Day 2

Tuesday, August 16, 2022 2:23 PM

Object Relational Mapping

ORM's turn java objects into database table entries

Most OOP apps store data in relational db's however we cannot just take an object and put it in a table

With JDBC we have been doing this manually

With an ORM the conversion will be done for you

Benefits of an ORM
- Map objects to tables for us
- Hide the implementation of SQL queries from OOP
- Provides automatic versioning and time stamping
- Provides caching for better support
- Best suited for large projects
- Injected transaction management
- Logging
- faster development

Hibernate / JPA

Java Persistence API (JPA)
- Standard Java API for accessing/persisting/managing data between objects and db's

This can be found javax.persistence
- Has its own query language JPQL
  - OOP query language
- JPA uses an EntityManager: interface to do crud operations

Hibernate: ORM Library for Java which implements JPA

flexible and powerful ORM solution to map Java classes to database tables

Implementation of JPA: when using hibernate we will use JPA imports

Has its query language similar to JPQL
- Hibernate Query Language (HQL)

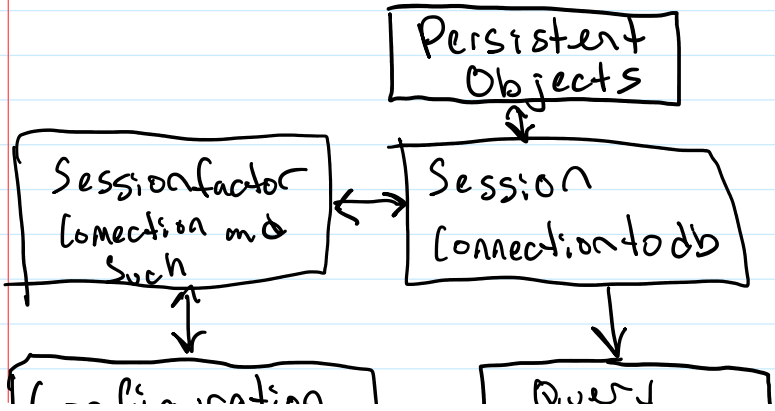
Hibernates Session interface extends JPA EntityManager
-has basic crud built in

Why we should not use JDBC anymore
- Large applications require very complex SQL
- Changing db's could require major refactoring
- We must manually convert objects
- Developers must understand SQL and Java
- The states of the objects must be fetched and managed

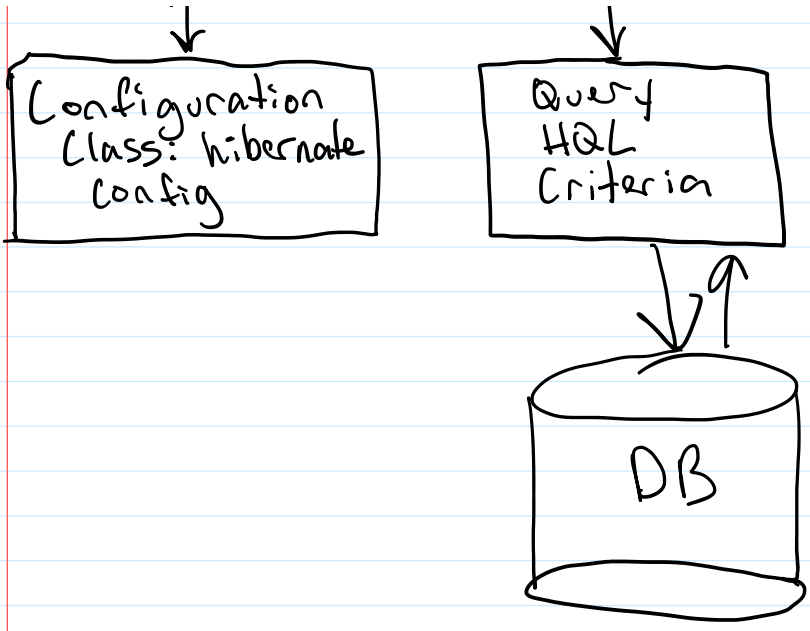
Hibernate Benefits
- transparent persistence ensures automatic connection between objects and db tables
- Hibernate is database independent
- Hibernate provides abstraction from creating connections, and basic queries
- Dual-level caching for performance

Hibernate Architecture

Hibernate is broken up into several key interfaces
- Hibernate architecture



POJO classes/models in Java
Annotated with JPA annotations



Transaction: used to actually persist/commit data to the database

Hibernate Configuration:
Properties can set in the hibernate.cfg.xml
Or in a class file
Or both a class and an XML