

JDBC Interview Questions

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

java.sql.Blob, since it does not extract any data from the database until you explicitly ask it to. The Java platform 2 type Blob wraps a database locator (which is essentially a pointer to byte). That pointer is a rather large number (between 32 and 256 bits in size) - but the effort to extract it from the database is insignificant next to extracting the full blob content. For insertion into the database, you should use a byte[] since data has not been uploaded to the database yet. Thus, use the Blob class only for extraction.

Conclusion: use the java.sql.Blob class for extraction whenever you can.

12) I have the choice of manipulating database data using a String or a java.sql.Clob. Which has best performance?

java.sql.Clob, since it does not extract any data from the database until you explicitly ask it to. The Java platform 2 type Clob wraps a database locator (which is essentially a pointer to char). That pointer is a rather large number (between 32 and 256 bits in size) - but the effort to extract it from the database is insignificant next to extracting the full Clob content. For insertion into the database, you should use a String since data need not be downloaded from the database. Thus, use the Clob class only for extraction.

Conclusion: Unless you always intend to extract the full textual data stored in the particular table cell, use the java.sql.Clob class for extraction whenever you can.

13) Do I need to commit after an INSERT call in JDBC or does JDBC do it automatically in the DB?

If your autoCommit flag (managed by the Connection.setAutoCommit method) is false, you are required to call the commit() method - and vice versa.

14) How can I retrieve only the first n rows, second n rows of a database using a particular WHERE clause ? For example, if a SELECT typically returns a 1000 rows, how do first retrieve the 100 rows, then go back and retrieve the next 100 rows and so on ?

Use the Statement.setFetchSize method to indicate the size of each database fetch. Note that this method is only available in the Java 2 platform. For Jdk 1.1.X and Jdk 1.0.X, no standardized way of setting the fetch size exists. Please consult the Db driver manual.

15) What does ResultSet actually contain? Is it the actual data of the result or some links to databases? If it is the

actual data then why can't we access it after connection is closed?

A ResultSet is an interface. Its implementation depends on the driver and hence ,what it "contains" depends partially on the driver and what the query returns.

For example with the Odbc bridge what the underlying implementation layer contains is an ODBC result set. A Type 4 driver executing a stored procedure that returns a cursor - on an oracle database it actually returns a cursor in the database. The oracle cursor can however be processed like a ResultSet would be from the client.

Closing a connection closes all interaction with the database and releases any locks that might have been obtained in the process.

16) What are SQL3 data types?

The next version of the ANSI/ISO SQL standard defines some new datatypes, commonly referred to as the SQL3 types. The primary SQL3 types are:

STRUCT: This is the default mapping for any SQL structured type, and is manifest by the java.sql.Struct type.

REF: Serves as a reference to SQL data within the database. Can be passed as a parameter to a SQL statement. Mapped to the java.sql.Ref type.

BLOB: Holds binary large objects. Mapped to the java.sql.Blob type.

CLOB: Contains character large objects. Mapped to the java.sql.Clob type.

ARRAY: Can store values of a specified type. Mapped to the java.sql.Array type.

You can retrieve, store and update SQL3 types using the corresponding getXXX(), setXXX(), and updateXXX() methods defined in ResultSet interface

17) How can I manage special characters (for example: " _ ' %) when I execute an INSERT query? If I don't filter the quoting marks or the apostrophe, for example, the SQL string will cause an error.

The characters "%" and "_" have special meaning in SQL LIKE clauses (to match zero or more characters, or exactly one character, respectively). In order to interpret them literally, they 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 underbar.

18) What is SQLJ and why would I want to use it instead of JDBC?

SQL/J is a technology, originally developed by Oracle Corporation, that enables you to embed SQL statements in Java. The purpose of the SQLJ API is to simplify the development

requirements of the JDBC API while doing the same thing. Some major databases (Oracle, Sybase) support SQLJ, but others do not. Currently, SQLJ has not been accepted as a standard, so if you have to learn one of the two technologies, I recommend JDBC.

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

All raw data types (including binary documents or images) should be read and uploaded to the database as an array of bytes, `byte[]`. Originating from a binary file,

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

20) How can I pool my database connections so I don't have to keep reconnecting to the database?

- you gets a reference to the pool
- you gets a free connection from the pool
- you performs your different tasks
- you frees the connection to the pool

Since your application retrieves a pooled connection, you don't consume your time to connect / disconnect from your data source.

21) Will a call to `PreparedStatement.executeQuery()` always close the `ResultSet` from the previous `executeQuery()`?

A `ResultSet` is automatically closed by the Statement that generated it when that Statement is closed, re-executed, or is used to retrieve the next result from a sequence of multiple results.

22) How do I upload SQL3 BLOB & CLOB data to a database?

Although one may simply extract BLOB & CLOB data from the database using the methods of the `java.sql.CLOB` and `java.sql.BLOB`, one must upload the data as normal java datatypes. The example below inserts a BLOB in the form of a `byte[]` and a CLOB in the form of a `String` into the database

```
Inserting SQL3 type data [BLOB & CLOB]
private void runInsert() {
    try {
        // Log
        this.log("Inserting values ... ");

        // Open a new Statement
        PreparedStatement stmtnt = conn.prepareStatement(
```

```

        "insert Lobtest (image, name) values (?, ?)");

// Create a timestamp to measure the insert time
Date before = new java.util.Date();

for(int i = 0; i < 500; i++) {
    // Set parameters
    stmt.setBytes(1, blobData);
    stmt.setString(2, "i: " + i + ";" + clobData);

    // Perform insert
    int rowsAffected = stmt.executeUpdate();
}

// Get another timestamp to complete the time measurement
Date after = new java.util.Date();
this.log(" ... Done!");
log("Total run time: " + (
    after.getTime() - before.getTime()));
// Close database resources
stmt.close();
} catch(SQLException ex) {
    this.log("Hmm... " + ex);
}
}

```

23) What is the difference between client and server database cursors?

What you see on the client side is the current row of the cursor which called a Result (ODBC) or ResultSet (JDBC). The cursor is a server-side entity only and remains on the server side.

24) Are prepared statements faster because they are compiled? if so, where and when are they compiled?

Prepared Statements aren't actually compiled, but they are bound by the JDBC driver. Depending on the driver, Prepared Statements can be a lot faster - if you re-use them. Some drivers bind the columns you request in the SQL statement. When you execute `Connection.prepareStatement()`, all the columns bindings take place, so the binding overhead does not occur each time you run the Prepared Statement. For additional information on Prepared Statement performance and binding see [JDBC Performance Tips](#) on IBM's website.

25) Is it possible to connect to multiple databases simultaneously? Can one extract/update data from multiple databases with a single statement?

In general, subject, as usual, to the capabilities of the specific driver implementation, one can connect to multiple databases at the same time. At least one driver (and probably others) will also handle commits across multiple connections. Obviously one should check the driver documentation rather than assuming these capabilities.

As to the second part of the question, one needs special middleware to deal with multiple databases in a single statement or to effectively treat them as one database. DRDA (Distributed Relational Database Architecture -- I, at least, make it rhyme with "Gerta") is probably most commonly used to accomplish this.

[Oracle](#) has a product called Oracle Transparent Gateway for IBM DRDA and [IBM](#) has a product called DataJoiner that make multiple databases appear as one to your application. No doubt there are other products available. [XOpen](#) also has papers available regarding DRDA.

26) Why do I get an UnsupportedOperationException?

JDBC 2.0, introduced with the 1.2 version of Java, added several capabilities to JDBC. Instead of completely invalidating all the older JDBC 1.x drivers, when you try to perform a 2.0 task with a 1.x driver, an UnsupportedOperationException will be thrown. You need to update your driver if you wish to use the new capabilities.

27) What advantage is there to using prepared statements if I am using connection pooling or closing the connection frequently to avoid resource/connection/cursor limitations?

The ability to choose the 'best' efficiency (or evaluate tradeoffs, if you prefer,) is, at times, the most important piece of a mature developer's skillset. This is YAA (Yet Another Area,) where that maxim applies. Apparently there is an effort to allow prepared statements to work 'better' with connection pools in JDBC 3.0, but for now, one loses most of the original benefit of prepared statements when the connection is closed. A prepared statement obviously fits best when a statement differing only in variable criteria is executed over and over without closing the statement.

However, depending on the DB engine, the SQL may be cached and reused even for a different prepared statement and most of the work is done by the DB engine rather than the driver. In addition, prepared statements deal with data conversions that can be error prone in straight ahead, built on the fly SQL; handling quotes and dates in a manner transparent to the developer, for example.

28) What is JDBC, anyhow?

JDBC is Java's means of dynamically accessing tabular data, and primarily data in relational databases, in a generic manner, normally using standard SQL statements.

29) Can I reuse a Statement or must I create a new one for each query?

When using a JDBC compliant driver, you can use the same Statement for any number of queries. However, some older drivers did not always "respect the spec." Also note that a Statement SHOULD automatically close the current ResultSet before executing a new query, so be sure you are done with it before re-querying using the same Statement.

30) What is a three-tier architecture?

A three-tier architecture is any system which enforces a general separation between the following three parts:

1. **Client Tier** or user interface
2. **Middle Tier** or business logic
3. **Data Storage Tier**

Applied to web applications and distributed programming, the three logical tiers usually correspond to the physical separation between three types of devices or hosts:

31) What separates one tier from another in the context of n-tiered architecture?

It depends on the application.

In a web application, for example, where tier 1 is a web-server, it may communicate with a tier 2 Application Server using RMI over IIOP, and subsequently tier 2 may communicate with tier 3 (data storage) using JDBC, etc.

Each of these tiers may be on separate physical machines or they may share the same box. The important thing is the functionality at each tier.

- Tier 1 - Presentation - should be concerned mainly with display of user interfaces and/or data to the client browser or client system.
- Tier 2 - Application - should be concerned with business logic

Tier 3+ - Storage/Enterprise Systems - should be focused on data persistence and/or communication with other Enterprise Systems.

32) What areas should I focus on for the best performance in a JDBC application?

These are few points to consider:

- Use a connection pool mechanism whenever possible.
- Use prepared statements. These can be beneficial, for example with DB specific escaping, even when used only once.
- Use stored procedures when they can be created in a standard manner. Do watch out for DB specific SP definitions that can cause migration headaches.
- Even though the jdbc promotes portability, true portability comes from NOT depending on any database specific data types, functions and so on.
- Select only required columns rather than using select * from Tablexyz.
- Always close Statement and ResultSet objects as soon as possible.
- Write modular classes to handle database interaction specifics.
- Work with DatabaseMetaData to get information about database functionality.
- Softcode database specific parameters with, for example, properties files.

- Always catch AND handle database warnings and exceptions. Be sure to check for additional pending exceptions.
- Test your code with debug statements to determine the time it takes to execute your query and so on to help in tuning your code. Also use query plan functionality if available.
- Use proper (and a single standard if possible) formats, especially for dates.
- Use proper data types for specific kind of data. For example, store birthdate as a date type rather than, say, varchar.

33) How can I insert multiple rows into a database in a single transaction?

```
//turn off the implicit commit

Connection.setAutoCommit(false);
//...your insert/update/delete goes here
Connection.Commit();
a new transaction is implicitly started.
```

34) How do I convert a java.sql.Timestamp to a java.util.Date?

While Timestamp extends Date, it stores the fractional part of the time within itself instead of within the Date superclass. If you need the partial seconds, you have to add them back in.

```
Date date = new Date(ts.getTime() + (ts.getNanos() / 1000000 ));
```

35) What is SQL?

SQL is a standardized language used to create, manipulate, examine, and manage relational databases.

36) Is Class.forName(Drivername) the only way to load a driver? Can I instantiate the Driver and use the object of the driver?

Yes, you can use the driver directly. Create an instance of the driver and use the connect method from the [Driver](#) interface. Note that there may actually be two instances created, due to the expected standard behavior of drivers when the class is loaded.

37) What's new in JDBC 3.0?

Probably the new features of most interest are:

- Savepoint support
- Reuse of prepared statements by connection pools
- Retrieval of auto-generated keys

- Ability to have multiple open ResultSet objects
- Ability to make internal updates to the data in Blob and Clob objects
- Ability to Update columns containing BLOB, CLOB, ARRAY and REF types
- Both java.sql and javax.sql (JDBC 2.0 Optional Package) are expected to be included with J2SE 1.4.

38) Why do I get the message "No Suitable Driver"?

Often the answer is given that the correct driver is not loaded. This may be the case, but more typically, the JDBC database URL passed is not properly constructed. When a Connection request is issued, the DriverManager asks each loaded driver if it understands the URL sent. If no driver responds that it understands the URL, then the "No Suitable Driver" message is returned.

39) When I create multiple Statements on my Connection, only the current Statement appears to be executed. What's the problem?

All JDBC objects are required to be threadsafe. Some drivers, unfortunately, implement this requirement by processing Statements serially. This means that additional Statements are not executed until the preceding Statement is completed.

40) Can a single thread open up multiple connections simultaneously for the same database and for same table?

The general answer to this is yes. If that were not true, connection pools, for example, would not be possible. As always, however, this is completely dependent on the JDBC driver.

You can find out the theoretical maximum number of active Connections that your driver can obtain via the DatabaseMetaData.getMaxConnections method.

41) Can I ensure that my app has the latest data?

Typically an application retrieves multiple rows of data, providing a snapshot at an instant of time. Before a particular row is operated upon, the actual data may have been modified by another program. When it is essential that the most recent data is provided, a JDBC 2.0 driver provides the ResultSet.refreshRow method.

42) What does normalization mean for java.sql.Date and java.sql.Time?

These classes are thin wrappers extending java.util.Date, which has both date and time components. java.sql.Date should carry only date information and a normalized instance has the time information set to zeros. java.sql.Time should carry only time information and a normalized instance has the date set to the Java epoch (January 1, 1970) and the milliseconds portion set to zero.

43) What's the best way, in terms of performance, to do multiple insert/update statements, a PreparedStatement or Batch Updates?

Because PreparedStatement objects are precompiled, their execution can be faster than that of Statement objects. Consequently, an SQL statement that is executed many times is often created as a PreparedStatement object to increase efficiency.

A CallableStatement object provides a way to call stored procedures in a standard manner for all DBMSes. Their execution can be faster than that of PreparedStatement object.

Batch updates are used when you want to execute multiple statements together. Actually, there is no conflict here. While it depends on the driver/DBMS engine as to whether or not you will get an actual performance benefit from batch updates, Statement, PreparedStatement, and CallableStatement can all execute the addBatch() method.

44) What is JDO?

JDO provides for the transparent persistence of data in a data store agnostic manner, supporting object, hierarchical, as well as relational stores.

45) What is the difference between setMaxRows(int) and SetFetchSize(int)? Can either reduce processing time?

setFetchSize(int) defines the number of rows that will be read from the database when the ResultSet needs more rows. The method in the java.sql.Statement interface will set the 'default' value for all the ResultSet derived from that Statement; the method in the java.sql.ResultSet interface will override that value for a specific ResultSet. Since database fetches can be expensive in a networked environment, fetch size has an impact on performance.

setMaxRows(int) sets the limit of the maximum number of rows in a ResultSet object. If this limit is exceeded, the excess rows are "silently dropped". That's all the API says, so the setMaxRows method may not help performance at all other than to decrease memory usage. A value of 0 (default) means no limit.

46) What is DML?

DML is an abbreviation for Data Manipulation Language. This portion of the SQL standard is concerned with manipulating the data in a database as opposed to the structure of a database. The core verbs for DML are SELECT, INSERT, DELETE, UPDATE, COMMIT and ROLLBACK.

47) What is DDL?

DDL is an abbreviation for Data Definition Language. 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 (and available to JDBC), it is often not supported in transactions.

48) How can I get information about foreign keys used in a table?

DatabaseMetaData.getImportedKeys() returns a ResultSet with data about foreign key columns, tables, sequence and update and delete rules.

49) How do I disallow NULL values in a table?

Null capability is a column integrity constraint, normally applied at table creation time. Note that some databases won't allow the constraint to be applied after table creation. Most databases allow a default value for the column as well. The following SQL statement displays the NOT NULL constraint:

```
CREATE TABLE CoffeeTable (  
    Type    VARCHAR(25)    NOT NULL,  
    Pounds  INTEGER        NOT NULL,  
    Price   NUMERIC(5, 2) NOT NULL  
)
```

50) What isolation level is used by the DBMS when inserting, updating and selecting rows from a database?

The answer depends on both your code and the DBMS. If the program does not explicitly set the isolation level, the DBMS default is used. You can determine the default using DatabaseMetaData.getDefaultTransactionIsolation() and the level for the current Connection with Connection.getTransactionIsolation(). If the default is not appropriate for your transaction, change it with Connection.setTransactionIsolation(int level).