**Язык HQL**

Язык который используется в **hibernate**. Несмотря на то что **hibernate** берет на себя множество задач, всё равно иногда приходится писать запросы вручную. Обычно в **hibernate** мы пишем запросы не с помощью языка **SQL**, а с помощью специального языка **HQL**.

Примеры запросов когда нам потребуется использовать язык **HQL**:

**1.** Вывести всех людей, у которых возраст > 20

**2.** Удалить всех людей, у которых имя начинается на “Т”

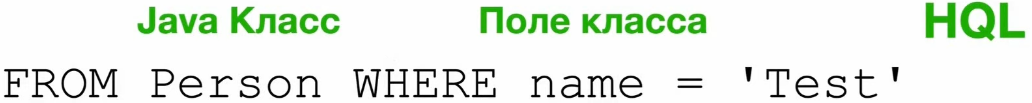
**3.** Обновить всех людей и всем поставить имя “Test”

**HQL и SQL** очень похожи, но есть различия. Главное различие в том, что **SQL** работает с таблицами базы данных, а язык **HQL** вообще не знает про таблицы в базе данных, он работает с сущностями(java классами). Тоесть в **HQL** мы не трогаем таблицы, которые у нас есть в базе данных, а работает исключительно с **java** объектами. (сущность эта те классы над которыми стоит аннотация **@Entity**).

**Пример кода SQL**



Пример кода **HQL**



**Свойства HQL**

* **HQL** был разработан, чтобы мы не спускались до уровня таблиц, а делали запрос используя наши сущности (**java** классы).
* Конвертируется в **SQL**
* В **hibernate** можно использовать и обычный **SQL** для сложных запросов, но лучше если это возможно использовать **HQL**.
* **Spring** приложение — это обычно комбинация **hibernate + Spring Data JPA** и **SQL** длянестандартных запросов**.**

**Примеры**

Здесь выбираем всех людей из таблицы **Person**, но в запросе указываем именно **java** класс.

Здесь выбираем всех людей у кого возраст больше 20, причем поле age это поле класса, а не столбец таблицы.

Здесь выбираем всех людей у кого имя начинается с буквы “T”.

Здесь обновляем имена на «TEST» всем людям кого возраст больше 30. Здесь используем метод ***executeUpdate()*** так как производим обновление таблицы базы данных.

Здесь удаляем всех людей у кого возраст больше 30. Здесь используем метод ***executeUpdate()*** так как производим обновление таблицы базы данных.