**Проблема N + 1**

У нас есть БД:

* Отношение Один ко многим.
* По молчанию — FetchType.LAZY.
* Вызов геттера на чеовее подгрузит связанные товары.

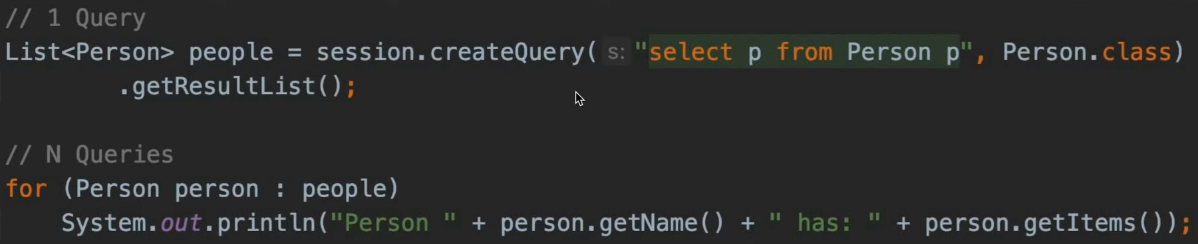
****

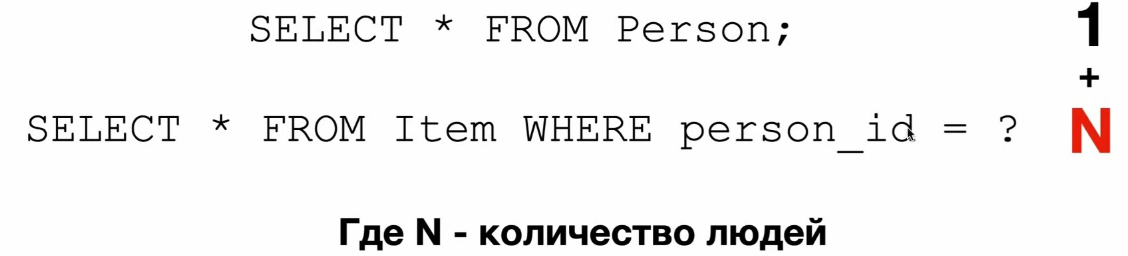
**Где появляется проблема n + 1 ?**

Предположим, что перед нами стоит следующая задача:

Необходимо получить товары для каждого человека и вывести их в консоль.

**Решение задачи:**

  
Сначала на сессии создадим какой нибудь HQL запрос для получения всех людей. Получим всех людей, а затем будем проходится с помощью цикла по всем этим людям и для каждого человека будем вызывать геттер и будем подгружать связанные товары. ЕСЛИ мы с помощью таких команд решим эту задачу, то у нас появится **проблема N + 1**. Потому-что у нас будет один запрос, чтобы получить всех людей, а затем после этого мы делаем **N** запросов в цикле для получения связанных сущностей.

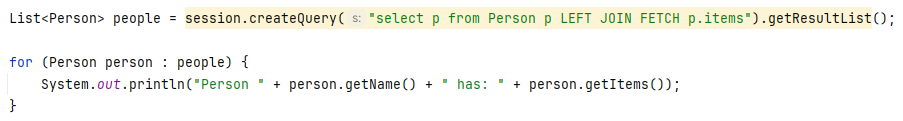


**Проблема заключается в том**, что если **N** большое (если много людей в базе данных) и для всех этих людей мы хотим получить список товаров, то эти запросы займут большое количество времени.

При этом **FetchType.EAGER** никак не поможет — все равно будут делаться **N** запросов к таблице **Item**, только теперь сразу после загрузки людей. Поэтому не ленивая загрузка эту проблему не решает.

**Решение проблемы N + 1**

Чтобы решить эту проблему, нам нужно делать объединение таблиц (например, **LEFT JOIN**) и работать уже с объедененой таблицей. Тоесть нужно сделать один запрос к базе данных где мы просто объеденяем таблицу **Person** с таблицей **Item**. В итоге получается одна большая объедененная таблица и там уже есть вся необходимая информация. Получается, что таким образом мы делаем не **N** запросов к базе данных, а делаем всего лишь один запрос.

  
В запросе **HQL** мы объеденяем две таблицы с помощью **LEFT JOIN FETCH**. Используя слово **FETCH** мы даём понять **Hibernate**`у, что мы не просто хотим соединить две таблицы, а еще хотим чтобы все данные из соединяемых таблиц были загружены сразу в оперативную память.