

# DBMS Technical Interview Questions

[@code\\_yatra](#)

Created this 15 days ago



Practice now

11 submissions



Public



608



1



## About

Here are some important DBMS (Database Management System) interview questions with their answers. These questions cover a variety of topics, including SQL, transactions, normalization, and database design.



## Content



Questions (34)



Answers (34)



Leaderboard



Comments



Your submissions

1. What is DBMS?



Answer

A Database Management System (DBMS) is a software system designed to manage databases. It provides an interface to interact with the data, allowing for data creation, retrieval, updating, and deletion (CRUD operations). DBMS also handles data security, integrity, and concurrency. Examples include MySQL, PostgreSQL, Oracle, and SQL Server.

2. What is a View in DBMS?



Answer

A View is a virtual table that represents a stored query. It does not store data physically but provides a way to access data. Views can be used to: Simplify complex queries. Provide a security mechanism by restricting access to specific rows or columns. Present data in a different format.

**Stop reading**

3. What are the different types of DBMS Models?



Answer

The four main types of DBMS models are:

Hierarchical Model: Data is represented in a tree-like structure. Each record has a parent-child relationship. Network Model: Similar to the hierarchical model but allows a record to have multiple parent-child relationships. Relational Model: Data is stored in tables (relations) and can be accessed via SQL. Object-Oriented Model: Data is represented as objects, similar to objects in object-oriented programming.

4. What is the difference between OLTP and OLAP?



Answer

OLTP (Online Transaction Processing): Used for day-to-day operations, handling large numbers of small transactions such as inserts, updates, and deletes. It focuses on fast query processing and data integrity.

OLAP (Online Analytical Processing): Used for complex queries and data analysis. It is designed for querying large amounts of historical data for decision-making and reporting purposes. OLAP systems are optimized for read-heavy operation

5. What are the different types of Indexes in DBMS?



Answer

Unique Index: Ensures that no two rows have the same value in the indexed columns. Composite Index: An index on multiple columns of a table. Bitmap Index: Uses bitmap vectors for indexing, typically used for columns with low cardinality (few unique values). Full-Text Index: Index for textual data, enabling fast text searches.

Clustered Index: The data is stored in the order of the index. Non-Clustered Index: The index is separate from the data and contains pointers to the data.

 **Stop reading**

6. What is a Deadlock? How can it be prevented?  

 Answer

A deadlock is a situation in which two or more transactions are waiting for each other to release resources, causing them to be stuck indefinitely. Deadlocks can be prevented by:

Timeouts: Transactions are automatically rolled back after waiting for a certain period. Lock Ordering: Ensuring that transactions acquire locks in a predefined order. Deadlock Detection and Recovery: Detecting deadlocks and rolling back one or more transactions to break the cycle.

7. What is a Transaction?  

 Answer

A Transaction is a sequence of one or more operations performed as a single unit of work. A transaction has the following ACID properties:

Atomicity: All operations in the transaction must complete; if one operation fails, the entire transaction is rolled back. Consistency: The database must transition from one valid state to another. Isolation: Transactions are isolated from each other until they are completed. Durability: Once a transaction is committed, its changes are permanent, even in the case of system failure.

8. What is the difference between a Clustered Index and a Non-Clustered Index?  

 Answer

Clustered Index: The rows of data are stored physically in the order of the index. A table can have only one clustered index. Non-Clustered Index: The index is stored separately from the data, and it contains pointers to the actual data rows. A table can have multiple non-clustered indexes.

9. What is an Index in DBMS?  

 Answer **Stop reading**

An Index is a database object that improves the speed of data retrieval operations. It is created on a table to allow faster searches and queries. Indexes are similar to the index in a book and allow the DBMS to find rows more quickly. There are two types of indexes:

Clustered Index: The data rows are stored in the order of the index. There can only be one clustered index per table. Non-Clustered Index: A separate structure from the data table that stores pointers to the data. A table can have multiple non-clustered indexes.

10. What is a Primary Key?  Answer

A Primary Key is a unique identifier for a record in a database table. It ensures that no two rows in the table have the same value for the primary key column(s). A primary key must always be unique, and it cannot contain NULL values.

11. What is Normalization? Why is it needed?  Answer

Normalization is the process of organizing the attributes and tables of a database to reduce redundancy and dependency. It involves decomposing a database into smaller, manageable pieces (tables) and ensuring that the relationships between the data are properly structured. It is needed to:

Reduce data redundancy. Avoid update anomalies (insertion, deletion, modification anomalies). Improve data integrity.

The different normal forms are 1NF (First Normal Form), 2NF (Second Normal Form), 3NF (Third Normal Form), and BCNF (Boyce-Codd Normal Form).

12. What is a Subquery in SQL?  Answer

A Subquery is a query within another query, enclosed in parentheses. The subquery provides data for the outer query.

Stop reading

Single-row subquery: Returns only one row of results. Multi-row subquery: Returns multiple rows of results. Correlated subquery: Refers to columns in the outer query and is executed for each row processed by the outer query.

13. What is a Join in SQL? Explain different types of Joins.



Answer

A Join is used to combine rows from two or more tables based on a related column between them. The common types of joins are:

INNER JOIN: Returns rows when there is a match in both tables. LEFT JOIN (OUTER JOIN): Returns all rows from the left table, and the matched rows from the right table. If there is no match, NULL is returned. RIGHT JOIN (OUTER JOIN): Returns all rows from the right table, and the matched rows from the left table. FULL JOIN (OUTER JOIN): Returns rows when there is a match in one of the tables. CROSS JOIN: Returns the Cartesian product of both tables, i.e., all possible combinations of rows.

14. What are the advantages of a Distributed DBMS?



Answer

A Distributed DBMS offers the following advantages:

Scalability: Can scale horizontally by adding more nodes or servers. Reliability: If one node fails, the system can still function with the remaining nodes. Data Locality: Data can be stored closer to users, reducing access time. Fault Tolerance: Redundancy can ensure data availability even in case of failures.

15. What is a Foreign Key?



Answer

A Foreign Key is a column (or a set of columns) in a table that links to the primary key of another table. It ensures referential integrity by enforcing a relationship

between two tables. A foreign key can contain duplicate values and can accept NULL values.


 **Stop reading**

16. Difference between DBMS and RDBMS? 

 Answer

DBMS: Manages databases, stores data in different structures (hierarchical, network, etc.), and does not support relationships or ACID properties.

RDBMS: A type of DBMS that stores data in tables (rows and columns), supports relationships (using primary and foreign keys), and enforces ACID properties for reliable transactions.

17. What is Big Data? How does it differ from traditional DBMS? 

 Answer

Big Data refers to extremely large datasets that cannot be processed efficiently using traditional database systems. Big data technologies (e.g., Hadoop, Spark) are designed to handle volume, velocity, and variety of data. Key differences:

Traditional DBMS: Structured data, relies on relational models, designed for transactional workloads. Big Data: Handles unstructured and semi-structured data, distributed processing, and fault tolerance.

18. What is the difference between DELETE, TRUNCATE, and DROP commands in SQL? 

 Answer

DELETE: Removes records from a table based on a condition, but the table structure remains. It is a DML command and can be rolled back.

TRUNCATE: Removes all rows from a table and resets any auto-increment counters. It is faster than DELETE and is a DDL command, so it cannot be rolled back in most cases.

DROP: Completely removes a table, including its structure, from the database. This operation cannot be undone.

19. Explain the concept of Locking



Stop reading

Answer

Locking is a mechanism used to manage concurrent access to database resources. When one transaction locks a resource, other transactions are blocked from accessing it until the lock is released.

Types of locks:

Shared Lock: Allows multiple transactions to read a resource, but not write to it.

Exclusive Lock: Allows only one transaction to access a resource, both for reading and writing.

20. What is the difference between 1NF, 2NF, and 3NF?



Answer

1NF (First Normal Form): Ensures that each column contains atomic (indivisible) values, and each column contains values of the same type.

2NF (Second Normal Form): Builds on 1NF and removes partial dependency, i.e., all non-key attributes must depend on the entire primary key.

3NF (Third Normal Form): Ensures that there is no transitive dependency, i.e., non-key attributes should not depend on other non-key attributes.

21. What is a Composite Key?



Answer

A Composite Key is a primary key that consists of two or more columns to uniquely identify a record in a table. This is necessary when a single column is not sufficient to uniquely identify a record.

Example: If a table stores data for multiple students in various courses, a composite key might consist of student\_id and course\_id to uniquely identify each enrollment record.

22. What is an Entity-Relationship (ER) Diagram?



 Answer **Stop reading**

An Entity-Relationship (ER) Diagram is a graphical representation of the entities in a database and their relationships. It helps in designing and modeling the database schema.

Entities represent objects or concepts (e.g., Employee, Department). Attributes describe the properties of entities (e.g., EmployeeID, Name). Relationships describe how entities are related (e.g., an Employee works in a Department).

23. What is a Join Condition in SQL?  Answer

A Join Condition is a condition used in SQL to combine rows from two or more tables based on a related column. It ensures that the rows from the tables are matched according to the specified criteria.

Common join conditions use the equality operator (=), such as `table1.id = table2.table1_id`, but other conditions can be used as well.

24. What is Referential Integrity?  Answer

Referential Integrity ensures that relationships between tables remain consistent. Specifically, it means that a foreign key in one table must match a primary key in another table (or be NULL). This prevents "orphaned" records, i.e., foreign key values that do not correspond to any primary key.

25. What is the difference between UNION and UNION ALL in SQL?  Answer

UNION: Combines the result sets of two or more queries, removing duplicate rows. It performs a DISTINCT operation. UNION ALL: Combines the result sets of two or more queries but does not remove duplicates. It is faster than UNION because it does not perform a DISTINCT operation.



26. What is an Aggregate Function?

 Stop reading Answer

Answer: An Aggregate Function performs a calculation on a set of values and returns a single result. Common aggregate functions in SQL are:

COUNT(): Returns the number of rows. SUM(): Returns the total sum of a numeric column. AVG(): Returns the average value of a numeric column. MAX(): Returns the maximum value. MIN(): Returns the minimum value.

27. What are Triggers in SQL?

 Answer

Triggers are special types of stored procedures that are automatically executed in response to specific events on a table or view, such as an insert, update, or delete. Example: A trigger can be used to automatically update a "LastModified" timestamp whenever a record is updated.

28. What is a Stored Procedure?

 Answer

A Stored Procedure is a precompiled collection of one or more SQL statements that can be executed as a single unit. Stored procedures are stored in the database and



Become QSet.io paid member! - 50% off (First 500)

29. What is a Self Join?

 Answer

A Self Join is a join where a table is joined with itself. It is useful when you need to compare rows within the same table. Example: A "Employee" table with an employee\_id and manager\_id can be self-joined to find the name of each employee's manager.

30. What is Denormalization?

**Stop reading**

Answer

Denormalization is the process of intentionally introducing redundancy into a database by merging tables or duplicating data. While normalization reduces redundancy, denormalization can improve query performance in some cases, especially for read-heavy applications. It is often used in OLAP systems where query performance is prioritized over storage efficiency.

31. What is Database Sharding?



Answer

Sharding is the process of splitting a large database into smaller, more manageable pieces, called "shards." Each shard is stored on a separate server or database instance. Sharding can improve performance and scalability, especially for large-scale applications. It is commonly used in distributed databases and NoSQL systems.

32. What is Data Redundancy and how can it be avoided?



Answer

Data Redundancy occurs when the same piece of data is stored in multiple places, leading to increased storage and potential inconsistencies. Redundancy can be avoided by:

Normalization: Organizing data into separate, related tables to eliminate duplication. Using foreign keys to link tables rather than duplicating data.

33. What is SQL Injection, and how can it be prevented?



Answer

SQL Injection is a type of security vulnerability where an attacker can execute arbitrary SQL code in a database by manipulating input data. It can be prevented by:

Using prepared statements and parameterized queries. Employing input validation and escaping dangerous characters. Limiting the database permissions for the

application.

Stop reading

34. Explain ACID Properties of a Transaction.

Answer

The ACID properties guarantee reliable processing of transactions:

Atomicity: The transaction is treated as a single unit, which either completes in full or does not execute at all (all or nothing). Consistency: A transaction brings the database from one valid state to another valid state. Isolation: Transactions are isolated from each other, so the intermediate states of a transaction are not visible to other transactions. Durability: Once a transaction is committed, its effects are permanent, even in the case of a system crash.

## More question sets



@code\_yatra

6 days ago

### TCS Coding Questions

Here is the most asked TCS technical interview coding questions

View



@code\_yatra

15 days ago

### DBMS Technical Interview Questions

Here are some important DBMS (Database Management System) interview questions with t

View