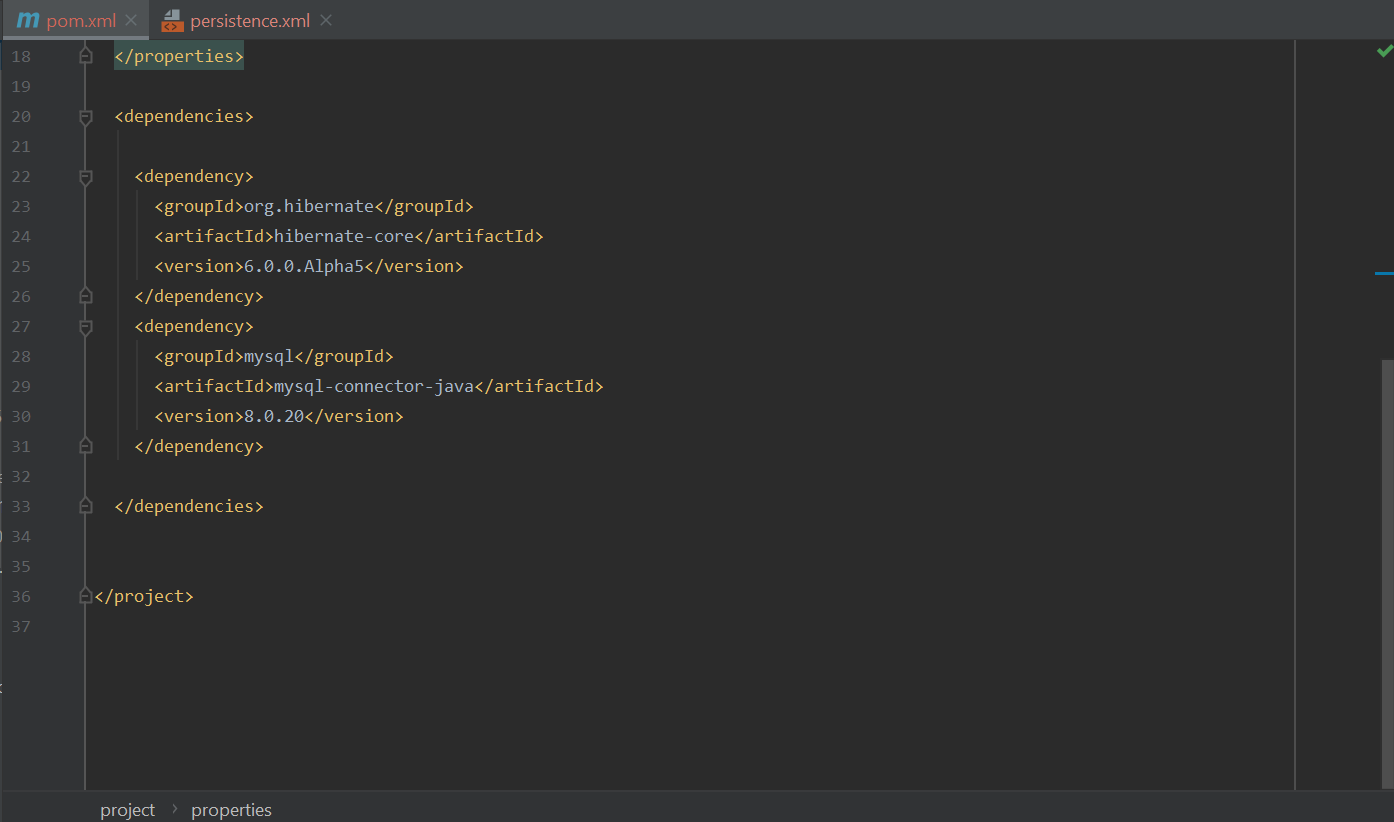
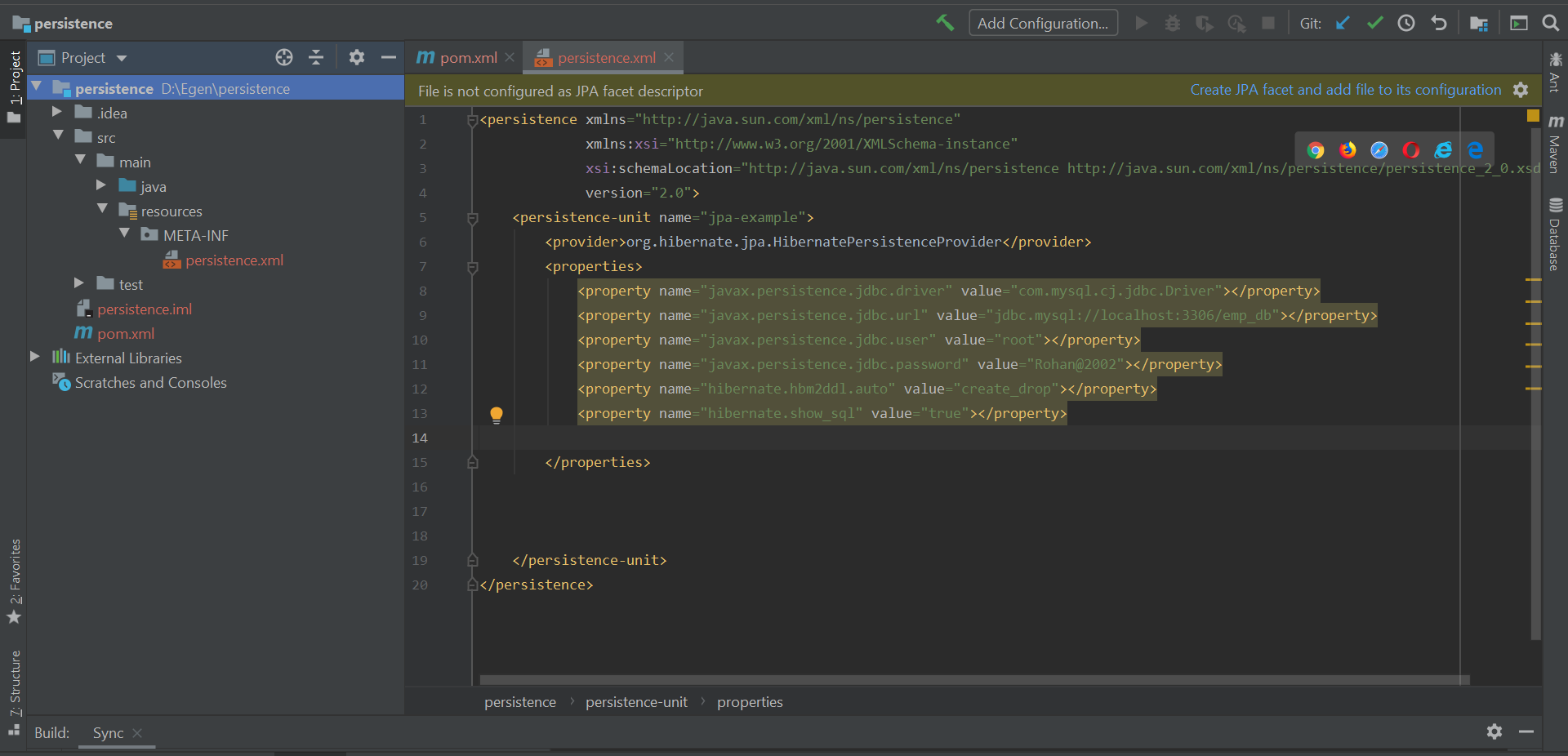
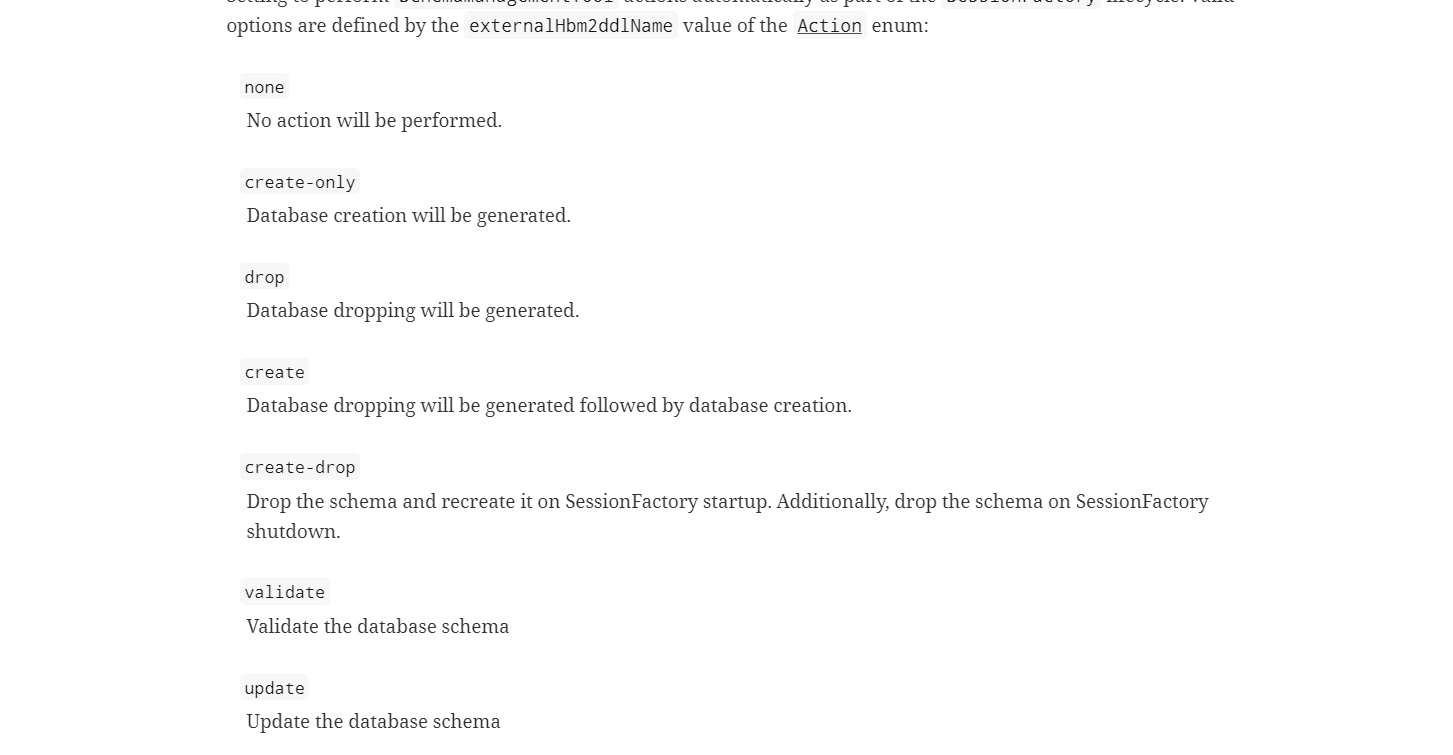
JPA: Java Persistence API

Connection to mysql requires hibernate core which lies on top of jpa

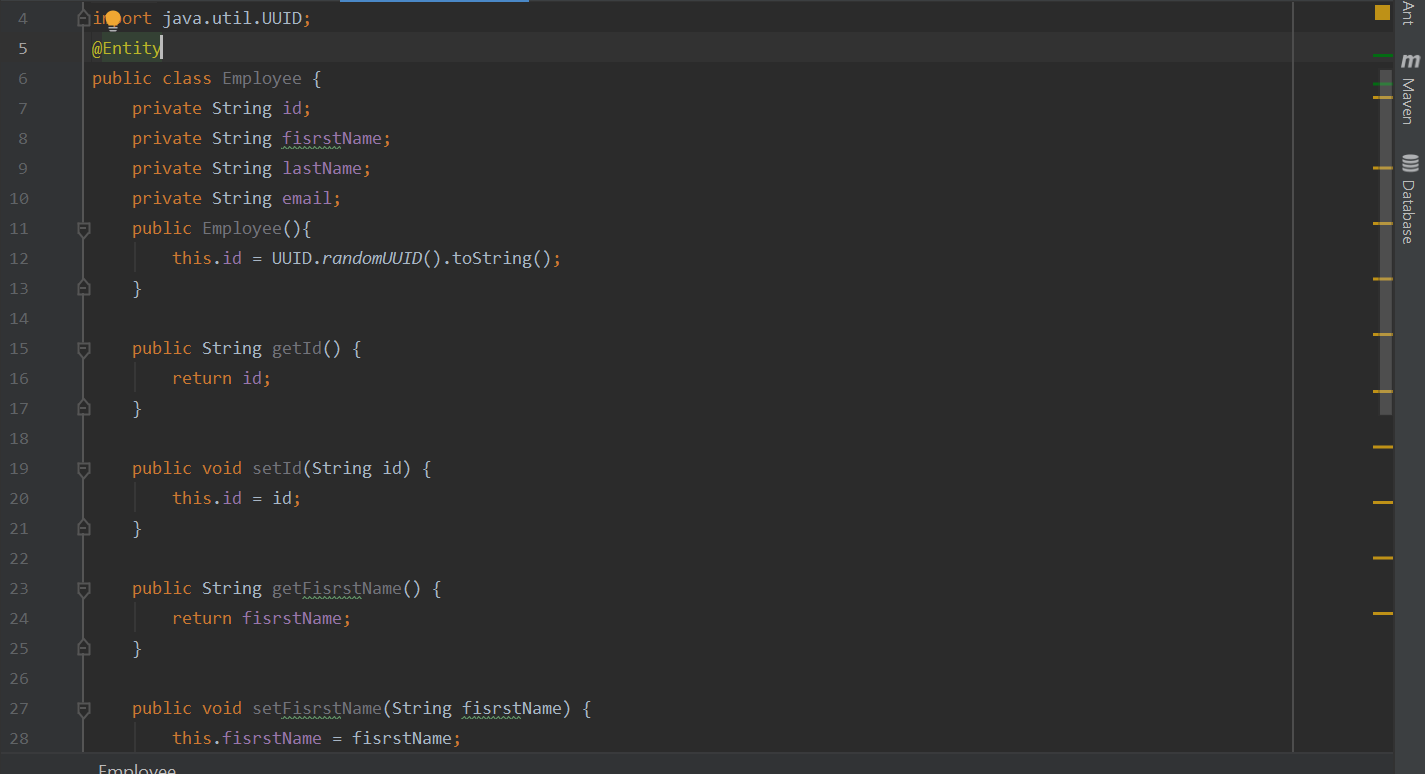




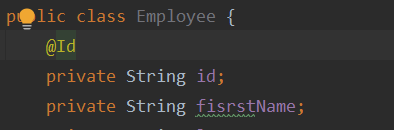
<https://docs.jboss.org/hibernate/orm/5.2/userguide/html_single/Hibernate_User_Guide.html#configurations-hbmddl>



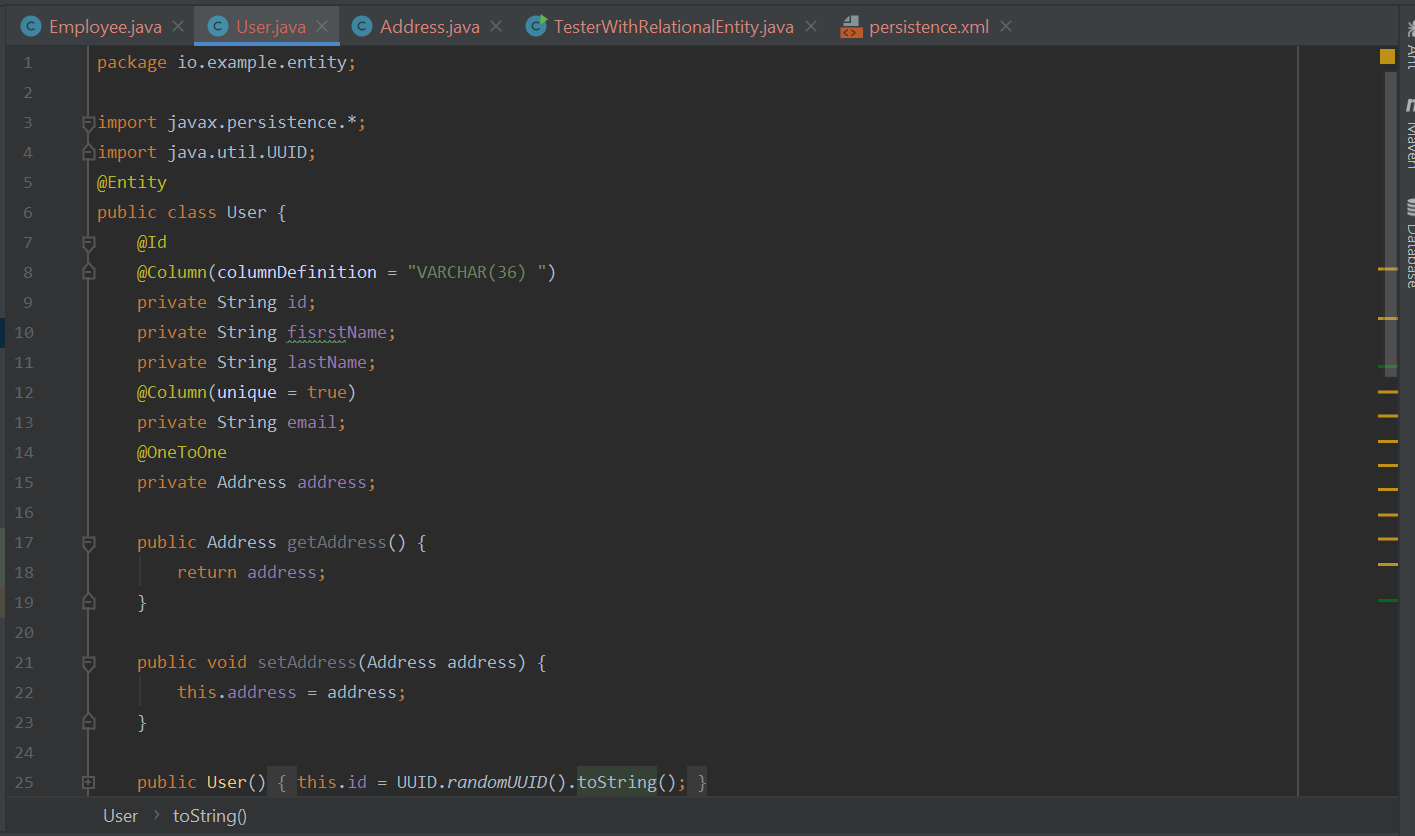
To tell java it’s a jpa entity

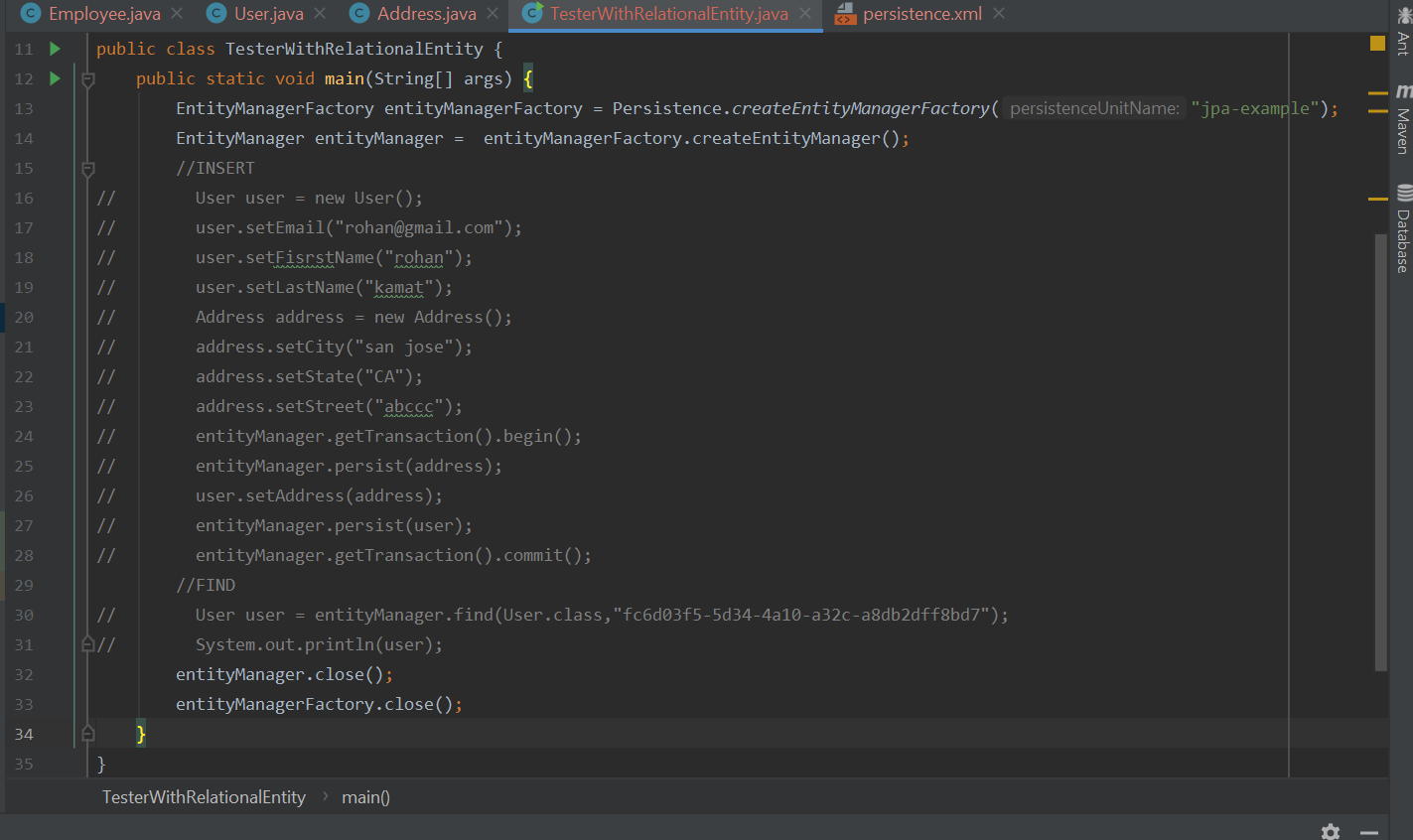


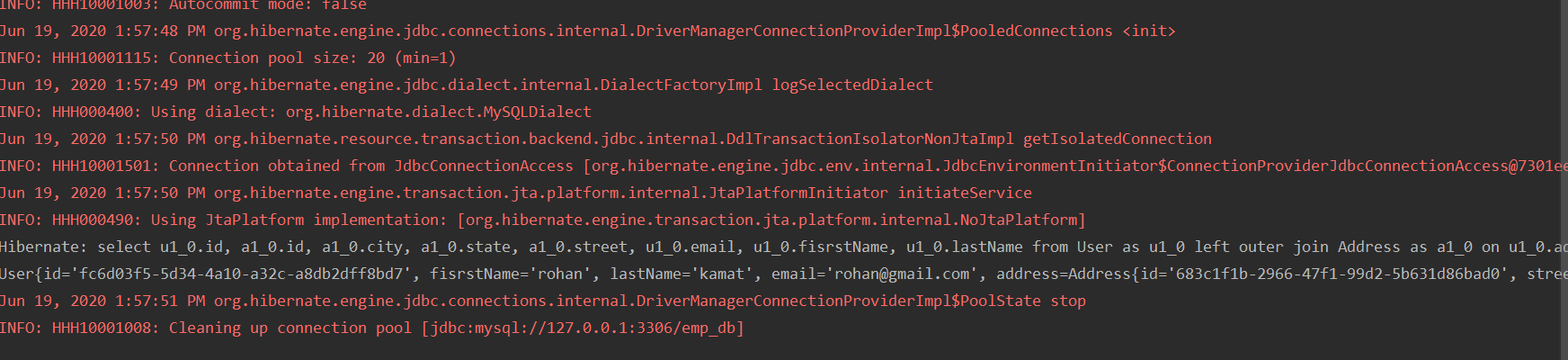
Primary key

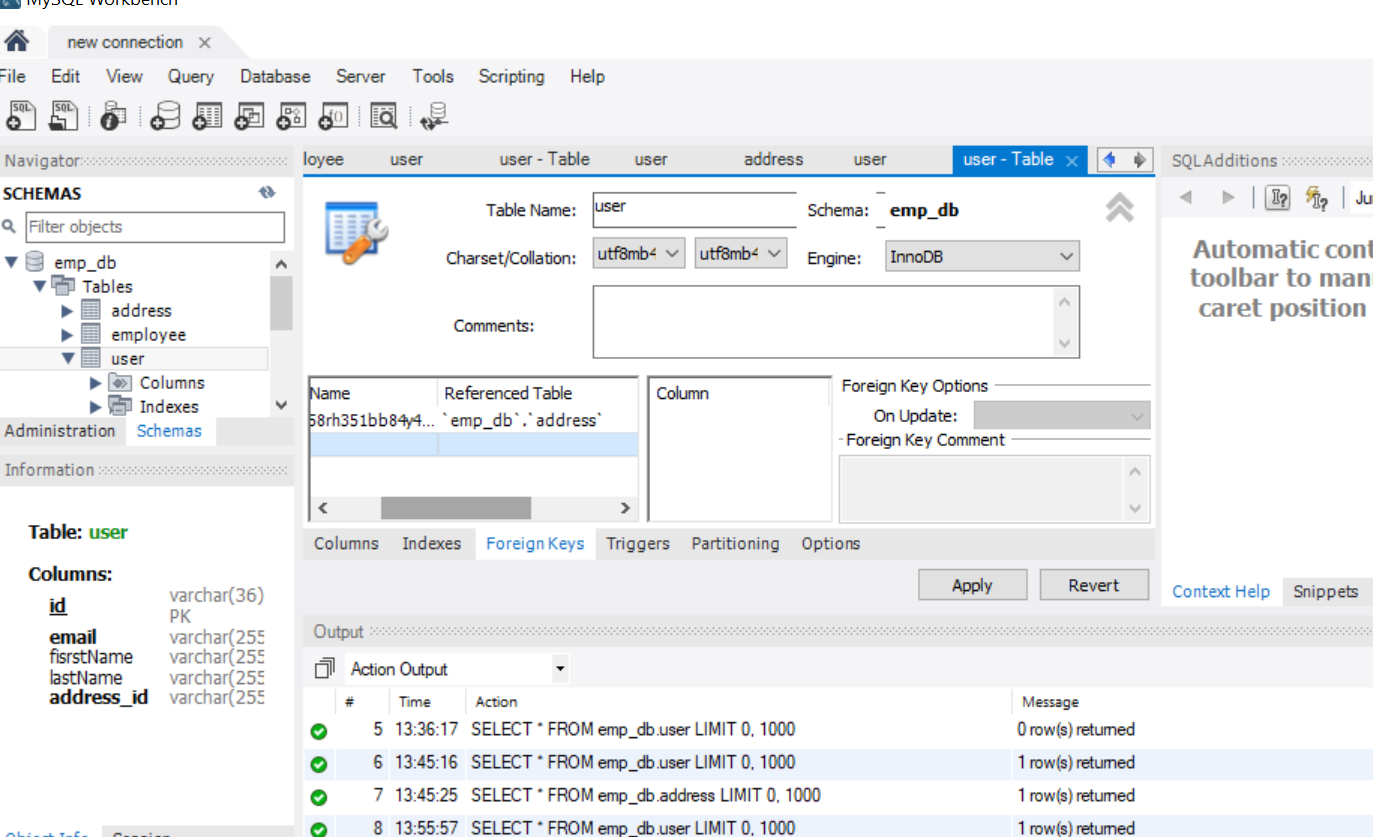


One to one mapping

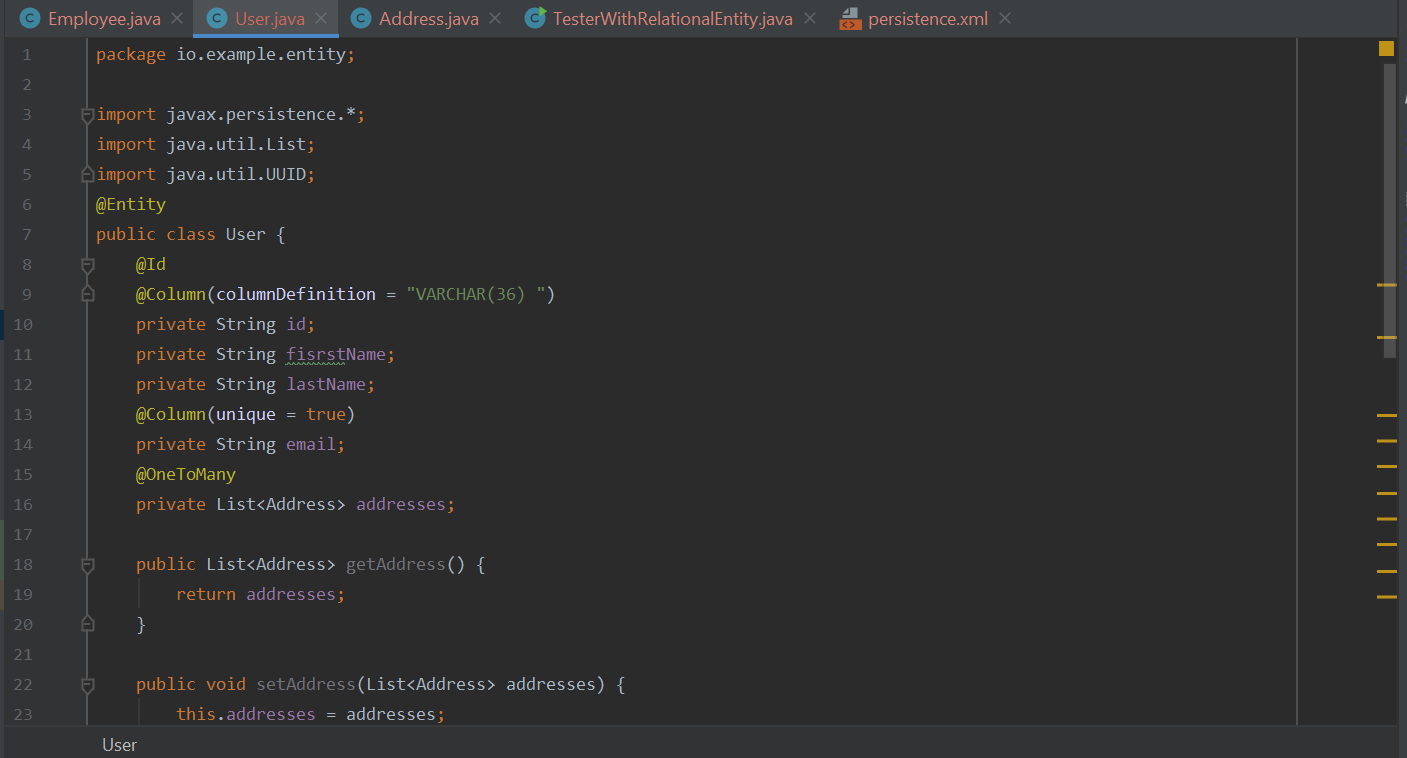


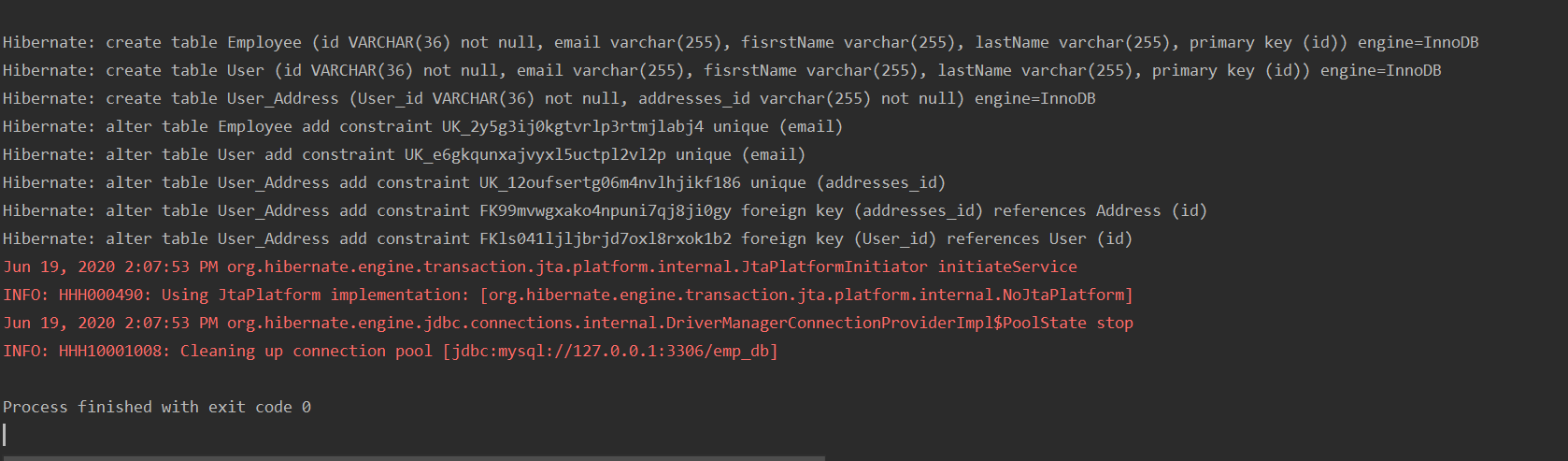


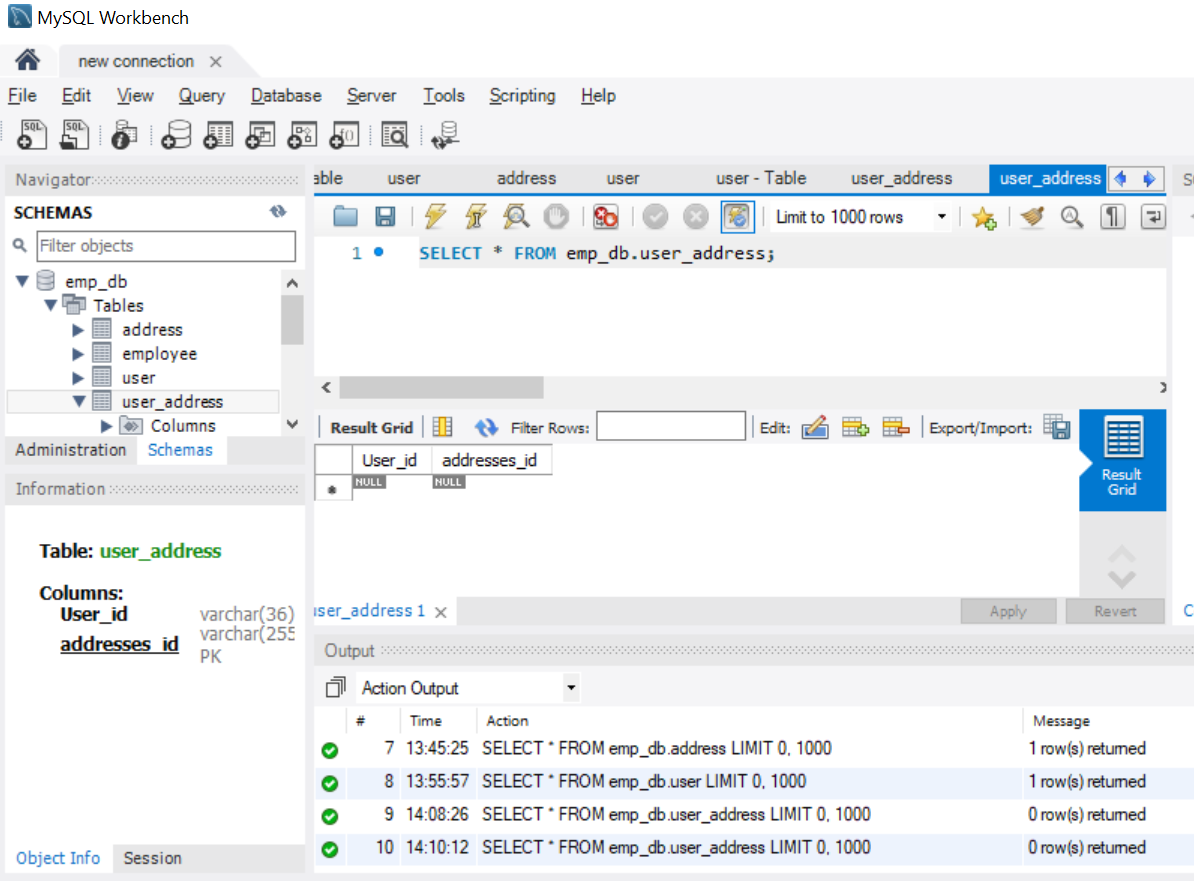




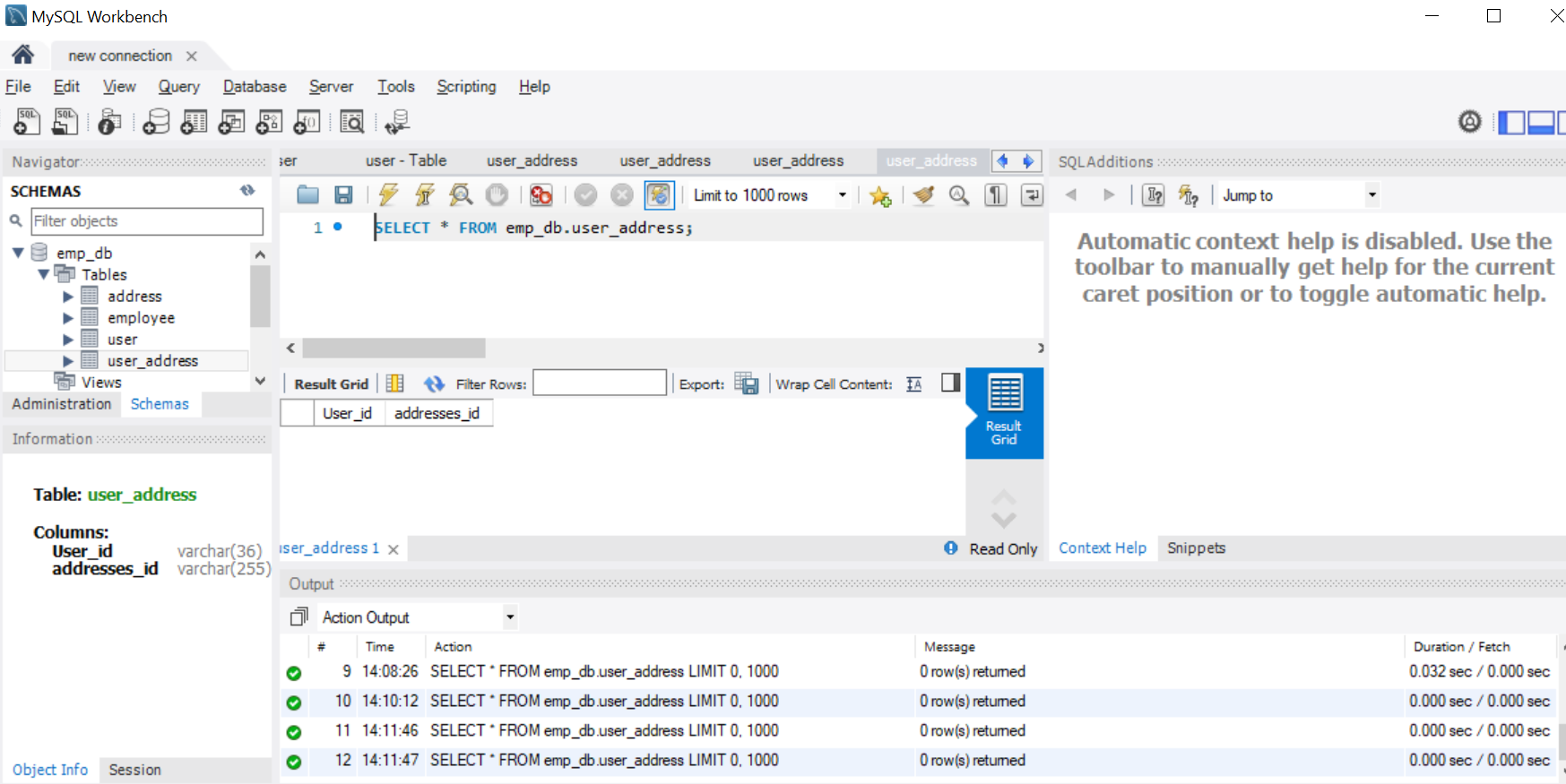
One to many

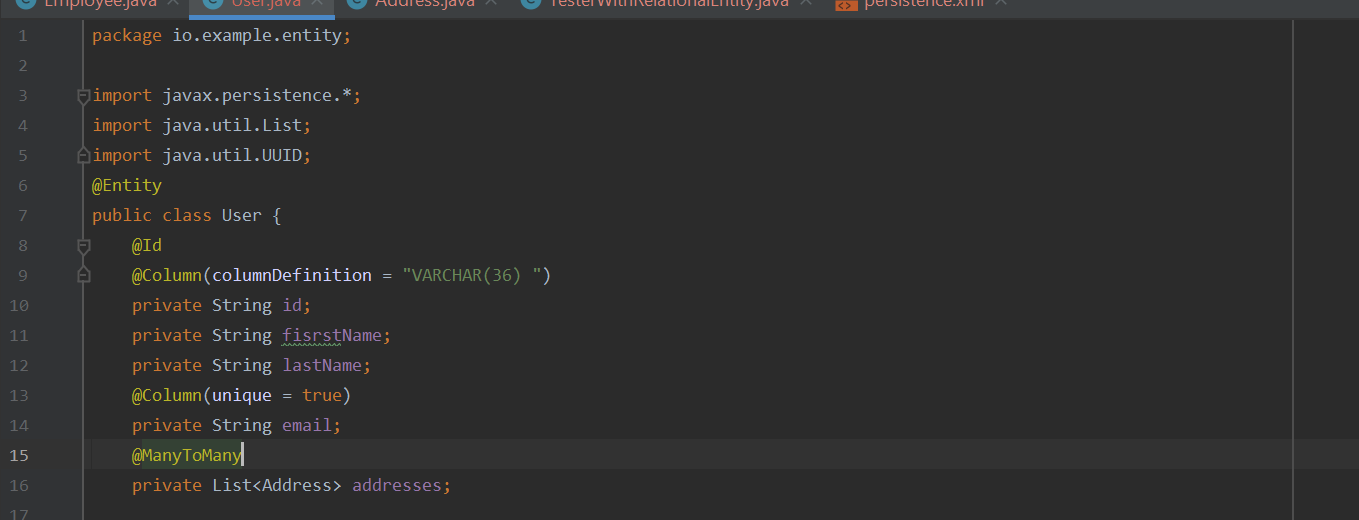


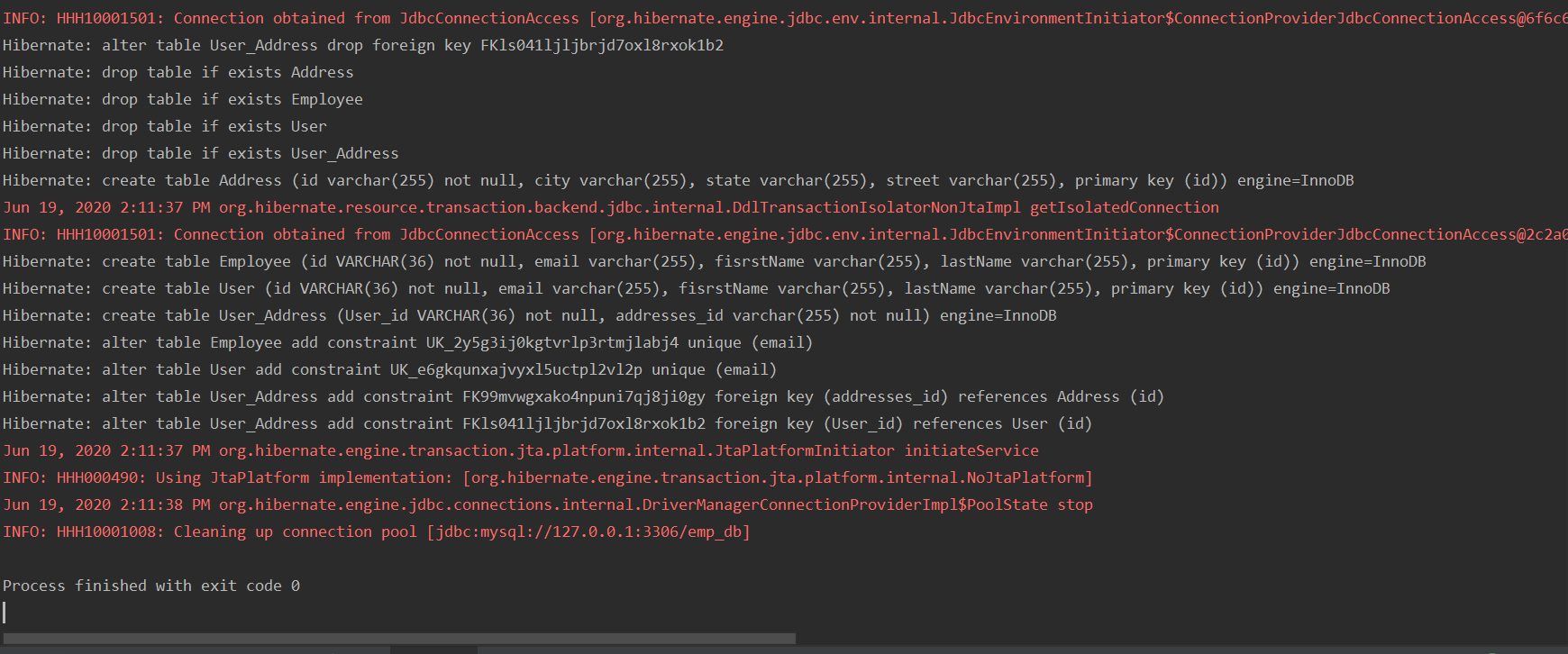




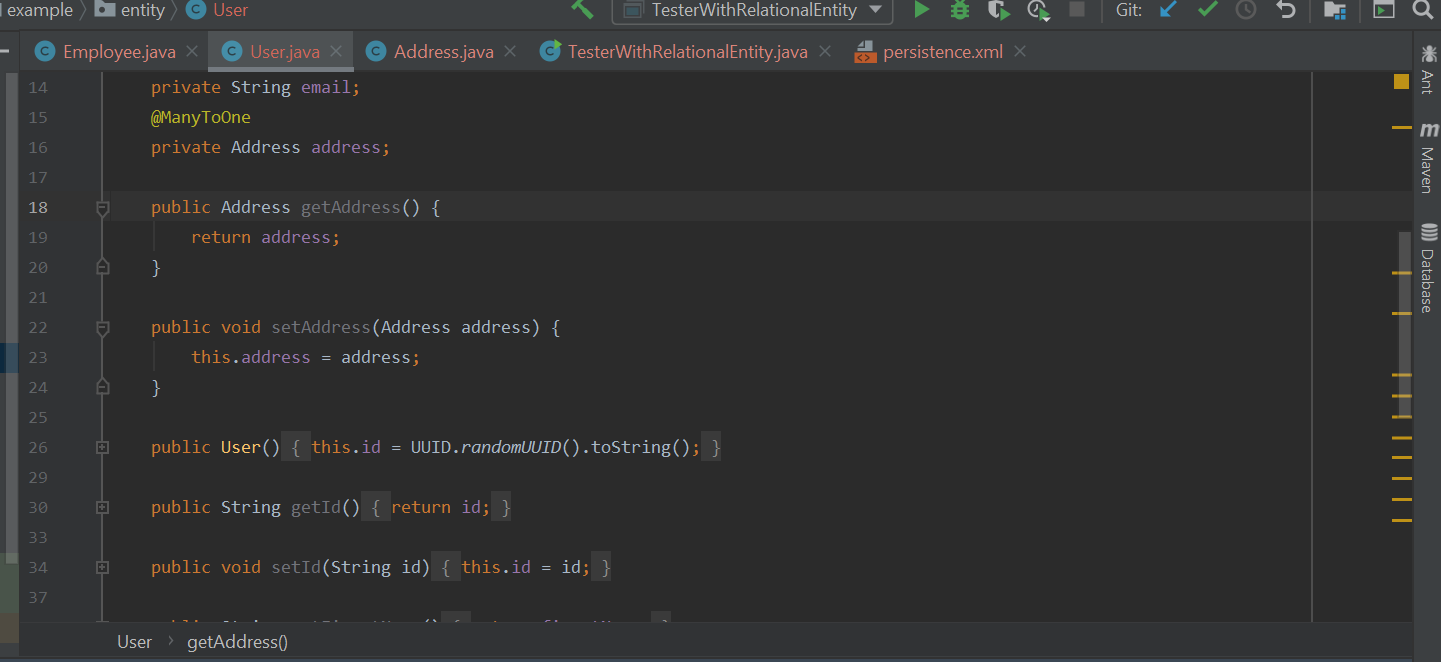
Many to many

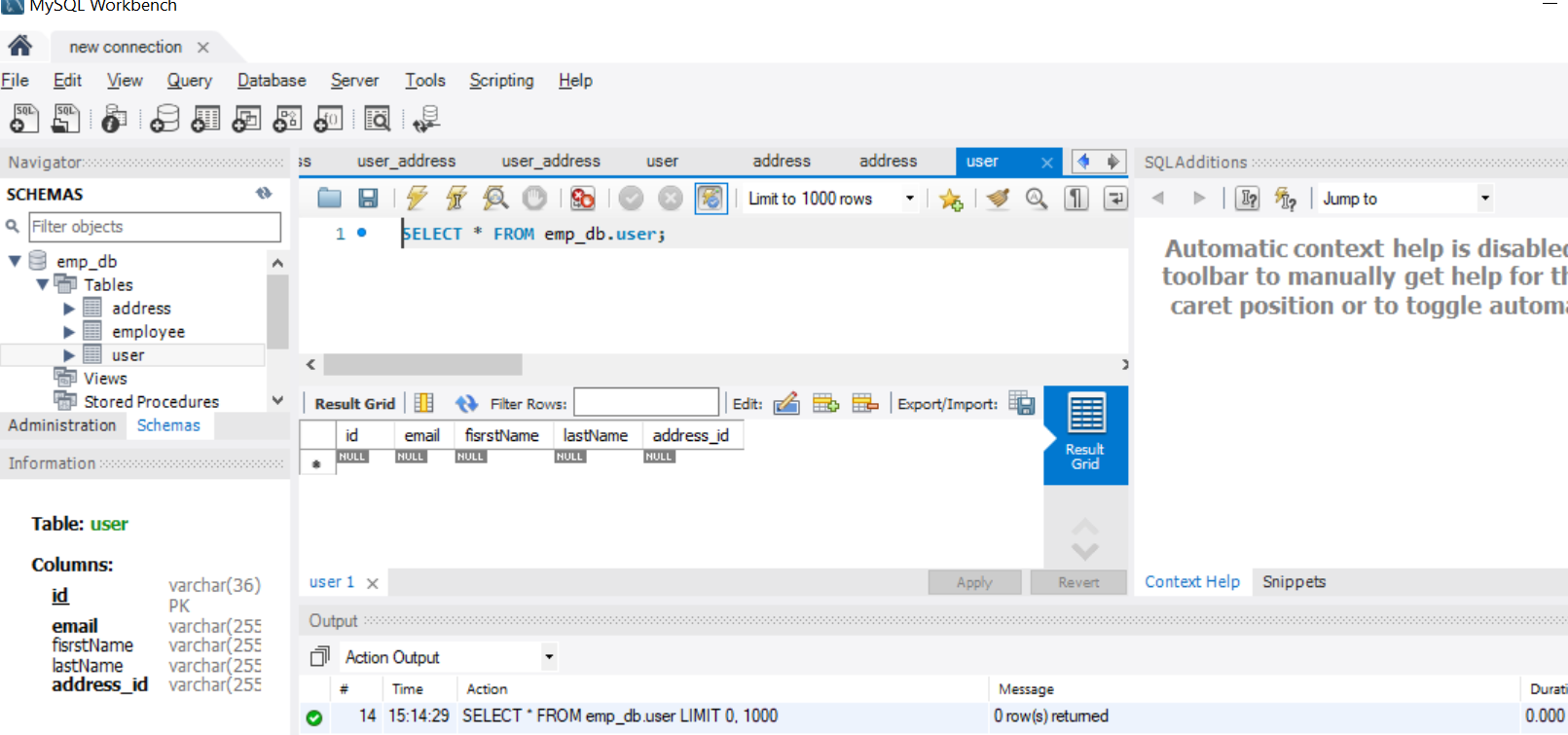
Address id is no longer the primary key the joint becomes a primary key





Many to one

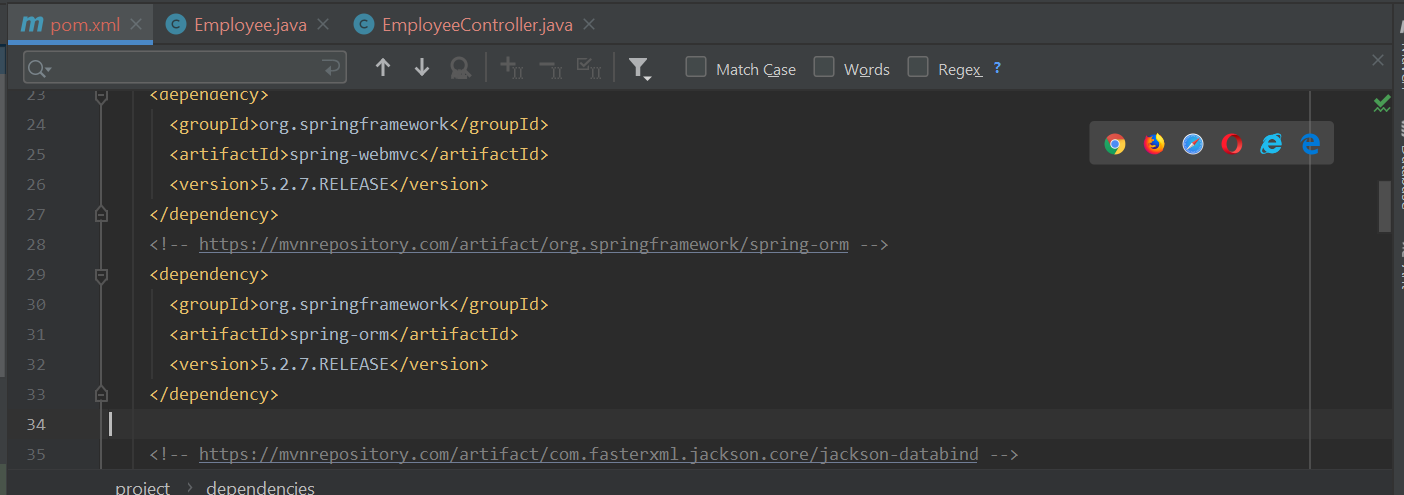




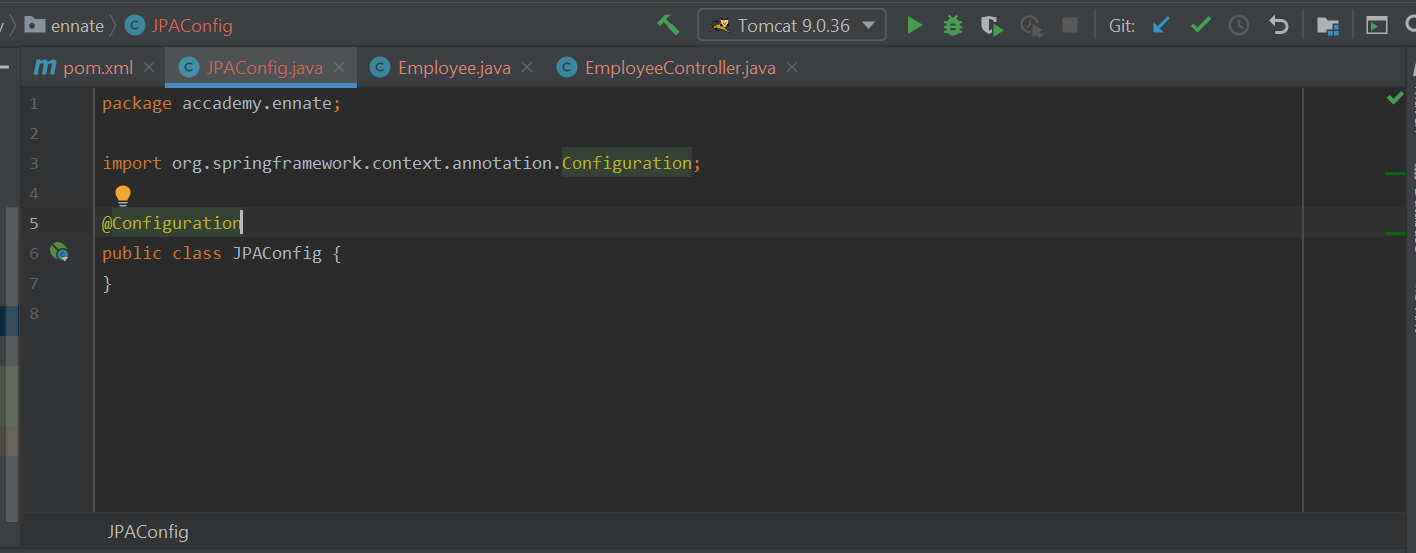
Spring boot

No need for persistence

ORM Package import



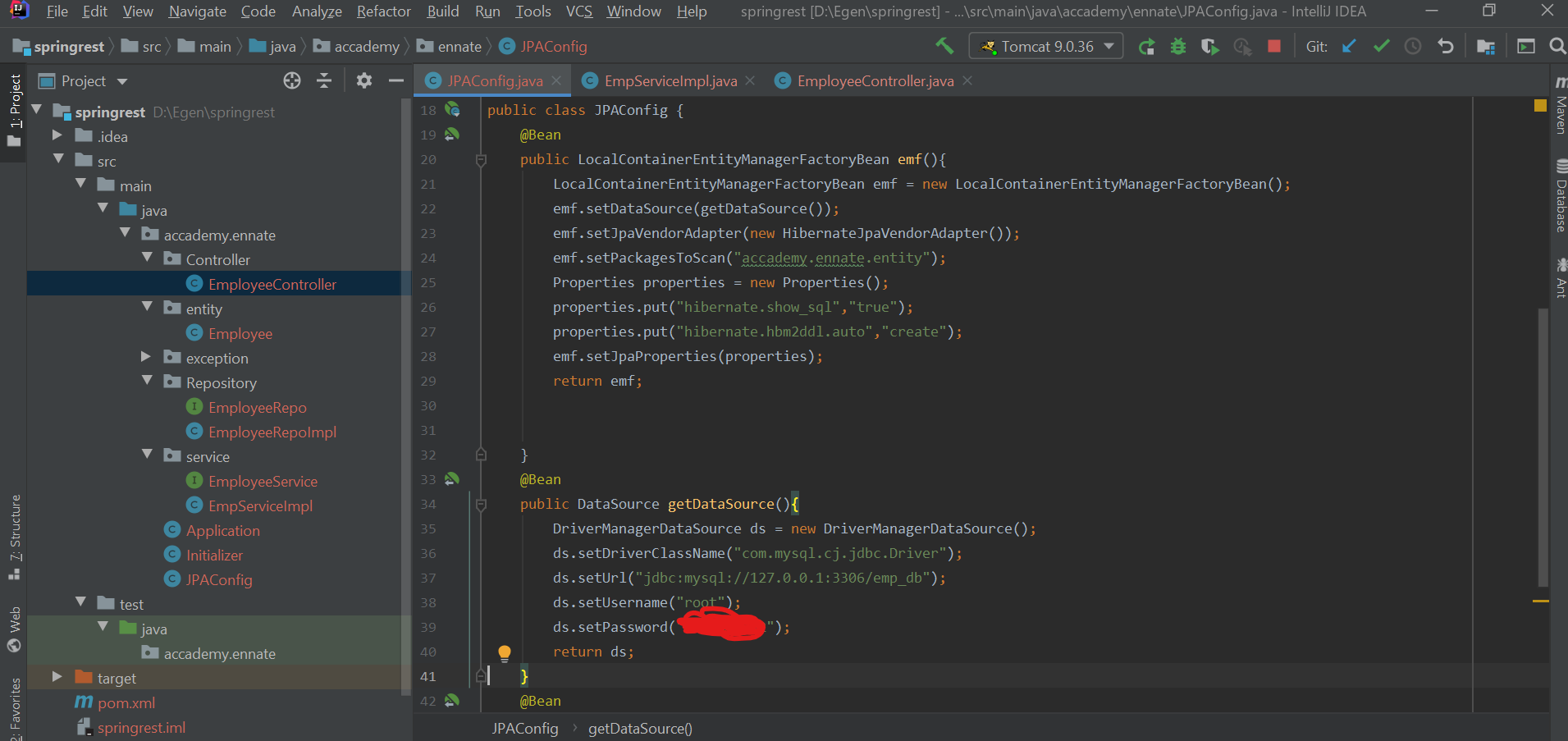
Use annotation instead of persistence.xml for spring boot



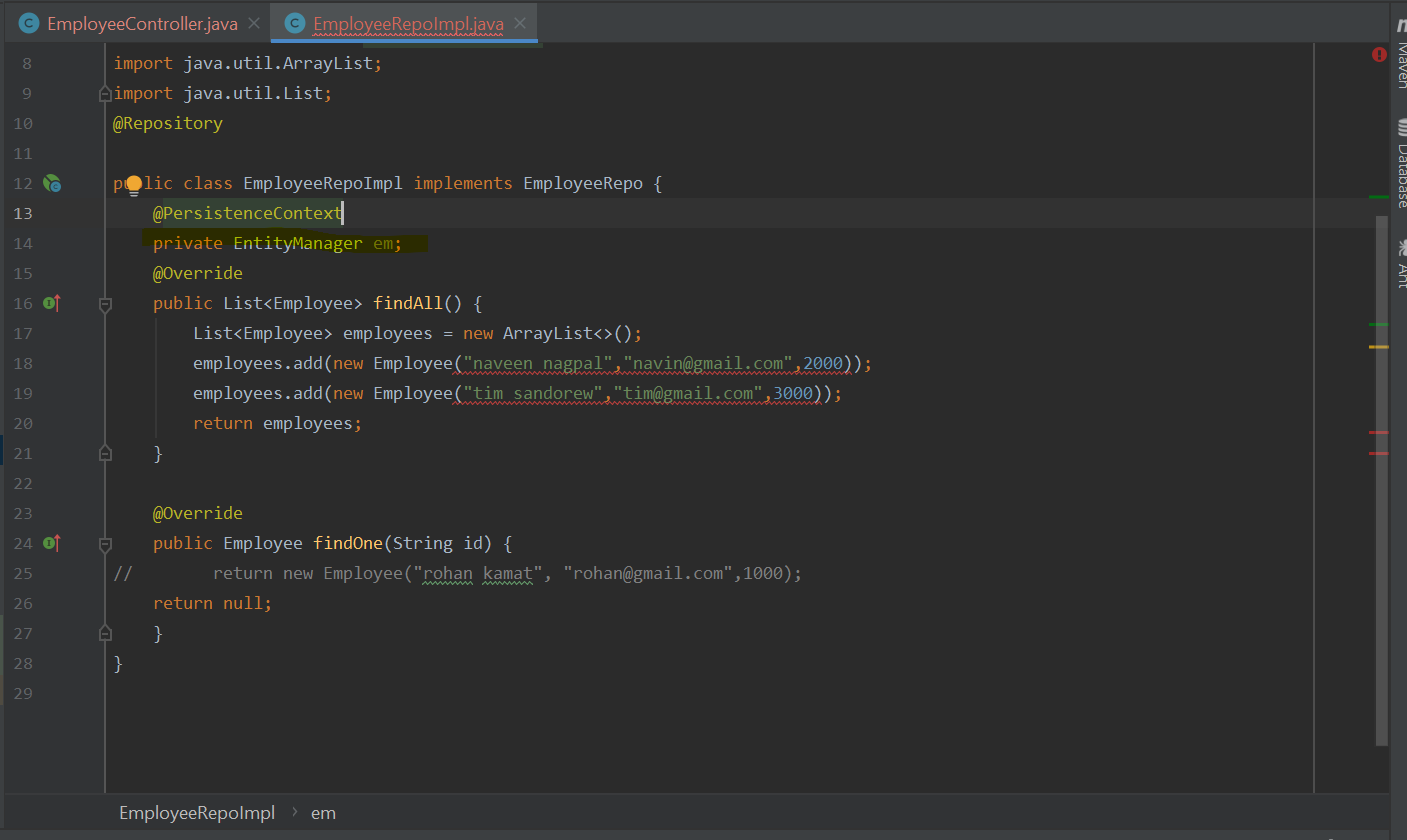
Load both configuration



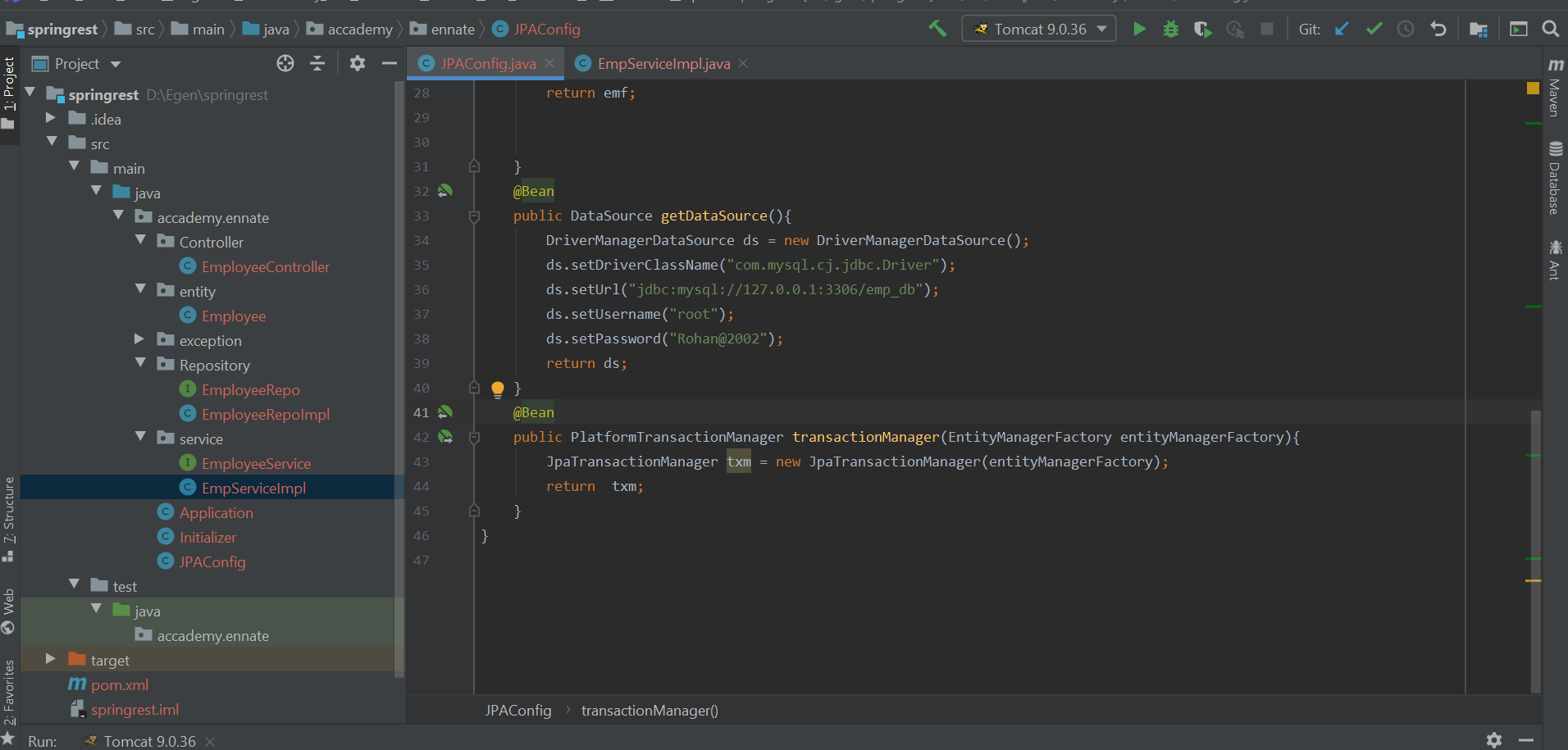
Setup the connection using the following, set entity package to scan



For entity manager

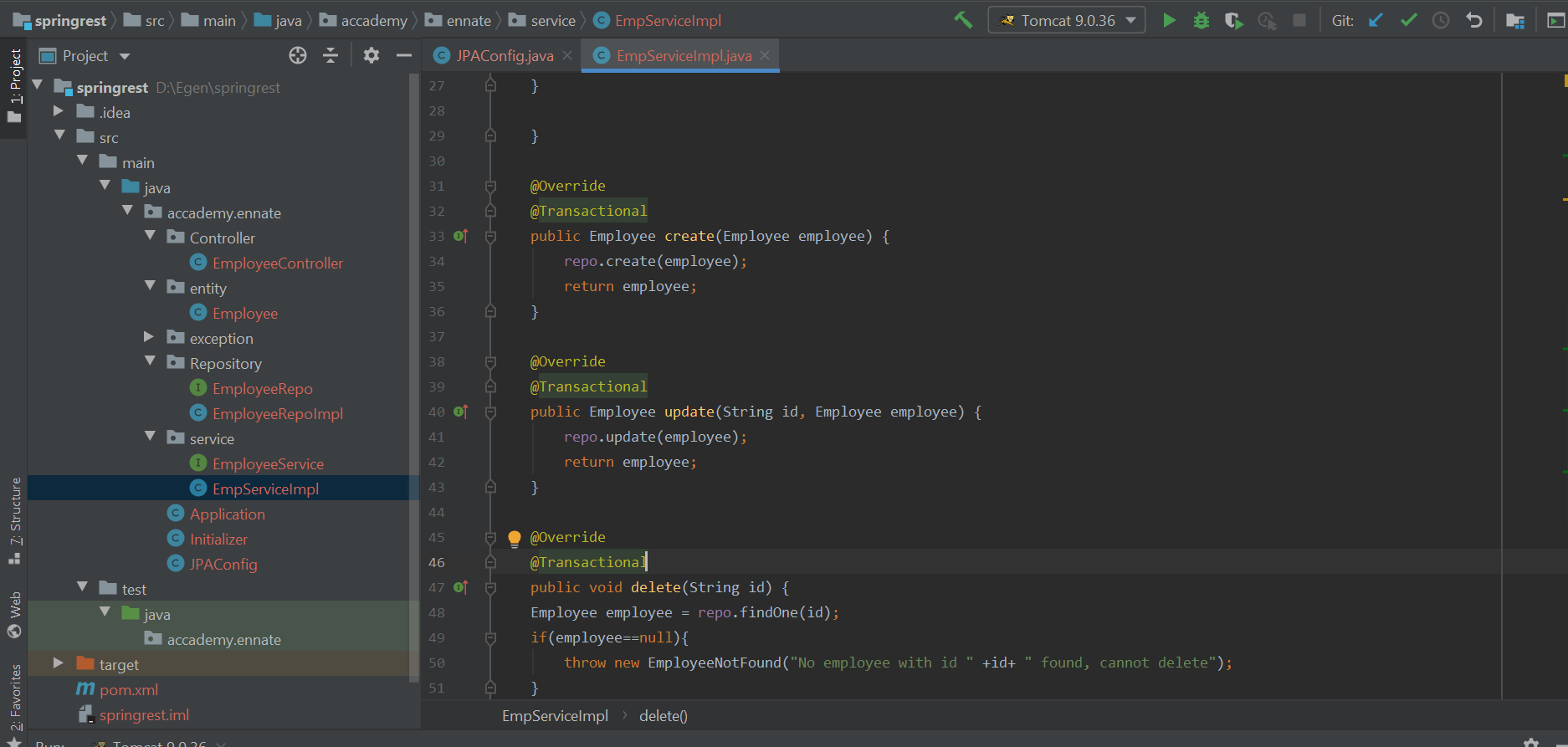


Transaction Management in Spring

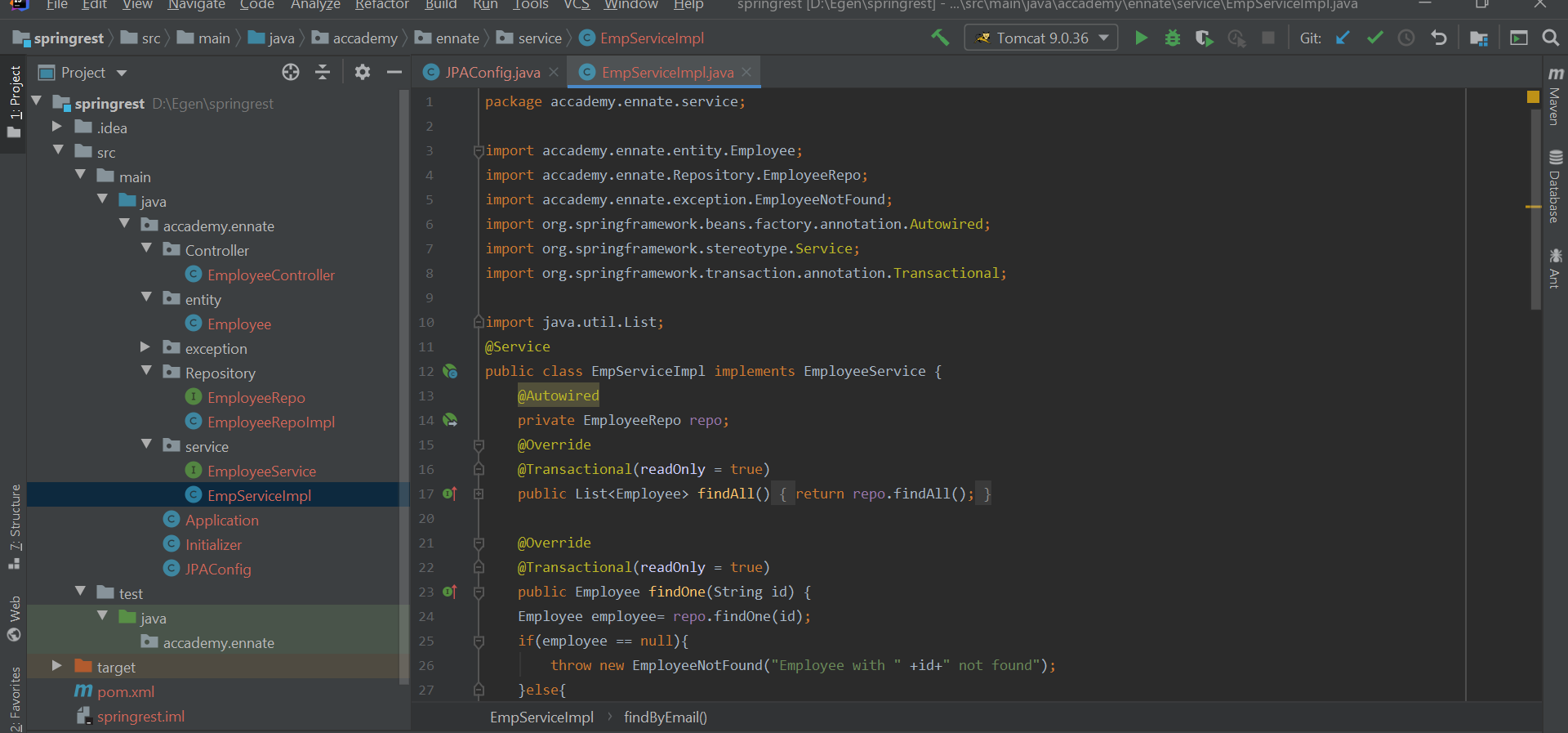


@Transactional annotation can be used, but have 3 options at repo, service and controller level.

As we have multiple calls in service (per method) it should be used at service layer(optimal) Helps for rolling back.

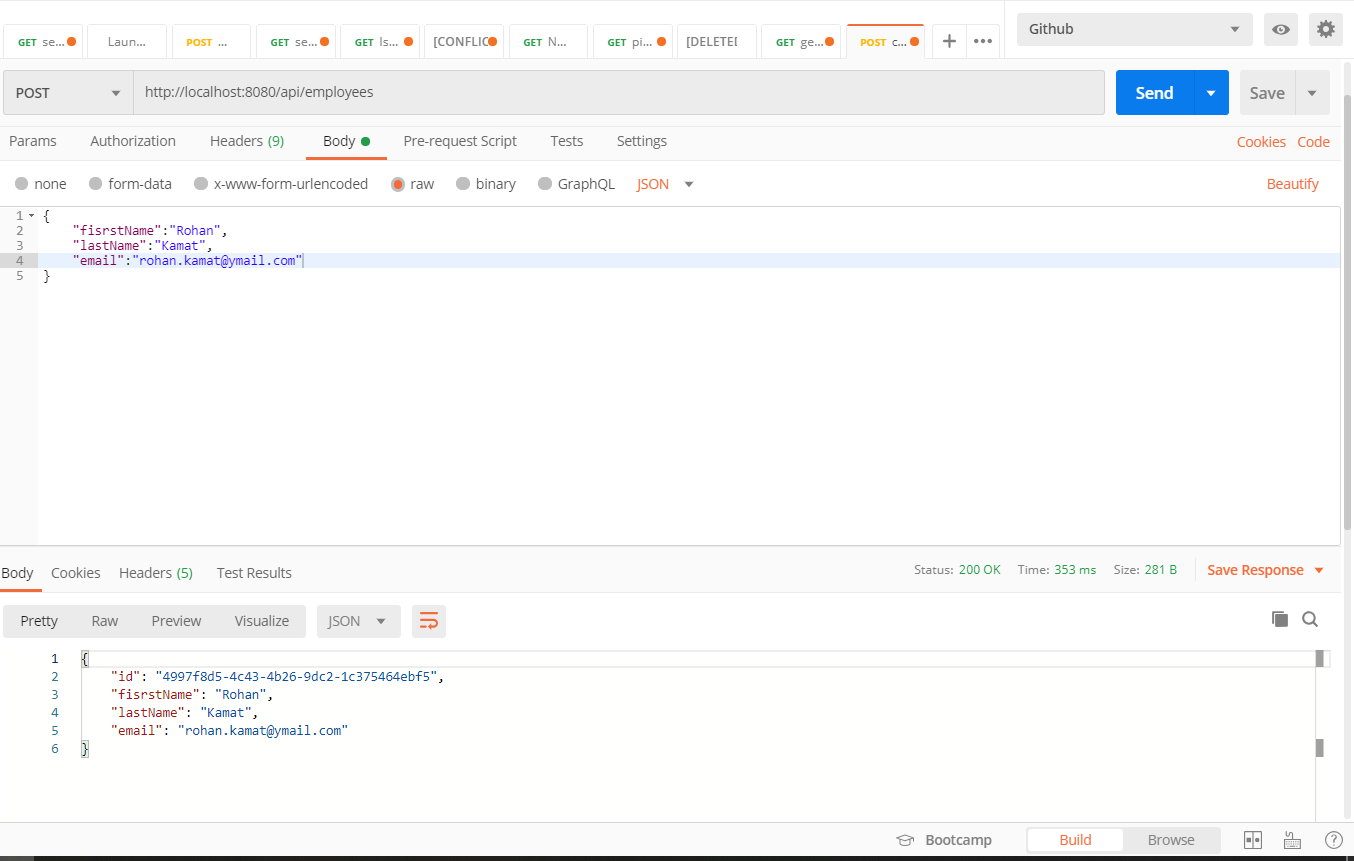


Read only optional



Testing Using Postman

Create Employee



Get all

