Skillbox

Hibernate

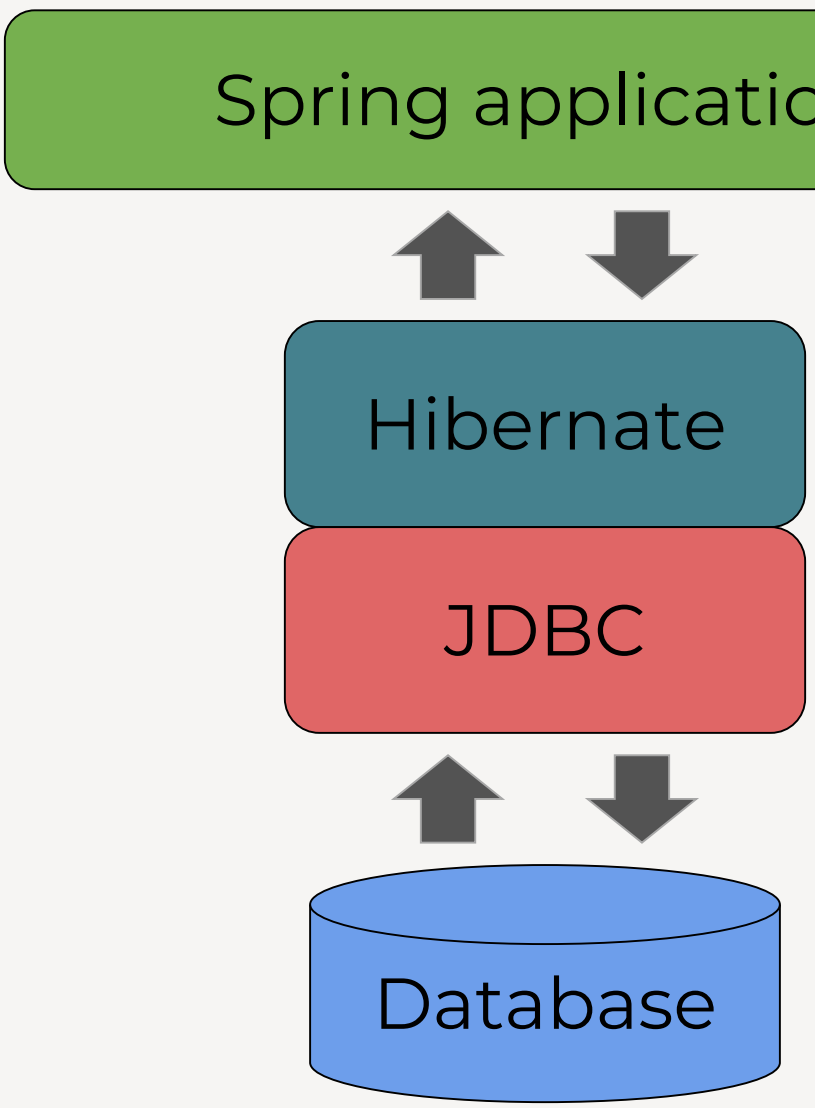
Кошаев К. А.

Java-разработчик

Skillbox

Hibernate framework

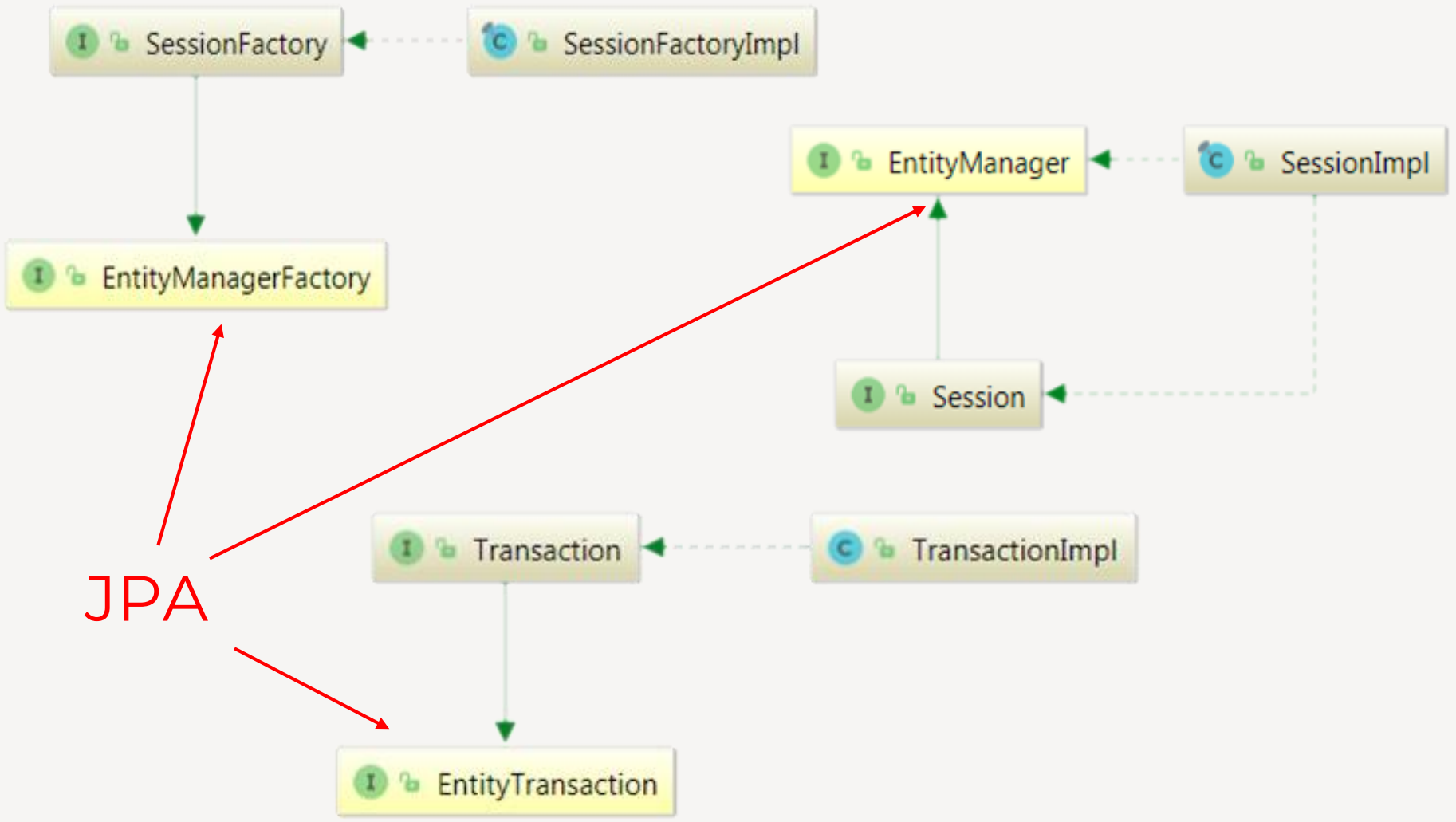
Hibernate — это java-фреймворк для ORM, т.е. объектно-реляционного отображения, который обеспечивает основу для сопоставления объектов приложения с таблицами реляционной базы данных в обоих направлениях



Skillbox

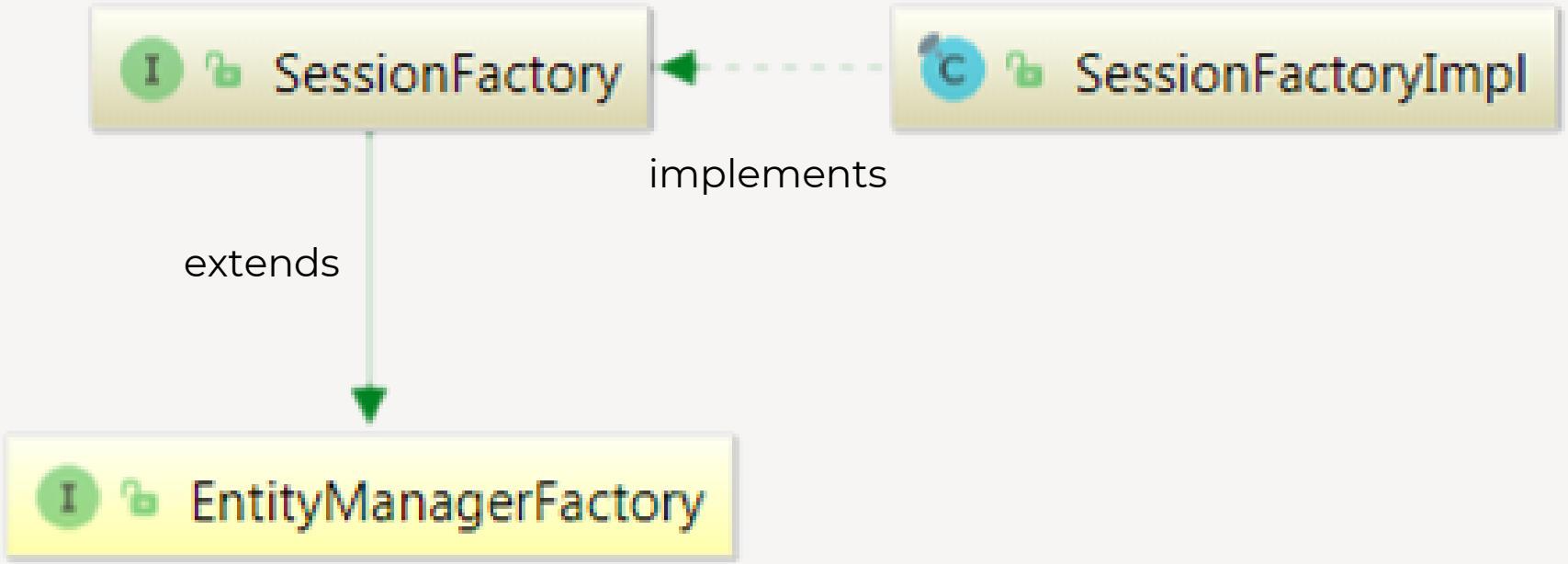
Архтектура Hibernate

Hibernate, как ORM-решение, эффективно «находится между» уровнем доступа к данным Java-приложения и реляционной базой данных. Приложение Java использует Hibernate API для загрузки, хранения данных, выполнения запросов и других операций с данными



SessionFactory

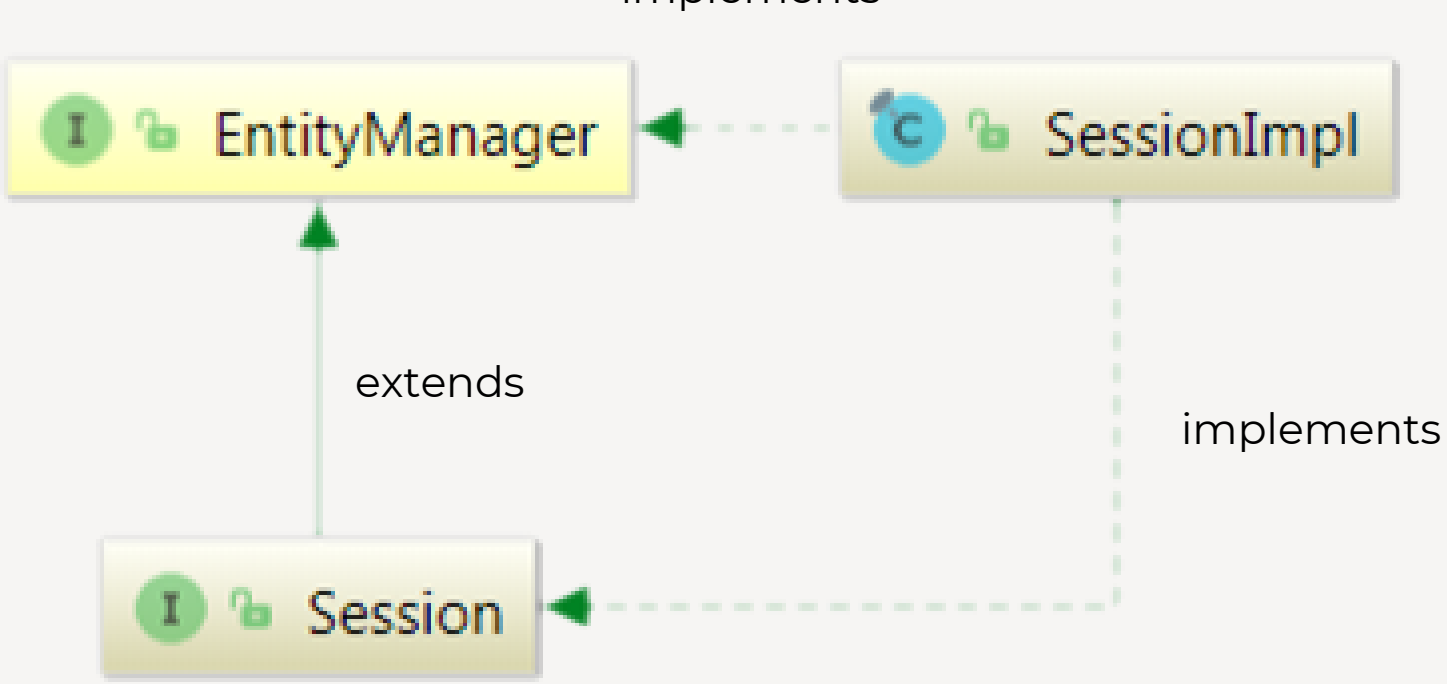
Интерфейс SessionFactory предоставляет потокобезопасное и неизменяемое сопоставление модели данных — entity-классов приложения с используемой для хранения этих данных базой



SessionFactory sessionFactory =
entityManagerFactory.unwrap(SessionFactory.class);

Session

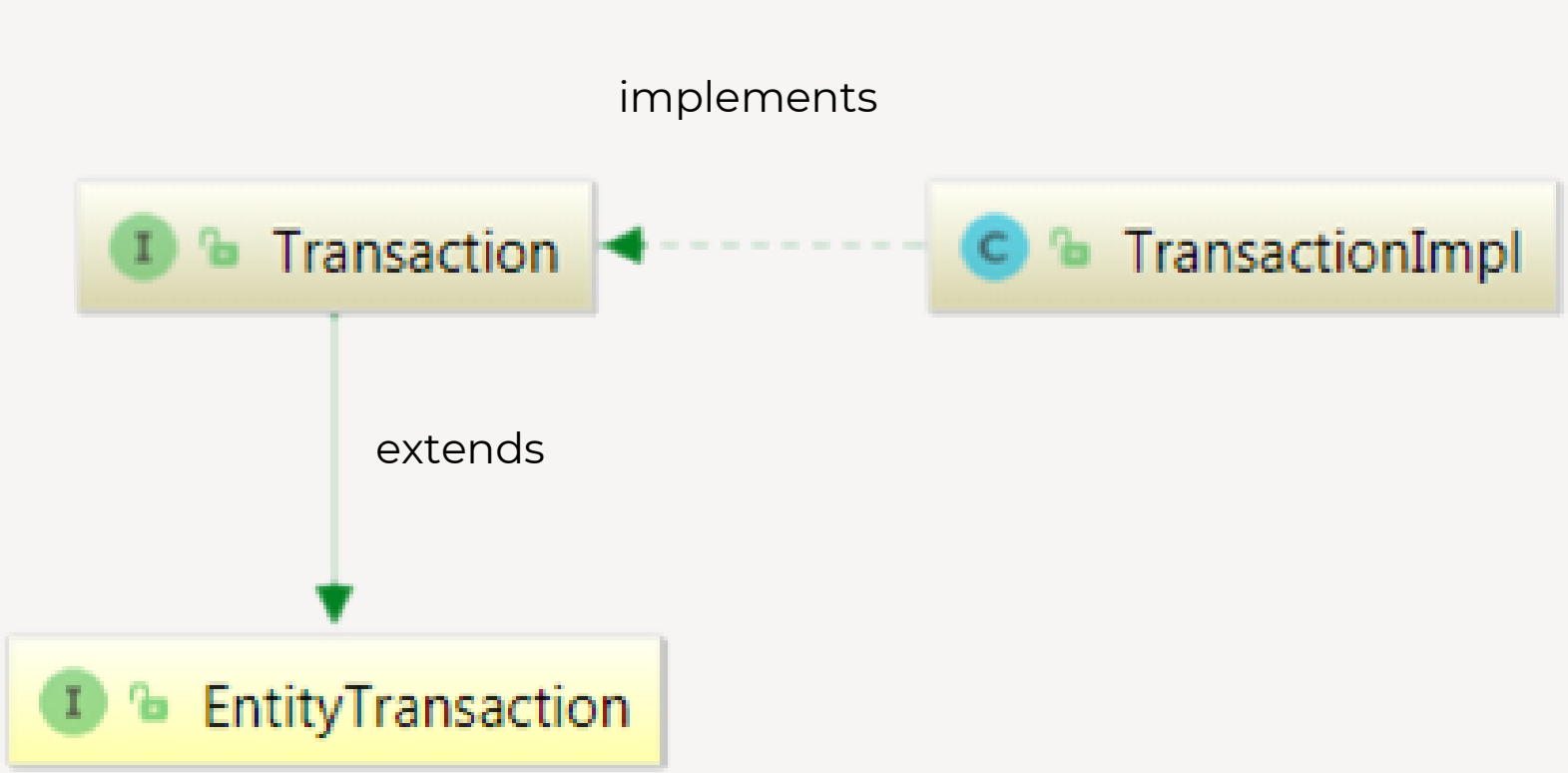
Session — это монопоточный объект с коротким жизненным циклом, концептуально моделирующий «единицу работы» в Hibernate



Session session = sessionFactory.openSession();

Transaction

Transaction также является монопоточным объектом с коротким жизненным циклом, который используется приложением для разграничения отдельных транзакций



Transaction

Transaction также является монопоточным объектом с коротким жизненным циклом, который используется приложением для разграничения отдельных транзакций

```
        Session session = factory.openSession();
Transaction tx = null;
Integer userID = null;
```

```
try {
    tx = session.beginTransaction();
    User user = new User(fname, lname, salary);
    userID = (Integer) session.save(user);
    tx.commit();
}catch(HibernateException e){
    if (tx!=null){tx.rollback();}
    e.printStackTrace();
}finally{
    session.close();
}
return userID;
```

Hibernate vs JDBC

- Hibernate исключает jdbc boilerplate
- Hibernate берёт на себя управление ресурсами
- Hibernate поддерживает lazy initialization запросов
- Механизмы кеширования Hibernate повышают производительность работы приложения с данными
- Hibernate поддерживает встроенное управление транзакциями

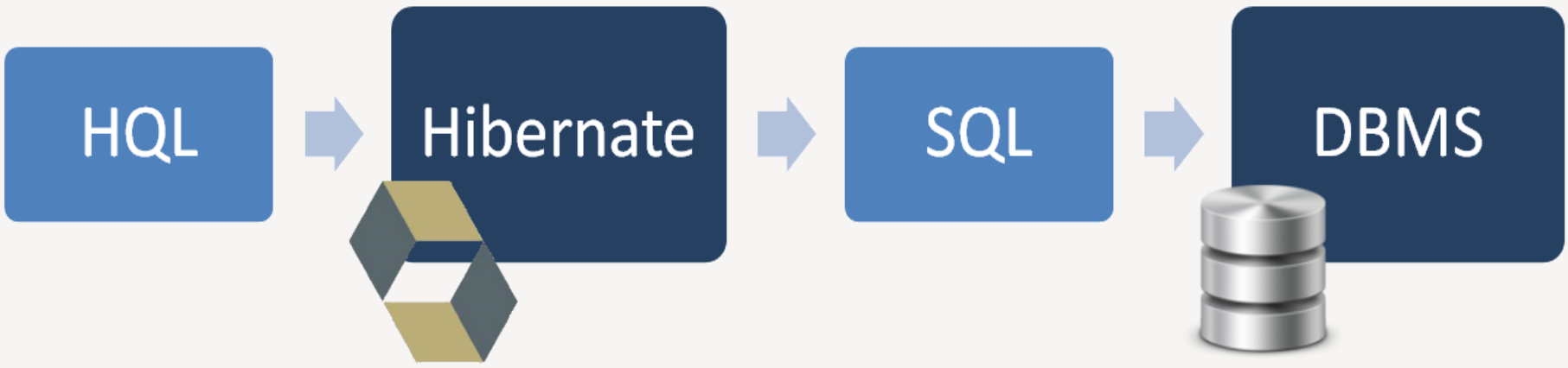
HQL

Hibernate Query Language (HQL) — это объектно-ориентированный язык запросов, похожий на SQL, но вместо работы с таблицами и столбцами HQL, работает с постоянными объектами и их свойствами

```
String hql = "FROM Student S WHERE S.id > 10 ORDER BY S.email DESC";

Query query = session.createQuery(hql);

List results = query.list();
```



Skillbox

HQL FROM

Hibernate Query Language (HQL) — это объектно-ориентированный язык запросов, похожий на SQL, но вместо работы с таблицами и столбцами HQL, работает с постоянными объектами и их свойствами

```
String hql = "FROM User";
Query query = session.createQuery(hql);
List results = query.list();


String hql = "FROM org.project.entity.User";
Query query = session.createQuery(hql);
List results = query.list();
```

Skillbox

HQL AS

Оператор AS можно использовать для присвоения псевдонимов entity-классам в рамках контекста запроса

```
String hql = "FROM BasicAuthStoreUser AS U";
Query query = session.createQuery(hql);
List results = query.list();


String hql = "FROM BasicAuthStoreUser U";
Query query = session.createQuery(hql);
List results = query.list();
```

Skillbox

HQL SELECT

Оператор SELECT обеспечивает больший контроль над набором результатов, чем оператор FROM

```
String hql = "SELECT User.name FROM User";
Query query = session.createQuery(hql);
List results = query.list();
```

```
String hql = "SELECT U.name FROM User U";
Query query = session.createQuery(hql);
List results = query.list();
```

Skillbox

HQL WHERE

Если понадобится собрать данные с учётом каких-то условий,
то для этого можно использовать в запросе оператор WHERE

```
String hql = "FROM User U WHERE U.id = 14";
Query query = session.createQuery(hql);
List results = query.list();
```

Skillbox

HQL ORDER BY

Чтобы отсортировать результаты HQL-запроса, можно использовать оператор ORDER BY. С его помощью можно упорядочивать результаты запросов по любому свойству фигурирующих в запросе persistence-объектов

```
String hql = "FROM User U WHERE U.age > 20 ORDER
+ BY U.rating DESC";
Query query = session.createQuery(hql);
List results = query.list();

String hql = "FROM User U WHERE U.age > 35 " +
"ORDER BY U.name DESC, U.rating ASC ";
Query query = session.createQuery(hql);
List results = query.list();
```

Skillbox

HQL GROUP BY

Оператор GROUP BY используется для определения групп выходных строк, к которым могут применяться агрегатные функции

```
String hql = "SELECT SUM(U.rating), U.name” +
“FROM User U GROUP BY U.name";
Query query = session.createQuery(hql);
List results = query.list();
```

Skillbox

HQL Named parameters

Hibernate поддерживает обработку именованных параметров в HQL-запросах

```
String hql = "FROM user E WHERE E.id = ” +
“:user_id";
Query query = session.createQuery(hql);
query.setParameter("user_id",10);
List results = query.list();
```

Skillbox

Спасибо за внимание!