

- 1. Q: What is a Database Management System (DBMS)?**
A: A DBMS is software used to manage and organize databases, enabling users to store, retrieve, update, and delete data efficiently.
- 2. Q: What are the advantages of using a DBMS?**
A: The advantages include data integrity, data security, data sharing, reduced data redundancy, and improved data retrieval and manipulation.
- 3. Q: What is data redundancy in a database?**
A: Data redundancy refers to the duplication of data in a database, which can lead to increased storage requirements and data inconsistency.
- 4. Q: What is a primary key in a database table?**
A: A primary key is a unique identifier for each record in a table, ensuring that each row can be uniquely identified.
- 5. Q: What is normalization in the context of databases?**
A: Normalization is the process of organizing data in a database to minimize data redundancy and dependency.
- 6. Q: What are the different levels of normalization?**
A: The levels of normalization are First Normal Form (1NF), Second Normal Form (2NF), Third Normal Form (3NF), and so on.
- 7. Q: What is a foreign key in a database table?**
A: A foreign key is a column or set of columns in a table that refers to the primary key of another table, creating a relationship between the two tables.
- 8. Q: What is the purpose of an index in a database?**
A: An index is used to speed up data retrieval operations by providing a quick access path to specific data within a table.
- 9. Q: What is a transaction in the context of databases?**
A: A transaction is a sequence of one or more database operations that are executed as a single unit of work, either all succeeding or all failing.
- 10. Q: What is ACID in database transactions?**
A: ACID stands for Atomicity, Consistency, Isolation, and Durability. It ensures that database transactions are reliable and maintain data integrity.
- 11. Q: What are the different types of database models?**
A: The main database models include the hierarchical model, network model, relational model, and object-oriented model.
- 12. Q: What is the difference between a database and a database management system?**
A: A database is a collection of structured data, while a database management system (DBMS) is software that manages and manipulates that data.
- 13. Q: What is the difference between a DBMS and a file system?**
A: A DBMS provides better data management, data integrity, and data security than a file system, which is primarily used for file storage and retrieval.
- 14. Q: What is a join operation in SQL?**
A: A join operation combines rows from two or more tables based on a related column between them.
- 15. Q: Explain the difference between INNER JOIN and OUTER JOIN in SQL.**
A: INNER JOIN returns only the matching rows from both tables, while OUTER JOIN returns matching rows and non-matching rows from one or both tables.
- 16. Q: What is a trigger in a database?**
A: A trigger is a stored procedure that automatically executes when a specific event occurs in the database, such as an update, insert, or delete operation.
- 17. Q: What is a database schema?**
A: A database schema is a blueprint that defines the structure of a database, including

tables, columns, relationships, and constraints.

18. Q: What is data warehousing?

A: Data warehousing is the process of collecting, storing, and organizing data from different sources to support business intelligence and reporting.

19. Q: What is OLAP (Online Analytical Processing)?

A: OLAP is a category of software tools used to analyze and present multi-dimensional data for decision-making and reporting purposes. Software tools that allow users to analyze large volumes of data from multiple perspectives quickly and interactively.

20. Q: What is NoSQL database?

A: NoSQL (Not Only SQL) databases are non-relational databases designed to handle large volumes of unstructured or semi-structured data.

21. Q: What is data mining in the context of databases?

A: Data mining is the process of discovering patterns, relationships, and useful information from large datasets using various techniques and algorithms.

22. Q: What is a database view?

A: A database view is a virtual table derived from one or more base tables in the database, presenting a filtered or summarized subset of data.

23. Q: What is data integrity in a database?

A: Data integrity refers to the accuracy, consistency, and reliability of data stored in a database.

24. Q: What is a transaction log in a database?

A: A transaction log is a file that records all the changes made to the database, ensuring that transactions can be rolled back or recovered in case of failures.

25. Q: What is the purpose of the COMMIT statement in SQL?

A: The COMMIT statement is used to save all the changes made during a transaction permanently into the database.

26. Q: What is the purpose of the ROLLBACK statement in SQL?

A: The ROLLBACK statement is used to undo all the changes made during a transaction and revert the database to its state before the transaction began.

27. Q: What is the difference between a clustered and a non-clustered index?

A: A clustered index determines the physical order of data in a table, whereas a non-clustered index creates a separate structure to store the index.

একটি Clustered Index হলো এমন একটি ইনডেক্স যা টেবিলের ডেটার ফিজিক্যাল অর্ডার নির্ধারণ করে।

অন্যদিকে, Non-Clustered Index হলো একটি আলাদা কাঠামো, যা মূল ডেটার ঠিকানার (location) সাথে ইনডেক্স ভ্যালু যুক্ত করে রাখে।

28. Q: What is a data dictionary in a database?

A: A data dictionary is a repository that stores metadata about the structure and contents of a database.

29. Q: How does a database handle concurrent transactions?

A: A database uses locking mechanisms and isolation levels to ensure that multiple transactions can execute simultaneously without interfering with each other.

30. Q: What is a stored procedure in a database?

A: A stored procedure is a precompiled set of SQL statements stored in the database, which can be executed as a single unit.

31. Q: What is a stored function in a database?

A: A stored function is similar to a stored procedure but returns a value, making it useful for calculations or data manipulations.

32. Q: What is database replication?

A: Database replication is the process of copying and maintaining a database on multiple servers to ensure data redundancy and availability.

33. Q: What is the difference between horizontal and vertical partitioning?

A: Horizontal partitioning divides a table into multiple smaller tables with the same columns, while vertical partitioning separates columns into different tables.

34. Q: What is data normalization and denormalization?

A: Data normalization is the process of reducing data redundancy and improving data integrity, while denormalization involves reintroducing redundancy to improve performance.

35. Q: What are composite keys in a database table?

A: Composite keys are primary keys that consist of multiple columns used together to uniquely identify a row in a table.

36. Q: What is an ER diagram in database design?

A: An Entity-Relationship (ER) diagram is a visual representation of the database's structure, showing entities, attributes, and relationships between them.

37. Q: What are surrogate keys, and why are they used?

A: Surrogate keys are artificially created unique identifiers used as primary keys in a table to avoid reliance on natural keys, which may change or be ambiguous.

38. Q: What is the difference between a heap table and a clustered table?

A: A heap table is an unordered collection of rows, while a clustered table has its rows stored in a specific order based on the clustered index.

39. Q: What is ACID compliance in databases?

A: ACID compliance ensures that database transactions are Atomic, Consistent, Isolated, and Durable, providing reliability and consistency to the data.

40. Q: What is the purpose of the GROUP BY clause in SQL?

A: The GROUP BY clause is used to group rows based on specified columns and allows performing aggregate functions on the grouped data.

41. Q: What is the role of a database administrator (DBA)?

A: A database administrator is responsible for managing and maintaining the database system, including performance tuning, security, and backup and recovery.

42. Q: Explain the differences between SQL and NoSQL databases.

A: SQL databases are relational and use structured query language, while NoSQL databases are non-relational and use various query languages like JSON, BSON, etc.

43. Q: What is the CAP theorem in database systems?

A: The CAP theorem states that a distributed database can have at most two of the following three properties: Consistency, Availability, and Partition Tolerance.

একটি distributed database হলো ডেটাবেইস যার ডেটা বিভিন্ন স্থানে (location), কিন্তু নেটওয়ার্কের মাধ্যমে সংযুক্ত এবং centralized control এর অধীনে থাকে।

44. Q: What is a data warehouse schema, and what are its types?

A: A data warehouse schema determines how data is organized and stored. Types include Star schema, Snowflake schema, and Galaxy schema.

45. Q: What is the purpose of database constraints?

A: Database constraints enforce rules and restrictions on data to maintain data integrity

and consistency.

46. Q: What is an ETL process in data warehousing?

A: ETL (Extract, Transform, Load) is a process that extracts data from various sources, transforms it to match the data warehouse schema, and loads it into the data warehouse.

47. Q: How does indexing impact database performance? A: Indexing improves data retrieval speed but may slow down data insertion and updates due to the overhead of maintaining the index.

48. Q: What is a data mart in data warehousing?

A: A data mart is a subset of a data warehouse, focusing on specific business needs or departments.

49. Q: What is the difference between a clustered and non-clustered index?

A: A clustered index determines the physical order of data in a table, while a non-clustered index creates a separate structure for fast data retrieval.

50. Q: What is the purpose of a database trigger? A: Database triggers are used to automatically execute predefined actions in response to specific events like data modification.

51. Q: How does a full-text search work in a database?

A: A full-text search allows users to search for words or phrases within text columns of a database efficiently.

52. Q: What is the purpose of the SQL SELECT statement?

A: The SQL SELECT statement retrieves data from one or more tables in a database.

53. Q: What is a self-join in SQL?

A: A self-join is a query where a table is joined with itself to combine related rows.

54. Q: What is the purpose of the HAVING clause in SQL?

A: The HAVING clause filters the results of aggregate functions applied to grouped data in SQL.

Aggregate functions এমন SQL ফাংশন যেগুলো একাধিক রেকর্ড (row) এর উপর কাজ করে এবং একটি মাত্র মান (value) রিটোর্ন করে।

55. Q: What are the advantages and disadvantages of using database views?

A: Advantages include data abstraction and security, while disadvantages involve performance overhead and complexity.

56. Q: What is database sharding?

A: Database sharding is a technique of partitioning a large database into smaller, more manageable pieces across multiple servers.

57. Q: What is a materialized view in a database?

A: A materialized view is a precomputed snapshot of a query result, stored for faster data retrieval.

58. Q: How can you improve the performance of database queries?

A: Performance can be improved by using indexes, optimizing queries, caching, and denormalization.

59. Q: What is a database schema migration?

A: Database schema migration is the process of updating a database's structure while preserving existing data.

ডেটাবেইস স্কিমা মাইগ্রেশন হলো এমন একটি প্রক্রিয়া, যার মাধ্যমে ডেটাবেইসের স্ট্রাকচার বা গঠন পরিবর্তন করা হয় —যেমন: নতুন কলাম যোগ করা, টেবিলের নাম পরিবর্তন করা,

ইনডেক্স তৈরি করা ইত্যাদি—তবে পুরনো ডেটা না মুছে বা নষ্ট না করে।

60. Q: What is the role of a data analyst in database management?

A: A data analyst analyzes data stored in databases to extract valuable insights for decision-making.

61. Q: What is a deadlock in a database, and how can it be resolved?

A: A deadlock occurs when two or more transactions are unable to proceed because they are each waiting for a resource that is locked by another transaction. Deadlocks can be resolved by using techniques such as deadlock detection and timeout mechanisms.

62. Q: What is the purpose of the CASCADE option in foreign keys? A: The CASCADE option ensures that when a record in the referenced table is updated or deleted, all related records in the referencing table are automatically updated or deleted.

63. Q: What is a database index, and how does it work?

A: A database index is a data structure that improves the speed of data retrieval operations by providing quick access to specific data within a table. It works similar to an index in a book, allowing faster lookup of specific data.

64. Q: What is a natural key in the context of databases?

A: A natural key is a column or set of columns in a table that already exists in the real world and can be used as a unique identifier for the records.

ধরো একটি students টেবিল আছে। যদি প্রতিটি শিক্ষার্থীর "Roll Number" অনন্য (unique) হয়, তাহলে সেই Roll Number-ই হতে পারে একটি Natural Key —কারণ এটি বাস্তব জীবনে আগে থেকেই ব্যবহৃত হচ্ছে।

65. Q: What is the purpose of the CHECK constraint in a database?

A: The CHECK constraint restricts the values that can be inserted or updated in a column based on a specified condition.

66. Q: What are ACID properties, and why are they essential for a database?

A: ACID properties (Atomicity, Consistency, Isolation, Durability) ensure that database transactions are reliable, maintain data integrity, and preserve consistency.

67. Q: What is a self-referential table in a database?

A: A self-referential table is a table in which a foreign key references the primary key of the same table, establishing a relationship within the table itself.

68. Q: What is a database index fragmentation, and how can it be resolved?

A: Index fragmentation occurs when data pages are out of order, leading to reduced query performance. It can be resolved by rebuilding or reorganizing indexes.

ইনডেক্স ফ্র্যাগমেন্টেশন ঘটে যখন ডেটাবেইসের ইনডেক্স পেজগুলো এলামেলো বা অগোছালো হয়ে যায়, ফলে কুয়েরি চালাতে সময় বেশি লাগে এবং পারফরম্যান্স কমে যায়।

69. Q: What is the difference between a left join and a right join in SQL?

70. A: A left join returns all rows from the left table and matching rows from the right table, while a right join returns all rows from the right table and matching rows from the left table.

71. Q: How can you ensure data security in a database system?

A: Data security can be ensured by implementing access controls, encryption, strong passwords, and regularly auditing and monitoring the database.

72. Q: What is database replication latency, and how can it affect data consistency?

A: Database replication latency refers to the delay in replicating data to secondary

servers. High latency can result in data inconsistencies between the primary and secondary servers.

73. Q: What is a foreign key constraint in a database?

A: A foreign key constraint ensures that the values in a column of one table correspond to the values of a primary key in another table, establishing a relationship between the two tables.

74. Q: What are the different types of database backups, and when should they be used?

A: Types of database backups include full backups (complete backup of the entire database), differential backups (backup of changes since the last full backup), and incremental backups (backup of changes since the last backup, whether full or incremental). Each type has specific use cases and recovery implications.

75. Q: What is the purpose of the DML (Data Manipulation Language) in SQL?

A: The DML is used to interact with the data in the database, allowing users to insert, update, delete, and retrieve data from tables.

76. Q: What is a surrogate data key, and how is it different from a natural data key?

A: A surrogate data key is an artificial unique identifier, usually an auto-incremented number, used as a primary key. In contrast, a natural data key is based on real-world data and may not always be unique or stable.

Surrogate data key হলো একটি কৃত্রিম (artificial) ইউনিক আইডেন্টিফায়ার, যা সাধারণত auto-incremented number বা সিস্টেম-জেনারেটেড আইডি হিসেবে ব্যবহৃত হয়। এটি টেবিলের প্রাইমারি কি (Primary Key) হিসেবে ব্যবহার করা হয়।

অন্যদিকে, Natural data key হলো এমন একটি কী যা বাস্তব জীবনের তথ্যের উপর ভিত্তি করে তৈরি, যেমন: জাতীয় পরিচয় নম্বর (NID), ইমেইল, পাসপোর্ট নম্বর ইত্যাদি।

77. Q: What is a B-tree index in a database?

A: A B-tree index is a balanced tree structure used to organize data for efficient data retrieval in a database.

78. Q: What is a data dictionary, and why is it important in database management?

A: A data dictionary is a repository that stores metadata about the database's structure and contents. It is crucial for data governance, documentation, and maintaining data consistency.

79. Q: What is a multi-version concurrency control (MVCC) in databases?

A: MVCC is a technique used to manage concurrent access to data by allowing multiple versions of the same data to coexist during transactions, improving database performance and concurrency.

80. Q: How does a database handle data recovery after a system crash?

A: Databases use transaction logs and other mechanisms to recover data to a consistent state after a system crash or failure.

81. Q: What is a database connection pool, and why is it used?

A: A database connection pool is a cache of established database connections that can be reused, reducing the overhead of establishing new connections and improving performance. **ডেটাবেইস কানেকশন পুল** হলো এমন একটি ক্যাশ বা সংরক্ষণাগার, যেখানে

আগে থেকেই তৈরি করা কিছু ডেটাবেইস কানেকশন রাখা থাকে, যাতে প্রতিবার নতুন করে কানেকশন তৈরির দরকার না হয়।

82. Q: What is data compression in a database, and how does it benefit storage?

A: Data compression is the process of reducing the size of data to save storage space, leading to lower storage costs and improved data retrieval speed.

83. Q: What is the difference between a clustered and non-clustered index in terms of data sorting?

A: A clustered index determines the physical order of data in a table, whereas a non-clustered index creates a separate structure to map the index to the data.

84. Q: What are materialized views, and how do they differ from regular views?

A: Materialized views are precomputed result sets stored in the database, while regular views are virtual tables derived from base tables. Materialized views offer faster query performance but require periodic updates.

Regular View (সাধারণভিটি):

- এটি একটি virtual table বা কল্পিত টেবিল।
- মূল টেবিল থেকে ডেটা নিয়ে প্রতিবার কুয়েরি চালানোর সময় ডেটা হিসেব করে দেখায়।
- ভিউতে নিজস্ব করে কোনো ডেটা সংরক্ষিত থাকে না।

Materialized View:

- এটি একটি precomputed result set — অর্থাৎ আগেই হিসেব করে রাখা হয়।
- ডেটাবেইসে সত্যিকারের টেবিলের মতো সংরক্ষিত থাকে।
- কুয়েরি চালালে খুব দ্রুত রেজাল্ট পাওয়া যায় কারণ আগেই হিসেব করা থাকে।
- তবে ডেটা আপডেট হলে এটিকে রিফ্রেশ বা আপডেট করতে হয়, না হলে পুরোনো ডেটা দেখাতে পারে।

85. Q: What is the role of a data architect in database management?

A: A data architect designs and plans the structure of databases, ensuring data integrity, scalability, and performance to meet business needs.

86. Q: What is the purpose of the ORDER BY clause in SQL?

A: The ORDER BY clause is used to sort the result of a SQL query based on specified columns, in either ascending or descending order.

87. Q: What is a multi-table join in SQL?

A: A multi-table join involves combining data from multiple tables using various join operations such as INNER JOIN, LEFT JOIN, RIGHT JOIN, and FULL OUTER JOIN.

88. Q: How does a database implement data replication across multiple servers?

A: Databases use replication techniques like master-slave and master-master replication to copy data across multiple servers in real-time or near real-time.

89. Q: What is an entity in the context of the Entity-Relationship (ER) model?

A: An entity represents a real-world object or concept that is stored as a table in a database, having attributes (columns) that describe its properties.

90. Q: What is the purpose of the COMMIT and ROLLBACK statements in transactions?

A: The COMMIT statement is used to save the changes made during a transaction, while the ROLLBACK statement is used to undo the changes and restore the database to its

state before the transaction.

91. Q: What is a data lake, and how does it differ from a data warehouse?

A: A data lake is a storage repository that holds a vast amount of raw and unstructured data, while a data warehouse is designed to store structured and processed data for business intelligence and reporting purposes.

92. Q: What are NoSQL database categories, and what types of data do they handle?

A: NoSQL databases are categorized into key-value stores, document stores, column-family stores, and graph databases, each designed to handle specific types of data and use cases.

93. Q: What is data governance in the context of database management?

A: Data governance is a set of policies, procedures, and controls that ensure data quality, security, and compliance within an organization.

94. Q: What is the purpose of the TRUNCATE statement in SQL?

A: The TRUNCATE statement is used to delete all records from a table quickly, but it cannot be rolled back like a DELETE statement.

95. Q: What is the role of an index in query optimization?

A: An index improves query performance by reducing the number of data pages that need to be scanned during data retrieval.

96. Q: What is the purpose of the UNIQUE constraint in a database?

A: The UNIQUE constraint ensures that the values in a column or a combination of columns are unique across all rows in the table.

97. Q: What is the difference between a database administrator (DBA) and a database developer?

A: A DBA is responsible for managing the database system, while a database developer is involved in designing and implementing database applications and queries.

98. Q: How does a database handle concurrency control to prevent data inconsistencies?

A: Databases use locking mechanisms, isolation levels, and transaction management to handle concurrent access and maintain data consistency.

99. Q: What is the role of a distributed database in managing data across multiple locations?

A: A distributed database allows data to be stored and accessed across multiple physical locations, enabling better data availability and disaster recovery.

100. Q: What is a NoSQL graph database, and how is it useful?

A: A NoSQL graph database uses graph structures to represent and store data, making it ideal for managing complex relationships and interconnected data.

101. Q: What is database replication lag, and how can it impact data consistency?

A: Database replication lag refers to the delay in replicating data from the primary to the secondary server. High replication lag can lead to data inconsistencies between the servers.

102. Q: What is the purpose of the SQL UPDATE statement, and how is it used?

A: The SQL UPDATE statement is used to modify existing records in a database table, allowing you to change the values of specific columns based on specified conditions.

103. Q: What is a NoSQL document store database, and when is it suitable for use?

A: A NoSQL document store database stores data as documents, typically in JSON or BSON format. It is suitable for storing and querying semi-structured or unstructured data with varying attributes.

NoSQL Document Store Database এমন এক ধরনের ডেটাবেইস যেখানে ডেটা ডকুমেন্ট আকারে সংরক্ষণ করা হয়, সাধারণত JSON বা BSON ফর্ম্যাটে। এটি বিশেষভাবে উপযুক্ত যখন ডেটার স্ট্রাকচার সব সময় একই না থাকে — অর্থাৎ ডেটা semi-structured বা unstructured হয়।

104. Q: What is database partitioning, and how does it improve performance?

A: Database partitioning involves dividing a large table into smaller, more manageable partitions to enhance query performance, data maintenance, and storage management.

105. Q: How does the SQL UNION operator work, and when should you use it?

A: The SQL UNION operator combines the result sets of two or more SELECT queries into a single result set. Use UNION when you want to merge data from multiple queries with the same structure.

106. Q: What is a database trigger, and what are its different types?

A: A database trigger is a special type of stored procedure that automatically executes in response to specific database events such as INSERT, UPDATE, or DELETE. Types include BEFORE and AFTER triggers.

107. Q: What is the purpose of a database connection string?

A: A database connection string contains the information required to establish a connection to a database, including the server address, username, password, and database name.

108. Q: What is data migration, and what are the challenges involved?

A: Data migration is the process of transferring data from one database system to another. Challenges include data mapping, data cleansing, and ensuring data integrity during the migration.

109. Q: What is a NoSQL column-family store database, and how is it different from other NoSQL databases?

A: A NoSQL column-family store database stores data in column families, where each row can have varying numbers of columns. It is suitable for large-scale data with varying attributes.

110. Q: How does database replication contribute to high availability and fault tolerance?

A: Database replication creates copies of data on multiple servers, allowing for automatic failover and ensuring data availability even if one server experiences a failure.

111. Q: What is database caching, and how does it improve performance?

A: Database caching stores frequently accessed data in memory, reducing the need to fetch data from disk, thereby improving query response times.

112. Q: What is database denormalization, and when should it be used?

A: Database denormalization involves introducing redundancy into a database to improve performance by reducing the need for complex joins. It should be used cautiously to maintain data consistency.

- 113. Q: How does the SQL INSERT statement work, and what precautions should be taken while inserting data?**

A: The SQL INSERT statement adds new records to a table. Precautions include validating data before insertion, handling primary key constraints, and using parameterized queries to prevent SQL injection.

- 114. Q: What are database constraints, and how do they ensure data integrity?**

A: Database constraints impose rules on the data stored in a table to prevent invalid or inconsistent data from being inserted or updated, thus maintaining data integrity.

- 115. Q: What is a database index seek, and how does it differ from a database index scan?**

A: A database index seek uses the index to find specific data efficiently, while an index scan reads through the entire index to locate the required data.

Index Seek is like finding a word using a dictionary's index — **fast and direct**, while **Index Scan** is like reading every word in a book — **slow and exhaustive**.

- 116. Q: What is the purpose of the SQL DELETE statement, and what are the implications of using it?**

A: The SQL DELETE statement removes rows from a database table. Using it without a WHERE clause can lead to accidental deletion of all records in the table.

- 117. Q: How do database transactions help maintain data consistency?**

A: Database transactions ensure that a series of database operations are treated as a single unit of work, either all succeeding or all failing, to maintain data consistency.

- 118. Q: What is a NoSQL key-value store database, and when is it appropriate to use?**

A: A NoSQL key-value store database stores data as key-value pairs and is suitable for high-speed data access, caching, and storing unstructured or semi-structured data.

- 119. Q: How does a database handle data indexing for large datasets?**

A: Databases use various data structures like B-trees and hash tables for indexing, which efficiently organize and store index data for quick data retrieval.

- 120. Q: What is a NoSQL graph database, and when is it beneficial?**

A: A NoSQL graph database is designed to store and process data with complex relationships and interconnectedness. It is beneficial for applications involving social networks, recommendation systems, and network analysis.

- 121. Q: What is a database query plan, and how does it impact query performance?**

A: A database query plan is an execution plan created by the database optimizer to retrieve data efficiently. A well-optimized query plan can significantly improve query performance.

- 122. Q: What is the purpose of the SQL MERGE statement, and how is it used?**

A: The SQL MERGE statement is used to perform an "upsert" Update + Insert = Upsert operation, which combines the functionality of both INSERT and UPDATE statements based on certain conditions.

- 123. Q: What are the different types of NoSQL data models, and how do they store data?** A: The main types of NoSQL data models are document stores, key-value stores, column-family stores, and graph databases, each designed to store and retrieve data in unique ways.

- 124. Q: What is database materialization, and when is it beneficial?**

A: Database materialization involves creating and storing the results of complex queries or views as actual tables. It is beneficial for frequently used queries to reduce computation time.

125. Q: What is data profiling in database management?

A: Data profiling is the process of analyzing and evaluating the quality of data in a database, identifying patterns, anomalies, and potential data issues.

126. Q: How does the SQL WHERE clause work, and what are its common operators?

A: The SQL WHERE clause filters data based on specified conditions, and common operators include "=", "<>", "<", ">", "<=", ">=", "LIKE," and "IN."

127. Q: What is data modeling, and why is it essential for designing a database?

A: Data modeling is the process of defining the structure and relationships of data in a database, which is essential for accurately representing business requirements and ensuring data integrity.

128. Q: What is a NoSQL key-range store database, and when is it suitable for use?

A: A NoSQL key-range store database stores data as ordered key-value pairs, allowing efficient range queries over keys. It is suitable for applications involving time-series data and range-based queries.

129. Q: How does horizontal partitioning differ from vertical partitioning in a database?

A: Horizontal partitioning divides a table's rows into smaller subsets, while vertical partitioning separates columns into different physical storage locations.

130. Q: What is database replication conflict resolution, and how is it handled?

A: Database replication conflict resolution involves resolving conflicts that may arise when the same data is updated concurrently on multiple servers. Techniques like timestamps or conflict detection algorithms are used to resolve conflicts.

131. Q: What is the purpose of a database trigger in the context of auditing?

A: Database triggers can be used for auditing changes to the database, allowing you to track modifications made to specific tables for security and compliance purposes.

132. Q: How does database clustering enhance performance and reliability?

A: Database clustering involves grouping multiple database servers together to distribute data and processing tasks, which improves performance and provides fault tolerance.

133. Q: What is a NoSQL time-series database, and when is it used?

A: A NoSQL time-series database is designed to efficiently store and analyze timestamped data, making it suitable for applications like IoT sensor data, financial data, and logs.

134. Q: What is the purpose of the SQL SELECT INTO statement?

A: The SQL SELECT INTO statement creates a new table and inserts the result set of a SELECT query into that new table.

135. Q: What is data warehousing schema design, and what are the commonly used schemas?

A: Data warehousing schema design involves organizing data for efficient querying and reporting. Commonly used schemas include star schema, snowflake schema, and galaxy schema.

136. Q: What is a NoSQL columnar database, and when is it appropriate to use?

A: A NoSQL columnar database stores data in columns rather than rows, making it suitable for analytics and scenarios where data is read and aggregated in column-based operations.

137. **Q: How does the SQL LIKE operator work, and when is it used in queries?** A: The SQL LIKE operator is used for pattern matching in queries, allowing you to search for data using partial or wildcard values.

138. **Q: What is the purpose of a database transaction log, and how is it used in recovery?**

A: The database transaction log records all changes to the database, providing a point-in-time recovery mechanism to restore the database to a consistent state after a failure.

139. **Q: What is NoSQL database sharding, and how does it help scale databases?**

A: NoSQL database sharding involves partitioning data across multiple servers to distribute the load and enable horizontal scaling for large datasets and high-performance applications.

NoSQL ডেটাবেইসে Sharding মানে হলো:

- বড় ডেটাসেটকে ছোট ছোট অংশে ভাগ করে
- আলাদা বিভিন্ন সার্ভারে সংরক্ষণ করা,
- যাতে লোড কমে, এবং
- অনেক ইউজার বা ট্রান্স্যাকশন একসাথে দ্রুত সেবা পেতে পারে।

140. **Q: What are the best practices for securing a database system?**

A: Best practices include using strong passwords, limiting user privileges, encrypting sensitive data, keeping the database software up-to-date, and implementing regular backups.

141. **Q: What is the role of a data warehouse architect in database management?**

A: A data warehouse architect designs and implements data warehouses to support business intelligence and analytics requirements, ensuring data accuracy and performance.

142. **Q: What is a NoSQL wide-column store database, and when is it appropriate to use?**

A: A NoSQL wide-column store database stores data in column families, where each row can have a varying number of columns. It is suitable for handling large amounts of sparsely populated data.

143. **Q: What is database query optimization, and how can it be achieved?**

A: Database query optimization involves improving the performance of queries through various techniques such as creating indexes, rewriting queries, and using caching.

144. **Q: What is a NoSQL distributed database, and what challenges does it address?**

A: A NoSQL distributed database spreads data across multiple nodes, addressing challenges like scalability, fault tolerance, and high availability in large-scale applications.

145. **Q: What is database encryption, and why is it important for data security?**

A: Database encryption is the process of converting sensitive data into ciphertext to protect it from unauthorized access. It ensures data confidentiality even if the database is compromised.

146. **Q: How does the SQL GROUP BY clause work, and when is it used?**

A: The SQL GROUP BY clause groups rows based on specified columns and is typically used with aggregate functions like SUM, AVG, COUNT, etc.

147. Q: What is a NoSQL document store database, and when is it not recommended?

A: A NoSQL document store database is suitable for flexible and unstructured data, but it may not be recommended for applications requiring complex querying or analytics.

148. Q: What is database data replication consistency, and how is it maintained?

A: Database data replication consistency ensures that data is the same across all replicated servers by using techniques like two-phase commit or eventual consistency.

149. Q: What is a database view materialization, and how does it impact performance?

A: Database view materialization involves creating a physical table from a view's definition. It can improve performance for complex views but may lead to increased storage requirements.

150. Q: What is the purpose of the SQL TRIGGER statement, and how is it used?

A: The SQL TRIGGER statement is used to define a trigger—a special type of stored procedure that is automatically executed in response to specified database events.

151. Q: What is NoSQL tunable consistency, and how does it address performance and data availability trade-offs?

A: NoSQL tunable consistency allows applications to configure the level of consistency they need based on performance and data availability requirements.

অ্যাপ্লিকেশন নিজের মতো করে ডেটার কনসিস্টেন্সি (Consistency) এর মাত্রা নির্ধারণ করতে পারে, যাতে পারফর্মেন্স এবং অ্যাভেইলেবিলিটি (Availability) মধ্যে সঠিক ব্যালেন্স রাখা যায়।

152. Q: What is a database connection pool, and how does it enhance performance?

A: A database connection pool is a cache of established database connections that can be reused to reduce the overhead of creating new connections and improve performance.

153. Q: What is a NoSQL event store database, and when is it used?

A: A NoSQL event store database is designed to store and manage event data, making it suitable for applications requiring event sourcing and event-driven architectures.

এটি এমন ডেটাবেইস যেখানে ঘটনাগুলোর (events) তালিকা সংরক্ষণ করা হয় — প্রতিটি পরিবর্তন বা অ্যাকশন (যেমন: অর্ডার করা, পেমেন্ট দেওয়া, লগইন ইত্যাদি) একটি ইভেন্ট হিসেবে রেকর্ড হয়।

154. Q: What is the purpose of the SQL JOIN operation, and what are its different types? A: The SQL JOIN operation combines rows from multiple tables based on related columns. Different types include INNER JOIN, LEFT JOIN, RIGHT JOIN, and FULL OUTER JOIN.

155. Q: What is database auditing, and why is it essential for data governance?

A: Database auditing involves tracking and recording changes to data and database activities for security, compliance, and data governance purposes.

156. Q: What is NoSQL eventual consistency, and how is it different from strong consistency?

A: NoSQL eventual consistency allows data changes to propagate across nodes over time, while strong consistency ensures that data is immediately consistent across all nodes.

যখন ডেটা পরিবর্তন করা হয়, তখন তা সাথে সব সার্ভারে আপডেট না-ও হতে পারে।

কিন্তু সময়ের সঙ্গে সঙ্গে সব নোডেই এক রকম ডেটা দেখা যাবে।

উদাহরণ:

তুমি একটি পোস্ট এডিট করলে হয়তো একসাথে সব জায়গায় সেই পরিবর্তন দেখা যাবে না, কিন্তু কিছু সময় পর সব সার্ভারে সেটি আপডেট হয়ে যাবে।

Strong Consistency (স্ট্রং কনসিস্টেন্সি):

- ডেটা আপডেট হওয়ার সঙ্গে সঙ্গে সব নোডেই তা একইরকম দেখা যায়।



তুমি যদি একটি ব্যাংক অ্যাপ ব্যবহার করো, এবং টাকা পাঠাও, তবে সেই তথ্য সাথে সাথেই সব সার্ভারে একরকম হওয়া দরকার — এটা স্ট্রং কনসিস্টেন্সির উদাহরণ।

উদাহরণ:

157. Q: What is database resource optimization, and how is it achieved?

A: Database resource optimization involves efficiently using hardware resources like CPU, memory, and disk to ensure optimal database performance and responsiveness.

158. Q: What is a NoSQL in-memory database, and when is it suitable for use?

A: A NoSQL in-memory database stores data primarily in RAM, offering extremely fast data access and is suitable for high-performance applications that require low-latency access.

159. Q: What is database change data capture, and how is it used?

A: Database change data capture tracks and captures data changes in real-time, enabling applications to react to data updates and propagate changes to other systems.

Change Data Capture (CDC) হলো এমন একটি প্রযুক্তি বা পদ্ধতি, যা রিয়েল-টাইমে ডেটাবেইসে কী পরিবর্তন (Insert, Update, Delete) ঘটেছে তা শনাক্ত ও ধরতে পারে, এবং সেই পরিবর্তনগুলো অন্য সিস্টেমে বা অ্যাপ্লিকেশনে প্রসারিত বা শেয়ার করতে পারে।

160. Q: What is a database backup strategy, and what are the common backup methods?

A: A database backup strategy outlines the plan for creating and storing database backups. Common methods include full backups, differential backups, and incremental backups.

161. Q: What is NoSQL tunable consistency, and how does it address performance and data availability trade-offs?

A: NoSQL tunable consistency allows applications to configure the level of consistency they need based on performance and data availability requirements.

162. Q: What is a NoSQL time-to-live (TTL) feature, and how is it used?

A: A NoSQL time-to-live (TTL) feature allows data to automatically expire after a specified period, which is useful for managing data retention and data cleanup in certain scenarios.

- One-time passwords (OTP)**
- সিস্টেমে অপ্রয়োজনীয় পুরনো ডেটা মুছে ফেলা**

- Storage অপটিমাইজেশন ও পরিষ্কার রাখা

163. Q: What is database data archiving, and why is it important for long-term data storage?

A: Database data archiving involves moving older or less frequently accessed data to secondary storage to free up space in the main database while still retaining access to historical records.

ডেটাবেইস ডেটা আর্কাইভিং মানে হলো —পুরনো বা কম ব্যবহৃত ডেটা প্রধান (main) ডেটাবেইস থেকে সরিয়ে দ্বিতীয় স্তরের স্টোরেজে (secondary storage) সংরক্ষণ করা, যাতে মূল ডেটাবেইস হালকা ও দ্রুতগতির থাকে, তবুও প্রযোজন হলে সেই পুরনো ডেটা পরবর্তীতে দেখা বা বিশ্লেষণ করা যায়।

164. Q: What is NoSQL partitioning, and how does it contribute to scalability?

A: NoSQL partitioning involves splitting data across multiple nodes, which allows the database to handle large datasets and handle read and write operations in parallel, contributing to scalability.

165. Q: What is database data masking, and how is it used to protect sensitive data?

A: Database data masking is a technique that replaces sensitive data with fictional or anonymized data to protect sensitive information from unauthorized access.

Database data masking হলো একটি কৌশল, যার মাধ্যমে সংবেদনশীল ডেটাকে (যেমন: নাম, ফোন নম্বর, কার্ড নম্বর) ক্রিমি বা গোপন ডেটা দিয়ে প্রতিস্থাপন (replace) করা হয়,

যাতে সেই তথ্য অনুমোদনহীন ব্যবহারকারীর কাছে প্রকাশ না পায়।

166. Q: What is the purpose of the SQL SET statement, and how is it used?

A: The SQL SET statement is used to assign values to variables that can be used in SQL queries, making queries more dynamic and flexible.

167. Q: What is a NoSQL data consistency model, and what are the different consistency levels?

A: A NoSQL data consistency model defines the guarantees provided by the database in terms of data consistency across replicas. Different levels include eventual consistency, strong consistency, and more.

NoSQL ডেটা কনসিস্টেন্সি মডেল এমন একটি ধারণা যা বলে দেয়, একটি ডেটা আপডেট হওয়ার পর বিভিন্ন সার্ভারে (রেপ্লিকা) সেই আপডেট কর্তৃতা দ্রুত বা নির্ভুলভাবে প্রতিফলিত হবে — অর্থাৎ, ডেটার মান সব জায়গায় একই থাকবে কিনা, তা নির্ভর করে কোন কনসিস্টেন্সি লেভেল ব্যবহৃত হচ্ছে তার উপর।

1. Strong Consistency (স্ট্রং কনসিস্টেন্সি)

- ডেটা আপডেট হওয়ার পর সব রেপ্লিকায় সাথে সাথে একই রকম দেখা যায়।
- ব্যবহারকারী সব সময় সঠিক এবং সর্বশেষ ডেটা পায়।
- ব্যবহৃত হয় **ব্যাংকিং, ট্রানজ্যাকশনাল সিস্টেমে।**

2. Eventual Consistency (ইভেন্যুটুয়াল কনসিস্টেন্সি)

- ডেটা আপডেট হয় ধীরে ধীরে — শুরুতে সব রেপ্লিকায় একসাথে না-ও দেখা যেতে পারে।
- কিন্তু কিছু সময় পরে সব জায়গায় একই ডেটা দেখা যাবে।
- ব্যবহৃত হয় **সোশ্যাল মিডিয়া, ক্যাশ সিস্টেমে।**

168. Q: What is database connection pooling, and how does it benefit performance and resource utilization?

A: Database connection pooling involves **reusing existing database connections** instead of creating new ones, which reduces connection overhead and enhances performance and resource efficiency.

169. Q: What is NoSQL read and write quorum, and how are they used in distributed databases?

A: NoSQL read and write quorums determine the number of replicas that must acknowledge read or write operations for data consistency and availability in distributed databases.

NoSQL-এর Read & Write Quorum হলো এমন একটি কৌশল যা নির্ধারণ করে—
একটি ডেটা লেখার (write) বা পড়ার (read) সময় কতটি রেপ্লিকা (replica server)
সেই অপারেশন গ্রহণ বা নিশ্চিত করতে হবে যাতে ডেটা সঠিক (consistent) ও উপলব্ধ (available) থাকে।

❖ কিভাবে কাজ করে?

Distributed Database-এ প্রতিটি ডেটা বিভিন্ন সার্ভারে কপি (replica) হয়ে থাকে।

তিনটি গুরুত্বপূর্ণ ভ্যালু থাকে:

- **N** = মোট রেপ্লিকার সংখ্যা (যতগুলো জায়গায় ডেটা আছে)
- **W** = Write quorum (লেখা কনফার্ম করার জন্য কমপক্ষে কতটি রেপ্লিকা লাগবে)
- **R** = Read quorum (পড়ার সময় কমপক্ষে কতটি রেপ্লিকা থেকে ডেটা আসবে)

নিয়ম: যদি $R + W > N$ হয়, তবে Strong Consistency নিশ্চিত করা সম্ভব হয়।

170. Q: What is database high availability, and how is it achieved in a database system?

A: Database high availability refers to the ability of a database system to **remain operational and accessible** even in the face of hardware failures, software issues, or network disruptions. It is achieved through **redundancy, failover mechanisms, and replication**.

Database High Availability (HA) বলতে বোঝায়, একটি ডেটাবেইস সিস্টেম এমনভাবে তৈরি করা, যাতে হার্ডওয়্যার ফেইল, সফটওয়্যার সমস্যা বা নেটওয়ার্ক বিভ্রাট হলেও সিস্টেম চলতে থাকে ও ব্যবহারকারীরা অ্যাক্সেস করতে পারে —অর্থাৎ ডাউনটাইম প্রায় শূন্যের কাছাকাছি থাকে।

171. Q: What is the purpose of the SQL DELETE CASCADE option, and when is it useful?

A: The SQL DELETE CASCADE option automatically deletes related records in child tables when a record in the parent table is deleted. It is useful to maintain referential integrity and handle cascading deletes.

172. Q: What is NoSQL ACID compliance, and how is it different from ACID in SQL databases?

A: NoSQL ACID compliance refers to the support for Atomicity, Consistency, Isolation, and Durability in NoSQL databases. However, the implementation of ACID properties in NoSQL databases may vary from traditional SQL databases.

NoSQL ACID Compliance মানে হলো—Atomicity, Consistency, Isolation, এবং Durability এই চারটি বৈশিষ্ট্য কোনো NoSQL ডেটাবেইসে কর্তৃত সমর্থিত বা বাস্তবায়িত হয়। তবে, SQL ডেটাবেইসের মতো করে NoSQL সবসময় পুরোপুরি ACID অনুসরণ করে না, কারণ NoSQL সিস্টেমে স্লেবিলিটি ও পারফরমেন্স-কে প্রাধান্য দেওয়া হয়।

173. Q: What is database partitioning key selection, and how is it determined?

A: Database partitioning key selection involves choosing the column or attribute used to determine how data is distributed across partitions. It is typically based on the access patterns and the nature of the data.

Database partitioning key selection মানে হলো—একটি সঠিক কলাম বা অ্যাট্ৰিবিউট (attribute) নির্বাচন করা, যার উপর ভিত্তি করে ডেটাবেইসের তথ্যগুলো বিভিন্ন পার্টিশনে ভাগ করা হবে। এই key বা কলামটি ডেটার অ্যাক্সেস প্যাটার্ন এবং ডেটার প্রাকৃতি (nature) অনুযায়ী বেছে নেওয়া হয়।

174. Q: What is NoSQL eventual consistency, and how is it implemented in distributed databases?

A: NoSQL eventual consistency allows data to be eventually consistent across replicas after a period of time. It is achieved through asynchronous replication and conflict resolution mechanisms.

175. Q: What is database hot standby, and how does it contribute to database availability?

A: Database hot standby involves maintaining a synchronized replica of the primary database that can take over in case of a failure, ensuring minimal downtime and high availability.

Database Hot Standby হলো একটি কৌশল, যেখানে প্রধান (primary) ডেটাবেইসের একটি সিঙ্ক্রেনাইজড কপি (replica) রিয়েল-টাইমে বা খুব দ্রুত সময়ে আপডেট হয়ে থাকে, যাতে মূল ডেটাবেইসে কোনো সমস্যা হলে এই কপিটি তৎক্ষণাত্মক দায়িত্ব নিতে পারে। এর ফলে সিস্টেম বন্ধ হওয়ার সময় (downtime) খুব কম হয় এবং High Availability বজায় থাকে।

176. Q: What is the purpose of the SQL ALTER TABLE statement, and how is it used?

A: The SQL ALTER TABLE statement is used to modify the structure of an existing table, such as adding or dropping columns, changing data types, or adding constraints.

177. Q: What is NoSQL consistency tuning, and how does it impact performance and data integrity?

A: NoSQL consistency tuning allows applications to configure the level of data consistency needed based on performance requirements, balancing consistency with potential data staleness.

178. Q: What is database data deduplication, and why is it beneficial for storage optimization?

A: Database data deduplication is the process of identifying and eliminating duplicate data in a database, which reduces storage requirements and improves data efficiency.

179. Q: What is NoSQL sharding key selection, and how is it determined?

A: NoSQL sharding key selection involves choosing the attribute or column used to determine how data is partitioned across shards in a distributed database. It should be selected based on even data distribution and access patterns.

NoSQL Sharding Key Selection বলতে বোঝায়—একটি উপযুক্ত অ্যাট্ৰিবিউট বা কলাম

নির্বাচন. যার ভিত্তিতে ডেটা বিভিন্ন shard (বিভাগ)-এ ভাগ করা হবে distributed NoSQL ডেটাবেইসে।

180. Q: What is database horizontal scaling, and how is it achieved in distributed databases?

A: Database horizontal scaling involves adding more servers or nodes to a distributed database system to handle increased data volume and workload. It can be achieved through partitioning and replication.

181. Q: What is the purpose of the SQL CASE statement, and how is it used?

A: The SQL CASE statement allows conditional logic in SQL queries, enabling you to perform different actions based on specified conditions.

182. Q: What is NoSQL query optimization, and how does it differ from SQL query optimization?

A: NoSQL query optimization involves optimizing queries for distributed databases, where data may be spread across multiple nodes. It differs from SQL query optimization, which is focused on optimizing queries for a single, centralized database.

NoSQL Query Optimization হলো—ডিস্ট্রিবিউটেড (বিভক্ত) ডেটাবেইসে লেখা query-গুলোর গতি ও কার্যকারিতা উন্নত করার প্রক্রিয়া, যেখানে ডেটা একাধিক সার্ভারে (node) ছড়িয়ে থাকে।

183. Q: What is database table partitioning, and how does it improve data retrieval performance?

A: Database table partitioning involves dividing a large table into smaller, more manageable partitions. It improves data retrieval performance by reducing the amount of data that needs to be scanned for specific queries.

184. Q: What is the CAP theorem, and how does it impact distributed database design?

A: The CAP theorem states that a distributed database system cannot simultaneously guarantee all three properties of Consistency, Availability, and Partition Tolerance. Database designers must choose a trade-off between these properties based on their specific requirements.

185. Q: What is database data compression, and how is it applied to save storage space?

A: Database data compression is the process of reducing the size of data stored in a database to save storage space. It is achieved by using algorithms that eliminate redundancies and use more efficient data representations.

186. Q: What is ACID-BASE, and how does it relate to database consistency in NoSQL systems?

A: ACID-BASE is an alternative model to the traditional ACID model for databases. It relaxes some of the ACID properties to achieve better scalability and availability in NoSQL databases, accepting eventual consistency instead of immediate consistency.

187. Q: What is the purpose of the SQL HAVING clause, and when is it used in queries?

A: The SQL HAVING clause filters data based on specified conditions after a GROUP BY operation, typically used with aggregate functions in GROUP BY queries.

188. Q: What is database data replication topology, and how does it affect replication performance and reliability?

A: Database data replication topology refers to the way data is copied and distributed across replica nodes. Different topologies (e.g., master-slave, master-master) impact

replication performance, latency, and data consistency.

189. Q: What is NoSQL query languages, and how do they differ from SQL?

A: NoSQL databases often use query languages specific to their data model, which can differ significantly from SQL in terms of syntax and capabilities.

NoSQL Query Languages হলো প্রতিটি NoSQL ডেটাবেইসের নিজস্ব কুয়েরি লেখার নিয়ম বা ভাষা, যা SQL-এর মতো হলেও ডেটা মডেল অনুযায়ী অনেকটাই আলাদা হয়ে থাকে।

190. Q: What is database data tiering, and how is it utilized for cost optimization?

A: Database data tiering involves categorizing data based on its access frequency and moving it to different storage tiers (e.g., fast SSD, slower HDD) to optimize costs and performance.

Database Data Tiering হলো এমন একটি কৌশল, যেখানে ডেটাকে তার ব্যবহারের পরিমাণ বা অ্যাক্সেসের ফ্রিকোয়েন্সি অনুযায়ী বিভিন্ন স্তরের স্টোরেজে ভাগ করে সংরক্ষণ করা হয়—

যেমন: দ্রুত SSD, ধীরগতির HDD ইত্যাদি।

191. Q: What is NoSQL serverless architecture, and how does it benefit database management?

A: NoSQL serverless architecture allows developers to focus on application logic without managing the underlying infrastructure, providing automatic scaling and cost optimization.

NoSQL Serverless Architecture এমন একটি ব্যবস্থা, যেখানে ডেভেলপারদের ইনফ্রাস্ট্রাকচার ম্যানেজ করার দরকার পড়ে না। তারা শুধু অ্যাপ্লিকেশনের লজিক নিয়ে কাজ করতে পারে, বাকি বিষয় (যেমন: সার্ভার ম্যানেজমেন্ট, ক্লেই়্যান্স ইত্যাদি) অটোমেটিকভাবে পরিচালিত হয়।

192. Q: What is database in-memory processing, and how does it improve query performance?

A: Database in-memory processing stores data in RAM for faster data access and processing, significantly improving query performance compared to disk-based databases.

193. Q: What is the role of a database performance tuning expert, and how do they optimize database performance?

A: A database performance tuning expert analyzes database configurations, queries, indexes, and hardware resources to identify and implement optimizations that improve overall database performance.

194. Q: What is NoSQL geospatial database, and when is it suitable for use?

A: A NoSQL geospatial database is designed to handle spatial data, enabling queries based on geographical locations, making it suitable for mapping, GIS, and location-based applications.

NoSQL Geospatial Database হলো এমন একটি ডেটাবেইস যা স্থানভিত্তিক (spatial) বা ভৌগলিক (geographical) ডেটা পরিচালনা করতে পারে। এটি ব্যবহারকারীদের লোকেশন ভিত্তিক কুয়েরি চালাতে দেয়, যেমন:

- “১ কিলোমিটারের মধ্যে কোন দোকান আছে?”
- “এই এলাকায় কোন ইউজার রয়েছে?”

এটি সাধারণত ম্যাপিং, GIS (Geographic Information Systems), এবং লোকেশন-ভিত্তিক অ্যাপ্লিকেশন তৈরিতে ব্যবহার করা হয়।

195. Q: What is database buffer pool management, and how does it impact data caching?

A: Database buffer pool management involves managing the cache of data pages in memory to improve query performance by reducing the need to read data from disk.

ডেটাবেইস বাফার পুল ম্যানেজমেন্ট হলো এমন একটি প্রক্রিয়া যেখানে ডেটার পেজগুলো (data pages) মেমোরিতে ক্যাশ করা হয়। এটি বারবার ডিস্ক থেকে ডেটা পড়ার দরকার কমিয়ে কুয়েরি পারফরম্যান্স উন্নত করে। ♦ উদাহরণস্বরূপ, frequently accessed ডেটা মেমোরিতে থাকলে, পরবর্তী রিড অপারেশনগুলো অনেক দ্রুত হয়।

196. **Q: What is NoSQL change data capture, and how is it used for real-time data processing?** A: NoSQL change data capture captures and streams data changes in real-time, allowing applications to react to data updates and propagate changes to downstream systems.
197. **Q: What is database table indexing strategy, and how is it determined?** A: Database table indexing strategy involves choosing the right columns and types of indexes to create for a table, based on the types of queries and access patterns performed on the data.
198. **Q: What is NoSQL database compaction, and how does it benefit data storage efficiency?** A: NoSQL database compaction is the process of reclaiming storage space by removing obsolete or deleted data and optimizing data organization, resulting in improved storage efficiency.
NoSQL Database Compaction হলো এমন একটি প্রক্রিয়া যা পুরানো বা মুছে ফেলা ডেটা সরিয়ে দিয়ে স্টোরেজ স্পেস পুনরুদ্ধার করে। এছাড়া এটি ডেটাকে নতুনভাবে সংগঠিত করে, ফলে রিড পারফরম্যান্স এবং স্টোরেজ ইফিষিয়েন্সি দ্বিগুণ হওয়া পরিপন্থ।
199. **Q: What is database read scaling, and how is it achieved in distributed databases?** A: Database read scaling involves distributing read queries across multiple replica nodes to handle read-intensive workloads and increase query throughput in a distributed database.
ডেটাবেইস রিড স্কেলিং হলো এমন একটি কৌশল, যেখানে রিড কুয়েরিগুলো একাধিক রিপ্লিকা নোডে ছড়িয়ে দেওয়া হয়। এর ফলে রিড পারফরম্যান্স বাড়ে, বিশেষ করে read-heavy অ্যাপ্লিকেশনে।
200. **Q: What is NoSQL BigTable database, and what are its use cases?** A: NoSQL BigTable database is a distributed, wide-column store database designed for large-scale, high-performance applications, such as Google's services, analytics platforms, and IoT data processing.
201. **Q: What is the purpose of the SQL CROSS JOIN statement, and when is it used in queries?** A: The SQL CROSS JOIN statement produces the Cartesian product of two or more tables, combining all rows from each table. It is used when you need to combine all possible combinations of rows between tables.
202. **Q: What is Oracle Database?** A: Oracle Database is a relational database management system (RDBMS) developed by Oracle Corporation. It is widely used for storing, managing, and retrieving data in various applications.
203. **Q: What are the main components of an Oracle database?** A: The main components are the instance (memory structures and background processes) and the database (data files, control files, and redo log files).
204. **Q: What is an Oracle tablespace?**
A: An Oracle tablespace is a logical storage container that groups related data files, providing better management of storage space and performance.
205. **Q: Explain the difference between a unique key and a primary key in Oracle.** A: A unique key ensures that the values in a column are unique, but it allows for NULL values. A primary key is a unique key that also enforces a NOT NULL

constraint.

206. Q: How can you create a new user in Oracle?

A: You can create a new user using the CREATE USER statement and specifying the username and password, along with other optional settings like default tablespace and temporary tablespace.

207. Q: What is the purpose of the Oracle Data Dictionary?

A: The Oracle Data Dictionary is a collection of database tables and views that store metadata about the database, including information about tables, columns, indexes, users, and privileges.

208. Q: Explain the difference between a clustered and non-clustered index in Oracle. A: A clustered index determines the physical order of data in a table, whereas a non-clustered index creates a separate structure to map the index to the data.

একটি Clustered Index হলো এমন একটি ইনডেক্স যা টেবিলের ডেটার ফিজিক্যাল অর্ডার নির্ধারণ করে।

অন্যদিকে, Non-Clustered Index হলো একটি আলাদা কাঠামো, যা মূল ডেটার ঠিকানার (location) সাথে ইনডেক্স ভ্যালু যুক্ত করে রাখে।

209. Q: What is an Oracle sequence, and how is it used?

A: An Oracle sequence is a database object that generates unique integers in a sequential order. It is commonly used to generate primary key values for tables.

210. Q: How can you backup an Oracle database, and what are the different backup methods?

A: Oracle databases can be backed up using the RMAN (Recovery Manager) utility. Different backup methods include full backups, incremental backups, and archive log backups.

211. Q: What is the purpose of the Oracle RAC (Real Application Clusters)?

A: Oracle RAC allows multiple instances to access a single Oracle database simultaneously, providing high availability, scalability, and load balancing.

212. Q: How can you tune the performance of an Oracle database?

A: Performance tuning in Oracle involves optimizing SQL queries, creating indexes, adjusting memory settings, and analyzing execution plans.

213. Q: What is an Oracle PL/SQL block, and how is it executed?

A: An Oracle PL/SQL block is a set of one or more SQL and PL/SQL statements that are executed together as a unit. It can be executed using anonymous blocks or stored procedures/functions.

214. Q: How can you find the number of rows affected by an SQL statement in Oracle?

A: You can use the SQL%ROWCOUNT attribute to retrieve the number of rows affected by the last SQL statement.

215. Q: Explain the difference between COMMIT and ROLLBACK in Oracle.

A: COMMIT is used to save the changes made during a transaction, while ROLLBACK is used to undo the changes and restore the database to its state before the transaction.

216. **Q: What is the purpose of the Oracle SQLPlus tool?**

A: SQLPlus is a command-line interface tool provided by Oracle for interacting with the Oracle Database. It is widely used for executing SQL queries and scripts.

217. **Q: What is an Oracle trigger, and when is it used?**

A: An Oracle trigger is a named PL/SQL block that is automatically executed when specific database events occur, such as INSERT, UPDATE, or DELETE on a table.

218. **Q: How can you find the execution plan for an SQL statement in Oracle?** A: You can use the EXPLAIN PLAN statement or the DBMS_XPLAN package to view the execution plan for an SQL statement.

219. **Q: What are Oracle hints, and how are they used in SQL queries?**

A: Oracle hints are special comments that provide instructions to the Oracle optimizer on how to execute a specific SQL statement. They are used to influence the execution plan.

220. **Q: Explain the difference between a VARCHAR and a CHAR data type in Oracle.**

A: Both VARCHAR and CHAR are used to store character data. The main difference is that VARCHAR stores variable-length strings, while CHAR stores fixed-length strings.

221. **Q: How can you grant privileges to a user in Oracle?**

A: You can use the GRANT statement to grant privileges such as SELECT, INSERT, UPDATE, DELETE, and others to a user on specific tables or database objects.