**Here are some important notes about SQL:**

**What is SQL?**

- SQL (Structured Query Language) is a programming language used for managing relational databases.

- It's used to store, manipulate, and retrieve data stored in a database.

* MySQL is a relational database management system
* MySQL is open-source
* MySQL is free
* MySQL is ideal for both small and large applications
* MySQL is very fast, reliable, scalable, and easy to use
* MySQL is cross-platform
* MySQL is compliant with the ANSI SQL standard
* MySQL was first released in 1995
* MySQL is developed, distributed, and supported by Oracle Corporation
* MySQL is named after co-founder Monty Widenius's daughter: My

**Basic SQL Concept?**

- Database: A collection of related data.

- Table: A collection of data organized into rows and columns.

- Column: A single field or attribute of a table (e.g., "name", "age", etc.).

- Row: A single record or entry in a table (e.g., a single customer's data).

- Primary Key: A unique identifier for each row in a table.

**SQL Commands**

- SELECT: Retrieves data from a database table.

- INSERT: Adds new data to a database table.

- UPDATE: Modifies existing data in a database table.

- DELETE: Deletes data from a database table.

- CREATE: Creates a new database table or other object.

- DROP: Deletes a database table or other object.

**SQL Syntax**

- SQL statements typically start with a command (e.g., SELECT, INSERT, etc.).

- SQL uses various clauses (e.g., WHERE, JOIN, ORDER BY) to specify conditions and options.

- SQL uses special characters (e.g., ;, ', ", =, <, >) to define syntax and operators.

**SQL Data Types**

- Integer: Whole numbers (e.g., 1, 2, 3).

- String: Text data (e.g., "hello", "goodbye").

- Date: Dates and times (e.g., "2024-06-28", "14:30:00").

- Boolean: True or false values.

This is just a brief overview, but I hope it helps! Let me know if you have any specific questions or need further clarification on any of these points.

**COMMIT**

The COMMIT command is used to permanently save all changes made during the current transaction to the database. Once a COMMIT is issued, all changes become permanent and cannot be undone by a ROLLBACK.

**ROLLBACK**

The ROLLBACK command is used to undo all changes made during the current transaction, reverting the database to its previous state before the transaction began. This is useful if an error occurs or if you decide not to apply the changes.

**LIKE Operator**

The LIKE operator is used in a WHERE clause to search for a specified pattern in a column.

**Wildcard Characters**

A wildcard character is used to substitute one or more characters in a string.

Wildcard characters are used with the [LIKE](https://www.w3schools.com/mysql/mysql_like.asp) operator. The LIKE operator is used in a WHERE clause to search for a specified pattern in a column.

**Difference between Like and Wildcards Operators?**

1. **Purpose**:

* LIKE is an operator used for pattern matching.
* Wildcards are characters used within the LIKE operator to define the pattern.

1. **Usage**:

* LIKE is used in SQL queries to specify that a pattern match is required.
* Wildcards are used to create flexible patterns within the LIKE operator.

1. **Flexibility**:

* The LIKE operator itself doesn’t specify the pattern. It requires wildcards to define what pattern to look for.
* Wildcards allow you to create complex patterns that can match a wide range of values.

## IN Operator

## The IN operator allows you to specify multiple values in a WHERE clause.

## The IN operator is a shorthand for multiple OR conditions.

## Join

## A JOIN clause is used to combine rows from two or more tables, based on a related column between them.

## Types of Joins in MySQL

## INNER JOIN: Returns records that have matching values in both tables

## LEFT JOIN: Returns all records from the left table, and the matched records from the right table

## RIGHT JOIN: Returns all records from the right table, and the matched records from the left table

## CROSS JOIN: Returns all records from both tables

## MySQL INNER JOIN  MySQL LEFT JOIN  MySQL RIGHT JOIN  MySQL CROSS JOIN

## Union Operator

## The UNION operator is used to combine the result-set of two or more SELECT statements.

## Every SELECT statement within UNION must have the same number of columns

## The columns must also have similar data types

## The columns in every SELECT statement must also be in the same order

## HAVING clause

## The HAVING clause was added to SQL because the WHERE keyword cannot be used with aggregate functions.

## What is the usage of COALESCE

## The COALESCE function in SQL is used to return the first non-null value in a list of arguments. It is particularly useful for handling null values in data, providing a way to substitute them with a default value or to ensure that null values do not disrupt calculations or queries.

## Key Differences

## Hierarchy:

## Database: Higher level, can contain multiple schemas.

## Schema: Lower level, a part of a database.

## Content:

## Database: Contains all data and objects, including schemas.

## Schema: Contains only the structure and definitions of related objects within a database.

## Purpose:

## Database: Manages overall data storage, retrieval, and integrity.

## Schema: Organizes objects logically, aids in permissions management, and helps to avoid name conflicts.

1.  **Database**: A container that holds all data and objects (schemas, tables, etc.).

##  ****Schema****: A logical grouping of related objects within a database to organize and manage them effectively.

## Primary Key

## Definition: A primary key is a column or a combination of columns that uniquely identifies each row in a table. Each table can have only one primary key.

## Characteristics:

## It must contain unique values.

## It cannot contain NULL values.

## It is used to enforce entity integrity.

## Usage:

## Typically used for the main identifier of the table (e.g., id column).

## Unique Key

## Definition: A unique key is a column or a combination of columns that must contain unique values across the table. A table can have multiple unique keys.

## Characteristics:

## It must contain unique values.

## It can contain NULL values (except when the column is part of the primary key).

## It is used to enforce unique constraints on columns that are not the primary identifier.

## Usage:

## Used to enforce uniqueness on columns that are not the primary key but still need to be unique (e.g., email addresses).

### **ON DELETE**

The **ON DELETE** clause specifies what happens when a referenced row in the parent table is deleted.

### **ON UPDATE**

The **ON UPDATE** clause specifies what happens when a referenced row in the parent table is updated.

### **CASCADE**

The **CASCADE** option means that the changes in the parent table will be propagated to the child table.

## ON DELETE CASCADE: When a row in the parent table is deleted, all related rows in the child table will also be deleted automatically.

## ON UPDATE CASCADE: When a row in the parent table is updated, all related rows in the child table will have their foreign key values updated automatically to match the new value.

**Foreign key**

A FOREIGN KEY is a field (or collection of fields) in one table, that refers to the [PRIMARY KEY](https://www.w3schools.com/mysql/mysql_primarykey.asp) in another table. Foreign keys are used to establish a relationship between two tables and enforce referential integrity.

**1. What are the String Data Types in MySQL?**

| **Type Name** | **Meaning** |
| --- | --- |
| CHAR | fixed-length nonbinary(character) string |
| VARCHAR | variable-length nonbinary string |
| BINARY | fixed-length binary string |
| VARBINARY | variable-length binary string |
| TINYBLOB | Very small BLOB(binary large object) |
| BLOB | Small BLOB |
| MEDIUMBLOB | Medium-sized BLOB |
| LONGBLOB | Large BLOB |
| TINYTEXT | A very small nonbinary string |
| TEXT | Small nonbinary string |
| MEDIUMTEXT | Medium-sized nonbinary string |
| LONGTEXT | Large nonbinary string |
| ENUM | An enumeration; each column value is assigned, one enumeration member |
| SET | A set; each column value is assigned zero or more set members |
| NULL | NULL in SQL is the term used to represent a missing value. A NULL value in a table is a value in a field that appears to be blank. This value is different than a zero value or a field that contains spaces. |

**2. How to add users in MySQL?**

You can add a User by using the CREATE command and specifying the necessary credentials. For example:

CREATE USER ‘testuser’ IDENTIFIED BY ‘sample password’;

**3. What is BLOB in MySQL?**

BLOB is an acronym that stands for a binary large object. It is used to hold a variable amount of data.  
There are four types of BLOB:

* TINYBLOB
* BLOB
* MEDIUMBLOB
* LONGBLOB

A BLOB can hold a very large amount of data. For example - documents, images, and even videos. You could store your complete novel as a file in a BLOB if needed.

**4. What are the Temporal Data Types in MySQL?**

| **Type Name** | **Meaning** |
| --- | --- |
| DATE | A date value, in ' CCYY-MM-DD ' Format |
| TIME | A Time value, in ' hh : mm :ss ' format |
| DATETIME | Date and time value, in ' CCYY-MM-DD hh : mm :ss ' format |
| TIMESTAMP | A timestamp value, in ' CCYY-MM-DD hh : mm :ss ' format |
| YEAR | A year value, in CCYY or YY format |

Example: To select the records with an Order Date of "2018-11-11" from a table:

SELECT \* FROM Orders WHERE OrderDate='2018-11-11'

**5. What is MySQL?**

MySQL is a database management system for web servers. It can grow with the website as it is highly scalable. Most of the websites today are powered by MySQL.

**6. What are the Numeric Data Types in MySQL?**

MySQL has numeric data types for integer, fixed-point, floating-point, and bit values, as shown in the table below. Numeric types can be signed or unsigned, except BIT. A special attribute enables the automatic generation of sequential integer or floating-point column values, which is useful for applications that require a series of unique identification numbers.

| **Type Name** | **Meaning** |
| --- | --- |
| TINYINT | Very Small Integer |
| SMALLINT | Small Integer |
| MEDIUMINT | Medium-sized Integer |
| INT | Standard Integer |
| BIGINT | Large Integer |
| DECIMAL | Fixed-point number |
| FLOAT | Single-precision floating-point number |
| DOUBLE | Double-precision floating-point number |
| BIT | Bit-field |

**7. How do you view a database in MySQL?**

One can view all the databases on the MySQL server host using the following command:

mysql> SHOW DATABASES;

**8. How to Delete Data From a MySQL Table?**

In MySQL, the DELETE statement is used to delete records from a table:

DELETE FROM table\_name

WHERE column\_name = value\_name

**9. How to create an Index in MySQL?**

In MySQL, there are different index types, such as a regular INDEX, a PRIMARY KEY, or a FULLTEXT index. You can achieve fast searches with the help of an index. Indexes speed up performance by either ordering the data on disk so it's quicker to find your result or, telling the SQL engine where to go to find your data.

Example: Adding indexes to the history table:

ALTER TABLE history ADD INDEX(author(10));

ALTER TABLE history ADD INDEX(title(10));

ALTER TABLE history ADD INDEX(category(5));

ALTER TABLE history ADD INDEX(year);

DESCRIBE history;

**10. How do you remove a column from a database?**

You can remove a column by using the DROP keyword:

ALTER TABLE classics DROP pages;

**11. How do you Insert Data Into MySQL?**

The INSERT INTO statement is used to add new records to a MySQL table:

INSERT INTO table\_name (column1, column2, column3,...)

VALUES (value1, value2, value3,...)

If we want to add values for all the columns of the table, we do not need to specify the column names in the SQL query. However, the order of the values should be in the same order as the columns in the table. The INSERT INTO syntax would be as follows:

INSERT INTO table\_name

VALUES (value1, value2, value3, ...);

**12. How do you create a table using MySQL?**

Use the following to create a table using MySQL:

CREATE TABLE history (

author VARCHAR(128),

title VARCHAR(128),

type VARCHAR(16),

year CHAR(4)) ENGINE InnoDB;

**13. How do you create a database in MySQL?**

Use the following command to create a new database called ‘books’:

CREATE DATABASE books;

**14. What are some of the common MySQL commands?**

| **Command** | **Action** |
| --- | --- |
| ALTER | To alter a database or table |
| BACKUP | To back-up a table |
| \c | To cancel Input |
| CREATE | To create a database |
| DELETE | To delete a row from a table |
| DESCRIBE | To describe a table's columns |
| DROP | To delete a database or table |
| EXIT(ctrl+c) | To exit |
| GRANT | To change user privileges |
| HELP (\h, \?) | Display help |
| INSERT | Insert data |
| LOCK | Lock table(s) |
| QUIT(\q) | Same as EXIT |
| RENAME | Rename a Table |
| SHOW | List details about an object |
| SOURCE | Execute a file |
| STATUS (\s) | Display the current status |
| TRUNCATE | Empty a table |
| UNLOCK | Unlock table(s) |
| UPDATE | Update an existing record |
| USE | Use a database |

**15. What are MySQL Database Queries?**

A query is a specific request or a question. One can query a database for specific information and have a record returned.

**16. How can you interact with MySQL?**

There are three main ways you can interact with MySQL:

* using a command line
* via a web interface
* through a programming language

**17. What does a MySQL database contain?**

A MySQL database contains one or more tables, each of which contains records or rows. Within these rows are various columns or fields that contain the data itself.

**18. What does SQL in MySQL stand for?**

The SQL in MySQL stands for Structured Query Language. This language is also used in other databases such as Oracle and Microsoft SQL Server.  One can use commands such as the following to send requests from a database:

SELECT title FROM publications WHERE author = ' J. K. Rowling’;

**19. What do you mean by ‘databases’?**

A database is a structured collection of data stored in a computer system and organized in a way to be quickly searched. With databases, information can be rapidly retrieved.

**20. What are some of the advantages of using MySQL?**

* Flexibility: MySQL runs on all operating systems
* Power: MySQL focuses on performance
* Enterprise-Level SQL Features: MySQL had for some time been lacking in advanced features such as subqueries, [**views**](https://www.scaler.com/topics/views-in-sql/), and stored procedures.
* Full-Text Indexing and Searching
* Query Caching: This helps enhance the speed of MySQL greatly
* Replication: One MySQL server can be duplicated on another, providing numerous advantages
* Configuration and Security.

**ORDER BY**

The ORDER BY keyword is used to sort the result-set in ascending or descending order.

**LIMIT CLAUSE**

The LIMIT clause is used to specify the number of records to return.

The LIMIT clause is useful on large tables with thousands of records. Returning a large number of records can impact performance.

**CREATE INDEX**

The CREATE INDEX statement is used to create indexes in tables.

Creating an index on a column in a table can significantly improve the performance of queries that search for specific values in that column.

Indexes are used to retrieve data from the database more quickly than otherwise. The users cannot see the indexes, they are just used to speed up searches/queries.

**Intermediate MySQL Interview Questions**

**1. What are the types of relationships used in MySQL?**

There are three categories of relationships in MySQL:

* **One-to-One**: Usually, when two items have a one-to-one relationship, you just include them as columns in the same table.
* **One-to-Many**: One-to-many (or many-to-one) relationships occur when one row in one table is linked to many rows in another table.
* **Many-to-Many**: In a many-to-many relationship, many rows in one table are linked to many rows in another table. To create this relationship, add a third table containing the same key column from each of the other tables

**2. What are the MySQL clients and utilities?**

Several MySQL programs are available to help you communicate with the server. For administrative tasks, some of the most important ones are listed here:

• **mysql**—An interactive program that enables you to send SQL statements to the server and to view the results. You can also use mysql to execute batch scripts (text files containing SQL statements).

• **mysqladmin**—An administrative program for performing tasks such as shutting down the server, checking its configuration, or monitoring its status if it appears not to be functioning properly.

• **mysqldump**—A tool for backing up your databases or copying databases to another server.

• **mysqlcheck and myisamchk**—Programs that help you perform table checking, analysis, and optimization, as well as repairs if tables become damaged. mysqlcheck works with MyISAM tables and to some extent with tables for other storage engines. myisamchk is for use only with MyISAM tables.

**3. What is the MySQL server?**

The server, mysqld, is the hub of a MySQL installation; it performs all manipulation of databases and tables.

**4. How many Triggers are possible in MySQL?**

There are six Triggers allowed to use in the MySQL database:

* Before Insert
* After Insert
* Before Update
* After Update
* Before Delete
* After Delete

**5. What are MySQL Triggers?**

A trigger is a task that executes in response to some predefined database event, such as after a new row is added to a particular table. Specifically, this event involves inserting, modifying, or deleting table data, and the task can occur either prior to or immediately following any such event.   
Triggers have many purposes, including:

* Audit Trails
* Validation
* Referential integrity enforcement

**6. How do you create and execute views in MySQL?**

Creating a view is accomplished with the CREATE VIEW statement. As an example:

CREATE

[OR REPLACE]

[ALGORITHM = {MERGE | TEMPTABLE | UNDEFINED }]

[DEFINER = { user | CURRENT\_USER }]

[SQL SECURITY { DEFINER | INVOKER }]

VIEW view\_name [(column\_list)]

AS select\_statement

[WITH [CASCADED | LOCAL] CHECK OPTION]

**7. What are MySQL “Views”?**

In MySQL, a view consists of a set of rows that is returned if a particular query is executed. This is also known as a ‘virtual table’. Views make it easy to retrieve the way of making the query available via an alias.   
The advantages of views are:

* Simplicity
* Security
* Maintainability

**Advanced MySQL Interview Questions**

**1. What are Transaction Storage Engines in MySQL?**

To be able to use MySQL’s transaction facility, you have to be using MySQL’s InnoDB storage engine (which is the default from version 5.5 onward). If you are not sure which version of MySQL your code will be running on, rather than assuming InnoDB is the default engine you can force its use when creating a table, as follows.

**2. What is Sharding in SQL?**

The process of breaking up large tables into smaller chunks (called shards) that are spread across multiple servers is called Sharding.   
The advantage of Sharding is that since the sharded database is generally much smaller than the original; queries, maintenance, and all other tasks are much faster.

**3. What is Scaling in MySQL?**

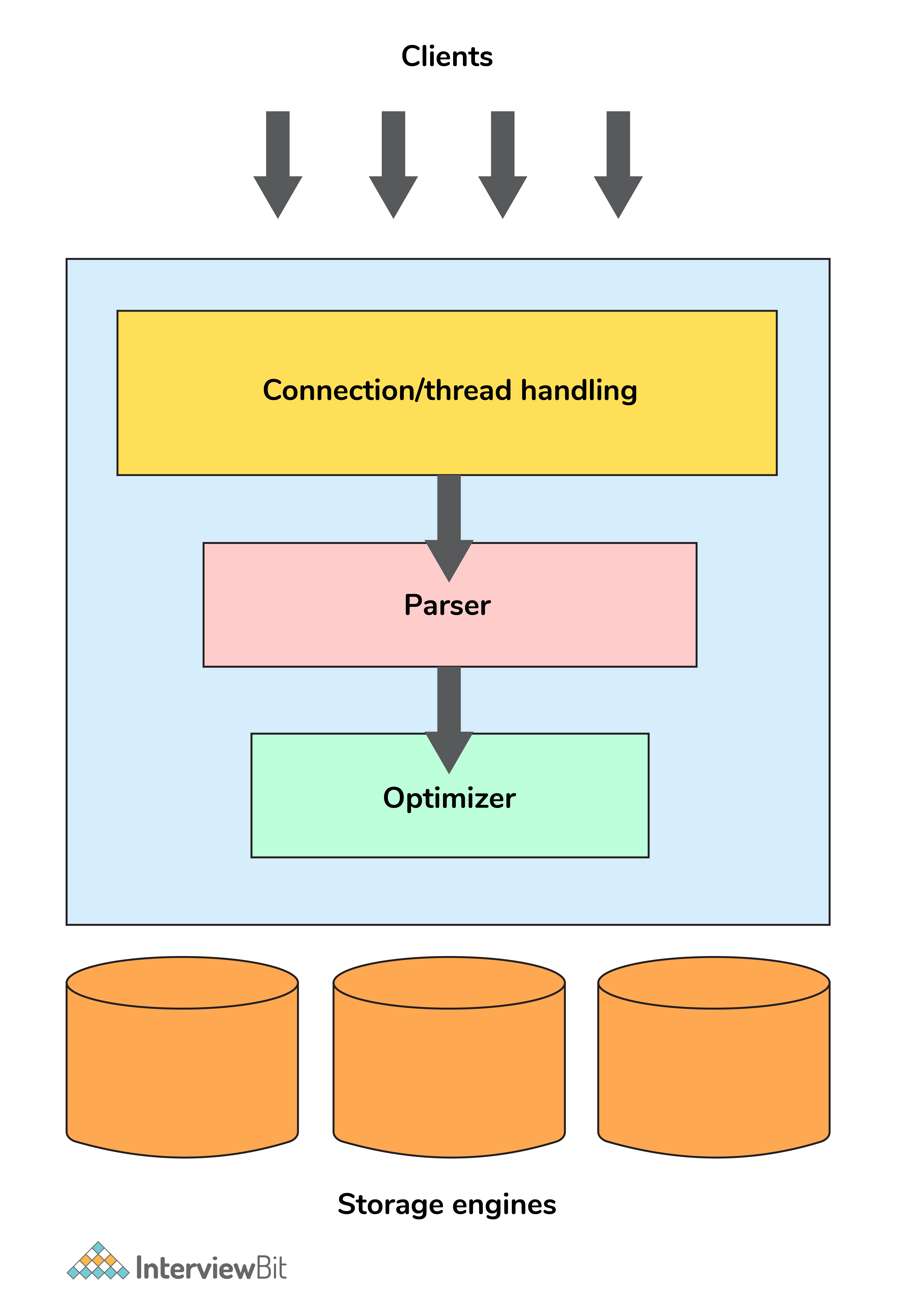
In MySQL, scaling capacity is actually the ability to handle the load, and it’s useful to think of load from several different angles such as:

* Quantity of data
* Number of users
* User activity
* Size of related datasets

**4. Can you explain the logical architecture of MySQL?**

The top layer contains the services most network-based client/server tools or servers need such as connection handling, authentication, security, and so forth.  
The second layer contains much of MySQL’s brains. This has the code for query parsing, analysis, optimization, caching, and all the built-in functions.

The third layer contains the storage engines that are responsible for storing and retrieving the data stored in MySQL.



**Data Definition Keywords:**

1. **CREATE**: Used to create databases, tables, indexes, and other objects.
   * Example: CREATE TABLE table\_name (column1 datatype, column2 datatype);
2. **DROP**: Deletes databases, tables, or other objects.
   * Example: DROP TABLE table\_name;
3. **ALTER**: Modifies the structure of an existing database object, like a table.
   * Example: ALTER TABLE table\_name ADD column\_name datatype;

**Data Manipulation Keywords:**

1. **SELECT**: Retrieves data from one or more tables.
   * Example: SELECT \* FROM table\_name;
2. **INSERT**: Adds new records to a table.
   * Example: INSERT INTO table\_name (column1, column2) VALUES (value1, value2);
3. **UPDATE**: Modifies existing records in a table.
   * Example: UPDATE table\_name SET column1 = value1 WHERE condition;
4. **DELETE**: Removes records from a table.
   * Example: DELETE FROM table\_name WHERE condition;

**Transaction Control Keywords:**

1. **COMMIT**: Saves all changes made in the current transaction.
   * Example: COMMIT;
2. **ROLLBACK**: Undoes all changes made in the current transaction.
   * Example: ROLLBACK;
3. **SAVEPOINT**: Sets a point within a transaction to which you can later roll back.
   * Example: SAVEPOINT savepoint\_name;

**Query Control Keywords:**

1. **WHERE**: Filters records based on specified conditions.
   * Example: SELECT \* FROM table\_name WHERE condition;
2. **ORDER BY**: Sorts the result set based on one or more columns.
   * Example: SELECT \* FROM table\_name ORDER BY column1 ASC;
3. **GROUP BY**: Groups rows that have the same values in specified columns.
   * Example: SELECT COUNT(\*), column\_name FROM table\_name GROUP BY column\_name;
4. **HAVING**: Filters groups created by the GROUP BY clause.
   * Example: SELECT COUNT(\*), column\_name FROM table\_name GROUP BY column\_name HAVING COUNT(\*) > 1;
5. **LIMIT**: Restricts the number of rows returned by a query.
   * Example: SELECT \* FROM table\_name LIMIT 5;

**Join and Set Operations Keywords:**

1. **JOIN**: Combines rows from two or more tables based on a related column.
   * Example: SELECT \* FROM table1 JOIN table2 ON table1.column = table2.column;
2. **UNION**: Combines the results of two or more SELECT statements.
   * Example: SELECT column1 FROM table1 UNION SELECT column1 FROM table2;
3. **INTERSECT** (not supported in MySQL): Returns rows that are common in two or more SELECT statements.
4. **EXCEPT** (not supported in MySQL): Returns rows from the first SELECT statement that are not in the second.

**Control Flow Keywords:**

1. **IF**: Allows conditional logic within queries.
   * Example: SELECT IF(condition, true\_value, false\_value);
2. **CASE**: Evaluates a list of conditions and returns one of multiple possible result expressions.
   * Example: SELECT CASE WHEN condition1 THEN result1 ELSE result2 END;

**Miscellaneous Keywords:**

1. **DISTINCT**: Removes duplicate rows from the result set.
   * Example: SELECT DISTINCT column FROM table\_name;
2. **LIKE**: Searches for a specified pattern in a column.
   * Example: SELECT \* FROM table\_name WHERE column\_name LIKE 'pattern%';
3. **IN**: Checks if a value matches any value in a list or subquery.
   * Example: SELECT \* FROM table\_name WHERE column\_name IN (value1, value2);
4. **BETWEEN**: Selects values within a specified range.
   * Example: SELECT \* FROM table\_name WHERE column\_name BETWEEN value1 AND value2;