FACULTY OF Engineering & Technology

Important Interfaces of **java.sql** package:

1. Driver interface
2. Connection interface
3. Statement interface
4. PreparedStatement interface
5. CallableStatement interface
6. ResultSet interface
7. ResultSetMetaData interface
8. DatabaseMetaData interface
9. RowSet interface

Important classes of **java.sql** package:

1. DriverManager class
2. Blob class
3. Clob class
4. Types class

Type-1 driver

1. Type-1 driver or JDBC-ODBC bridge driver uses ODBC(provided bu window os) driver to connect to the database.
2. The JDBC-ODBC bridge driver converts JDBC method calls into the ODBC calls first.
3. Type-1 driver is also called Universal driver because it can be used to connect to any of the databases.
4. The ODBC bridge driver is needed to be installed in individual client machines.
5. Type-1 driver isn't written in java, that's why it isn't a portable driver.
6. This driver software is built-in with JDK so no need to install separately.
7. It is a database independent driver.
8. ODBC calls again changes into database specific call. So this one is the slowest and hence also called snail driver

Type-2 driver

1. The Native API driver uses the client -side native libraries of the database. Hence it is database dependent
2. This driver converts JDBC method calls into native calls of the database API.
3. Driver needs to be installed separately in individual client machines
4. The Vendor client library needs to be installed on client machine.
5. Type-2 driver isn't written in java(mostly written in c or c++), so it is platform dependent also

Type-3 driver

1. The Network Protocol driver uses middleware (application server) that converts JDBC calls directly or indirectly into the vendor-specific database protocol.
2. Maintenance of Network Protocol driver becomes costly because it requires database-specific coding to be done in the middle tier.
3. They are fully developed in java so it is platform independent driver.
4. It does not interact directly with the database so also database independent.
5. Cost is high due to extra middleware IDS server.

Type-4 driver

1. Type-4 driver is also called native protocol driver or thin driver.
2. This driver interact directly with database. It does not require any native database library that is why it is also known as Thin Driver.
3. Does not require any native library and Middleware server, so no client-side or server-sid installation.
4. It is fully written in Java language, hence they are portable drivers.
5. They are database dependent driver
6. Performance is very high.