123F
B456G
10 50 1000
Allows values that satisfy condition.
INVALID DATA
alice.smith@email.com
ERROR: Length mut violate hot condition!
ERROR: Length must

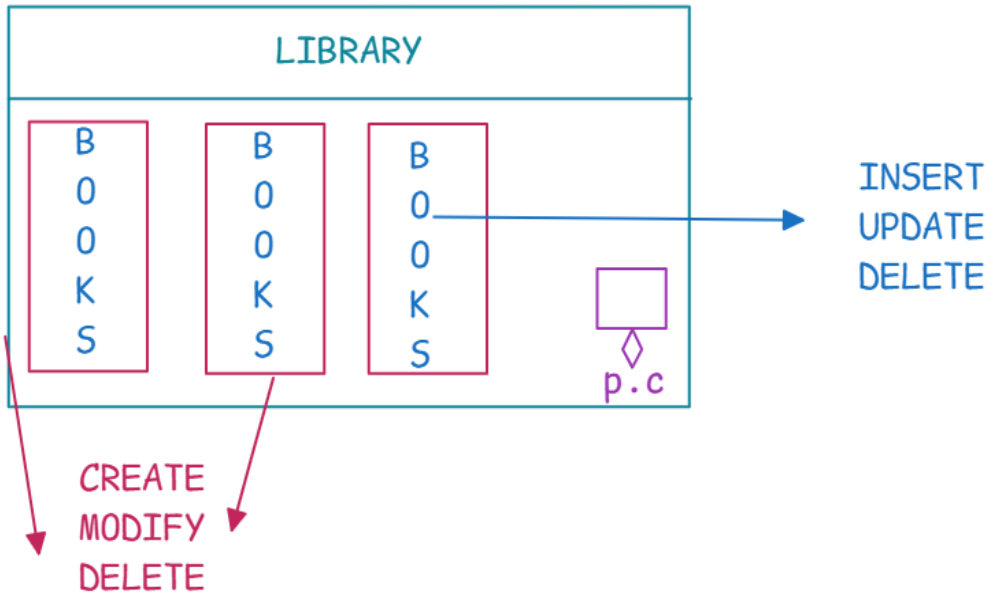
Overview of SQL Statements :

Overview of SQL Statements means how many Statements we have inside SQL

Types of SQL Statements :

1. DDL - DATA DEFINITION LANGUAGE
2. DML - DATA MANIPULATION LANGUAGE
3. TCL - TRANSACTION CONTROL LANGUAGE
4. DCL - DATA CONTROL LANGUAGE
5. DQL - DATA QUERY LANGUAGE

Example:



DATA DEFINITION LANGUAGE - DDL :

Data Definition Language(DDL) is used to CREATE or BUILD a Table into a Database and MODIFY or DELETE the Table from the Database

DATA MANIPULATION LANGUAGE - DML :

Data Manipulation Language(DML) is used to INSERT the Data/ Values/Records or UPDATE the Data inside the Table or DELETE the Data from the Table

TRANSACTION CONTROL LANGUAGE - TCL :

Transaction Control Language(TCL) is used to control the Transactions on the Database

TRANSACTIONS - Whatever operations performed on the DML Operations

And also, in order to Save the Data, we will be using TCL

DATA CONTROL LANGUAGE - DCL :

Data Control Language(DCL) is used to Control the flow of Data between the Users

DATA QUERY LANGUAGE - DQL :

Data Query Language(DQL) is used to Retrieve/Fetch → Getting the Data from the Database

DATA QUERY LANGUAGE - DQL :

Data Query Language(DQL) is used to **Retrieve/Fetch** → Getting the Data from the Database

We have 4 statements in DQL :

1. SELECT
2. PROJECTION
3. SELECTION
4. JOINS

> SELECT :

It is used to Retrieve/fetch the Data from the Table and display it

> PROJECTION :

It is used to Retrieve/fetch the Data from the Table by selecting only the Columns

> SELECTION :

It is used to Retrieve/fetch the Data from the Table by selecting both Rows and Columns

> JOINS :

It is used to Retrieve/fetch the Data from Multiple Tables Simultaneously

Projection :

It is used to Retrieve/fetch the Data from the Table
by selecting only the Columns

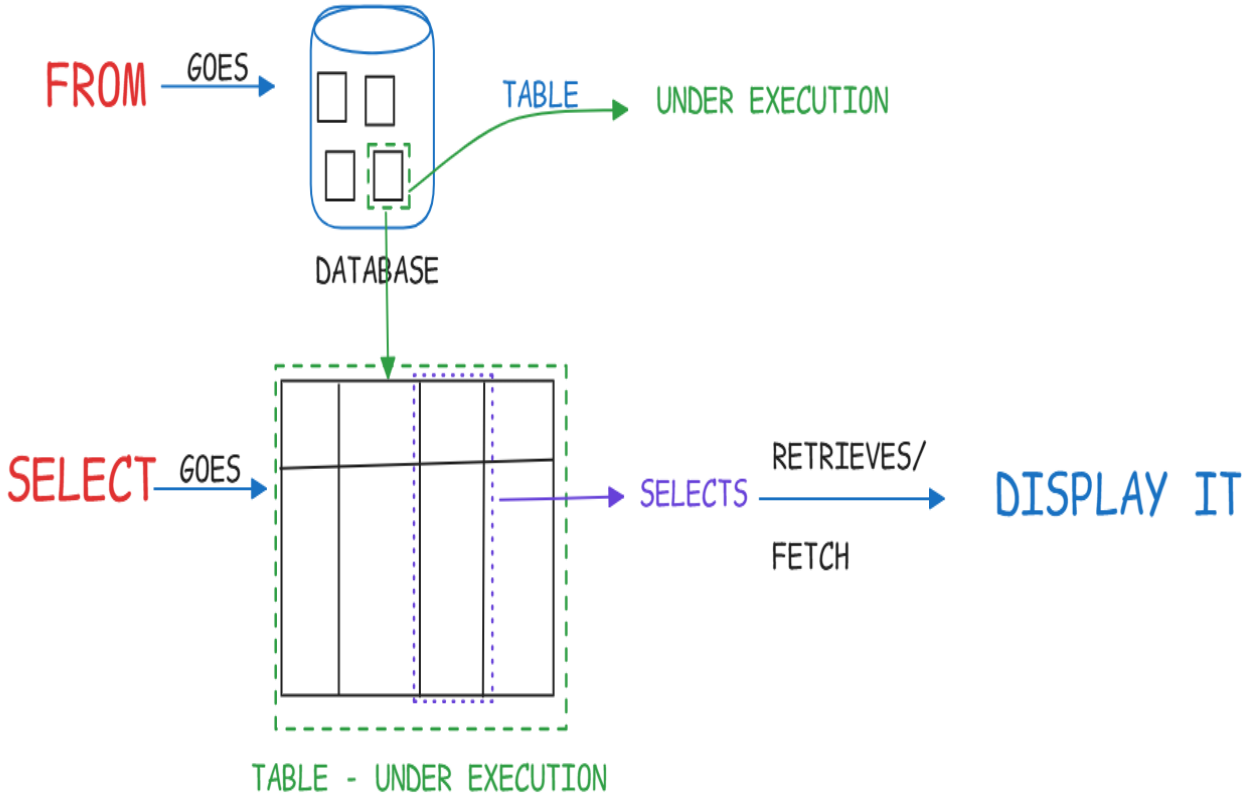
Syntax :

```
SELECT * / [DISTINCT] COLUMN_NAME / EXPRESSION [ALIAS]

FROM  TABLE_NAME ;
```

CLAUSES

ORDER OF EXECUTION :

1. FROM
2. SELECT



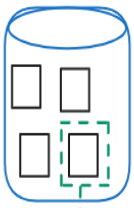
EMP

EMPNO	ENAME	PH_NO	SAL
1	Stark	1234567890	2000
2	Steve	0123654987	2500
3	Natasha	9876543210	3000
4	Bruce	8899774422	1000

WRITE A QUERY TO DISPLAY THE NAMES OF THE EMPLOYEES

```
SELECT NAMES
FROM EMPLOYEES;
```

COLUMN_NAME :



1

WRITE A QUERY TO DISPLAY THE NAMES OF THE EMPLOYEES

```
SELECT ENAME
FROM EMP ;
```

2

EMP

EMPNO	ENAME	PH_NO	SAL
1	Stark	1234567890	2000
2	Steve	0123654987	2500
3	Natasha	9876543210	3000
4	Bruce	8899774422	1000

FINAL O/P :

ENAME
Stark
Steve
Natasha
Bruce

UNDER EXECUTION

```
SQL> -- WRITE A QUERY TO DISPLAY NAMES OF THE EMPLOYEES
SQL> SELECT NAMES
  2  FROM EMPLOYEES;
FROM EMPLOYEES
     *
ERROR at line 2:
ORA-00942: table or view does not exist


SQL> SELECT ENAME
  2  FROM EMP;

ENAME
----------
SMITH
ALLEN
WARD
JONES
MARTIN
BLAKE
CLARK
SCOTT
KING
TURNER
ADAMS
JAMES
FORD
MILLER

14 rows selected.
```

WRITE A QUERY TO DISPLAY THE NAMES,EMPLOYEE NUMBER AND SALARY OF THE EMPLOYEES

```
SELECT ENAME,EMPNO,SAL

FROM EMP ;
```

FINAL O/P :

EMPNO	ENAME	SAL
1	Stark	2000
2	Steve	2500
3	Natasha	3000
4	Bruce	1000

```
SQL> -- WAQTD NAMES,EMPLOYEE NUMBER AND SALARY OF THE EMPLOYEES
SQL> SELECT ENAME,EMPNO,SAL
  2  FROM EMP;

ENAME             EMPNO          SAL
---------- ---------- ----------
SMITH              7369          800
ALLEN              7499         1600
WARD               7521         1250
JONES              7566         2975
MARTIN             7654         1250
BLAKE              7698         2850
CLARK              7782         2450
SCOTT              7788         3000
KING               7839         5000
TURNER             7844         1500
ADAMS              7876         1100
JAMES              7900          950
FORD               7902         3000
MILLER             7934         1300

14 rows selected.
```

WRITE A QUERY TO DISPLAY THE DETAILS OF THE EMPLOYEES

```
SELECT *

FROM EMP ;
```

FINAL O/P :

EMPNO	ENAME	PH_NO	SAL
1	Stark	1234567890	2000
2	Steve	0123654987	2500
3	Natasha	9876543210	3000
4	Bruce	8899774422	1000

```
SQL> -- WAQTD THE DETAILS OF THE EMPLOYEES
SQL> SELECT *
  2  FROM EMP;

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7369 SMITH      CLERK           7902 17-DEC-80        800                    20
      7499 ALLEN      SALESMAN        7698 20-FEB-81       1600        300         30
      7521 WARD       SALESMAN        7698 22-FEB-81       1250        500         30
      7566 JONES      MANAGER         7839 02-APR-81       2975                    20
      7654 MARTIN     SALESMAN        7698 28-SEP-81       1250       1400         30
      7698 BLAKE      MANAGER         7839 01-MAY-81       2850                    30
      7782 CLARK      MANAGER         7839 09-JUN-81       2450                    10
      7788 SCOTT      ANALYST         7566 19-APR-87       3000                    20
      7839 KING       PRESIDENT            17-NOV-81       5000                    10
      7844 TURNER     SALESMAN        7698 08-SEP-81       1500          0         30
      7876 ADAMS      CLERK           7788 23-MAY-87       1100                    20
      7900 JAMES      CLERK           7698 03-DEC-81        950                    30
      7902 FORD       ANALYST         7566 03-DEC-81       3000                    20
      7934 MILLER     CLERK           7782 23-JAN-82       1300                    10

14 rows selected.

SQL> SELECT EMP.*
  2  FROM EMP;

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7369 SMITH      CLERK           7902 17-DEC-80        800                    20
      7499 ALLEN      SALESMAN        7698 20-FEB-81       1600        300         30
      7521 WARD       SALESMAN        7698 22-FEB-81       1250        500         30
      7566 JONES      MANAGER         7839 02-APR-81       2975                    20
      7654 MARTIN     SALESMAN        7698 28-SEP-81       1250       1400         30
      7698 BLAKE      MANAGER         7839 01-MAY-81       2850                    30
      7782 CLARK      MANAGER         7839 09-JUN-81       2450                    10
      7788 SCOTT      ANALYST         7566 19-APR-87       3000                    20
      7839 KING       PRESIDENT            17-NOV-81       5000                    10
      7844 TURNER     SALESMAN        7698 08-SEP-81       1500          0         30
      7876 ADAMS      CLERK           7788 23-MAY-87       1100                    20
      7900 JAMES      CLERK           7698 03-DEC-81        950                    30
      7902 FORD       ANALYST         7566 03-DEC-81       3000                    20
      7934 MILLER     CLERK           7782 23-JAN-82       1300                    10

14 rows selected.
```

DISTINCT :

DISTINCT is a Keyword which is used to remove the
Duplicated / repeated / same Values / records
from the Result Table

Characteristics of DISTINCT Keyword :

1. Remove Duplicate Rows -

   > Returns only the unique rows in the result set

   > Duplicates are Eliminated based on the columns
   present in SELECT List

2. DISTINCT Keyword has to be written as a First Argument Immediately
   after passing SELECT Clause

   EX: `SELECT DISTINCT ENAME` ✓ `SELECT DISTINCT ENAME,SAL,EMPNO` ✓

   `SELECT ENAME DISTINCT` ✗ `SELECT EMPNO,DISTINCT ENAME` ✗

3. Column names should be written after this DISTINCT Keyword

   `SELECT DISTINCT ENAME`

   `SELECT ENAME DISTINCT`

4. We can pass multiple columns to DISTINCT, it will remove the combination
   of columns in which values are repeated.

EXAMPLE:

WRITE A QUERY TO DISPLAY THE DIFFERENT SALARIES OF THE EMPLOYEES

EMP

EMPNO	ENAME	PH_NO	SAL
1	Stark	1234567890	2000
2	Steve	0123654987	2500
3	Natasha	9876543210	3000
4	Bruce	8899774422	1000
5	Stark	1234567890	2000
6	Bruce	8899774422	1000

```
SELECT DISTINCT SAL

FROM EMP ;
```

SAL
2000
2500
3000
1000

→ FINAL O/P

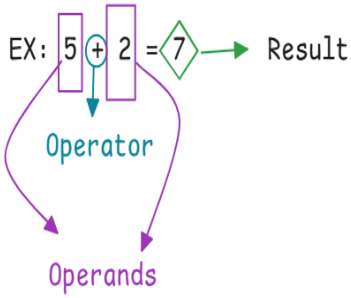
```
SQL> -- WAQTD DIFFERENT SALARIES OF THE EMPLOYEES
SQL> SELECT DISTINCT SAL
  2  FROM EMP;

       SAL
----------
       800
      1600
      1250
      2975
      2850
      2450
      3000
      5000
      1500
      1100
       950
      1300

12 rows selected.
```

EXPRESSION :

Any Statement which gives a result is known as an Expression

OR

An Expression is a Combination of Operands and Operators that evaluates to a Single Value



1. Operands :

An Operand is the Value, Column Names on which the Operator acts (Constants)

EX : SAL * 12 = YEARLY SALARY / ANNUAL SALARY

SAL + 100 = TOTAL (SAL+BONUS)

2. Operators :

An Operator is a Symbol that tells what action to be performed on the Operands

EXAMPLE:

WRITE A QUERY TO DISPLAY THE ANNUAL SALARIES OF THE EMPLOYEES

EMP

EMPNO	ENAME	PH_NO	SAL
1	Stark	1234567890	2000
2	Steve	0123654987	2500
3	Natasha	9876543210	3000
4	Bruce	8899774422	1000
5	Stark	1234567890	2000
6	Bruce	8899774422	1000

```
SELECT SAL*12
FROM EMP ;
```

SAL*12
24000
30000
36000
12000
24000
12000

→ FINAL O/P :

ALIAS :

An ALIAS Name temporarily renames a Column or an Expression in the Result Table / Final output for better Readability

1. With AS keyword
2. With Underscore _
3. Without AS keyword --> " "

EXAMPLE:

WRITE A QUERY TO DISPLAY THE ANNUAL SALARIES OF THE EMPLOYEES

EMP

EMPNO	ENAME	PH_NO	SAL
1	Stark	1234567890	2000
2	Steve	0123654987	2500
3	Natasha	9876543210	3000
4	Bruce	8899774422	1000
5	Stark	1234567890	2000
6	Bruce	8899774422	1000

```
SELECT SAL*12 AS AnnualSalary
FROM EMP ;
```

AnnualSalary
24000
30000
36000
12000
24000
12000

→ FINAL O/P

EMP

EMPNO	ENAME	PH_NO	SAL
1	Stark	1234567890	2000
2	Steve	0123654987	2500
3	Natasha	9876543210	3000
4	Bruce	8899774422	1000
5	Stark	1234567890	2000
6	Bruce	8899774422	1000

```
SELECT SAL*12 Annual_Salary
FROM EMP ;
```

Annual_Salary
24000
30000
36000
12000
24000
12000

→ FINAL O/P

EMP

EMPNO	ENAME	PH_NO	SAL
1	Stark	1234567890	2000
2	Steve	0123654987	2500
3	Natasha	9876543210	3000
4	Bruce	8899774422	1000
5	Stark	1234567890	2000
6	Bruce	8899774422	1000

```
SELECT SAL*12"Annual_Salary"
FROM EMP ;
```

Annual_Salary
24000
30000
36000
12000
24000
12000

→ FINAL O/P

```
SQL> -- WAQTD ANNUAL SALARIES OF THE EMPLOYEES
SQL>
SQL> SELECT SAL*12
  2  FROM EMP;

    SAL*12
----------
      9600
     19200
     15000
     35700
     15000
     34200
     29400
     36000
     60000
     18000
     13200
     11400
     36000
     15600

14 rows selected.

SQL> SELECT SAL*12 AS ANNUALSALARY
  2  FROM EMP;

ANNUALSALARY
------------
        9600
       19200
       15000
       35700
       15000
       34200
       29400
       36000
       60000
       18000
       13200
       11400
       36000
       15600

14 rows selected.
```

```
SQL> SELECT SAL*12 ANNUAL_SALARY
  2  FROM EMP;

ANNUAL_SALARY
-------------
         9600
        19200
        15000
        35700
        15000
        34200
        29400
        36000
        60000
        18000
        13200
        11400
        36000
        15600

14 rows selected.

SQL> SELECT SAL*12ANNUAL_SALARY
  2  FROM EMP;

ANNUAL_SALARY
-------------
         9600
        19200
        15000
        35700
        15000
        34200
        29400
        36000
        60000
        18000
        13200
        11400
        36000
        15600

14 rows selected.
```

```
SQL> SELECT SAL*12 ANNUAL SALARY
  2  FROM EMP;
SELECT SAL*12 ANNUAL SALARY
                  *
ERROR at line 1:
ORA-00923: FROM keyword not found where expected


SQL> SELECT SAL*12 "ANNUAL_SALARY"
  2  FROM EMP;

ANNUAL_SALARY
-------------
         9600
        19200
        15000
        35700
        15000
        34200
        29400
        36000
        60000
        18000
        13200
        11400
        36000
        15600

14 rows selected.

SQL> SELECT *,SAL*12
  2  FROM EMP;
SELECT *,SAL*12
        *
ERROR at line 1:
ORA-00923: FROM keyword not found where expected


SQL> SELECT *,ENAME
  2  FROM EMP;
SELECT *,ENAME
        *
ERROR at line 1:
ORA-00923: FROM keyword not found where expected
```

```
SQL> -- WAQTD THE DETAILS OF THE EMPLOYEES ALONG WITH THEIR ANNUAL SALARY
SQL> SELECT EMP.* , SAL*12 "ANN_SAL"
  2  FROM EMP;

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO    ANN_SAL
---------- ---------- --------- ---------- --------- ---------- ---------- ---------- ----------
      7369 SMITH      CLERK           7902 17-DEC-80        800                    20       9600
      7499 ALLEN      SALESMAN        7698 20-FEB-81       1600        300         30      19200
      7521 WARD       SALESMAN        7698 22-FEB-81       1250        500         30      15000
      7566 JONES      MANAGER         7839 02-APR-81       2975                    20      35700
      7654 MARTIN     SALESMAN        7698 28-SEP-81       1250       1400         30      15000
      7698 BLAKE      MANAGER         7839 01-MAY-81       2850                    30      34200
      7782 CLARK      MANAGER         7839 09-JUN-81       2450                    10      29400
      7788 SCOTT      ANALYST         7566 19-APR-87       3000                    20      36000
      7839 KING       PRESIDENT            17-NOV-81       5000                    10      60000
      7844 TURNER     SALESMAN        7698 08-SEP-81       1500          0         30      18000
      7876 ADAMS      CLERK           7788 23-MAY-87       1100                    20      13200
      7900 JAMES      CLERK           7698 03-DEC-81        950                    30      11400
      7902 FORD       ANALYST         7566 03-DEC-81       3000                    20      36000
      7934 MILLER     CLERK           7782 23-JAN-82       1300                    10      15600

14 rows selected.

SQL> SELECT EMP.*
  2  FROM EMP;

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7369 SMITH      CLERK           7902 17-DEC-80        800                    20
      7499 ALLEN      SALESMAN        7698 20-FEB-81       1600        300         30
      7521 WARD       SALESMAN        7698 22-FEB-81       1250        500         30
      7566 JONES      MANAGER         7839 02-APR-81       2975                    20
      7654 MARTIN     SALESMAN        7698 28-SEP-81       1250       1400         30
      7698 BLAKE      MANAGER         7839 01-MAY-81       2850                    30
      7782 CLARK      MANAGER         7839 09-JUN-81       2450                    10
      7788 SCOTT      ANALYST         7566 19-APR-87       3000                    20
      7839 KING       PRESIDENT            17-NOV-81       5000                    10
      7844 TURNER     SALESMAN        7698 08-SEP-81       1500          0         30
      7876 ADAMS      CLERK           7788 23-MAY-87       1100                    20
      7900 JAMES      CLERK           7698 03-DEC-81        950                    30
      7902 FORD       ANALYST         7566 03-DEC-81       3000                    20
      7934 MILLER     CLERK           7782 23-JAN-82       1300                    10

14 rows selected.
```

Selection :

It is used to Retrieve/fetch the Data from the Table
by selecting both rows and columns

Syntax :

```
SELECT * / [DISTINCT] COLUMN_NAME / EXPRESSION [ALIAS]
FROM   TABLE_NAME
WHERE  <FILTER-CONDITIONS> ;
```

↓

CLAUSES

ORDER OF EXECUTION :

1. FROM
2. WHERE
3. SELECT

WAY OF WRITING

1. SELECT
2. FROM
3. WHERE

Characteristics of WHERE Clause :

1. The WHERE Clause executes Row - By - Row
2. It executes after the Execution of FROM clause
3. We can pass the Filter condition inside the WHERE clause
4. We can pass Multiple Conditions in the WHERE Clause with the help of Logical Operators

EXAMPLE :

WRITE A QUERY TO DISPLAY THE NAMES OF THE EMPLOYEES WHOSE SAL IS GREATER THAN 1000

EMP

EMPNO	ENAME	PH_NO	SAL
1	Stark	1234567890	2000
2	Steve	0123654987	2500
3	Natasha	9876543210	3000
4	Bruce	8899774422	1000

```
SELECT ENAME

FROM EMP

WHERE SAL > 1000 ;
```

UNDER EXECUTION

SAL > 1000

2000 > 1000

2500 > 1000

3000 > 1000

1000 > 1000

O/P OF WHERE CLAUSE:

EMPNO	ENAME	PH_NO	SAL
1	Stark	1234567890	2000
2	Steve	0123654987	2500
3	Natasha	9876543210	3000

FINAL O/P:

ENAME
Stark
Steve
Natasha

```
SQL> -- WRITE A QUERY TO DISPLAY THE NAMES OF THE EMPLOYEES WHOSE SAL IS GREATER THAN 1000
SQL> SELECT ENAME
  2  FROM EMP
  3  WHERE SAL > 1000 ;

ENAME
----------
ALLEN
WARD
JONES
MARTIN
BLAKE
CLARK
SCOTT
KING
TURNER
ADAMS
FORD
MILLER

12 rows selected.

SQL> SELECT EMP.*
  2  FROM EMP;

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7369 SMITH      CLERK           7902 17-DEC-80        800                    20
      7499 ALLEN      SALESMAN        7698 20-FEB-81       1600        300         30
      7521 WARD       SALESMAN        7698 22-FEB-81       1250        500         30
      7566 JONES      MANAGER         7839 02-APR-81       2975                    20
      7654 MARTIN     SALESMAN        7698 28-SEP-81       1250       1400         30
      7698 BLAKE      MANAGER         7839 01-MAY-81       2850                    30
      7782 CLARK      MANAGER         7839 09-JUN-81       2450                    10
      7788 SCOTT      ANALYST         7566 19-APR-87       3000                    20
      7839 KING       PRESIDENT            17-NOV-81       5000                    10
      7844 TURNER     SALESMAN        7698 08-SEP-81       1500          0         30
      7876 ADAMS      CLERK           7788 23-MAY-87       1100                    20
      7900 JAMES      CLERK           7698 03-DEC-81        950                    30
      7902 FORD       ANALYST         7566 03-DEC-81       3000                    20
      7934 MILLER     CLERK           7782 23-JAN-82       1300                    10

14 rows selected.
```

```
SQL> -- WAQTD DETAILS OF THE EMPLOYEES WHO HAVE HIRED AFTER THE YEAR 1981
SQL> SELECT EMP.*
  2  FROM EMP
  3  WHERE HIREDATE > '1981' ;
WHERE HIREDATE > '1981'
                 *
ERROR at line 3:
ORA-01861: literal does not match format string


SQL> SELECT EMP.*
  2  FROM EMP
  3  WHERE HIREDATE > '01-JAN-1982' ;

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7788 SCOTT      ANALYST         7566 19-APR-87       3000                    20
      7876 ADAMS      CLERK           7788 23-MAY-87       1100                    20
      7934 MILLER     CLERK           7782 23-JAN-82       1300                    10

SQL> SELECT EMP.*
  2  FROM EMP
  3  WHERE HIREDATE > '31-DEC-1981' ;

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7788 SCOTT      ANALYST         7566 19-APR-87       3000                    20
      7876 ADAMS      CLERK           7788 23-MAY-87       1100                    20
      7934 MILLER     CLERK           7782 23-JAN-82       1300                    10
```

```
SQL> -- WAQTD ALL THE DETAILS OF EMPLOYESS WHO IS WORKING AS CLERK / WHOSE DESIGNATION IS CLERK
SQL> SELECT EMP.*
  2  FROM EMP
  3  WHERE JOB = CLERK;
WHERE JOB = CLERK
            *
ERROR at line 3:
ORA-00904: "CLERK": invalid identifier


SQL> SELECT EMP.*
  2  FROM EMP
  3  WHERE JOB = 'CLERK';

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7369 SMITH      CLERK           7902 17-DEC-80        800                    20
      7876 ADAMS      CLERK           7788 23-MAY-87       1100                    20
      7900 JAMES      CLERK           7698 03-DEC-81        950                    30
      7934 MILLER     CLERK           7782 23-JAN-82       1300                    10

SQL> cl scr

SQL> SELECT EMP.*
  2  FROM EMP;

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7369 SMITH      CLERK           7902 17-DEC-80        800                    20
      7499 ALLEN      SALESMAN        7698 20-FEB-81       1600        300         30
      7521 WARD       SALESMAN        7698 22-FEB-81       1250        500         30
      7566 JONES      MANAGER         7839 02-APR-81       2975                    20
      7654 MARTIN     SALESMAN        7698 28-SEP-81       1250       1400         30
      7698 BLAKE      MANAGER         7839 01-MAY-81       2850                    30
      7782 CLARK      MANAGER         7839 09-JUN-81       2450                    10
      7788 SCOTT      ANALYST         7566 19-APR-87       3000                    20
      7839 KING       PRESIDENT            17-NOV-81       5000                    10
      7844 TURNER     SALESMAN        7698 08-SEP-81       1500          0         30
      7876 ADAMS      CLERK           7788 23-MAY-87       1100                    20
      7900 JAMES      CLERK           7698 03-DEC-81        950                    30
      7902 FORD       ANALYST         7566 03-DEC-81       3000                    20
      7934 MILLER     CLERK           7782 23-JAN-82       1300                    10

14 rows selected.
```

WRITE A QUERY TO DISPLAY THE NAMES OF THE EMPLOYEES WHOSE SAL IS GREATER THAN 1000 , WITH EMPNO 4

OPERATORS

Operators :

An Operator is a Symbol that tells what action to be performed on the Operands

OR

An Operator is a Symbol which is used to Perform Specific Task

Types of Operators :

1. ARITHMETIC OPERATORS (+, -, *, /)
2. COMPARISON OPERATORS (= , !=)
3. RELATIONAL OPERATORS (>, <, >=, <=)
4. CONCATENATION OPERATORS (||)
5. LOGICAL OPERATORS (AND, OR, NOT)
6. SPECIAL OPERATORS

   (IN, NOT IN, BETWEEN, NOT BETWEEN, IS, IS NOT, LIKE, NOT LIKE)
7. SUB-QUERY OPERATORS (ALL, ANY, EXISTS, NOT EXISTS)
8. SET OPERATORS (UNION, UNION ALL, INTERSECT, MINUS)

CONCATENATION OPERATORS :

> It is used to Join/Combine more than one string

> Symbol: ||

```
SQL>--WAQTD THE NAMES OF EMP'S WITH A SALUTATION 'MR.' / NAMES OF EMP'S IN THE FOLLOWING FORMAT EX: MR. X
SQL> SELECT 'MR. ' || ENAME "NAMES"
  2  FROM EMP;

NAMES
--------------
MR. SMITH
MR. ALLEN
MR. WARD
MR. JONES
MR. MARTIN
MR. BLAKE
MR. CLARK
MR. SCOTT
MR. KING
MR. TURNER
MR. ADAMS
MR. JAMES
MR. FORD
MR. MILLER

14 rows selected.

SQL> SELECT 'MR. ' || ENAME
  2  FROM EMP;

'MR.'||ENAME
--------------
MR. SMITH
MR. ALLEN
MR. WARD
MR. JONES
MR. MARTIN
MR. BLAKE
MR. CLARK
MR. SCOTT
MR. KING
MR. TURNER
MR. ADAMS
MR. JAMES
MR. FORD
MR. MILLER

14 rows selected.
```

LOGICAL OPERATORS :

> It is used to pass Multiple Conditions inside the "WHERE" Clause

> Types of Logical Operators:

1. AND
2. OR
3. NOT

AND Operator :

> AND Operator is a type Logical Operator which returns "TRUE" when all the conditions are satisfied

OR Operator :

> OR Operator is a type Logical Operator which returns "TRUE" when any one of the condition is satisfied

NOT Operator :

> NOT Operator is a type Logical Operator which is used to reverse the given Input

WRITE A QUERY TO DISPLAY THE NAMES OF THE EMPLOYEES WHOSE SAL IS GREATER THAN 2000 and working in deptno 20

EMP

EMPNO	ENAME	SAL	Ph_No	DEPTNO
123	Stark	2000	3210654987	10
210	Steve	1500	1234567895	20
312	Natasha	5000	3123023698	20
187	Peter	1000	7788994455	40
230	Bruce	900	9988665544	10
364	Thor	2500	3215478961	20

```
SELECT ENAME

FROM EMP

WHERE SAL > 2000  AND DEPTNO = 20 ;
```

1. SAL > 2000 AND 2. DEPTNO = 20

UNSELECTED	2000 > 2000	10 = 20	UNSELECTED
	1500 > 2000	20 = 20	
SELECTED	5000 > 2000	20 = 20	SELECTED
UNSELECTED	1000 > 2000	40 = 20	UNSELECTED
	900 > 2000	10 = 20	
SELECTED	2500 > 2000	20 = 20	SELECTED

O/P OF WHERE CLAUSE

EMPNO	ENAME	SAL	Ph_No	DEPTNO
312	Natasha	5000	3123023698	20
364	Thor	2500	3215478961	20

FINAL O/P:

ENAME
Natasha
Thor

```
SQL> SELECT EMP. *
  2  FROM EMP;

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7369 SMITH      CLERK          7902 17-DEC-80        800                    20
      7499 ALLEN      SALESMAN       7698 20-FEB-81       1600        300         30
      7521 WARD       SALESMAN       7698 22-FEB-81       1250        500         30
      7566 JONES      MANAGER        7839 02-APR-81       2975                    20
      7654 MARTIN     SALESMAN       7698 28-SEP-81       1250       1400         30
      7698 BLAKE      MANAGER        7839 01-MAY-81       2850                    30
      7782 CLARK      MANAGER        7839 09-JUN-81       2450                    10
      7788 SCOTT      ANALYST        7566 19-APR-87       3000                    20
      7839 KING       PRESIDENT           17-NOV-81       5000                    10
      7844 TURNER     SALESMAN       7698 08-SEP-81       1500          0         30
      7876 ADAMS      CLERK          7788 23-MAY-87       1100                    20
      7900 JAMES      CLERK          7698 03-DEC-81        950                    30
      7902 FORD       ANALYST        7566 03-DEC-81       3000                    20
      7934 MILLER     CLERK          7782 23-JAN-82       1300                    10

14 rows selected.

SQL> -- WRITE A QUERY TO DISPLAY THE NAMES OF THE EMPLOYEES WHOSE SAL IS GREATER
SQL> THAN 2000 and working in deptno 20
SQL> SELECT ENAME
  2  FROM EMP
  3  WHERE SAL > 2000 AND DEPTNO = 20;

ENAME
----------
JONES
SCOTT
FORD

SQL> -- WAQTD ALL THE DETAILS OF EMP'S WHOSE SAL IS GREATER THAN 3000 AND WORKING IN DEPTNO 30
SQL> SELECT EMP.*
  2  FROM EMP
  3  WHERE SAL > 3000 AND DEPTNO = 30;

no rows selected
```

WRITE A QUERY TO DISPLAY THE NAMES OF THE EMPLOYEES WHOSE SAL IS GREATER THAN 2000 or working in deptno 20

EMP

EMPNO	ENAME	SAL	Ph_No	DEPTNO
123	Stark	2000	3210654987	10
210	Steve	1500	1234567895	20
312	Natasha	5000	3123023698	20
187	Peter	1000	7788994455	40
230	Bruce	900	9988665544	10
364	Thor	2500	3215478961	20

```
SELECT ENAME

FROM EMP

WHERE SAL > 2000  OR DEPTNO = 20 ;
```

1. SAL > 2000 OR 2. DEPTNO = 20

	1. SAL > 2000		2. DEPTNO = 20	
UNSELECTED	2000 > 2000		10 = 20	UNSELECTED
UNSELECTED	1500 > 2000		20 = 20	SELECTED
SELECTED	5000 > 2000		20 = 20	SELECTED
UNSELECTED	1000 > 2000		40 = 20	UNSELECTED
UNSELECTED	900 > 2000		10 = 20	UNSELECTED
SELECTED	2500 > 2000		20 = 20	SELECTED

O/P OF WHERE CLAUSE

EMPNO	ENAME	SAL	Ph_No	DEPTNO
210	Steve	1500	1234567895	20
312	Natasha	5000	3123023698	20
364	Thor	2500	3215478961	20

FINAL O/P:

ENAME
Steve
Natasha
Thor

```
SQL> -- WAQD THE NAMES OF THE EMPLOYEES WHERE SAL IS GREATER THAN 2000 OR DEPTNO = 20
SQL> SELECT ENAME
  2  FROM EMP
  3  WHERE SAL > 2000 OR DEPTNO = 20;

ENAME
----------
SMITH
JONES
BLAKE
CLARK
SCOTT
KING
ADAMS
FORD

8 rows selected.

SQL> SELECT EMP.*
  2  FROM EMP;

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7369 SMITH      CLERK           7902 17-DEC-80        800                    20
      7499 ALLEN      SALESMAN        7698 20-FEB-81       1600        300         30
      7521 WARD       SALESMAN        7698 22-FEB-81       1250        500         30
      7566 JONES      MANAGER         7839 02-APR-81       2975                    20
      7654 MARTIN     SALESMAN        7698 28-SEP-81       1250       1400         30
      7698 BLAKE      MANAGER         7839 01-MAY-81       2850                    30
      7782 CLARK      MANAGER         7839 09-JUN-81       2450                    10
      7788 SCOTT      ANALYST         7566 19-APR-87       3000                    20
      7839 KING       PRESIDENT            17-NOV-81       5000                    10
      7844 TURNER     SALESMAN        7698 08-SEP-81       1500          0         30
      7876 ADAMS      CLERK           7788 23-MAY-87       1100                    20
      7900 JAMES      CLERK           7698 03-DEC-81        950                    30
      7902 FORD       ANALYST         7566 03-DEC-81       3000                    20
      7934 MILLER     CLERK           7782 23-JAN-82       1300                    10

14 rows selected.
```

```
SQL> SELECT EMP.*
  2  FROM EMP
  3  WHERE DEPTNO=10 OR DEPTNO = 20 OR DEPTNO = 30;

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7369 SMITH      CLERK           7902 17-DEC-80        800                    20
      7499 ALLEN      SALESMAN        7698 20-FEB-81       1600        300         30
      7521 WARD       SALESMAN        7698 22-FEB-81       1250        500         30
      7566 JONES      MANAGER         7839 02-APR-81       2975                    20
      7654 MARTIN     SALESMAN        7698 28-SEP-81       1250       1400         30
      7698 BLAKE      MANAGER         7839 01-MAY-81       2850                    30
      7782 CLARK      MANAGER         7839 09-JUN-81       2450                    10
      7788 SCOTT      ANALYST         7566 19-APR-87       3000                    20
      7839 KING       PRESIDENT            17-NOV-81       5000                    10
      7844 TURNER     SALESMAN        7698 08-SEP-81       1500          0         30
      7876 ADAMS      CLERK           7788 23-MAY-87       1100                    20
      7900 JAMES      CLERK           7698 03-DEC-81        950                    30
      7902 FORD       ANALYST         7566 03-DEC-81       3000                    20
      7934 MILLER     CLERK           7782 23-JAN-82       1300                    10

14 rows selected.

SQL> SELECT EMP.*
  2  FROM EMP
  3  WHERE DEPTNO=10 AND DEPTNO = 20 AND DEPTNO = 30;

no rows selected
```

```
SQL> --WAQTD DETAILS OF EMP'S WHO ARE WORKING AS CLERK,SALESMAN,MANAGER
SQL> SELECT EMP.*
  2  FROM EMP
  3  WHERE JOB='CLERK' OR JOB='SALESMAN' OR JOB='MANAGER';

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7369 SMITH      CLERK           7902 17-DEC-80        800                    20
      7499 ALLEN      SALESMAN        7698 20-FEB-81       1600        300         30
      7521 WARD       SALESMAN        7698 22-FEB-81       1250        500         30
      7566 JONES      MANAGER         7839 02-APR-81       2975                    20
      7654 MARTIN     SALESMAN        7698 28-SEP-81       1250       1400         30
      7698 BLAKE      MANAGER         7839 01-MAY-81       2850                    30
      7782 CLARK      MANAGER         7839 09-JUN-81       2450                    10
      7844 TURNER     SALESMAN        7698 08-SEP-81       1500          0         30
      7876 ADAMS      CLERK           7788 23-MAY-87       1100                    20
      7900 JAMES      CLERK           7698 03-DEC-81        950                    30
      7934 MILLER     CLERK           7782 23-JAN-82       1300                    10

11 rows selected.
```

WRITE A QUERY TO DISPLAY THE NAMES AND DEPTNO OF THE EMPLOYEES WHO ARE NOT WORKING IN DEPTNO 10 / NAMES AND DEPTNO OF THE EMPLOYEES EXCEPT IN DEPTNO 10

EMP

EMPNO	ENAME	SAL	Ph_No	DEPTNO
123	Stark	2000	3210654987	10
210	Steve	1500	1234567895	20
312	Natasha	5000	3123023698	20
187	Peter	1000	7788994455	40
230	Bruce	900	9988665544	10
364	Thor	2500	3215478961	20

```
SELECT ENAME,DEPTNO

FROM EMP

WHERE NOT DEPTNO = 10 ;
```

NOT DEPTNO = 10

10 = 10 UNSELECTED

20 = 10 SELECTED

20 = 10

40 = 10 SELECTED

10 = 10 UNSELECTED

20 = 10 SELECTED

FINAL O/P:

ENAME	DEPTNO
STEVE	20
NATASHA	20
PETER	40
THOR	20

```
SQL> --WAQTD THE NAMES OF THE EMPLOYEES WHO ARE NOT WORKING IN DEPTNO 10 /
SQL> --NAMES AND DEPTNO OF THE EMPLOYEES EXCEPT IN DEPTNO 10
SQL>
SQL> SELECT ENAME,DEPTNO
  2  FROM EMP
  3  WHERE NOT DEPTNO=10;

ENAME          DEPTNO
---------- ----------
SMITH              20
ALLEN              30
WARD               30
JONES              20
MARTIN             30
BLAKE              30
SCOTT              20
TURNER             30
ADAMS              20
JAMES              30
FORD               20

11 rows selected.
```

```
SQL> -- WAQTD THE DETAILS OF THE EMPLOYEES EXCEPT WHO ARE WORKING AS MANAGER
SQL> SELECT EMP.*
  2  FROM EMP
  3  WHERE NOT JOB='MANAGER';

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7369 SMITH      CLERK           7902 17-DEC-80        800                    20
      7499 ALLEN      SALESMAN        7698 20-FEB-81       1600        300         30
      7521 WARD       SALESMAN        7698 22-FEB-81       1250        500         30
      7654 MARTIN     SALESMAN        7698 28-SEP-81       1250       1400         30
      7788 SCOTT      ANALYST         7566 19-APR-87       3000                    20
      7839 KING       PRESIDENT            17-NOV-81       5000                    10
      7844 TURNER     SALESMAN        7698 08-SEP-81       1500          0         30
      7876 ADAMS      CLERK           7788 23-MAY-87       1100                    20
      7900 JAMES      CLERK           7698 03-DEC-81        950                    30
      7902 FORD       ANALYST         7566 03-DEC-81       3000                    20
      7934 MILLER     CLERK           7782 23-JAN-82       1300                    10

11 rows selected.
```

SPECIAL OPERATORS :

> Special Operators are the type of Operators where we have eight types of Special Operators in SQL

Types of Special Operators :

> IN	> NOT IN
> BETWEEN	> NOT BETWEEN
> IS	> IS NOT
> LIKE	> NOT LIKE

1. IN Operator :

> IN Operator is a type of Special Operator which is a Multi-Valued Operator used to accept Multiple Values at the R.H.S

> Example :

```
deptno = 10 or deptno = 20 or deptno = 30 or deptno = 40
deptno in (10,20,30,40)
```

> Syntax : COLUMN_NAME/EXPRESSION IN (V1,V2,V3,V4,...,VN);

> We can also pass Single Values at the R.H.S using IN Operator

> Example : deptno in 10

> Whenever we are passing Multiple Values, Brackets are Mandatory

WRITE A QUERY TO DISPLAY THE NAMES AND DEPTNO OF THE EMPLOYEES WHO ARE working in deptno 10,20,30

EMP

EMPNO	ENAME	SAL	Ph_No	DEPTNO
123	Stark	2000	3210654987	10
210	Steve	1500	1234567895	20
312	Natasha	5000	3123023698	20
187	Peter	1000	7788994455	40
230	Bruce	900	9988665544	10
364	Thor	2500	3215478961	20

```
SELECT ENAME , DEPTNO

FROM EMP

WHERE DEPTNO IN (10,20,30) ;
```

DEPTNO IN 10,20,30

10 IN 10 SELECTED

20 IN 10
20 IN 10 UNSELECTED
40 IN 10

10 IN 10 SELECTED

20 IN 10 UNSELECTED

10 IN 20 UNSELECTED

20 IN 20 SELECTED

20 IN 20 SELECTED

40 IN 20
10 IN 20 UNSELECTED

20 IN 20 SELECTED

10 IN 30
20 IN 30
20 IN 30
40 IN 30
10 IN 30
20 IN 30 UNSELECTED

FINAL O/P:

ENAME	DEPTNO
STARK	10
STEVE	20
NATASHA	20
BRUCE	10
THOR	20

```
SQL> --WAQTD THE NAMES AND DEPTNO OF THE EMP'S WHO ARE WORKING IN DEPTNO 10,20,30
SQL>
SQL> SELECT ENAME, DEPTNO
  2  FROM EMP
  3  WHERE DEPTNO IN(10,20,30);

ENAME          DEPTNO
---------- ----------
SMITH              20
ALLEN              30
WARD               30
JONES              20
MARTIN             30
BLAKE              30
CLARK              10
SCOTT              20
KING               10
TURNER             30
ADAMS              20
JAMES              30
FORD               20
MILLER             10

14 rows selected.

SQL> --WAQTD ALL THE DETAILS OF THE EMP'S WHO ARE WORKING AS CLERK, MANAGER
SQL>
SQL> SELECT EMP.*
  2  FROM EMP
  3  WHERE JOB IN('CLERK','MANAGER');

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7369 SMITH      CLERK           7902 17-DEC-80        800                    20
      7566 JONES      MANAGER         7839 02-APR-81       2975                    20
      7698 BLAKE      MANAGER         7839 01-MAY-81       2850                    30
      7782 CLARK      MANAGER         7839 09-JUN-81       2450                    10
      7876 ADAMS      CLERK           7788 23-MAY-87       1100                    20
      7900 JAMES      CLERK           7698 03-DEC-81        950                    30
      7934 MILLER     CLERK           7782 23-JAN-82       1300                    10

7 rows selected.
```

2. NOT IN Operator :

> NOT IN Operator is similar to IN Operator instead of selecting the Records/Values at R.H.S, it will reject the values

> Syntax : COLUMN_NAME/EXPRESSION NOT IN (V1,V2,V3,V4,...,VN);

WRITE A QUERY TO DISPLAY THE NAMES AND DEPTNO OF THE EMPLOYEES WHO ARE not working in deptno 10,20,30

EMP

EMPNO	ENAME	SAL	Ph_No	DEPTNO
123	Stark	2000	3210654987	10
210	Steve	1500	1234567895	20
312	Natasha	5000	3123023698	20
187	Peter	1000	7788994455	40
230	Bruce	900	9988665544	10
364	Thor	2500	3215478961	20

```
SELECT ENAME , DEPTNO
FROM EMP
WHERE DEPTNO NOT IN (10,20,30) ;
```

DEPTNO NOT IN 10,20,30

10 IN 10
20 IN 10 — UNSELECTED
20 IN 10

40 IN 10 — SELECTED

10 IN 10
20 IN 10 — UNSELECTED

10 IN 20
20 IN 20 — UNSELECTED
20 IN 20

40 IN 20 — SELECTED

10 IN 20
20 IN 20 — UNSELECTED

10 IN 30
20 IN 30 — UNSELECTED
20 IN 30

40 IN 30 — SELECTED

10 IN 30
20 IN 30 — UNSELECTED

FINAL O/P:

ENAME	DEPTNO
PETER	40

```
SQL> --WAQTD THE DETAILS OF THE EMP'S WHO ARE NOT WORKING IN DEPTNO 10,20,30
SQL>
SQL> SELECT EMP.*
  2  FROM EMP
  3  WHERE JOB NOT IN('CLERK','MANAGER');

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7499 ALLEN      SALESMAN       7698 20-FEB-81       1600        300         30
      7521 WARD       SALESMAN       7698 22-FEB-81       1250        500         30
      7654 MARTIN     SALESMAN       7698 28-SEP-81       1250       1400         30
      7788 SCOTT      ANALYST        7566 19-APR-87       3000                    20
      7839 KING       PRESIDENT           17-NOV-81       5000                    10
      7844 TURNER     SALESMAN       7698 08-SEP-81       1500          0         30
      7902 FORD       ANALYST        7566 03-DEC-81       3000                    20

7 rows selected.

SQL> --WAQTD THE NAMES OF THE EMP'S WHO ARE NOT WORKING IN DEPTNO 10,20,30
SQL>
SQL> SELECT EMP.*
  2  FROM EMP
  3  WHERE DEPTNO NOT IN(10,20,30);

no rows selected
```

3. BETWEEN Operator :

> BETWEEN Operator is a type of Special Operator which is used whenever we have a Range of Values

> Syntax :

```
COLUMN_NAME/EXPRESSION BETWEEN LOWER_RANGE_VALUE AND HIGHER_RANGE_VALUE;
```

WRITE A QUERY TO DISPLAY THE NAMES AND SALARY OF THE EMP'S WHO IS EARNING SALARY BETWEEN 2000 AND 5000

EMP

EMPNO	ENAME	SAL	Ph_No	DEPTNO
123	Stark	2000	3210654987	10
210	Steve	1500	1234567895	20
312	Natasha	5000	3123023698	20
187	Peter	1000	7788994455	40
230	Bruce	900	9988665544	10
364	Thor	2500	3215478961	20

```
SELECT ENAME , SAL

FROM EMP

WHERE SAL BETWEEN 2001 AND 4999 ;
```

4. NOT BETWEEN Operator :

> NOT BETWEEN Operator is similar to BETWEEN Operator instead of selecting the Records/Values in Range, it will reject the values in the Range

> Syntax :

```
COLUMN_NAME/EXPRESSION NOT BETWEEN LOWER_RANGE_VALUE AND HIGHER_RANGE_VALUE;
```

```
SQL> --WAQTD THE NAMES AND SAL OF THE EMP'S WHO IS EARNING SALARY BETWEEN 2000 AND 5000
SQL>
SQL> SELECT ENAME, SAL
  2  FROM EMP
  3  WHERE SAL BETWEEN 2000 AND 5000;

ENAME             SAL
---------- ----------
JONES            2975
BLAKE            2850
CLARK            2450
SCOTT            3000
KING             5000
FORD             3000

6 rows selected.              ------> NOT EXPECTED ANSWER

SQL> SELECT ENAME, SAL
  2  FROM EMP
  3  WHERE SAL BETWEEN 2001 AND 4999;

ENAME             SAL
---------- ----------
JONES            2975
BLAKE            2850
CLARK            2450
SCOTT            3000
FORD             3000

------------> EXPECTED ANSWER <--------------
```

```
SQL> --WAQTD DETAILS OF EMPLOYEES WHO WERE HIRED AFTER '01-MAY-81' AND BEFORE 03-DEC-81'
SQL>
SQL> SELECT EMP.*
  2  FROM EMP
  3  WHERE HIREDATE BETWEEN '01-MAY-81' AND '03-DEC-81';

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7654 MARTIN     SALESMAN        7698 28-SEP-81       1250       1400         30
      7698 BLAKE      MANAGER         7839 01-MAY-81       2850                    30
      7782 CLARK      MANAGER         7839 09-JUN-81       2450                    10
      7839 KING       PRESIDENT            17-NOV-81       5000                    10
      7844 TURNER     SALESMAN        7698 08-SEP-81       1500          0         30
      7900 JAMES      CLERK           7698 03-DEC-81        950                    30
      7902 FORD       ANALYST         7566 03-DEC-81       3000                    20

7 rows selected.                  ------> NOT EXPECTED ANSWER

SQL> SELECT EMP.*
  2  FROM EMP
  3  WHERE HIREDATE BETWEEN '02-MAY-81' AND '02-DEC-81';

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7654 MARTIN     SALESMAN        7698 28-SEP-81       1250       1400         30
      7782 CLARK      MANAGER         7839 09-JUN-81       2450                    10
      7839 KING       PRESIDENT            17-NOV-81       5000                    10
      7844 TURNER     SALESMAN        7698 08-SEP-81       1500          0         30

------------> EXPECTED ANSWER <--------------
```

```
SQL> --WAQTD THE NAMES OF THE EMP'S WHO IS EARNING SALARY IN THE RANGE OF  2000 AND 5000
SQL>
SQL> SELECT ENAME, SAL
  2  FROM EMP
  3  WHERE SAL BETWEEN 2000 AND 5000;

ENAME           SAL
---------- ----------
JONES           2975
BLAKE           2850
CLARK           2450
SCOTT           3000
KING            5000
FORD            3000

6 rows selected.

SQL> --WAQTD DETAILS OF EMPLOYEES WHO WERE NOT HIRED IN THE RANGE OF  '01-MAY-81' AND 03-DEC-81'
SQL>
SQL> SELECT EMP.*
  2  FROM EMP
  3  WHERE HIREDATE NOT BETWEEN '01-MAY-81' AND '03-DEC-81';

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7369 SMITH      CLERK           7902 17-DEC-80        800                    20
      7499 ALLEN      SALESMAN        7698 20-FEB-81       1600        300         30
      7521 WARD       SALESMAN        7698 22-FEB-81       1250        500         30
      7566 JONES      MANAGER         7839 02-APR-81       2975                    20
      7788 SCOTT      ANALYST         7566 19-APR-87       3000                    20
      7876 ADAMS      CLERK           7788 23-MAY-87       1100                    20
      7934 MILLER     CLERK           7782 23-JAN-82       1300                    10

7 rows selected.
```

```
SQL> --WAQTD THE NAMES AND SALARY OF THE EMP'S WHO IS NOT EARNING SALARY IN THE RANGE BETWEEN 2000 AND 5000
SQL>
SQL> SELECT ENAME, SAL
  2  FROM EMP
  3  WHERE SAL NOT BETWEEN 2001 AND 4999;

ENAME            SAL
---------- ----------
SMITH             800
ALLEN            1600
WARD             1250
MARTIN           1250
KING             5000
TURNER           1500
ADAMS            1100
JAMES             950
MILLER           1300

9 rows selected.

SQL> --WAQTD THE NAMES AND SALARY OF THE EMP'S WHO IS NOT EARNING SALARY IN THE RANGE OF 2000 AND 5000
SQL>
SQL> SELECT ENAME, SAL
  2  FROM EMP
  3  WHERE SAL NOT BETWEEN 2000 AND 5000;

ENAME            SAL
---------- ----------
SMITH             800
ALLEN            1600
WARD             1250
MARTIN           1250
TURNER           1500
ADAMS            1100
JAMES             950
MILLER           1300

8 rows selected.
```

5. IS Operator :

> IS Operator is a type of Special Operator which is used to compare with NULL Values

> Syntax :

```
COLUMN_NAME/EXPRESSION IS NULL;
```

WRITE A QUERY TO DISPLAY THE NAMES AND SALARY OF THE EMP'S WHO IS NOT EARNING SALARY

EMP

EMPNO	ENAME	SAL	Ph_No	DEPTNO
123	Stark	2000	3210654987	10
210	Steve		1234567895	20
312	Natasha	5000	3123023698	20
187	Peter	1000	7788994455	40
230	Bruce		9988665544	10
364	Thor	2500	3215478961	20

```
SELECT ENAME , SAL

FROM EMP

WHERE SAL IS NULL ;
```

```
SQL> -- WAQTD THE DETAILS OF THE EMP'S WHO IS NOT EAARNING COMMISSION
SQL>
SQL> SELECT EMP.*
  2  FROM EMP
  3  WHERE COMM = 0;

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7844 TURNER     SALESMAN       7698 08-SEP-81       1500          0         30

SQL> SELECT EMP.*
  2  FROM EMP
  3  WHERE COMM = NULL;

no rows selected

SQL> SELECT EMP.*
  2  FROM EMP
  3  WHERE COMM IS NULL;

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7369 SMITH      CLERK          7902 17-DEC-80        800                    20
      7566 JONES      MANAGER        7839 02-APR-81       2975                    20
      7698 BLAKE      MANAGER        7839 01-MAY-81       2850                    30
      7782 CLARK      MANAGER        7839 09-JUN-81       2450                    10
      7788 SCOTT      ANALYST        7566 19-APR-87       3000                    20
      7839 KING       PRESIDENT           17-NOV-81       5000                    10
      7876 ADAMS      CLERK          7788 23-MAY-87       1100                    20
      7900 JAMES      CLERK          7698 03-DEC-81        950                    30
      7902 FORD       ANALYST        7566 03-DEC-81       3000                    20
      7934 MILLER     CLERK          7782 23-JAN-82       1300                    10

10 rows selected.

SQL> -- WAQTD THE DEATILS OF THE EMP'S WHO IS NOT HAVING MGR
SQL> SELECT EMP.*
  2  FROM EMP
  3  WHERE MGR IS NULL;

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7839 KING       PRESIDENT           17-NOV-81       5000                    10
```

6. IS NOT Operator :

> IS NOT Operator is similar to IS Operator instead of selecting the NULL Records/Values, it will reject the NULL Values

> Syntax :

```
COLUMN_NAME/EXPRESSION IS NOT NULL;
```

WRITE A QUERY TO DISPLAY THE NAMES AND SALARY OF THE EMP'S WHO IS EARNING SALARY

EMP

EMPNO	ENAME	SAL	Ph_No	DEPTNO
123	Stark	2000	3210654987	10
210	Steve		1234567895	20
312	Natasha	5000	3123023698	20
187	Peter	1000	7788994455	40
230	Bruce		9988665544	10
364	Thor	2500	3215478961	20

```
SELECT ENAME , SAL

FROM EMP

WHERE SAL IS NOT NULL ;
```

```
SQL> -- WAQTD THE DETAILS OF EMP'S WHO IS EARNING COMMISSION
SQL>
SQL> SELECT EMP.*
  2  FROM EMP
  3  WHERE COMM IS NOT NULL;

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7499 ALLEN      SALESMAN        7698 20-FEB-81       1600        300         30
      7521 WARD       SALESMAN        7698 22-FEB-81       1250        500         30
      7654 MARTIN     SALESMAN        7698 28-SEP-81       1250       1400         30
      7844 TURNER     SALESMAN        7698 08-SEP-81       1500          0         30

SQL> -- WAQTD THE DETAILS OF EMP'S WHO IS HAVING MGR
SQL>
SQL> SELECT EMP.*
  2  FROM EMP
  3  WHERE MGR IS NOT NULL;

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7369 SMITH      CLERK           7902 17-DEC-80        800                    20
      7499 ALLEN      SALESMAN        7698 20-FEB-81       1600        300         30
      7521 WARD       SALESMAN        7698 22-FEB-81       1250        500         30
      7566 JONES      MANAGER         7839 02-APR-81       2975                    20
      7654 MARTIN     SALESMAN        7698 28-SEP-81       1250       1400         30
      7698 BLAKE      MANAGER         7839 01-MAY-81       2850                    30
      7782 CLARK      MANAGER         7839 09-JUN-81       2450                    10
      7788 SCOTT      ANALYST         7566 19-APR-87       3000                    20
      7844 TURNER     SALESMAN        7698 08-SEP-81       1500          0         30
      7876 ADAMS      CLERK           7788 23-MAY-87       1100                    20
      7900 JAMES      CLERK           7698 03-DEC-81        950                    30
      7902 FORD       ANALYST         7566 03-DEC-81       3000                    20
      7934 MILLER     CLERK           7782 23-JAN-82       1300                    10

13 rows selected.
```

7. LIKE Operator :

> LIKE Operator is a type of Special Operator which is used for Pattern Matching Process

> It can be achieved with the help of two Characters:

1. % (PERCENTILE) --> it'll accept any No. of character and any No. of time

when to go for % --> when No. of Char/Digits are Unknown

2. _ (UNDERSCORE) --> it'll accept only one character at a time

when to go for _ --> when No. of Char/Digits are Exactly Known

> Syntax :

```
COLUMN_NAME/EXPRESSION LIKE 'PATTERN TO MATCH' ;
```

PATTERN TO MATCH --> It is also called as Wild Cards

> whose name starts with character 'S'

```
--> ENAME LIKE 'S%';
```

> Ends with character 'A' --> ENAME LIKE '%A';

> Whose name contains 'T' in it --> ENAME LIKE '%T%';

> two consecutive characters 'OO' --> ENAME LIKE '%OO%';

> Whose name contains exactly 2 'A' --> ENAME LIKE '%E%E%';

> starts with char 'N' and 3rd char as 'T'
last but one character as 'H' and ends with 'A'

```
--> ENAME LIKE 'N_T%HA';
```

> Who are hired in the month of march --> HIREDATE LIKE '%MAR%';

(OR) HIREDATE LIKE '__-MAR-__';

> Who are hired in the year of 82 --> HIREDATE LIKE '%82'

(OR) HIREDATE LIKE '__-___-82';

WRITE A QUERY TO DISPLAY THE NAMES AND SALARY OF THE EMP'S WHOSE NAME STARTS WITH S

EMP

EMPNO	ENAME	SAL	Ph_No	DEPTNO
123	Stark	2000	3210654987	10
210	Steve		1234567895	20
312	Natasha	5000	3123023698	20
187	Peter	1000	7788994455	40
230	Bruce		9988665544	10
364	Dr.Doom	2500	3215478961	20

```
SELECT ENAME , SAL

FROM EMP

WHERE ENAME LIKE 'S%' ;
```

```
SQL> -- WAQTD THE DETAILS OF THE EMP'S WHOSE NAME STARTS WITH CHARACTER S
SQL>
SQL> SELECT EMP.*
  2  FROM EMP
  3  WHERE ENAME LIKE 'S%';

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7369 SMITH      CLERK           7902 17-DEC-80        800                    20
      7788 SCOTT      ANALYST         7566 19-APR-87       3000                    20

SQL> -- WAQTD THE DETAILS OF EMP'S WHOSE NAME ENDS WITH CHARACTER N
SQL>
SQL> SELECT EMP.*
  2  FROM EMP
  3  WHERE ENAME LIKE '%N';

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7499 ALLEN      SALESMAN        7698 20-FEB-81       1600        300         30
      7654 MARTIN     SALESMAN        7698 28-SEP-81       1250       1400         30
```

```
SQL> -- WAQTD THE DETAILS OF EMP'S WHO IS HIRED IN THE MONTH DEC
SQL> SELECT EMP.*
  2  FROM EMP
  3  WHERE HIREDATE LIKE '%DEC%';

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7369 SMITH      CLERK           7902 17-DEC-80        800                    20
      7900 JAMES      CLERK           7698 03-DEC-81        950                    30
      7902 FORD       ANALYST         7566 03-DEC-81       3000                    20

SQL> SELECT EMP.*
  2  FROM EMP
  3  WHERE HIREDATE LIKE '__-DEC-__';

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7369 SMITH      CLERK           7902 17-DEC-80        800                    20
      7900 JAMES      CLERK           7698 03-DEC-81        950                    30
      7902 FORD       ANALYST         7566 03-DEC-81       3000                    20

SQL> -- WAQTD THE DETAILS OF EMP'S WHO IS HIRED IN THE YEAR 81
SQL>
SQL> SELECT EMP.*
  2  FROM EMP
  3  WHERE HIREDATE LIKE '%81';

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7499 ALLEN      SALESMAN        7698 20-FEB-81       1600        300         30
      7521 WARD       SALESMAN        7698 22-FEB-81       1250        500         30
      7566 JONES      MANAGER         7839 02-APR-81       2975                    20
      7654 MARTIN     SALESMAN        7698 28-SEP-81       1250       1400         30
      7698 BLAKE      MANAGER         7839 01-MAY-81       2850                    30
      7782 CLARK      MANAGER         7839 09-JUN-81       2450                    10
      7839 KING       PRESIDENT            17-NOV-81       5000                    10
      7844 TURNER     SALESMAN        7698 08-SEP-81       1500          0         30
      7900 JAMES      CLERK           7698 03-DEC-81        950                    30
      7902 FORD       ANALYST         7566 03-DEC-81       3000                    20

10 rows selected.
```

8. NOT LIKE Operator :

> NOT LIKE Operator is similar to LIKE Operator instead of selecting the Pattern, it will reject the Pattern

> Syntax :

```
COLUMN_NAME/EXPRESSION NOT LIKE 'PATTERN TO MATCH' ;
```

WRITE A QUERY TO DISPLAY THE NAMES AND SALARY OF THE EMP'S WHOSE NAME DOES NOT ENDS WITH E

EMP

EMPNO	ENAME	SAL	Ph_No	DEPTNO
123	Stark	2000	3210654987	10
210	Steve		1234567895	20
312	Natasha	5000	3123023698	20
187	Peter	1000	7788994455	40
230	Bruce		9988665544	10
364	Dr.Doom	2500	3215478961	20

```
SELECT ENAME , SAL

FROM EMP

WHERE ENAME NOT LIKE '%E'  ;
```

```
SQL> -- WAQTD THE DETAILS OF THE EMP'S WHOSE NAME DOESN'T STARTS WITH CHARACTER A
SQL> SELECT EMP.*
  2  FROM EMP
  3  WHERE ENAME NOT LIKE 'A%';

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7369 SMITH      CLERK           7902 17-DEC-80        800                    20
      7521 WARD       SALESMAN        7698 22-FEB-81       1250        500         30
      7566 JONES      MANAGER         7839 02-APR-81       2975                    20
      7654 MARTIN     SALESMAN        7698 28-SEP-81       1250       1400         30
      7698 BLAKE      MANAGER         7839 01-MAY-81       2850                    30
      7782 CLARK      MANAGER         7839 09-JUN-81       2450                    10
      7788 SCOTT      ANALYST         7566 19-APR-87       3000                    20
      7839 KING       PRESIDENT            17-NOV-81       5000                    10
      7844 TURNER     SALESMAN        7698 08-SEP-81       1500          0         30
      7900 JAMES      CLERK           7698 03-DEC-81        950                    30
      7902 FORD       ANALYST         7566 03-DEC-81       3000                    20
      7934 MILLER     CLERK           7782 23-JAN-82       1300                    10

12 rows selected.

SQL> -- WAQTD THE DETAILS OF EMP'S WHOSE NAME DOESN'T ENDS WITH CHARACTER S
SQL>
SQL> SELECT EMP.*
  2  FROM EMP
  3  WHERE ENAME NOT LIKE '%S';

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7369 SMITH      CLERK           7902 17-DEC-80        800                    20
      7499 ALLEN      SALESMAN        7698 20-FEB-81       1600        300         30
      7521 WARD       SALESMAN        7698 22-FEB-81       1250        500         30
      7654 MARTIN     SALESMAN        7698 28-SEP-81       1250       1400         30
      7698 BLAKE      MANAGER         7839 01-MAY-81       2850                    30
      7782 CLARK      MANAGER         7839 09-JUN-81       2450                    10
      7788 SCOTT      ANALYST         7566 19-APR-87       3000                    20
      7839 KING       PRESIDENT            17-NOV-81       5000                    10
      7844 TURNER     SALESMAN        7698 08-SEP-81       1500          0         30
      7902 FORD       ANALYST         7566 03-DEC-81       3000                    20
      7934 MILLER     CLERK           7782 23-JAN-82       1300                    10

11 rows selected.
```

```
SQL> -- WAQTD THE DETAILS OF EMP'S WHO IS NOT HIRED IN THE MONTH DEC
SQL>
SQL> SELECT EMP.*
  2  FROM EMP
  3  WHERE HIREDATE NOT LIKE '%DEC%';

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7499 ALLEN      SALESMAN        7698 20-FEB-81       1600        300         30
      7521 WARD       SALESMAN        7698 22-FEB-81       1250        500         30
      7566 JONES      MANAGER         7839 02-APR-81       2975                    20
      7654 MARTIN     SALESMAN        7698 28-SEP-81       1250       1400         30
      7698 BLAKE      MANAGER         7839 01-MAY-81       2850                    30
      7782 CLARK      MANAGER         7839 09-JUN-81       2450                    10
      7788 SCOTT      ANALYST         7566 19-APR-87       3000                    20
      7839 KING       PRESIDENT            17-NOV-81       5000                    10
      7844 TURNER     SALESMAN        7698 08-SEP-81       1500          0         30
      7876 ADAMS      CLERK           7788 23-MAY-87       1100                    20
      7934 MILLER     CLERK           7782 23-JAN-82       1300                    10

11 rows selected.

SQL> -- WAQTD THE DETAILS OF EMP'S WHO IS NOT HIRED IN THE YEAR 81
SQL>
SQL> SELECT EMP.*
  2  FROM EMP
  3  WHERE HIREDATE NOT LIKE '%81';

     EMPNO ENAME      JOB              MGR HIREDATE         SAL       COMM     DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
      7369 SMITH      CLERK           7902 17-DEC-80        800                    20
      7788 SCOTT      ANALYST         7566 19-APR-87       3000                    20
      7876 ADAMS      CLERK           7788 23-MAY-87       1100                    20
      7934 MILLER     CLERK           7782 23-JAN-82       1300                    10

SQL>
```