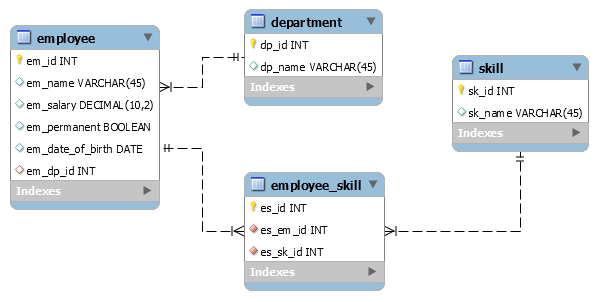
**Hands on 3**

**Create payroll tables and bean mapping**   
  
To demonstrate one to many, many to one and many to many relationships in Hibernate, a schema with entities employee, department and skill will be used. In this hands on we will setup the tables and data, which forms the basis for learning the mappings in Hibernate.  
  
**Schema Structure**  
  
  
  
Follow steps below to create necessary tables:

* Open mysql client in command line
* Use the source command to execute the payroll.sql script file available in spring-data-jpa-files folder. The following command assumes that spring-data-jpa-files folder is in D:.

mysql> source D:\spring-data-jpa-files\payroll.sql

Define bean mapping

* Open orm-learn project in Eclipse
* Create model classes Employee, Department and Skill in com.cognizant.orm-learn.model package
* Define each model should have @Entity and @Table annotations.
* Each id field should be have @Id annotation and @GeneratedValue(strategy = GenerationType.IDENTITY) annotation. @GeneratedValue annotation ensures auto increment of id creation.
* Define @Column against each field.
* Define getters, setters and toString() methods
* Employee
  + private int id;
  + private String name;
  + private double salary;
  + private boolean permanent;
  + private Date dateOfBirth;
* Department
  + private int id;
  + private String name;
* Skill
  + private int id;
  + private String name;
* Create appropriate repository interfaces EmployeeRepository, DepartmentRepository and SkillRepository in repository package

**CODE:**

**payroll.sql**

CREATE TABLE department ( id INT AUTO\_INCREMENT PRIMARY KEY, name VARCHAR(100) NOT NULL );

CREATE TABLE employee ( id INT AUTO\_INCREMENT PRIMARY KEY, name VARCHAR(100), salary DOUBLE, permanent BOOLEAN, date\_of\_birth DATE, department\_id INT, FOREIGN KEY (department\_id) REFERENCES department(id) );

CREATE TABLE skill ( id INT AUTO\_INCREMENT PRIMARY KEY, name VARCHAR(100) );

CREATE TABLE employee\_skill ( employee\_id INT, skill\_id INT, PRIMARY KEY (employee\_id, skill\_id), FOREIGN KEY (employee\_id) REFERENCES employee(id), FOREIGN KEY (skill\_id) REFERENCES skill(id) );

## **Employee.java**

@Entity @Table(name = "employee") public class Employee { @Id @GeneratedValue(strategy = GenerationType.IDENTITY) private int id;

@Column(name="name")  
private String name;  
  
@Column(name="salary")  
private double salary;  
  
@Column(name="permanent")  
private boolean permanent;  
  
@Column(name="date\_of\_birth")  
private Date dateOfBirth;  
  
@ManyToOne  
@JoinColumn(name="department\_id")  
private Department department;  
  
@ManyToMany  
@JoinTable(  
 name="employee\_skill",  
 joinColumns=@JoinColumn(name="employee\_id"),  
 inverseJoinColumns=@JoinColumn(name="skill\_id")  
)  
private Set<Skill> skills;  
 }

## **Department.java**

@Entity

@Table(name = "department")

public class Department {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private int id;

@Column(name="name")

private String name;

@OneToMany(mappedBy="department")

private List<Employee> employees;

}

## **Skill.java**

@Entity

@Table(name="skill")

public class Skill {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private int id;

@Column(name="name")

private String name;

@ManyToMany(mappedBy="skills")

private Set<Employee> employees;

}

**EmployeeRepository.java**

public interface EmployeeRepository extends JpaRepository<Employee, Integer> {

}

**DepartmentRepository.java**

public interface DepartmentRepository extends JpaRepository<Department, Integer> {

}

**SkillRepository.java**

public interface SkillRepository extends JpaRepository<Skill, Integer> {

}

**OUTPUT:**

INFO Spring Data JPA repositories initialized

INFO Hibernate validated successfully

INFO Tomcat started on port 8080

INFO Application started