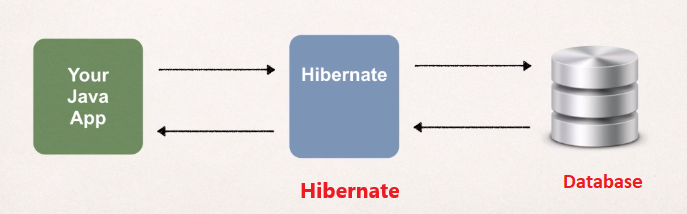
**18. Introduction to Hibernate**

**Hibernate**:

Hibernate is a framework for persisting / saving Java objects in a database.

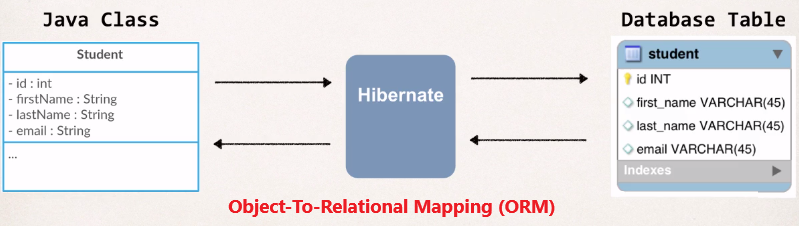


**Benefits of Hibernate**:

1. Hibernate handles all of the low-level SQL
2. Minimizes the amount of JDBC code we have to develop
3. Hibernate provides the Object-to-Relational Mapping (ORM)

**Object-To-Relational Mapping (ORM)**:

The developer defines mapping between Java class (**Student.class**) and database table. Here we have a Java class that has four field “id”, “firstName”, “lastName”, and “email”. We have the Hibernate framework in the middle and at the right we have the actual database table. The name of the table is **Student** and it has four fields “id” (primary key), “first\_name”, “last\_name”, and “email”. Here Hibernate creates one-to-one mapping between java **Student.class** and database **student** table.



Now we can setup this mapping by using a configuration file using XML or by using Java annotations.

**Saving a Java Object with Hibernate**:

// create Java object

Student theStudent = **new** Student(“John”, “Doe”, “john@luv2code.com");

// save it to database

**int** theId = (Integer) session.save(theStudent);

Here **Session** is a special hibernate object. And in the background Hibernate will store the data into the database. Here ***session.save()*** return an int value.

**Retrieving a Java Object with Hibernate**:

// create Java object

Student theStudent = **new** Student(“John”, “Doe”, “john@luv2code.com");

// save it to database

**int** theId = (Integer) session.save(theStudent);

// now retrieve from database using the primary key

Student myStudent = session.get(Student.**class**, theId);

**Querying for Java Objects**:

Query query = session.createQuery("from Student");List<Student> students= query.list();

Return a list of Student object from the database.

18. Introduction to Hibernate