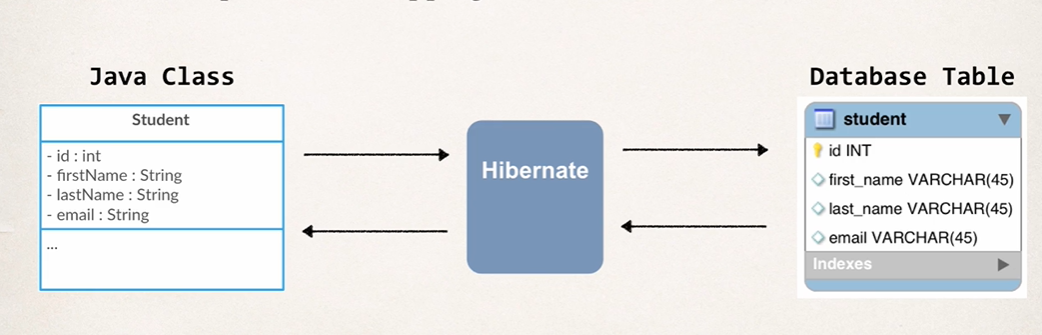
**What is ORM?**

ORM stands for object relational mapping.it is the process of mapping a Java class to a database table and its fields to perform database operations.



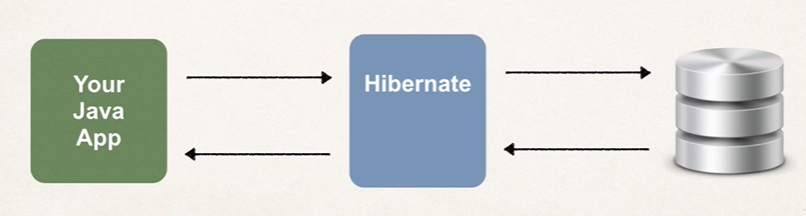
**What is Spring ORM?**

Just like how Spring made it easy for us to use JDBC by providing a Spring JDBCtemplate.

It also makes it easy to use ORM tools like Hibernate by providing **SessionFactory.**

**What is hibernate?**

A framework for saving java objects in database.



**Benefits of Hibernate:**

* Hibernate handles all of the low-level sql queries
* Minimize the amount of JDBC code you have to develop
* Hibernate provide object relational Mapping

**Use of SessionFactory**

--Reads the hibernate config file

---Creates Session objects

---only creates once in app

**Use of Session**

Wraps a JDBC connection

Main object used to save/retrieve objects

Retrieved from SessionFactory

**Steps to implements Hibernate-ORM tool**

* + - * Required dependencies for hibernate implementation

1. Spring-core—5.3.25
2. Spring-context
3. my-connector-java
4. Hibernate-core
5. Spring-orm

* Create Database
* Create table
* Provide hibernate configuration
  1. LocalSessionFactoryBean-
     1. Datasource
     2. PackagesToScan
     3. HibernateProperties
* Create Entity class
* Mapped Entity class fields with Database table columns using JPA Annotations

@Table

@Column

@ID

@Enitiy

Q. Why we are using JPA Annotation instead of Hibernate ?

Answer: JPA is a standard specification. Hibernate is an implementation of the JPA specification.

Hibernate implements all of the JPA annotations.

**The Hibernate team recommends the use of JPA annotations as a best practice.**

* Create dao layer
* Create service layer
* Create Test class