JDBC - Interview Questions

Dear readers, these **JDBC Interview Questions** have been designed specially to get you acquainted with the nature of questions you may encounter during your interview for the subject of **JDBC**. As per my experience good interviewers hardly plan to ask any particular question during your interview, normally questions start with some basic concept of the subject and later they continue based on further discussion and what you answer:

What is JDBC?

JDBC stands for Java Database Connectivity, which is a standard Java API for database-independent connectivity between the Java programming language and a wide range of databases.

Describe a general JDBC Architecture.

General JDBC Architecture consists of two layers JDBC API (This provides the application-to-JDBC Manager connection) and JDBC Driver API (This supports the JDBC Manager-to-Driver Connection).

What are the common JDBC API components?

JDBC API consists of following interfaces and classes DriverManager, Driver, Connection, Statement, ResultSet, SQLException.

What is a JDBC DriverManager?

JDBC DriverManager is a class that manages a list of database drivers. It matches connection requests from the java application with the proper database driver using communication subprotocol.

What is a JDBC Driver?

JDBC driver is an interface enabling a Java application to interact with a database. To connect with individual databases, JDBC requires drivers for each database. The JDBC driver gives out the connection to the database and implements the protocol for transferring the query and result between client and database.

What is a connection?

Connection interface consists of methods for contacting a database. The connection object represents communication context.

What is a statement?

Statement encapsulates an SQL statement which is passed to the database to be parsed, compiled, planned and executed.

What is a ResultSet?

These objects hold data retrieved from a database after you execute an SQL query using Statement objects. It acts as an iterator to allow you to move through its data. The java.sql.ResultSet interface represents the result set of a database query.

What are types of ResultSet?

There are three constants which when defined in result set can move cursor in resultset backward, forward and also in a particular row.

- **ResultSet.TYPE_FORWARD_ONLY** − The cursor can only move forward in the result set.
- **ResultSet.TYPE_SCROLL_INSENSITIVE** − The cursor can scroll forwards and backwards, and the result set is not sensitive to changes made by others to the database that occur after the result set was created.
- **ResultSet.TYPE_SCROLL_SENSITIVE** − The cursor can scroll forwards and backwards, and the result set is sensitive to changes made by others to the database that occur after the result set was created.

What are the basic steps to create a JDBC application?

Following are the basic steps to create a JDBC application

- Import packages containing the JDBC classes needed for database programming.
- Register the JDBC driver, so that you can open a communications channel with the database.
- Open a connection using the DriverManager.getConnection () method.
- Execute a query using an object of type Statement.
- Extract data from result set using the appropriate ResultSet.getXXX () method.
- Clean up the environment by closing all database resources relying on the JVM's garbage collection.

What are JDBC driver types?

There are four types of JDBC drivers

- **JDBC-ODBC Bridge plus ODBC driver** − also called Type 1 calls native code of the locally available ODBC driver.
- **Native-API, partly Java driver** − also called Type 2 calls database vendor native library on a client side. This code then talks to database over network.
- **JDBC-Net, pure Java driver** − also called Type 3 the pure-java driver that talks with the server-side middleware that then talks to database.
- **Native-protocol, pure Java driver** − also called Type 4 the pure-java driver that uses database native protocol.

When should each of the JDBC driver type be used?

Following is a list as to when the four types of drivers can be used

- If you are accessing one type of database, such as Oracle, Sybase, or IBM, the preferred driver type is 4.
- If your Java application is accessing multiple types of databases at the same time, type 3 is the preferred driver.
- Type 2 drivers are useful in situations where a type 3 or type 4 driver is not available yet for your database.
- The type 1 driver is not considered a deployment-level driver and is typically used for development and testing purposes only.

Which type of JDBC driver is the fastest one?

JDBC Net pure Java driver(Type 4) is the fastest driver because it converts the JDBC calls into vendor specific protocol calls and it directly interacts with the database.

Does the JDBC-ODBC Bridge support multiple concurrent open statements per connection?

No. You can open only one Statement object per connection when you are using the JDBC-ODBC Bridge.

What are the standard isolation levels defined by JDBC?

The standard isolation levels are

- **TRANSACTION_NONE**
- **TRANSACTION_READ_COMMITTED**

- **TRANSACTION_READ_UNCOMMITTED**
- **TRANSACTION_REPEATABLE_READ**
- **TRANSACTION_SERIALIZABLE**

What is the design pattern followed by JDBC?

JDBC architecture decouples an abstraction from its implementation. Hence JDBC follows a bridge design pattern. The JDBC API provides the abstraction and the JDBC drivers provide the implementation. New drivers can be plugged-in to the JDBC API without changing the client code.

What are the different types of JDBC Statements?

Types of statements are

- **Statement** − regular SQL statement.
- **PreparedStatement** − more efficient than statement due to pre-compilation of SQL.
- **CallableStatement** − to call stored procedures on the database.

What is difference between statement and prepared statement?

Prepared statements offer better performance, as they are pre-compiled. Prepared statements reuse the same execution plan for different arguments rather than creating a new execution plan every time. Prepared statements use bind arguments, which are sent to the database engine. This allows mapping different requests with same prepared statement but different arguments to execute the same execution plan. Prepared statements are more secure because they use bind variables, which can prevent SQL injection attack.

How do you register a driver?There are 2 approaches for registering the Driver

- **Class.forName()** − This method dynamically loads the driver's class file into memory, which automatically registers it. This method is preferable because it allows you to make the driver registration configurable and portable.
- **DriverManager.registerDriver()** − This static method is used in case you are using a non-JDK compliant JVM, such as the one provided by Microsoft.

What are the benefits of JDBC 4.0?

Here are few advantages of JDBC 4.0

- Auto loading of JDBC driver class. In the earlier versions we had to manually register and load drivers using class.forName.

- Connection management enhancements. New methods added to javax.sql.PooledConnection.
- DataSet Implementation of SQL using annotations.
- SQL XML support.

What do you mean by fastest type of JDBC driver?

JDBC driver performance or fastness depends on a number of issues Quality of the driver code, size of the driver code, database server and its load, Network topology, Number of times your request is translated to a different API.

In real time project which driver did you use?

Tell about your real time experience.

How do you create a connection object?

There are 3 overloaded DriverManager.getConnection() methods to create a connection object

getConnection(String url, String user, String password)Using a database URL with a username and password. For example

```
String URL = "jdbcoraclethin@amrood1521EMP";
String USER = "username";
String PASS = "password"
Connection conn = DriverManager.getConnection(URL, USER, PASS);
getConnection(String url)Using only a database URL. For example
```

```
String URL = "jdbcoraclethinusername/password@amrood1521EMP";
Connection conn = DriverManager.getConnection(URL);
getConnection(String url, Properties prop)Using a database URL and a Properties object
```

```
String URL = "jdbcoraclethin@amrood1521EMP";
Properties info = new Properties( );
info.put( "user", "username" );
info.put( "password", "password" );
```

How can I determine whether a Statement and its ResultSet will be closed on a commit or rollback?

Use the DatabaseMetaData methods supportsOpenStatementsAcrossCommit() and supportsOpenStatementsAcrossRollback() to check.

Is there a practical limit for the number of SQL statements that can be added to an instance of a Statement object?

The specification makes no mention of any size limitation for Statement.addBatch(), this is dependent, on the driver.

How cursor works in scrollable result set?

There are several methods in the ResultSet interface that involve moving the cursor, like beforeFirst(), afterLast(), first(), last(), absolute(int row), relative(int row), previous(), next(), getRow(), moveToInsertRow(), moveToCurrentRow().

How can you view a result set?

ResultSet interface contains get methods for each of the possible data types, and each get method has two versions

- One that takes in a column name.
- One that takes in a column index.

For e.g. getInt(String columnName), getInt(int columnIndex)

How do you update a result set?

ResultSet interface contains a collection of update methods for updating the data of a result set. Each update method has two versions for each data type

- One that takes in a column name.
- One that takes in a column index.

These methods change the columns of the current row in the ResultSet object, but not in the underlying database. To update your changes to the row in the database, you need to invoke one of the following methods

updateRow(), deleteRow(), refreshRow(), cancelRowUpdates(), insertRow()

How does JDBC handle the data types of Java and database?

The JDBC driver converts the Java data type to the appropriate JDBC type before sending it to the database. It uses a default mapping for most data types. For example, a Java int is converted to an SQL INTEGER.

What causes "No suitable driver" error?

"No suitable driver" is occurs during a call to the DriverManager.getConnection method, may be of any of the following reason

- Due to failing to load the appropriate JDBC drivers before calling the getConnection method.
- It can be specifying an invalid JDBC URL, one that is not recognized by JDBC driver.
- This error can occur if one or more the shared libraries needed by the bridge cannot be loaded.

How do you handle SQL NULL values in Java?

SQL's use of NULL values and Java's use of null are different concepts. There are three tactics you can use

- Avoid using getXXX( ) methods that return primitive data types.
- Use wrapper classes for primitive data types, and use the ResultSet object's wasNull( ) method to test whether the wrapper class variable that received the value returned by the getXXX( ) method should be set to null.
- Use primitive data types and the ResultSet object's wasNull( ) method to test whether the primitive variable that received the value returned by the getXXX( ) method should be set to an acceptable value that you've chosen to represent a NULL.

What does setAutoCommit do?

When a connection is created, it is in auto-commit mode. This means that each individual SQL statement is treated as a transaction and will be automatically committed right after it is executed. By setting auto-commit to false no SQL statements will be committed until you explicitly call the commit method.

Why will you set auto commit mode to false?

Following are the reasons

- To increase performance.
- To maintain the integrity of business processes.
- To use distributed transactions.

What is SavePoint? Give an example.

A savepoint marks a point that the current transaction can roll back to. Instead of rolling all of its changes back, it can choose to roll back only some of them. For example, suppose you

- start a transaction.
- insert 10 rows into a table.
- set a savepoint.
- insert another 5 rows.
- rollback to the savepoint.
- commit the transaction.

After doing this, the table will contain the first 10 rows you inserted. The other 5 rows will have been deleted by the rollback. A savepoint is just a marker that the current transaction can roll back to.

What are SQL warnings?

SQLWarning objects are a subclass of SQLException that deal with database access warnings. Warnings do not stop the execution of an application, as exceptions do. They simply alert the user that something did not happen as planned. A warning can be reported on a Connection object, a Statement object (including PreparedStatement and CallableStatement objects), or a ResultSet object. Each of these classes has a getWarnings method.

Why would you use a batch process?

Batch Processing allows you to group related SQL statements into a batch and submit them with one call to the database.

What are the steps followed to create a batch process?

Typical sequences of steps to use Batch Processing with Statement or PrepareStatement Object are

- In case of Batch processing using PrepareStatement object, create SQL statements with placeholders.
- Create a Statement or PrepareStatement object using either createStatement() or prepareStatement() methods respectively.
- Set auto-commit to false using setAutoCommit().
- Add as many as SQL statements you like into batch using addBatch() method on created statement object.
- Execute all the SQL statements using executeBatch() method on created statement object.
- Finally, commit all the changes using commit() method.

What is a Stored Procedure and how do you call it in JDBC?

A stored procedure is a group of SQL statements that form a logical unit and perform a particular task. For example operations on an employee database (hire, fire, promote, lookup) could be coded as stored procedures executed by application code. Stored procedures can be called using CallableStatement class in JDBC API. For example the following code demonstrates this

```
CallableStatement cs = con.prepareCall("{call MY_SAMPLE_STORED_PROC}");
ResultSet rs = cs.executeQuery();
```

What is JDBC SQL escape syntax?

The escape syntax gives you the flexibility to use database specific features unavailable to you by using standard JDBC methods and properties.

The general SQL escape syntax format is as follows

```
{keyword 'parameters'}.
```

JDBC defines escape sequences that contain the standard syntax for the following language features

- Date, time, and timestamp literals (d, t, ts Keywords).
- Scalar functions such as numeric, string, and data type conversion functions(fn Keyword).
- Outer joins(oj Keyword)
- Escape characters for wildcards used in LIKE clauses(escape Keyword).
- Procedure calls(call Keyword).

What is a transaction?

A transaction is a logical unit of work. To complete a logical unit of work, several actions may need to be taken against a database. Transactions are used to provide data integrity, correct application semantics, and a consistent view of data during concurrent access.

How will you insert multiple rows into a database in a single transaction?

Follow steps as below

```
//turn off the implicit commit
Connection.setAutoCommit(false);
//..your insert/update/delete goes here
```

```
Connection.Commit();
//a new transaction is implicitly started.
```

When will you get the message "No Suitable Driver"?

When a Connection request is issued, the DriverManager asks each loaded driver if it understands the URL sent. When the URL passed is not properly constructed, then the "No Suitable Driver" message is returned.

What is the difference between execute, executeQuery, executeUpdate?

- **boolean execute()** - Executes the any kind of SQL statement.
- **ResultSet executeQuery()** - This is used generally for reading the content of the database. The output will be in the form of ResultSet. Generally SELECT statement is used.
- **int executeUpdate()** - This is generally used for altering the databases. Generally DROP TABLE or DATABASE, INSERT into TABLE, UPDATE TABLE, DELETE from TABLE statements will be used in this. The output will be in the form of int which denotes the number of rows affected by the query.

Why do you have to close database connections in Java?

You need to close the resultset, the statement and the connection. If the connection has come from a pool, closing it actually sends it back to the pool for reuse. We can do this in the finally{} block, such that if an exception is thrown, you still get the chance to close this.

What is the use of blob, clob datatypes in JDBC?

These are used to store large amount of data into database like images, movie etc which are extremely large in size.

Resultset is an interface, how does it support rs.Next()?

Every vendor of Database provides implementation of ResultSet & other interfaces, through the Driver.

What is Connection Pooling?

Connection Pooling is a technique used for reuse of physical connections and reduced overhead for your application. Connection pooling functionality minimizes expensive operations in the creation and closing of sessions.Database vendor's help multiple clients to share a cached set of

connection objects that provides access to a database. Clients need not create a new connection everytime to interact with the database.

How do you implement connection pooling?

If you use an application server like WebLogic, WebSphere, jBoss, Tomcat. , then your application server provides the facilities to configure for connection pooling. If you are not using an application server then components like Apache Commons DBCP Component can be used.

Out of byte[] or a java.sql.Blob, which has best performance when used to manipulate data from database?

java.sql.Blob has better performance as it does not extract any data from the database until you explicitly ask it to.

Out of String or a java.sql.Clob, which has best performance when used to manipulate data from database?

java.sql.Clob has better performance as it does not extract any data from the database until you explicitly ask it to.

Suppose the SELECT returns 1000 rows, then how to retrieve the first 100 rows, then go back and retrieve the next 100 rows?

Use the Statement.setFetchSize method to indicate the size of each database fetch.

What does the Class.forName("MyClass") do?

Class.forName("MyClass")

- Loads the class MyClass.
- Execute any static block code of MyClass.
- Returns an instance of MyClass.

When you say Class.forName() loads the driver class, does it mean it imports the driver class using import statement?

No, it doesn't. An import statement tells the compiler which class to look for. Class.forName() instructs the Classclass to find a class-loader and load that particular Class object into the memory used by the JVM.

What we set the attribute Concurrency in ResultSet?

The ResultSet concurrency determines whether the ResultSet can be updated, or only read. A ResultSet can have one of two concurrency levels

- **ResultSet.CONCUR_READ_ONLY** − means that the ResultSet can only be read.
- **ResultSet.CONCUR_UPDATABLE** − means that the ResultSet can be both read and updated.

What are the differences between setMaxRows(int) and SetFetchSize(int)?

The difference between setFetchSize(int) and setMaxRow(int) are

- setFetchSize(int) defines the number of rows that will be read from the database when the ResultSet needs more rows. setFetchSize(int) affects how the database returns the ResultSet data.
- setMaxRows(int) method of the ResultSet specifies how many rows a ResultSet can contain at a time. setMaxRows(int) affects the client side JDBC object.

What is a RowSet?

A JDBC RowSet object holds tabular data in a way that makes it more flexible and easier to use than a result set. A RowSet objects are JavaBeans components.

What are different types of RowSet objects?

There are two types of RowSet

- **Connected** A connected RowSet Object is permanent in nature. It doesn't terminate until the application is terminated.
- **Disconnected** A disconnected RowSet object is ad-hoc in nature. Whenever it requires retrieving data from the database, it establishes the connection and closes it upon finishing the required task. The data that is modified during disconnected state is updated after the connection is re-established.

What is a "dirty read"?

In typical database transactions, say one transaction reads and changes the value while the second transaction reads the value before committing or rolling back by the first transaction. This reading process is called as 'dirty read'. Because there is always a chance that the first transaction might rollback the change which causes the second transaction reads an invalid value.

Which isolation level prevents dirty read in JDBC, connection class?

TRANSACTION_READ_COMMITTED prevents dirty reads.

What is Metadata and why should you use it?

JDBC API has two Metadata interfaces DatabaseMetaData & ResultSetMetaData. The meta data provides comprehensive information about the database as a whole. The implementation for these interfaces is implemented by database driver vendors to let users know the capabilities of a Database.

How to Connect to an Excel Spreadsheet using JDBC in Java?

Follow the steps below

First setup the new ODBC datasource. Goto Administrative Tools−>Data Sources (ODBC) −>System DSN tab−>Add−>Driver do Microsoft Excel(*.xls)−>Finish. Now give the Data Source Name (SampleExcel) & Description. Next, click Select Workbook and point to your excel sheet.

In the code make to following code additions

```
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
Connection conn = DriverManager.getConnection("jdbcodbcSampleExcel","","");
stmt = conn.createStatement();
sql = "select * from [Sheet1$]";
rs=stmt.executeQuery(sql);
```

Where Sheet1 is the excel sheet name.

What is difference between JDBC, JNDI and Hibernate?

- Hibernate is an Object−Relational Mapping tool. It maps Objects to relational data.
- The Java Naming and Directory Interface (JNDI) is an API to access different naming and directory services. You use it to access something stored in a directory or naming service without haveing to code specifically to that naming or directory service.
- Java DataBase Connectivity (JDBC) API is an API to access different relational databases. You use it to access relational databases without embedding a dependency on a specific database type in your code.

What is Next ?

Further you can go through your past assignments you have done with the subject and make sure you are able to speak confidently on them. If you are fresher then interviewer does not expect you will answer very complex questions, rather you have to make your basics concepts very strong.

Second it really doesn't matter much if you could not answer few questions but it matters that whatever you answered, you must have answered with confidence. So just feel confident during your interview. We at tutorialspoint wish you best luck to have a good interviewer and all the very best for your future endeavor. Cheers :-)