35+ Genpact Interview Questions and Answers

Genpact is a worldwide professional services organization specializing in business transformation. The Genpact interview process includes both Technical and Human Resources interviews as integral components. Getting a job at Genpact requires successfully navigating through these interviews.

35+ Genpact Interview Questions and Answers

In this post, you will learn the frequently asked technical questions during Genpact interviews along with their answers. These questions are divided into three categories i.e. basic, intermediate, and advanced based on the difficulty level of the questions. This post will assist you in the preparation of your interview.

These Genpact interview questions are broadly divided into major categories such as:

1. **Basic Genpact Interview Questions**
2. **Intermediate Genpact Interview Questions**
3. **Advanced Genpact Interview Questions**

Basic Genpact Interview Questions

1. What do you know about Genpact?

Genpact is a worldwide professional services organization that focuses on digital transformation, consulting, analytics, and business process services. Headquartered in New York City, the company distributes operations across more than 25 countries, catering to clients in industries such as banking, financial services, insurance, healthcare, life sciences, manufacturing, high-tech consumer goods, retail, and more.

2. Why do you want to work for Genpact?

Here, the interviewer is trying to determine whether you are genuinely interested in the company and whether your goals and values align with theirs. Therefore, you should learn about the company before going for the interview. Try to align your answer with the vision or goal of the company.

Apart from this, you can also mention that you are a keen learner and the environment of an international company like Genpact is perfect for you to launch your career, and advance your knowledge and abilities.

3. What are your strengths and weaknesses?

This question is asked to assess your self-awareness. While answering you should keep the requirements mentioned by the company in the job description, in mind. It means your answer should be relevant to the job description. To make an impression you can describe a situation that supports your claim. For example, if you say you're a great leader, then you should mention a situation proving your leadership skills. However, you should refrain from exaggerating your strengths.

In case of weaknesses, you should not mention something that could ruin your chance of getting the job. You should be honest but refrain from mentioning a flaw that is too personal or critical to the job. Also, you should show your willingness to overcome your weaknesses by mentioning some scenarios supporting your claim. For example, you can say that your weakness is that you feel uncomfortable when you are not able to complete a task on a given time. Or you do not like to submit work until and unless you are not satisfied with your work.

4. How do you handle pressure?

To properly respond to this question, you should give specific examples of times when you worked under pressure and handled the situation efficiently. For example, you can say that you had to complete multiple projects in a short period. However, you were able to finish all those projects ahead of schedule and avoid unnecessary stress because you made a schedule outlining how you would divide each project into smaller assignments.

5. What value do you bring to Genpact?

As an answer to this question, you should mention the skills the employer is looking for and how you have demonstrated them in the past. You can add your achievements and how you can use them for the benefit of the company. Overall, your answer should reflect that you have taken time to learn about the organization and the job role along with why you are a good fit for the company.

For example, you can mention that you have experience working on a particular technology, which seems to be the requirement of this role. You can add that you have been doing all of this in the same industry that Genpact operates in, so you won't need any extra time to learn the ins and outs of the industry.

6. What types of operating systems are you familiar with? Give some examples as well.

Operating systems can be divided into several types. Those types include network operating systems, multitasking operating systems, time-sharing operating systems, batch operating systems, real-time operating systems, multitasking operating systems, distributed operating systems, and mobile operating systems. UNIX, Multics, Amoeba, VMware ESXi, Novell NetWare, VxWorks, Linux, Windows, macOS.

7. What are real-time Operating Systems?

Real-time operating systems (RTOS) are the type of operating systems designed to meet the stringent timing requirements of real-time systems. Real-time systems are those in which the correctness of the system's behavior depends not only on the logical result of computation but also on the time at which the results are produced. These systems must respond to external stimuli or events within a predetermined and predictable time frame. Examples of real-time operating systems are airline traffic control systems, Command Control Systems, airline reservation systems, Heart pacemakers, Network Multimedia Systems, robots, etc.

8. What are the types of real-time Operating Systems?

Real-time operating systems can be classified into three different types and those types are:

Hard Real-Time Operating Systems: These systems have strict timing constraints. This means that the task must be completed in a specific period otherwise it will result in a serious failure. VxWorks and RTLinux are examples of Hard Real-Time Operating Systems.

Soft Real-Time Operating Systems: These systems have timing constraints, but occasional misses may be tolerable without any serious consequences. Examples of such software include Windows CE and Linux with real-time extensions.

Firm Real-Time Operating Systems: A firm real-time system is one in which a few missed deadlines won't result in the system failing, but missing multiple deadlines could have negative results.

9. What do you know about threads in Operating Systems?

In operating systems, a thread is the smallest unit of execution within a process. It represents an independent sequence of instructions that can be scheduled for execution. Threads within a process share the same resources, such as memory space, while each thread maintains its registers and stack. Thread states include running, ready, and blocked, and synchronization mechanisms like locks and semaphores are used to manage simultaneous access to shared resources.

10. What do you understand by multithreading?

Multithreading is a programming and execution model that enables the simultaneous execution of multiple threads within the same process. These threads share the same resources, such as memory space and open files, within the process while having their local data, registers, and stack. Multithreading allows for parallelism, improving the overall performance of an application by enabling simultaneous execution of multiple tasks.

11. What is a process in Operating Systems?

In operating systems, a process represents the execution of a program in a computer's memory. It is an independent and self-contained unit that consists of the program code, execution state, memory space, and system resources. Each process operates in isolation from others, ensuring data integrity and security. Processes are managed by the operating system's process scheduler, which determines the order and duration of their execution on the CPU. Each process

has its own address space, including a code section, data section, and stack, preventing one process from directly accessing the memory of another.

12. What are user-level threads?

User-level threads are threads managed by user-level software without direct involvement from the operating system kernel. In this threading model, a user-level thread library or runtime environment handles thread creation, scheduling, and synchronization within the application. User-level threads offer flexibility for developers to implement custom threading models and are often more portable across different operating systems.

13. What are kernel-level threads?

Kernel-level threads are threads that are directly managed and supported by the operating system kernel. These threads are visible to and managed by the operating system. Each kernel-level thread represents an independent unit of execution with its program counter, register set, and execution state. Because the operating system is directly involved in managing kernel-level threads, it can make decisions about thread scheduling, prioritization, and resource allocation. Kernel-level threads typically provide better parallelism and can fully leverage multi-core systems, as the operating system kernel can schedule threads on different processor cores.

14. What is the data independence?

Data independence is a concept in database management systems that refers to the separation of the database schema from the applications that use the data. It allows changes to be made in the database structure without affecting the application programs that access the data.

Intermediate Genpact Interview Questions

15. What are the types of data independence?

Data independence is divided into two main types: logical data independence and physical data independence. Both types ensure that changes in the database schema do not affect the application programs that use the data.

1. **Logical Data Independence:** Logical data independence refers to the ability to modify the logical structure or the middle level of the database management system without affecting the higher-level layers of the application program. If a new table is added, existing application programs should continue to function without modification. Similarly, modifications to existing tables or relationships should not require changes to the application code.
2. **Physical Data Independence:** Physical data independence refers to the ability to modify the physical storage structure of the data without affecting the application programs. Changes to the way data is stored, indexed, or organized at the physical level should not require modifications to the application code. For example, transitioning from one storage technology to another or reorganizing the data storage structure should be transparent to applications.

16. Why is multiple inheritance not supported in Java?

Multiple inheritance is not supported in Java to avoid the complexities and ambiguities that can arise from the “diamond problem”. The diamond problem occurs when a class inherits from two classes that have a common base class. It leads to ambiguity in method resolution if there are conflicting implementations in the parent classes. However, Java can support multiple inheritance through interfaces, allowing a class to implement multiple interfaces to achieve a form of polymorphism.

17. Why can’t we override a private or static method in Java?

In Java, the private methods are intentionally designed to be inaccessible outside the class in which they are defined, and as a result, they are not visible to subclasses. Since overriding implies providing a different implementation for a method in a subclass, private methods do not participate in this polymorphic behavior, and attempts to declare a private method with the same signature in a subclass are treated as a separate, unrelated method rather than an override.

Similarly, static methods are associated with the class itself, rather than instances of the class, and they are resolved at compile time based on the reference type, not the actual object type. Therefore, they are not subject to dynamic method dispatch, which is a fundamental mechanism in method overriding. The concept

of overriding involves selecting the most specific method implementation based on the actual type of the object, which does not apply to static methods. As a result, while it is possible to declare a static method with the same signature in a subclass, it is considered method hiding rather than overriding.

18. What is Data Warehousing?

A data warehouse is a storage system used to store large amounts of data. Data can flow into a data warehouse from multiple sources. It aggregates the data into a single, central, data store to support data analysis, machine learning, artificial intelligence (AI), and data mining.

19. What is the Entity Relationship (ER) model in DBMS?

The Entity-Relationship (ER) model is a data model used in database design to represent the relationships between entities in a system. In the ER model, entities are represented as tables, and the relationships between entities are depicted using lines connecting them. Each entity is defined by its attributes, and relationships can be one-to-one, one-to-many, or many-to-many.

20. What is a Transaction in DBMS?

A transaction is a logical unit of work that consists of one or more database operations, such as inserts, updates, or deletions. Transactions ensure the consistency and integrity of the database by enforcing the ACID properties i.e. Atomicity, Consistency, Isolation, and Durability.

21. Explain the ACID properties in DBMS.

The ACID properties, standing for Atomicity, Consistency, Isolation, and Durability, are a set of characteristics that guarantee the reliability and integrity of transactions in Database Management Systems.

1. **Atomicity:** This property is to make sure that a transaction is treated as a single, indivisible unit of work. Either all operations within the transaction are completed, and the changes are applied to the database, or none are applied. If any part of the transaction fails, the entire transaction is rolled back, leaving the database in its original state.

2. **Consistency:** Consistency ensures that a transaction brings the database from one valid state to another, following predefined principles of integrity. The execution of a transaction should not violate any integrity rules defined for the database. If a transaction results in a violation, it is rolled back, and the database remains consistent.
3. **Isolation:** Isolation ensures that the intermediate state of a transaction is not visible to other simultaneous transactions. Transactions execute as if they are the only transactions in the system, preventing interference between them.
4. **Durability:** Durability guarantees that once a transaction is committed, its changes are permanent and survive any subsequent system failures. The committed changes are stored persistently in the database, ensuring that, even in the event of a power outage, crash, or other failures, the database can be recovered to a consistent state.

22. What do you understand by the term garbage collection in Java?

Garbage collection is the process of automatically identifying and reclaiming memory occupied by objects that are no longer in use, freeing up resources and preventing memory leaks.

23. What is synchronization in Java?

Java synchronization refers to the ability to manage how many threads can simultaneously access a shared resource. In the multithreading concept, inconsistent results are produced when multiple threads attempt to access the shared resources simultaneously. For threads to communicate reliably, synchronization is required.

24. What is the System.gc() function used for?

In Java, the `System.gc()` function is a method that suggests to the Java Virtual Machine (JVM) to perform garbage collection. However, it’s important to note that calling `System.gc()` does not guarantee immediate garbage collection. The JVM is responsible for managing its garbage collection, and calling `System.gc()` is merely a suggestion to the JVM, not a command. The JVM may choose to ignore

the suggestion based on its own internal heuristics and memory management strategies.

25. What is a scalar function in DBMS?

In a Database Management System, a scalar function refers to a function that operates on a single value and returns a single result. Scalar functions are designed to manipulate or transform individual data items within a database. These functions are to perform diverse operations on a per-row basis. Examples of scalar functions include mathematical operations like ABS or ROUND, string manipulations such as CONCAT or SUBSTRING, and date/time functions like NOW or DATEADD.

26. What do you know about ‘this’ keyword in Java?

In Java, the ‘this’ keyword is a reference variable that is used to refer to the current object within a class. It is used when there is a need to distinguish between instance variables and parameters or local variables with the same name. When a method is called, and it has parameters with the same names as instance variables, the ‘this’ keyword helps differentiate between the two, making it clear that the reference is to the instance variable. Apart from this, it is commonly used within constructors to invoke another constructor in the same class or to pass the current object as a parameter to other methods. By using ‘this’, we can avoid naming conflicts in our code.

27. In DBMS, what is a self-join query used for?

A self-join query is used when we want to join a table with itself in an SQL query. It is done when a table contains a hierarchical relationship, and it allows for the retrieval of information by establishing connections between rows within the same table.

28. What is normalization in DBMS?

Normalization is a process of organizing and structuring relational databases to reduce data redundancy and improve data integrity. The goal of normalization is to eliminate data anomalies and inconsistencies that can arise from storing redundant information. It involves breaking down large tables into smaller, related tables and organizing the data to adhere to specific rules, known as normal forms.

29. What is meant by 1st, 2nd, and 3rd normal forms?

The normalization process involves organizing the data into tables and ensuring that relationships between tables are well-defined. The first three normal forms are stages of this normalization process.

1. **First Normal Form (1NF):** A table is in 1NF if it contains only atomic values. It means that each column in a 1NF table must hold only a single, indivisible piece of data. There should be no repeating groups or arrays in any column.
2. **Second Normal Form (2NF):** A table is in 2NF if it is in 1NF and all non-key attributes are fully functionally dependent on the primary key. In simple words, a table is in 2NF if it has a primary key, and all other columns are fully dependent on the entire primary key, not just a part of it.
3. **Third Normal Form (3NF):** A table is in 3NF if it is in 2NF and all the attributes are functionally dependent only on the primary key. In other words, no transitive dependencies should exist. If A is functionally dependent on B, and B is functionally dependent on C, then A should not be dependent on C.

Advanced Genpact Interview Questions

30. How would you differentiate between UNIQUE and DISTINCT keywords in DBMS?

UNIQUE: The “UNIQUE” keyword is typically used as a constraint in the database schema to ensure that each value in a column is unique. When a column is declared as “UNIQUE,” it means that each value in that column must be unique across all rows in the table. Attempting to insert or update a row with a value that already exists in the “UNIQUE” column will result in a constraint violation error.

DISTINCT: The “DISTINCT” keyword is used in queries to retrieve unique values from a specific column or combination of columns in the result set. When included in a SELECT statement, “DISTINCT” filters out duplicate values, ensuring that unique values are returned. It is applied at the query level, not at the schema level like “UNIQUE.”

31. How are DROP and TRUNCATE commands different?

The DROP command is used to delete database objects entirely, including tables, views, indexes, or even databases. When the DROP command is used on a table, it removes the entire table along with all its data, and the table structure is permanently deleted from the database. Whereas, the TRUNCATE command is used to remove all the rows from a table but retains the table structure for future use. TRUNCATE removes all the rows from a table, resetting any associated identity columns or sequences. However, it does not delete the table itself, and the table structure remains intact.

32. What are Hard Real-Time (HRT) Systems?

Hard Real-Time (HRT) systems represent a class of computing systems in which task execution schedules and accuracy are important. In other words, tasks must not only produce accurate results but must also meet strict deadlines. These systems are commonly found in safety-critical environments such as aerospace, automotive, medical devices, and industrial control systems, where failure to meet a deadline could have severe consequences.

33. What do ‘intension’ and ‘extension’ mean in database management systems?

The intension of a database encompasses its design, schema, and metadata, representing the blueprint of the data model. It includes information about the data types, relationships, constraints, and rules that govern the organization of data within the database. In simple words, the intension describes the inherent properties and structure of the data, providing a conceptual framework for understanding how information is stored and related.

On the other hand, the extension of a database refers to the actual data instances or records stored in the database at a given point in time. It represents the concrete, tangible data that adheres to the specifications outlined in the intension. The extension is dynamic and can change as data is inserted, updated, or deleted within the database. While the intention defines the schema and rules governing the data, the extension is the instantiation of that schema with specific data values.

34. What are the different levels of data abstraction in Database Management Systems?

In DBMS, data abstraction refers to the process of hiding the complex details of the database implementation and providing users with simplified views of the data. There are three main levels of data abstraction:

1. **Physical Level:** The physical level is the lowest level of abstraction. It deals with the actual storage and retrieval of data on the hardware. It includes details such as data structures, storage mechanisms, and indexing methods.
2. **Logical Level:** The logical level provides a bridge between the physical implementation and the way users perceive and interact with the data. It involves defining the database schema, specifying relationships between entities, and establishing integrity constraints. Users interact with the database at this level through query languages like SQL, focusing on the organization and structure of the data without concern for how it is physically stored.
3. **View Level or External Level:** The highest level of abstraction is the view level, which deals with the user interface and how different users or applications perceive the data. At this level, users are presented with customized views or subsets of the data based on their specific needs. Views are created using query languages and define how users can access and manipulate the data.

35. How is pattern matching done in SQL?

Pattern matching in SQL is typically performed using the ‘LIKE’ operator, which allows users to search for patterns within text data. The ‘LIKE’ operator is commonly used in conjunction with wildcard characters to represent unknown or variable parts of a string. The two main wildcard characters used in SQL for pattern matching are the percent sign ‘%’ and underscore ‘_’. The percent sign ‘%’ is used to represent zero or more characters and the underscore ‘_’ is used to represent a single character.

36. How do we create objects in DBMS?

Creating objects in DBMS refers to defining and setting up database elements such as tables, views, indexes, stored procedures, and more. In SQL-based relational database systems like MySQL, PostgreSQL, or Microsoft SQL Server, creating a table involves using the `CREATE TABLE` statement.

37. What is Index Hunting in DBMS?

The process of enhancing a collection of indexes is referred to as index hunting. This is done since indexes reduce the time needed to process queries and increase query performance. Utilizing the query optimizer, the process involves recommending the most efficient queries. Index hunting contains a comprehensive assessment of the indexes, query distribution, and overall performance impact. This evaluation aids in selecting and refining indexes to achieve optimal query execution.

38. How are the operators == and equals () different from each other in Java?

In Java, the `==` operator checks whether two object references point to the same memory address, while the `equals()` method can be overridden in a class to compare the content of objects based on specific criteria. When 'equals()' is invoked, we can define their logic for comparing the internal attributes of two objects to determine if they are considered equal.

39. What is a Java Singleton class and how is it created?

A Singleton class in Java is a class that is designed to have only one instance. This is used when exactly one object is needed to coordinate actions across the system, such as a configuration manager, logging service, or database connection pool. The Singleton pattern limits a class's instantiation to one "singleton" instance and offers a global point of access to that instance.

The implementation involves making the class's constructor private to prevent external instantiation and providing a static method within the class to return the single instance.

40. What is an aggregate function in DBMS?

An aggregate function is a function that performs an operation on a set of values and returns a single value as a result. These functions are commonly used with the GROUP BY statement to summarize and analyze data across multiple rows. Examples of aggregate functions include COUNT, which returns the number of rows in a group; SUM, which calculates the sum of values; AVG, which computes

the average of values; and MIN and MAX, which determine the minimum and maximum values, respectively.

41. What are time-sharing operating systems?

Time-sharing operating systems, also known as multitasking or multitasking systems, are computer operating systems designed to allow multiple users to simultaneously share and access a single computer's resources. The primary goal of time-sharing systems is to optimize the utilization of computing resources, particularly the CPU, by dividing its processing time among multiple users or tasks. In a time-sharing environment, each user or task is allocated a small time slice or quantum during which they can execute their programs or commands. The system rapidly switches between different tasks, giving the illusion that each user has a dedicated machine. Examples include Unix, Linux, and various versions of the Windows operating system.