

Q. What is DBMS? Is it a system soft or will it fit in business?

Ans: DBMS stands for Database Management System.

⇒ DBMS is a System software responsible for the Creation, Retrieval and update and Management of the database.

⇒ It ensures that our data is consistent, organized and is easily accessible by serving as an interface between the database and end users.

Q. Why do we use DBMS & NOT File System.

⇒ Because, there is a several problem in file system -

- ✓ (a) Complex process to retrieve the data.
- ✓ (b) Loss of data on Concurrent Access.
- ✓ (c) Data Redundancy (Means Data Repetition).
- ✓ (d) Data Isolation is not there.

Q. What is RDBMS.

⇒ Relational Database Management System is a type of database management system that stores data in a structured format using tables, rows and columns.

⇒ It follows the principle of Relational database's where data is organized into relations (tables) and can be accessed using SQL (Structured Query Language).

### (Features of RDBMS)

1). (Data is stored in tables) ⇒ Each table has rows (records) and columns (Fields).

2). (User SQL for data Management) ⇒ SQL is used for querying, updating and managing data.

### 3). (Support ACID properties)

⇒ Ensures data integrity with -

- a). Atomicity
- b). Consistency
- c). Isolation
- d). Durability

4). (Enforces Relationship between tables)

⇒ Uses primary keys and Foreign keys

to maintain relationships.

5). (Normalization)  
⇒ Reduces data redundancy.  
⇒ Improves efficiency.

### ⇒ Examples of RDBMS

✓ a). MySQL

✓ b). PostgreSQL

✓ c). Oracle database

✓ d). Microsoft SQL server

✓ e). IBM DB2

4). What are the different database Management System other than RDBMS?

### ⇒ a). (Distributed databases)

Amazon

Redshift

TeraData

Netezza

### b). (NoSQL database)

MongoDB

Cassandra

### c). (Graph databases) ⇒ Amazon Neptune

### d). (Cloud databases)

AWS

Azure

Oracle Cloud

### 5). What is Entity Integrity.

⇒ It ensures Tables must have a column or set of columns which uniquely identify a row.

⇒ It Enforced through : Primary key

### Q. What is Referential Integrity.

⇒ So basically, Referential Integrity ensures that the value for 1-table is coming from another table.  
↓

It means, the Foreign key value should always refers to the primary key value.

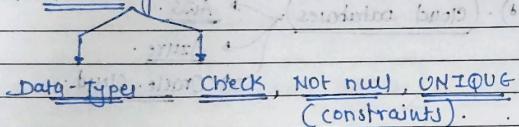
⇒ Referential Integrity enforces through: Foreign key.

### Q. What is Domain Integrity.

⇒ It ensures that the column values should be defined.

⇒ E.g. ⇒ M or F in Gender.

⇒ Enforced through:



### Q. Why we use Constraints in SQL

⇒ Because, Constraints are used to specify the rules concerning data in the table.

### Q. What are the Constraints in SQL.

⇒ a). (Not Null) ⇒ Restricts NULL value from being inserted into a column.

b) (Check) ⇒ Verifies that all values in a field satisfy a condition.

c) (Default) ⇒ Automatically assign a default value if no value has been specified for the field.

d) (Unique) ⇒ Ensures unique values to be inserted into the field.

e) (Primary key) ⇒ Uniquely identify each record in a table.

f) (Foreign key) ⇒ Used to link one or more than one table together.

⇒ A foreign key matches the primary key field of another table.

⇒ It means a foreign-key field in one table refers to the primary-key field of the other table.

### Q. What are the various levels of constraints.

⇒ There are two levels of a constraint.

Column level constraints	Table level constraints
--------------------------	-------------------------

## 11. Column level Vs Table level constraints

Features	Column-level constraints	Table-level constraints
1. Definition.	# Constraints applied to a <u>single column</u> at the time of column definition.	# Constraints applied to one or more columns separately after defining all columns.
2. Scope.	# <u>Limited to a specific column</u> .	Can apply to <u>a multiple column</u> in a table.
3. Syntax	Defined directly in the column definition.	Defined separately after all columns are declared.
4. Usage.	# Used for constraints like - ✓ a) NOT NULL ✓ b) UNIQUE ✓ c) DEFAULT etc..	# Used for constraints like - ✓ a) primary key ✓ b) Foreign key ✓ c) CHECK ✓ d) UNIQUE on <u>multiple columns</u> .

## 12. How can you select unique records from a table?

→ Using DISTINCT keyword

## 13. What is a Primary key?

→ The Primary key constraint is used to uniquely identify each record in a table.

### i. Primary key has a property

NOT NULL  
UNIQUE

→ A table in SQL is strictly restricted to have one and only one primary key, which is comprised of single or multiple column.

## 14. What is a Foreign key?

→ A Foreign key comprises of single or collection of fields in a table that essentially refer to the PRIMARY KEY in another table.

→ It ensures Referential Integrity between 2 tables.

15

### What are different types of Relationship in SQL?

- ⇒ a). (One-to-One) ⇒ This can be defined as the relationship between two tables where each record in one table is associated with the maximum of one record in the other table.

### b). (One-to-Many & Many-to-One)

- ⇒ This is the most commonly used relationship where a record in a table is associated with multiple records in the other table.

### c). (Many-to-Many)

- ⇒ This is used in cases when multiple instances on both sides are needed for defining a relationship.

### d). (Self Referencing Relationship)

- ⇒ This is used when a table needs to define a relationship with itself.

16

### Different Types of SQL commands:

- ⇒ a). DDL ⇒ Data Definition Language.  
b). DML ⇒ Data Manipulation Language.  
c). DCL ⇒ Data Control Language.  
d). TCL ⇒ Transaction Control Language.

17

### What is DDL Command?

#### ⇒ Data Definition Language.

- ⇒ Used to define the database schema. A description of how the data should reside in the database.
- a). Create
  - b). Alter
  - c). Drop
  - d). Truncate
  - e). Comment
  - f). Rename.

18

### What is DML command? ⇒ deals with data manipulation

#### ⇒ Data Manipulation language (Command).

- a). Insert
- b). Update
- c). Delete

19

### What is DCL command?

#### ⇒ Data control language.

- Commands which deals with the rights, permissions and other controls of the database system.

- a). Grant
- b). Revoke

20

### What is TCL command?

- Savepoint
- Commit
- Rollback
- Set Transaction

#### ⇒ Transaction control language.

- Commands deals with the transaction within a database.

## Q1. Difference between Commit Vs Rollback.

Features Definition:

### COMMIT

### ROLLBACK

1. Definition:

- # Saves all changes made during the current transaction permanently.
- # Undoes all changes made during the current transaction.

2. Effect:

- changes become permanent and cannot be undone.
- Restore the database to the state before the transaction started.

When to use:  
After successfully completing all operations in a transaction.

When an error occurs or you want to discard the changes.

When the Transaction is Aborted or Failed then Rollback occurs.

3. Reversibility

### Irreversible

- Once committed, changes cannot be undone.

### Reversible

- Restores the database to its previous state.

## Q2. Difference between Delete, Drop and Truncate.

### DELETE

### TRUNCATE

### DROP

1. DML

DCL

DDL

2. WHERE condition can be used.

WHERE condition can't be used.

3. Slow

Fast

Fast

## Q3. Delete Vs Truncate

### Delete

### Truncate

① State Command is used to delete specific rows.

② Truncate command is used to delete all rows from the table.

③ It is a DML command.

④ It is a DCL command.

⑤ Delete command is slower than truncate command.

⑥ Truncate command is faster than delete command.

⑦ There may be WHERE clause in a delete command in order to filter the record.

⑧ While there may not be WHERE clause in truncate command.

⑨ We can rollback the data even after using delete command.

⑩ While in this command we can't rollback.

## 25. Difference betw Union &amp; Union ALL

- ⇒ a) (UNION) ⇒ only keeps unique records
- ⇒ b) (UNION- ALL) ⇒ UNION ALL gives all values including duplicates

## 26. What is an Alias

- ⇒ Alias is a temporary name assigned to the table or table column for the purpose of a particular SQL query.

E.g. `SELECT A.empname AS 'Employee' B.empname AS 'Manager' FROM employee A, employee B WHERE A.empid = B.empid;`

## 27. What is a Join ? List its different types.

Join is used to combine records (rows) from two or more tables in a SQL database based on a related column between the two.

A) (INNER JOIN) ⇒ Retriever records that having matching values in both tables involved in the join. It gives exact matching rows.

Eg. `SELECT * FROM Table A JOIN Table B ON`

`SELECT * FROM Table A INNER JOIN Table B ON`

`Table-A.column1 = Table-B.column1`

[8]. LEFT (OUTER JOIN) JOIN :-

⇒ Retrieved all the records/rows from the left table & the matched records/rows from the right table.

[9]. (RIGHT (OUTER) JOIN)

⇒ Retrieved all the records/rows from the right table & the matched records/rows from the left table.

[10]. FULL OUTER JOIN

⇒ Full outer join fetches the common rows along with unmatching data from both the tables.

[28]. What is a Self-Join

⇒ A self join is a type of regular join where a table is joined to itself based on some relation between its own columns.

⇒ Self-join uses the INNER JOIN or LEFT JOIN clause and a table alias is used to assign different names to the table within the query.

Example: SELECT A.emp\_id AS "Emp ID", A.emp\_name AS "Employee",  
B.emp\_id AS "Sup. ID", B.emp\_name AS "Manager"  
FROM employee A, employee B  
WHERE A.emp\_sup = B.emp\_id;

[29]. What are the different Operators available in SQL

- ⇒ a). Arithmetic Operators: (+, -, \*, /, %)
- b). Logical Operators: (AND, OR, NOT)
- c). Comparison (Relational Operators):

(=, !=, <>, >, <, >=, <=)

- d). Bitwise Operators: (|, &, ^, ~, <<, >>)

[30]. CHAR Vs VARCHAR datatype in SQL

## CHAR

CHAR stands for "Character".

## VARCHAR

VARCHAR stands for "Variable Character".

- ⇒ CHAR datatype is used to store character strings of fixed length.

- ⇒ VARCHAR datatype is used to store character strings of variable length.

- ⇒ We should use the CHAR datatype when we expect the data values in a column are of the same length.

- ⇒ We should use the VARCHAR datatype when we expect the data values in a column are of variable length.

- ⇒ Better performance than VARCHAR

- ⇒ Performance not good as compared to CHAR.

- ⇒ Size = (2000 Bytes)

- ⇒ Size = (4000 Bytes).

### Q1. What is Normalization?

⇒ Way of organizing structured data in the database efficiently.

### Q2. Advantages of Normalization

- ⇒
  - a) Better database organization.
  - b) More tables with Smaller rows.
  - c) Efficient data access.
  - d) Remove anomalies.
  - e) Reduction of redundant & duplicate data.
  - f) Ensure consistent data after modification.

### Q3. What is Functional Dependency.

⇒ Also known as database dependency.

⇒ defines as the relationship which occurs when one attribute in a relation uniquely determine another attributes.

### Q4. What are the Different Normal Forms?

- ⇒
  - a) 1NF
  - b) 2NF
  - c) 3NF
  - d) BCNF (Boyce Codd NF).

### Q5. First Normal Form (1NF)

⇒ Ensures that each column contains only atomic (indivisible) values.

⇒ Removes duplicate columns and ensures each column has a unique name.

### Q6. Second Normal Form (2NF)

⇒ Must be in 1NF.

⇒ Removes partial dependency.  
(i.e., No non-key column should depend on only part of a composite primary key).

### Q7. Third Normal Form (3NF)

⇒ Must be in 2NF.

⇒ Removes transitive dependencies.  
(i.e., No non-key column should depend on another non-key column).

### Q8. Boyce-Codd Normal Form (BCNF)

⇒ A stronger version of 3NF.

⇒ Ensures that every determinant is a candidate key.

9.

### Fourth Normal Form (4NF)

- Multi-be in BCNF.
- Fourth Normal Form

### Fifth Normal Form (5NF)

- Ensures that data cannot be recombined.
- From simple table without class.

- What is Denormalization?
- Ensures that data cannot be recombined.
  - From simple table without class.

- b) DOLTF : (Quine; Translational Processing)
- Quines involved in such database are generally simple, need fast response time and return relatively few records.
  - These systems are generally designed for a large audience of end users who conduct short translation.
  - DOLTF stands for Online Translation Processing.

- c) OLTF : (Quine; Translational Processing)
- OLTF stands for Online Translation Processing: It is a class of software applications capable of supporting translation.
  - Such systems are generally applied to a class of users who conduct short translation.
  - Details with complete join dependencies.

- d) OLAP : (Quine; Analytical Processing)
- Quines involved in such database are generally simple, need fast response time and return relatively few records.
  - OLAP is a class of software programs which are characterized by relatively slow response time of outcome.
  - OLAP performs high level aggregation.

- e) DOLAP : (Quine; Analytical Processing)
- DOLAP stands for Online Analytical Processing.
  - Such systems are widely used for data mining.
  - Such systems are multimensional schema.
  - For OLAP systems, the effective effectiveness measures response time.
  - Such systems are widely used for data mining.

- What is Denormalization?
- Denormalization is the reverse of normalization.
  - Where, the normalized schema is converted into a schema which has a redundant information.
  - The reason for performing denormalization is to overcome the disadvantages of query processor.

- The overheads produced in query processor by an over-normalized structure.

- The overheads produced in query processor by an over-normalized structure.

例. `#include <iomanip>`  
    `int main()`  
    `{`     `cout << uppercase << fixed << setprecision(3);`  
        `cout << "UPPER() : " << uppercase << "A" << endl;`  
        `cout << "LOWER() : " << lowercase << "A" << endl;`  
        `cout << "FLOOR() : " << floor(1.5) << endl;`  
        `cout << "CEIL() : " << ceil(1.5) << endl;`  
    `}`

$\Rightarrow$  scalar function returns a single value based on the input value.

Q. What are scalar function?  
Eg.  $\text{Avg}()$ ,  $\text{Count}()$ ,  $\text{Min}()$ ,  $\text{Max}()$

Aggregates function performs operations on a collection of values to return a single value.

What are Aggregate Functions? [38]

function in a query.

Therefore, WHERE clause is applied to each row returned by a query.

Having clause is only used with the group B

Diffusion heat, Reynold's number, a, MHD, diffusion

Q. Different culture  $\Rightarrow$  Remote duplicate values from result set.

(LIMIT culture)  $\rightarrow$  Replaces the numbers of species per

Date:	
Page No.:	
Page No.:	
Page No.:	

(Hunting clause)  $\Rightarrow$  Filters groups after aggregation

Values in specified columns and is often used with aggregate function. (sum, avg, count etc)

(group by clause)  $\rightarrow$  groups: the rows left the same  
order: descending order:

→ Which closure → Hitler the roles based on a (polish) Order by closure → start the result set to ascending

What are common clauses used with SELECT query in SQL?

outer cavity. The long term problem depends on the rate of diffusion.

A subquery that A subquery inside

There are 2 types in question. Correlated

A subsequence is a subset of terms of a sequence. It also has the same limit as the original sequence.

What is a superstring? What are its types?

- Q1. (What is a View)  $\Rightarrow$  A database object that no value is called "View".
- $\Rightarrow$  A View is a Virtual table based on the result-set of an SQL statement.
- $\Rightarrow$  Views contain rows and columns just like a real table.
- $\Rightarrow$  A View is created by combining one or more tables.
- $\Rightarrow$  A View is a query that defines a plain for evaluating query which has the result-set formatted (cut it into pieces).
- $\Rightarrow$  The output is provided faster.
- $\Rightarrow$  b). A larger number of queries can be executed in less time.
- $\Rightarrow$  c). Reduces more storage complexity.
- $\Rightarrow$  d). Reduces the number of queries can be executed in less time.
- $\Rightarrow$  e). Reduces the number of queries can be executed in less time.
- $\Rightarrow$  f). Reduces the number of queries can be executed in less time.
- $\Rightarrow$  g). Reduces the number of queries can be executed in less time.
- $\Rightarrow$  h). Reduces the number of queries can be executed in less time.
- $\Rightarrow$  i). Reduces the number of queries can be executed in less time.
- $\Rightarrow$  j). Reduces the number of queries can be executed in less time.
- Q2. What are Views used for?
- $\Rightarrow$  FROM table\_name
- $\Rightarrow$  SELECT columns
- $\Rightarrow$  CREATE VIEW view\_name AS
- $\Rightarrow$  a). Trigger is a procedural code in a database.
- $\Rightarrow$  b). Triggers are automatically triggered when specific events occurs on a particular table.
- $\Rightarrow$  c). Triggers are automatically triggered when specific column updating triggers are invoked upon them.
- $\Rightarrow$  d). Triggers are available in MySQL.
- Q3. (What do you understand by query optimization)
- $\Rightarrow$  The Advertisers are as follows:-
- $\Rightarrow$  a). Query optimizer is provided faster.
- $\Rightarrow$  b). A larger number of queries can be executed in less time.
- $\Rightarrow$  c). Reduces more storage complexity.
- $\Rightarrow$  d). Reduces the number of queries can be executed in less time.
- $\Rightarrow$  e). Reduces the number of queries can be executed in less time.
- $\Rightarrow$  f). Reduces the number of queries can be executed in less time.
- $\Rightarrow$  g). Reduces the number of queries can be executed in less time.
- $\Rightarrow$  h). Reduces the number of queries can be executed in less time.
- $\Rightarrow$  i). Reduces the number of queries can be executed in less time.
- $\Rightarrow$  j). Reduces the number of queries can be executed in less time.
- Q4. (What are Views used for?)
- $\Rightarrow$  a). Resticting access to data.
- $\Rightarrow$  b). Making complex queries to simple forms.
- $\Rightarrow$  c). Enabling data independence.
- $\Rightarrow$  d). Providing different views of same data.
- $\Rightarrow$  e). Inserting complex queries to simple forms.
- $\Rightarrow$  f). Deleting data from table.
- $\Rightarrow$  g). Inserting data into table.
- $\Rightarrow$  h). Updating data in table.
- $\Rightarrow$  i). Deleting data from table.
- $\Rightarrow$  j). After DELETE.
- $\Rightarrow$  k). BEFORE INSERT.
- $\Rightarrow$  l). AFTER INSERT.
- $\Rightarrow$  m). BEFORE UPDATE.
- $\Rightarrow$  n). AFTER UPDATE.
- $\Rightarrow$  o). BEFORE DELETE.
- $\Rightarrow$  p). AFTER DELETE.
- Q5. Triggers are available in MySQL Table.

45) What is a stored procedure?

→ A stored procedure is a set of SQL statements which can be combined together to form a program. It can be reused and shared by multiple programs.

→ A stored procedure can be used at a module level or system level. It can be created, modified, deleted, etc. It is re-usable and can be referred to from anywhere.

46) Advantages of stored procedure

1) Reduces network traffic and provides better security to the data.

2) Helps to create shared procedure if more than one application needs to execute it.

ON table\_name FOR EACH ROW

BEFORE

END;

- trigger code

- variable declaration

AFTER

{ insert | update | delete }

CREATE TRIGGER trigger\_name  
[ before | after ]

(SQL)

47) How to create a trigger in MySQL

48) How to create shared procedure in MySQL to execute it.



56. [What are the Indexes in SQL] [Ans]
- They allow quick access to rows based on the primary key.
57. [Types of Indexes]:
- Primary Key Index  $\Rightarrow$  Automatically created when a primary key is defined.
  - Unique Index  $\Rightarrow$  Ensures all values in a column are unique.
  - Composite Index  $\Rightarrow$  Created on multiple columns for faster performance.
  - Full Text Index  $\Rightarrow$  Used for text searches.
  - Spatial Index  $\Rightarrow$  Used for geographic data types.

58. [Index vs Full Text Procedure] [Ans]
- Using select statement.
59. [You cannot call procedure if you can call a function] [Ans]
- Trigger is a stored-procedure that runs automatically on insert, update, or delete.
60. [Indexes in SQL database objects that improve the speed of data retrieval, comparisons that improve the speed of one or more columns, enhancing query performance.]:
- Indexes in SQL are database objects that improve the speed of data retrieval, comparisons that improve the speed of one or more columns, enhancing query performance.
61. [Tiger is a stored-procedure that runs automatically on insert, update, or delete. It performs some specific tasks.]:
- Trigger can execute stored procedures, can take input as a parameter.
62. [Tiger can execute procedure based on invoked explicitly by the user.]:
- Tiger cannot take return value.
63. [Tiger cannot handle stored procedure can take input as a parameter.]:
- We can't use transaction and roll back inside a stored procedure. Till the transaction begins the transaction will be inside a statement like begin transaction.
64. [Tiger cannot handle a value.]:
- A value is returned by a stored procedure can be used in a trigger.
65. [We can't use transaction and roll back inside a stored procedure.]:
- We can't use transaction and roll back inside a stored procedure.

CN employees (name, department):

DROP INDEX idx\_name ON employee;

(Dropping out index) .6

d) Creating a composite index

158. (Hello to declare an Index in MySQL)

CREATE Table employee (

सांख्यिकीय विवरण (Statistical Methods)

department varchar (50),

.. (Regular index) departmental column

CREATE UNIQUE INDEX idx\_unique\_email

On employee (email):

### 6. Creating a unique index

## MySQL

MySQL is currently the most popular database management system software used for managing the database. It is open-source database software which is supported by Oracle Company. It is fast, scalable and easy to use database management system.

For manipulating data, these queries are insert records, update records, deleterecords, select records, create tables, drop tables, etc.

### Database

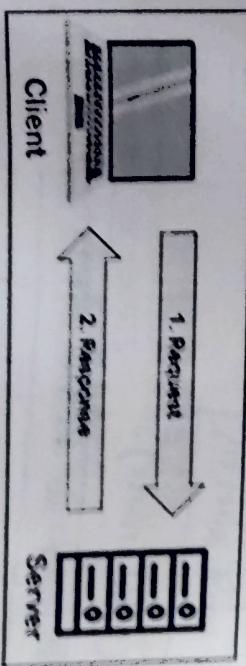
- A database is an application that stores the organized collection of records. It can be accessed and manage by the user very easily.
- It allows us to organize data into tables, rows, columns, and indexes to find the relevant information very quickly.
- Each database contains distinct API for performing database operations such as creating, managing, accessing, and searching the data it stores.
- There are many databases available like MySQL, SQLite, Oracle, MongoDB, PostgreSQL, etc.

### How MySQL Works?

MySQL follows the working of Client-Server Architecture.

This model is designed for the end-users called clients to access the resources from a central computer known as a server using network services.

Here, the clients make requests through a graphical user interface (GUI), and the server will give the desired output as soon as the instructions are matched.



### • How to create table

```
create table table name (student_id, student_name);
```

↑  
column name.

int not null auto-increment primary key.

query for create table

```
create table (Table Name) (column_name);
```

Insert record into table

```
insert into (Table name) (column_name) values (' ', ' ', ' ');
```

update record

```
update (Table Name) set (column_name) = ' ' where id = 1;
```

delete record

```
delete from (Table name) where student_id = 2;
```

whole record यांचे अनेकांक

```
select * from (table name) where student_id = 1;
```

whole record यांचे अनेकांक

```
select * from (table name);
```

table मध्ये किती count आणे लागेचा असेहा रूप

```
select count(*) from (table name);
```

# How to find 3rd Highest salary

⇒ select salary from student\_table ORDER BY salary)

## Data Type in MySQL-

A Data Type specifies a particular type of data, like integer, floating points, Boolean, etc. It also identifies the possible values for that type, the operations that can be performed on that type, and the way the values of that type are stored. In MySQL, each database table has many columns and contains specific data types for each column.

## Clauses-

Q1 Where-clause is used with SELECT, INSERT, UPDATE and DELETE clause to filter results. It specifies a specific position where you have to do the operation.

## Syntax- WHERE conditions;

Example- SELECT \* FROM student WHERE city = 'pune'; SELECT \* FROM student WHERE city = 'pune'  
AND id < 5;

Import

select \* from student where salary>=25000 AND salary<=48000;

Q2 DISTINCT- clause is used to remove duplicate records from the table and fetch only the unique records. The DISTINCT clause is only used with the SELECT statement.

Syntax- SELECT DISTINCT expressions FROM tables [ WHERE conditions]; Example- SELECT DISTINCT city FROM student;

Q3 FROM- clause is used to select some records from a table. It can also be used to retrieve records from multiple tables using JOIN condition.  
Q4 ORDER BY- Clause is used to sort the records in ascending or descending order.

## Syntax-

✓ SELECT expressions FROM tables [ WHERE conditions] ORDER BY expression [ ASC | DESC ];

Example- SELECT \* FROM student WHERE city = 'pune' ORDER BY salary;

DESC LIMIT 2,1;

## Trigger

- ① Trigger is a stored procedure that runs automatically when various events happens. (e.g. update, insert, delete).

## Stored procedure

- ① Stored procedure are a pieces of the code in written in SQL to do some specific task.

- ② Trigger can execute automatically based on the events.
- ③ It cannot take input as parameter
- ④ Trigger cannot return values.
- ⑤ We can't we transaction inside a trigger because begin transaction statement and rollback inside a stored procedure.

select count(\*) from (table name) where student\_city = 'mumbai';

Table Name .  
Column name .

## MySQL Key

### 1. Primary Key-

MySQL primary key is a single or combination of the field, which is used to identify each record in a table uniquely.

- ✓ It cannot be null or empty.
- ✓ It can contain only one primary key.
- ✓ It always contains unique value into a column.

### column Rules:

- ✓ 1. The primary key column value must be unique.
- ✓ 2. Each table can contain only one primary key.
- ✓ 3. The primary key column cannot be null or empty.

### 2. Foreign Key-

The foreign key is used to link one or more than one table together. It is also known as the referencing key.

A foreign key matches the primary key field of another table. It means a foreignkey field in one table refers to the primary key field of the other table.

### Aggregate function-

- ✓ 1. MySQL count() function is used to returns the count of an expression. It allows us to count all rows or only some rows of the table that matches specified condition.

```
mysql> SELECT COUNT(emp_name) FROM employees;
```

- ✓ 2. The MIN() & MAX() function in MySQL is used to return the minimum value and maximum in a set of values from the table.

```
mysql> SELECT MIN(income) AS Minimum_Income FROM employees;
```

### ③ Unique key

→ Unique key is a column or set of columns that uniquely identify each record in a table.

→ It means column cannot store duplicate value.

### ④ Composite key

→ It is a combination of two or more columns in a table that can be used to uniquely identify each row in a table when columns are combined.

→ Column used in the composite key can have different data type.

→ Composite key can be added in two ways -

- ✓ Using CREATE statement.
- ✓ Using ALTER statement.

## SQL Commands-

### DDL Statements

DDL is short name of Data Definition Language, which deals with database schemas and descriptions, of how the data should reside in the database.

- CREATE - used to create a database table
- ALTER - alters the structure of the existing database
- DROP - delete objects from the database
- TRUNCATE - remove all records from a table, including all spaces allocated for the records are removed
- COMMENT - add comments to the data dictionary
- RENAME - rename an object.

### DML Statements

DML is short name of Data Manipulation Language which deals with data manipulation and includes most common SQL statements such SELECT, INSERT, UPDATE, DELETE, etc., and it is used to store, modify, retrieve, delete and update data in a database.

- SELECT - retrieve data from a database
- INSERT - insert data into a table
- UPDATE - updates existing data within a table
- DELETE - Delete all records from a database table
- MERGE - UPSERT operation (insert or update)
- CALL - call a PL/SQL or Java subprogram
- EXPLAIN PLAN - interpretation of the data access path
- LOCK TABLE - concurrency Control

### DCL Statements

DCL is short name of Data Control Language which includes commands such as GRANT and mostly concerned with rights, permissions and other controls of the database system.

- GRANT - allow users access privileges to the database
- REVOKE - withdraw users access privileges given by using the GRANT

## WHERE clause

## HAVING clause

① This clause is implemented in row operations.

② This clause is implemented in column operation.

③ It does not allow to work with aggregate function.

③ This clause can be used with SELECT, UPDATE & DELETE statement.

③ Can only be used with SELECT statement.

## Stored procedure

## Function

① Purpose to use procedures is to modify the database where return values is not required. Like delete, update, drop.

① Functions are not used for any kind of updating because functions are not allowed to change anything in the database is only used for plain queries.

② Stored procedure neither contain any parameter nor return any value.

② Function should contain at least one parameter & should return a value.

③ We can call functions from stored procedures.

③ We can't call stored procedures from functions.

command

### TCL Statements-

TCL is short name of **Transaction Control Language** which deals with a transaction within a database.

- **COMMIT** - commits a Transaction
- **ROLLBACK** - rollback a transaction in case of any error occurs
- **SAVEPOINT** - to rollback the transaction making points within groups
- **SET TRANSACTION** - specify characteristics of the transaction.

### MySQL-

#### Table create

Syntax- CREATE TABLE table\_name (column\_name column\_type);

Example-

```
CREATE TABLE User (
    id int not null
    auto_increment,
    LastName
    varchar(255),
    FirstName
    varchar(255),
    Address
    varchar(255),
    City
    varchar(255),
    Salary
    varchar(255),
    primary key(id)
);
Table
insert-
```

#### Syntax-

```
insert into table_name ( column1, column2,...columnN )VALUES(value1,
value2,...valueN );
```

## Commit

- ① commit statement ~~permanently~~ save the state, when all the statements are executed successfully ~~without~~ any error.

- ② When the transaction is successful, COMMIT is applied.

## Rollback

- ① In Rollback statement, if any operation fail during the completion of transaction, it cannot permanent save the change & we can undo them using this statement.

- ② When the transaction is about failure then ROLLBACK occurs.

## Primary Key

## Unique Key

- ① We cannot store null values in primary key column.

- ① We can store null values in unique key column, but only one null is allowed.

- ② We cannot modify primary key column value.

- ② We can modify the unique key column value.

### **Example-**

For single insert records-

insert into User (lastName,firstName,Address,City,Salary) values ('naik', 'ashok','pimpri','pune', 45000),('patil','ram','kothrud','pune', 35000);

'jay','chinchwad','pune', 25000);

For multiple insert records-

insert into User (lastName,firstName,Address,City,Salary) values ('naik', 'ashok','pimpri','pune', 45000),('patil','ram','kothrud','pune', 35000);

Table

update-

### **Syntax**

UPDATE table\_name SET column\_name1 = new-value1,

column\_name2=new-value2, ... [WHERE Clause]

### **Example**

update user set firstName = 'rohan' where lastName='kulkarni' //Single record or field or

update user set firstName = 'rohan' where id=3

update employee set username = 'rohan', password='kulkarni' where id=1;

//multiplefield in same record.

update user set city = 'mumbai' where City='pune'; //updating multiple record formultiple user.

Table

delete

### **Syntax-**

DELETE FROM table\_name WHERE condition; //specific

data deleteSET SQL\_SAFE\_UPDATES = 0;

delete from user; //it will delete the all the table data not structure

Example-

delete from user where id=3;

Add

⇒ ALTER TABLE table-name ADD column-name datatype;

Example ⇒

ALTER TABLE user ADD email varchar(150);

) modify

⇒ ALTER TABLE table-name MODIFY column-name datatype;  
datatype;

Example ⇒

ALTER TABLE user MODIFY email varchar(150);

) drop

ALTER TABLE table-name DROP column-name column-name;

Example ⇒

ALTER TABLE user DROP email;

### Table select

Syntax-

SELECT field\_name1, field\_name2, ..., field\_nameN FROM  
table\_name [WHERE condition]

Example-

select \* from user;

select \* from user where

city='pune' Table Alter-(Add,

Delete, Modify | Syntax-Add

ALTER TABLE table\_name ADD column\_name  
datatype; Example- .

ALTER TABLE user

ADD Email varchar(255);

### Syntax-Modify

ALTER TABLE table\_name MODIFY COLUMN column\_name datatype;  
Example-

ALTER TABLE user

modify Email varchar(125);

### Syntax-Drop

ALTER TABLE table\_name DROP COLUMN  
column\_name; Example-

ALTER TABLE user

drop Email;

## Delete

- ① Delete command is used to delete specified rows.
- ② It is a DML command (Data Manipulation Language).
- ③ Delete command is slower than truncate command.
- ④ There may be WHERE clause in delete command in order to filter the record.

- ⑤ Truncate command is used to delete all rows from table.
- ⑥ It is a DDL command (Data Definition Language).
- ⑦ Truncate command is faster than delete command.
- ⑧ While there may not be WHERE clause in truncate command.
- ⑨ While in this command, we can't rollback.

## Truncate

## Difference between delete and truncate?

S.NO	Delete	Truncate
1.	The DELETE command is used to delete specified rows(one or more).	While this command is used to delete all the rows from a table.
2.	It is a DML(Data Manipulation Language)	While it is a DDL(Data Definition Language) command.
3.	There may be WHERE clause in DELETE command in order to filter the records.	While there may not be WHERE clause in TRUNCATE command.
4.	In the DELETE command, a tuple is locked before removing it.	While in this command, data page is locked before removing the table data.
5.	We can rollback the data even after using DELETE command.	While in this command, we can rollback.
6.	DELETE command is slower than TRUNCATE command.	While TRUNCATE command is faster than DELETE command.

## How to Write Inner Join query:

```
Create Table Customer (customer_id int not null auto-increment  
primary key , customer_name varchar (50));
```

```
Create Table Account (customer_id int, account_id int  
not null auto-increment primary key , account_balance  
varchar (150), foreign key (customer_id) references  
customer (customer_id));
```

### Inner Join

```
Select customer. customer_id, customer_name, accounts_id,  
account_balance  
FROM customer Inner JOIN Account  
ON customer. customer_id = Account. customer_id;
```

### Left Join

### Right Join

### Left Join

### Right Join

## MySQL-Join Query

Why?

If I want to retrieve the data from multiple table then go for join query.

By using one select statement, we can retrieve the data from multiple table.

It is used to retrieve data from multiple tables. It is performed whenever you need to fetch records from two or more tables.

There are three types of MySQL joins:

- ✓ MySQL INNER JOIN (or sometimes called simple join)
- ✓ MySQL LEFT OUTER JOIN (or sometimes called LEFT JOIN)
- ✓ MySQL RIGHT OUTER JOIN (or sometimes called RIGHT JOIN)

### Inner Join-

It gives you exactly matching rows called inner join.

Syntax-

```
SELECT columns FROM table1 INNER JOIN table2 ON table1.column =  
table2.column;
```

Example-

Table 1-

```
create table customers(
```

customerid int(10) primary key auto\_increment not null, customername

```
varchar(32),email varchar(32), phone varchar(125));
```

customerid	customername	email	phone
1	Ajay	a@gmail.com	0000000000
2	Abhishek	ab@gmail.com	8888888888
3	Ram	ram@gmail.com	7777777777
4	Rohit	rohit@gmail.com	6666666666
5	Archana	an@gmail.com	7777000000
6	Shivani	shivani@gmail.com	8888888888

Write the Inner  
join query.

create table accounts( customerid int,  
accountsid int primary key auto\_increment not null,  
accountstype varchar(10), balance int(50), foreign key(customerid)  
references customers(customerid));

customerid	accountsid	accountstype	balance
1	1	saving	5000
2	2	current	7500

Inner Join;

Select customers.customerid, customername, email, accountsid, balance from  
customers INNER JOIN accounts on customers.customerid = accounts.customerid;

customerid	customername	email	accountsid	balance
1	jeevan	jk@gmail.com	1	5000
2	ashok	ashok@gmail.com	2	7500

2. Left outer join - The LEFT OUTER JOIN returns all rows from the left hand  
table (Table 1) specified in the ON condition and only those rows from the other  
table where the join condition is fulfilled.

Syntax - SELECT columns FROM table1 LEFT [OUTER] JOIN table2

ON table1.column = table2.column;

Example - customers.

select customername, email, accountsid, balance from customers left join  
accounts on customers.customerid = accounts.customerid;

customerid	customername	email	accountsid	balance
1	Ajay	aj@gmail.com	1	5000
2	Ashok	ak@gmail.com	2	7500
3	soham	sm@gmail.com	3	1000
4	Ram	r@gmail.com	4	NULL
	Rohan	rohan@gmail.com	5	NULL

3. Right outer join - The MySQL Right Outer Join returns all rows from the RIGHT-handtable (Table 2) specified in the ON condition and only those rows from the other table where the join condition is fulfilled.

Syntax- SELECT columns FROM table1 RIGHT [OUTER] JOIN table2ON table1.column = table2.column;

Example-

select customername, email, accountsid, balance from customers  
accounts customerid = accounts.customerid;

customername	email	accountsid	balance
Ajay	aj@gmail.com	1	5000
Ashok	ak@gmail.com	2	7500
soham	sm@gmail.com	3	1000
Ram	r@gmail.com	4	NULL
Rohan	rohan@gmail.com	5	NULL

4. Full outer Join-  $\Rightarrow$  also known as Union join.

In MySQL it is the combination of left join union right join.

select \* from customers [left join] accounts on customers.customerid = accounts.customerid [union] select \* from customers [right join] accounts on customers.customerid = accounts.customerid;

customerid	customername	email	phone	customerid	accountsid	accountstype	balance
1	Ajay	aj@gmail.com	88889977666	1	1	saving	5000
2	Ashok	ak@gmail.com	89899888888	2	2	current	7500
3	soham	sm@gmail.com	77767777777	3	3	saving	1000
4	Ram	r@gmail.com	88888888888	4	4	NULL	NULL
	Rohan	rohan@gmail.com	88888888888	5	5	NULL	NULL
				6	6	NULL	NULL
				7	7	saving	2000

# Important Questions

## Highest Salary Calculate Query

Finding Nth highest salary in a table is the most common question asked in interviews.

salary
90000
75000
45000

First solution- with using sub-query 1<sup>st</sup> Highest salary

```
SELECT salary FROM (SELECT salary FROM student ORDER BY salary DESC LIMIT 2) ASs ORDER BY salary LIMIT 1;
```

## 2<sup>nd</sup> Highest salary

```
SELECT salary FROM (SELECT salary FROM student ORDER BY salary DESC LIMIT 2) ASs ORDER BY salary LIMIT 1;
```

## Description-

In this solution, we have first sorted all salaries from Employee table in descending order, so that 2 highest salaries come at top of the result set.

After that we took just two records by using LIMIT 2.

Again we did the same thing but this time we sort the result set on ascending order, so that second highest salary comes at top.

Now we print that salary by using LIMIT.

## 3<sup>rd</sup> Highest salary

```
SELECT salary FROM (SELECT salary FROM student ORDER BY salary DESC LIMIT 3) ASs ORDER BY salary LIMIT 1;
```

## Second solution- without using sub-

query 2<sup>nd</sup> Highest Salary calculates.

```
SELECT salary FROM student ORDER BY salary DESC LIMIT 1,1;
```

Column name → Table name

LIMP

index from 0  
always starts

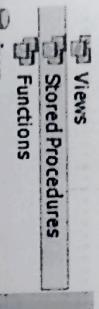
Here (1, 1) -> second 1 is used for to fetch only one row or record.

First 1 is used to calculate the 2<sup>nd</sup> highest salary.

\*\*\*\*\*

Stored Procedures>>demo

- ① Step one >> right click on Stored Procedures then new tab open which having below syntax



- ② Step 2 >> you can add your procedure name in single quote

```
CREATE PROCEDURE 'new_procedure' ()  
BEGIN  
SELECT * FROM student;  
END
```

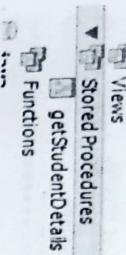
name  
procedure

- ③ Step 3 >> add the query that you want to store in procedure.

```
CREATE PROCEDURE 'getStudentDetails' ()  
BEGIN  
SELECT * FROM student;  
END
```

1 2 3 4 5

(4) Step 4> click on apply and finish after that check the stored procedure tab



(5) Step 5>> then you can call stored procedure by hitting the below query

Call (your stored procedure name)

e.g. call getStudentDetails;

For dynamic data for extra knowledge only

re: getAccountDetails

D:

CREATE DEFINER='root'@'localhost' PROCEDURE `getAccountDetails` (In id INT)

```
1 • CREATE DEFINER='root'@'localhost' PROCEDURE `getAccountDetails` (In id INT)
2 BEGIN
3   select * from customers where customer_id=id;
4 END
```

means input

call getAccountDetails(2);