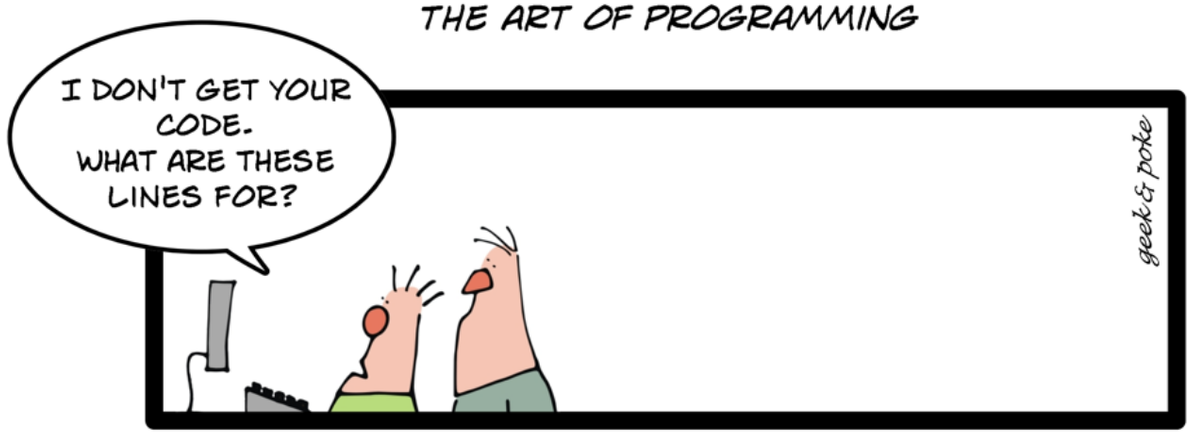
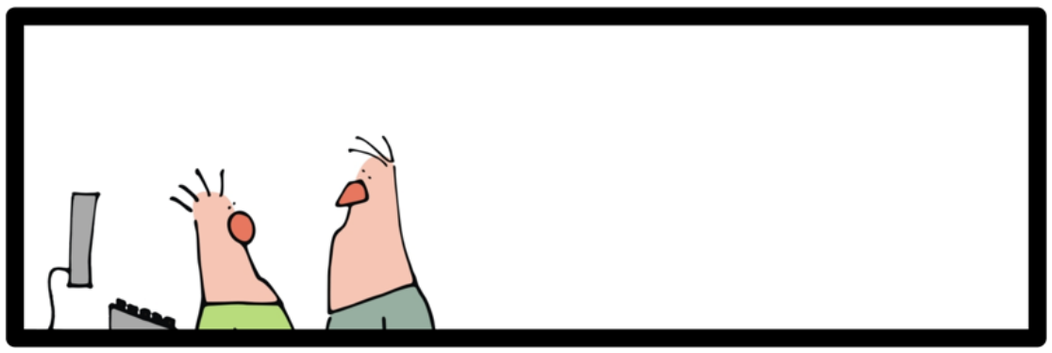
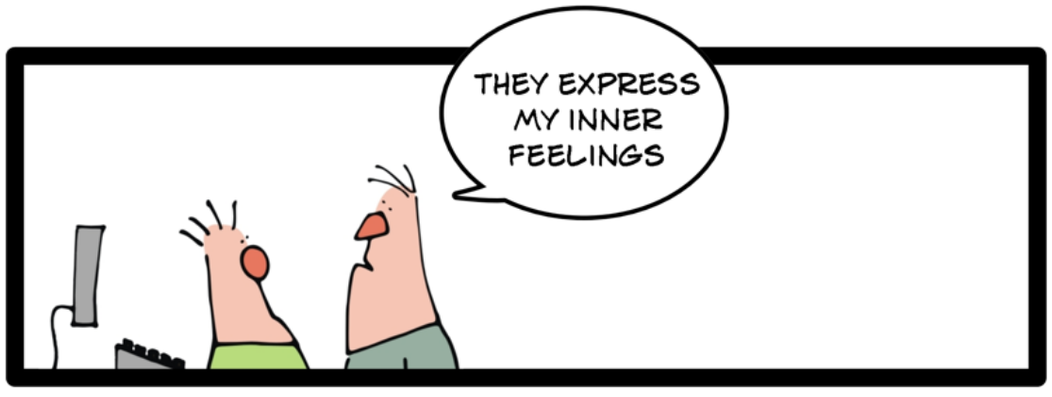
THE ART OF HIBERNATE OPTIMIZATION

Bryce Penberthy
Daugherty Business Solutions

THE ART OF PROGRAMMING







PROGRAMMERS ARE ARTISTS

TECHNIQUES

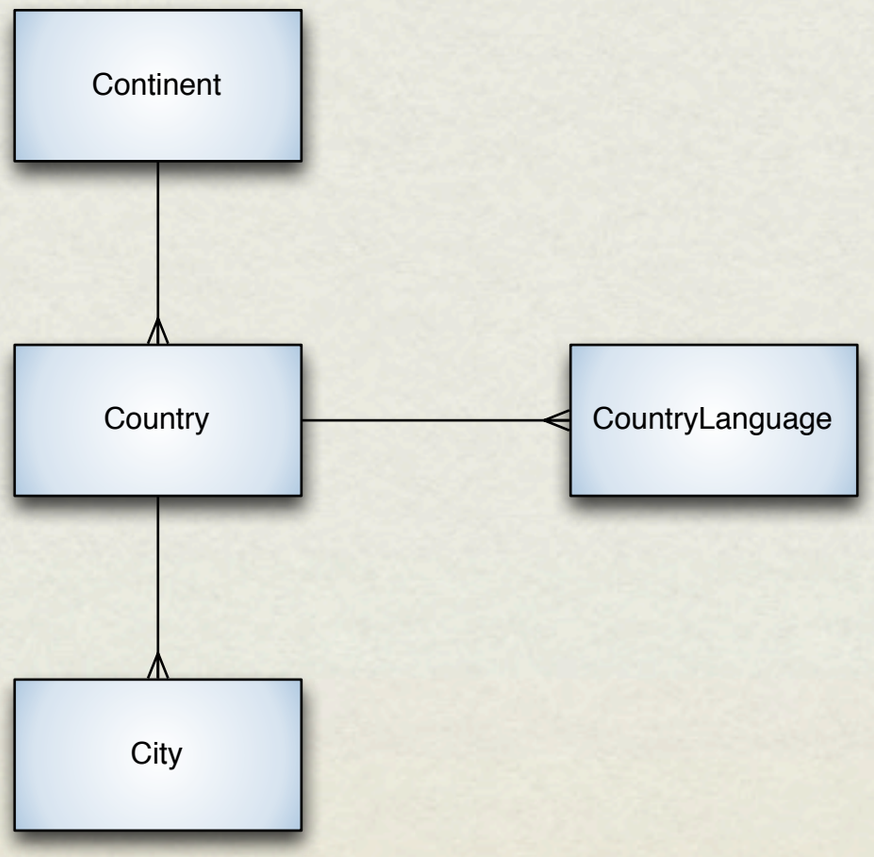
- Tune your HQL fetching strategy
- Use paging when needed
- Execute custom native SQL queries

YOUR CANVAS

- Install / Set up your favorite IDE
- For Eclipse, install eGit & m2e plugins
- Example code can be found on GitHub: https://github.com/brycep/world-example

EXAMPLE DATABASE

- MySQL world database example / Converted to HSQL
  http://dev.mysql.com/doc/world-setup/en/world-setup.html



FIRST STEPS
TAKE CONTROL OF YOUR SQL

- Turn on SQL output
- Modify HQL to reduce the number of database queries Hibernate needs to make.
- Force collections to be eager loaded by using JOINs

THE PRE-FETCH

From

```
@Override
public Country find(String countryName)  {
    return (Country) entityManager.createQuery(
                    "from Country country where country.name = :countryName")
            .setParameter("countryName", countryName).getSingleResult();
}
```

To

```
@Override
public Country find(String countryName)  {
    return (Country) entityManager.createQuery(
                    "select country “ +
                    “from Country country left join fetch country.cities “ +
                    “where country.name = :countryName")
            .setParameter("countryName", countryName).getSingleResult();
}
```

TURN THE PAGE

- Very large data sets can often times be broken up
- Returning partial results can improve response times

RETURNING PAGES

JPA

```
Query query = entityManager.createQuery("select c from City c");
return (List<City>)query.getResultList().subList(fromIndex, toIndex);
```

Hibernate

```
Query query = sessionFactory.getCurrentSession()
        .createQuery("select c from City c");
query.setFirstResult(fromIndex);
query.setMaxResults(toIndex - fromIndex);
return query.list();
```

DOWN TO THE WIRE

- Do you need a the power of a native query? Put it in Hibernate!

```
@NamedNativeQuery(name="findLanguagesForCountry",
            query="select CountryCode, Language, IsOfficial, Percentage from CountryLanguage where CountryCode = :countryCode",
            resultSetMapping="languageMapping")
@SqlResultSetMapping(name="languageMapping",
            entities={
                  @EntityResult(entityClass=CountryLanguage.class, fields= {
                        @FieldResult(name="pk.countryCode", column="CountryCode"),
                        @FieldResult(name="pk.language", column="Language"),
                        @FieldResult(name="isOfficial", column="IsOfficial"),
                        @FieldResult(name="percentage", column="Percentage")
                  })
})
```

THAT'S ALL FOR NOW!

- Feel free to check out the code, use it for your own apps, presentations or anything you want. Have fun!

- Any Questions?