

SQL

Alessandro Ap. Antonio

Lucas Maestro

SQL (*Structured Query Language*)

- SQL (Linguagem de Consulta Estruturada)
- SQL é uma linguagem padrão para armazenar, manipular e recuperar dados em bancos de dados.
- A linguagem SQL é dividida em 4 subconjuntos de acordo com as operações que queremos efetuar sobre um banco de dados.

SQL

- DDL - Linguagem de Definição de Dados (*Data Definition Language*)
- DML - Linguagem de Manipulação de Dados (*Data Manipulation Language*).
- DCL - Linguagem de Controle de Dados (*Data Control Language*)
- DTL - Linguagem de Transação de Dados (*Data Transaction Language*)

SQL – DDL

- Criação de um Banco de Dados
 - `create database nome_BD`
 - `drop database nome_BD`
- Comandos para criação de Tabelas
 - `create table`
 - define a estrutura da tabela, suas restrições de integridade e cria uma tabela vazia
 - `alter table`
 - modifica a definição de uma tabela
 - Inclui, Exclui e Altera **Atributos** (**Campos**);
 - Inclui e Exclui **RI**s (**Restrições de Integridade**)
 - `drop table`
 - remove uma tabela com todas as suas tuplas

Base de Dados

- Criar base de dados

```
CREATE DATABASE nome_banco;
```

- Excluir base de dados

```
DROP DATABASE nome_banco;
```

- Exibir base de dados

```
SHOW DATABASES;
```

- Acessar base de dados

```
USE nome_banco;
```

Tipos de dados

- **Números Inteiros:**

- TINYINT
- SMALLINT
- MEDIUMINT
- INT or INTEGER
- BIGINT

- **Ponto Flutuante (Reais)**

- FLOAT
- DOUBLE, DOUBLE PRECISION, REAL
- DECIMAL, NUMERIC

- **Datas:**

- DATE
- DATETIME
- TIMESTAMP
- TIME
- YEAR

- **Texto**

- CHAR
- VARCHAR

- **Dados Binários**

- TINYBLOB
- BLOB
- MEDIUMBLOB
- LONGBLOB

- **Listas:**

- ENUM
- SET

```
CREATE TABLE nome_tabela (  
    nome_atributo_1 tipo_1 [[NOT]NULL][UNIQUE]  
    [{, nome_atributo_n    tipo_n}]  
    [, PRIMARY KEY (nome(s)_atributo(s))]   
    [{, FOREIGN KEY (nome_atributo) REFERENCES nome_tabela}] )
```


Exemplos de Criação de Tabelas

```
CREATE TABLE ambulatorios (  
    amb_id integer NOT NULL,  
    amb_andar int NOT NULL,  
    amb_capacidade smallint,  
    PRIMARY KEY(amb_id)  
);
```

```
CREATE TABLE medicos (  
    med_id integer,  
    med_nome varchar(40) NOT NULL,  
    med_idade smallint NOT NULL,  
    med_especialidade varchar(20),  
    med_CPF char(11) UNIQUE,  
    med_cidade varchar(30),  
    amb_id integer,  
    PRIMARY KEY(med_id),  
    FOREIGN KEY(amb_id) REFERENCES ambulatorios (amb_id)  
);
```

SQL – Alter Table

```
ALTER TABLE nome_tabela
```

```
ADD [COLUMN] nome_atributo_1 tipo_1 [{RIs}]  
[{, nome_atributo_n tipo_n [{RIs}]]}]
```

```
MODIFY [COLUMN] nome_atributo_1 tipo_1 [{RIs}]  
[{, nome_atributo_n tipo_n [{RIs}]]}]
```

```
DROP COLUMN nome_atributo_1  
[{, nome_atributo_n }]
```

```
ADD CONSTRAINT nome_RI_1 def_RI_1  
[{, nome_RI_n def_RI_n}]
```

```
DROP CONSTRAINT nome_RI_1  
[{, nome_RI_n}]
```

```
[ADD | DROP] [PRIMARY KEY ... | FOREIGN KEY ...]
```

Exemplos de Alteração De Tabelas

```
ALTER TABLE ambulatorios  
    ADD nome VARCHAR(30)
```

```
ALTER TABLE medicos DROP PRIMARY KEY
```

```
ALTER TABLE pacientes DROP COLUMN doenca, DROP  
    COLUMN cidade
```

```
ALTER TABLE Funcionarios  
    ADD FOREIGN KEY(amb_id)REFERENCES ambulatorios
```

```
ALTER TABLE Funcionarios  
    ADD constraint fk_amb_id  
    FOREIGN KEY(amb_id) REFERENCES ambulatorios
```

SQL – Índices

- Definidos sobre atributos para acelerar consultas a dados
- Índices são definidos automaticamente para chaves primárias

```
CREATE [UNIQUE] INDEX nome_indice ON  
nome_tabela (nome_atributo_1[{, nome_atributo_n }])
```

```
DROP INDEX nome_indice ON nome_tabela
```

Exemplos

- `CREATE UNIQUE INDEX indPac_CPF ON Pacientes (CPF)`
- `DROP INDEX indPac_CPF ON Pacientes`

Conectar e desconectar

Conectando ao banco de dados MySQL

```
shell> mysql -h host -u usuario -p
```

Enter password:

```
mysql>
```

O parâmetro **-h** é utilizado para
Conexão Remota

Não é Necessário em conexão Local

-h host → servidor onde se encontra o banco

-u usuário → nome do usuário do banco

-p → solicita a senha do usuário.

Desconectando ao banco de dados MySQL

Para desconectar do banco basta digitar o comando **quit** ou **exit**

```
mysql> quit  
Bye
```

```
C:\Users\Alessandro>
```

```
mysql> exit  
Bye
```

```
C:\Users\Alessandro>
```

```
mysql> select version(),current_date;
```

```
mysql> select version(),current_date, current_time;
```

version()	current_date	current_time
5.1.30-community	2016-04-27	09:26:14

1 row in set (0.00 sec)

Comando

Titulo das colunas

Resultados da consulta

Resultado do comando e tempo de processamento

Comandos

- Os comando podem ser digitados em uma linha ou em varias linhas.

```
mysql> select user(), now();
```

user()	now()
root@localhost	2016-04-27 10:30:14

```
1 row in set (0.00 sec)
```

```
mysql> select user(),  
-> now();
```

user()	now()
root@localhost	2016-04-27 10:29:23

```
1 row in set (0.00 sec)
```


Status do Prompt MySQL

Prompt	Status
mysql>	Pronto para um novo comando
?>	Aguardando a próxima linha do comando
'>	Aguardando fechar aspas simples
">	Aguardando fechar aspas dupla

Exercícios

1. Crie um BD com nome **Clinica**

2. Crie as seguintes tabelas neste BD, considerando que os atributos em **vermelho** são chaves primárias e os em **azul** são chaves estrangeiras:

Ambulatorios: **amb_id**(int) (não nulo), amb_andar (numeric(3)) (não nulo), amb_capacidade (smallint)

Medicos: **med_id** (int), med_nome (varchar(40)) (não nulo), med_idade (smallint) (não nulo), med_especialidade (char(20)), med_CPF (char(11)) (único), med_cidade (varchar(30)), **amb_id** (int)

Pacientes: **pac_id** (int), pac_nome (varchar(40)) (não nulo), pac_idade (smallint) (não nulo), pac_cidade (char(30)), pac_CPF (char(11)) (único), pac_doenca (varchar(40)) (não nulo)

Funcionarios: **fun_id** (int), fun_nome (varchar(40)) (não nulo), fun_idade (smallint), fun_CPF (char(11)) (único), fun_cidade (varchar(30)), fun_salario (decimal(10,2)), fun_cargo (varchar(20))

Consultas: **med_id**(int), **pac_id** (int), **con_data** (date), **con_hora** (time)

3. Crie a coluna **amb_id** (int) na tabela **Funcionarios**

4. Crie os seguintes índices:

- **Medicos:** CPF (único)
- **Pacientes:** doença

5. Remover o índice **doença** em **Pacientes**

6. Remover as colunas **cargo** e **amb_id** da tabela de **Funcionarios**