**Введение в Hibernate**

**ORM** **(Object-Relational Mapping)** - технология программирования, которая связывает базы данных с концепциями объектно-ориентированных языков программирования, создавая «виртуальную объектную базу данных» тоесть, отображение объектов в таблице и наоборот отображение таблиц в объекты. ORM это такая вещь которая берет на себя это отображение между объектами и таблицами в реляционных базах данных.

**JPA** – стандартная спецификация для **java**, которая предназначена для того, чтобы сохранять объеты в базы данных. А **Hibernate** это всего лишь один из **JPA** библиотек для **java**. Есть еще много других библиотек которые занимаются сохранением **java** объектов в базу данных. Все эти библиотеки подчиняются единой спецификации — спецификации **JPA**.

**Hibernate** – это одна из реализаций  **ORM**. **Hibernate –** это **ORM** котоырй за нас автоматищзирует отображение java объектов в строки в таблице базы данных. Тоесть сейячас не нужно букдет писать SQL код для работы с базой данных.

Под капотом у **Hibernate** тот же самый JDBC API, но он за нас берет написания SQL кода.

**Простой пример для сохранения объекта в таблице БД:**



Чтобы сделать запрос с помощью **Hibernate** необходимо сначала получить **Session**.

**Session** -это объект для взаимодействия с **Hibernate.** Когда хотим что-то делат ьс БД чепрез **Hibernate –** получаем сессию. Объект **Session** получаем из объекта **SessionFactory**(паттерн Фаброика). На объекте **Session** можно вызвать:

* **save**
* **upgrade**
* **get**
* **и т. д.**

**Транзакция**

Перед тем как сохранить что то, мы видим, что мы начинаем транзакцию тоесть, вызываем метод ***beginTransaction()*** на объекте **Session** и после того как вызвали метод ***save()*** – транзакцию завершаем. Между началом транзакции и завершением транзакции у нас есть какая то опреация. Если мы в одну транзакцию поместили несколько операций с базой данных, то все эти операции с базой данных будут восприниматься как одна большая операция с БД.

**Зачем нужно помещать несколько операций с БД в одну транзакцию ?**

**1.** Нужныо для поддержания согласованности данных в таблице.

**2.** Транзакции могут блокировать доступ к БД (пример: пока один пользователь пишет в таблицу данные, из нее нельзя читать данные, обновлять, удалять и т.д.)

**3.** Транзакции можно применять (**COMMIT**) и откатывать (**ROLLBACK**).