DBMS Interview Questions:

1. What is DBMS?

- **Answer**: A Database Management System (DBMS) is software that provides an interface to interact with databases, allowing users to store, retrieve, and manage data.
- **Example**: Examples of DBMS include MySQL, PostgreSQL, Oracle, and SQL Server.

2. What are the types of DBMS?

- **Answer**: There are four types of DBMS:
  - **Hierarchical DBMS**
  - **Network DBMS**
  - **Relational DBMS**
  - **Object-oriented DBMS**

3. What is a primary key?

- **Answer**: A primary key is a unique identifier for a record in a database table.
- **Example**:

```
CREATE TABLE Students (
    StudentID INT PRIMARY KEY,
    Name VARCHAR(50)
);
```


4. What is a foreign key?

- **Answer**: A foreign key is a column or set of columns in a table that references the primary key of another table.
- **Example**:

```
CREATE TABLE Orders (
    OrderID INT PRIMARY KEY,
    CustomerID INT,
    FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)
);
```

5. What is normalization?

- **Answer**: Normalization is the process of organizing data in a database to minimize redundancy and dependency.
- **Example**: Converting a database with repeating groups into a set of related tables.

6. What are the types of normalization?

- **Answer**:
  - **1NF**: Ensures atomicity of data.
  - **2NF**: Removes partial dependency.
  - **3NF**: Removes transitive dependency.
  - **BCNF**: A stricter version of 3NF.

7. What is denormalization?

- **Answer**: Denormalization is the process of introducing redundancy into a database to improve query performance at the cost of data integrity.
- **Example**: Adding redundant columns to a table to avoid joins.

8. What is a composite key?

- **Answer**: A composite key is a combination of two or more columns used to uniquely identify a row in a table.
- **Example**:

```
CREATE TABLE Enrollment (
    StudentID INT,
    CourseID INT,
    PRIMARY KEY (StudentID, CourseID)
);
```


9. What is a database index?

- **Answer**: An index is a database object that improves the speed of data retrieval operations.
- **Example**:

```
CREATE INDEX idx_name ON Students (Name);
```

10. What is a view in SQL?

- **Answer**: A view is a virtual table based on the result of a SELECT query.
- **Example**:

```
CREATE VIEW EmployeeDetails AS
SELECT Name, Department FROM Employees WHERE Salary > 50000;
```

11. What is the difference between DROP, DELETE, and TRUNCATE?

- **Answer**:
    - DROP: Removes the table and its structure.
    - DELETE: Removes data from a table but retains its structure.
    - TRUNCATE: Removes all data from a table and resets identity columns.
- **Example**:

DROP TABLE Employees;

DELETE FROM Employees WHERE Salary < 50000;

TRUNCATE TABLE Employees;

12. What is a JOIN operation in SQL?

- **Answer**: A JOIN operation is used to combine rows from two or more tables based on a related column.
- **Example**:

SELECT Employees.Name, Departments.DepartmentName

FROM Employees

INNER JOIN Departments ON Employees.DepartmentID = Departments.DepartmentID;

13. What is the difference between INNER JOIN and OUTER JOIN?

- **Answer**:
    - **INNER JOIN**: Returns only matching rows.
    - **OUTER JOIN**: Returns all rows, even if there is no match.
- **Example**:

SELECT * FROM Employees INNER JOIN Departments ON Employees.DepartmentID = Departments.DepartmentID;

SELECT * FROM Employees LEFT OUTER JOIN Departments ON Employees.DepartmentID = Departments.DepartmentID;

14. What is a GROUP BY clause in SQL?

- **Answer**: The GROUP BY clause is used to group rows that have the same values into summary rows, often used with aggregate functions like COUNT(), SUM(), AVG().
- **Example**:

SELECT Department, COUNT(*) FROM Employees GROUP BY Department;

15. What is an aggregate function in SQL?

- **Answer**: Aggregate functions perform a calculation on a set of values and return a single value.
- **Examples**: COUNT(), SUM(), AVG(), MAX(), MIN().

- **Example**:

SELECT AVG(Salary) FROM Employees;

16. What is a subquery in SQL?

- **Answer**: A subquery is a query nested inside another query.
- **Example**:

SELECT Name FROM Employees WHERE DepartmentID = (SELECT DepartmentID FROM Departments WHERE DepartmentName = 'HR');

17. What is the HAVING clause in SQL?

- **Answer**: The HAVING clause is used to filter groups created by the GROUP BY clause.

SELECT Department, COUNT(*) FROM Employees GROUP BY Department HAVING COUNT(*) > 5;

18. What is the difference between WHERE and HAVING?

- **Answer**:
    - WHERE: Filters rows before grouping.
    - HAVING: Filters groups after grouping.
- **Example**:

SELECT Department, COUNT(*) FROM Employees WHERE Salary > 50000 GROUP BY Department HAVING COUNT(*) > 5;

19. What is a stored procedure?

- **Answer**: A stored procedure is a precompiled collection of SQL statements that can be executed as a single unit.
- **Example**:

CREATE PROCEDURE GetEmployeeDetails AS

BEGIN

SELECT * FROM Employees;

END;

20. What is a trigger in SQL?

- **Answer**: A trigger is a set of SQL statements that are automatically executed in response to certain events on a table or view.
- **Example**:

CREATE TRIGGER UpdateEmployeeSalary

AFTER UPDATE ON Employees

FOR EACH ROW

```
BEGIN
    UPDATE SalaryHistory SET NewSalary = NEW.Salary WHERE EmployeeID = OLD.EmployeeID;
END;
```

---

21. What is an EXCEPT operator in SQL?

- **Answer**: The EXCEPT operator returns the difference between two SELECT queries.
- **Example**:

```
SELECT Name FROM Employees
EXCEPT
SELECT Name FROM Contractors;
```

22. What is the IN operator in SQL?

- **Answer**: The IN operator is used to filter records based on a list of values.
- **Example**:

```
SELECT * FROM Employees WHERE Department IN ('HR', 'IT');
```

23. What is the BETWEEN operator in SQL?

- **Answer**: The BETWEEN operator is used to filter records within a specified range.
- **Example**:

```
SELECT * FROM Employees WHERE Salary BETWEEN 30000 AND 60000;
```

24. What is a NULL value in SQL?

- **Answer**: A NULL value represents the absence of a value in a table.
- **Example**:

```
SELECT * FROM Employees WHERE Salary IS NULL;
```

25. What is a database index?

- **Answer**: A database index is used to improve the speed of data retrieval operations.
- **Example**:

```
CREATE INDEX idx_name ON Employees (Name);
```

26. What is a materialized view?

- **Answer**: A materialized view stores the result of a query physically, and can be refreshed periodically.
- **Example**:

```
CREATE MATERIALIZED VIEW EmployeeSalaries AS
```

```
SELECT Department, AVG(Salary) FROM Employees GROUP BY Department;
```

27. What is a composite key?

- **Answer**: A composite key is a primary key that consists of more than one column.
- **Example**:

```
CREATE TABLE Enrollment (
    StudentID INT,
    CourseID INT,
    PRIMARY KEY (StudentID, CourseID)
);
```

28. What is a candidate key?

- **Answer**: A candidate key is a set of columns that could uniquely identify rows in a table, but is not necessarily the primary key.
- **Example**:

```
CREATE TABLE Students (
    StudentID INT,
    Email VARCHAR(50),
    PRIMARY KEY (StudentID)
);
```

29. What is a CASE statement in SQL?

- **Answer**: The CASE statement allows conditional logic in SQL queries.
- **Example**:

```
SELECT Name,
    CASE
        WHEN Salary > 50000 THEN 'High'
        ELSE 'Low'
    END AS SalaryLevel
FROM Employees;
```

30. What is a database transaction?

- **Answer**: A transaction is a unit of work that is performed against a database, and it is either completed in full or not at all.
- **Example**:

```
BEGIN TRANSACTION;
```

```
UPDATE Accounts SET Balance = Balance - 500 WHERE AccountID = 1;

UPDATE Accounts SET Balance = Balance + 500 WHERE AccountID = 2;

COMMIT;
```

---

31. What are ACID properties?

- **Answer**: ACID properties ensure that database transactions are processed reliably.
    - **Atomicity**: A transaction is all or nothing.
    - **Consistency**: The database is in a valid state before and after the transaction.
    - **Isolation**: Transactions are isolated from each other.
    - **Durability**: Changes are permanent once committed.
- **Example**: If you transfer money between two accounts, all the steps (debit and credit) must either complete successfully or none of them should be executed.

32. What is a deadlock in DBMS?

- **Answer**: A deadlock occurs when two or more transactions are waiting for each other to release locks, resulting in a cycle of dependencies.
- **Example**:
    - Transaction 1 locks Resource A and waits for Resource B.
    - Transaction 2 locks Resource B and waits for Resource A.

33. What is the ROLLBACK command?

- **Answer**: The ROLLBACK command is used to undo changes made by the current transaction.
- **Example**:

```
ROLLBACK;
```

34. What is the COMMIT command?

- **Answer**: The COMMIT command is used to save the changes made by a transaction.
- **Example**:

```
COMMIT;
```

35. What is the difference between INNER JOIN and OUTER JOIN?

- **Answer**:
    - INNER JOIN: Returns only matching rows.
    - OUTER JOIN: Returns all rows, even if there is no match.
- **Example**:

SELECT * FROM Employees INNER JOIN Departments ON Employees.DepartmentID = Departments.DepartmentID;

SELECT * FROM Employees LEFT OUTER JOIN Departments ON Employees.DepartmentID = Departments.DepartmentID;

36. What is a SELF JOIN?

- **Answer**: A SELF JOIN is a join where a table is joined with itself.
- **Example**:

SELECT E1.Name AS Employee, E2.Name AS Manager

FROM Employees E1

LEFT JOIN Employees E2 ON E1.ManagerID = E2.EmployeeID;

37. What is a UNION operator in SQL?

- **Answer**: The UNION operator is used to combine the results of two SELECT queries and remove duplicates.
- **Example**:

SELECT Name FROM Employees

UNION

SELECT Name FROM Contractors;

38. What is a UNION ALL operator in SQL?

- **Answer**: The UNION ALL operator combines the results of two SELECT queries without removing duplicates.
- **Example**:

SELECT Name FROM Employees

UNION ALL

SELECT Name FROM Contractors;

39. What is the DISTINCT keyword in SQL?

- **Answer**: The DISTINCT keyword is used to return only unique values.
- **Example**:

SELECT DISTINCT Department FROM Employees;

40. What is a schema in a database?

- **Answer**: A schema is a collection of database objects, such as tables, views, indexes, and procedures, that define the structure of a database.

41. What is a CLUSTERED INDEX?

- **Answer**: A clustered index determines the physical order of data in a table. A table can have only one clustered index.
- **Example**:

CREATE CLUSTERED INDEX idx_employee_name ON Employees(Name);

42. What is a NON-CLUSTERED INDEX?

- **Answer**: A non-clustered index does not affect the physical order of data in the table. A table can have multiple non-clustered indexes.
- **Example**:

CREATE NONCLUSTERED INDEX idx_employee_salary ON Employees(Salary);

43. What is the EXPLAIN command in SQL?

- **Answer**: The EXPLAIN command is used to display the execution plan of a query, helping to analyze performance.
- **Example**:

EXPLAIN SELECT * FROM Employees WHERE Department = 'HR';

44. What is the HAVING clause used for?

- **Answer**: The HAVING clause is used to filter the results of a GROUP BY query.
- **Example**:

SELECT Department, COUNT(*) FROM Employees GROUP BY Department HAVING COUNT(*) > 5;

45. What is a transaction log in DBMS?

- **Answer**: A transaction log is a file that records all changes made to the database to support recovery in case of failure.

46. What is the LIMIT clause in SQL?

- **Answer**: The LIMIT clause is used to restrict the number of rows returned by a query.
- **Example**:

SELECT * FROM Employees LIMIT 10;

47. What is the difference between DELETE and TRUNCATE?

- **Answer**:
    - DELETE: Removes data but allows for rollback.
    - TRUNCATE: Removes all data and cannot be rolled back.
- **Example**:

DELETE FROM Employees WHERE Department = 'HR';

TRUNCATE TABLE Employees;

48. What is a foreign key constraint?

- **Answer**: A foreign key constraint ensures referential integrity by enforcing a relationship between two tables.
- **Example**:

```
CREATE TABLE Orders (
    OrderID INT PRIMARY KEY,
    CustomerID INT,
    FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)
);
```

49. What is the SELECT INTO statement in SQL?

- **Answer**: The SELECT INTO statement is used to create a new table from the results of a SELECT query.
- **Example**:

```
SELECT * INTO NewEmployees FROM Employees WHERE Department = 'HR';
```

50. What is a FOREIGN KEY constraint in DBMS?

- **Answer**: A foreign key constraint ensures that values in one table correspond to values in another table, ensuring referential integrity.
- **Example**:

```
CREATE TABLE Orders (
    OrderID INT PRIMARY KEY,
    CustomerID INT,
    FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)
);
```

51. What is the difference between a TRUNCATE and DELETE statement in SQL?

- **Answer**:
    - DELETE: Removes records one at a time and logs each removal, which can be rolled back.
    - TRUNCATE: Removes all records and cannot be rolled back.

52. What is a database schema?

- **Answer**: A schema is the organizational blueprint of a database, which defines its structure, including tables, columns, relationships, views, etc.

53. What is the difference between a JOIN and a UNION?

- **Answer**:
    - JOIN: Combines rows from two or more tables based on a related column.
    - UNION: Combines the results of two SELECT queries into a single result set, removing duplicates.

54. What is a STORED PROCEDURE?

- **Answer**: A stored procedure is a set of SQL statements that are stored in the database and can be executed as a single unit.
- **Example**:

```
CREATE PROCEDURE GetEmployeeDetails

AS

BEGIN

   SELECT * FROM Employees;

END;
```

55. What is a TRIGGER?

- **Answer**: A trigger is a set of SQL statements that are automatically executed in response to specific events on a table or view.
- **Example**:

```
CREATE TRIGGER UpdateEmployeeSalary

AFTER UPDATE ON Employees

FOR EACH ROW

BEGIN

   UPDATE SalaryHistory SET NewSalary = NEW.Salary WHERE EmployeeID = OLD.EmployeeID;

END;
```

56. What is a VIEW in SQL?

- **Answer**: A view is a virtual table based on the result of a SELECT query. It does not store data but provides a way to access data from one or more tables.
- **Example**:

```
CREATE VIEW EmployeeDetails AS

SELECT Name, Department FROM Employees WHERE Salary > 50000;
```

57. What is a CONSTRAINT in SQL?

- **Answer**: A constraint is a rule that limits the type of data that can be inserted into a table to ensure data integrity.
- **Example**:

```
CREATE TABLE Employees (
    EmployeeID INT PRIMARY KEY,
    Name VARCHAR(100),
    Salary DECIMAL(10, 2) CHECK (Salary > 0)
);
```

58. What is the SELECT statement used for in SQL?

- **Answer**: The SELECT statement is used to retrieve data from one or more tables.
- **Example**:

```
SELECT Name, Salary FROM Employees WHERE Department = 'IT';
```

59. What is normalization in DBMS?

- **Answer**: Normalization is the process of organizing data to minimize redundancy and dependency by dividing large tables into smaller ones and defining relationships.
- **Example**: Normalizing a table by creating separate tables for Employees and Departments.

60. What is denormalization in DBMS?

- **Answer**: Denormalization is the process of combining normalized tables to reduce the number of joins, improving query performance.
- **Example**: Merging Employees and Departments tables for faster retrieval.

---

61. What is the purpose of the GROUP BY clause in SQL?

- **Answer**: The GROUP BY clause is used to group rows that have the same values in specified columns into summary rows, like counting or summing.
- **Example**:

```
SELECT Department, COUNT(*) FROM Employees GROUP BY Department;
```

62. What is a PRIMARY KEY?

- **Answer**: A PRIMARY KEY is a column or set of columns that uniquely identifies each row in a table.
- **Example**:

```
CREATE TABLE Employees (
    EmployeeID INT PRIMARY KEY,
    Name VARCHAR(100)
);
```

63. What is a FOREIGN KEY?

- **Answer**: A FOREIGN KEY is a column or a set of columns in one table that refers to the PRIMARY KEY of another table, ensuring referential integrity.
- **Example**:

```
CREATE TABLE Orders (
    OrderID INT PRIMARY KEY,
    EmployeeID INT,
    FOREIGN KEY (EmployeeID) REFERENCES Employees(EmployeeID)
);
```

64. What is the difference between CHAR and VARCHAR?

- **Answer**:
    - CHAR: A fixed-length string.
    - VARCHAR: A variable-length string.
- **Example**:

```
CREATE TABLE Employees (
    Name CHAR(100),
    Address VARCHAR(255)
);
```

65. What is the AUTO_INCREMENT property?

- **Answer**: The AUTO_INCREMENT property is used to automatically generate a unique number for a primary key field.
- **Example**:

```
CREATE TABLE Employees (
    EmployeeID INT AUTO_INCREMENT PRIMARY KEY,
    Name VARCHAR(100)
);
```

66. What is the difference between TRUNCATE and DELETE commands?

- **Answer**:
    - DELETE: Removes rows one at a time and can be rolled back.
    - TRUNCATE: Removes all rows in a table and cannot be rolled back.
- **Example**:

```
DELETE FROM Employees WHERE Department = 'HR';
TRUNCATE TABLE Employees;
```

67. What is an INDEX in DBMS?

- **Answer**: An index is a database object that improves the speed of data retrieval operations on a table at the cost of additional space.
- **Example**:

```
CREATE INDEX idx_employee_name ON Employees(Name);
```

68. What is a COMPOSITE KEY?

- **Answer**: A composite key is a primary key that consists of more than one column.
- **Example**:

```
CREATE TABLE Orders (
    OrderID INT,
    ProductID INT,
    PRIMARY KEY (OrderID, ProductID)
);
```

69. What is the difference between DROP and DELETE?

- **Answer**:
    - DROP: Removes a table or database permanently.
    - DELETE: Removes rows from a table but retains the table structure.
- **Example**:

```
DROP TABLE Employees;
DELETE FROM Employees WHERE Department = 'HR';
```

70. What is the difference between INNER JOIN and OUTER JOIN?

- **Answer**:
    - INNER JOIN: Returns only the rows that have matching values in both tables.
    - OUTER JOIN: Returns all rows from one table and the matching rows from the other table.
- **Example**:

```
SELECT * FROM Employees INNER JOIN Departments ON Employees.DepartmentID = Departments.DepartmentID;
SELECT * FROM Employees LEFT OUTER JOIN Departments ON Employees.DepartmentID = Departments.DepartmentID;
```

71. What is the difference between UNION and UNION ALL?

- **Answer**:

    - UNION: Combines results from two queries and removes duplicates.
    - UNION ALL: Combines results without removing duplicates.
- **Example**:

SELECT Name FROM Employees UNION SELECT Name FROM Contractors;

SELECT Name FROM Employees UNION ALL SELECT Name FROM Contractors;

72. What is a VIEW in SQL?

- **Answer**: A view is a virtual table that contains the result of a SELECT query. It does not store data but allows you to query it like a regular table.
- **Example**:

CREATE VIEW EmployeeView AS

SELECT Name, Department FROM Employees WHERE Salary > 50000;

73. What is the EXPLAIN command in SQL?

- **Answer**: The EXPLAIN command provides the execution plan of a query, showing how the database will execute the query and optimize it.
- **Example**:

EXPLAIN SELECT * FROM Employees WHERE Department = 'HR';

74.What are the main components of an ER diagram?

- **Answer: The main components of an ER diagram are:**
    - **Entities: Represent objects or things within the system.**
    - **Attributes: Properties or characteristics of entities.**
    - **Relationships: Connections between entities.**
- **Example: In a university database, Student and Course are entities, Enrolled is the relationship, and attributes could include StudentName, CourseName.**

75. What is the difference between a Strong Entity and a Weak Entity in ER modeling?

- **Answer:**
    - **Strong Entity: Can exist independently and has a primary key.**
    - **Weak Entity: Cannot exist without a related strong entity and lacks a primary key.**
- **Example: A Student is a strong entity, while a CourseEnrollment might be a weak entity because it depends on the Student entity for its existence.**

76. What is a TRANSACTION in DBMS?

- **Answer**: A transaction is a unit of work that is performed against a database. It is a sequence of operations performed as a single logical unit.

- **Example**: A transaction might include multiple steps such as updating an account balance and logging the transaction.

77. What is a CHECK constraint in SQL?

- **Answer**: The CHECK constraint is used to limit the range of values that can be entered into a column.
- **Example**:

```
CREATE TABLE Employees (
    EmployeeID INT PRIMARY KEY,
    Salary DECIMAL(10, 2) CHECK (Salary > 0)
);
```

78. What is ACID in DBMS?

- **Answer**: ACID stands for:
    - **Atomicity**: A transaction is either fully completed or fully rolled back.
    - **Consistency**: The database is in a valid state before and after the transaction.
    - **Isolation**: Transactions are isolated from each other.
    - **Durability**: Changes made by a transaction are permanent.
- **Example**: In a banking transaction, money is either fully transferred or not transferred at all.

79. What is CAP Theorem in DBMS?

- **Answer**: The CAP Theorem states that in a distributed data store, you can only achieve two out of the following three guarantees:
    - **Consistency**: Every read gets the most recent write.
    - **Availability**: Every request receives a response (either success or failure).
    - **Partition Tolerance**: The system continues to function despite network partitions.
- **Example**: In a distributed database, if a network partition occurs, you may need to choose between consistency and availability.

80. What is a CLUSTERED INDEX?

- **Answer**: A clustered index sorts the data rows based on the index key. Each table can have only one clustered index because the data rows can only be sorted in one way.
- **Example**:

```
CREATE CLUSTERED INDEX idx_employee_name ON Employees(Name);
```

81. What is DEADLOCK in DBMS?

- **Answer**: A deadlock occurs when two or more transactions are blocked, each waiting for the other to release a lock, resulting in a circular dependency.

- **Example**: Transaction 1 holds a lock on Resource A and waits for Resource B, while Transaction 2 holds a lock on Resource B and waits for Resource A.

82. What is DATABASE MIGRATION?

- **Answer**: Database migration is the process of transferring data between different types of databases or versions of a database.
- **Example**: Migrating data from an on-premise MySQL database to a cloud-based PostgreSQL database.

83. What is the difference between TRUNCATE and DELETE?

- **Answer**:
    - TRUNCATE: Removes all rows from a table and cannot be rolled back.
    - DELETE: Removes specific rows and can be rolled back.
- **Example**:

```
DELETE FROM Employees WHERE Department = 'HR';

TRUNCATE TABLE Employees;
```

84. What is SHARDING in DBMS?

- **Answer**: Sharding is the process of breaking up large databases into smaller, more manageable pieces called shards, often distributed across multiple servers.
- **Example**: A large customer database might be split into shards based on geographic regions.

---

85. What is Referential Integrity in DBMS?

- **Answer**: Referential integrity ensures that relationships between tables remain consistent. When one table references another (via a foreign key), referential integrity ensures that foreign key values correspond to primary key values in the referenced table.
- **Example**:

```
CREATE TABLE Orders (
    OrderID INT PRIMARY KEY,
    CustomerID INT,
    FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)
);
```

86. What is SQL Injection?

- **Answer**: SQL injection is a code injection technique that exploits vulnerabilities in an application's software by manipulating SQL queries to execute arbitrary SQL code.
- **Example**: An attacker could input malicious SQL code into a form input field like '; DROP TABLE Employees; -- to delete a table.

87. What is a Self Join in SQL?

- **Answer**: A self join is a join where a table is joined with itself to combine rows based on a related column.
- **Example**:

```
SELECT E1.EmployeeID, E1.Name, E2.Name AS Manager
FROM Employees E1, Employees E2
WHERE E1.ManagerID = E2.EmployeeID;
```

88.What is a Master-Slave Replication in DBMS?

- **Answer: Master-Slave replication refers to a configuration where one database (master) acts as the primary source of data, and other databases (slaves) replicate the data from the master. The master handles write operations, while slaves handle read operations, providing high availability and load balancing.**
- **Example: In a typical setup, the master database holds all the data, and the slave databases synchronize with it. Any changes to the master (inserts, updates, deletes) are replicated to the slave databases.**

89. What is the Entity-Relationship (ER) Model?

- **Answer: The Entity-Relationship (ER) model is a high-level conceptual data model that describes the data and its relationships using entities, attributes, and relationships. It helps in designing the structure of a database.**
- **Example:**
    - **Entities: Customer, Order**
    - **Attributes: CustomerID, OrderID**
    - **Relationship: Customer places Order.**

90. What is a Transaction Log in DBMS?

- **Answer**: A transaction log is a record of all transactions that have been executed against a database. It helps in recovering the database in case of a crash.
- **Example**: A database system might log all insert, update, and delete operations to ensure durability and recovery.

91. What is Database Clustering?

- **Answer**: Database clustering is a method of connecting multiple database servers to form a cluster to provide high availability and scalability.

- **Example**: A database cluster might involve multiple nodes that handle read and write operations in parallel, improving performance.

92. What is the RAID technology in databases?

- **Answer**: RAID (Redundant Array of Independent Disks) is a technology that combines multiple hard drives into a single unit to improve performance and redundancy.
- **Example**: RAID 1 (mirroring) stores the same data on two or more drives to ensure redundancy.

93. What is DBMS Architecture and discuss it advantages?

- **Answer: DBMS architecture defines the design of the system and how different components interact with each other. There are generally three types of DBMS architecture:**
    - **1-Tier Architecture: The database and application are on the same system.**
    - **2-Tier Architecture: The database and application are on separate systems, with the client communicating directly with the database.**
    - **3-Tier Architecture: The client, application server, and database are separate, with the application server acting as an intermediary between the client and database.**
- **Example: In a 3-tier architecture, the client interacts with the application server, which in turn interacts with the database**

advantages of DBMS Architecture:

- **Answer: Some advantages include:**
    - **Data independence.**
    - **Efficient data management.**
    - **Improved data security.**
    - **Easier data backup and recovery.**
- **Example: A 3-tier architecture provides a separation of concerns, making it easier to maintain and scale the system.**

94. What is the role of Indexing in a database?

- **Answer**: Indexing improves the speed of data retrieval operations on a database table by creating a structure that allows for faster searching.
- **Example**:

CREATE INDEX idx_employee_name ON Employees(Name);

95. What is the difference between Clustered and Non-clustered indexes?

- **Answer**:
    - **Clustered Index**: The data is stored in the index itself, and the table can only have one clustered index.

    - **Non-clustered Index**: The index is stored separately from the data, and multiple non-clustered indexes can exist.

- **Example**:
    - Clustered Index: EmployeeID could be a clustered index because the rows are sorted based on the employee ID.
    - Non-clustered Index: EmployeeName could be a non-clustered index.

96. What is SQL vs NoSQL databases?

- **Answer**:
    - **SQL Databases**: Structured databases that use SQL for querying and are based on a relational model (e.g., MySQL, PostgreSQL).
    - **NoSQL Databases**: Non-relational databases that store data in a flexible, schema-less format (e.g., MongoDB, Cassandra).
- **Example**: SQL is used in applications requiring structured data with relationships, while NoSQL is used in applications that need scalability and flexibility.

97. What is a Stored Procedure in SQL?

- **Answer**: A stored procedure is a precompiled collection of one or more SQL statements that can be executed as a unit.
- **Example**:

```
CREATE PROCEDURE GetEmployeeInfo (IN emp_id INT)
BEGIN
  SELECT Name, Department FROM Employees WHERE EmployeeID = emp_id;
END;
```

98. What is the purpose of Triggers in DBMS?

- **Answer**: A trigger is a set of SQL statements that automatically executes when a specific event (INSERT, UPDATE, DELETE) occurs on a table.
- **Example**:

```
CREATE TRIGGER before_insert_employee
BEFORE INSERT ON Employees
FOR EACH ROW
SET NEW.created_at = NOW();
```

99. What is a Database Schema?

- **Answer**: A database schema is the structure that defines the organization of data, including tables, columns, relationships, and constraints.

- **Example**: A schema might define a Customers table with columns CustomerID, Name, and Email.

100. What is Replication in DBMS?

- **Answer**: Replication is the process of copying data from one database server to another to ensure data availability and redundancy.
- **Example**: A master-slave replication setup where the master database updates data and the slave databases replicate the changes.