

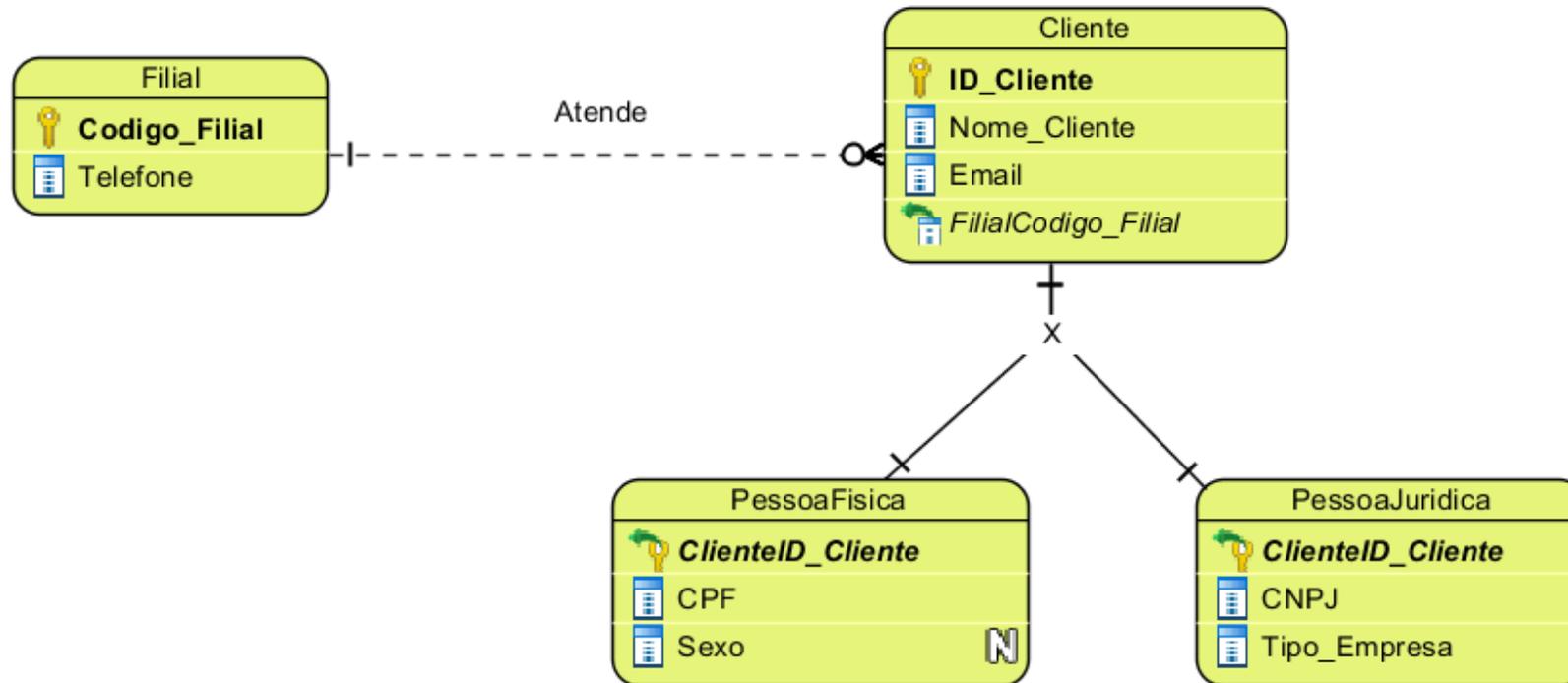
# Generalização e Especialização

ORM Aplicado

# Gen/Espec

- Especialização:
  - Definição de uma entidade que é um subconjunto de uma outra entidade
  - Processo de classificar o conjunto de entidades em conjunto de entidades especializados
- Generalização:
  - Definição de uma entidade que é um superconjunto de uma outra entidade.
  - Processo de generalizar vários conjuntos de entidades em um só conjunto de entidade

# Gen/Espec



# Herança

Os atributos dos conjuntos de entidades de nível superior (genérico) são herdados pelos conjuntos de entidades de nível inferior (específicos)

Em geral, atributos usados como critério não são herdados pelos Conjuntos de Entidades específicos

Numa hierarquia de especialização, os conjuntos de entidades específicos participam de todos os relacionamentos definidos para o Conjunto de Entidade genérico

A Herança de propriedades (atributos e relacionamentos) se propaga em todos os níveis de uma hierarquia de especialização

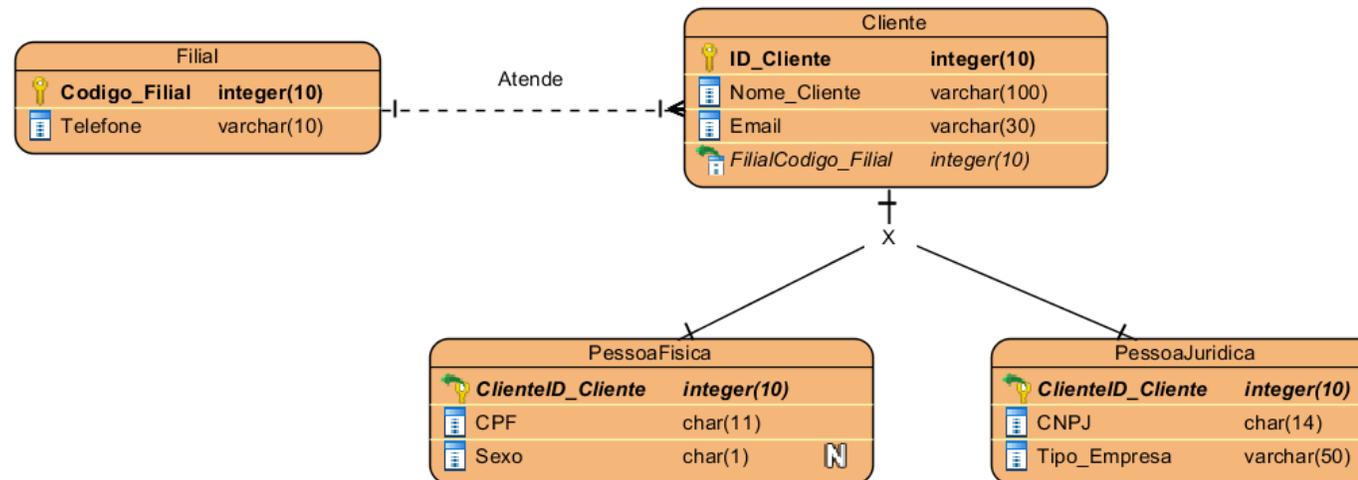
# Herança

Diferentemente da Orientação a Objetos, não se faz uma composição entre entidades, mas uma relação PK/FK entre a entidade mãe e a(s) entidade(s) filha(s).

Para recuperação completa do registro, deve-se fazer uma operação de consulta com cláusula de junção (JOIN) entre super entidade e sub entidade.

# DDL Aplicado ao Hibernate

**Exemplo:** Uma empresa tem diversas filiais que atendem a diversos clientes, no entanto, um cliente é atendido apenas por uma filial (Caracterizada por um código e seu telefone de atendimento). Os clientes atendidos devem ser armazenados por suas características relevantes (ID (único), nome e e-mail). Os clientes podem ser pessoas físicas ou jurídicas. Das pessoas físicas é obrigatório saber o CPF e pode-se informar o sexo. Das pessoas jurídicas, é obrigatório saber o CNPJ e o tipo de estabelecimento comercial



# DDL Aplicado ao Hibernate

- Considerando o HibernateUtil feito, mapeando as entidades e uma database de nome ORMHerancaUm
- Omitindo imports, Getters/Setters e toString()

```
@Entity
@Table(name = "filial")
public class Filial {

    @Id
    @Column(name = "codigo_filial")
    @NotNull
    private int codigo;

    @Column(name = "telefone", length = 10)
    @NotNull
    private String telefone;
}
```

# DDL Aplicado ao Hibernate

- A herança pode ser tratada com 3 estratégias, conforme definição ORM:
  - TABLE\_PER\_CLASS – Todas as propriedades de uma classe estão em sua tabela, portanto, nenhuma junção é necessária. (Estratégia de generalização)
  - SINGLE\_TABLE – As entidades de diferentes classes com um ancestral comum são colocadas em uma única tabela. (Estratégia de generalização)
  - JOINED – Cada classe tem sua tabela, e consultar uma entidade de subclasse requer a junção das tabelas. (Estratégia de especialização)

# DDL Aplicado ao Hibernate

- HibernateUtil deve mapear TODAS as classes e subclasses
- Para aplicar a especialização, que diminui a incidência de *nulls*, a estratégia utilizada será a JOINED

```
@Entity
@Table(name = "cliente")
@Inheritance(strategy = InheritanceType.JOINED)
public class Cliente {

    @Id
    @Column(name = "id_cliente")
    @NotNull
    private int id;

    @Column(name = "nome", length = 100)
    @NotNull
    private String nome;

    @Column(name = "email", length = 30)
    @NotNull
    private String email;

    @ManyToOne
    @JoinColumn(name = "codigo_filial")
    @NotNull
    private Filial filial;
```

```
@Entity
@Table(name = "pessoa_fisica")
public class PessoaFisica extends Cliente {

    @Column(name = "cpf", length = 11)
    @NotNull
    private String cpf;

    @Column(name = "sexo", length = 1)
    @Null
    private String sexo;
```

```
@Entity
@Table(name = "pessoa_juridica")
public class PessoaJuridica extends Cliente {

    @Column(name = "cnpj", length = 14)
    @NotNull
    private String cnpj;

    @Column(name = "tipo_empresa", length = 50)
    @NotNull
    private String tipo;
```

# DDL Aplicado ao Hibernate

- Geração DDL

```
Problems Javadoc Declaration Console × Git Staging
<terminated> Principal (5) [Java Application] C:\Program Files\Java\jdk-11.0.9\bin\javaw.exe (12 de mai de 2022 11:11:43 - 11:11:48)
INFO: HHH10001501: Connection obtained from JdbcConnectionAccess [org.hibernate.engine.jdbc.env.internal.JdbcEnvironmentInitiator$ConnectionProviderJdbcConnectionAccess@4c731956] for (non-JTA)
Hibernate: create table cliente (id_cliente integer not null, email varchar(30) not null, nome varchar(100) not null, codigo_filial integer not null, primary key (id_cliente)) engine=InnoDB
Hibernate: create table filial (codigo_filial integer not null, telefone varchar(10) not null, primary key (codigo_filial)) engine=InnoDB
Hibernate: create table pessoa_fisica (cpf varchar(11) not null, sexo varchar(1) not null, id_cliente integer not null, primary key (id_cliente)) engine=InnoDB
Hibernate: create table pessoa_juridica (cnpj varchar(14) not null, tipo_empresa varchar(50) not null, id_cliente integer not null, primary key (id_cliente)) engine=InnoDB
Hibernate: alter table cliente add constraint FK3itw7eryw54cfi5nf8ght24ud foreign key (codigo_filial) references filial (codigo_filial)
Hibernate: alter table pessoa_fisica add constraint FK6hssyf14h9l0e9p8pfjofk10p foreign key (id_cliente) references cliente (id_cliente)
Hibernate: alter table pessoa_juridica add constraint FKp81xan09bwu6nly3afyta5sn0 foreign key (id_cliente) references cliente (id_cliente)
```

# Exercício

- Fazer o Modelagem ER, aplicando modelo físico e implementar em projeto Java com Hibernate para a geração DDL

A empresa XYZ quer informatizar seus dados e tem, preliminarmente, que cadastrar seus funcionários. Todos recebem uma identificação interna, e devem cadastrar seu nome, data de nascimento, seu salário, apenas um telefone de contato (celular). Quando o funcionário for um atendente, é preciso saber seu horário de entrada, seu horário de saída e seu e-mail institucional. Quando o funcionário for entregador, é importante saber o número e a categoria da CNH. O atendente não tem clientes fixos, podendo atender vários clientes sendo que, um cliente pode ser atendido por diversos atendentes. É importante saber a data e a hora do atendimento. É importante frisar que, dos clientes, é armazenado CPF, nome, telefone de contato (celular), um e-mail e um pronome de tratamento.