3. Create a table in an available MySQL database using JDBC program.

Solution:

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.SQLException;

public class INSERT_DATA {

    public static void main(String[] args) {
        // Database URL, username, and password
        String url = "jdbc:mysql://localhost:3306/firstdatabase"; //
Replace with your database name
        String user = "IshanKRajani"; // Replace with your MySQL username
        String password = "IshanKRajani@1234"; // Replace with your MySQL
password

        // SQL query to insert data
        String insertSQL = "INSERT INTO Employees (Name, Age, Position)
VALUES (?, ?, ?)";

        // Insert data into the table
        try (Connection conn = DriverManager.getConnection(url, user,
password);
             PreparedStatement pstmt = conn.prepareStatement(insertSQL)) {

            // Set the values for the PreparedStatement
            pstmt.setString(1, "Dev Thakkar");
            pstmt.setInt(2, 30);
            pstmt.setString(3, "Software Engineer");

            // Execute the insert operation
            int rowsAffected = pstmt.executeUpdate();
            System.out.println(rowsAffected + " row(s) inserted.");

            // Inserting another record as an example
            pstmt.setString(1, "Harshil");
            pstmt.setInt(2, 25);
            pstmt.setString(3, "Data Analyst");

            rowsAffected = pstmt.executeUpdate();
            System.out.println(rowsAffected + " row(s) inserted.");

        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
}
```

Step-by-step explanation of the code:

1. Import Statements

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.Statement;
import java.sql.SQLException;

- **`Connection`**: Represents a connection to the database. It is used to send SQL commands and manage communication with the database.
- **`DriverManager`**: This is used to establish a connection with the database by providing connection details (URL, username, password).
- **`Statement`**: Used to execute SQL queries (like creating tables or inserting data) on the database.
- **`SQLException`**: A class for handling SQL exceptions, which are thrown when there is an issue with database operations.

2. Class Declaration

public class INSERT_TABLE {

- This defines a public class called `INSERT_TABLE`, which is the main class of the program.

3. Main Method

public static void main(String[] args) {

- This is the entry point for the Java program. When the program is run, it starts executing from here.

4. Database Connection Details

```
String url = "jdbc:mysql://localhost:3306/firstdatabase"; // Replace with your database name
String user = "IshanKRajani"; // Replace with your MySQL username
String password = "IshanKRajani@1234"; // Replace with your MySQL password
```

- **`url`**: The database connection URL. It specifies the type of database (`jdbc:mysql`), the host (`localhost`), the port (`3306`), and the database name (`firstdatabase`).
- **`user`**: The username for accessing the MySQL database. In this case, it's set to `"IshanKRajani"`.
- **`password`**: The password for the specified MySQL username. It is set as `"IshanKRajani@1234"` here.

5. SQL Query for Creating the Table

```
String createTableSQL = "CREATE TABLE Employees ("
        + "ID INT PRIMARY KEY AUTO_INCREMENT, "
        + "Name VARCHAR(100), "
        + "Age INT, "
        + "Position VARCHAR(100))";
```

- **`createTableSQL`**: This is a string that contains the SQL query to create a table named **`Employees`**.
    - The table has four columns:
        - **`ID`**: An integer that is the primary key of the table and automatically increments with each new record.
        - **`Name`**: A variable character field (up to 100 characters) for the employee's name.
        - **`Age`**: An integer field to store the employee's age.
        - **`Position`**: A variable character field (up to 100 characters) to store the employee's job position.

6. Database Connection and Statement Creation

```
try (Connection conn = DriverManager.getConnection(url, user, password);
     Statement stmt = conn.createStatement()) {
```

- **`Connection conn = DriverManager.getConnection(url, user, password)`**: This line creates a connection to the database using the `DriverManager.getConnection()` method. It connects to the database using the URL, username, and password provided earlier.
- **`Statement stmt = conn.createStatement()`**: This creates a `Statement` object that will be used to execute SQL queries.

7. Executing the SQL Query

```
stmt.executeUpdate(createTableSQL);
System.out.println("Table 'Employees' created successfully.");
```

- **`stmt.executeUpdate(createTableSQL)`**: This method executes the SQL query stored in `createTableSQL` to create the table in the database. `executeUpdate()` is used for SQL statements that modify the database (like `INSERT`, `UPDATE`, `DELETE`, or `CREATE`).
- **`System.out.println("Table 'Employees' created successfully.");`**: If the table is created successfully, a message is printed to the console.

8. Exception Handling

```
} catch (SQLException e) {
   e.printStackTrace();
}
```

- **`catch (SQLException e)`**: If any exception occurs during the execution (like issues with the database connection or SQL syntax), it is caught here.
- **`e.printStackTrace()`**: If an exception occurs, the details of the exception are printed to the console. This helps in debugging and understanding the cause of the error.

9. End of the Class

```
}
```

- This marks the end of the `main()` method and the class `INSERT_TABLE`.

What the Code Does:

- It connects to a MySQL database (`firstdatabase` on `localhost` with the username IshanKRajani and password IshanKRajani@1234).

- It creates a table named `Employees` in the database with columns for `ID`, `Name`, `Age`, and `Position`.
- If the table is created successfully, it prints a success message. If any error occurs, it prints the error details.

Potential Issues:

- If the database `firstdatabase` does not exist, the connection will fail.
- The table creation will fail if a table named `Employees` already exists.