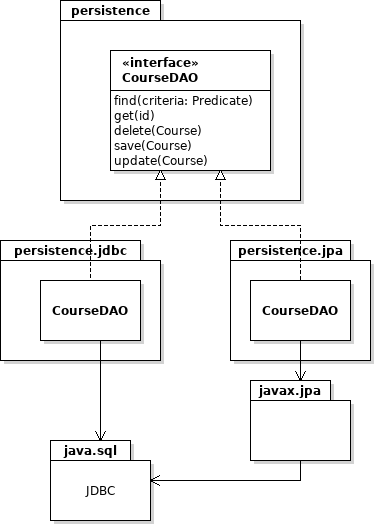
## Week 12 Topics

1. Review Quiz 2 answers

2. Intro to some Database Software

3. Design a Persistence Component

This week: implement DAO using JDBC

Next class: implement DAO using JPA (better approach)

Java Persistence API (JPA) is a standard that is part

of the Java EE specification.

JPA is a specification that anyone can implement.

Popular implementations are EclipseLink and Hibernate.

4. Designing an ORM

a. The problem and requirements

b. Two designs: Active Objects and Data Access Objects

5. Java Classpath & Project Dependencies

JAR files

Other external dependencies

### How JDBC Works

To use JDBC you first create a connection to a database, using a Connection object. Connection objects are specific to the type of database, e.g. MySQL Connection, Oracle Connection.

Use the Connection object to create a Statement. Statement is a reusable command object that you use to execute SQL commands and get the results. You can specify attributes for a Statement (such as how many results it can hold). Connection can also create another kind of Statement called a PreparedStatement, which has better performance and security.

SQL "SELECT" queries return results as a ResultSet object, which lets you iterate over rows in the result. A ResultSet also contains *metadata* you can use to discover information about the results.



Database

*Connect and authenticate*

*execute*

### How does DriverManager know *what* *database* to use?

The first parameter to DriverManager.getConnection( ) is a URL.

The **url** parameter identifies the location of the database server and the *kind* of database (DB2, MySQL, etc). Here are some examples of URLs:

**jdbc:mysql://se.cpe.ku.ac.th/world** MySQL database on a server

**jdbc:mysql://localhost:3306/world** MySQL database server on this host

**jdbc:sqlite:/database/world** SQLite database on this host

**jdbc:hsqldb:file:/database/world** HSQLDB embedded database on this host

If the URL starts with "jdbc:sqlite" then DriverManager looks for a **registered driver**for "sqlite". It instantiates a Connection class provided by that driver (org.sqlite.jdbc4.JDBC4Connection) and returns it. Your code uses the Connection object to create other objects (Statement, PreparedStatement, etc). The Sqlite JDBC4Connection class returns concrete implementations of Statement, PreparedStatement, etc.