C207 - BD

Arthur Openheimer

Inatel

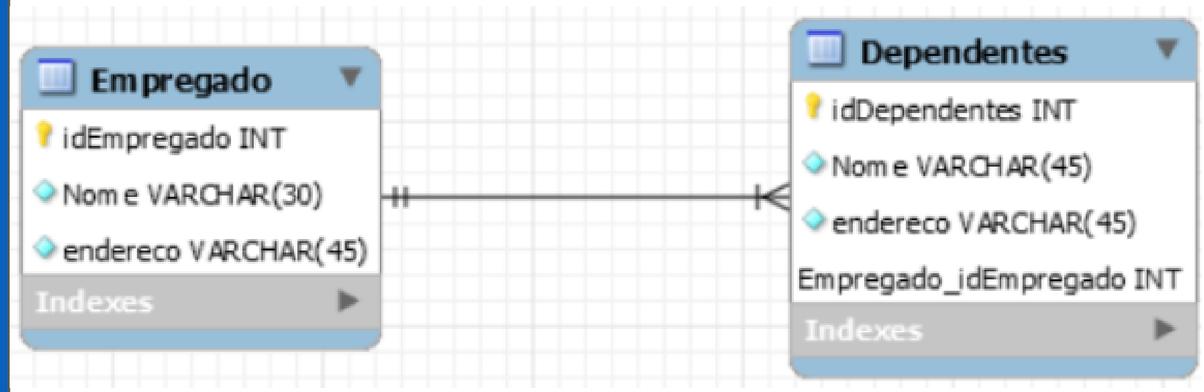
Criando tabelas com chaves estrangeiras

CONSTRAINT fk_tabelas FOREIGN KEY (atributo1) REFERENCES Nome_Tabela2 (atributo2) ON UPDATE CASCADE ON DELETE CASCADE;

- **atributo1** é da tabela que RECEBE a chave primária da outra tabela através do **atributo2**
- **On Delete/On Update**
  - São regras opcionais que fazem com que a tabela original do atributo2 receba alterações quando uma determinada ação é executada:
  - "**No Action**" ou "**Restrict**" - Não alteram ou deletam nada quando uma restrição é quebrada
  - "**Cascade**" - Altera ou deleta todos os atributos que estão envolvidos no relacionamento
  - "**Set Null**" - Seta como null os atributos envolvidos na restrição

Inatel

Relacionamentos 1:1 e 1:N

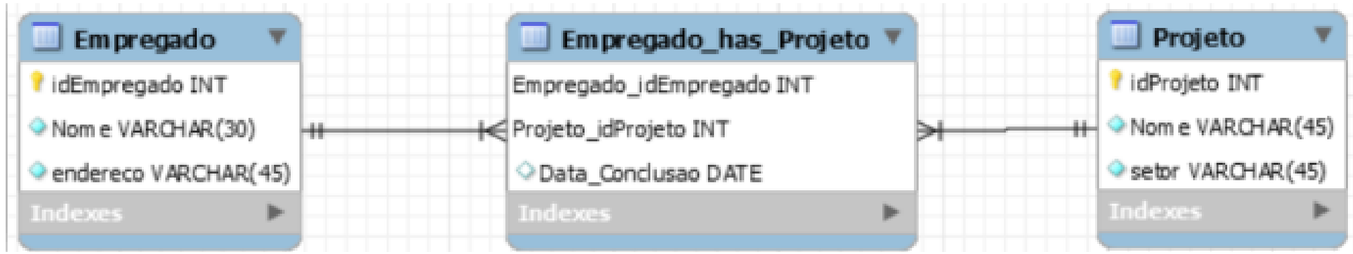



```
CREATE TABLE IF NOT EXISTS Empregado (
  idEmpregado INT NOT NULL AUTO_INCREMENT,
  Nome VARCHAR(30) NOT NULL,
  endereco VARCHAR(45) NOT NULL,
  PRIMARY KEY (idEmpregado));
```

```
CREATE TABLE IF NOT EXISTS Dependentes (
  idDependentes INT NOT NULL AUTO_INCREMENT,
  Nome VARCHAR(45) NOT NULL,
  endereco VARCHAR(45) NOT NULL,
  Empregado_idEmpregado INT NOT NULL,
  PRIMARY KEY (idDependentes),
  CONSTRAINT fk_Dependentes_Empregado
    FOREIGN KEY (Empregado_idEmpregado)
    REFERENCES Empregado (idEmpregado)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION);
```

Inatel

Relacionamento N:N




```sql
CREATE TABLE IF NOT EXISTS Empregado (
  idEmpregado INT NOT NULL AUTO_INCREMENT,
  Nome VARCHAR(30) NOT NULL,
  endereco VARCHAR(45) NOT NULL,
  PRIMARY KEY (idEmpregado));
```

```sql
CREATE TABLE IF NOT EXISTS Projeto (
  idProjeto INT NOT NULL AUTO_INCREMENT,
  Nome VARCHAR(45) NOT NULL,
  setor VARCHAR(45) NOT NULL,
  PRIMARY KEY (idProjeto));
```

```sql
CREATE TABLE IF NOT EXISTS Empregado_has_Projeto (
  Empregado_idEmpregado INT NOT NULL,
  Projeto_idProjeto INT NOT NULL,
  Data_Conclusao DATE,
  PRIMARY KEY (Empregado_idEmpregado, Projeto_idProjeto),
  CONSTRAINT fk_Empregado_has_Projeto_Empregado
    FOREIGN KEY (Empregado_idEmpregado)
    REFERENCES Empregado (idEmpregado),
  CONSTRAINT fk_Empregado_has_Projeto_Projeto1
    FOREIGN KEY (Projeto_idProjeto)
    REFERENCES Projeto (idProjeto)
);
```

Inatel

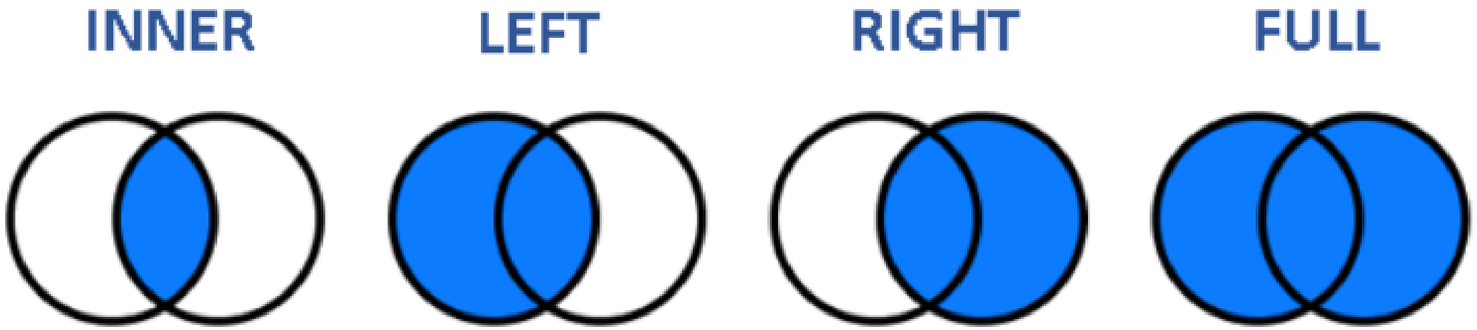
SELECT

SELECT <lista de atributos> FROM <lista de tabelas> WHERE <condição>;

- Parâmetros opcionais do select:
  - **GROUP BY** - Para agrupar registros selecionados em grupos
  - **HAVING** - Para expressar a condição que deve satisfazer cada grupo
  - **ORDER BY** - Para ordenar os registros selecionados numa ordem específica
- Parâmetros opcionais do Where:
  - **AND**, **NOT** e **OR**, juntamente com os operadores <,>,=,<>,<= e >=
  - (**NOT**) **BETWEEN** - Para especificar intervalos de valores
  - **LIKE** - Para comparar textos
  - **IN** - Para buscar dados de valores específicos dentro do WHERE
- Funções de agregação:
  - AVG - Para calcular a média dos valores de um campo determinado
  - COUNT - Para contar o número de registros da seleção
  - SUM - Para somar todos os valores de um atributo
  - MAX - Para devolver o valor mais alto de um atributo especificado
  - MIN - Para devolver o valor mais baixo de um atributo especificado

SELECT com JOIN

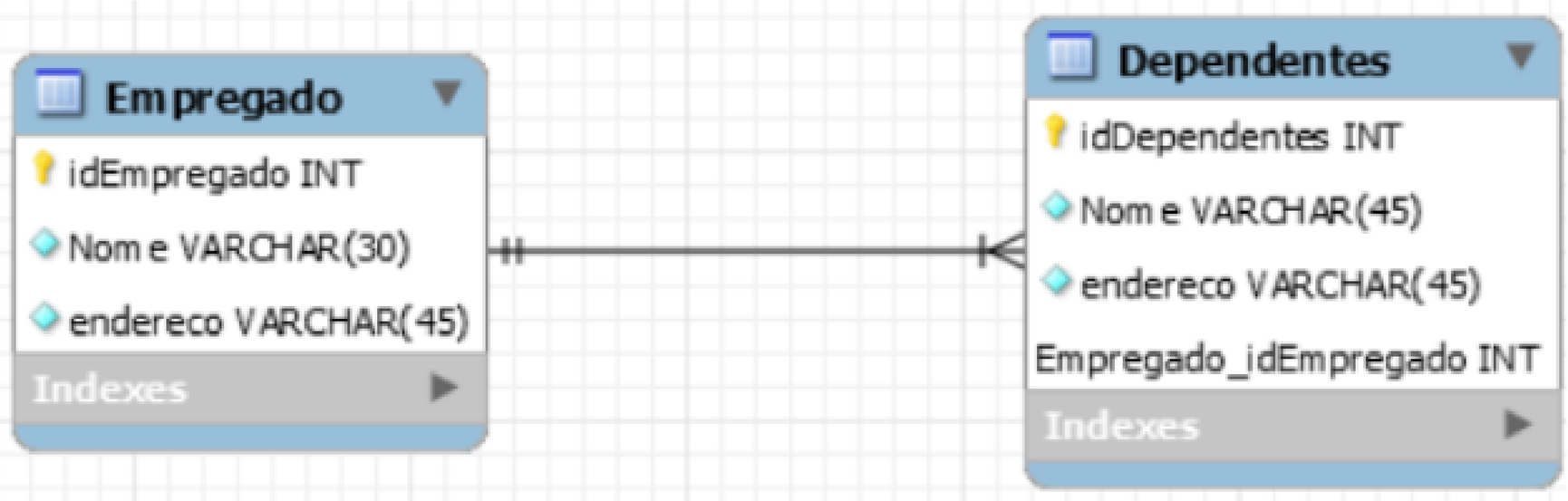
O JOIN é usado para fazer junções no SQL, ele é usado dentro do comando SELECT para juntar colunas de diferentes tabelas em um relacionamento, trazendo informações mais completas sobre algo



Inatel

SELECT com JOIN

Para relacionamentos 1:1 e 1:N



Inatel

SELECT com JOIN

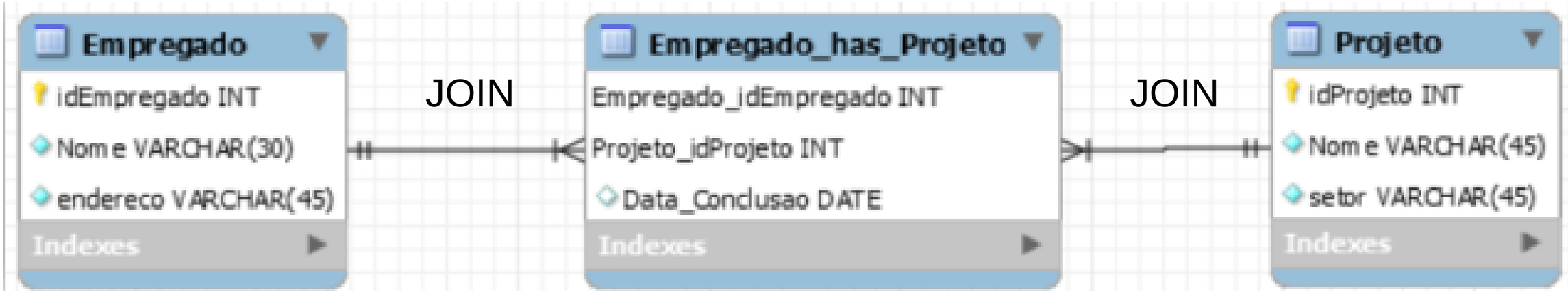
Para relacionamentos 1:1 e 1:N

```
SELECT Empregado.nome, Dependentes.nome FROM Empregado JOIN Dependentes;

SELECT E.nome, D.nome FROM Empregado AS E INNER JOIN Dependentes AS D;

SELECT E.nome, D.nome FROM Empregado AS E JOIN Dependentes AS D ON
D.empregado_idEmpregado = E.idEmpregado;

SELECT E.nome, D.nome FROM Empregado AS E JOIN Dependentes AS D ON
D.empregado_idEmpregado = E.idEmpregado WHERE E.idEmpregado = 1 ORDER BY E.nome;
```

Inatel

SELECT com JOIN

Para relacionamentos N:N



Inatel

SELECT com JOIN

Para relacionamentos N:N

```
SELECT E.nome, P.nome, EP.data_conclusao FROM Empregado AS E JOIN
Empregado_has_Projeto AS EP ON E.idEmpregado = EP.empregado_id JOIN Projeto AS P
ON P.idProjeto = EP.Projeto_id ORDER BY EP.data_conclusao;
```

Inatel