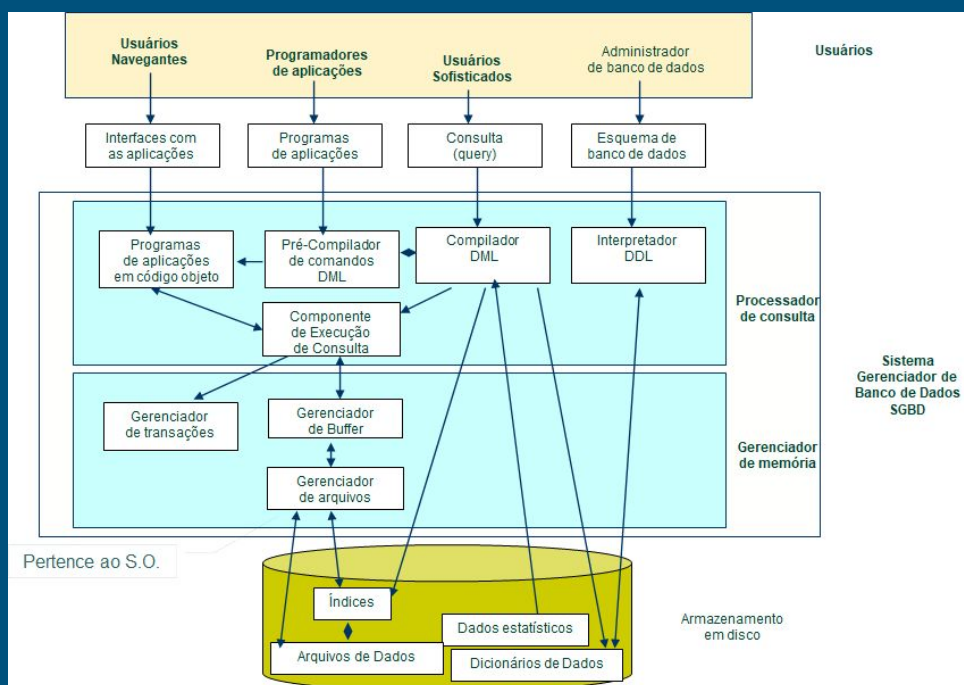


Introdução à Programação

Banco de Dados

emerson@paduan.pro.br

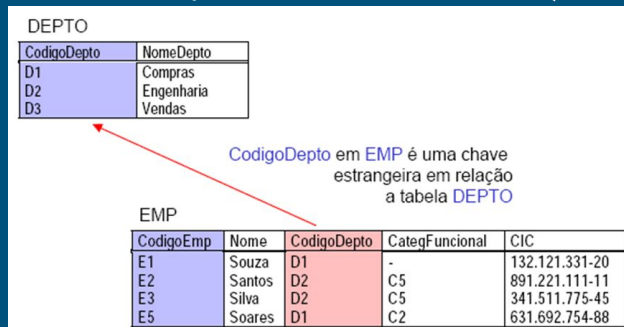


emerson@paduan.pro.br

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

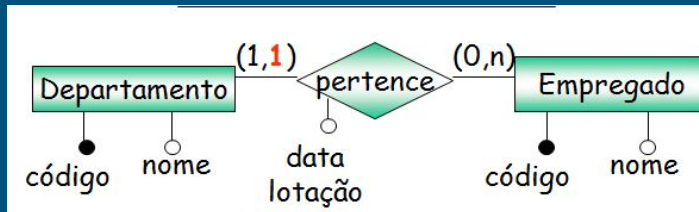
emerson@paduan.pro.br

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 1 : n



Departamento (CodDept, Nome)

Empregado (CodEmp, Nome, CodDept)

CodDept referencia Departamento

emerson@paduan.pro.br

SGBD MySQL



emerson@paduan.pro.br

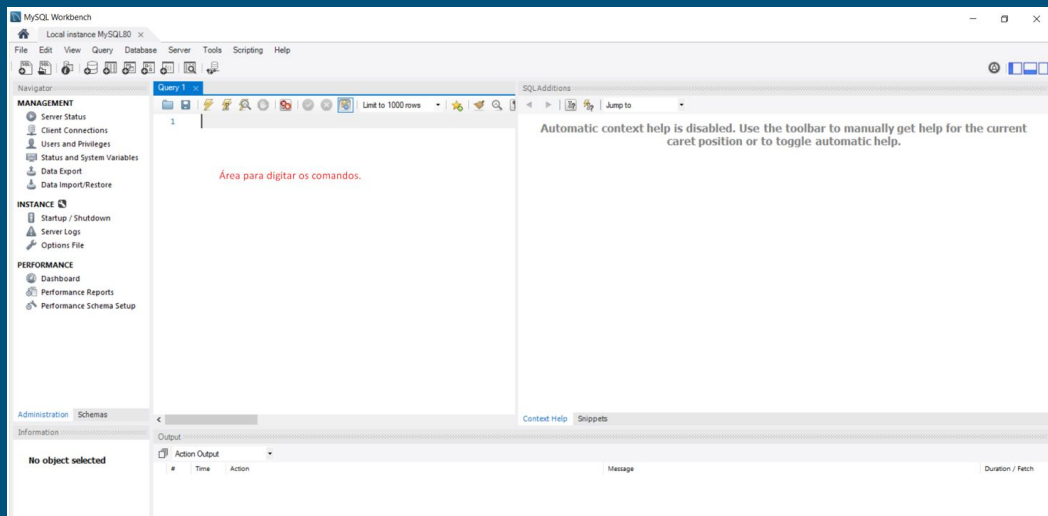
Tipos de dados

MySQL DATA TYPES

DATE TYPE	SPEC	DATA TYPE	SPEC
CHAR	String (0 - 255)	INT	Integer (-2147483648 to 2147483647)
VARCHAR	String (0 - 255)	BIGINT	Integer (-9223372036854775808 to 9223372036854775807)
TINYTEXT	String (0 - 255)	FLOAT	Decimal (precise to 23 digits)
TEXT	String (0 - 65535)	DOUBLE	Decimal (24 to 53 digits)
BLOB	String (0 - 65535)	DECIMAL	"DOUBLE" stored as string
MEDIUMTEXT	String (0 - 16777215)	DATE	YYYY-MM-DD
MEDIUMBLOB	String (0 - 16777215)	DATETIME	YYYY-MM-DD HH:MM:SS
LONGTEXT	String (0 - 4294967295)	TIMESTAMP	YYYYMMDDHHMMSS
LOBLOB	String (0 - 4294967295)	TIME	HH:MM:SS
TINYINT	Integer (-128 to 127)	ENUM	One of preset options
SMALLINT	Integer (-32768 to 32767)	SET	Selection of preset options
MEDIUMINT	Integer (-8388608 to 8388607)	BOOLEAN	TINYINT(1)

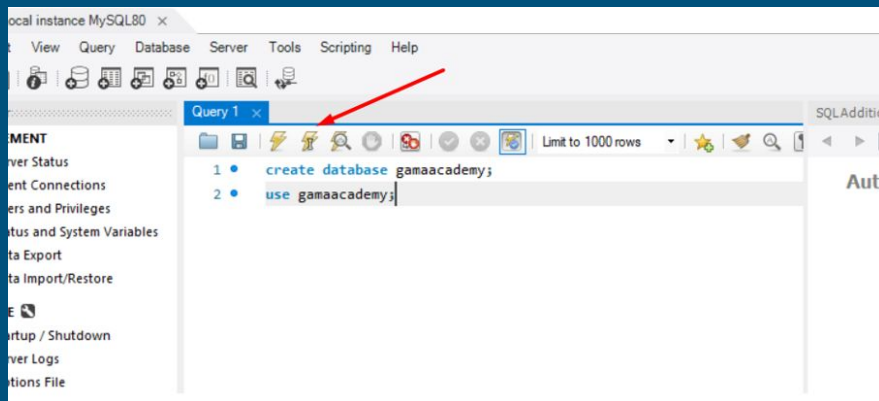
emerson@paduan.pro.br

MySQL Workbench



emerson@paduan.pro.br

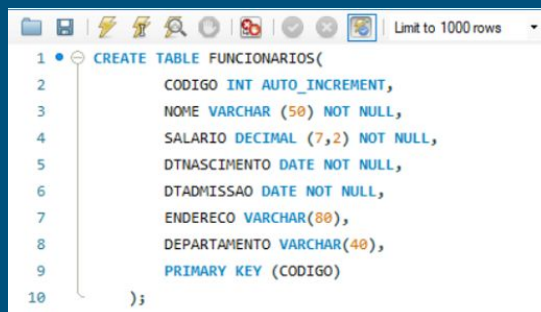
SQL Language



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

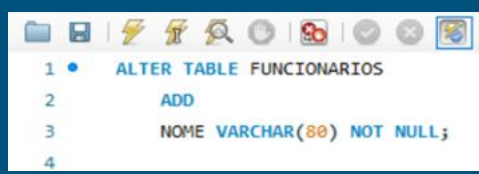
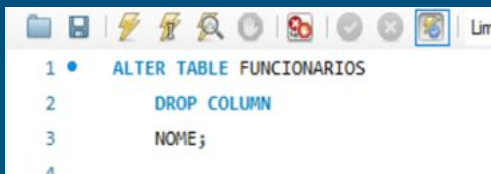
SQL Language

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.

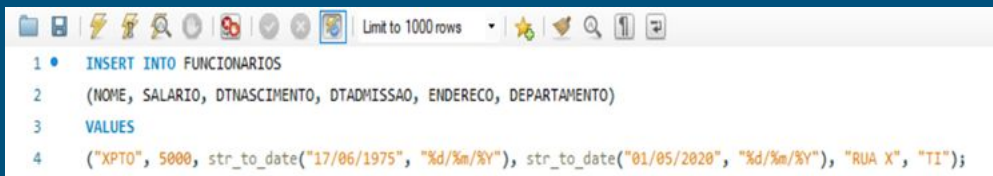
Sintaxe:

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

emerson@paduan.pro.br

SQL Language

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



```
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");
```

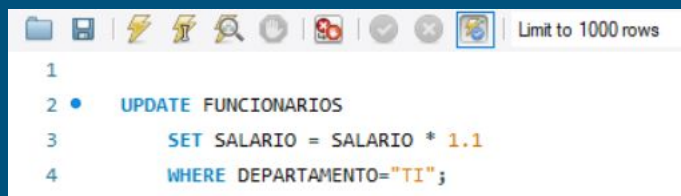
* Para zerar o auto incremento:

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>]

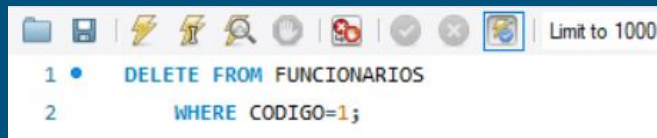


```
1
2 • UPDATE FUNCIONARIOS
3     SET SALARIO = SALARIO * 1.1
4     WHERE DEPARTAMENTO="TI";
```

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

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).

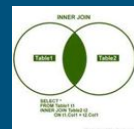
A operação JUNÇÃO (JOIN) é usada para combinar registros relacionados, dentro de um único registro. Esta operação nos permite processar os relacionamentos entre as relações.

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