CS544 EA

Hibernate

Alternate Query Types

Constraints

- Uses Java Objects to create a query
  - Use this **instead of concatenating JPQL strings**!
  - Has all the same features as JPQL

```
CriteriaBuilder cb = em.getCriteriaBuilder();
CriteriaQuery<Person> cq = cb.createQuery(Person.class);
Root<Person> person = cq.from(Person.class);
Join<Person, Address> address = person.join("address");
cq.where(
          cb.and(
                    cb.equal(person.get("firstName"), "John"),
                    cb.equal(address.get("city"), "Fairfield")
               )
     );

TypedQuery<Person> query = em.createQuery(cq);
List<Person> ppl = query.getResultList();
```

Same as JPQL:
from Person p where
p.firstName = ‘John’ and
p.address.city = ‘Fairfield’

For more info see: https://docs.oracle.com/javaee/6/tutorial/doc/gjrij.html

Stored Procedure Queries

- To execute a MySQL **stored procedure**:

```
create procedure calculate(
     IN x int, IN y int, OUT sum int, OUT prod int)
begin
     select x + y into sum;
     select x * y into prod;
end
```

```
StoredProcedureQuery query = em.createStoredProcedureQuery("calculate");

query.registerStoredProcedureParameter("x", Integer.class, ParameterMode.IN);
query.registerStoredProcedureParameter("y", Integer.class, ParameterMode.IN);
query.registerStoredProcedureParameter("sum", Integer.class, ParameterMode.OUT);
query.registerStoredProcedureParameter("prod", Integer.class, ParameterMode.OUT);

query.setParameter("x", 2);
query.setParameter("y", 3);
query.execute();

int sum = (int) query.getOutputParameterValue("sum");
int prod = (int) query.getOutputParameterValue("prod");
System.out.println("sum: " + sum + " prod: " + prod);
```

Native Queries

- Write SQL and receive Objects

```
Query query = em.createNativeQuery("SELECT * FROM Person", Person.class);
List<Person> ppl = query.getResultList();
ppl.forEach(x -> System.out.println(x.getFirstName()));
```

- Without the second parameter
  - native query returns `Object[]`

```
Query query = em.createNativeQuery("SELECT * FROM Person");
List<Object[]> ppl = query.getResultList();
ppl.forEach(x -> System.out.println(x[1]));
```