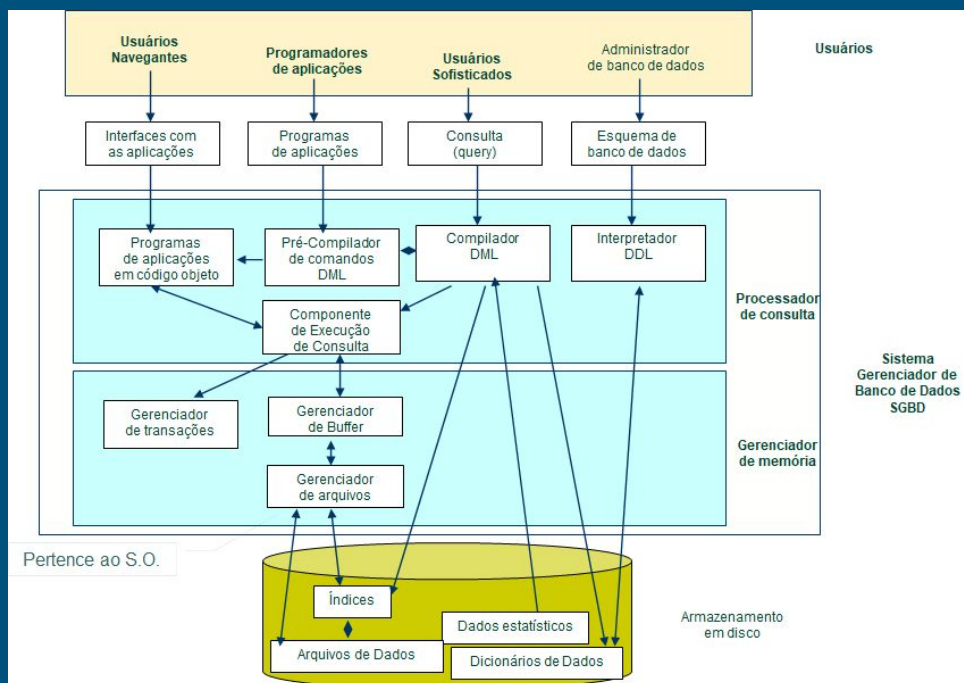


Aula 06

Banco de Dados

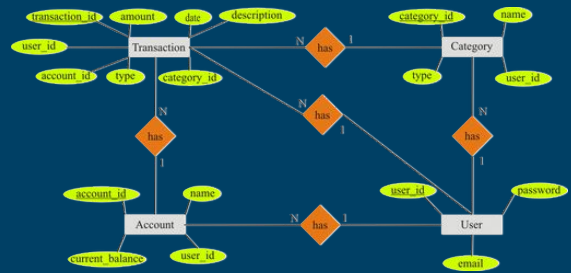
emerson@paduan.pro.br



emerson@paduan.pro.br

Modelo Entidade Relacionamento

Diagrama
Entidade-Relacionamento (DER)



Conceitos fundamentais

- Entidade
- Atributo
- Relacionamento
- Cardinalidade
- Generalização/Especialização

Entidade

➤ Entidade

- representa objetos concretos (uma pessoa, um automóvel) ou abstratos (um departamento, um endereço)
- o nome é usado no singular
- no DER representada por um retângulo contendo o nome da entidade

Cliente

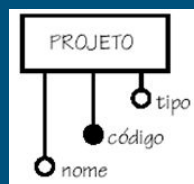
Departamento

emerson@paduan.pro.br

Atributos

➤ Atributos

- são as propriedades sobre as quais temos interesse na Entidade
- ligadas à entidade à qual pertencem



emerson@paduan.pro.br

Como identificar

- Como identificar as entidades?
 - A partir de uma descrição, identificar os objetos “participantes” da descrição.
- E os atributos?
 - Características, informações, que descrevem esses objetos

emerson@paduan.pro.br

Relacionamento

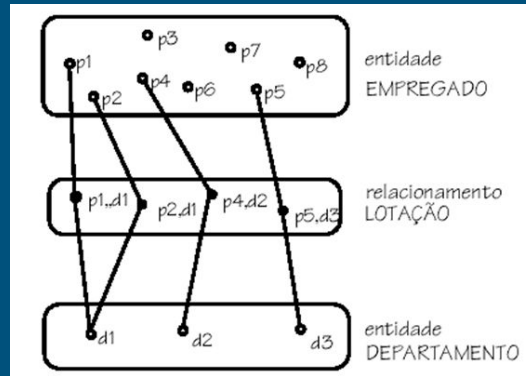
- Relacionamento
 - Conjunto de associações entre entidades
 - Observe os verbos



emerson@paduan.pro.br

Relacionamento

- Relacionamento
 - Diagrama de ocorrências



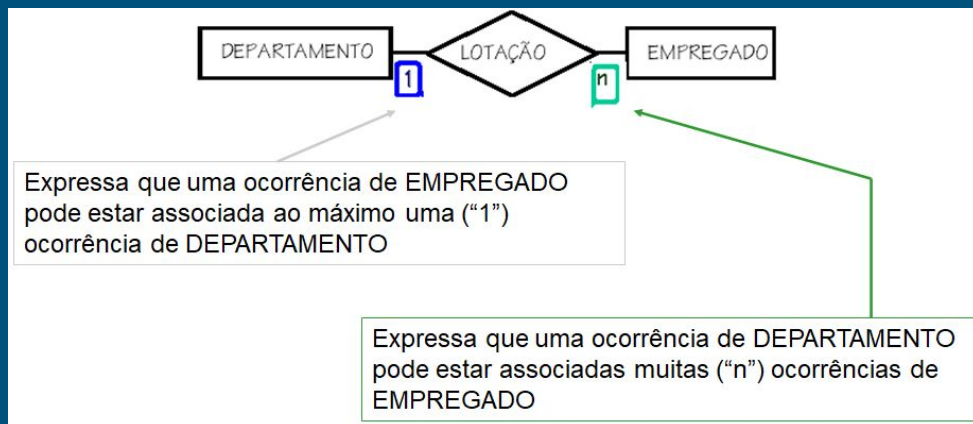
emerson@paduan.pro.br

Cardinalidade

- Cardinalidade
 - Representa quantas ocorrências de uma entidade podem estar associadas a uma determinada ocorrência de entidade através do relacionamento
 - Pode ser usada para classificar relacionamentos binários (2 entidades)
 - $n : n$ (muitos para muitos)
 - $1 : n$ (um para muitos)
 - $1 : 1$ (um para um)

emerson@paduan.pro.br

Leitura



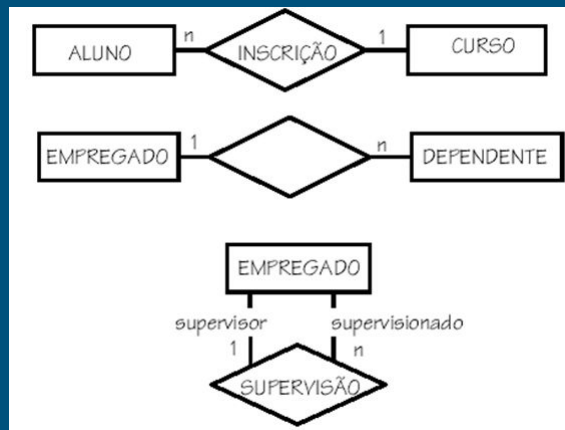
emerson@paduan.pro.br

Relacionamentos 1 : 1



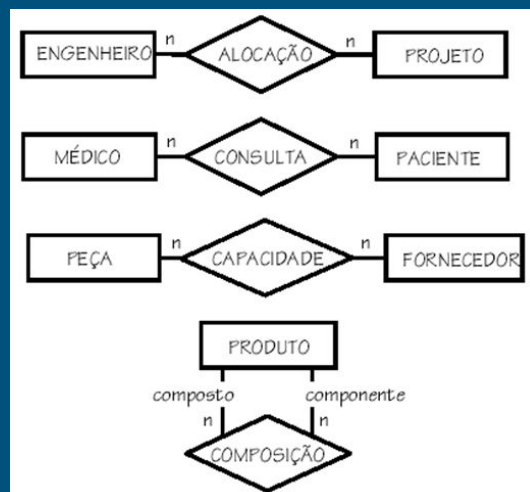
emerson@paduan.pro.br

Relacionamentos 1 : N



emerson@paduan.pro.br

Relacionamentos N : N



emerson@paduan.pro.br

Terminologias

Emp

CódigoEmp	Nome	CódigoDepto	CategFuncional
E5	Souza	D1	C5
E3	Santos	D2	C5
E2	Silva	D1	C2
E1	Soares	D1	—

Coluna (atributo) →

nome do campo (nome do atributo) →

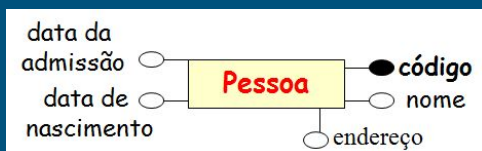
valor do campo (valor do atributo) →

linha (tupla) →

emerson@paduan.pro.br

Entidades

- Toda entidade gera uma tabela
- Cada atributo de uma entidade corresponde a uma coluna da tabela
- Atributos identificadores correspondem às colunas que compõem a chave-primária da tabela



Pessoa (CódigoPess, Nome, Endereço, DataAdm, DataNasc)

emerson@paduan.pro.br

Chaves

Um chave primária é uma coluna, ou uma combinação de colunas, cujos valores distinguem uma linha das demais dentro de uma tabela.

Uma chave estrangeira é uma coluna, ou uma combinação de colunas, cujos valores referenciam a chave primária de outra tabela. (relacionamentos)

DEPTO	
CodigoDepto	NomeDepto
D1	Compras
D2	Engenharia
D3	Vendas

CodigoDepto em EMP é uma chave estrangeira em relação a tabela DEPTO

EMP				
CodigoEmp	Nome	CodigoDepto	CategFuncional	CIC
E1	Souza	D1	-	132.121.331-20
E2	Santos	D2	C5	891.221.111-11
E3	Silva	D2	C5	341.511.775-45
E5	Soares	D1	C2	631.692.754-88

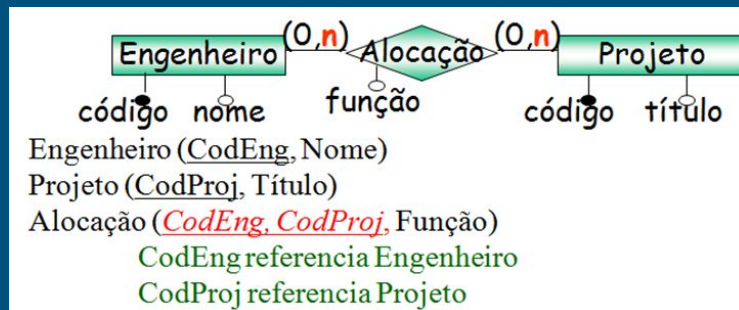
emerson@paduan.pro.br

Relacionamentos

- Tabela própria: O relacionamento é implementado como uma tabela própria contendo as colunas correspondentes aos identificadores das entidades relacionadas.
- Adição de colunas: Adicionar colunas correspondentes aos identificadores da entidade com cardinalidade máxima 1 à outra entidade envolvida no relacionamento.
- Fusão de tabelas: Somente é possível quando o relacionamento é do tipo 1:1.

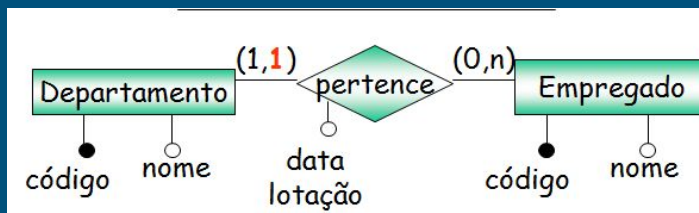
emerson@paduan.pro.br

Relacionamentos n: n



emerson@paduan.pro.br

Relacionamentos 1 : n



Departamento (CodDept, Nome)
Empregado (CodEmp, Nome, CodDept)
CodDept referencia Departamento

emerson@paduan.pro.br

SGBD MySQL



emerson@paduan.pro.br

Antes de iniciar...

Instalações

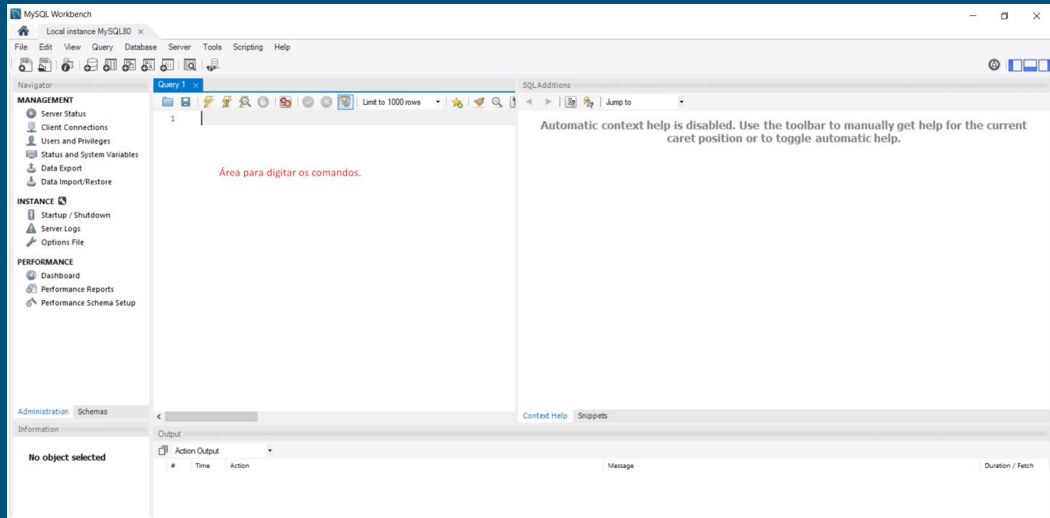
Download:

<https://dev.mysql.com/downloads/windows/installer/8.0.html>

Como instalar:

<https://www.mysqltutorial.org/install-mysql/>

MySQL Workbench



emerson@paduan.pro.br

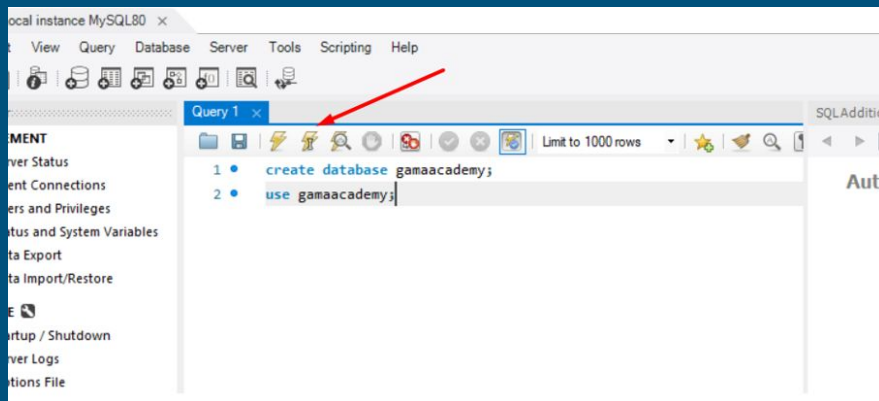
SQL Language

DDL - Data Definition Language (Criação/Manipulação das estruturas)

DML - Data Manipulation Language (Manipulação dos dados)

emerson@paduan.pro.br

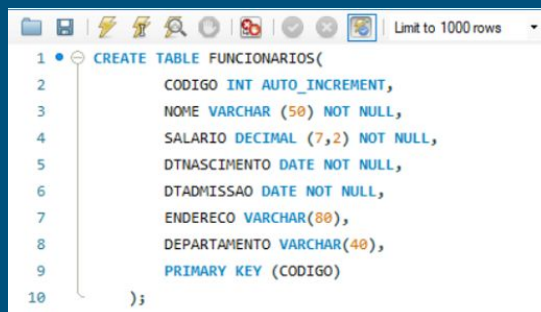
SQL Language



emerson@paduan.pro.br

SQL Language

CREATE TABLE <nome-tabela> (<nome-coluna> <tipo-do-dado> [NOT NULL]
PRIMARY KEY (nome-coluna-chave) FOREIGN KEY
(nome-coluna-chave-estrangeira) REFERENCES <nome-tabela-pai>
(nome-coluna-chave-primária));



emerson@paduan.pro.br

SQL Language

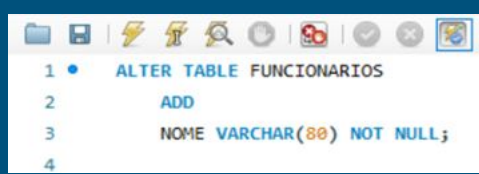
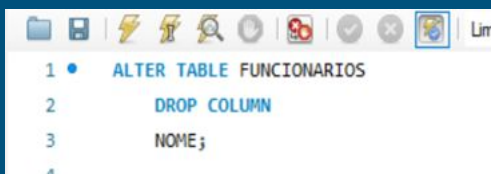
ALTER TABLE <nome-tabela>

DROP COLUMN <nome-coluna>

ADD <nome-coluna> <tipo-do-dado> [NOT NULL] [NOT NULL WITH DEFAULT]

RENAME <nome-coluna> <novo-nome-coluna>

MODIFY <nome-coluna> <tipo-do-dado> [NULL] [NOT NULL]



emerson@paduan.pro.br

SQL Language

DROP TABLE

Deletar a estrutura e os dados existentes em uma tabela. Após a execução deste comando estarão deletados todos os dados, estrutura e índices de acessos que estejam a ela associados.

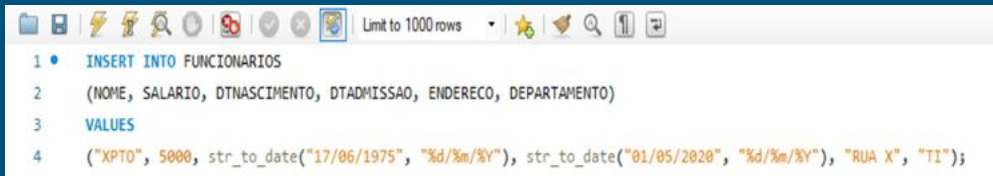
Sintaxe:

DROP TABLE [IF EXISTS]<nome-tabela>;

emerson@paduan.pro.br

SQL Language

INSERT INTO <nome-tabela> [(<nome-coluna>, [<nome-coluna>],...)] VALUES (<conteúdo>, [<conteúdo>], ...);



The screenshot shows a SQL query editor with a toolbar at the top. The query text is as follows:

```
1 • INSERT INTO FUNCIONARIOS
2 (NOME, SALARIO, DTNASCIMENTO, DTADMISSAO, ENDereco, DEPARTAMENTO)
3 VALUES
4 ("XPTO", 5000, str_to_date("17/06/1975", "%d/%m/%Y"), str_to_date("01/05/2020", "%d/%m/%Y"), "RUA X", "TI");
```

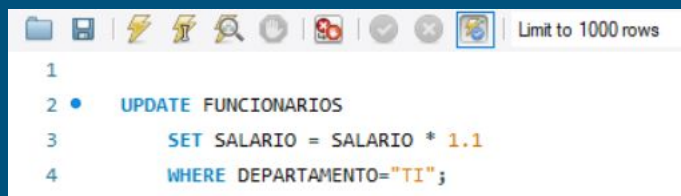
* Para zerar o auto incremento:

ALTER TABLE <TABELA> AUTO_INCREMENT=1

emerson@paduan.pro.br

SQL Language

UPDATE <nome-tabela> SET <nome-coluna> = <novo conteúdo para o campo> [<nome-coluna> = <novo conteúdo para o campo>] [WHERE <condição>]



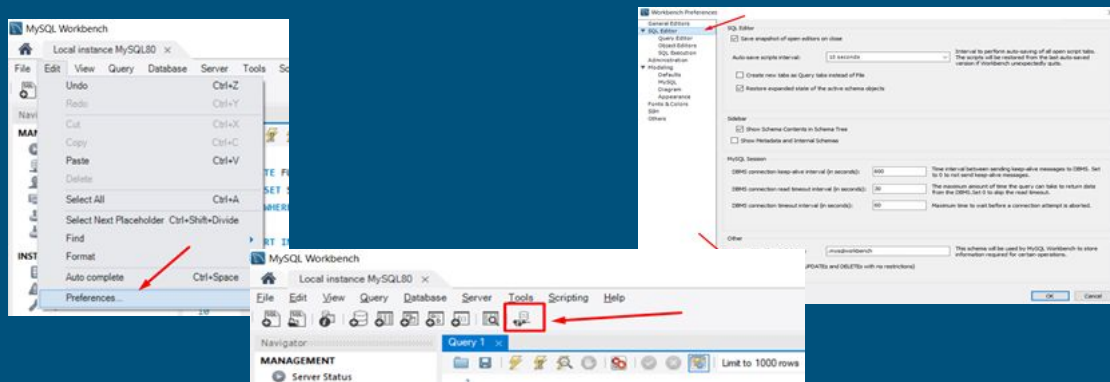
The screenshot shows a SQL query editor with a toolbar at the top. The query text is as follows:

```
1
2 • UPDATE FUNCIONARIOS
3     SET SALARIO = SALARIO * 1.1
4     WHERE DEPARTAMENTO="TI";
```

emerson@paduan.pro.br

SQL Language

Para que o Update funcione no Workbench é preciso realizar as seguintes alterações:



emerson@paduan.pro.br

SQL Language

DELETE FROM <nome-tabela> [WHERE <condição>]



emerson@paduan.pro.br

SQL Language

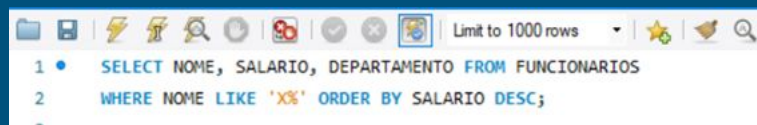
```
SELECT tabela1.coluna, tabela2.coluna
FROM tabela1
[CROSS JOIN tabela2] |
[NATURAL JOIN tabela2] |
[JOIN tabela2 USING (nome_coluna)] |
[JOIN tabela2
ON(tabela1.nome_coluna = tabela2.nome_coluna)] |
[LEFT|RIGHT|FULL OUTER JOIN tabela2
ON (tabela1.nome_coluna = tabela2.nome_coluna)];
```

emerson@paduan.pro.br

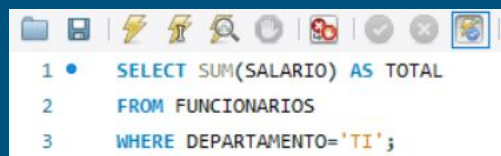
SQL Language



1 • SELECT * FROM FUNCIONARIOS;



1 • SELECT NOME, SALARIO, DEPARTAMENTO FROM FUNCIONARIOS
2 WHERE NOME LIKE 'XX' ORDER BY SALARIO DESC;
3



1 • SELECT SUM(SALARIO) AS TOTAL
2 FROM FUNCIONARIOS
3 WHERE DEPARTAMENTO='TI';

emerson@paduan.pro.br

SQL Language

SELECT

Pesquisas em múltiplas tabelas

Se faz necessário quando precisamos recuperar informações que estão distribuídas em uma ou várias tabelas, a esta necessidade damos o nome de JUNÇÕES(JOIN).

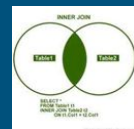
A operação JUNÇÃO (JOIN) é usada para combinar registros relacionados, dentro de um único registro. Esta operação nos permite processar os relacionamentos entre as relações.

emerson@paduan.pro.br

SQL Language

SELECT – INNER JOIN

Pesquisas em múltiplas tabelas



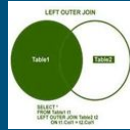
Nesse tipo de junção, caso sejam unidas duas tabelas serão exibidos todos os dados existentes nas duas tabelas envolvidas na consulta.

```
Limit to 1000 rows
1 • INSERT INTO DEPENDENTES VALUES (1,'FILH01', str_to_date('26/11/1998','%d/%m/%Y'), 2);
2 • INSERT INTO DEPENDENTES VALUES (2,'FILH02', str_to_date('23/01/2020','%d/%m/%Y'), 2);
3 • INSERT INTO DEPENDENTES VALUES (3,'FILH01 - OUTRO', str_to_date('20/04/2017','%d/%m/%Y'), 3);
4
5 • SELECT D.NOME AS FILHO,
6       date_format(D.DTNASCIMENTO, '%d/%m/%Y') AS NASCIMENTO,
7       F.NOME AS COLABORADOR
8       FROM FUNCIONARIOS AS F
9       INNER JOIN DEPENDENTES AS D
10      ON F.CODIGO=D.CODFUNCIONARIO
11     WHERE F.NOME='XPTO';
```

emerson@paduan.pro.br

SQL Language

SELECT – LEFT JOIN



Todas as linhas da tabela à esquerda serão recuperadas, independentemente da existência de ocorrências relacionadas na tabela da direita. Preserva as linhas sem correspondência da primeira tabela (esquerda), juntando-as com a linha nula da segunda tabela(direita).

```
1 • INSERT INTO FUNCIONARIOS
2 (NOME, SALARIO, DTNASCIMENTO, DTADMISSAO, ENDereco, DEPARTAMENTO)
3 VALUES
4 ("REGINA", 30000, str_to_date("18/10/1992", "%d/%m/%Y"), str_to_date("04/05/2020", "%d/%m/%Y"), "AVENIDA PAULISTA", "TI");
5
6 • SELECT D.NOME AS FILHO,
7       date_format(D.DTNASCIMENTO, '%d/%m/%Y') AS NASCIMENTO,
8       F.NOME AS COLABORADOR
9 FROM FUNCIONARIOS AS F
10 LEFT JOIN DEPENDENTES AS D
11 ON F.CODIGO=D.CODFUNCIONARIO;
```

emerson@paduan.pro.br

SQL Language

Outros comandos:

SHOW DATABASES

Exibe todos os bancos de dados que o usuário tenha acesso.

DESC <TABELA>

Exibe a estrutura da tabela informada.

CURDATE()

Retorna a data atual do sistema e pode ser utilizado em select's, insert's e etc.

emerson@paduan.pro.br