INTRODUCTION

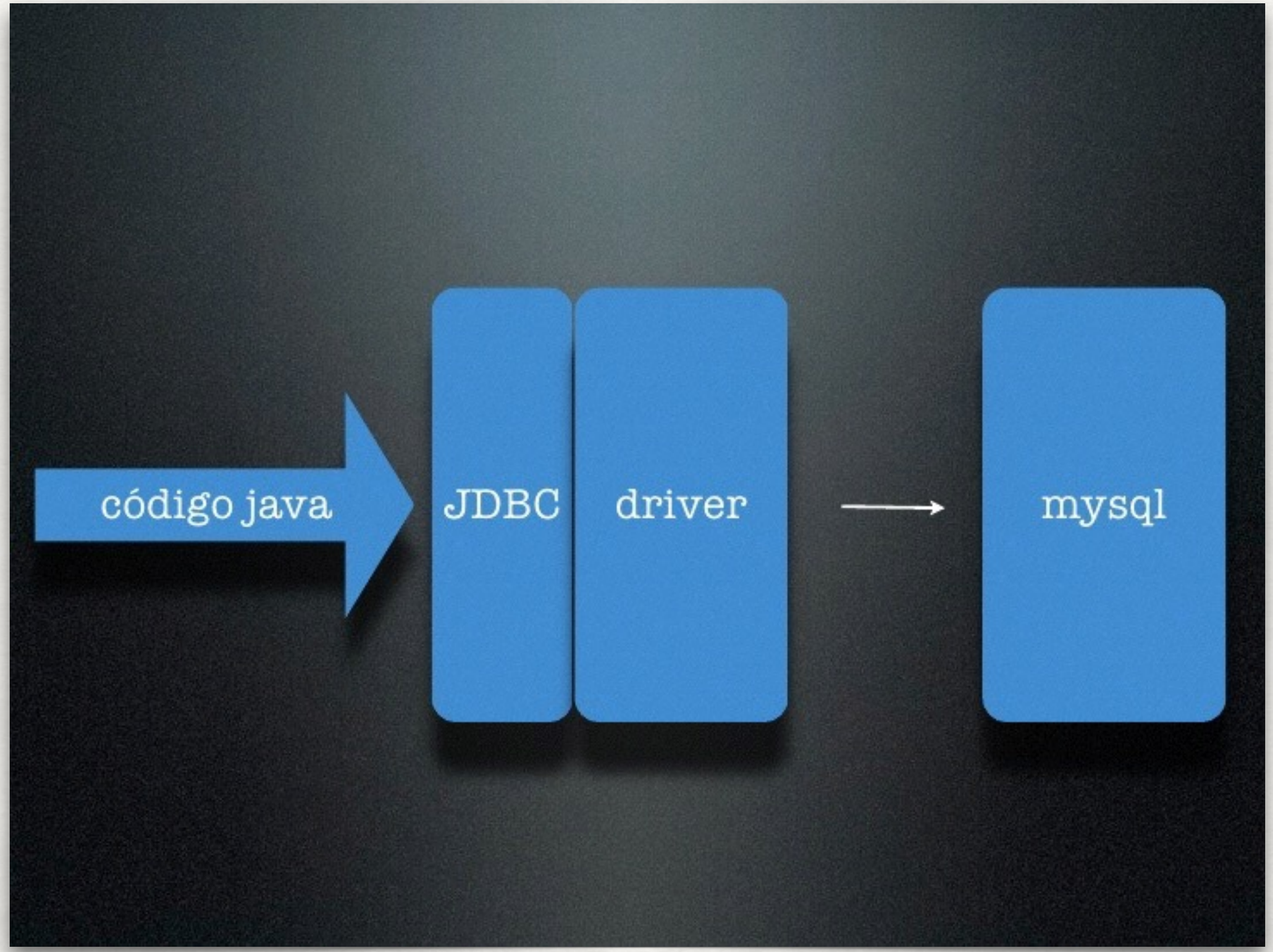
JPA + HIBERNATE

Marcus Piancó - macpj@ic.ufal.br

AGENDA

- MOTIVATION
- WHY TO USE JPA+HIBERNATE?
- WHAT IS JPA AND HIBERNATE?
- PROJECT
  - INSTALLATION
  - CONFIGURATION
  - FIRST EXAMPLE
  - EXTENDED FIRST EXAMPLE
  - SOME ANNOTATIONS
  - RELATIONSHIPS

MOTIVATION

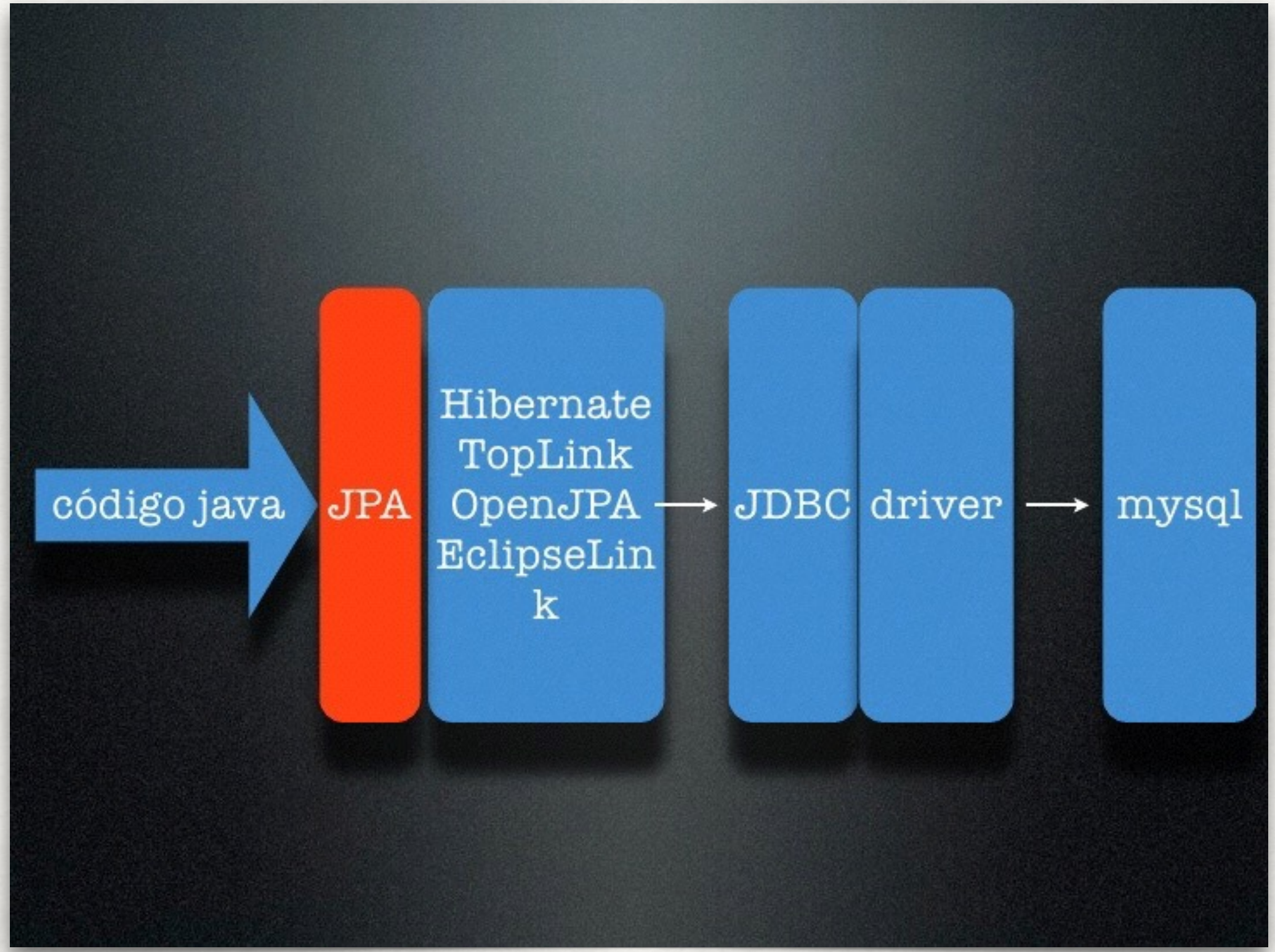


MOTIVATION

```java
import java.sql.*;

public class FirstExample {
   // JDBC driver name and database URL
   static final String JDBC_DRIVER = "com.mysql.jdbc.Driver";
   static final String DB_URL = "jdbc:mysql://localhost/EMP";

   //  Database credentials
   static final String USER = "username";
   static final String PASS = "password";

   public static void main(String[] args) {
   Connection conn = null;
   Statement stmt = null;
   try{
      //STEP 2: Register JDBC driver
      Class.forName("com.mysql.jdbc.Driver");

      //STEP 3: Open a connection
      System.out.println("Connecting to database...");
      conn = DriverManager.getConnection(DB_URL,USER,PASS);

      //STEP 4: Execute a query
      System.out.println("Creating statement...");
      stmt = conn.createStatement();
      String sql;
      sql = "SELECT id, first, last, age FROM Employees";
      ResultSet rs = stmt.executeQuery(sql);

      //STEP 5: Extract data from result set
      while(rs.next()){
         //Retrieve by column name
         int id  = rs.getInt("id");
         int age = rs.getInt("age");
         String first = rs.getString("first");
         String last = rs.getString("last");

         //Display values
         System.out.print("ID: " + id);
         System.out.print(", Age: " + age);
         System.out.print(", First: " + first);
         System.out.println(", Last: " + last);
      }
```

```java
//STEP 6: Clean-up environment
      rs.close();
      stmt.close();
      conn.close();
   }catch(SQLException se){
      //Handle errors for JDBC
      se.printStackTrace();
   }catch(Exception e){
      //Handle errors for Class.forName
      e.printStackTrace();
   }finally{
      //finally block used to close resources
      try{
         if(stmt!=null)
            stmt.close();
      }catch(SQLException se2){
      }// nothing we can do
      try{
         if(conn!=null)
            conn.close();
      }catch(SQLException se){
         se.printStackTrace();
      }//end finally try
   }//end try
   System.out.println("Goodbye!");
}//end main
}//end FirstExample
```

WHY TO USE JPA + HIBERNATE



WHAT IS JPA AND HIBERNATE?

JPA

Java Persistence API

JPA é uma camada que descreve uma interface comum para frameworks ORM.

HIBERNATE

O Hibernate é um framework ORM, ou seja, a implementação física do que você usará para persistir, remover, atualizar ou buscar dados no SGBD.

INSTALLATION

https://github.com/MarcusPianco/jpa-hibernate-curse

CONFIGURATION

- Configure Library Path (pom.xml/Maven);
- Configure MySql, User(root) and Data Base;
  - user=“root”;
  - password=“admin”.
- Configure hibernate.cfg.xml (show with more details);
- Configure components and modules.

FIRST EXAMPLE

FIRST EXAMPLE

```java
@Entity
public class User {

    private int id;

    private Calendar birthDay;

    private Integer age;

    private Status status;

    private String name;

    @Temporal(TemporalType.DATE)
    public Calendar getBirthDay()
{
        return birthDay;
    }
```

```java
public static void main(String[] args) {

        // Criando Novo Usuário
        User user = new User();

        user.setId(1);

        user.setName("Baldoino Neto");
        user.setAge(20);
        user.setLogUser(Calendar.getInstance());
        user.setStatus(Status.activate);

        // Cria uma Sessão de Comunicação com o Banco
                SessionFactory sessionFactory = new
Configuration().configure("./META-INF/
hibernate.cfg.xml").buildSessionFactory();

                //Abrindo uma sessão junto ao Banco ed Dados
                Session session =
sessionFactory.openSession();

                session.beginTransaction();

                //Salvar no Banco
                session.save(user);

                //Enviar a sessão para o banco
                session.getTransaction().commit();

                //Fecha a sessão com o banco
                session.close();
    }
```

FIRST EXAMPLE

```
Jul 31, 2016 7:51:30 PM org.hibernate.Version logVersion
INFO: HHH000412: Hibernate Core {5.0.0.Final}
Jul 31, 2016 7:51:30 PM org.hibernate.cfg.Environment <clinit>
INFO: HHH000206: hibernate.properties not found
Jul 31, 2016 7:51:30 PM org.hibernate.cfg.Environment buildBytecodeProvider
INFO: HHH000021: Bytecode provider name : javassist
Jul 31, 2016 7:51:30 PM org.hibernate.annotations.common.reflection.java.JavaReflectionManager <clinit>
INFO: HCANN000001: Hibernate Commons Annotations {5.0.0.Final}
Jul 31, 2016 7:51:30 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl configure
WARN: HHH000402: Using Hibernate built-in connection pool (not for production use!)
Jul 31, 2016 7:51:30 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl buildCreator
INFO: HHH000401: using driver [com.mysql.jdbc.Driver] at URL [jdbc:mysql://localhost:3306/curse]
Jul 31, 2016 7:51:31 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl buildCreator
INFO: HHH000046: Connection properties: {user=root, password=****}
Jul 31, 2016 7:51:31 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl buildCreator
INFO: HHH000006: Autocommit mode: false
Jul 31, 2016 7:51:31 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl configure
INFO: HHH000115: Hibernate connection pool size: 20 (min=1)
Jul 31, 2016 7:51:31 PM org.hibernate.dialect.Dialect <init>
INFO: HHH000400: Using dialect: org.hibernate.dialect.MySQLDialect
Jul 31, 2016 7:51:31 PM org.hibernate.tool.hbm2ddl.SchemaUpdate execute
INFO: HHH000228: Running hbm2ddl schema update
Hibernate: select next_val as id_val from hibernate_sequence for update
Hibernate: update hibernate_sequence set next_val= ? where next_val=?
Hibernate: insert into User (birthDay, name, status, id) values (?, ?, ?, ?)
```

EXTENDED

FIRSTEXAMPLE

SOME ANNOTATIONS

@Entity = Declares the class as an entity

@Id = Declares the identifier property of this entity

@Table = Add characteristics to the table/Entity

@Temporal = Used to define date format to persistence.

@Transient = Used in data that are used in operations in run time.

@Enumerated = Change enumeration persistence (String,…)

@OneToOne= Relationship 1….1

@OneToMany = RelationShip: 1….n

@ManyToOne = Relationship: n….1

@ManyToMany = Relationship: n …. n

RELATIONSHIP

- @OneToOne
- @OnToMany
- @ManyToOne
- @ManyToMany

QUESTIONS?