

SQL (*Structured Query Language*)

- Linguagem comercial para BD relacional
 - padrão ISO desde a década de 80
 - SQL-86; SQL-89; SQL-92; ...; SQL:2023
 - não é apenas uma linguagem de consulta!
 - como o nome sugere...
- Base Formal
 - álgebra relacional e cálculo relacional
- Funcionalidades principais
 - definição de dados (DDL):
 - estrutura das tabelas
 - visões e autorizações de acesso
 - restrições de integridade (RIs)
 - transações
 - manipulação de dados (DML): Inclusão, Alteração, Exclusão, Consulta

SQL - DDL

- Comandos para definição de esquemas
 - **Create table**
 - define a estrutura da tabela, suas restrições de integridade e cria uma tabela vazia
 - **Alter table**
 - modifica a definição de uma tabela (inclusão / exclusão / alteração atributos; inclusão / exclusão RIs)
 - **Drop table**
 - remove uma tabela com todas as suas tuplas

SQL – *Create Table*

```
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}]  
)
```

- Principais tipos de dados
 - *int, smallint, tinyint, float, real,*
numeric(tamanho[,nro_casas_decimais]),
char(tamanho), varchar(tamanho), text, date,
time, datetime, ...

Exemplos de Criação de Tabela

```
CREATE TABLE Ambulatorios (  
    nroa                int,  
    andar               numeric(3) NOT NULL,  
    capacidade          smallint,  
    PRIMARY KEY (nroa)  
)
```

```
CREATE TABLE Medicos (  
    codm                int,  
    nome                varchar(40) NOT NULL,  
    idade               smallint NOT NULL,  
    especialidade       char(20),  
    CPF                 numeric(11) NOT NULL UNIQUE,  
    cidade              varchar(30),  
    nroa                int,  
    PRIMARY KEY (codm),  
    FOREIGN KEY (nroa) REFERENCES Ambulatorios  
)
```

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 def_RI_1
    [{, nome_RI_n def_RI_n}]
|
[ADD | DROP] [PRIMARY KEY ... | FOREIGN KEY ...]
```

Exemplos de Alteração de Tabelas

```
ALTER TABLE Ambulatorios  
    ADD COLUMN nome VARCHAR(30)
```

```
ALTER TABLE Medicos  
    DROP PRIMARY KEY
```

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

```
ALTER TABLE Funcionarios  
    ADD FOREIGN KEY(nroa) REFERENCES Ambulatorios  
(ou)
```

```
ALTER TABLE Funcionarios  
    ADD CONSTRAINT fk_nroa  
    FOREIGN KEY(nroa) REFERENCES Ambulatorios
```

SQL – Índices

- Definidos sobre atributos para acelerar consultas
- Chaves primárias são automaticamente indexadas
- Instruções:

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

```
DROP INDEX nome_indice ON nome_tabela
```

- Exemplos

```
CREATE INDEX indPac_CPF ON Pacientes(CPF)  
USING btree
```

```
DROP INDEX indPac_CPF ON Pacientes
```

Exercícios

- 1) Criar as seguintes tabelas, considerando que os atributos sublinhados fazem parte da chave primária e os em itálico são chaves estrangeiras (não acentuem as palavras!):
 - a) **Ambulatorios**: nroa (int), andar (numeric(2)) (não nulo), capacidade (smallint)
 - b) **Medicos**: codm (int), nome (varchar(40)) (não nulo), idade (smallint) (não nulo), cidade (varchar(40)), CPF (numeric(11)) (não nulo e único), especialidade (varchar(30)), *nroa* (int)
 - c) **Pacientes**: codp (int), nome (varchar(40)) (não nulo), idade (smallint) (não nulo), cidade (varchar(40)), CPF (numeric(11)) (não nulo e único), doença (varchar(40)) (não nulo)
 - d) **Funcionarios**: codf (int), nome (varchar(40)) (não nulo), idade (smallint) (não nulo), cidade (varchar(40)), CPF (numeric(11)) (não nulo e único), salário (numeric(10)), cargo (varchar(40))
 - e) **Consultas**: *codm* (int), *codp* (int), data (date), hora (time)
- 2) Alterar a tabela Funcionarios, removendo o atributo cargo
- 3) Criar um índice para o atributo cidade na tabela Pacientes