What is JDBC?

JDBC stands for Java database connectivity.

It's Java's way of providing a consistent way to connect to a wide variety of databases, including relational, NoSQL, and object-oriented databases.

It abstracts the complexities of connecting to different databases, through a common interface.

This means we can use the same code to interact with different databases, without needing to rewrite, or have a separate version of the application logic for each one.

You can think of JDBC as a middleman, that sits between a Java application and a data source.

You can also use JDBC with spreadsheets and flat files, meaning you can use structured query language to interact with these files.

What's a JDBC Driver?

To use a particular data source from an application, you'll need a JDBC driver for that data source.

A driver is simply a Java library, containing classes that implement the JDBC API.

Drivers are usually provided by the database vendor, either as a library jar, or a java module, which we can import into our application.

The current version of the JDBC API, is JDBC 4.3.

For backwards compatibility, it contains all the methods that were in previous JDBC versions.

You can find this information in the java.sql package API documentation.

What's the JDBC Driver used for?

At the most basic level, these drivers allow us to:

- Connect to the database. Each database may have a different mechanism to establish a connection to it.
- Execute SQL statements. These statements can be DML statements, any of the CRUD statements for example, or they can be DDL statements.
- Execute Stored procedures. This sends a request to the database to execute a procedure or function stored in the database.
- Retrieve and process results. This could be a set of data from a select statement, or a count of rows updated or inserted.
- Handling Database Exceptions.

Differences between JDBC 4.0 and previous versions

Since there have been changes to the way that JDBC works over time, you want to be sure you're working with a driver that supports at least JDBC 4.0.

Some Features in JDBC 4.0 and later include:

- Auto-loading of JDBC drivers.
- Built-in Connection Pooling.
- Distributed Transactions.
- Row streaming.
- SQLXML support.
- Try-With-Resources.

java.sql and javax.sql

JDBC consists of two packages, java.sql which is core JDBC, and javax.sql, which provides the API for server side data source access.

The java.sql package contains most of the types you'll need to communicate with your data source.

But javax.sql provides two alternative types, for several of the most significant types.

Purpose	java.sql	javax.sql
Makes a connection with a driver	DriverManager	DataSource
Query Results	ResultSet	RowSet

DriverManager should be replaced with DataSource in most cases, because it's newer and supports a lot more functionality.

For querying results, there's the standard ResultSet type, but the RowSet interface provides many advantages.