

BANCO DE DADOS

Linguagem SQL

**DDL - Linguagem de Definição de Dados –
Comandos Básicos**

SQL (*STRUCTURED QUERY LANGUAGE*)

Esquema para o modelo Relacional do contexto didático:

EMPRESA.

FUNCIONARIO (ident, nome, sobrenome, endereco, dtnasc, salario, sexo, supident, dnumero)

DEPENDENTE (fident, nome, dt_nasc, sexo, relacionamento)

DEPARTAMENTO (numero, nome, gident, dtinicio)

LOCALIZACOES (dnumero, localizacao)

PROJETO (numero, nome, localizacao, dnumero)

TRABALHA_EM (pnumero, fident, horas)

Algumas simplificações nos nomes de variáveis foram realizadas para facilitar o uso delas nas consultas. A chave primária, em cada relação, está sublinhada.

SQL (*STRUCTURED QUERY LANGUAGE*)

- Linguagem de banco de dados com recursos para **definição** de dados e **manipulação** de dados.
- Permite especificar restrições que devem ser impostas aos dados possibilitando a implementação da integridade e segurança da informação armazenada.

SQL

SQL é uma linguagem padrão para SGBDs comerciais. Há um padrão, estabelecido pelo *American National Standards Institute* (ANSI) e pela *International Standards Organization* (ISO).

Muito importante saber que as implementações de produtos comerciais **apresentam pequenas alterações** em relação ao SQL padrão.

SQL

Estudaremos nesta aula:

- Comando de definição de dados (DDL – *Data Definition Language*)
 - CREATE, ALTER, DROP
- Comandos de manipulação de dados (DML – *Data Manipulation Language*)
 - INSERT, UPDATE, DELETE

SQL

O comando **CREATE** permite criar objetos de banco de dados

- CREATE SCHEMA
- CREATE DOMAIN
- **CREATE TABLE**
- CREATE INDEX
- CREATE VIEW
- CREATE TRIGGER
- ...

SQL – CREATE TABLE

O comando CREATE TABLE é usado para especificar as tabelas do banco de dados, atribuindo-lhes um nome e listando seus atributos e restrições iniciais.

CREATE TABLE FUNCIONARIO

(
 atributos ...
 restrições ...
);

CREATE TABLE FUNCIONARIO

(

ident INT,

nome VARCHAR(50) NOT NULL,

sobrenome VARCHAR(50) NOT NULL,

endereco VARCHAR(200),

dtnasc DATE,

salario NUMERIC(10,2),

sexo CHAR,

supident INT,

dnumero INT NOT NULL,

PRIMARY KEY (ident)

);


```
CREATE TABLE FUNCIONARIO
```

Nome do objeto (nome da tabela)

```
(
```

Tipo do atributo (restrição de domínio)

```
ident
```

```
INT,
```

```
nome
```

```
VARCHAR(50)
```

```
NOT NULL,
```

```
sobrenome
```

```
VARCHAR(50)
```

```
NOT NULL,
```

```
endereco
```

```
VARCHAR(200),
```

```
dtnasc
```

```
DATE,
```

```
salario
```

```
NUMERIC(10,2),
```

```
sexo
```

```
CHAR,
```

```
supident
```

```
INT,
```

```
dnumero
```

```
INT
```

```
NOT NULL,
```

```
PRIMARY KEY (ident)
```

Restrição de chave

```
);
```

Restrição para valores nulos

CREATE TABLE FUNCIONARIO

(

ident INT,

nome VARCHAR(50) NOT NULL,

sobrenome VARCHAR(50) NOT NULL,

endereco VARCHAR(200),

dtnasc DATE,

salario NUMERIC(10,2),

sexo CHAR,

supident INT,

dnumero INT NOT NULL,

PRIMARY KEY (ident)

);

Chaves estrangeiras?

CREATE TABLE FUNCIONARIO

(

ident

TIPO,

....

....

....

....

....

....

supident

TIPO,

dnumero

TIPO

NOT NULL,

PRIMARY KEY (ident),

FOREIGN KEY (supident) **REFERENCES** FUNCIONARIO (ident),

FOREIGN KEY (dnumero) **REFERENCES** DEPARTAMENTO (numero)

);

CREATE TABLE FUNCIONARIO

(

ident

TIPO,

....

....

....

....

....

....

supident

TIPO,

dnumero

TIPO

NOT NULL,

PRIMARY KEY (ident),

FOREIGN KEY (supident) REFERENCES FUNCIONARIO (ident),

FOREIGN KEY (dnumero) REFERENCES DEPARTAMENTO (numero)

);

A tabela
DEPARTAMENTO já está
criada?

SQL – ALTER TABLE

O comando ALTER TABLE é usado para alterações nas definições dos objetos criados no banco de dados.

ALTER TABLE FUNCIONARIO
especificações de alterações ... ;

O comando ALTER pode ser usado com outros objetos do banco de dados.

```
CREATE TABLE FUNCIONARIO
```

```
(
```

```
....
```

```
....
```

```
....
```

```
dnumero
```

```
TIPO_ID
```

```
NOT NULL,
```

```
....
```

```
);
```

```
CREATE TABLE DEPARTAMENTO
```

```
(
```

```
numero
```

```
TIPO_ID
```

```
NOT NULL,
```

```
....
```

```
);
```

```
ALTER TABLE FUNCIONARIO
```

```
ADD CONSTRAINT FUNC_DEPTO FOREIGN KEY (dnumero)  
REFERENCES DEPARTAMENTO (numero);
```

SQL – OUTROS EXEMPLOS

ALTER TABLE FUNCIONARIO

DROP endereco **CASCADE**;

ou

ALTER TABLE FUNCIONARIO

DROP endereco **RESTRICT**;

ALTER TABLE FUNCIONARIO

ALTER supident **SET DEFAULT** 1163;

ALTER TABLE FUNCIONARIO

ADD ENDERECO VARCHAR(200);

SQL – DROP

O comando **DROP** é usado para alterações nas definições dos objetos criados no banco de dados.

DROP TABLE FUNCIONARIO **CASCADE**;

O comando DROP pode ser usado com outros objetos existentes no banco de dados.

SQL – INSERT

O comando **INSERT** permite inserir um registro em uma tabela.

INSERT INTO FUNCIONARIO

VALUES (1163, 'Claudia', 'Morais', 'Rua A – 1 –SP', '1973-02-25',
5558.00, 'F', 1, 4);

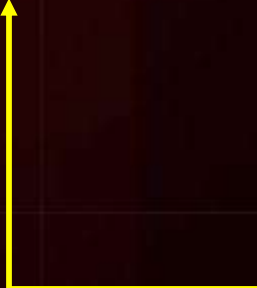
INSERT INTO FUNCIONARIO (ident, nome, sobrenome, dnumero)

VALUES (1164, 'Jorge', 'Vila Verde', 1);

SQL – INSERT BY SELECT

INSERT INTO DEPTS_INFO (DNOME, NO_FUNC, T_SALARIO)

SELECT ...
FROM ...



Uma consulta SQL para extração de informação existente em outras tabelas do banco de dados. Retorna um conjunto de registros e cada um deles será inserido na tabela DEPTS_INFO pelo comando INSERT INTO.

SQL - UPDATE

O comand UPDATE permite modificar os valores dos atributos de um ou mais registros de uma tabela.

UPDATE	PROJETO	novos valores
SET	localizacao = 'Minas Gerais', dnumero = 5	
WHERE	numero = 1;	

condição para indicar os registros que devem ter seus valores modificados, **importantíssimo** para filtrar **o que realmente deve ser alterado.**

SQL - UPDATE

```
UPDATE FUNCIONARIO
SET    salario = salario * 1.1
WHERE  dnumero IN ( SELECT ...
                    FROM   ...
                    WHERE  ... );
```

Uma consulta SQL que retorna um conjunto de registros.
A condição do comando UPDATE indica que *dnumero* deve pertencer ao conjunto de registros retornado.

SQL - DELETE

O comando DELETE remove registros de uma tabela.


DELETE FROM FUNCIONARIO;

DELETE FROM FUNCIONARIO

WHERE ident = 1167;

DELETE FROM FUNCIONARIO

WHERE nome = 'Caio';



Remove apenas
um registro, pois
ident é chave
primária.

SQL - DELETE

DELETE	FUNCIONARIO	
WHERE	dnumero	IN
		(SELECT ...
		FROM ...
		WHERE ...);



Uma consulta SQL que retorna um conjunto de registros.
A condição do comando DELETE indica que *dnumero* deve pertencer ao conjunto de registros retornado.

Estes slides estão baseados na bibliografia:

- Elmasri, Ramez; Navathe, Shamkant B. Sistemas de Banco de Dados. Pearson, 7a edição, 1128p., 2018.

BANCO DE DADOS

Linguagem SQL

**DDL - Linguagem de Definição de Dados –
Comandos Básicos**