ESTRITAMENTE **CONFIDENCIAL**

ELABORATA
INFORMÁTICA

LINGUAGEM JAVA

Hibernate Relacionamento um-para-muitos

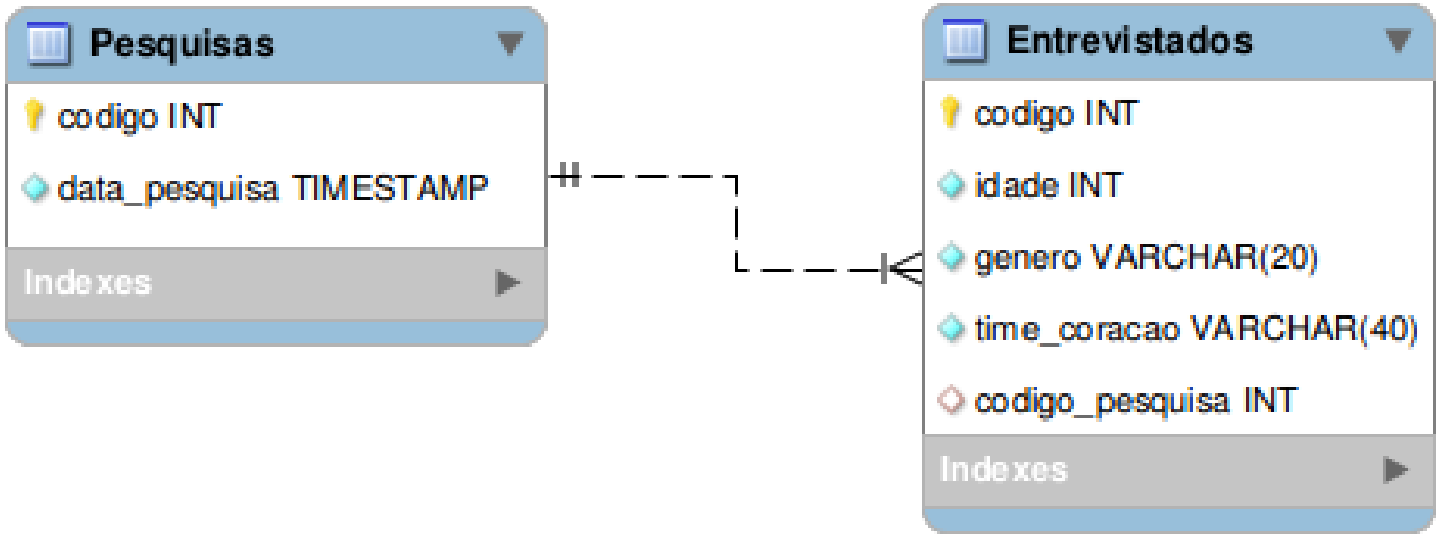
Relacionamento um-para-muitos

- O relacionamento um-para-muitos acontece quando duas tabelas em um banco de dados relacional estão relacionadas e na primeira tabela existe um registro que possuem um ou mais registros relacionados a ele na segunda tabela.

Exemplos

- Funcionário e dependentes
- Nota fiscal e itens
- Curso e disciplinas

Modelo MRN um-para-muitos



Script SQL um-para-muitos

```
CREATE TABLE Pesquisas
(
  codigo INT NOT NULL AUTO_INCREMENT,
  data_pesquisa TIMESTAMP NOT NULL,
  PRIMARY KEY (codigo)
);

CREATE TABLE Entrevistados
(
  codigo INT NOT NULL AUTO_INCREMENT,
  idade INT NOT NULL,
  genero VARCHAR(20) NOT NULL,
  time_coracao VARCHAR(40) NOT NULL,
  codigo_pesquisa INT NULL,
  PRIMARY KEY (codigo),
  FOREIGN KEY (codigo_pesquisa)
       REFERENCES Pesquisas (codigo) ON UPDATE NO ACTION
);
```

Um-para-muitos no Hibernate

ELABORATA INFORMÁTICA

- Além das annotations @Entity, @Table, @Id, @GeneratedValue e @Column utilizadas no mapeamento de uma única tabela, para mapear um relacionamento um-para-muitos no Hibernate é necessário utilizar as annotations @OneToMany, @ManyToOne e @JoinColumn. Na classe que representa a tabela principal deverá ser um arquivo que implemente a interface Set (Collections) para armazenar todos os objetos filhos. Na classe filha deverá ter um atributo que represente a classe principal.

Exemplo

```
@Entity
@Table(name = "pesquisas")
public class PesquisaModel {
  @Id
  @GeneratedValue(strategy = GenerationType.IDENTITY)
  @Column(name = "codigo", unique = true)
  private Integer codigo;

  @Column(name = "data")
  @Temporal(TemporalType.TIMESTAMP)
  private Date data;

  @OneToMany(fetch = FetchType.LAZY, mappedBy = "pesquisa",
      cascade = CascadeType.ALL)
  private Set<EntrevistadoModel> entrevistados =
      new HashSet<>(0);

  // Getters e Setters
}
```

Exemplo

```
@Entity
@Table(name = "entrevistados")
public class EntrevistadoModel {
  @Id
  @GeneratedValue(strategy = GenerationType.IDENTITY)
  @Column(name = "codigo")
  private int codigo;

  @Column(name = "idade")
  private int idade;

  @Column(name = "genero")
  private String genero;

  @Column(name = "time")
  private String time;

  @ManyToOne(fetch = FetchType.LAZY)
  @JoinColumn(name = "codigo_pesquisa", nullable = false)
  private PesquisaModel pesquisa;
  ...
```

Persistindo um-para-muitos

- Após as classes serem devidamente anotadas para estabelecer o relacionamento um-para-muitos, é preciso preencher os objetos corretamente para que o dados sejam devidamente inseridos nas tabelas.

Exemplo

```
...
PesquisaModel p = new PesquisaModel();

EntrevistadoModel e1 = new EntrevistadoModel();
e1.setIdade(18);
e1.setGenero("Masculino");
e1.setTime("Coritiba");
e1.setPesquisa(p);

EntrevistadoModel e2 = new EntrevistadoModel();
e2.setIdade(21);
e2.setGenero("Feminino");
e2.setTime("Coritiba");
e2.setPesquisa(p);

Set<EntrevistadoModel> entrevistados = new HashSet<>();
entrevistados.add(e1);
entrevistados.add(e2);
...
```

Exemplo

```
...
p.setData(new Date());
p.setEntrevistados(entrevistados);

Session session =
HibernateUtils.getSessionFactory().openSession();
Transaction t = null;
try {
  t = session.beginTransaction();
  session.save(p);
  session.flush();
  t.commit();
} catch (Exception e) {
  e.printStackTrace();
  if (t != null)
  t.rollback();
} finally {
  session.close();
}
...
```

OBRIGADO!

ESTRITAMENTE **CONFIDENCIAL**

ELABORATA
INFORMÁTICA

www.**elaborata**.com.br

Horário de Atendimento Comercial
Segunda à sexta – das 9:00h às 19:30h e
Sábado - das 8:00h às 15:00h.

Rua Monsenhor Celso, 256 - 1º Andar
Centro - Curitiba - PR

41.3324.0015
41.99828.2468

cursos@elaborata.com.br