CLICK FOR NOTES

GALGOTIAS UNIVERSITY
(Established under Galgotias University Uttar Pradesh Act No. 14 of 2011)

Question Bank for DBMS of Unit 1 and Unit2

1. What is the full form of DBMS?
   database management system

2. What is the objective to study DBMS database?

   The objectives of DBMS can be narrated as follows:

- Eliminate redundant data.
- Make access to the data easy for the user.
- Provide for mass storage of relevant data.
- Protect the data from physical harm and un-authorised systems.
- Allow for growth in the data base system.
- Make the latest modifications to the data base available immediately.
- Allow for multiple users to be active at one time.
- Provide prompt response to user requests for data.

3. What is tuple?

In **Database Management System** (DBMS), most of the time we need to store the data in tabular format . This kind of data storage model is also called a **Relational model** and the system which leverages the relational model is called **Relational Database Management System** (RDBMS). These relations (or tables) consist of rows and columns. But in DBMS, we call these rows **"Tuples"** and a row **"Tuple"**.

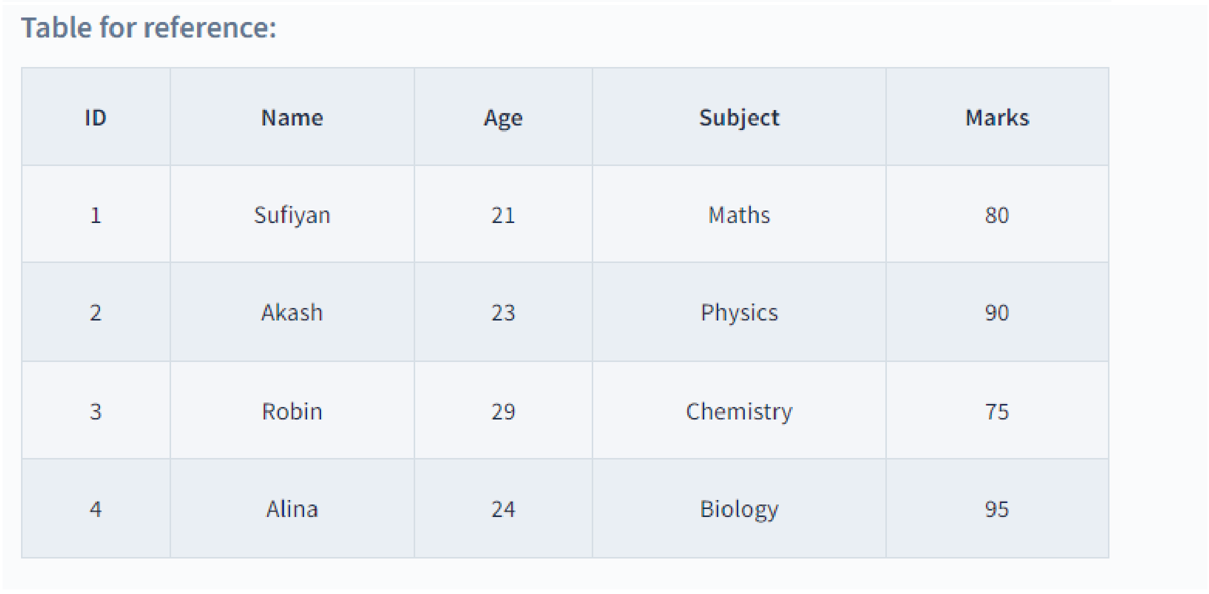


Table for reference:

ID	Name	Age	Subject	Marks
1	Sufiyan	21	Maths	80
2	Akash	23	Physics	90
3	Robin	29	Chemistry	75
4	Alina	24	Biology	95

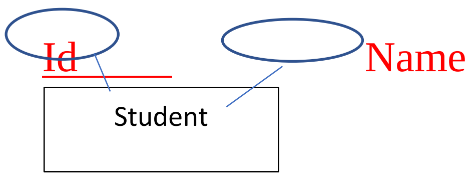
4. Degree of Relationship is ~~defined by~~

In DBMS, a degree of relationship represents the number of entity types that associate in a relationship. For example, we have two entities, one is a student and the other is a bag and they are connected with the primary key and foreign key. So, here we can see that the degree of relationship is 2 as 2 entities are associating in a relationship.

Types of degree

Now, based on the number of linked entity types, we have 4 types of degrees of relationships.

1. Unary
2. Binary
3. Ternary
4. N-ary



5. Here, which is key ?

   ID

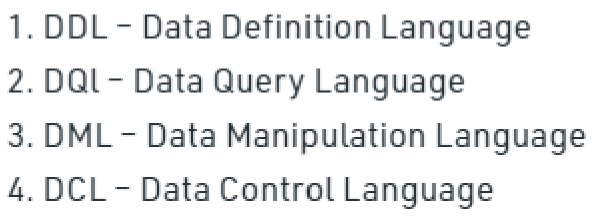
6. Properties/characteristics that describ~~e entit~~ies is called

   Attribute. A characteristic or trait of an entity type that describes the entity, for example, the Person entity type has the Date of Birth attribute.

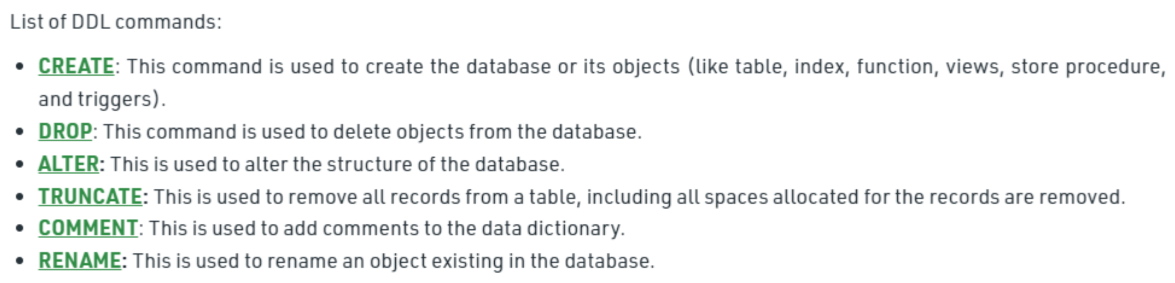
7. What is the full form of DCL?

   Data Control Language

**-extra



1. DDL – Data Definition Language
2. DQL – Data Query Language
3. DML – Data Manipulation Language
4. DCL – Data Control Language



List of DDL commands:

- **CREATE**: This command is used to create the database or its objects (like table, index, function, views, store procedure, and triggers).
- **DROP**: This command is used to delete objects from the database.
- **ALTER**: This is used to alter the structure of the database.
- **TRUNCATE**: This is used to remove all records from a table, including all spaces allocated for the records are removed.
- **COMMENT**: This is used to add comments to the data dictionary.
- **RENAME**: This is used to rename an object existing in the database.

List of DQL:

- **SELECT**: It is used to retrieve data from the database.

List of DML commands:

- **INSERT** : It is used to insert data into a table.
- **UPDATE:** It is used to update existing data within a table.
- **DELETE** : It is used to delete records from a database table.
- **LOCK:** Table control concurrency.
- **CALL:** Call a PL/SQL or JAVA subprogram.
- **EXPLAIN PLAN:** It describes the access path to data.

List of DCL commands:

- **GRANT:** This command gives users access privileges to the database.
- **REVOKE:** This command withdraws the user's access privileges given by using the GRANT command.

List of TCL commands:

- **COMMIT:** Commits a Transaction.
- **ROLLBACK:** Rollbacks a transaction in case of any error occurs.
- **SAVEPOINT:**Sets a savepoint within a transaction.
- **SET TRANSACTION:** Specify characteristics for the transaction.

8. CREATE is ______

a. DDL b. DML

c. DCL d. TCL

9. ALTER is ______

a. DDL b. DML

c. DCL d. TCL

10. UPDATE is ______

a. DDL b. DML

c. DCL d. TCL

11. COMMIT is

a. DDL b. DML

c. DCL d. TCL

12. ROLLBACK is

a. DDL b. DML

c. DCL d. TCL

13. SELECT is

a. DDL b. DML

c. DCL d. TCL

14. .INSERT is

a. DDL b. DML

c. DCL d. TCL

15. Which one of the following is a set of one or more attributes taken collectively to uniquely identify a record?

a. Candidate key b.Subkey

c. Super key d. Foreign key

16. An attribute in a relation is a foreign key if the ________ key from one relation is used as an attribute in that relation.

a. Candidate

b. Primary

c. Super

d. Sub

17. GRANT is

a. DDL b. DML

c. DCL d. TCL

18. REVOKE is

a. DDL b. DML

c. DCL d. TCL

19. A is all those set of attributes which can uniquely identify a row. However, any subset of these set of attributes would not identify a row uniquely.

candidate key

Primary Key

Foreign Key

All these

20. What is the full form of TCL?

a. Transaction Control Language b. Tele Communication Language

c. Transaction Connection Language d. None of these

21. Write difference between file system and Database system.

file system	Database system
• A file system is a software that manages and organizes the files in a storage medium. It controls how data is stored and retrieved. • The file system provides the details of data representation and storage of data. • Storing and retrieving of data can’t be done efficiently in a file system. • It does not offer data recovery processes. • The file system doesn’t have a crash recovery mechanism. • Protecting a file system is very difficult. • In a file management system, the redundancy of data is greater. • The file system offers lesser security. • File System allows you to stores the data as isolated data files and entities. • It doesn’t offer backup and recovery of data if it is lost. • There is no efficient query processing in the file system.	• DBMS or Database Management System is a software application. It is used for accessing, creating, and managing databases. • DBMS gives an abstract view of data that hides the details • DBMS is efficient to use as there are a wide variety of methods to store and retrieve data. • There is a backup recovery for data in DBMS. • DBMS provides a crash recovery mechanism • DBMS offers good protection mechanism. • The redundancy of data is low in the DBMS system. • Data inconsistency is low in a database management system. • Database Management System stores data as well as defined constraints and interrelation. • DBMS system provides backup and recovery of data even if it is lost. • You can easily query data in a database using the SQL language.

22. Explain one-tier, two-tier and three-tier architecture of DBMS.

A **Database Architecture** is a representation of DBMS design. It helps to design, develop, implement, and maintain the database management system. A DBMS architecture allows dividing the database system into individual components that can be independently modified, changed, replaced, and altered. It also helps to understand the components of a database.

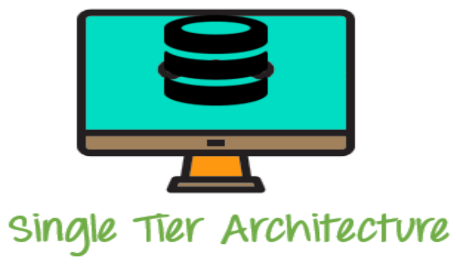
Types of DBMS Architecture

There are mainly three types of DBMS architecture:

- One Tier Architecture (Single Tier Architecture)
- Two Tier Architecture
- Three Tier Architecture

1-Tier Architecture

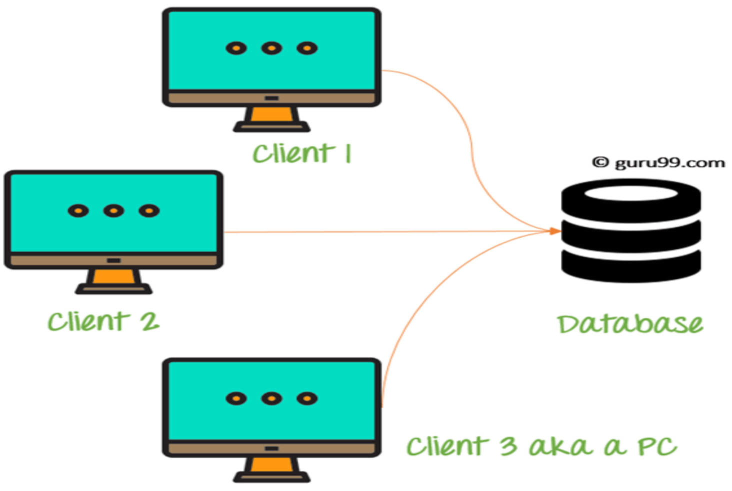
1 Tier Architecture in DBMS is the simplest architecture of Database in which the client, server, and Database all reside on the same machine. A simple one tier architecture example would be anytime you install a Database in your system and access it to practice SQL queries. But such architecture is rarely used in production.



1 Tier Architecture Diagram

2-Tier Architecture

A **2 Tier Architecture** in DBMS is a Database architecture where the presentation layer runs on a client (PC, Mobile, Tablet, etc.), and data is stored on a server called the second tier. Two tier architecture provides added security to the DBMS as it is not exposed to the end-user directly. It also provides direct and faster communication.



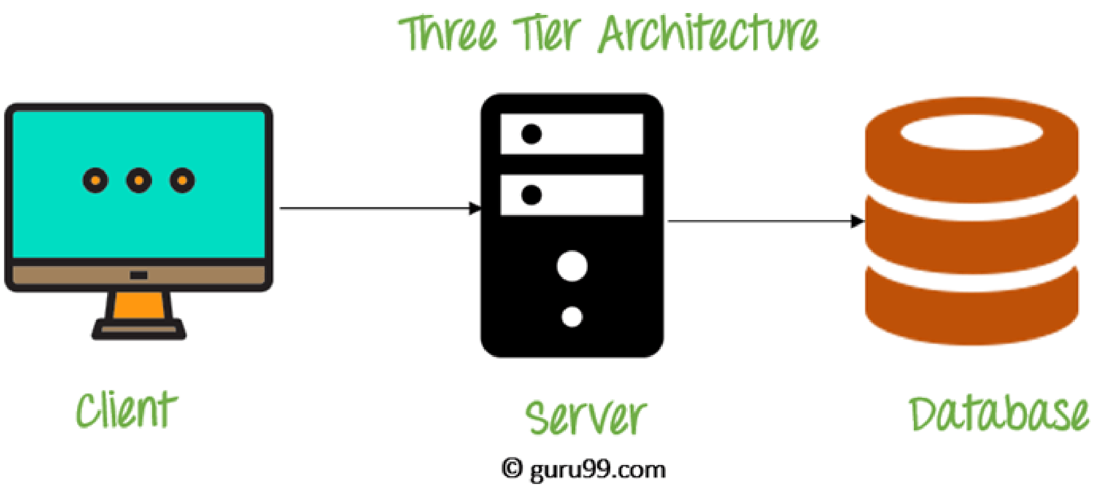
2 Tier Architecture Diagram

3-Tier Architecture

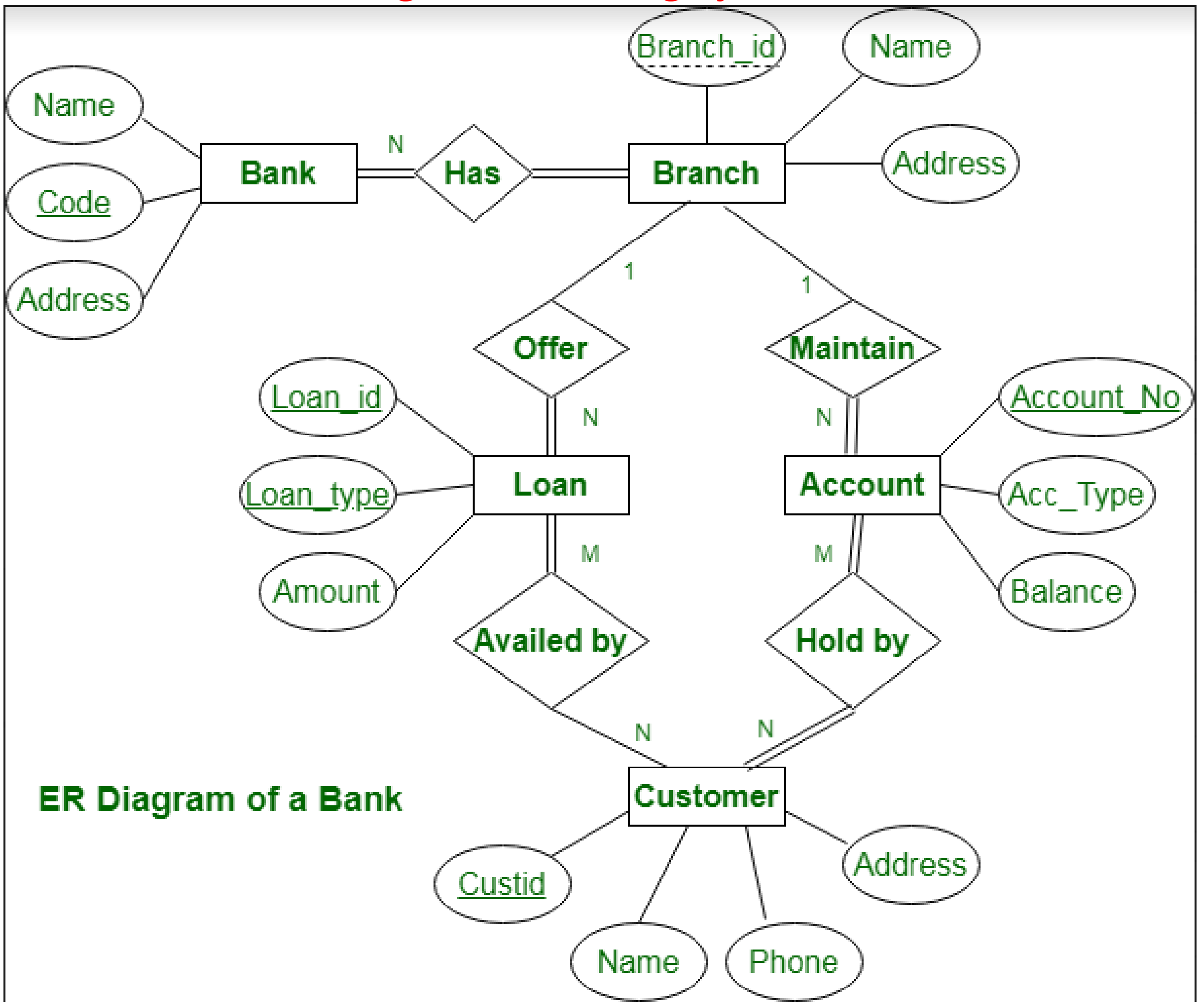
A **3 Tier Architecture** in DBMS is the most popular client server architecture in DBMS in which the development and maintenance of functional processes, logic, data access, data storage, and user interface is done independently as separate modules. Three Tier architecture contains a presentation layer, an application layer, and a database server.

3-Tier database Architecture design is an extension of the 2-tier client-server architecture. A 3-tier architecture has the following layers:

1. Presentation layer (your PC, Tablet, Mobile, etc.)
2. Application layer (server)
3. Database Server



23. Draw the E-R diagram of Banking System.



24. Explain Cartesian Product in DBMS with example.

Cartesian product operation

It combines R1 and R2 without any condition. It is denoted by X.

Degree of R1 XR2 = degree of R1 + degree of R2

{degree = total no of columns}

Example

Consider R1 table –

RegNo	Branch	Section
1	CSE	A
2	ECE	B
3	CIVIL	A
4	IT	B

Table R2

Name	RegNo
Bhanu	2
Priya	4

R1 X R2

RegNo	Branch	Section	Name	RegNo
1	CSE	A	Bhanu	2
1	CSE	A	Priya	4
2	ECE	B	Bhanu	2
2	ECE	B	Priya	4
3	CIVIL	A	Bhanu	2
3	CIVIL	A	Priya	4
4	IT	B	Bhanu	2
4	IT	B	Priya	4

25. Write SQL query to retrieve the Name and Address of Student from Student table.

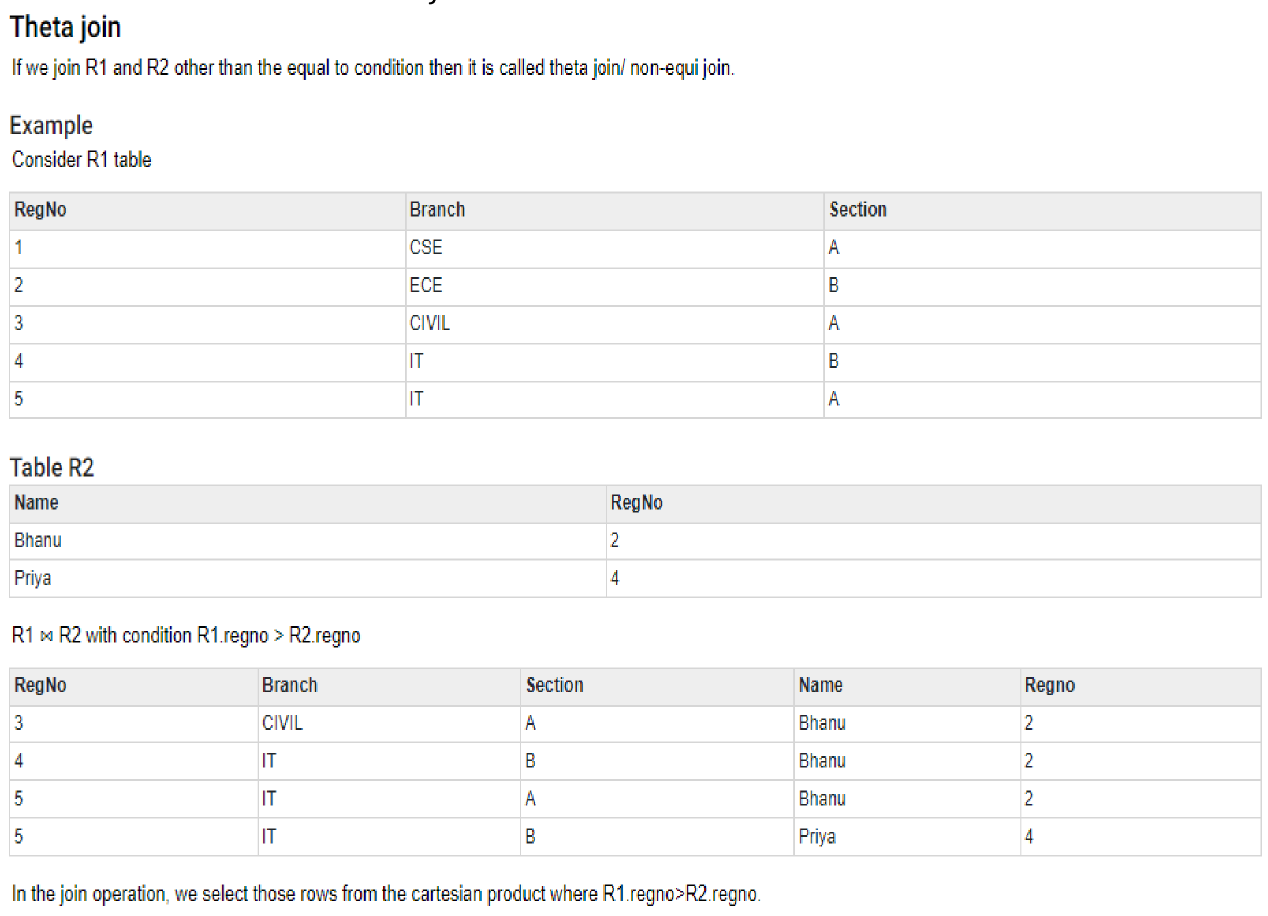
SELECT Name, Address from Students;

26. Explain join in Relational Algebra

Join operation combines the relation R1 and R2 with respect to a condition. It is denoted by ⋈.

The different types of join operation are as follows –

- Theta join
- Natural join
- Outer join – It is further classified into following types –
  - Left outer join.
  - Right outer join.
  - Full outer join.



Theta join

If we join R1 and R2 other than the equal to condition then it is called theta join/ non-equi join.

Example

Consider R1 table

RegNo	Branch	Section
1	CSE	A
2	ECE	B
3	CIVIL	A
4	IT	B
5	IT	A

Table R2

Name	RegNo
Bhanu	2
Priya	4

R1 ⋈ R2 with condition R1.regno > R2.regno

RegNo	Branch	Section	Name	Regno
3	CIVIL	A	Bhanu	2
4	IT	B	Bhanu	2
5	IT	A	Bhanu	2
5	IT	B	Priya	4

In the join operation, we select those rows from the cartesian product where R1.regno>R2.regno.

Natural join

If we join R1 and R2 on equal condition then it is called natural join or equi join. Generally, join is referred to as natural join.

Natural join of R1 and R2 is −

{ we select those tuples from cartesian product where R1.regno=R2.regno}

R1 ⋈ R2

Regno	Branch	Section	Name
2	-	-	Bhanu
4	-	-	priya

Outer join

It is an extension of natural join to deal with missing values of relation.

Consider R1 and R2 shown below −

Table R1

RegNo	Branch	Section
1	CSE	A
2	ECE	B
3	CIVIL	A
4	IT	B
5	IT	A

Table R2

Name	Regno
Bhanu	2
Priya	4
Hari	7

Outer join is of three types. These are explained below −

Left outer join

It is denoted by R1 ⟕ R2.

RegNo	Branch	Section	Name	Regno
2	-	-	Bhanu	2
4	-	-	Priya	4
1	-	-	NULL	NULL
3	-	-	NULL	NULL
5	-	-	NULL	NULL

Here all the tuples of R1(left table) appear in output.

The mismatching values of R2 are filled with NULL.

Left outer join = natural join + mismatch / extra tuple of R1

Full outer join

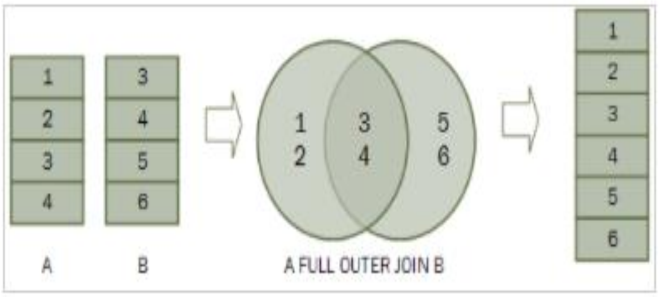
It is denoted by R1 ⋈ R2.

Full outer join=left outer join U right outer join.

RegNo	Branch	Section	Name	Regno
2	-	-	Bhanu	2
4	-	-	Priya	4
1	-	-	NULL	NULL
3	-	-	NULL	NULL
5	-	-	NULL	NULL
NULL	NULL	NULL	Hari	7

Example

Given below is the picture of full outer join −



27. Write SQL query to get the Students Name from Student table whose Address is Greater Noida.
SELECT Name from Students WHERE Address=’ Greater Noida’;

28. Write SQL query to retrieve the name of students whose age is between 10 and 20.
select name from students where age between 10 and 20;

29. Write SQL query to retrieve the name of students whose age is greater than 20.
select name from students where age >20;

30. Write SQL query to retrieve the name of student has got the maximum marks.
select name ,max(marks) as max_marks from student;

31. Write SQL query to retrieve the name of student has got the minimum marks.

select name ,min(marks) as min_marks from student;

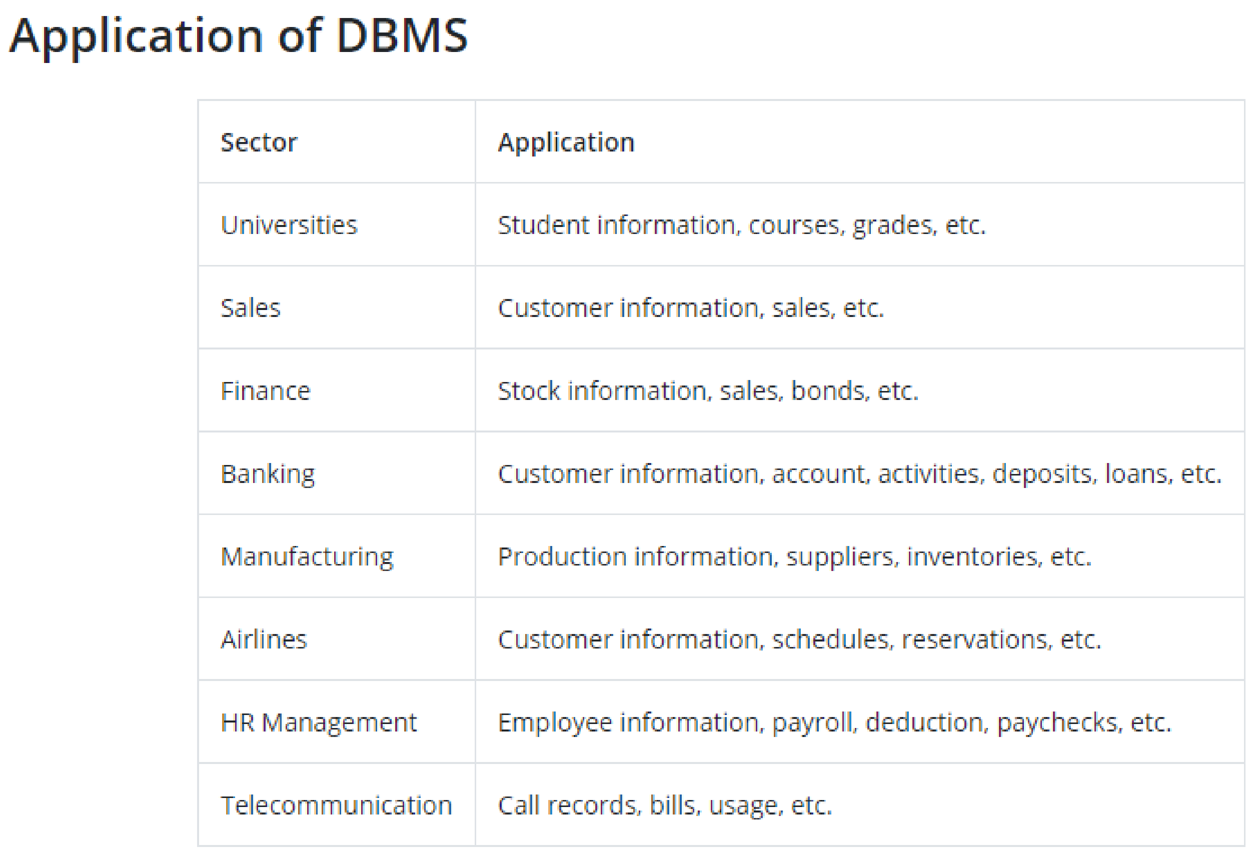
32. Write SQL query to retrieve the name of student whose address in Greater Noida.

select name from student where address in (‘Greater Noida’);

33. Write SQL query to retrieve the name of student has got the maximum marks.

select name ,max(marks) as max_marks from student;

34. What are the applications of DBMS?



Application of DBMS

Sector	Application
Universities	Student information, courses, grades, etc.
Sales	Customer information, sales, etc.
Finance	Stock information, sales, bonds, etc.
Banking	Customer information, account, activities, deposits, loans, etc.
Manufacturing	Production information, suppliers, inventories, etc.
Airlines	Customer information, schedules, reservations, etc.
HR Management	Employee information, payroll, deduction, paychecks, etc.
Telecommunication	Call records, bills, usage, etc.

35. Explain Select in Relational Algebra with example.

Select operation chooses the subset of tuples from the relation that satisfies the given condition mentioned in the syntax of selection. The selection operation is also known as horizontal partitioning since it partitions the table or relation horizontally.

Notation:

$\sigma_{c}(R)$

where 'c' is selection condition which is a boolean expression(condition), we can have a single condition like Roll= 3 or combination of condition like X>2 AND Y<1,
symbol 'σ (sigma)' is used to denote select(choose) operator,

R is a relational algebra expression, whose result is a relation. The boolean expression specified in condition 'c' can be written in the following form:

```
<attribute name> <comparison operator> <constant value> or <attribute name>
```

where, <attribute name> is obviously name of an attribute of relation,

<comparison operator> is any of the operator {<, >, =, <=, >=, !=} and,
<constant value> is constant value taken from the domain of the relation.

Example-1:

$\sigma_{\text{Place = 'Mumbai' or Salary >= 1000000}}(\text{Citizen})$

$\sigma_{\text{Department = 'Analytics'}}(\sigma_{\text{Location = 'NewYork'}}(\text{Manager}))$

36. Explain Project in Relational Algebra with example.

Project operation selects (or chooses) certain attributes discarding other attributes. The Project operation is also known as vertical partitioning since it partitions the relation or table vertically discarding other columns or attributes.

Notation:

$\pi_{A}(R)$

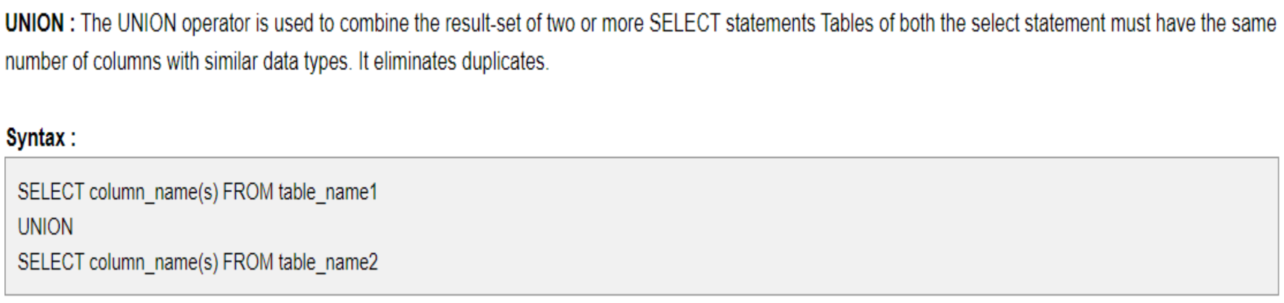
where 'A' is the attribute list, it is the desired set of attributes from the attributes of relation(R),
symbol 'π(pi)' is used to denote the Project operator,
R is generally a relational algebra expression, which results in a relation.

Example –

$\pi_{\text{Age}}(\text{Student})$

$\pi_{\text{Dept, Sex}}(\text{Emp})$

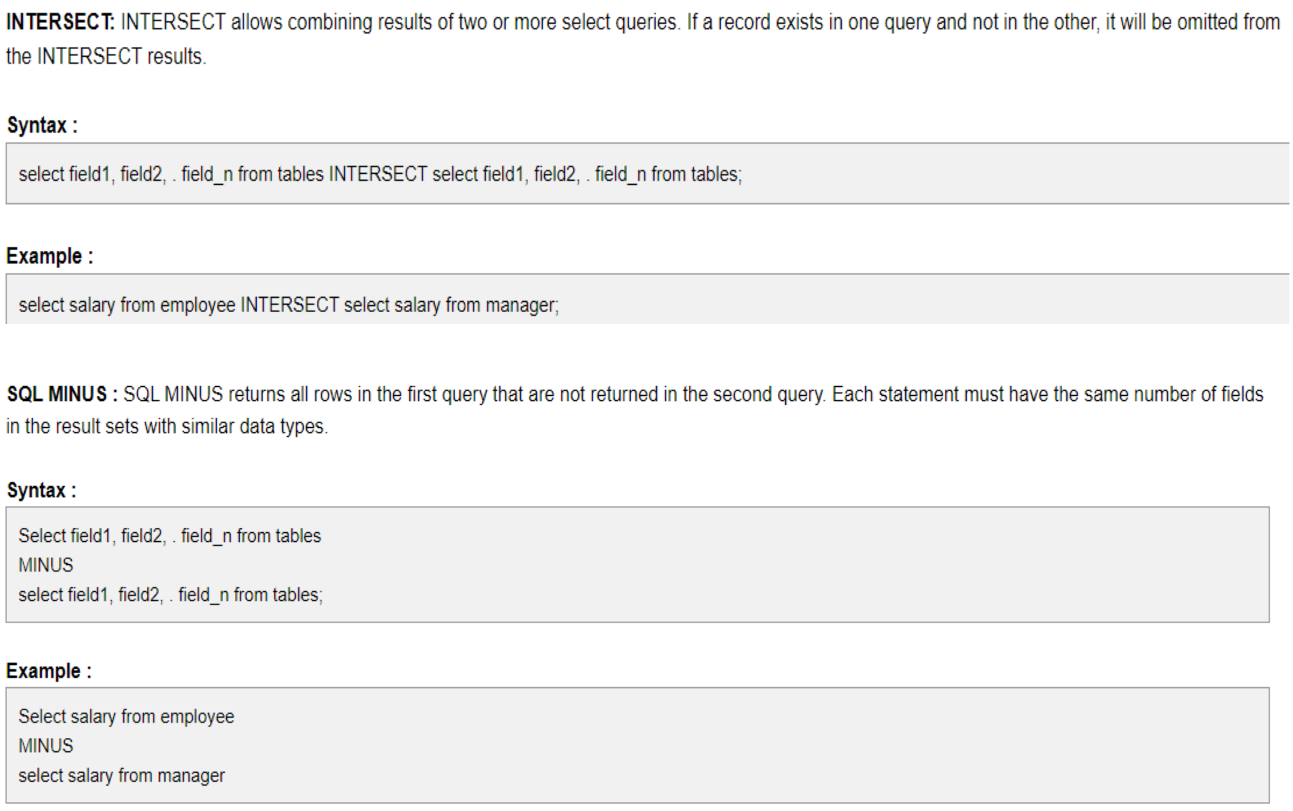
37. Explain Union, Intersection, Minus with example.



UNION : The UNION operator is used to combine the result-set of two or more SELECT statements Tables of both the select statement must have the same number of columns with similar data types. It eliminates duplicates.

Syntax :

```
SELECT column_name(s) FROM table_name1
UNION
SELECT column_name(s) FROM table_name2
```



INTERSECT: INTERSECT allows combining results of two or more select queries. If a record exists in one query and not in the other, it will be omitted from the INTERSECT results.

Syntax :

```
select field1, field2, . field_n from tables INTERSECT select field1, field2, . field_n from tables;
```

Example :

```
select salary from employee INTERSECT select salary from manager;
```

SQL MINUS : SQL MINUS returns all rows in the first query that are not returned in the second query. Each statement must have the same number of fields in the result sets with similar data types.

Syntax :

```
Select field1, field2, . field_n from tables
MINUS
select field1, field2, . field_n from tables;
```

Example :

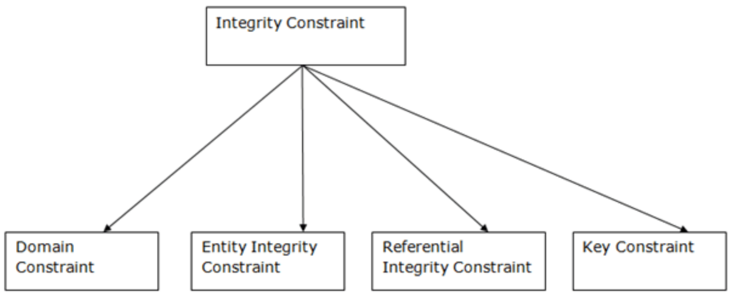
```
Select salary from employee
MINUS
select salary from manager
```

38. Explain integrity constraints, entity integrity, referential integrity, Keys constraints, Domain constraints.

Integrity Constraints

- Integrity constraints are a set of rules. It is used to maintain the quality of information.
- Integrity constraints ensure that the data insertion, updating, and other processes have to be performed in such a way that data integrity is not affected.
- Thus, integrity constraint is used to guard against accidental damage to the database.

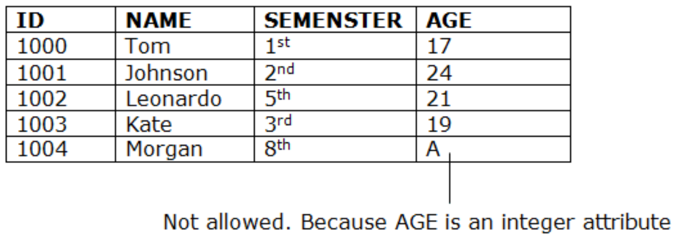
Types of Integrity Constraint



1. Domain constraints

- Domain constraints can be defined as the definition of a valid set of values for an attribute.
- The data type of domain includes string, character, integer, time, date, currency, etc. The value of the attribute must be available in the corresponding domain.

Example:



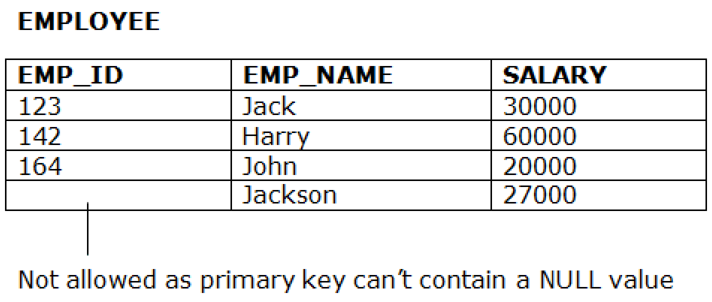
ID	NAME	SEMENSTER	AGE
1000	Tom	1st	17
1001	Johnson	2nd	24
1002	Leonardo	5th	21
1003	Kate	3rd	19
1004	Morgan	8th	A

Not allowed. Because AGE is an integer attribute

2. Entity integrity constraints

- The entity integrity constraint states that primary key value can't be null.
- This is because the primary key value is used to identify individual rows in relation and if the primary key has a null value, then we can't identify those rows.
- A table can contain a null value other than the primary key field.

Example:



EMPLOYEE

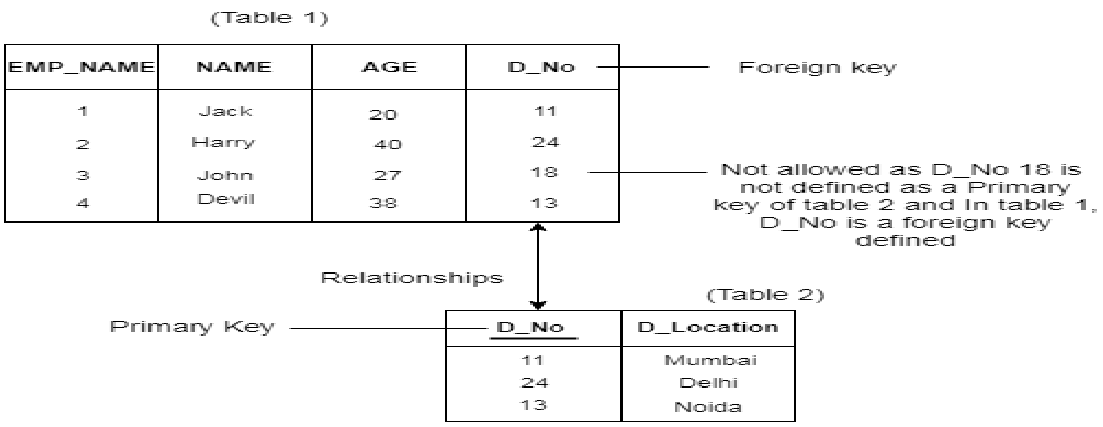
EMP_ID	EMP_NAME	SALARY
123	Jack	30000
142	Harry	60000
164	John	20000
	Jackson	27000

Not allowed as primary key can't contain a NULL value

3. Referential Integrity Constraints

- A referential integrity constraint is specified between two tables.
- In the Referential integrity constraints, if a foreign key in Table 1 refers to the Primary Key of Table 2, then every value of the Foreign Key in Table 1 must be null or be available in Table 2.

Example:


Foreign key
Not allowed as D_No 18 is not defined as a Primary key of table 2 and In table 1, D_No is a foreign key defined
Relationships
Primary Key

(Table 1)

EMP_NAME	NAME	AGE	D_No
1	Jack	20	11
2	Harry	40	24
3	John	27	18
4	Devil	38	13

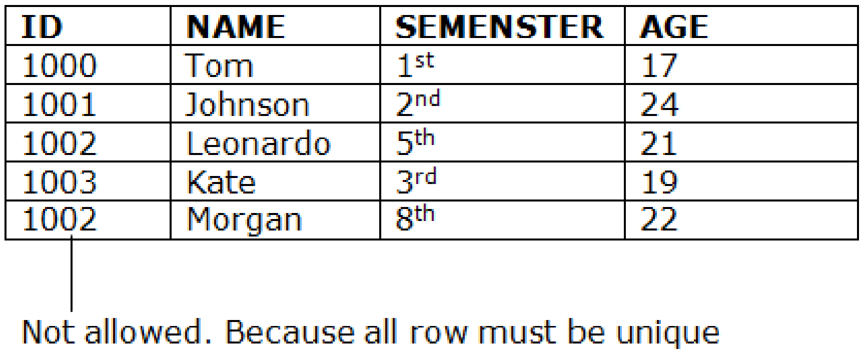
(Table 2)

D_No	D_Location
11	Mumbai
24	Delhi
13	Noida

4. Key constraints

- Keys are the entity set that is used to identify an entity within its entity set uniquely.
- An entity set can have multiple keys, but out of which one key will be the primary key. A primary key can contain a unique and null value in the relational table.

Example:



ID	NAME	SEMENSTER	AGE
1000	Tom	1st	17
1001	Johnson	2nd	24
1002	Leonardo	5th	21
1003	Kate	3rd	19
1002	Morgan	8th	22

Not allowed. Because all row must be unique

39. Explain SQL operators.

What is an Operator in SQL?

An operator is a reserved word or a character used primarily in an SQL statement's WHERE clause to perform operation(s), such as comparisons and arithmetic operations. These Operators are used to specify conditions in an SQL statement and to serve as conjunctions for multiple conditions in a statement.

- Arithmetic operators
- Comparison operators
- Logical operators
- Operators used to negate conditions

SQL Arithmetic Operators

Assume **'variable a'** holds 10 and **'variable b'** holds 20, then

Operator	Description	Example
+ (Addition)	Adds values on either side of the operator.	a + b will give 30
- (Subtraction)	Subtracts right hand operand from left hand operand.	a - b will give -10
* (Multiplication)	Multiplies values on either side of the operator.	a * b will give 200
/ (Division)	Divides left hand operand by right hand operand.	b / a will give 2
% (Modulus)	Divides left hand operand by right hand operand and returns remainder.	b % a will give 0

SQL Comparison Operators

Assume **'variable a'** holds 10 and **'variable b'** holds 20, then –

Operator	Description	Example
=	Checks if the values of two operands are equal or not, if yes then condition becomes true.	(a = b) is not true.
!=	Checks if the values of two operands are equal or not, if values are not equal then condition becomes true.	(a != b) is true.
<>	Checks if the values of two operands are equal or not, if values are not equal then condition becomes true.	(a <> b) is true.
>	Checks if the value of left operand is greater than the value of right operand, if yes then condition becomes true.	(a > b) is not true.
<	Checks if the value of left operand is less than the value of right operand, if yes then condition becomes true.	(a < b) is true.
>=	Checks if the value of left operand is greater than or equal to the value of right operand, if yes then condition becomes true.	(a >= b) is not true.
<=	Checks if the value of left operand is less than or equal to the value of right operand, if yes then condition becomes true.	(a <= b) is true.
!<	Checks if the value of left operand is not less than the value of right operand, if yes then condition becomes true.	(a !< b) is false.
!>	Checks if the value of left operand is not greater than the value of right operand, if yes then condition becomes true.	(a !> b) is true.

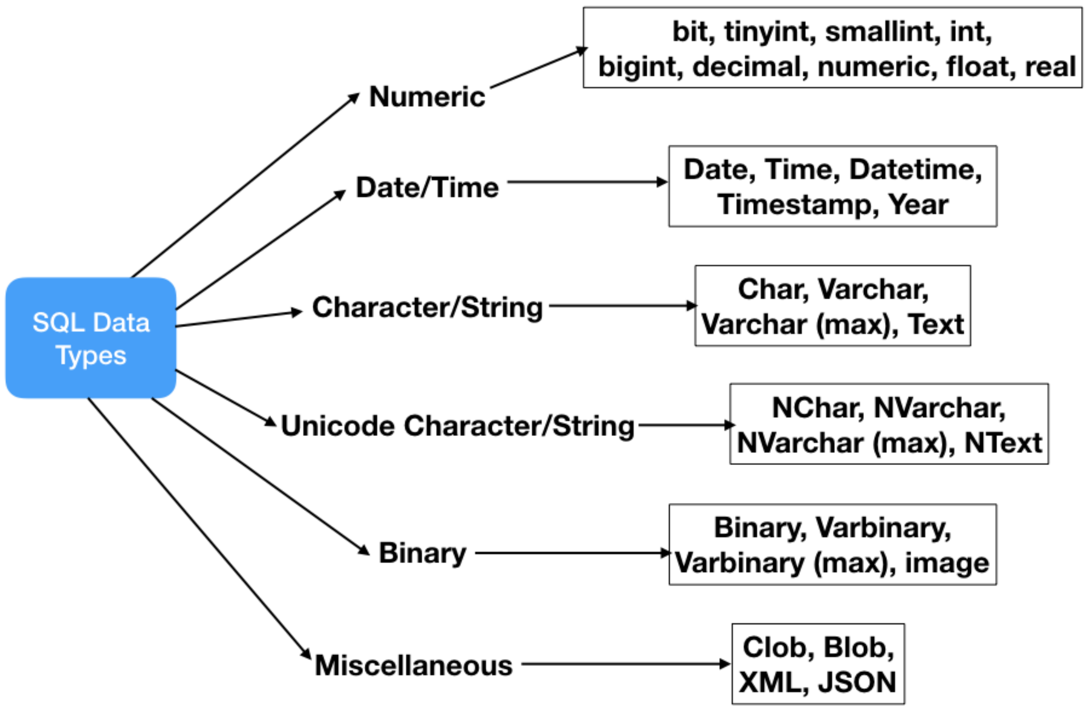
SQL Logical Operators

Here is a list of all the logical operators available in SQL.

Sr.No.	Operator & Description
1	**ALL** The ALL operator is used to compare a value to all values in another value set.
2	**AND** The AND operator allows the existence of multiple conditions in an SQL statement's WHERE clause.
3	**ANY** The ANY operator is used to compare a value to any applicable value in the list as per the condition.
4	**BETWEEN** The BETWEEN operator is used to search for values that are within a set of values, given the minimum value and the maximum value.
5	**EXISTS** The EXISTS operator is used to search for the presence of a row in a specified table that meets a certain criterion.
6	**IN** The IN operator is used to compare a value to a list of literal values that have been specified.
7	**LIKE** The LIKE operator is used to compare a value to similar values using wildcard operators.
8	**NOT** The NOT operator reverses the meaning of the logical operator with which it is used. Eg: NOT EXISTS, NOT BETWEEN, NOT IN, etc. **This is a negate operator.**

9	**OR** The OR operator is used to combine multiple conditions in an SQL statement's WHERE clause.
10	**IS NULL** The NULL operator is used to compare a value with a NULL value.
11	**UNIQUE** The UNIQUE operator searches every row of a specified table for uniqueness (no duplicates).

40. Explain the datatypes of SQL.


SQL Data Types
Numeric
bit, tinyint, smallint, int, bigint, decimal, numeric, float, real
Date/Time
Date, Time, Datetime, Timestamp, Year
Character/String
Char, Varchar, Varchar (max), Text
Unicode Character/String
NChar, NVarchar, NVarchar (max), NText
Binary
Binary, Varbinary, Varbinary (max), image
Miscellaneous
Clob, Blob, XML, JSON

CLICK FOR NOTES