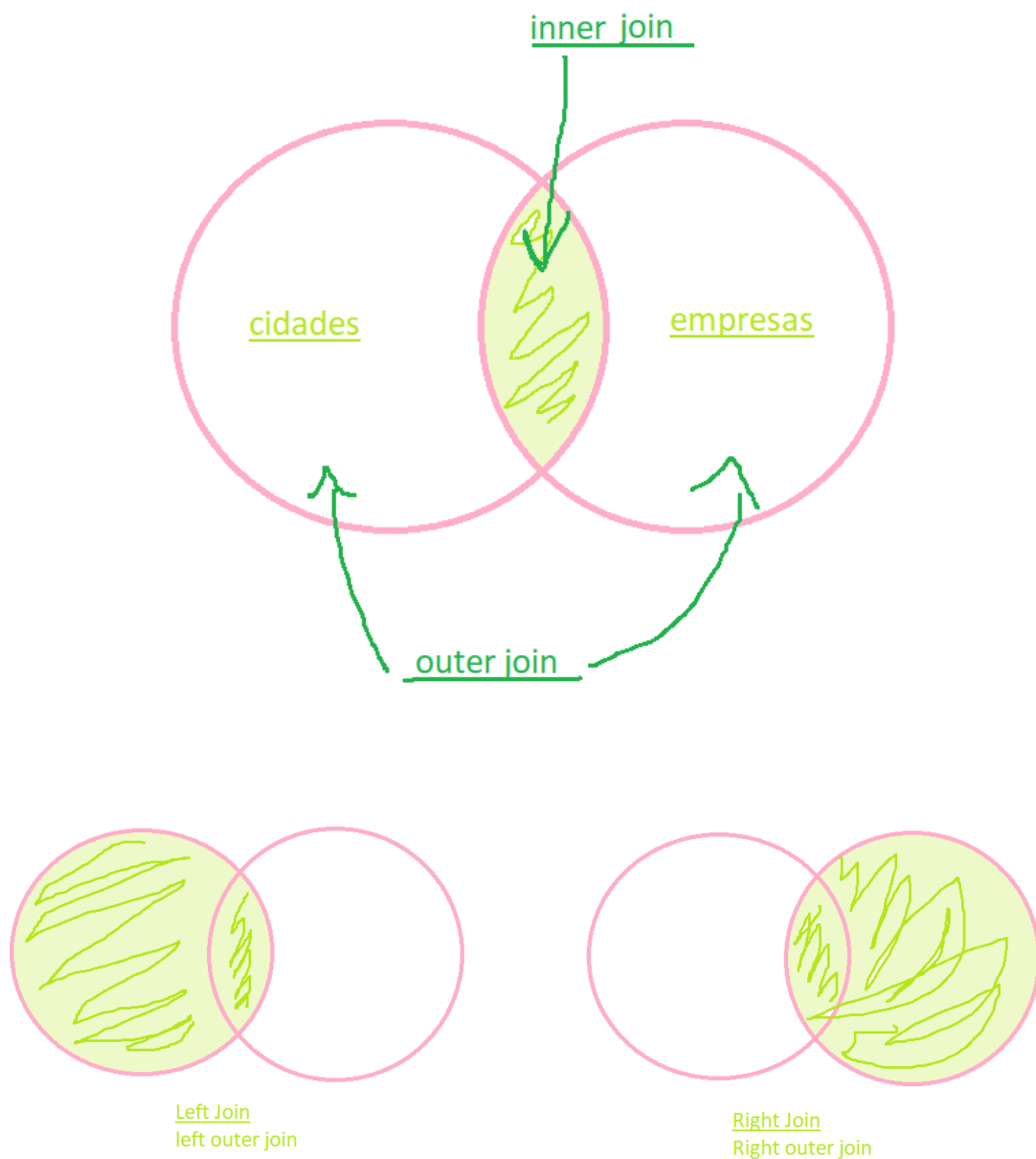
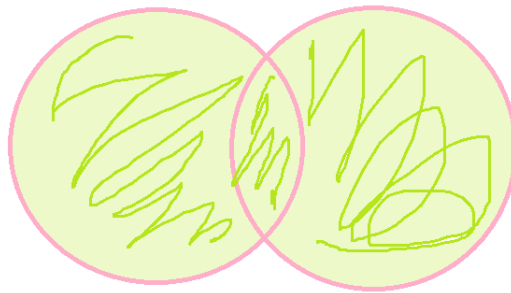


# Banco de Dados

## SQL





Full join

## JDBC - Java Database Connectivity

- É uma API com uma série de funcionalidades, com interfaces e classes, que te ajudarão a interagir com o banco de dados.

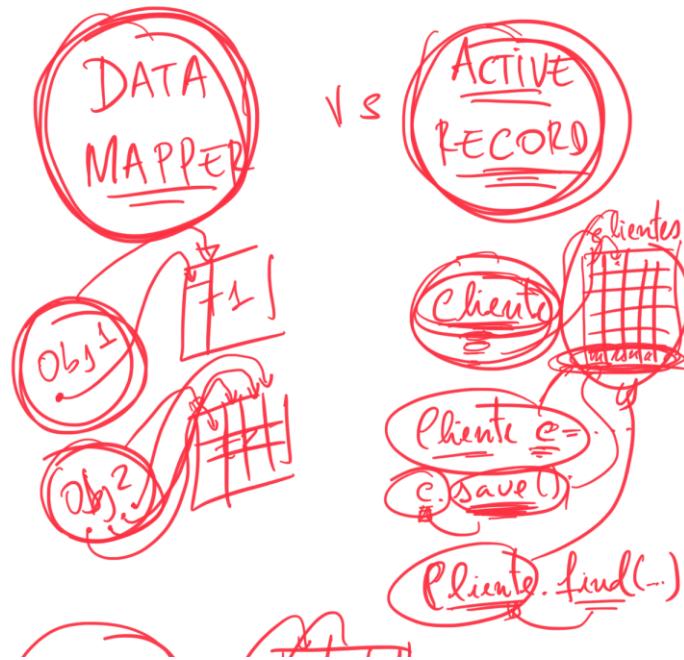
## DAO - Data Access Object

**porta** - é o acesso usado para abrir uma conexão/ uma comunicação por padrão, no sql, se vc nao seleciona nenhuma porta, ele seleciona a 3306

```
1 package padroes.observer;  
2  
3 @FunctionalInterface  
4 public interface ObservadorChegadaAniversariante  
5     public void chegou(EventoChegadaAniversariant  
6 }  
7
```

annotation

## Java Persistence API (JPA)



## Active Record

No Active Record o objeto possui suas operações básicas de CRUD nele mesmo. Dessa forma, fica mais fácil consultá-lo.

## Data Mapper

Já no Data Mapper sua classe não possuirá nenhum método relativo ao CRUD ou suas operações. Somente haverá possíveis *getters* e *setters* dos atributos (dependendo da linguagem) e alguns métodos como `clone()`, `toString()` e outros, se necessário. Dessa forma esse objeto fica mais "limpo".

Sendo assim, para trabalhar com um objeto no padrão Data Mapper, você deverá **passar o objeto para uma classe terceira que realizará as operações implementadas** (geralmente um DAO).

**ORM=** Object Relational Mapper



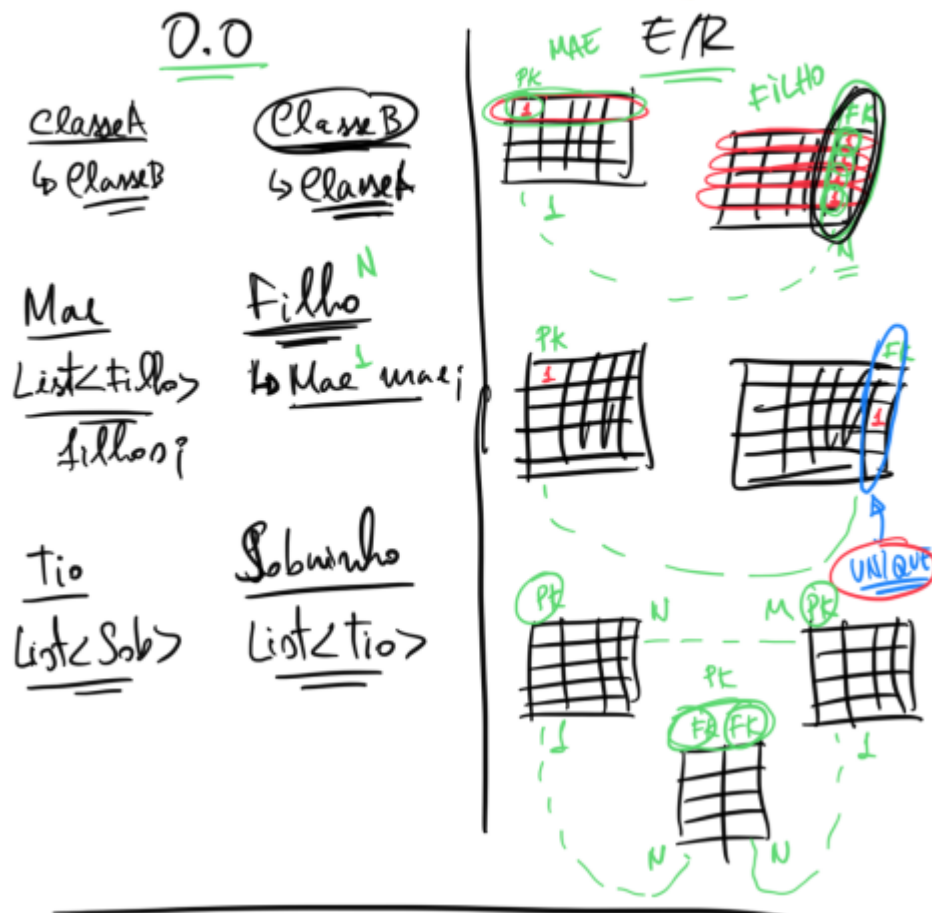


**O EntityManagerFactory** cria um entity manager específico. Ele deve ser usado para criar instâncias de um EntityManager.

**Um EntityManager** mapeia um conjunto de classes a um banco de dados particular. Este conjunto de classes é chamado de persistence unit (unidade de persistência).

**JPQL**= vc está acessando um objeto, não uma tabela.

## RELA CIONAMENTOS



**Na E/R:** A relação n:n é virtual, com uma outra tabela com as foreign keys intermediando a relação.

**No O.O:** Todas as relações são físicas, e temos relações bidirecionais também.

**JPA:** @OneToOne, @OneToMany | @ManyToOne, @ManyToMany.

**EAGER (geralmente 1pra1 ou MuitospraUm (...toOne)):** o relacionamento vai ser eager (ansioso) → quando eu obtenho um, ele já traz os outros.

**LAZY (geralmente MuitospraMuitos ou uma Lista (...toMany)):** o relacionamento vai ser lazy (obtem os dados de forma tardia) → só vai retornar esses dados se o usuário pedir, ele não traz junto com ele no primeiro momento.

# Banco de Dados Não Relacional (NoSQL)

**NoSQL:** Not only SQL

MongoDB:

- Documento → JSON
- Pode ter vários bancos de dados em uma única instância do MongoDB
- DB → Collections → Docs