



Banco de dados 1

Linguagem SQL – DDL e DML

Professor: Victor Hugo L. Lopes



Banco de Dados 1

Agenda:
Introdução à linguagem
de dados;
DDL;
DML;
CRUD;
Introdução à linguagem
SQL.





Banco de Dados 1

Linguagem de consulta - SQL

Structure Query Language

Implementa conceitos dos BDs relacionais

Cria/altera/remove recursos físicos

Gerenciamento do SGBD

Salva/altera/remove/consulta dados em BDs

Dois grandes grupos:

- DDL: Linguagem de Definição de Dados
- DML: Linguagem de Manipulação de Dados



Banco de Dados 1

Linguagem de consulta - SQL

Por que precisamos da linguagem SQL?

A algebra relacional fornece uma notação formal e concisa para representar consultas em um banco de dados, entretanto, os sistemas de bancos de dados comerciais exigem uma linguagem mais amigável.



Banco de Dados 1

Linguagem de consulta - SQL

O que é a linguagem SQL?

Embora seja referida como uma linguagem de consulta estruturada (SQL: Structured Query Language, ou linguagem de consulta estruturada), ela permite fazer mais do que consultas em um banco de dados.

É uma linguagem que permite definir a estrutura dos dados, modificar dados em um bd e especificar regras de restrições em modelos físicos.



Banco de Dados 1

Linguagem de consulta - SQL

De onde veio a SQL?

A IBM desenvolveu a versão original da SQL, originalmente se chamando Sequel, na década de 1970.

Muitos produtos hoje aceitam a linguagem SQL, que se estabeleceu como linguagem padrão de bancos de dados relacionais.

O American National Standards Institute e a International Organization for Standardization passaram a publicar padrões SQL, como o SQL-86, SQL-89, SQL-2003...



Banco de Dados 1

Linguagem de consulta - SQL

A linguagem se divide em:

Linguagem de definição de dados (DDL)

- Criação e gerenciamento de esquemas
- Gestão das integridades
- Construção de recursos extras, como views
- Gestão de autorização

Linguagem de manipulação de dados (DML)

- Toda manipulação possível com os dados



Banco de Dados 1

Linguagem de consulta - SQL

Definição de dados

O conjunto de relações em um BD precisa ser especificado para o sistema por meio de uma linguagem de definição de dados, a DDL.

Permite a criação dos esquemas para cada relação, o domínio dos valores associados a cada atributo, as restrições de integridade, o conjunto dos índices a serem mantidos para cada relação, bem como as informações de segurança e autorização e estrutura de armazenamento físico.



Banco de Dados 1

Linguagem de consulta - SQL

Tipos de domínios básicos (Depende do SGBD e de sua versão)*

- Numéricos
- Data e Hora
- Textuais
- Tipos especiais (arquivos, espaciais...)

<https://dev.mysql.com/doc/refman/5.7/en/data-types.html>



Banco de Dados 1

Linguagem de consulta - SQL

Uma breve parada!!!

Conhecendo o SGBD MySQL



Banco de Dados 1

Linguagem de consulta - SQL

*

fazendo uso do SGBD via terminal

Windows:

Acesse a pasta do postgresql, que provavelmente está em <pasta de instalação>\bin

- `mysql -u nome_usuario -p <enter>`

Linux (ou Windows com Path definido):

- `mysql -u nome_usuario -p <enter>`
- `mysql -h <ip> -u nome_usuario -p <enter>`



Banco de Dados 1

Linguagem de consulta - SQL

```
C:\Arquivos de Programas\MySQL\MySQL Server 8.0\bin>mysql -u root -p
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 14
Server version: 8.0.16 MySQL Community Server - GPL

Copyright (c) 2000, 2019, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

```
Prompt de Comando - mysql -u root -p

C:\>mysql -u root -p
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 9
Server version: 8.0.16 MySQL Community Server - GPL

Copyright (c) 2000, 2019, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```



Banco de Dados 1

Linguagem de consulta - SQL

*

fazendo uso do SGBD via terminal

Se o login for bem sucedido, será apresentado o nome do usuário logado seguido de um símbolo composto por dois caracteres, conforme:

mysql>_ Indica que você está logado, aguardando comando

Os comandos DEVEM ser terminados com “.”;

Comandos podem ser inseridos em múltiplas linhas:

```
mysql> select * from tabela  
-> inner join outra_tabela;
```



Banco de Dados 1

Linguagem de consulta - SQL

fazendo uso do SGBD via terminal

<code>\q</code>	- sair (ou exit;)
<code>\u <nome_banco></code>	-muda o banco de dados (ou use <nome_banco>;)
<code>\s</code>	- status do SGBD (ou status;)
<code>\c</code>	- limpar comando (ou clear;)
<code>\h</code>	- help (ou help;)
<code>show databases;</code>	- lista os Bds;
<code>show tables;</code>	- lista as tabelas de um BD (precisa estar selecionado);
<code>Describe <tabela>;</code>	- mostra a estrutura da tabela;
<code>Select database();</code>	- mostrar o BD atual;



Banco de Dados 1

Linguagem de consulta - SQL

Retornando de onde paramos!

1. Acesse o MySQL;
 - `mysql -u <usuario> -p <enter>`
2. Crie o banco, ou alterne para um existente:
 - `Create database <nome>;`
 - `\u <nome> ou use <nome>;`



Banco de Dados 1

Linguagem de consulta - SQL

Definição básica de esquema de relação*

Utiliza-se o comando CREATE TABLE:

```
create          table          <nome_da_relacao>  
(<sequencia_de_campos>, <sequencia_de_restricoes>);
```




Banco de Dados 1

Linguagem de consulta - SQL

Definição básica de esquema de relação

ex.:

```
create table ALUNO (  
    matricula int not null auto_increment,  
    nome varchar(100),  
    nascimento date,  
    primary key (matricula)  
);
```



Banco de Dados 1

Linguagem de consulta - SQL

Definindo a chave primária como auto incremento em **postgresql**:

```
create table ALUNO (  
    matricula serial not null,  
    primary key (matricula)  
);
```



Banco de Dados 1

Linguagem de consulta - SQL

Criando uma relação com base em outra existente:

```
CREATE TABLE new_tbl LIKE orig_tbl;
```

Ou

```
CREATE TABLE new_tbl AS SELECT * FROM orig_tbl;
```



Banco de Dados 1

Linguagem de consulta - SQL

Definindo valores *default**:

```
CREATE TABLE estudante (  
    idEstudante int not null auto_increment primary key,  
    matricula int(20) default '10'  
);
```

```
CREATE TABLE t1 (  
    i INT DEFAULT -1,  
    c VARCHAR(10) DEFAULT "",  
    price DOUBLE(16,2) DEFAULT '0.00'  
);
```



Banco de Dados 1

Linguagem de consulta - SQL

Definindo restrições (*constraint*)*:

Unique – Unicidade

```
Create table pessoa (  
    idPessoa int,  
    nome varchar(100) UNIQUE  
);
```

OU

```
create table pessoa(  
    idPessoa int,  
    nome varchar(100));
```

```
Create UNIQUE index unq_nome on pessoa (nome);
```

A coluna “nome” não aceitará valores duplicados.



Banco de Dados 1

Linguagem de consulta - SQL

Definindo restrições (*constraint*)*:

Primary Key – Chave primária

```
Create table pessoa (  
    idPessoa int primary key,  
    nome varchar(100) UNIQUE  
);
```

OU

```
create table pessoa(  
    idPessoa int not null auto_increment,  
    nome varchar(100), primary key (idPessoa));
```

```
Alter table pessoa add PRIMARY KEY (idPessoa);
```

```
Alter table pessoa modify idPessoa int(11) AUTO_INCREMENT primary  
key;
```



Banco de Dados 1

Linguagem de consulta - SQL

Definindo restrições (*constraint*)*:

Foreign Key – Chave estrangeira

```
[CONSTRAINT [symbol]] FOREIGN KEY  
    [index_name] (col_name, ...)  
    REFERENCES tbl_name (col_name,...)  
    [ON DELETE reference_option]  
    [ON UPDATE reference_option]
```

reference_option:

RESTRICT | CASCADE | SET NULL | NO ACTION | SET DEFAULT



Banco de Dados 1

Linguagem de consulta - SQL

Definindo restrições (*constraint*)*:

Foreign Key – Chave estrangeira

```
CREATE TABLE parent (  
    id INT NOT NULL,  
    PRIMARY KEY (id)  
) ENGINE=INNODB;
```

```
CREATE TABLE child (  
    id INT,  
    parent_id INT, constraint  
    FOREIGN KEY (parent_id)  
        REFERENCES parent(id)  
        ON DELETE CASCADE  
) ENGINE=INNODB;
```




Banco de Dados 1

Linguagem de consulta - SQL

Definindo restrições (*constraint*)*:

Foreign Key – Chave estrangeira

E se a tabela já existir?

```
Alter table <tabela> add constraint foreign key [nome]  
(coluna) references <tabela_pai> (coluna)  
On delete [regra]  
On update [regra];
```

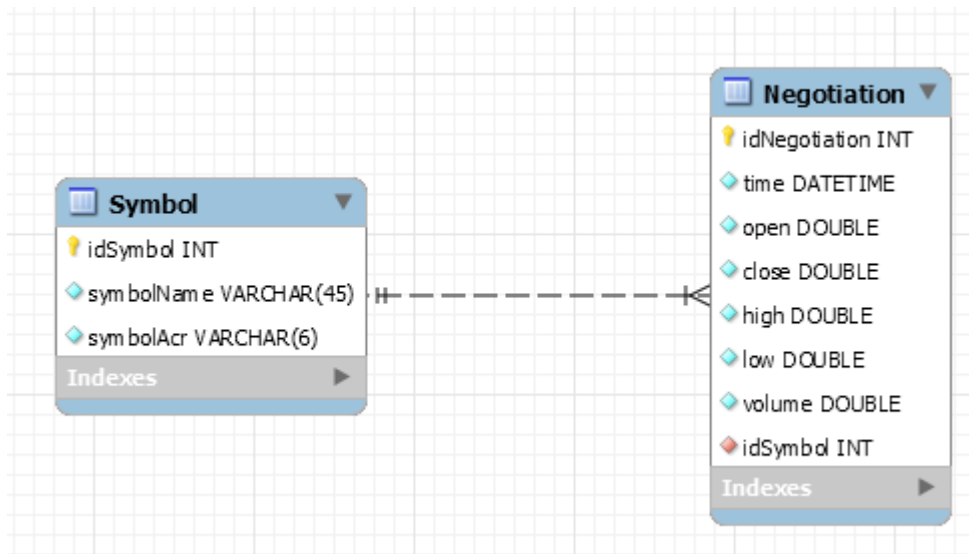


Banco de Dados 1

Linguagem de consulta - SQL

exercitando 1

Implemente o BD abaixo usando DDL em MySQL





Banco de Dados 1

Linguagem de consulta - SQL

exercitando 2

Dado o DDL abaixo, desenhe as relações usando DER

```
mysql> create table fornecedor(  
-> idFornecedor int not null auto_increment,  
-> nomeFornecedor varchar(100) not null UNIQUE,  
-> PRIMARY KEY (idfornecedor));
```

```
mysql> create table produto(  
-> idProduto int not null auto_increment,  
-> nomeProduto varchar(100) not null,  
-> codigoProduto int not null,  
-> precoProduto double,  
-> idFornecedor int not null,  
-> PRIMARY KEY (idProduto));
```

```
mysql> alter table produto  
-> add constraint foreign key fk_fornecedor (idFornecedor)  
-> references fornecedor (idFornecedor)  
-> ON DELETE RESTRICT  
-> ON UPDATE CASCADE;
```



Banco de Dados 1

Linguagem de consulta - SQL

exercitando 3

Projete um banco de dados para a sua agenda de compromissos diários. Após completo o modelo lógico, implemente suas relações no banco de dados.



Banco de Dados 1

Linguagem de consulta - SQL

Deletando uma relação

`drop table <nome_tabela>;`

Exclui toda a relação

ex.:

`Drop table aluno;`



Banco de Dados 1

Linguagem de consulta - SQL

Alterando uma relação

O comando `alter table` serve para modificar uma relação, adicionando novas colunas ou restrições à uma relação existente, além de excluir colunas existentes. Sintaxe:

`Alter table <nome_tabela> <ações a seguir>`

ex.:

```
alter table tabela1 add column telefone varchar(13);
```



Banco de Dados 1

Linguagem de consulta - SQL

Alterando uma relação

Alterando o nome de uma coluna da relação.

```
Alter table table_name change old_column_name  
new_column_name datatype(length);
```

ex.:

```
alter table tabela1 change telefone tele varchar(20);  
Alter table table1 change No id int(5),  
modify Image img varchar(40)
```



Banco de Dados 1

Linguagem de consulta - SQL

Alterando uma relação

Alterando o tipo de dados de uma coluna da relação.

```
Alter table table_name modify column_name  
datatype(length);
```

ex.:

```
Alter table table1 modify Game_name char(15);
```




Banco de Dados 1

Linguagem de consulta - SQL

Alterando uma relação

Criando restrição de não nulo em coluna existente.

```
ALTER table yourTableName  modify column_name data  
type constraint;
```

ex.:

```
alter table cliente modify nome varchar(100) not null;
```

```
alter table cliente modify nome varchar(100) null;
```



Banco de Dados 1

Linguagem de consulta - SQL

Alterando uma relação

Definindo a chave primária em tabela já existente.

```
Alter      table      <nome_tabela>      add      constraint  
<nome_restricao> primary key (nome_coluna);
```

ex.:

```
alter table cliente add constraint pk_cliente primary key  
(idcliente);
```



Banco de Dados 1

Linguagem de consulta - SQL

Alterando uma relação

Definindo a chave primária auto_increment.

```
ALTER table <table>  modify <column_name> data type  
constraint;
```

OU

```
ALTER  TABLE  <table>  CHANGE  <column_name>  
<column_name> data type auto_increment;
```

ex.:

```
alter table cliente modify idCliente int(11) auto_increment;
```



Banco de Dados 1

Linguagem de consulta - SQL

Alterando uma relação

Removendo o auto_increment.

```
ALTER TABLE mytable CHANGE my_col my_col INT(10)  
UNSIGNED NOT NULL
```

ex.:

```
alter table cliente CHANGE idCliente idCliente int(11) NOT  
NULL;
```



Banco de Dados 1

Linguagem de consulta - SQL

Alterando uma relação

Removendo a chave primária em tabela já existente.

```
Alter    table    <nome_tabela>    drop    constraint  
<nome_restricao>;
```

OU

```
Alter table <nome_tabela> drop PRIMARY KEY;
```

ex.:

```
alter table cliente drop constraint pk_cliente;
```

```
Alter table cliente drop PRIMARY KEY;
```



Banco de Dados 1

Linguagem de consulta - SQL

Alterando uma relação

Excluindo uma coluna de uma relação.

`Alter table <nome_tabela> drop column <coluna>;`

ex.:

`alter table cliente drop column tele;`



Banco de Dados 1

Linguagem de consulta - SQL

Alterando uma relação

Criando relacionamento entre tabelas existentes

```
Alter      table      <nome_tabela>      add      constraint  
<nome_restricao>  foreign  key  (<chave_estrangeira>)  
references <tabela_pai> (<chave_primaria_pai>) on delete  
<regra> on update <regra>;
```

ex.:

```
ALTER TABLE peca ADD CONSTRAINT depto_pecas_fk  
FOREIGN KEY (idDepto)  
REFERENCES depto (idDepto)  
ON DELETE NO ACTION  
ON UPDATE NO ACTION;
```

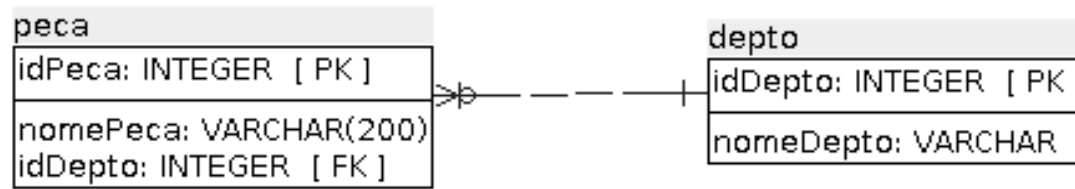


Banco de Dados 1

Linguagem de consulta - SQL

exercitando:

Implemente em MySQL o banco de dados abaixo:



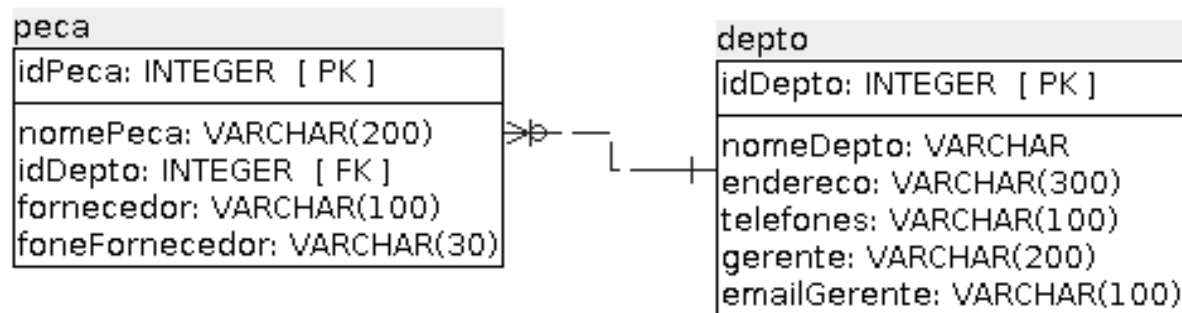


Banco de Dados 1

Linguagem de consulta - SQL

exercitando:

Agora, altere o banco implementado para seguir as alterações fornecidas pelo projetista.





Banco de Dados 1

Linguagem de consulta - SQL

exercitando:

Por fim, normalize as tabelas, e refatore o esquema.

