



Projeto de Banco de Dados

Prof^a: Flavia Garcia

AULA 4

SQL

- Structured Query Language
- Linguagem utilizada por vários SGBD's como a linguagem padrão para o acesso às bases de dados.
- Linguagem padrão para a manipulação de dados em ambiente relacional.

SQL

- Considerando o modelo lógico abaixo:

CLIENTE (cod_cli, nome_cli, endereco, cidade, cep, uf)

VENDEDOR (cod_vend, nome_vend, sal_fixo, faixa_comiss)

PEDIDO (num_ped, prazo_entr, cd_cli, cd_vend)

ITEM_PEDIDO (no_ped, cd_prod, qtd_ped)

PRODUTO (cod_prod, unid_prod, desc_prod, val_unit)

SQL

- **Linguagem de definição de dados (DDL)**
 - Permite ao usuário a definição da estrutura e organização dos dados armazenados, e das relações existentes entre eles.
- **Linguagem de manipulação de dados (DML)**
 - Permite a um usuário, ou a um programa de aplicação, a execução de operações de inclusão, remoção, seleção ou atualização de dados previamente armazenados na base de dados.

Criação/Exclusão do Banco de Dados

- Create database

- Sintaxe:

- Create database pedido;

- Drop database

- Sintaxe:

- Drop database pedido;

Linguagem de definição de dados (DDL)

- CREATE TABLE

- Comando utilizado para a criação de tabelas.

- Sintaxe:

- CREATE TABLE <nome_tabela>
(<descrição das colunas>,
<descrição das chaves>);

Linguagem de definição de dados (DDL)

- CREATE TABLE

- <nome_tabela> dever ser substituído pelo nome da tabela a ser criada.
- <descrição das colunas> deve ser substituída pela relação das colunas da tabela e seus respectivos tipos de dados (por exemplo, smallint, char, varchar, integer, number, float e etc).
- <descrição das chaves> deve ser substituída pela lista das colunas que são tratadas como chaves estrangeiras.

Linguagem de definição de dados (DDL)

- Exemplo:

```
create table cliente (  
    cod_cli    smallint    not null,  
    nome_cli   varchar(40) not null,  
    endereco  varchar(40)   null,  
    cidade    varchar(20)   null,  
    cep       char(08)      null,  
    uf        char(02)      null,  
    primary key (cod_cli));
```


Linguagem de definição de dados (DDL)

- Exemplo:

```
create table vendedor(  
    cod_vend    smallint    not null,  
    nome_vend   varchar(40) not null,  
    sal_fixo    decimal(9,2) not null,  
    faixa_comiss char(01)   not null,  
    primary key (cod_vend));
```

Linguagem de definição de dados (DDL)

- Exemplo:

```
create table produto(  
    cod_prod    smallint    not null,  
    unid_prod   char(03)    not null,  
    desc_prod   varchar(20) not null,  
    val_unit    number(9,2) not null,  
    primary key (cod_prod));
```

Linguagem de definição de dados (DDL)

- Exemplo:

```
create table pedido(  
    num_ped    smallint  not null,  
    prazo_entr smallint  not null,  
    cd_cli     smallint  not null,  
    cd_vend    smallint  not null,  
    Primary Key (num_ped),  
    Foreign Key (cd_cli) References Cliente (cod_cli),  
    Foreign Key (cd_vend) References Vendedor(cod_vend));
```

Linguagem de definição de dados (DDL)

- Exemplo:

```
create table item_pedido(  
    no_ped    smallint  not null,  
    cd_prod   smallint  not null,  
    qtd_ped   float     not null,  
    Primary Key (no_ped , cd_prod),  
    Foreign Key (no_ped) References Pedido (num_ped),  
    Foreign Key (cd_prod) References Produto (cod_prod));
```

Criação de Chave Primária

- Sintaxe:

Primary Key (<nome dos campos>)

- Exemplos:

Primary Key (cd_prod)

Primary Key (no_ped , cd_prod)

Criação de Chave Estrangeira

- Sintaxe:

Foreign Key <nome da chave estrangeira> (coluna)

References <nome da tabela> (coluna)

On Update <ação>

On Delete <ação>

Opção	Descrição
<i>nome-chave-estrangeira</i>	Nome opcional da constraint.
<i>lista-de-colunas</i>	Lista de colunas da tabela que faz referência a outra tabela.
<i>nome-tabela</i>	Nome da tabela em que está a chave primária.

Criação de Chave Estrangeira

- Onde:

Opção	Descrição
<i>nome-chave-estrangeira</i>	Nome opcional da constraint.
<i>lista-de-colunas</i>	Lista de colunas da tabela que faz referência a outra tabela.
<i>nome-tabela</i>	Nome da tabela em que está a chave primária.
<i>ação</i>	Determina qual ação o banco de dados deve tomar quando for excluída ou alterada uma linha da tabela que contém referência a esta chave. Pode ser SET NULL (altera o conteúdo da coluna para Nulo, perdendo a referência, sem deixar valores inconsistentes), SET DEFAULT (altera o conteúdo da coluna para o valor especificado na cláusula DEFAULT se houver), CASCADE (exclui ou altera todos os registros que se relacionam a eles), NO ACTION (em caso de alteração, não modifica os valores que se relacionam a eles) ou RESTRICT (não permite a exclusão da chave primária).

Linguagem de definição de dados (DDL)

- ALTER TABLE

- Permite a alteração na estrutura da tabela.

- Acrescentar novas colunas

```
ALTER TABLE tabela  
  ADD nome-coluna tipo-de-dado constraints [, nome-coluna tipo-de-  
    dado constraints, ...]
```

- Adicionar chave primária

```
ALTER TABLE cliente  
  ADD PRIMARY KEY (CDCLIENTE)
```


Linguagem de definição de dados (DDL)

- Modificar colunas:

```
ALTER TABLE cliente  
    MODIFY email varchar(100) NOT NULL
```

Linguagem de definição de dados (DDL)

Exemplo 1 - Exclusão de coluna:

```
ALTER TABLE cliente  
DELETE email
```

Exemplo 2 - Exclusão de constraint de tabela:

```
ALTER TABLE cliente  
DELETE primary key
```

Exemplo 3 - Exclusão de constraint de tabela:

```
ALTER TABLE cliente  
DELETE FOREIGN KEY pedido_cliente_fk
```

Linguagem de definição de dados (DDL)

- DROP TABLE

- Comando que permite a exclusão de tabelas do banco de dados.
- Sintaxe:

DROP TABLE <nome_tabela>;

onde:

<nome_tabela> dever ser substituído pelo nome da tabela a ser excluída.

Linguagem de definição de dados (DDL)

- Exemplos:

```
drop table item_pedido;
```

```
drop table pedido;
```

```
drop table vendedor;
```

```
drop table produto;
```

```
drop table cliente;
```

Tipos de Dados

Tipo de Dado	Descrição
INTEGER ou INT	Número positivo ou negativo inteiro. O número de bytes que pode ser utilizado varia em função do banco de dados utilizado.
SMALLINT	Mesma função do INTEGER, mas ocupa cerca da metade do espaço.
NUMERIC	Número positivo ou negativo de ponto flutuante. Normalmente, deve-se informar o tamanho total do campo e definir quantas casas decimais devem ser armazenadas após a vírgula.
DECIMAL	Semelhante ao NUMERIC, mas, em alguns bancos de dados, poderá ter uma maior precisão após a vírgula.
REAL	Número de ponto flutuante de simples precisão. A diferença básica é que os valores serão armazenados em representação exponencial, portanto serão arredondados para o nível mais próximo de precisão.
DOUBLE PRECISION	Número de ponto flutuante de dupla precisão. Comporta-se como o REAL, mas permite maior aproximação de resultados.
FLOAT	Número de ponto flutuante em que você define o nível de precisão (número de dígitos significativos).

Tipos de Dados

DATE	Permite armazenar datas.
TIME	Permite armazenar horários.
TIMESTAMP	Permite armazenar uma combinação de data e hora.
CHARACTER ou CHAR	Permite armazenar cadeias de caracteres (letras, símbolos e números). O tamanho deve ser informado e será fixo, ou seja, mesmo que não utilizado totalmente, será ocupado o espaço fisicamente. O valor definido será o tamanho máximo da cadeia armazenada.
CHARACTER VARYING ou VARCHAR	Permite armazenar cadeias de caracteres, mas com tamanho variável. Nesse caso, especifica-se o tamanho máximo da coluna. Se for utilizado menos espaço que o máximo definido, o espaço restante não será ocupado.