* SUPER ASIDE
  + Cascade is a keyword in sl that cause commands to propagate to other tables
  + On update or delete
  + If I delete a row in role, I also delete all users that had that role.
* Hibernate?
* Is a framework, made by Gavin King
  + What is an orm
    - Object Relational Mapper
    - This means we map our objects (models) to tables in our database
    - Movie model and tie it to a movie table in a db
  + What are the benefits of an orm?
    - We don’t have to write sql queries anymore
    - It allows us to use a sql db without having to worry about what dialect of sql we are using
    - Allows to manipulate database in a more OOP way
  + What are the downsides of an orm
    - It’s more complicated
    - It has more upfront configuration
    - It makes errors harder to trace
* How do I configure Hibernate
  + There is a special class hibernate has called Configuration
  + Using an instance of COnfiguration object from hibernate we can build the rest of the objects we need
  + We build that Configuration using an xml file
  + What are some important tags for my xml
    - Hibernate.dialect - what sql dialect
    - Hibernate.connection.username
    - Hibernate.connection.password
    - Hibernate.connection.url
    - Hibernate.connection.pool\_size
      * Just like pg library for node we utilize connection pooling in hibernate
      * (Benefits of connection pooling review)
  + What is the hbm2ddl tag
    - This tag has 4 different values
      * Create - when the program starts up, it completely rebuilds the database schema using the current database model mappings
      * Update - will only rerun ddl statements if they have changed since last time
      * Validate - just make sure the database and the mappings have the same effect or it won’t start
      * None - do nothing with ddl
  + What are mapping tags
    - These tags tell hibernate what classes or files we use to define the database mappings
* Mappings
  + What are the 2 ways to generate object mappings in hibernate?
    - We can create a .hbm.xml file that contains many xml tags that correspond to fields on a class and how to relate those fields to the database
    - We can add annotations on a class and above fields and put the relational information there
  + What annotations are required in a mapping
    - @Entity - tells hibernate this is a model mapping
    - @Id - this one denotes which field is the primary key on the object
  + How does this all work?
    - By default Hibernate assumes the column name in the database is the exact same as the field name on the object
    - Hibernate knows all of the different objects we have mapped and can tell if we a have a field of another mapped object that it will have to resolve a FK relationship ( hibernate will write your joins for you )
  + Annotations for multiplicity?
    - @ManyToOne - with this we put a @JoinColumn ( we put this on the many side )
    - @OneToMany - mappedBy property (we put this on the one side )
    - @ManyToMany - @JoinTable - joinColumn and inverseJoinColumn
      * Actually builds the join table when doing auto ddl
  + Annotations for manipulating the column in the db?
    - @Column - specify column name, column type and such forth
* JDBC?
  + Java database connectivity
  + It’s basically just the pg library but for java
* Interfaces of Hibernate
  + What are the interfaces of Hibernate
  + Explain the purpose of each and some methods you might use
* Object States
  + What are the states of an object in hibernate
  + How can I get an object into each of those states
* Caching
  + What is l1 caching?
  + How does it relate to object states
  + What is l2 caching
* Proxies
  + When do I get a proxy in Hibernate
  + How do I resolve a proxy
  + Exceptions related?
  + Why are proxies good