Java – RDBMS & Database Programming with JDBC Assignment

Introduction to JDBC

QUES 1: What is JDBC (Java Database Connectivity)?
JDBC stands for java database connectivity. JDBC API is to connect and execute the query with the database .it is used to write programs required to access database.

QUES 2: Importance of JDBC in Java Programming.
JDBC is a java database API used for making connection between java application and various database, basically it is used for stable connection between database and application API. Multiple applications can connect to different types of database.

QUES 3: JDBC Architecture: Driver Manager, Driver, Connection, Statement, and ResultSet
DRIVER MANAGER: it plays an important role in the JDBC architecture. It uses some database specific driver to connect to enterprise application to database.
DRIVER: to communicate with data source through jdbc we need jdbc driver.
CONNECTION, STATEMENT AND RESULTSET is interfaces of jDBC API.

JDBC Driver Types

QUES 4: Overview of JDBC Driver
JDBC drivers are client side adapters that translate request from java program into a protocol understood by the DBMS. These drives are software components that implement the interface in the JDBC API, allowing java application to interact with a database.
Types:
Type 1: JDBC-ODBC Bridge Driver
The JDBC-ODBC bridge driver uses ODBC driver to connect to the database. The JDBC-ODBC bridge driver converts JDBC method calls into the ODBC function calls. This is now discouraged because of thin driver.

Type 2: Native-API Driver
The Native API driver uses the client-side libraries of the database. The driver converts JDBC method calls into native calls of the database API. It is not written entirely in java.

Type 3: Network Protocol Driver
The Network Protocol driver uses middleware (application server) that converts JDBC calls directly or indirectly into the vendor-specific database protocol. It is fully written in java.

Type 4: Thin Driver o Comparison and Usage of Each Driver Type
The thin driver converts JDBC calls directly into the vendor-specific database protocol. That is why it is known as thin driver. It is fully written in Java language.

Steps for Creating JDBC Connections
Step-by-Step Process to Establish a JDBC Connection:
1. Import the JDBC packages

2. Register the JDBC driver
Here we can load driver class into memory at runtime.
Class.forName("com.mysql.cj.jdbc.Driver");
3. Open a connection to the database
Establish a connection using connection class
Connection con = DriverManager.getConnection(url,user,password);
Con refer to connection interface
url is uniform resource locator which is created as ,
String url = "jdbc:mysql://localhost:3306/db";
User and password are from which SQL command prompt.
4. Create a statement
Statement smt = con.createStatement();
5. Execute SQL queries
String sql = insert into table name (id,name) values (101,xyz);
6. Process the result set
Int status = smt.excecuteUpdate(sql);
7. Close the connection
Con.close()

Types of JDBC Statements

QUES Overview of JDBC Statements:
Statement: Executes simple SQL queries without parameters.
Syntax : statement statement = connection.createStatement();

PreparedStatement: Precompiled SQL statements for queries with parameters.
IINSERT INTO info VALUES ("Aryan",30);
INSERT INTO info VALUES ("Kriva",25);

CallableStatement: Used to call stored procedures.
callableStatement cstmt = con.prepareCall("{callProcedureName(?,?)}");

JDBC CRUD Operations (Insert, Update, Select, Delete)

Insert: Adding a new record to the database.
Update: Modifying existing records.
Select: Retrieving records from the database.
Delete: Removing records from the database.

ResultSet Interface

QUES What is ResultSet in JDBC?
It is very important part of JDBC connection because it excecute the query.
ResultSet rs = stmt.executeQuery("SELECT * FROM your_table");

Navigating through ResultSet (first, last, next, previous)
We can navigate in different ways using above given methods
First() = used for move to first row in the resultset

Last () = used for move to last row in the resultset
Next() = used for move to next row in the resultset
Previous() = used for move to previous row in the resultset

Working with ResultSet to retrieve data from SQL queries
getInt(int columnIndex), getString, getDouble, getBoolean, getData, getObject
These methods retrieve data from the current row in the resultset. And also you can retrieve data by column index or column name.

7. Database Metadata

QUES : What is DatabaseMetaData?
Generally, Data about data is known as metadata. The DatabaseMetaData interface provides methods to get information about the database you have connected with like, database name, database driver version, maximum column length etc

QUES : Importance of Database Metadata in JDBC .
This metadata provides valuable insights into the structure of the database which is used for various advantages like generation of dynamic queries and validations of the data and schema introspection.

QUES: Methods provided by DatabaseMetaData (getDatabaseProductName, getTables, etc.)
Commonly used methods of DatabaseMetaData interface
public String getDriverName()throws SQLException: it returns the name of the JDBC driver.
public String getDriverVersion()throws SQLException: it returns the version number of the JDBC driver.
public String getUserName()throws SQLException: it returns the username of the database.
public String getDatabaseProductName()throws SQLException: it returns the product name of the database.
public String getDatabaseProductVersion()throws SQLException: it returns the product version of the database.
public ResultSet getTables(String catalog, String schemaPattern, String tableNamePattern, String[] types)throws SQLException: it returns the description of the tables of the specified catalog. The table type can be TABLE, VIEW, ALIAS, SYSTEM TABLE, SYNONYM etc.

8. ResultSet Metadata

QUES: What is ResultSetMetaData?
The getMetaData() method of the ResultSet interface retrieves the ResultSetMetaData object, which contains information about the types and properties of the columns in a ResultSet object

QUES: Importance of ResultSet Metadata in analyzing the structure of query results
The ResultSetMetaData provides information about the obtained ResultSet object like, the number of columns, names of the columns, datatypes of the columns, name of the table etc.

QUES: Methods in ResultSetMetaData (getColumnCount, getColumnName, getColumnType)
getColumnCount() = Retrieves the number of columns in the current ResultSet object
getColumnLabel() = Retrieves the suggested name of the column for use.

getColumnName() = Retrieves the name of the column.
getTableName() = Retrieves the name of the table.

10. Practical Example 1: Swing GUI for CRUD Operations

QUES: Introduction to Java Swing for GUI development
Java Swing is a part of Java Foundation Classes (JFC) that is used to create window-based applications. It is built on the top of AWT (Abstract Windowing Toolkit) API and entirely written in Java.

11. Practical Example 2:

QUES: What is a CallableStatement?
CallableStatement interface is used to call the stored procedures and functions.

QUES: How to call stored procedures using CallableStatement in JDBC .
Create a database connection
Create a SQl string
Create CallableStatement object (CallableStatement cs = con.prepareCall(sql_string);
Set the input parameters
Call stored procedure

QUES: Working with IN and OUT parameters in stored procedures
Procedure with IN parameter = An IN parameter is used to take a parameter as input such as attributes.
Procedure with OUT parameter = An out parameter is used to pass a parameter as output or display like the select operator, but implicity. The value of an OUT parameter can be change inside the procedure and its new value is passed back to the calling program.