

PART I: Short Questions (Exercise Answers)

Q1. Define JDBC.

- **JDBC (Java Database Connectivity)** is a Java API that allows Java programs to **connect and interact with databases**.

Q2. Why should we use JDBC?

- JDBC enables:
 - Storing and retrieving data from databases.
 - Executing SQL queries directly in Java.
 - Making platform-independent database applications.

Q3. What is API?

- **API (Application Programming Interface):** A set of classes and methods used to interact with software components.
- Example: JDBC API to access databases.

Q4. What is MySQL?

- **MySQL** is an **open-source relational database management system (RDBMS)** that stores and manages data in tables.

Q5. Explain advantages of MySQL.

1. Open-source (free).
2. Cross-platform (Windows, Linux, Mac).
3. Supports large databases.
4. Secure and reliable.
5. Widely used with PHP, Java, and other technologies.

Q6. Explain connecting to the database.

- Steps to connect Java with database using JDBC:
 1. Register driver.
 2. Establish connection using DriverManager.
 3. Create Statement object.
 4. Execute queries.
 5. Close connection.

Q7. What is Sequential Access?

- Reading records **one by one in sequence** from database.
- Example: Iterating through ResultSet using next().

Q8. What is Random Access?

- Accessing records **in any order** using index.
- Example: Jumping to specific row using scrollable ResultSet.

Q9. Why we import JDBC package?

Chapter # 8

- To use JDBC classes and interfaces.
- Import statement:

```
import java.sql.*;
```

Q10. Explain Register JDBC Driver.

- First step in JDBC, loads the database driver.
- Example:

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

Q11. Explain Connection Object.

- Represents a connection (session) between Java program and database.
- Example:

```
Connection con = DriverManager.getConnection(
"jdbc:mysql://localhost:3306/test","root","password");
```

✓ PART II: Long Questions (Exercise Answers)

Q1. Write a note on MySQL.

- MySQL is a **popular relational database** that uses SQL for querying data.
- **Features:**
 - Supports large datasets.
 - Provides data security and backup.
 - Works well with Java (via JDBC).
- Example use cases: Banking systems, Online shopping, School management systems.

Q2. Write a note on executeQuery() method.

- Used to **execute SELECT statements** in SQL.
- Returns a ResultSet containing retrieved rows.
- Example:

```
Statement stmt = con.createStatement();
ResultSet rs = stmt.executeQuery("SELECT * FROM students");
while(rs.next()){
    System.out.println(rs.getInt(1)+" "+rs.getString(2));
}
```

Q3. Write a note on executeUpdate() method.

- Used to execute **INSERT, UPDATE, DELETE, and DDL statements**.
- Returns number of rows affected.
- Example:

```
Statement stmt = con.createStatement();
int i = stmt.executeUpdate("INSERT INTO students VALUES(1,'Ali')");
System.out.println(i+" record inserted");
```

Q4. Write a note on DML and DDL.

- **DML (Data Manipulation Language):** Deals with manipulation of data.
 - Commands: INSERT, UPDATE, DELETE, SELECT.
- **DDL (Data Definition Language):** Deals with structure of database.
 - Commands: CREATE, ALTER, DROP, TRUNCATE.

★ **Extra Important Short Questions with Answers**

Q1. What are the main JDBC classes/interfaces?

- DriverManager → Manages drivers.
- Connection → Connects to database.
- Statement → Executes SQL queries.
- ResultSet → Stores query results.

Q2. Difference between Statement and PreparedStatement.

- **Statement:** Executes simple SQL queries.
- **PreparedStatement:** Used for dynamic queries, prevents SQL injection.

Q3. What is ResultSet?

- A table of data returned by executeQuery().
- Methods: next(), getInt(), getString().

Q4. What is DriverManager in JDBC?

- A class that manages JDBC drivers and establishes connections.

Q5. What is SQL Injection?

- A security attack where malicious SQL is inserted in queries.
- Prevented using **PreparedStatement**.

★ **Extra Important Long Questions with Answers**

Q1. Explain JDBC Architecture.

1. **JDBC API** (Java application side).
2. **JDBC Driver Manager** (manages drivers).
3. **JDBC Drivers** (bridge between Java & DB).
4. **Database** (MySQL, Oracle, etc.).

Q2. Explain JDBC Steps with Example.

1. Import package → import java.sql.*;
2. Load driver → Class.forName("com.mysql.cj.jdbc.Driver");
3. Establish connection → DriverManager.getConnection(...)
4. Create statement → con.createStatement()
5. Execute query → executeQuery() or executeUpdate()
6. Process results.
7. Close connection.

Q3. Difference between DDL and DML with Examples.

DDL Commands	Example	DML Commands	Example
CREATE	CREATE TABLE students	INSERT	INSERT INTO students VALUES(1,'Ali')
ALTER	ALTER TABLE students ADD age INT	UPDATE	UPDATE students SET name='Asad' WHERE id=1
DROP	DROP TABLE students	DELETE	DELETE FROM students WHERE id=1