

BANCO DE DADOS 2

SÉRGIO CARLOS PORTARI JÚNIOR

Sumário:

1) Trabalhando com Constraints:

- **NOT NULL**
- **UNIQUE**
- **PRIMARY KEY**
- **FOREIGN KEY**
- **DEFAULT**

2) Componentes do Select:

- **DISTINCT**
 - **WHERE**
 - **AND e OR**
 - **ORDER BY**
- 

2) Visão Geral

- A **SQL** é dividida em alguns aspectos:

.Data Manipulation Language (DML) – Linguagem de Manipulação de Dados, são as **consultas**, inserções, modificações e exclusões de tuplas (registros).

- SELECT: Consulta registros;
- INSERT: Adiciona linhas a uma tabela;
- UPDATE: Altera linhas;
- DELETE: Remove linhas;

2) Visão Geral

- A **SQL** é dividida em alguns aspectos:

.Data Definition Language (DDL) – Linguagem de Definição de Dados, dá suporte à criação, exclusão e modificação das tabelas e visões;

- CREATE: cria a estrutura;
- ALTER: modifica a estrutura;
- DROP: remove a estrutura;
- RENAME: modifica o nome;
- TRUNCATE: exclui todas as linhas de uma tabela;

2) Visão Geral

- O que é uma **Constraint**?

- Para Matemática é uma condição para que a solução de um problema seja satisfeito;
- Para Banco de Dados limita um tipo de dado para que ele seja persistido em uma tabela do Banco de Dados;
- Exemplos de **CONSTRAINTS**:
 - 1) **NOT NULL**: não pode ser nulo;
 - 2) **UNIQUE**: o valor deve ser único entre todos os registros;
 - 3) **PRIMARY KEY**: chave primária;
 - 4) **FOREIGN KEY**: chave estrangeira;
 - 5) **CHECK**: limitar valores;
 - 6) **DEFAULT**: valor padrão;

3) Formato de uma consulta Básica;

- Exemplo 1, trabalhando com a Constraint **NOT NULL**:

```
CREATE TABLE Pessoas  
(  
idPessoa int NOT NULL,  
sobreNome varchar(255) NOT NULL,  
primeiroNome varchar(255),  
endereco varchar(255),  
cidade varchar(255)  
)
```

3) Formato de uma consulta Básica;

- Exemplo 2, trabalhando com a Constraint **UNIQUE**:

```
CREATE TABLE Pessoas  
(  
idPessoa int NOT NULL,  
sobreNome varchar(255) NOT NULL,  
primeiroNome varchar(255),  
endereco varchar(255),  
cidade varchar(255),  
UNIQUE (idPessoa)  
)
```

Se já criou a tabela, utilize

```
ALTER TABLE Pessoas ADD CONSTRAINT uc_PessoaID UNIQUE  
(idPessoa);
```

3) Formato de uma consulta Básica;

- Exemplo 3, trabalhando com a Constraint **UNIQUE**:

```
CREATE TABLE Pessoas(  
idPessoa int NOT NULL,  
sobreNome varchar(255) NOT NULL,  
primeiroNome varchar(255),  
endereco varchar(255),  
cidade varchar(255),  
CONSTRAINT uc_PessoaID UNIQUE (idPessoa,sobreNome)  
)
```

```
ALTER TABLE Pessoas  
ADD CONSTRAINT uc_PessoaIDsN UNIQUE (idPessoa,sobreNome)  
ALTER TABLE Pessoas  
DROP INDEX uc_PessoaIDsN
```


3) Formato de uma consulta Básica;

- Exemplo 4, trabalhando com a Constraint **PRIMARY KEY**:

```
CREATE TABLE Pessoas
(
idPessoa int NOT NULL,
sobrenome varchar(255) NOT NULL,
primeiroNome varchar(255),
endereco varchar(255),
cidade varchar(255),
PRIMARY KEY (idPessoa)
)
```

3) Formato de uma consulta Básica;

- Exemplo 5, trabalhando com a Constraint **PRIMARY KEY**:

```
CREATE TABLE Pessoas(  
idPessoa int NOT NULL,  
sobreNome varchar(255) NOT NULL,  
primeiroNome varchar(255),  
endereco varchar(255),  
cidade varchar(255),  
CONSTRAINT pk_PessoaID PRIMARY KEY (idPessoa,sobreNome)  
)
```

```
ALTER TABLE Pessoas  
ADD CONSTRAINT pk_PessoaID PRIMARY KEY (idPessoa,sobreNome)  
ALTER TABLE Pessoas  
DROP PRIMARY KEY
```

3) Formato de uma consulta Básica;

- Exemplo 6, trabalhando com a Constraint **FOREIGN KEY**:

idPessoa	primeiroNome	segundoNome	endereço	cidade
1	John	Hansen	Timoteivn, 10	Sandess
2	Harry	Svendson	Stots, 32	Sandess
3	Ana	Pattersen	Prince, 120	Stavanger

idCompra	compraNumero	idPessoa
1	12314	3
2	12352	3
3	12342	2
4	12345	2
5	12358	3

3) Formato de uma consulta Básica;

- Exemplo 6, trabalhando com a Constraint **FOREIGN KEY**:

```
CREATE TABLE Compras
(
idCompra int NOT NULL,
compraNumero int NOT NULL,
idPessoa int,
PRIMARY KEY (idCompra),
CONSTRAINT fk_porOrdem FOREIGN KEY (idPessoa)
REFERENCES Pessoa(idPessoa)
)
```

```
ALTER TABLE Compras
ADD CONSTRAINT fk_PorOrdem FOREIGN KEY (idPessoa)
REFERENCES Pessoas(idPessoa)
```

3) Formato de uma consulta Básica;

- Exemplo 8, trabalhando com a Constraint **DEFAULT**:

```
CREATE TABLE Pessoas(  
idPessoa int NOT NULL,  
sobreNome varchar(255) NOT NULL,  
primeiroNome varchar(255),  
endereco varchar(255),  
cidade varchar(255) DEFAULT 'Sandness'  
)
```

```
ALTER TABLE Pessoas  
CHANGE cidade  
cidade varchar(255) DEFAULT 'Sandness'
```

2) Visão Geral

- Uso de **DISTINCT**:

- É utilizado para retornar um valor apenas, ou seja, não apresenta valores repetidos:

```
SELECT DISTINCT Cidade FROM Pessoas
```

- Resultado:

Sandness
Stavanger

2) Visão Geral

- Uso de **WHERE**:

- É utilizado para extrair resultados que seguem um

```
SELECT column_name(s)  
FROM table_name  
WHERE column_name operator value
```

```
SELECT * FROM Pessoas  
WHERE Cidade='Sandness'
```

Operadores:

=, <>, >, <, >=, <=,

BETWEEN (Entre)

LIKE (pesquisando um padrão específico, utilizando o '%')

IN (dentro de uma lista)

2) Visão Geral

- Uso de **AND** e **OR**:
 - Operadores lógicos para validação de instrução:

```
SELECT * FROM Pessoa  
WHERE primeironome='Harry'  
AND sobrenome='Svendson'
```

- Uso de **ORDER BY**:
 - instrução de ordenação dos registros;

```
SELECT * FROM Pessoa  
ORDER BY sobrenome
```


Exercícios:

- 1) Crie o código SQL das tabelas utilizadas nesta apresentação, inclusive os Inserts

- 2) Utilizando as tabelas dadas como exemplo nesta apresentação crie as seguintes sintaxes:
 - a) SQL para buscar somente as pessoas que fizeram compras;
 - b) SQL para listar as pessoas que tem o primeironome começados por "A";
 - c) As pessoas que fizeram compras e que moram em "Sandess" ou o primeiro nome termina com "A";
 - d) Exiba todas as compras ordenadas pelo nome do comprador;

Referências Bibliográficas:

Sistemas de Gerenciamento de Bancos de Dados - 3^aed. -
R.Ramakrishnan, J.Gehrke - McGraw-Hill