

# Data Base Management System :-

1). What is database or DBMS?

DBMS stands for Data Base management system and is used to store, retrieve and update data in computer systems.

There are 2 types of DBMS:-

1) RDBMS - Relation DBMS.

Ex:- MySQL, MSSQL, PostgreSQL.

2) NOSQL :- Non Relational DBMS.

Ex:- MongoDB, cassandra, Neo4j. → in nodes on edges form.  
JSON format ↳ focuses

2) Write the SQL statement to create a table.

```
CREATE TABLE Person (
    PERSONID int NOTNULL,
    personname varchar(50) NOTNULL,
    Age int,
    PRIMARY KEY (PERSONID),
```

3) What is a Foreign Key?

A Foreign key is a key used to link two tables together. It is a field (or collection of fields) in one table that refers to the primary key in another table.

```
CREATE TABLE Orders (
```

```
OrderID int NOTNULL,
OrderNumber int NOTNULL,
CustomerID int,
PRIMARY KEY (OrderID),
FOREIGN KEY (CustomerID)
    REFERENCES Person(PERSONID)
```

Person		Orders	
ID	Name	OrderID	CustomerID
1	Rahul	1423	3
2	Kunal	1424	1
3	Anoop	1425	2

refrencing table

4) Difference b/w Primary key & unique key

Primary key

unique key

1) Only one primary key that can be defined in a table.

1) More than one unique key

Primary key

2) cannot accept NULL values.

3). Create clustered index  $\hookrightarrow$  fast

```
CREATE TABLE Person (
```

PersonID int NOTNULL,  
lastName varchar(255) NOTNULL,  
firstName varchar(255),  
Age int,  
PRIMARY KEY (PersonID)

)

unique key.

can accept one NULL value.

3) Create non-clustered index.

```
CREATE TABLE Person (
```

PersonID int NOTNULL  
citizenID int, UNL  
Name varchar(255) NOTNULL,  
Age int  
UNIQUE (citizenID)  
)

Q5) Difference b/w Drop, Delete, and Truncate?

\* Delete :- (DML)

It is used to delete one or more rows of a table based on some condition.

DELETE FROM Customers WHERE CustomerName = 'Prao';

Log is generated we can rollback so it is slow

\* Drop (Drop table / Drop database) :- (DDL)

It is used to delete the complete table or db, schema is removed.

```
DROP TABLE Logs;
```

```
DROP DATABASE Logs_DB;
```

\* Truncate :- (DDL)

It is used to clear the data inside the table. The table schema remains intact.

```
TRUNCATE TABLE Logs;
```

No log is generated so it is fast.

Q6) What are the different types of languages available in DBMS?

4 types:- DQL, DML

1) DQL - Data Definition Language

consist of commands to define the database schema.

Ex. Create, Drop, Alter, Rename

- 2) DQL - Data query language  
 consist of commands to get data from tables.  
 Ex - select
- 3) DML - Data manipulation language.  
 Commands used to edit data inside the db.  
 Ex - Insert, Update, Delete.
- 4). DCL - Data control language -  
 commands used to control rights & permission  
 to db. Ex - Grant, Revoke

Q6 Difference b/w Union and Union All.

Union:-  
 The SQL UNION clause / operator is used to combine the results of two or more SELECT statements without returning any duplicate rows.

A, B UNION A, C = A, B, C.  $\rightarrow$  rows

select city from customers

UNION  
 select city from suppliers ORDER BY city;

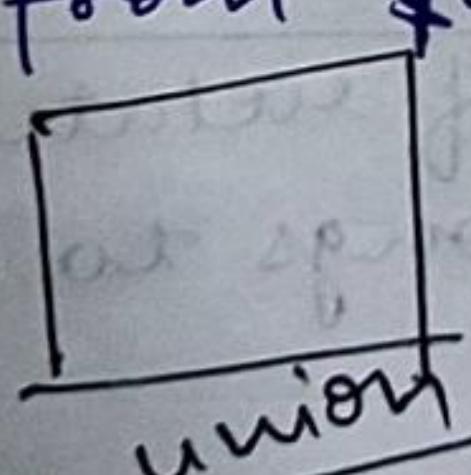
UNION ALL-

The union all operator is used to combine result of two select statements including duplicate rows.

A, B UNION ALL A, C = A, A, B, C  $\rightarrow$  rows

select city from customers UNION ALL  
 select city from suppliers ORDER BY city;

Person	
ID	Name
1	Ram
2	Shyam
3	Mohan
4	Dheeraj



1
2
3
4

same table  
 (NO Duplicate)

2
3
4
1
2
3
4

contains Duplicate Rows.

select \* from p.  
 UNION ALL  
 select \* from person

Q7). What are ACID's properties in DBMS? - ①

Atomicity :-

Entire transaction takes place at once or  
doesn't happen at all.

Ex

	User	Money	Before transaction
Transfer	A	15,000	$A = A - 5000 = 10,000$
5000	B	5000	$B = B + 5000 = 10,000$

	User	Money	After Transaction
	A	10,000	
	B	10,000	

consistency :-

The database must be consistent before & after  
the transaction.

Ex Money Before Transaction = After Transaction  
 $15,000 + 5,000 = 10,000 + 10,000$

Isolation :-

multiple transactions happen independently  
without interfering with one another. The intermediate state of a transaction is invisible to other transaction.

Ex Trying to convert parallel transaction (process)  
in CPU scheduling to series / serial process.

Durability :-

once transaction has happened, the changes stay  
there even if a system failure occurs afterwards.

Q8 What are different types of relationships

one to one :- One order belongs to only  
one customer.

one to many :-

Ex one company have many employs.

many to many :- eg:-

one person can work on many project &  
each project can have many person working  
on it.

## Q9 Different types of SQL Joins.

- SQL Joins:- (There must be common attributes)  
 ✓ statement is used to combine data on rows from two or more tables based on a common field b/w them. Different types of joins are as follows:-
- Cross Join . Product
  - Natural Join
  - Equi Join
  - self Join
  - Outer
  - Left      Right      full

✓ Need of DBMS:- 'Employee'

E-NO	E-Name	Address
1	Ram	Delhi
2	Varun	Chd
3	Rani	Chd
4	Amit	Delhi
5	Nitin	Noida

DepNo	Name	Eno
D1	HR	1
D2	IT	2
D3	MRKT	4
D4	Finance	5

Q) What is the name of employee whose name is Varun?  
 (here No need of Join)

select Address FROM Employee WHERE E-Name = 'Varun'.

Q). find E-name of 'Employee' whose working as HR.  
 (Required two tables)

Join = cross product + select statement (condn)  
 (some conan)

### 6 Natural Join

Find the Emp Names = 12 rows  
 who is working in a department. (on above table)

[Emp, Dept] means crossproduct

SELECT E-name from

Emp, Dept

where Emp.Eno = Dept.Eno

Op

Ram
Varun
Amit

Emp		Dept	
ENO	E-name	DeptNo	ENO
1	Ram	D1	1
1	Ram	D2	2
1	Ram	D3	4
2	Varun	D1	1
2	Varun	D2	2
2	Varun	D3	4
3	Rani	D1	1
3	Rani	D2	2
3	Rani	D3	4
4	Amit	D1	1
4	Amit	D2	2
4	Amit	D3	4

(6)

or common attribute name must same

SELECT E-name FROM Emp NATURAL JOIN Dept

II)

'Self Join'

T <sub>1</sub>	C <sub>1</sub>	T <sub>2</sub>	C <sub>1</sub>
S <sub>1</sub>	C <sub>1</sub>	S <sub>1</sub>	C <sub>1</sub>
S <sub>1</sub>	C <sub>1</sub>	S <sub>2</sub>	C <sub>2</sub>
S <sub>1</sub>	C <sub>1</sub>	S <sub>1</sub>	C <sub>2</sub>
S <sub>2</sub>	C <sub>2</sub>	S <sub>1</sub>	C <sub>1</sub>
S <sub>2</sub>	C <sub>2</sub>	S <sub>2</sub>	C <sub>2</sub>
S <sub>2</sub>	C <sub>2</sub>	S <sub>1</sub>	C <sub>2</sub>
S <sub>1</sub>	C <sub>2</sub>	S <sub>1</sub>	C <sub>1</sub>
S <sub>1</sub>	C <sub>2</sub>	S <sub>2</sub>	C <sub>2</sub>
S <sub>1</sub>	C <sub>2</sub>	S <sub>1</sub>	C <sub>2</sub>

T <sub>1</sub>	X	T <sub>2</sub>	Sid	C-id	Since
S <sub>1</sub>	C <sub>1</sub>	S <sub>1</sub>	C <sub>1</sub>	2016	
S <sub>2</sub>	C <sub>2</sub>	S <sub>2</sub>	C <sub>2</sub>	2017	
S <sub>1</sub>	C <sub>2</sub>	S <sub>1</sub>	C <sub>2</sub>	2018	

cross product + some condn.

Select T<sub>1</sub>.sid from study as T<sub>1</sub>, study as T<sub>2</sub>

where T<sub>1</sub>.sid = T<sub>2</sub>.sid and

T<sub>1</sub>.C-id <> T<sub>2</sub>.C-id, (Not equal).

III Equi Join :-

find the Emp name who worked in a department having location same as their address?

E-no	E-name	Address
1	Ram	Delhi
2	Vasun	Chd
3	Rani	Chd
4	Amit	Delhi

Dept No	Location	E-no
D <sub>1</sub>	Delhi	1
D <sub>2</sub>	Pune	2
D <sub>3</sub>	Patna	4

cross product

1	Ram	Delhi	D <sub>1</sub>	Delhi	1
1	Ram	Delhi	D <sub>2</sub>	Pune	2
1	Ram	Delhi	D <sub>3</sub>	Patna	4
2	Vasun	Chd	D <sub>1</sub>	Delhi	1
2	Vasun	Chd	D <sub>2</sub>	Pune	2
2	Vasun	Chd	D <sub>3</sub>	Patna	4
3	Rani	Chd	D <sub>1</sub>	Delhi	1
3	Rani	Chd	D <sub>2</sub>	Pune	2
3	Rani	Chd	D <sub>3</sub>	Patna	4
4	Amit	Delhi	D <sub>1</sub>	Delhi	1
4	Amit	Delhi	D <sub>2</sub>	Pune	2
4	Amit	Delhi	D <sub>3</sub>	Patna	4

SELECT E-name

FROM

Emp, Dept,

WHERE

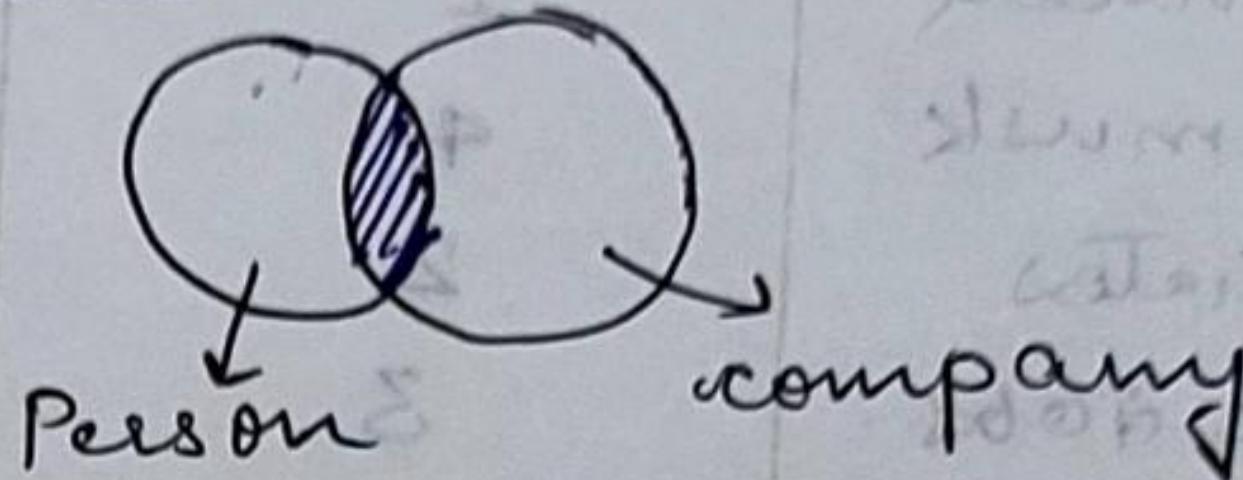
Emp.E-No = Dept.E-no

and

Emp.Address =  
Dept.Location.

#### IV Inner Join :-

return records that have matching values in both the tables. same as Natural Join but display no duplicate attributes.



Person ID	Name	Company ID	CompanyName	Founder
1	Elon Musk	1	Tesla	1
2	Bill Gates	2	Microsoft	2
3	Steve Jobs	3	Apple	3
4	Satya Nadella	4	SpaceX	1
		5	Twitter	NULL

left Join.

PersonID	Name	companyID	Company
1	Elon Musk	1	Tesla
1	Elon Musk	4	SpaceX
2	Bill Gates	2	Microsoft
3	Steve Jobs	3	Apple.

Q. List the persons who are founders of one or more companies.

```
SELECT Person.PersonID, Person.Name,
       company.CompanyID, company.CompanyName .
FROM Person INNER JOIN company
ON Person.PersonID = company.Founder.
```

#### V Left Join:-

Returns all records from left table, and the matched record from the right table.

Q. List the persons who are founders of zero or more companies along with the company name if present

```
SELECT Person.PersonID, Person.Name,
       company.CompanyID, company.CompanyName
FROM Person LEFT JOIN company
ON Person.PersonID = company.Founder.
```

left join

Person ID	Name	company ID	company
1	Elon musk	1	Tesla
2	Elon musk	4	SpaceX
2	Bill Gates	2	Microsoft
3	Steve Jobs	3	Apple
4	Satya Nadella	NULL	NULL

### VI Right Join :-

Returns all records from the right table, and the matched records from the left table.

List all the companies along with their founder if present.

SELECT Person.PersonID, Person.Name,  
 company.companyID, company.CompanyName  
 FROM Person RIGHT JOIN company  
 ON person.PersonID = company.Founder

PersonID	Name	CompanyID	Company
1	Elon Musk	1	Tesla
2	Bill Gates	2	Microsoft
3	Steve Jobs	3	Apple
4	Elon Musk	4	SpaceX
5	NULL	5	Twitter

### VII Full Join :-

return all records when there is either left or right table . This is a UNION of Left Join and Right Join.

List all the persons with their company if present and all companies with their founders if present.

SELECT Person.PersonID, Person.Name,  
 company.companyID, company.Name  
 FROM Person FULL JOIN company  
 ON person.PersonID = company.Founder

PersonID	Name	CompanyID	Company
1	Elon Musk	1	Tesla
2	Elon Musk	4	SpaceX
3	Bill Gates	2	Microsoft
4	Steve Jobs	3	Apple
5	Satya Nadella	NULL	NULL
NULL	NULL	5	Twitter

### (DDL)

SQL syntax - Create table

1) `create table Table_name (col1name datatype, col2name datatype, col3name datatype);`

`desc Table-name`

---

Ex.

```
create Table emp
(
    id int,
    name varchar(20),
    salary number(10)
);
desc.emp
```

2) Alter command (DDDL)

↳ Add column `table name`  
`alter table employee add varchar(10);`

↳ Remove column  
`alter table employee drop column address;`

↳ Modify datatype  
`alter table employee modify id varchar(20);`

↳ Rename column  
`alter table employee rename column id to rollno;`

↳ alter table employee rename to emp1;

↳ Add constraints

`alter table emp1 add primary key (roll-no);`

↳ Remove constraints

↳ Remove column / table

```
create table employee
(
    id int,
    name varchar(10)
);
```

rename table name

④ Alter (DDL) change on structure.

update syntax.

\* UPDATE table name

SET col1 = value1,  
col2 = value2 ...

WHERE condition.

② data

ID	Name	Salary
1	Ram	10000
2	Amit	20000

update data set salary = salary \* 10

⑤ update (DML)

on data.

Ex

UPDATE students

set name = 'Ram',  
city = 'Chennai' WHERE  
student ID = 10

# Constraints:- we apply condition on columns.

1) unique:-

2) Not NULL

\* 3) Primary key:- U+Not null.

4) check:- age int check (age > 18)

\* 5) Foreign key:-

6) Default:- salary int default 1000

### Queries

Ques1:- write a SQL

Query to display maximum salary from emp table.

select max(salary) from emp.

O/P 50000

Emp			
E_id	E-name	Dept	Salary
1	Ram	HR	10000
2	Amit	MRKT	20000
3	Ravi	HR	30000
4	Nilesh	MRKT	40000
5	Vasun	IT	50000

Ques2 Write a SQL query to display employee name who is taking maximum salary.

select E-name from emp

where salary = (select max(salary) from emp)

Ques3 Write a SQL query to display second highest salary from Emp table?

select <sup>max</sup>(salary) from emp 40000

where salary <> (select .Max(salary) from emp);

50000

Ques 4 Write a SQL query to display employee name who is taking second highest salary. (11)

Select E-name from Emp

Where salary = (Select Max(salary) from emp  
Where salary <> (Select Max(salary)  
from emp))

Ques 5 (Group by clause)

Write a query to display all the dept names alongwith no. of employ working in that.

Select dept, count(\*) from Emp

Dept	No. of Employees
HR	2
MRKT	2
IT	1

group by dept  
Here we can use only dept on aggregate fn in GroupBy. IT

Ques 6 Write a query to display all the dept names where no. of emps are less than 2.

Select dept from emp group by dept having count(\*) < 2;

HR	2
MRKT	2
IT	1

Print name:-

Select E-name from emp where dept IN ( )

Ques 7 Write a query to display highest salary department wise and name of emp who is taking that salary.

Select E-name from Emp

where salary IN (select max(salary) from emp group by dept);  
30000  
40000  
50000

Ques 8). find the name of emps who are working on project

Eid	Ename	City	Erld	Pid	P-name	Location
1	Raj	Chd	1	P1	IoT	Banglore
2	Vamsi	Delhi	5	P2	Big Data	Delhi
3	Nitin	Chd	3	P3	Retail	Mumbai
4	Robert	Bang	4	P4	Android	Hyderabad
5	Ammy	Chd				

Select E-name from emp where Eid IN  
(select distinct(Eid) from Project)  
(1, 5, 3, 4)

(12)

## Aggregate Function :-

1) max 3) min 3) count,

4) Avg 5) sum.

Select count(\*) from emp;  
// 6.

Select count(salary) from  
emp. // 5.  
not count null.

Select distinct(count(salary))  
from emp // 4

Select distinct(sum(salary)) // 110000  
from emp

$$\text{Avg(salary)} = \frac{\text{sum(salary)}}{\text{count(salary)}} = \frac{110000}{5} = 22000$$

E-id	E-name	Dept	Salary
1	Ram	HR	10000
2	Amit	MRKT	20000
3	Ram	HR	20000
4	Nitin	MRKT	30000
5	Vaish	IT	50000
6	Sandy	Testing	Null

Ques Find the nth highest salary using SQL.

2nd highest

Select <sup>TOP 1</sup> salary from

(Select DISTINCT TOP 2 salary  
From Employees  
Order by Employee.Salary DESC)

id salary

1 8000

2 7000

3 6000

8000  
7000

Result

Order by salary → 7000  
on 8000

1) Find the employees with top n distinct salaries.

2) Find the lowest salary among the salaries

this will give nth largest salary

3) Find the details of employee.

③ Select \* from employee where sal =

② Select min(sal) from employee

Where sal IN (

③ Select distinct top N  
sal from Employee  
order by sal desc)

# Normalization:-  
 a technique of organizing the data into multiple related tables, to minimize Data Redundancy. (13)

What is Data Redundancy? why should we reduce it?  
 → is repetition of similar data at multiple places.

→ Repetition of data increases the size of Database problems:- issue due to data redundancy.

- 1) Insertion Anomaly:- if want to add 5th student then needs to repeat data.
- 2) Deletions:- loss of Data.
- 3) updation A:- if Mr. X leave and Mr. Y came.

Ex

'Student Table'

roll no	Name	branch	nod	office-tel
1	Akon	CSE	Mrx	5337
2	Bkon	CSE	Mrx	5337
3	Ckon	CSE	Mrx	5337
4	Dkon	CSE	Mrx	5337

seen Normalization

roll no	Name	Branch
New Student Table		

Branch	nod	office
New Branch Table		

Types of Normalization:-

1) First Normal Form:- step 1 of normalization

4 Rules:-

- 1) Each column should contain values that are of the same type. (Date type can't contain string)
- 2) Each column should have atomic values. Not like C, C++ and X, Y.
- 3) Each column should have unique name.
- 4) Order in which data is stored doesn't matter.

2) Second Normal Form (2NF):-

- 1) It should be in 1st Normal Form
- 2) And, It should not have any partial dependency.

Partial Dependency:-

When any attribute in the table depends only on a part of the primary composite key and not on the complete primary key.

ER :-  $\begin{array}{ccc} A & \xrightarrow{\text{depends}} & B \\ \text{Prime,} & \text{non prime} & \end{array}$

$\begin{array}{c} AXY \\ \text{candidate} \\ \text{key} \end{array}$

then B depends on A rather than entire key is partial depend

student-id	name	reg-no	branch	subject-id	subject-name
PK	Student Table			PK	Table

Here Teacher col only depends on subject and not on student

score-id	student-id	subject-id	marks	Teacher
				composite key

In 2nf of score table

score-id	stud-id	subj-id	marks
	X8N1		320
	X8N1		320

subj-id	subj-name	Teacher

② Third Normal Form (3NF)

It should be in 2NF. and There should not be transitive dependencies

# Transitive dependencies:-  
is when some attributes of the table depends on some other attributes which is not prime attributes.

Score Table

score-id	student-id	subject-id	marks	exam-name	total-marks

prime attributes/  
composite key

depends on  
not a part  
of primary key

This is transitive dependency.  
It increases the complexity in the db.

A  $\rightarrow$  B  
non-prime non-prime  
prime prime  
This is transitive dep.

BCNF (Boyce-Codd Normal Form) 3.5 NF:-

- It should be in the 3rd normal form,
- for any dependency A  $\rightarrow$  B, A should be a super key. Which means for, A  $\rightarrow$  B if A is non-prime and B is a prime attribute

prime attribute  $\rightarrow$  non-prime attribute

## keys:-

A DBMS key is an attribute or set of attributes which help uniquely identify the records.

### ① Super key:-

superset of all key

Sid Reg-id Name Branch Email.

(A) Sid Sid + Reg-id Sid + Reg-id + Email.

(B) Reg-id Reg-id + Email

(C) Email Email + SID

An attribute or set of attribute that are used to identify the table.

### ② Candidate key:-

It is nothing but a minimal subset of super key

~~{Sid}~~ {~~{Sid, Reg-id}~~} {~~Sid + Reg-id + Email~~}

If any subset of a super key is super key then key cannot be candidate key.

### ③ Primary key:-

The candidate key chosen to uniquely identify each row of data.

### ④ Alternate key:-

other than PK.

### ⑤ Composite key:-

Any key with more than one attribute is called composite key.

Ex (SID, REG-ID) (Email, SID) etc.

### ⑥ Compound key:-

If a composite key has at least one attribute

which is a foreign key is called compound key.

Ex If we have a composite key (REG-ID, BRANCH code)

then it is anntk Bcz branch attribute is fk.

### ⑦ Surrogate key:-

If a relation has no attribute which can be used to identify the data stored in it, then we create an attribute for purpose.

It adds no meaning of data but helps to identify data uniquely.

Like command :-

(16)

1). Find employee detail whose name starting with 'A'.

Select \* from Emp, Where name LIKE 'A%'

2) Find emp detail whose name ending with 'n'.  
Where name Like '%n'.

3). Whose name consist 'ee'.  
Like '% ee %'.

4), whose name contains 'a' in a second place.  
Like '\_a%'.

5). whose name contains 'a' in second place and name should contains five character.  
Like '-a---'.

Emp	
ID	Name
1	Vaishnavi
2	Arun
3	Karuna
4	Amit
5	Ranjeet
6	Ajeet.

# Entity, Entity type, Entity set :-

student → Entity type → collection of entity having common attribute.

ID	Name	Age
e1	Ram	14
e2	Shyam	12
e3	John	13

entity  
(each record)

Entity set :- collection of one or more entity.

Attributes :- are properties that describes entity  
Domain :- set of permitted value.

ER Model :-

Diagrammatic representations of ER model which represent diff entities (real world existence) and their relationship among each other.

Strong Entity  
has its own Primary  
Key.

Father works on company  
has its own id (PK)

Weak Entity  
does not  
depend on strong entity

Family member  
depends on father