

JDBC INTERVIEW QUESTIONS

[LEAVE A COMMENT](#)

How can we 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))
```

What are the new things in JDBC 3.0?

The new features 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.

When do you 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.

Earlier we said we can use same connection but when I try to 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.

Explain some tips to improve the 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.

How does the JDBC work?

The Java Database Connectivity (JDBC) is used to whenever a Java application should

communicate with a relational database for which a JDBC driver exists.

Main JDBC classes:

DriverManager. Manages a list of database drivers.

Driver. The database communications link, handling all communication with the

database.

Connection. Interface with all the methods for contacting a database

Statement. Encapsulates an SQL statement which is passed to the database to be parsed,

compiled, planned and executed.

ResultSet. The answer/result from a statement. A ResultSet is a list which encapsulates

all outgoing results from a given SQL query.

Related Tutorials :

1. [What is JDBC](#)
2. [Spring JDBC](#)
3. [Accessing JDBC from EJBs](#)

4.

Spring Interview Questions

5.

Java Interview Questions

