

JDBC Viva Questions with Answers

1. What is JDBC?

Answer:

JDBC (Java Database Connectivity) is a standard Java API used to connect Java applications with relational databases such as MySQL, Oracle, PostgreSQL, and SQL Server.

It allows Java programs to:

- Establish database connections
- Execute SQL queries
- Retrieve results
- Perform CRUD operations
- Handle transactions

Example:

```
Connection con = DriverManager.getConnection(url, user, password);
```

2. What are the steps involved in JDBC programming?

Answer:

The standard JDBC workflow consists of:

1. Import JDBC packages
 2. Load the JDBC driver
 3. Establish a connection
 4. Create Statement/PreparedStatement
 5. Execute SQL query
 6. Process ResultSet
 7. Close all resources
-

3. What is the purpose of Class.forName()?

Answer:

It loads the JDBC driver class into memory and registers it with the DriverManager.

```
Class.forName("com.mysql.cj.jdbc.Driver");
```

In JDBC 4.0 and later, explicit loading is usually optional because the driver is auto-registered.

4. What is DriverManager?

Answer:

DriverManager is a class that manages JDBC drivers and establishes a connection to the database.

Example:

```
Connection con =
```

```
DriverManager.getConnection(url,user,password);
```

5. What is Connection?

Answer:

Connection represents an active communication session between a Java application and the database.

It is required before executing any SQL statement.

6. Difference between Statement and PreparedStatement?

Statement	PreparedStatement
Static SQL	Parameterized SQL
Slow	Faster
SQL Injection possible	Prevents SQL Injection
Compiled every time	Precompiled
Less secure	More secure

7. Why is PreparedStatement faster?

Answer:

PreparedStatement compiles the SQL statement only once.
When executed repeatedly with different values, only the parameters change.
Therefore:

- Less parsing
 - Less compilation
 - Better performance
-

8. What is SQL Injection?

Answer:

SQL Injection is an attack where malicious SQL commands are inserted through user input.

Example:

Instead of entering

Harish

User enters

' OR 1=1 --

This may return every record.

PreparedStatement prevents this attack.

9. What is ResultSet?

Answer:

ResultSet is an object that stores records returned by a SELECT query.

Example

```
ResultSet rs = st.executeQuery("SELECT * FROM student");
```

10. Why do we use rs.next()?

Answer:

Initially, the cursor is positioned before the first row.

next() moves the cursor to the next row.

It returns:

- true → row exists
 - false → no more rows
-

11. Difference between executeQuery(), executeUpdate(), and execute()?

Answer:

executeQuery()

- Used for SELECT
- Returns ResultSet

executeUpdate()

- Used for INSERT, UPDATE, DELETE
- Returns number of affected rows

execute()

- Used when SQL type is unknown
 - Returns boolean
-

12. What does executeUpdate() return?

Answer:

It returns the number of rows affected.

Example

2

means two rows were inserted or updated.

13. Why can't executeQuery() execute INSERT?

Answer:

executeQuery() expects a ResultSet.

INSERT returns the number of affected rows.

Hence SQLException occurs.

14. Why can't executeUpdate() execute SELECT?

Answer:

SELECT returns records.
executeUpdate() expects an integer.
Therefore an exception occurs.

15. What is Auto Commit?**Answer:**

By default, every SQL statement is committed immediately.
con.setAutoCommit(false);
disables automatic commit.

16. Difference between Commit and Rollback?**Commit**

- Saves changes permanently.

Rollback

- Cancels uncommitted changes.
-

17. What is Transaction?**Answer:**

A transaction is a group of SQL statements executed as a single unit.

Either:

- all succeed
 - all fail
-

18. What are ACID properties?**Answer:**

A – Atomicity

C – Consistency

I – Isolation

D – Durability

These ensure reliable transactions.

19. What is a Scrollable ResultSet?**Answer:**

It allows movement in both forward and backward directions.

Methods include

- next()
 - previous()
 - first()
 - last()
 - absolute()
-

20. Difference between Forward Only and Scrollable ResultSet?

Forward Only:

- Move only forward.

Scrollable:

- Move in both directions.
-

21. Difference between TYPE_SCROLL_INSENSITIVE and TYPE_SCROLL_SENSITIVE?**TYPE_SCROLL_INSENSITIVE**

- Does not reflect database changes.

TYPE_SCROLL_SENSITIVE

- Reflects database changes.
-

22. What is Updatable ResultSet?**Answer:**

An Updatable ResultSet allows updating records directly through the ResultSet without writing explicit SQL UPDATE statements.

23. Why is Primary Key required for Updatable ResultSet?**Answer:**

The JDBC driver must uniquely identify the row being updated. Without a Primary Key, it cannot determine which row to modify, leading to a NotUpdatable exception.

24. Explain moveToInsertRow().**Answer:**

It moves the cursor to a special insert buffer where values for a new row can be assigned before calling insertRow().

25. What does updateRow() do?**Answer:**

It writes the modified column values in the current row back to the database.

26. Difference between updateRow() and executeUpdate()?**updateRow()**

- Updates through ResultSet.

executeUpdate()

- Executes an SQL UPDATE statement.
-

27. Why is ResultSet initially positioned before the first row?**Answer:**

This design allows iteration using:
while(rs.next())
ensuring the first row is processed correctly.

28. What is a Cursor?**Answer:**

A cursor is a pointer that indicates the current row in a ResultSet.

29. What happens if rs.next() returns false?**Answer:**

There are no more rows to process, and iteration should stop.

30. What is SQLException?**Answer:**

SQLException is thrown when a database-related error occurs, such as invalid SQL, connection failures, or constraint violations.

31. Why should JDBC resources be closed?**Answer:**

Closing ResultSet, Statement, and Connection prevents resource leaks and frees database connections.

32. What is Connection Pooling?**Answer:**

Connection pooling reuses database connections instead of creating a new one each time, improving performance in enterprise applications.

33. What is Batch Processing?**Answer:**

Batch processing groups multiple SQL statements and sends them to the database in one operation, reducing network overhead.

34. What is ResultSetMetaData?**Answer:**

It provides information about the columns in a `ResultSet`, such as column names, types, and count.

35. What is `DatabaseMetaData`?

Answer:

It provides information about the database itself, including database name, version, supported features, and tables.

36. Difference between `ResultSetMetaData` and `DatabaseMetaData`?

- **`ResultSetMetaData`:** Information about query result columns.
 - **`DatabaseMetaData`:** Information about the database and driver.
-

37. What is `CallableStatement`?

Answer:

`CallableStatement` is used to execute stored procedures in the database.

38. What is the advantage of `PreparedStatement` over `Statement`?

Answer:

`PreparedStatement` offers:

- Better performance
 - Protection against SQL Injection
 - Easier parameter handling
 - Reusability
-

39. Why do `PreparedStatement` parameter indexes start from 1?

Answer:

JDBC follows SQL parameter numbering, where the first placeholder (?) is index **1**, not **0**.

40. Can a `PreparedStatement` be reused?

Answer:

Yes. A `PreparedStatement` can be executed multiple times with different parameter values without recompiling the SQL.

41. What happens if you forget to set a parameter in `PreparedStatement`?

Answer:

A `SQLException` is thrown because not all placeholders have been assigned values.

42. What is the difference between `getInt()` and `setInt()`?

Answer:

- `setInt()` sends an integer value to the database.
 - `getInt()` retrieves an integer value from the `ResultSet`.
-

43. What is SQL parsing?

Answer:

SQL parsing is the process where the database checks the SQL syntax and creates an execution plan. `PreparedStatement` avoids repeated parsing for the same SQL.

44. Why is Type-4 JDBC Driver the most commonly used?

Answer:

Type-4 is a pure Java driver that communicates directly with the database using its native protocol. It is portable, fast, and does not require native libraries.

45. What is the difference between `execute()` and `executeUpdate()`?

Answer:

- `execute()` returns a boolean and is useful when the SQL type is unknown.
 - `executeUpdate()` returns the number of affected rows for DML statements.
-

46. What causes a `NotUpdatable` exception?

Answer:

Common causes include:

- No primary key in the table.
 - Query involving joins or computed columns.
 - ResultSet not created with CONCUR_UPDATABLE.
-

47. How do you disable Auto Commit?**Answer:**

```
con.setAutoCommit(false);
```

This allows multiple SQL statements to be treated as a single transaction.

48. Why is exception handling important in JDBC?**Answer:**

Database operations can fail due to connection issues, SQL syntax errors, or constraint violations. Proper exception handling ensures that errors are managed gracefully and resources are released.

49. What are the benefits of using try-with-resources in JDBC?**Answer:**

try-with-resources automatically closes JDBC resources (Connection, Statement, and ResultSet), reducing boilerplate code and preventing resource leaks.

50. Why should you use PreparedStatement instead of concatenating SQL strings?**Answer:**

PreparedStatement:

- Prevents SQL Injection.
- Improves performance through precompilation.
- Handles data types safely.
- Makes the code cleaner and easier to maintain.