



Banco de Dados

📅 Created	@May 3, 2022 7:00 PM
👤 Created by	
🏷️ Tags	Banco de Dados
☰ Property	

SQL

Tipos de dados:

- **Caracteres:**

- Char(n) → datatype que aceita como valor qualquer dígito, sendo que o espaço ocupado no disco é de um dígito por caractere. É possível utilizar até 8mil dígitos.
- Varchar(n) → datatype que aceita como valor qualquer dígito, sendo que o espaço ocupado no disco é de um dígito por caractere. É possível utilizar até 8mil dígitos. A diferença pro Char é que usamos Varchar quando não sabemos o tamanho fixo do campo.
- Text → qualquer dígito pode ser usado neste datatype, sendo ocupado 1 byte por caractere, o equivalente a 2.147.483.647 bytes.

- **Numéricos Inteiros:**

- Bigint – Aceita valores entre -2^{63} e $2^{63}-1$, sendo que esse datatype ocupa 8 bytes.
- Int – Os valores aceitos aqui variam entre -2^{31} a $2^{31}-1$. Ocupa 4 bytes.
- Smallint – Aceita valores entre -32768 até 32767 e ocupa 2 bytes.
- Tinyint – Os valores aceitos aqui variam entre 0 e 255, ocupa apenas 1 byte.

- Bit – É um tipo de dado inteiro (conhecido também como booleano), cujo valor pode corresponder a NULL, 0 ou 1. Podemos converter valores de string TRUE e FALSE em valores de bit, sendo que TRUE corresponde a 1 e FALSE a 0.
- **Numéricos Exatos:**
 - Decimal(P,S) – Os valores aceitos variam entre -10^{38-1} e 10^{38-1} , sendo que o espaço ocupado varia de acordo com a precisão. Se a precisão for de 1 a 9, o espaço ocupado é de 5 bytes. Se a precisão é de 10 a 19, o espaço ocupado é de 9 bytes, já se a precisão for de 20 a 28, o espaço ocupado é de 13 bytes, e se a precisão for de 29 a 38, o espaço ocupado é de 17 bytes.
 - Numérico(P,S) – Considerado um sinônimo do datatype decimal, o numérico também permite valores entre -10^{38-1} e 10^{38-1} e o espaço ocupado é o mesmo do anterior.
- **Numéricos Aproximados:**
 - Float[(n)] – O mesmo que double precision quando o valor de n é 53, este datatype aceita valores entre $-1.79E + 308$ e $1.79E + 308$. O espaço ocupado varia de acordo com o valor de n. Se esse valor estiver entre 1 e 24, a precisão será de 7 dígitos, sendo que o espaço ocupado será de 4 bytes. Se o valor de n estiver entre 25 e 53, sua precisão será de 15 dígitos, assim sendo o espaço ocupado será de 8 bytes.
 - Real – Este datatype é similar ao float(n) quando o valor de n é 24. Os valores aceitos variam entre $-3.40E + 38$ e $3.40E + 38$. Esse datatype ocupa 4 bytes.
- **Numéricos Monetários:**
 - Money – Este datatype aceita valores entre -2^{63} e 2^{63-1} , sendo que 8 bytes são ocupados.

- Smallmoney – É possível usar valores entre -2^{31} e $2^{31}-1$, sendo que 4 bytes são ocupados.
- **Data e Hora:**
 - Datetime – Permite o uso de valores entre 1/1/1753 e 31/12/9999. Este datatype ocupa 8 bytes e sua precisão atinge 3.33 milissegundos.
 - Smalldatetime – Aceita o uso de valores entre 1/1/1900 e 06/06/2079, sendo que sua precisão é de 1 minuto e ocupa 4 bytes em disco.
- Binários
- Especiais

SQL E POSTGRE

Database - Aula 1

Schemas → Query Tool → comandos

1 - Construir as tabelas para o seguinte esquema:

Fornecedor(codigo, nome, cidade);

Venda(codPeca, codForn, quantidade, data);

Peca(codPeca, nome, descricao);

Restrições:

Nome da peça não pode ser nulo

Quantidade tem que ser maior que 0

```
create table Fornecedor(
codigo int primary key,
nome varchar (100),
cidade varchar (100)
);

create table Peca(
codPeca int primary key,
nome varchar (50) not null,
descricao varchar (300)
);
```

```

create table Venda (
codPeca int not null,
codigo int not null,
quantidade int check (quantidade > 0),
data date,
constraint fk_codPeca foreign key (codPeca) references Peca(codPeca),
constraint fk_codigo foreign key (codigo) references Fornecedor(codigo)
);

insert into Peca values (1, 'Chave de fenda', 'Chave para parafusos');
insert into Fornecedor values (1, 'João Oliveira', 'Porto Alegre');
insert into Venda values (1,1,10,'2022-05-05');
insert into Venda values (1,1,10,'2022-05-06');

```

2 - Construir as tabelas para o seguinte esquema:

Aluno(Nome, RA, Idade, DataNasc)

Professor (Nome, NFunc, Idade, Titulação)

Disciplina(Sigla, Nome, NCred, NFunc, Livro)

Turma(Sigla, Numero, NAlunos)

Matrícula(Sigla, Numero, RA, Ano, Nota)

```

create table Aluno (
nome varchar (100) not null,
ra int,
dataNasc date,
constraint pk_ra primary key (ra)/ chave primaria que não esta na primeira linha - no final /
);
create table Professor (
nome varchar (100) not null,
nFunc int,
idade int,
titulacao varchar (100),
primary key (nFunc) / outra forma de declarar chave primaria /
);
create table Disciplina (
sigla varchar (10),
nome varchar (50),
nCred int,
nFunc int,
livro varchar (100),
primary key (sigla),
constraint fk_nFunc foreign key (nFunc) references Professor(nFunc) / referencia chave estrangeira /
);
create table Turma (
sigla varchar (10),
numero int,
nAlunos int,

```

```
primary key (numero),  
constraint fk_sigla foreign key (sigla) references Disciplina(sigla)  
);  
create table Matricula (  
sigla varchar (10),  
numero int,  
ra int,  
ano date,  
nota float,  
constraint fk_sigla foreign key (sigla) references Disciplina(sigla),  
constraint fk_numero foreign key (numero) references Turma(numero),  
constraint fk_ra foreign key (ra) references Aluno(ra)  
);
```