

MySQL Interview Questions

1) 2nd maximum salary in a table

To find the second maximum salary in a table, you can use the following SQL query:

```
SELECT MAX(salary) FROM table_name WHERE salary < (SELECT MAX(salary) FROM table_name);
```

2) Alter table

The ALTER TABLE statement is used to modify the structure of an existing table in a database. You can use this statement to add or remove columns, change column data types, add or remove indexes, and modify table constraints.

3) Create table

The CREATE TABLE statement is used to create a new table in a database. You can use this statement to define the columns, data types, and constraints for the new table.

4) Drop, Delete and Truncate table difference

DROP TABLE is used to remove a table from a database. It permanently deletes the table and all of its data. DELETE is used to remove rows from a table. It can be used to delete specific rows or all rows in a table. TRUNCATE TABLE is used to remove all rows from a table. It is faster than DELETE because it does not log individual row deletions.

5) Foreign, Primary, Unique, Super key

In MySQL, a foreign key is a column or a set of columns in a table that refers to the primary key of another table. A primary key is a column or a set of columns that uniquely identifies each row in a table. A unique key is a column or a set of columns that must contain unique values. A super key is a set of columns that can uniquely identify each row in a table.

6) How to add a column in a table

To add a column to an existing table, you can use the ALTER TABLE statement with the ADD COLUMN clause. For example, the following query adds a new column called "new_column" to an existing table called "table_name":

```
ALTER TABLE table_name ADD COLUMN new_column datatype;
```

7) Fetch Odd/Even records in the table

To fetch odd or even records from a table, you can use the following SQL queries:

```
SELECT * FROM table_name WHERE id % 2 = 1; -- fetch odd records
```

```
SELECT * FROM table_name WHERE id % 2 = 0; -- fetch even records
```

8) Joins SQL

In SQL, join is used to combine rows from two or more tables based on a related column between them. There are different types of joins in SQL, including inner join, left join, right join, and full outer join.

9) SQL/MySQL difference

SQL is a standard query language used to manage relational databases. MySQL is an open-source relational database management system that uses SQL to manage databases. MySQL is a popular choice for web applications and is widely used in combination with PHP to create dynamic websites. The main difference between SQL and MySQL is that SQL is a standard language used to manage relational databases, while MySQL is a specific implementation of a relational database management system.

Java Interview Questions

1) Linear or Non-Linear Data structures

Data structures can be broadly classified into linear and non-linear data structures. Linear data structures include arrays, linked lists, stacks, and queues, where data is arranged in a linear sequence. Non-linear data structures include trees, graphs, and heaps, where data is arranged in a hierarchical or non-linear manner.

2) Data-types in Java

Java supports a variety of data types, including primitive data types like byte, short, int, long, float, double, char, and boolean, and reference data types like classes, interfaces, and arrays.

3) Singleton class

Singleton class is a design pattern in Java that restricts the instantiation of a class to a single object. It ensures that only one instance of the class is created and provides a global point of access to that instance.

4) Operations of stacks

A stack is a data structure that follows the Last In First Out (LIFO) principle. The operations that can be performed on a stack are push, pop, peek, and isEmpty.

5) Best sorting algorithm

The best sorting algorithm depends on the data to be sorted and the context in which it is used. Some commonly used sorting algorithms are quicksort, mergesort, heapsort, and insertion sort.

6) Complexity of sorting algorithm

The time complexity of sorting algorithms is usually measured in terms of the number of comparisons and swaps performed by the algorithm. The best-case, worst-case, and average-case time complexity can be calculated for each algorithm.

7) JDK, JRE & JVM difference

JDK stands for Java Development Kit, which is a software development kit used to develop Java applications. JRE stands for Java Runtime Environment, which is a runtime environment used to run Java applications. JVM stands for Java Virtual Machine, which is a virtual machine that provides a platform-independent execution environment for Java programs.

8) Access modifiers (public, private, protected etc.)

Access modifiers in Java are used to control the visibility and accessibility of class members (variables, methods, and inner classes). Public members can be accessed from any class, private

members can only be accessed from within the same class, and protected members can be accessed from within the same class or any subclass. Default (package-private) members can be accessed from within the same package.

9) Static variables and methods

Static variables and methods belong to the class rather than an instance of the class. They can be accessed using the class name and do not require an instance of the class to be created. Static variables have a single value for all instances of the class, while non-static variables have a separate value for each instance.

10) Packages

Packages in Java are used to group related classes and interfaces together. They help in organizing the code and avoiding naming conflicts. A package is defined using the package keyword and can contain one or more classes or interfaces. The naming convention for packages is to use reverse domain name notation.

11) Object oriented programming- Inheritance, Abstraction, Encapsulation, Polymorphism

Object oriented programming (OOP) is a programming paradigm that revolves around the use of objects. In OOP, there are four key concepts: Inheritance, Abstraction, Encapsulation, and Polymorphism.

Inheritance is the ability of a class to inherit properties and behavior from another class. It allows for code reuse and enables the creation of hierarchies of classes.

Abstraction is the process of hiding implementation details while showing only the necessary information to the user. It is achieved through abstract classes and interfaces.

Encapsulation is the process of wrapping data and methods together into a single unit, preventing outside access to the inner workings of the object. It helps in maintaining the integrity of the object and enables code reusability.

Polymorphism is the ability of an object to take on multiple forms or behaviors. It can be achieved through method overloading and method overriding.

12) Compile time and Run time Polymorphism

Polymorphism is the ability of an object to take on multiple forms. In Java, there are two types of polymorphism: compile-time and runtime polymorphism.

Compile-time polymorphism is achieved through method overloading, where multiple methods can have the same name but with different parameters. The compiler decides which method to execute based on the number and types of arguments passed.

Runtime polymorphism is achieved through method overriding, where a subclass can override the implementation of a method inherited from its superclass. The method called at runtime depends on the actual type of the object, not on the reference type.

13) Interface

An interface is a collection of abstract methods and constant variables that can be implemented by a class. It defines a contract between a class and the outside world, specifying what methods and properties a class should have. A class can implement multiple interfaces.

14) Constructor

A constructor is a special method that is called when an object of a class is created. It is used to initialize the object's state and allocate memory. In Java, a constructor has the same name as the class and can be overloaded to accept different parameters.

15) New, Final, super, this keywords

In Java, the new keyword is used to create an object of a class. The final keyword is used to define a constant variable or to prevent a class or method from being overridden. The super keyword is used to call a constructor or method from the superclass. The this keyword is used to refer to the current object.

16) Throw and throws, Difference between both of them

The throw keyword is used to explicitly throw an exception in Java, while the throws keyword is used to declare the types of exceptions that a method may throw. The main difference between them is that the throw keyword is used to throw an exception, while the throws keyword is used to declare that a method may throw an exception.

17) Exception handling

Exception handling is the process of handling unexpected or exceptional situations that may occur during program execution. In Java, exceptions are objects that represent errors or unexpected events that occur during program execution.

18) Try & catch statement, finally block

The try-catch statement is used to catch and handle exceptions in Java. It allows you to catch specific exceptions and take appropriate actions. The finally block is used to execute code that should always be executed, regardless of whether an exception is thrown or not.

19) POM, Maven, Beans

POM (Page Object Model) is a design pattern used in software testing to create a modular and maintainable test code. Maven is a build automation tool used primarily for Java projects. It helps manage project dependencies and automate the build process. Beans are reusable software components that can be manipulated visually in an IDE (Integrated Development Environment) and used in Java applications.

20) Multithreading

Multithreading is the ability of a program to perform multiple tasks concurrently. In Java, multithreading is achieved through the Thread class, which allows multiple threads of execution to run concurrently within a single program. Multithreading can help improve program performance and responsiveness. However, it requires careful handling to avoid issues such as race conditions, deadlocks, and thread starvation. Some common techniques for managing multithreading include synchronization, locking, and atomic variables. Java also provides higher-level concurrency constructs such as Executors, which allow for easy creation and management of thread pools, and synchronized collections, which enable safe access to shared data structures. Proper understanding and implementation of multithreading can significantly improve the efficiency and performance of Java programs.

21) Difference between threads and process

The main difference between a thread and a process is that a process is an instance of a running program while a thread is a subset of a process. A process has its own memory space, while threads share the same memory space within a process. Each process runs in a separate address space and has its own copy of data and code, while threads share the same data and code but have their own execution stacks.

22) Deadlock

Deadlock is a situation in which two or more threads are blocked waiting for each other to release the resources they need to proceed. This can happen when two or more threads have a dependency on each other, and each thread is waiting for the other to complete its task. Deadlocks can cause a program to become unresponsive, and careful programming is needed to avoid them.

23) Difference between abstraction and interface

Abstraction and interface are both mechanisms for achieving abstraction in Java, but they differ in their implementation.

Abstraction is achieved through abstract classes, which can have both abstract and non-abstract methods. Abstract classes can also have state, i.e., instance variables. A class can extend only one abstract class.

An interface, on the other hand, is a collection of abstract methods and constant variables. It does not have any implementation and cannot have any state. A class can implement multiple interfaces.

24) Difference between static and final

The static keyword is used to define class-level variables and methods that can be accessed without creating an instance of the class. The final keyword is used to define constant variables that cannot be changed after initialization, or to prevent a class or method from being overridden.

25) Types of inheritance

There are four types of inheritance in Java:

Single Inheritance: A class can only inherit from a single superclass.

Multilevel Inheritance: A class can inherit from a superclass, which in turn can inherit from another superclass.

Hierarchical Inheritance: Multiple classes can inherit from a single superclass.

Multiple Inheritance: A class can inherit from multiple super classes, which is not supported in Java, but can be achieved through interfaces.

26) Difference between Multi-level and Multiple inheritance

Multilevel inheritance is a type of inheritance in which a class inherits from a superclass, which in turn inherits from another superclass. Multiple inheritance is a type of inheritance in which a class inherits from multiple super classes. Multiple inheritance is not directly supported in Java, but it can be achieved through interfaces, where a class can implement multiple interfaces.

27) Why wait and notify is declared in Object class instead of Thread?

The wait() and notify() methods are used for inter-thread communication in Java. They are declared in the Object class because they are used to control the behavior of an object's monitor, which is a fundamental synchronization mechanism in Java. Thread is just one of the entities that can hold a lock on an object's monitor, so it wouldn't make sense to declare these methods in the Thread class.

28) Why does Java not support operator overloading?

Java does not support operator overloading because it can lead to ambiguity and make the code harder to read and understand. Instead, Java provides a set of well-defined operators that work with specific types and follow a consistent set of rules.

29) How to create a thread-safe singleton in Java using double-checked locking?

To create a thread-safe singleton in Java using double-checked locking, you can declare the singleton instance as a volatile variable and use a synchronized block to create the instance if it has not already been created. Here's an example:

Copy code

```
public class MySingleton {  
  
    private static volatile MySingleton instance;
```



```

private MySingleton() {}

public static MySingleton getInstance() {
    if (instance == null) {
        synchronized (MySingleton.class) {
            if (instance == null) {
                instance = new MySingleton();
            }
        }
    }
    return instance;
}
}

```

30) Can you override the static method in Java? If I create the same method in the subclass, is it a compile-time error?

No, you cannot override a static method in Java because static methods are bound to the class, not the object. If you create the same method in a subclass, it is not considered an override and is simply a new method with the same name. It is not a compile-time error, but it may lead to confusion and unexpected behavior if the wrong method is called.

31) What is the difference between an Array List and vector in Java?

Both Array List and Vector are classes that implement the List interface in Java. The main difference between them is that Vector is synchronized and therefore thread-safe, whereas Array List is not. Vector is also generally considered to be less efficient than Array List because of the overhead of synchronization.

32) How do you ensure that the N thread can access N resources without deadlock?

To ensure that N threads can access N resources without deadlock, you can use a resource allocation algorithm such as the banker's algorithm or the wait-for graph algorithm. These algorithms ensure that resources are allocated in a safe and deadlock-free manner by tracking resource usage and predicting future needs.

33) Can you access a non-static variable in the static context?

No, you cannot access a non-static variable in the static context because non-static variables are associated with a specific instance of a class, whereas static variables are associated with the class itself. In other words, non-static variables require an instance of the class to be created, whereas static variables do not.

34) How can we make anything immutable in Java?

To make an object immutable in Java, you can follow these steps:

Declare the class as final to prevent inheritance.

Declare all fields as final to prevent modification.

Do not provide any setters for the fields.

If the object contains mutable fields, ensure that they are not exposed to the outside world.

Use defensive copying to ensure that any returned values are also immutable.

35) In a distributed server system, how do we manage load balancing?

Load balancing in a distributed server system is typically managed using a load balancer, which distributes incoming requests across multiple servers to ensure that no single server becomes overloaded. There are several types of load balancers, including hardware load balancers, software load balancers, and DNS-based load balancers.

36) Diamond problem

The diamond problem is a problem that occurs in multiple inheritance when two or more parent classes of a subclass have a common method with the same name and signature. This can lead to ambiguity and make it unclear which method should be called. In Java, this problem is solved using default methods, which provide a default implementation of the method that can be overridden in the subclass if necessary.

37) Give a real-life example of multithreading.

A real-life example of multithreading is a web server that handles multiple requests simultaneously. Each request can be handled by a separate thread, allowing the server to handle multiple requests at once and improve performance.

38) What is notify and notify all?

notify() and notifyAll() are methods used for inter-thread communication in Java. They are used to wake up threads that are waiting on an object's monitor. notify() wakes up a single thread that is waiting on the monitor, whereas notifyAll() wakes up all threads that are waiting on the monitor.

39) HashMap Implementation Theory

HashMap is an implementation of the Map interface in Java that uses a hash table to store key-value pairs. The hash table is implemented as an array of buckets, where each bucket contains a linked list of entries. To store a key-value pair in the hash table, the key is first hashed to determine the bucket index, and the entry is added to the linked list in that bucket. When retrieving a value from the hash table, the key is hashed to determine the bucket index, and then the linked list in that bucket is searched for the entry with the matching key.

40) What is a constructor?

A constructor is a special method in a class that is used to create and initialize an object of that class. It is called when an object is created using the new keyword and is used to set the initial values of the object's state. Constructors can have parameters, which are used to pass values to the object's state during initialization. If a class does not define a constructor, a default constructor is automatically provided by the compiler.

41) How exception handling works in Java?

In Java, exceptions are objects that represent errors or exceptional conditions that occur during program execution. When an exception occurs, the program throws an exception object, which contains information about the error. The exception is then caught by an exception handler, which is a block of code that is designed to handle the exception. The exception handler can either handle the exception or pass it up the call stack to a higher-level exception handler. If an exception is not caught by any exception handler, the program terminates.

42) What are different types of exceptions?

There are two types of exceptions in Java: checked exceptions and unchecked exceptions. Checked exceptions are exceptions that the compiler requires to be handled by the calling code. Examples of checked exceptions include `IOException` and `ClassNotFoundException`. Unchecked exceptions, on the other hand, are exceptions that do not need to be handled by the calling code. Examples of unchecked exceptions include `NullPointerException` and `ArrayIndexOutOfBoundsException`.

43) What is multi-threading and multitasking in Java?

Multithreading is a technique in which a program can perform multiple tasks concurrently by creating multiple threads of execution. Each thread executes a separate set of instructions, allowing the program to perform multiple tasks simultaneously. Multitasking, on the other hand, is the ability of an operating system to run multiple programs or processes concurrently, allowing the user to perform multiple tasks simultaneously. Java supports both multithreading and multitasking.

44) What are wait and sleep methods?

`wait()` and `sleep()` are methods used for thread synchronization in Java. `wait()` is used to pause the current thread and release the object's monitor, allowing other threads to access the object. It is typically used in conjunction with `notify()` or `notifyAll()` to implement inter-thread communication. `sleep()`, on the other hand, is used to pause the current thread for a specified amount of time. It does not release the object's monitor, and is typically used for simple timing operations.

45) Difference between HashSet and HashMap

`HashSet` and `HashMap` are both implementations of the `Set` and `Map` interfaces in Java, respectively. `HashSet` stores elements as keys in a `HashMap` with null values, while `HashMap` stores key-value pairs. `HashSet` ensures that all elements are unique and does not allow duplicate elements, while `HashMap` allows duplicate values but not duplicate keys. Additionally, `HashSet` is unordered, while `HashMap` maintains the order of the elements based on their insertion order or an explicitly specified order.

46) How many ways to make a thread?

There are two ways to create a thread in Java: by extending the `Thread` class and by implementing the `Runnable` interface. When a class extends the `Thread` class, it inherits the `run()` method, which is used to define the thread's behavior. When a class implements the `Runnable`

interface, it must provide an implementation of the `run()` method. The `Runnable` interface is typically preferred over extending the `Thread` class because it allows for better code organization and flexibility.

47) Compile-time errors vs runtime errors

Compile-time errors are errors that occur during compilation and prevent the program from being compiled. They are usually caused by syntax errors, type errors, or other issues that violate the language's rules. Runtime errors, on the other hand, are errors that occur during program execution and cause the program to terminate or behave unexpectedly. They are usually caused by issues such as null pointer dereference, array out of bounds, or division by zero.

What is a final class?

A final class is a class that cannot be subclassed. When a class is declared as `final`, it cannot be extended by any other class. This is typically used to prevent the class from being modified or overridden, ensuring that the class's behavior remains consistent across all instances. Additionally, final classes can be optimized by the compiler for improved performance, since the class cannot be modified at runtime.

30. What is the finally block?

The finally block is a block of code that is executed regardless of whether an exception is thrown or not. It is typically used to release resources, such as file handles or network connections, that were acquired in a try block. The finally block is guaranteed to be executed, even if an exception is thrown or the program terminates unexpectedly.

What is the need for the finally block?

The finally block is needed to ensure that critical resources are released, even in the case of an exception or unexpected termination of the program. Without the finally block, resources may not be properly released, leading to resource leaks or other issues. The finally block ensures that the program always exits cleanly, even in the case of unexpected errors.

What is the difference between `final`, `finally`, and `finalize`?

`final` is a keyword used to declare that a variable, method, or class cannot be modified. `finally` is a keyword used to define a block of code that is guaranteed to be executed, regardless of whether an exception is thrown or not. `finalize` is a method that is called by the garbage collector when an object is no longer needed, allowing the object to perform any necessary cleanup operations before it is destroyed.

What are generic classes? And what are its uses?

Generic classes are classes that can work with multiple types of data, allowing for greater code reuse and flexibility. They are defined using type parameters, which are placeholders for the actual types of data that will be used. Generic classes can be used to create reusable data structures, algorithms, and other types of code that work with different types of data. Some of the most commonly used generic classes in Java include ArrayList, HashMap, and HashSet.

Can we use the Object class instead of a generic class for collection?

Yes, it is possible to use the Object class instead of a generic class for collection. However, this approach is generally not recommended, as it can lead to issues with type safety and performance. When using the Object class, it is necessary to cast objects to the appropriate type when retrieving them, which can lead to runtime errors if the wrong type is used. Additionally, using the Object class can result in unnecessary boxing and unboxing operations, leading to reduced performance.

Start() method calls the run method for thread creation. Why don't we directly use the run() method?

The start() method is used to create a new thread of execution and begin running the thread's code in parallel with the main thread. When the start() method is called, the JVM creates a new thread and invokes the run() method in the new thread. If the run() method were called directly, it would execute in the context of the main thread, rather than in a new thread of execution. This would not allow for parallelism and would not provide the benefits of multithreading.

Spring/Spring boot Interview Questions

1) Spring mvc and Spring boot difference

Spring MVC is a framework for building web applications using the Model-View-Controller (MVC) pattern. It provides various features such as controller, views, and model components for building web applications. Spring Boot is an extension of the Spring Framework that simplifies the process of building standalone, production-grade Spring-based applications by providing a pre-configured environment.

2) Spring and Spring boot difference

The main difference between Spring and Spring Boot is that Spring is a framework that provides various features for building complex applications, while Spring Boot is an extension of the Spring Framework that provides a pre-configured environment for building standalone, production-grade Spring-based applications. Spring Boot is designed to reduce the amount of

boilerplate code needed to configure and run Spring applications, while still leveraging the power of the Spring Framework.

3) ORM

ORM stands for Object-Relational Mapping, which is a technique used to map objects to relational database tables. It provides a way to map Java objects to database tables and vice versa. Some popular ORM frameworks for Java include Hibernate, MyBatis, and JPA.

4) Spring security

Spring Security is a powerful and highly customizable authentication and access-control framework for Spring-based applications. It provides various features such as authentication, authorization, and security for web applications.

5) Types of authentication in Rest API

There are several types of authentication methods in REST API, including:

Basic Authentication: This method uses a username and password for authentication.

Token-Based Authentication: This method uses a token to authenticate users.

OAuth 2.0 Authentication: This method uses an access token to authenticate users.

JSON Web Token (JWT) Authentication: This method uses a JSON Web Token to authenticate users.

6) Inject and auto wired annotation difference

The @Inject and @Autowired annotations are used for dependency injection in Spring. The main difference between the two is that @Autowired is a Spring-specific annotation while @Inject is a Java EE specification. Both annotations can be used to inject dependencies, but @Autowired is more commonly used in Spring applications.

7) Rest controller, controller annotation and their difference

@Controller is an annotation used to mark a class as a controller in Spring. @RestController is a specialized version of the @Controller annotation that is used to create RESTful web services. The main difference between the two is that @RestController combines the functionality of @Controller and @ResponseBody annotations into a single annotation.

8) Response body, Request parameter and path variable and their difference

@RequestBody is an annotation used to map the HTTP request body to a Java object.
@RequestParam is an annotation used to extract a single parameter from the request URL.
@PathVariable is an annotation used to extract a parameter from the request URL. The main difference between the three is that @RequestBody maps the entire request body to a Java object, while @RequestParam and @PathVariable extract individual parameters from the request URL.

9) Mapping in spring boot

Mapping in Spring Boot refers to the process of mapping HTTP requests to specific controller methods. Spring Boot provides various annotations such as @RequestMapping, @GetMapping, @PostMapping, etc. to map requests to controller methods based on the request type and URL.

10) How to connect with MySQL

To connect with MySQL in Spring Boot, you can use the Spring Data JPA framework. This framework provides an easy-to-use API for working with databases in Spring applications. You can configure the database connection properties in the application.properties file and use the @Repository annotation to create data access objects (DAOs) for interacting with the database.

11) How to implement join in spring boot

To implement joins in Spring Boot, you can use the Spring Data JPA framework. This framework provides various annotations such as @OneToMany, @ManyToOne, @ManyToMany, etc. to map relationships between entities in a database. You can use these annotations to define relationships between entities and use JPQL or native SQL queries to perform joins between tables.

12) What is bean, scope of bean?

In Spring framework, a bean is a Java object that is managed by the Spring IoC (Inversion of Control) container. A bean is defined in the Spring configuration file and its scope defines the lifecycle of the bean instance.

13) What is Kafka?

Apache Kafka is a distributed streaming platform that is used to publish and subscribe to streams of records. It is designed to be fast, scalable, and fault-tolerant.

14) What is Redis?

Redis is an in-memory data structure store that is used as a database, cache, and message broker. It supports a wide range of data structures such as strings, hashes, lists, sets, and sorted sets.

15) What are DI and IOC?

DI (Dependency Injection) is a design pattern that allows objects to be created and dependencies to be injected at runtime. IOC (Inversion of Control) is a design principle where the control flow of a program is inverted to allow for loose coupling and better code organization.

16) What is Spring?

Spring is an open-source Java framework that provides support for building web applications and services. It is based on the principles of DI and IOC and provides a wide range of features such as MVC architecture, security, data access, and integration with other frameworks.

17) Beans in Spring

In Spring, a bean is a Java object that is managed by the Spring IoC container. Beans are defined in the Spring configuration file and can be configured with various properties such as dependencies, scope, and lifecycle callbacks.

18) MVC architecture

MVC (Model-View-Controller) is a software architecture pattern that separates an application into three main components: the model (data and business logic), the view (user interface), and the controller (handles user input and updates the model and view accordingly). It is commonly used in web applications to separate concerns and promote maintainability and extensibility.