

Домашнее задание:

Решите несколько практических заданий, связанных с использованием CriteriaBuilder в Hibernate для создания запросов с использованием WHERE, нескольких условий и ограничений результатов выборки:

Задание 1: Простой WHERE запрос

Создайте запрос с использованием CriteriaBuilder, который выберет все курсы с продолжительностью более 50 часов.

Задание 2: Сложное WHERE условие

Создайте запрос, который выберет учителей старше 30 лет и курсы с продолжительностью менее 40 часов.

Задание 3: Использование параметров

Создайте запрос, который выберет все курсы, у которых цена выше 120_000. Значение цены должно быть параметром, который передается в запрос при выполнении.

Задание 4: Ограничение результатов

Создайте запрос, который выберет первые 5 курсов, отсортированных по продолжительности в убывающем порядке.

Задание 5: Использование OR

Создайте запрос, который выберет курсы с продолжительностью более 40 часов или учителей старше 35 лет.

Задание 6: Использование BETWEEN

Создайте запрос, который выберет курсы с ценой, находящейся в определенном диапазоне от 50_000 до 150_000.

Попробуйте решить эти задания, и, если у вас возникнут вопросы или нужна помощь, не стесняйтесь спрашивать.

Ссылка на обновленный дамп базы top:

<https://drive.google.com/file/d/118XgthGRb1mhZ8VtqzUQUbT6TBBqvIM0/view?usp=sharing>

Также почитайте про **проблему N + 1**: [N+1 проблема в Hibernate - SYSOUT](#)

Проект с занятия:

```

import jakarta.persistence.criteria.CriteriaBuilder;
import jakarta.persistence.criteria.CriteriaQuery;
import jakarta.persistence.criteria.Root;
import models.Course;
import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.Transaction;
import org.hibernate.boot.Metadata;
import org.hibernate.boot.MetadataSources;
import org.hibernate.boot.registry.StandardServiceRegistry;
import org.hibernate.boot.registry.StandardServiceRegistryBuilder;

import java.util.List;

public class Main {
    public static void main(String[] args) {
        StandardServiceRegistry registry = new StandardServiceRegistryBuilder()
            .configure("hibernate.cfg.xml").build();
        Metadata metadata = new MetadataSources(registry).getMetadataBuilder().build();
        SessionFactory sessionFactory = metadata.getSessionFactoryBuilder().build();
        Session session = sessionFactory.openSession();

        Transaction transaction = session.beginTransaction();

        // todo Способ с hql
        String hql = "From " + Course.class.getSimpleName() + " Where price > 120000";
        List<Course> resultList = session.createQuery(hql).getResultList();
        resultList.forEach(course -> System.out.println(course.getPrice()));

        // todo Способ с Criteria API
        CriteriaBuilder builder = session.getCriteriaBuilder();
        CriteriaQuery<Course> query = builder.createQuery(Course.class);
        Root<Course> root = query.from(Course.class);
        // Запрос типа: select * from courses where price <= 100_000 order_by price
        DESC;
        query.select(root).where(builder.lessThanOrEqualTo(root.get("price"), 100_000))
        .orderBy(builder.desc(root.get("price")));

        // Запрос с несколькими условиями
        query.select(root)
            .where(builder.and(
                builder.lessThanOrEqualTo(root.get("price"), 10000),
                builder.lessThan(root.get("duration"), 18)
            ));

        // Сохранение результатов с ограничением результат запроса с помощью LIMIT
        (setMaxResults)
        List<Course> resultList =
        session.createQuery(query).setMaxResults(3).getResultList();

        // Вывод результатов с помощью forEach
        resultList.forEach(course ->
            System.out.println(course.getName() + " - " + course.getPrice() + " " +
            course.getDuration()));

        transaction.commit();
        session.close();
        sessionFactory.close();
    }
}

```