Persistencia en BD

Taller de Programación

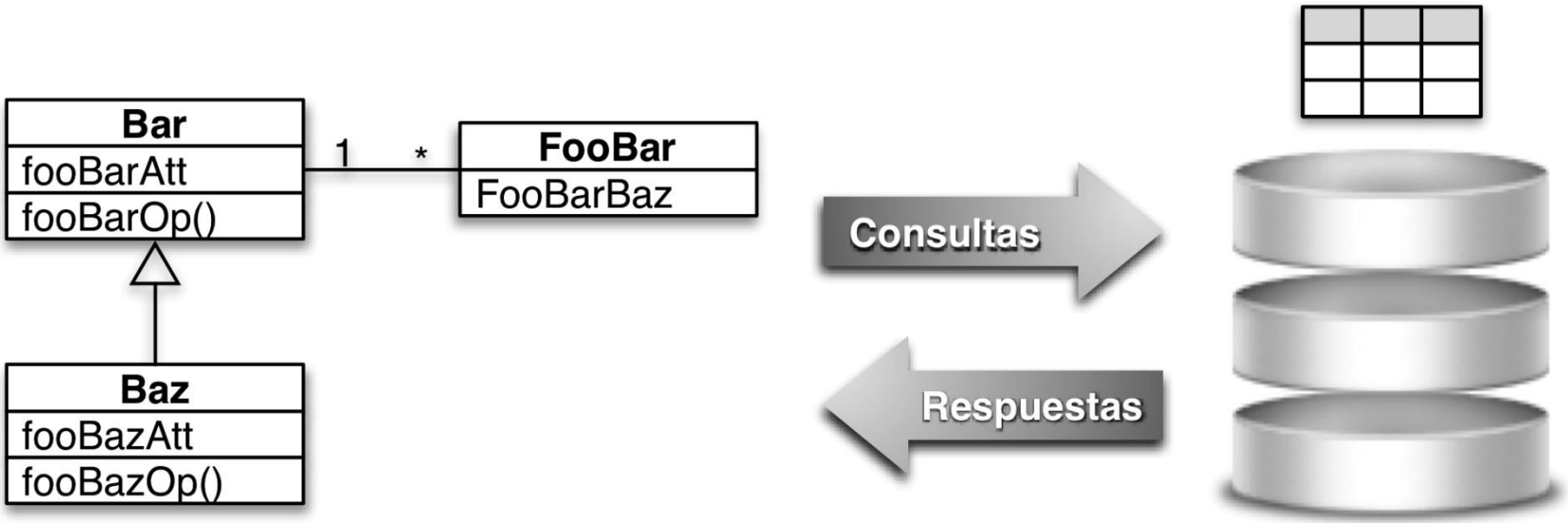
Instituto de Computación –Facultad de Ingeniería
Universidad de la República

Contenido

- Mecanismos de Persistencia
  - Acceso directo a Base de datos (JDBC)
    - Ejemplo
  - Impedance Mismatch
  - Mapeador Objeto Relacional (JPA)
    - Ejemplo

Mecanismos de Persistencia

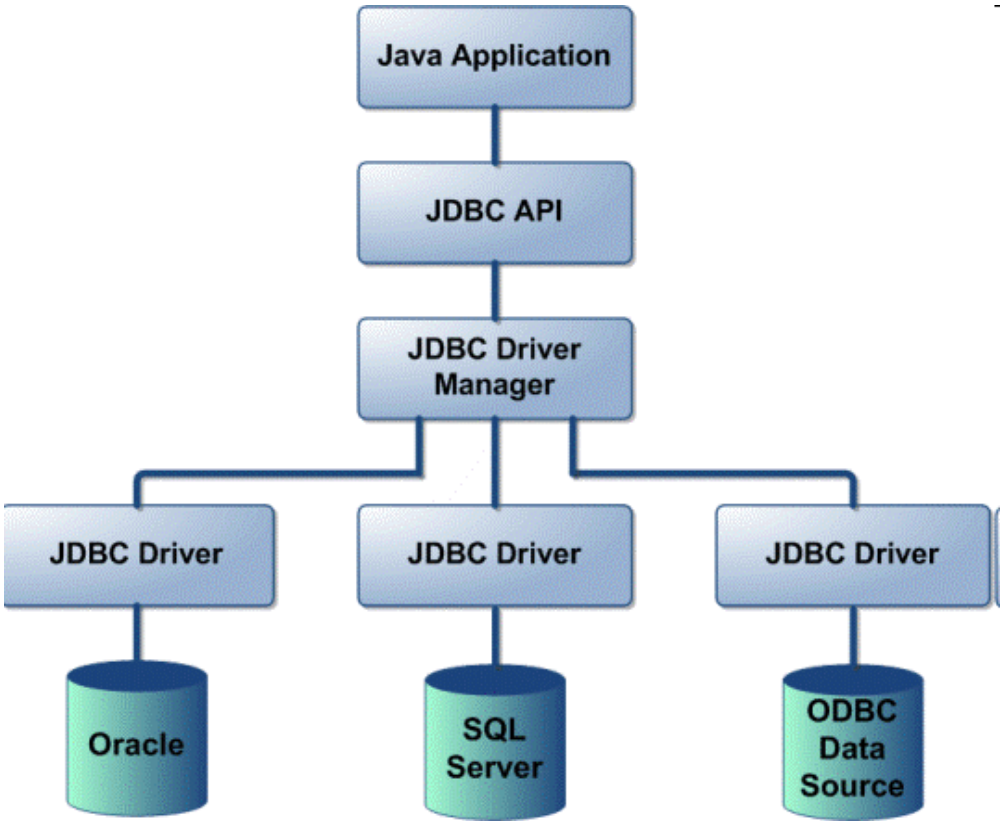
- Acceso directo a base de datos



Mecanismos de Persistencia

- Acceso directo a Base de Datos
  - Ventaja
    - Buen desempeño debido a que el acceso a la base se realiza en forma directa
  - Desventaja
    - Mayor complejidad para el diseño de la persistencia.
  - Herramienta Java
    - JDBC (Java Database Connectivity)

Arquitectura JDBC



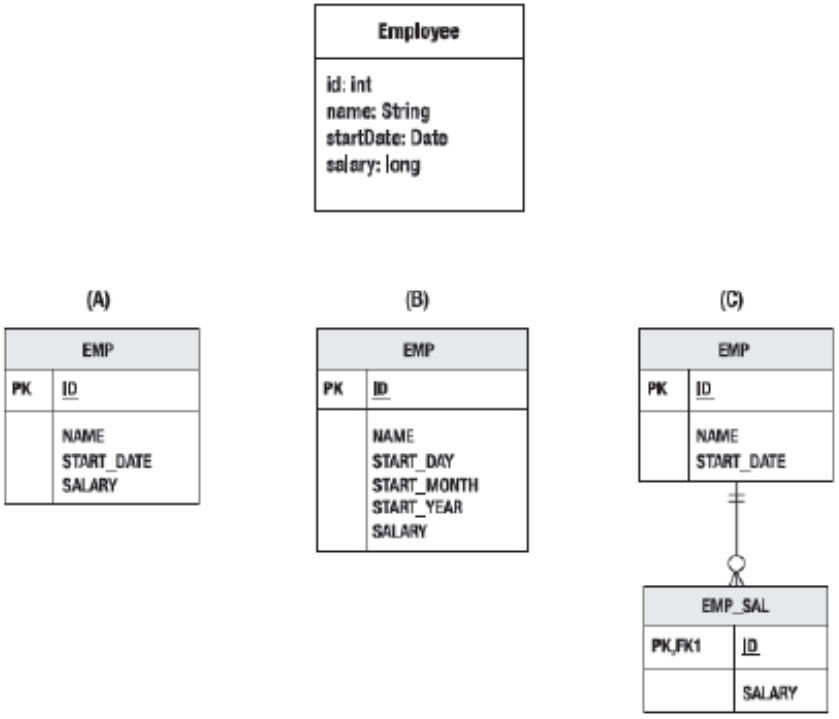
JDBC - Ejemplo

```
Class.forName("org.hsqldb.jdbcDriver");
connection = DriverManager.getConnection("jdbc:hsqldb:" + database, "sa", "");

Statement statement = null;
ResultSet resultSet = null;

statement = connection.createStatement();
resultSet = statement.executeQuery(sql);

while (resultSet.next()) {
    System.out.println(resultSet.getString("title") + " (" +
                       resultSet.getString("url") + ")");

}

resultSet.close();
statement.close();
connection.close();
```

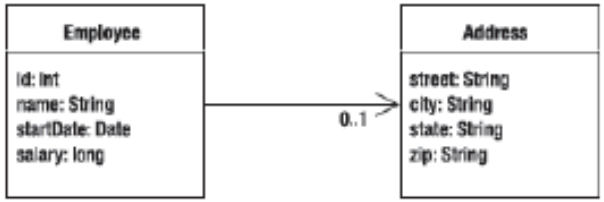
JDBC - Diseño

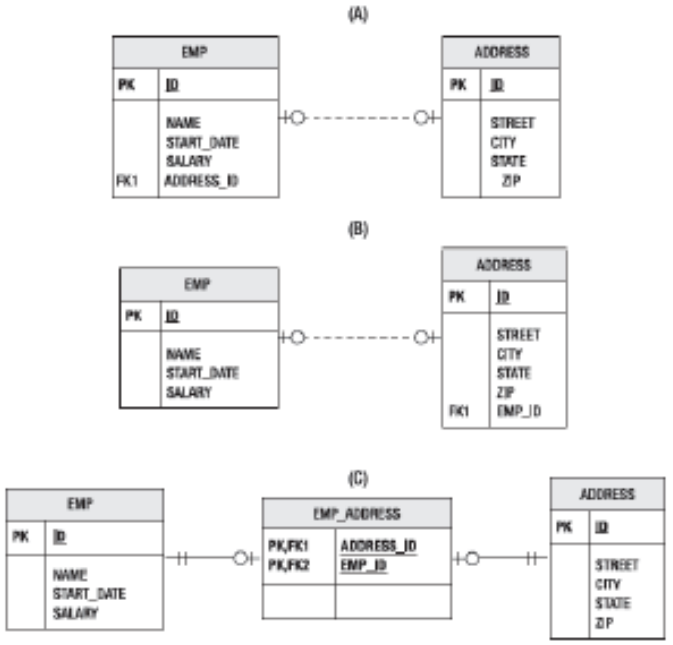
- Patrón de Diseño DAO:
  - Encapsula todo el acceso a una fuente de datos utilizando la capa de acceso a datos.
  - Cada objeto de negocio a ser persistido crea un DAO con la información necesaria.
  - Luego utiliza el objeto DAO para obtener y guardar información del origen de datos.
- Problemas al tener que “mapear” entidades
  - Impedance Mismatch

Impedance Mismatch

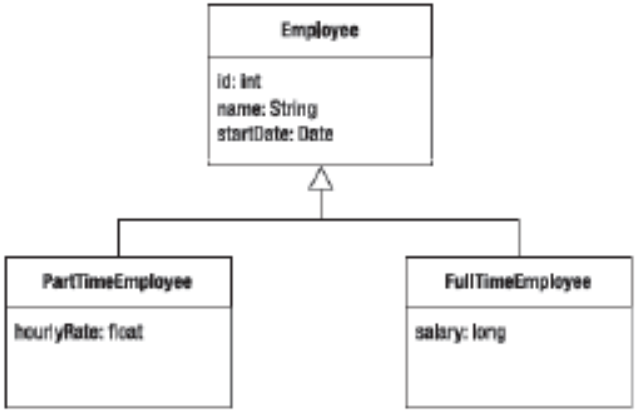


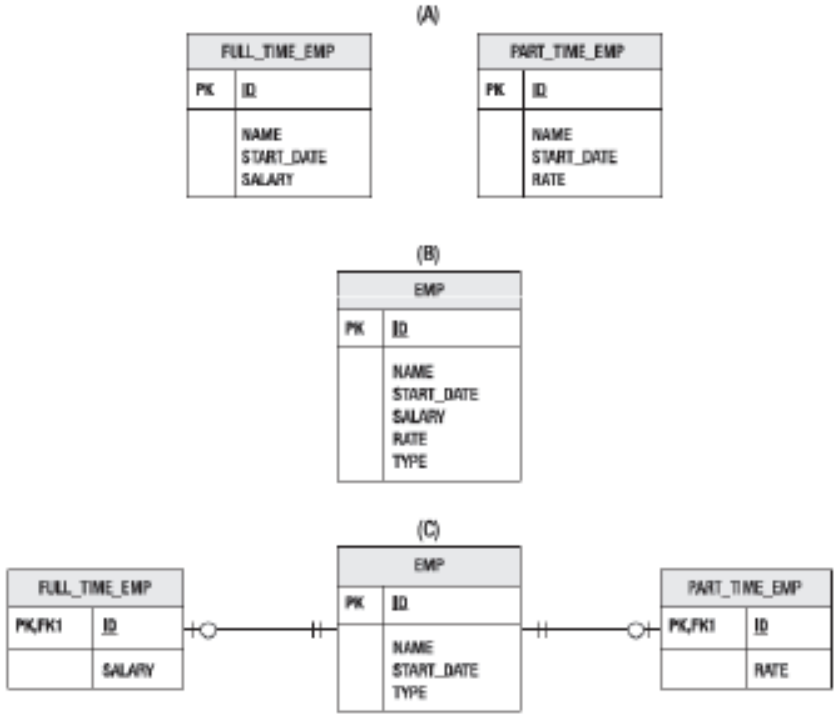
Impedance Mismatch





Impedance Mismatch



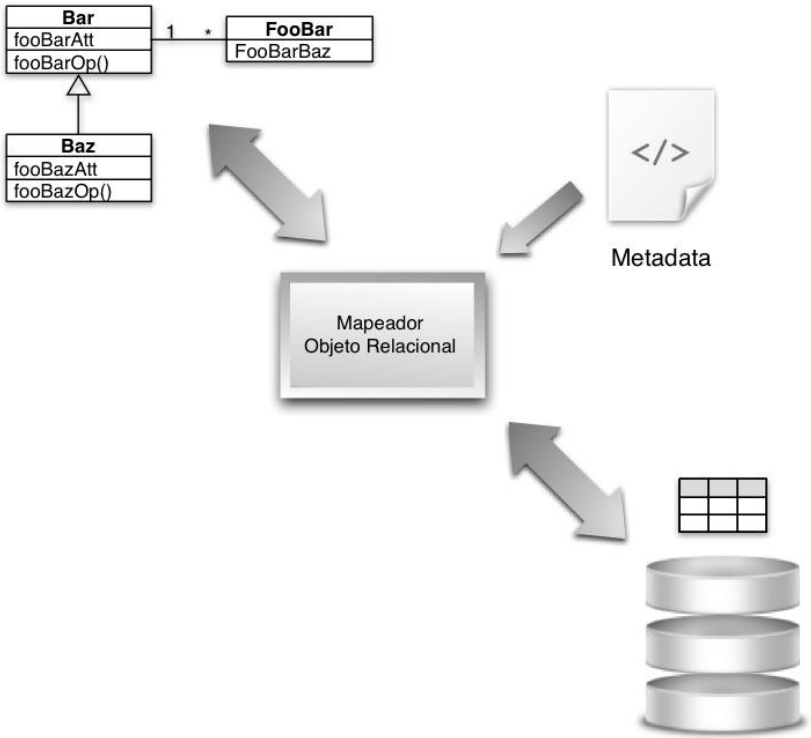


Diferencias entre modelos

OO Model (Java)	Relational Model
Object, classes	Table, rows
Attributes, properties	Columns
Identity	Primary key
Relationship/reference to other entity	Foreign key
Inheritance/polymorphism	Not supported
Methods	Indirect parallel to SQL logic, stored procedures, triggers.
Code is portable	Not necessarily portable, depending on vendor

Mecanismos de Persistencia

- Mapeador Objeto - Relacional



Mecanismos de Persistencia

- Mapeador Objeto - Relacional
  - Ventaja
    - Reduce la cantidad de código necesario para lograr la persistencia. La aplicación resultante es fácil de mantener.
  - Desventaja
    - Requiere Mantener metadata sobre las entidades.
  - Herramienta Java
    - Hibernate
    - OpenJPA

JPA – Conceptos - EntityBean

- Se anotan con @Entity
- Tienen una propiedad anotada con @Id
- Constructor sin argumentos public/protected
- No puede ser final
- Puede extender de otra
- Puede ser abstracta
- Es un POJO (Plain Old Java Objects), objeto liviano que no implementan ninguna interfaz

JPA – Conceptos - EntityBean

```
@Entity
public class Employee {

@Id private int id;
private String name;
private long salary;

public Employee() {}
public Employee(int id) { this.id = id; }

public int getId() { return id; }
public void setId(int id) { this.id = id; }
```

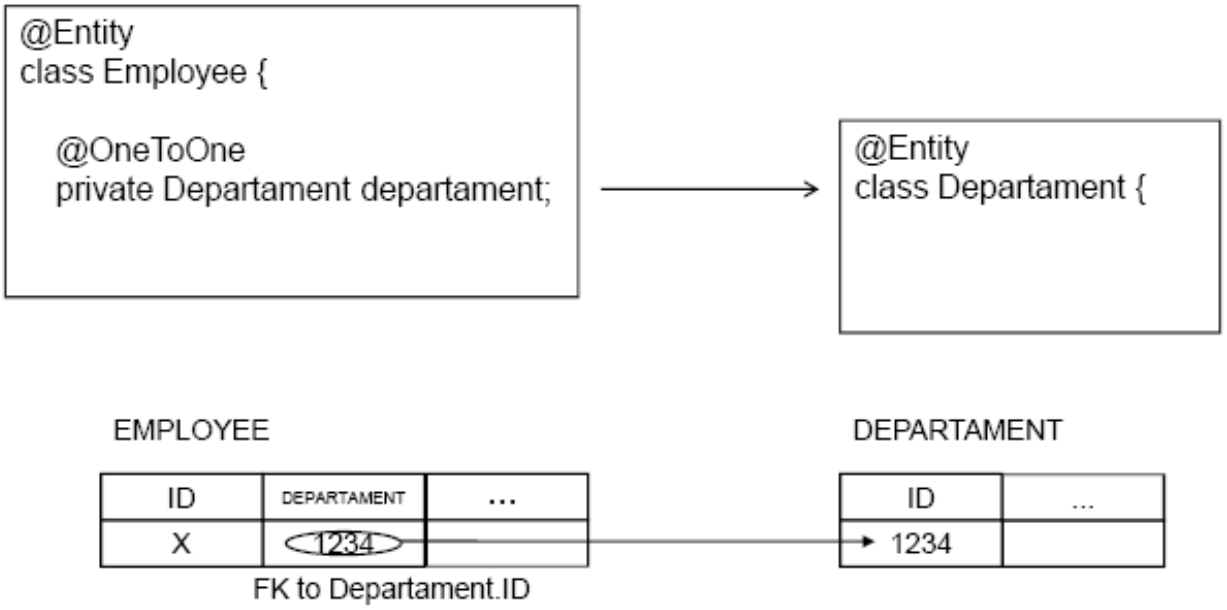
JPA – Conceptos - EntityBean

```
public String getName() { return name; }

public void setName(String name) {
  this.name = name;
}
public long getSalary() { return salary; }

public void setSalary (long salary) {
  this.salary = salary; }
}
```

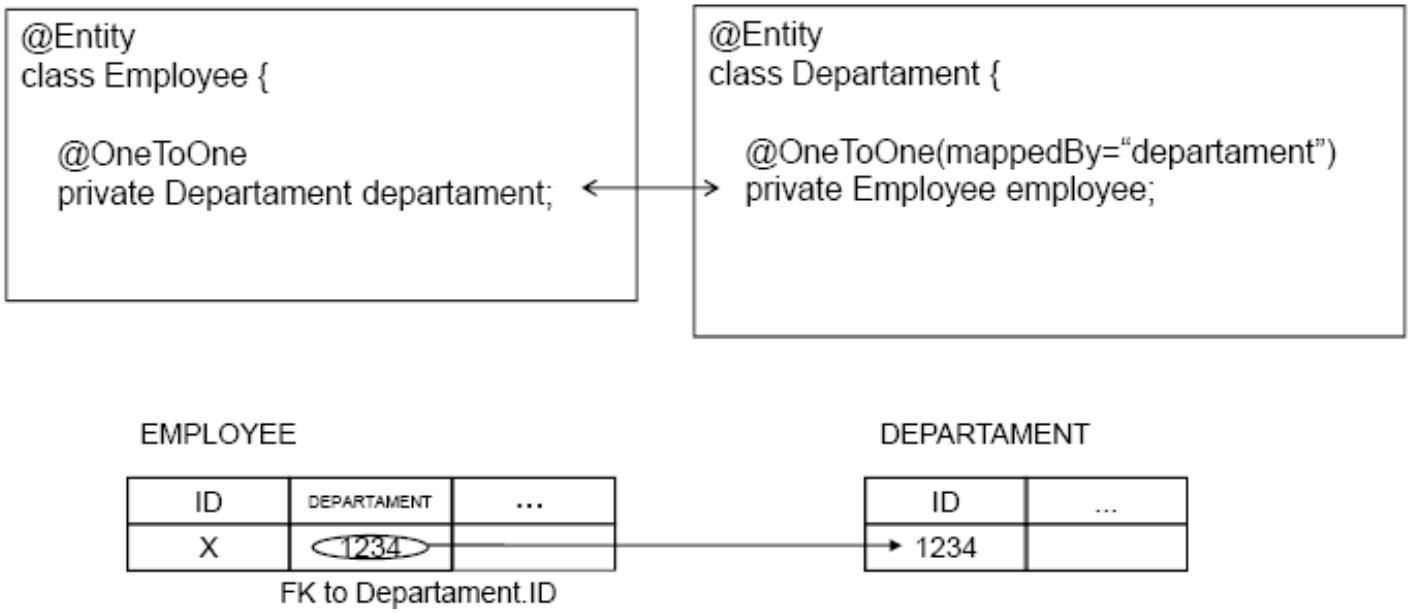
Relaciones entre entidades

- Relaciones Unidireccionales
  - Sólo se puede navegar desde una entidad a la otra
- Relaciones Bidireccionales
  - Desde cualquiera de las dos entidades es posible navegar a la otra
- Tipos
  - one-to-one
  - one-to-many
  - many-to-one
  - many-to-many

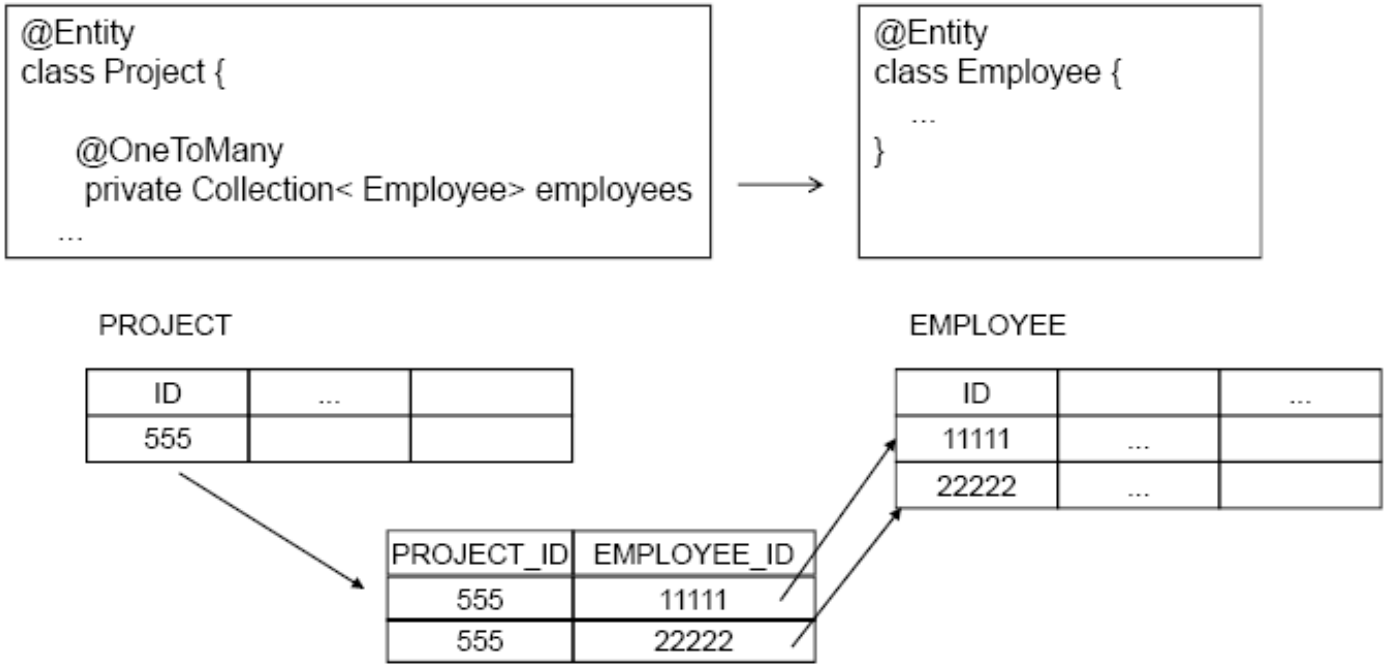
OneToOne - Unidireccional



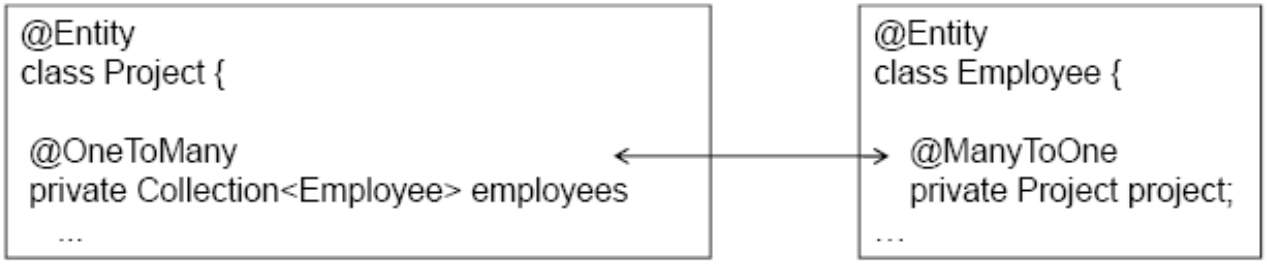
OneToOne - Bidireccional



OneToMany - Unidireccional



OneToMany - Bidireccional



ManyToMany

```
@Entity
class Project {

 @ManyToMany(mappedBy="projects")
 private Collection<Employee> employees
```

```
@Entity
class Employee {

 @ManyToMany
  private Collection< Project> projects
```

PROJECT

ID	...	
555		

EMPLOYEE

ID		...
11111	...	
22222	...	

PROJECT_ID	EMPLOYEE_ID
555	11111
555	22222
666	11111
666	22222

PROJECT_EMPLOYEE

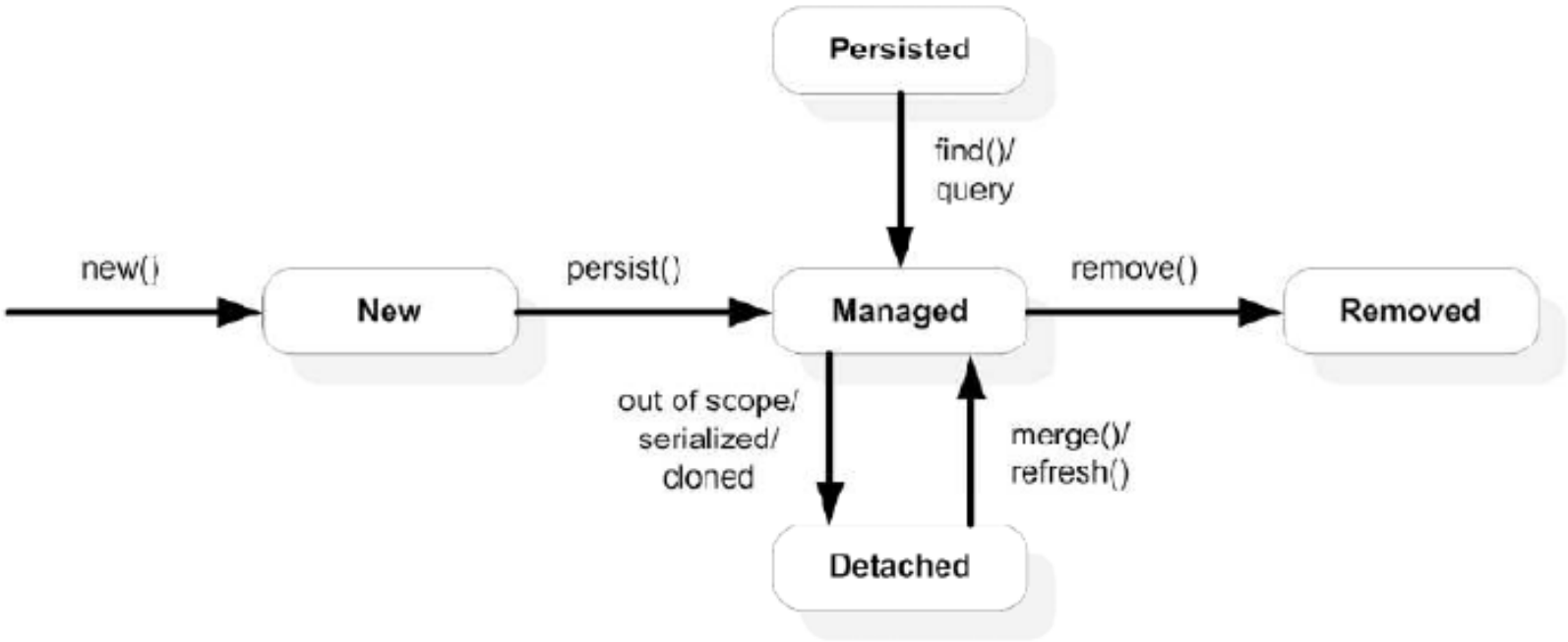
Entity Manager

- El API del EntityManager es la parte más importante e interesante de JPA
- Maneja el ciclo de vida de las entidades
- Actúa de puente entre el mundo O.O y el mundo relacional

Entity Manager - Metodos

- `public void persist(Object entity);`
- `public <T> T merge(T entity);`
- `public void remove(Object entity);`
- `public <T> T find(Class<T> entityClass,`
- `Object primaryKey);`
- `public void flush();`
- `public void setFlushMode(FlushModeType flushMode);`
- `public FlushModeType getFlushMode();`
- `public void refresh(Object entity);`
- `public Query createQuery(String jpqlString);`
- `public Query createNamedQuery(String name);`

Ciclo de vida de una entidad



JPQL

- El procesador de consultas de JPA, transforma una consulta JPQL en una consulta SQL


```
SELECT c
FROM Category c
WHERE c.categoryName LIKE :categoryName
ORDER BY c.categoryId
```

JPQL

- Una consulta puede tener los siguientes elementos:
  - SELECT
  - FROM
  - WHERE
  - ORDER BY
  - GROUP BY
  - HAVING

Consultas con la interface Query

```
Query query = em.createQuery("SELECT i
FROM Item i");
```

```
SELECT i FROM Item i
WHERE i.initialPrice = ?1
```

```
query.setParameter(1,100)
```

```
SELECT i FROM Item i
WHERE i.initialPrice = :price
```

```
query.setParameter("price",100)
```

```
List categories = query.getResultList();
```

Empaquetado

- Se debe definir un descriptor de la unidad de persistencia
  - META-INF/persistence.xml

```
<persistence>
   <persistence-unit name="tsi2_PU">
     <jta-data-source>java:myDS</jta-data-source>
     <properties>
             <property name="hibernate.show_sql" value="false" />
             <property name="hibernate.dialect"
                    value="org.hibernate.dialect.PostgtreSQLDialect" />
             <property name="hibernate.hbm2ddl.auto"
                     value="update" />
     </properties>
   </persistence-unit>
</persistence>
```

Ejemplo JPA

- HSQLDB
- Facturas – Lineas
- Persistir
- Consultas

Referencias

- OpenJPA
  - http://openjpa.apache.org/
- JPA Tutorial
  - http://java.sun.com/javaee/5/docs/tutorial/doc/bnbpz.html
- HSQLDB
  - http://hsqldb.org/
- Pro EJB 3 Java Persistence API
  - Mike Keith. Apress 2006
- Mastering Enterprise JavaBeans 3.0
  - Rima Patel Sriganesh.
- EJB 3 in Action. Wiley 2006.
  - Debu Panda. Manning 2007
- Java EE (Sitio oficial Sun)
  - http://java.sun.com/javaee/