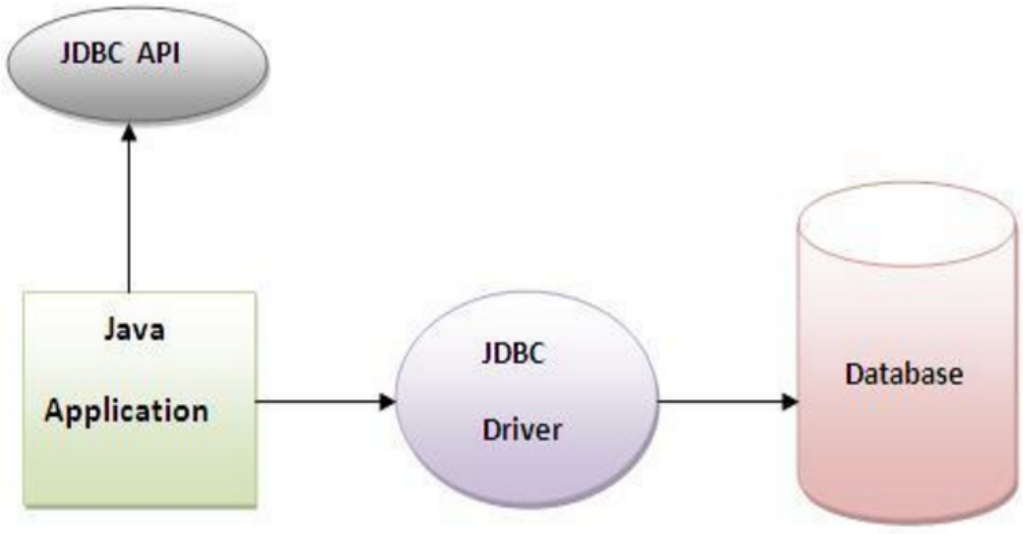
JDBC Interview Question Answer

What is Java JDBC ?

Java JDBC is a java API to connect and execute query with the database. JDBC API uses jdbc drivers to connect with the database.



What are the types of statements in JDBC?

JDBC API has 3 Interfaces and their key features of these are as follows:

Statement: which is used to run simple SQL statements like select and update. Statement interfaces use for general-purpose access to your database. It is useful when you are using static SQL statements at runtime. The Statement interface cannot accept parameters.

Prepared Statement: A SQL statement is pre-compiled and stored in a Prepared Statement object. It is used to run Pre compiled SQL. This object can then be used to efficiently execute this statement multiple times. The object of Prepared Statement class can be created using Connection.prepareStatement() method. This extends Statement interface.

Callable Statement: This interface is used to execute the stored procedures. This extends Prepared Statement interface. The object of Callable Statement class can be created using Connection.prepareCall() method.

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.

Why use JDBC ?

Before JDBC, ODBC API was the database API to connect and execute query with the database. But, ODBC API uses ODBC driver which is written in C language (i.e. platform dependent and unsecured). That is why Java has defined its own API (JDBC API) that uses JDBC drivers (written in Java language).

What restrictions are placed on method overriding?

The restriction on method overloading is the signature of the method.
• The signature is the number, type, and order of the arguments passed to a method.
• Overridden methods must have the same name, argument list, and return type.
• Any method which has the same name cannot have the same signature.
• They can have the same return types in the same scope.
• The compiler uses the signature to detect which overloaded method to refer when a overloaded method is called.
• If two methods have the same name and signature the compiler will throw a runtime error.

What does setAutoCommit do?

setAutoCommit() invoke the commit state query to the database. To perform batch updation we use the setAutoCommit() which enable us to execute more than one statement together, which in result minimize the database call and send all statement in one batch.

setAutoCommit() allowed us to commit the transaction commit state manually the default values of the setAutoCommit() is true.

What are types of JDBC drivers?

There are four types of drivers defined by JDBC as follows:

• JDBC/ODBC These require an ODBC (Open Database Connectivity) driver for the database to be installed. It is used for local connection.

• Native API (partly-Java driver): This type of driver uses a database API to interact with the database. It also provides no host redirection.

• Network Protocol Driver: It makes use of a middle-tier between the calling program and the database. The client driver communicates with the net server using a database-independent protocol and the net server translates this protocol into database calls.

• Native Protocol Drive: This has a same configuration as a type 3 driver but uses a

wire protocol specific to a particular vendor and hence can access only that vendor's database.

Why Prepared Statements are faster?

Prepared execution is faster than direct execution for statements executed more than three or four times because the statement is compiled only once. Prepared statements and JDBC driver are linked with each other. We can bind drivers with columns by triggering the query into the database. When we execute Connection.prepareStatement(), all the columns bindings take place, in oder to reduce the time.

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

The difference between setFetchSize and setMaxRow are:

• setFetchSize(int) defines the number of rows that will be read from the database when the ResultSet needs more rows whereas setMaxRows(int) method of the ResultSet specifies how many rows a ResultSet can contain at a time.

• In setFetchSize(int), method in the java.sql.Statement interface will set the 'default' value for all the ResultSet derived from that Statement whereas in setMaxRow(int) default value is 0, i.e. all rows will be included in the ResultSet.

• the setMaxRows affects the client side JDBC object while the setFetchSize affects how the database returns the ResultSet data.

How can I manage special characters when I execute an INSERT query?

The special characters meaning in SQL can be preceded with a special escape character in strings, e.g. "\". In order to specify the escape character used to quote these characters, include the following syntax on the end of the query: {escape 'escape-character'}

For example, the query

SELECT NAME FROM IDENTIFIERS WHERE ID LIKE '_%' {escape '\'}
finds identifier names that begin with an underscore.

How can you load the drivers?

It is very simple and involves just one line of code to load the driver or drivers we want to use.
For example, We want to use the JDBC-ODBC Bridge driver, the following code will load it:
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
Driver documentation will give you the class name to use. For instance, if the class name is jdbc.DriverHELLO, you would load the driver with the following line of code:
Class.forName("jdbc.DriverHELLO");

How do I insert an image file (or other raw data) into a database?

All raw data types should be read and uploaded to the database as an array of bytes, byte[].

- Originating from a binary file.
- Read all data from the file using a FileInputStream.
- Create a byte array from the read data.
- Use method setBytes(int index, byte[] data); of java.sql.PreparedStatement to upload the data.

Discuss the significances of JDBC

The significances are given below:

- JDBC is the acronym stands for Java Database Connectivity.
- Java Database Connectivity (JDBC) is a standard Java API .
- It's purpose is to interact with the relational databases in Java.
- JDBC is having a set of classes & interfaces which can be used from any Java application.
- By using the Database Specific JDBC drivers, it interacts with a database without the applications of RDBMS.

ⓓ Name the new features added in JDBC 4.0.

The major features introduced in JDBC 4.0 are :

• Auto-loading by JDBC driver class.
• Enhanced Connection management
• RowId SQL enabled.
• DataSet implemented by SQL by using Annotations
• Enhancements of SQL exception handling
• Supporting SQL XML files.

ⓓ Is it possible to connect to multiple databases simultaneously? Using single statement can one update or extract data from multiple databases?

Yes, it is possible but it depends upon the capabilities of the specific driver implementation, we can connect to multiple databases at the same time. We doing following steps:

• Minimum one driver will be used to handle the commits transaction for multiple connections.
• To update and extract data from the different database we use single statement for this we need special middleware to deal with multiple databases in a single statement or to effectively treat them as one database.

ⓓ What is the benefit of having JdbcRowSet implementation? Why do we need a JdbcRowSet like wrapper around ResultSet?

The JdbcRowSet implementation is a wrapper around a ResultSet object has following advantages over ResultSet:
• It makes possible to use the ResultSet object as a JavaBeans component.
• A JdbcRowSet can be used as a JavaBeans component, thus it can be created and configured at design time and executed at run time.
• It can be used to make a ResultSet object scrollable and updatable. All RowSet objects are by default scrollable and updatable.

ⓓ I have the choice of manipulating database data using a byte[] or a java.sql.Blob. Which has best performance?

We use java.sql.Blob, because of following reason:
• It does not extract any data from the database until we trigger a query to the databse.
• use byte[] for inserting data in the database when data is not upload in the database till yet.
• java.sql.Blob is used when extraction of the data is performed.

How do Java applications access the database using JDBC?

Java applications access the database using JDBC by :

• Communicating with the database for Loading the RDBMS specific JDBC driver
• Opening the connection with database
• Sending the SQL statements and get the results back.
• Creating JDBC Statement object which contains SQL query.
• Executing statement to return the resultset(s) containing the tuples of database table which is a result of SQL query.
• Processing the result set.
• Closing the connection..

Explain the life cycle of JDBC.

The life cycle for a servlet comprises of the following phases:
• DriverManager : for managing a list of database drivers.
• Driver : for communicating with the database.
• Connection : for interfacing with all the methods for connecting a database.
• Statement : for encapsulating an SQL statement for passing to the database which had been parsed, compiled, planned and executed.
• ResultSet: for representing a set of rows retrieved for the query execution.

Describe how the JDBC application works.

A JDBC application may be divided into two layers:

• Driver layer
• Application layer
• The Driver layer consists of DriverManager class & the JDBC drivers.

• The Application layer begins after putting a request to the DriverManager for the connection.
• An appropriate driver is chosen and used for establishing the connection.
• This connection is linked to the application layer.
• The application needs the connection for creating the Statement kind of objects by which the results are obtained.

Briefly tell about the JDBC Architecture.

The JDBC Architecture consists of two layers:

1.The JDBC API
2.The JDBC Driver API

• The JDBC API provides the application-JDBC Manager connection.
• The JDBC Driver API supports the JDBC Manager-to-Driver Connection.

• The JDBC API interacts with a driver manager, database-specific driver for providing transparent connectivity for the heterogeneous databases.
• The JDBC driver manager authenticates that the correct driver has been used to access each data source.
• The driver manager supports multiple concurrent drivers connected to the multiple heterogeneous databases.

What are DML and DDL?

Data Manipulation Language (DDL) this portion of the SQL standard is concerned with manipulating the data in a database as opposed to the structure of a database. The DML deals with the SELECT, INSERT, DELETE, UPDATE, COMMIT and ROLLBACK.

Data Definition Language (DDL) this portion of the SQL standard is concerned with the creation, deletion and modification of database objects like tables, indexes and views. The core verbs for DDL are CREATE, ALTER and DROP. While most DBMS engines allow DDL to be used dynamically, it is often not supported in transactions.

Explain Basic Steps in writing a Java program using JDBC.

JDBC makes the interaction with RDBMS simple and intuitive. When a Java application needs to access database :

• Load the RDBMS specific JDBC driver because this driver actually communicates with the database.
• Open the connection to database, for sending SQL statements and get results back.
• Create JDBC Statement object containing SQL query.
• Execute statement which returns result set. ResultSet contains the tuples of database table as a result of SQL query.
• Process the result set.
• Close the connection.

What does the JDBC Driver interface do?

• The JDBC Driver interface provides vendor-specific customized implementations of the abstract classes.

• It is provided normally by the JDBC API.
• For each vendor the driver provides implementations of the java.sql.Connection, , PreparedStatement, Driver,Statement, ResultSet and CallableStatement.

How a database driver can be loaded with JDBC 4.0 / Java 6?

• By providing the JAR file , the driver must be properly configured.
• The JAR file is placed in the classpath.
• It is not necessary to explicitly load the JDBC drivers by using the code like Class.forName() to register in the JDBC driver.
• The DriverManager class looks after this, via locating a suitable driver at the time when the DriverManager.getConnection() method is called.
• This feature provides backward-compatibility, so no change is needed in the existing JDBC code.

What is a Statement ?

• The Statement acts just like a vehicle via which SQL commands are sent.
• By the connection objects, we create the Statement kind of objects.
Statement stmt = conn.createStatement();
• This method returns the object, which implements the Statement interface.

What is represented by the connection object?

• The connection object represents the communication context
• All the communication with the database is executed via the connection objects only.
• Connection objects are used as the main linking elements.

.Define PreparedStatement.

• A Preparedstatement is an SQL statement which is precompiled by the database.
• By precompilation, the prepared statements improve the performance of the SQL commands that are executed multiple times (given that the database supports prepared statements).
• After compilation, prepared statements may be customized before every execution by the alteration of predefined SQL parameters.
Code:

```
PreparedStatement pstmt = conn.prepareStatement("UPDATE data= ? WHERE vl = ?");
pstmt.setBigDecimal(1, 1200.00);
pstmt.setInt(2, 192);
```

Differentiate between a Statement and a PreparedStatement.

• A standard Statement is used for creating a Java representation for a literal SQL statement and for executing it on the database.
• A PreparedStatement is a precompiled Statement.
• A Statement has to verify its metadata in the database every time.
• But ,the prepared statement has to verify its metadata in the database only once.
• If we execute the SQL statement, it will go to the STATEMENT.
• But, if we want to execute a single SQL statement for the multiple number of times, it'll go to the PreparedStatement.

What is the function of setAutoCommit?

• When a connection is created, it is in auto-commit mode.
• This means that each individual SQL statement is to be treated as a single transaction .
• The setAutoCommit will be automatically committed just after getting executed.
• The way by which two or more statements are clubbed into a transaction to disable the auto-commit mode is : con.setAutoCommit (false);
• Once auto-commit mode is disabled, no SQL statements will be committed until we call the method 'commit' explicitly.

Code :

```
con.setAutoCommit(false);
PreparedStatement updateSales = con.prepareStatement( "UPDATE COFFEE SALES = ? WHERE COF_NAME LIKE ?");
updateSales.setInt(1, 50); updateSales.setString(2, "Colombian");
updateSales.executeUpdate();
PreparedStatement updateTotal =
con.prepareStatement("UPDATE COFFEES SET TOTAL = TOTAL + ? WHERE COF_NAME LIKE ?");
updateTotal.setInt(1, 50);
updateTotal.setString(2, "Colombian");
updateTotal.executeUpdate();
con.commit();
con.setAutoCommit(true);
```