

ADMINISTRAÇÃO DE BANCO DE DADOS

Linguagem SQL

Prof. Ms. Helton Souza Lima



Linguagem SQL - Introdução

- O modelo relacional encontra-se padronizado pela indústria de informática.
 - Ele é chamado de padrão SQL (Structured Query Language).
- O padrão SQL define precisamente uma interface SQL para a definição de tabelas, para as operações sobre as mesmas (seleção, projeção, junção, e outras) e para a definição de regras de integridade de bancos de dados.
- A interface SQL é, portanto, implementada em todos os sistemas de bancos de dados relacionais existentes.
- Por quê a indústria tem interesse em padronizar os sistemas de bancos de dados? A razão é muito simples: a existência de padrões facilita a interoperabilidade (comunicação entre máquinas, entre programas).

Linguagem SQL - Introdução

- **SQL** (Structured Query Language): desenvolvida pela IBM (70) como parte do sistema System R. A SQL foi inicialmente chamada de SEQUEL
- É a linguagem de consulta padrão para os SGBDR's
- Já existem padrões propostos: ANSI-SQL(SQL-89), SQL-92 e padrões mais recentes: SQL:1999 e SQL:2003.
- Linguagem de Definição de Dados (**DDL**): fornece comandos para definições de esquemas de relação, criação/remoção de tabelas, criação de índices e modificação de esquemas.
- Linguagem de Manipulação de Dados (**DML**): inclui uma linguagem de consulta baseada na álgebra relacional e cálculo relacional de tupla. Compreende comandos para inserir, consultar, remover e modificar tuplas num BD.
- Linguagem de Controle de acesso e segurança (**DCL**)

Data Definition Language - DDL

- Utilizada para a criação do esquema do banco de dados
- As principais ações desta linguagem são:
 - CREATE TABLE
 - ALTER TABLE
 - DROP TABLE
- Manipulação de índices
 - CREATE INDEX, ALTER INDEX, DROP INDEX

Data Definition Language - DDL

```
CREATE TABLE table_name
```

```
(
```

```
column_name1 data_type(size),
```

```
column_name2 data_type(size),
```

```
...
```

```
);
```

http://www.w3schools.com/sql/sql_datatypes.asp

MySQL Data Types (Version 8.0)

In MySQL there are three main data types: string, numeric, and date and time.

String Data Types

Data type	Description
CHAR(size)	A FIXED length string (can contain letters, numbers, and special characters). The <i>size</i> parameter specifies the column length in characters - can be from 0 to 255. Default is 1
VARCHAR(size)	A VARIABLE length string (can contain letters, numbers, and special characters). The <i>size</i> parameter specifies the maximum string length in characters - can be from 0 to 65535
BINARY(size)	Equal to CHAR(), but stores binary byte strings. The <i>size</i> parameter specifies the column length in bytes. Default is 1
VARBINARY(size)	Equal to VARCHAR(), but stores binary byte strings. The <i>size</i> parameter specifies the maximum column length in bytes.
TINYBLOB	For BLOBs (Binary Large Objects). Max length: 255 bytes
TINYTEXT	Holds a string with a maximum length of 255 characters
TEXT(size)	Holds a string with a maximum length of 65,535 bytes
BLOB(size)	For BLOBs (Binary Large Objects). Holds up to 65,535 bytes of data
MEDIUMTEXT	Holds a string with a maximum length of 16,777,215 characters
MEDIUMBLOB	For BLOBs (Binary Large Objects). Holds up to 16,777,215 bytes of data
LONGTEXT	Holds a string with a maximum length of 4,294,967,295 characters
LONGBLOB	For BLOBs (Binary Large Objects). Holds up to 4,294,967,295 bytes of data

Data Definition Language - DDL

- As colunas podem ter as seguintes restrições
 - NOT NULL
 - UNIQUE
 - DEFAULT

Data Definition Language - DDL

- Restrições de integridade e de domínio
 - PRIMARY Key (colunas)
 - FOREIGN KEY (colunas) REFERENCES tabela_base (colunas)
 - CHECK(condição)

Criando uma tabela

```
CREATE TABLE nome_tabela
```

```
(
```

```
nome_coluna1 data_type(tamanho) not null,
```

```
nome_coluna2 data_type(tamanho),
```

```
...
```

```
CONSTRAINT pk_sigla1 PRIMARY KEY (nome_coluna1),
```

```
CONSTRAINT fk_sigla2 FOREIGN KEY (column_name2) REFERENCES
```

```
nome_tabela02(nome_coluna_outra_tabela)
```

```
);
```

Criando uma tabela

CREATE TABLE EMPREGADOS

```
(  
  matricula number(10) not null,  
  nome varchar2(50) not null,  
  endereco varchar2(250),  
  funcao varchar2(50),  
  salario number(10,2),  
  constraint pk_emp primary key (matricula)  
);
```

EMPREGADOS				
<u>Matrícula</u>	Nome	Endereço	Função	Salário
100	Ana	R. Pedro I, 12, A. Branco	Secretária	500,00
250	Pedro	R. J. Silva, 24, Liberdade	Engenheiro	1500,00
108	André	R. Itália, 33, B. Nações	Técnico	950,00
210	Paulo	R. Pará, 98, B. Estados	Engenheiro	1810,00
105	Sônia	R. Oliveira, 76, A. Branco	Engenheiro	2500,00

Criando uma tabela

```
CREATE TABLE DEPARTAMENTOS
```

```
(
```

```
codigo varchar2(10) not null,
```

```
nome varchar2(50) not null,
```

```
constraint pk_depto primary key (codigo)
```

```
);
```

DEPARTAMENTOS	
<u>Código</u>	Nome
D1	Compras
D2	Médico
D3	Educação

Criando uma tabela

```
CREATE TABLE ATUACOES
```

```
(
```

```
  empregado number(10) not null,
```

```
  departamento varchar2(10) not null,
```

```
  funcao varchar2(30),
```

```
  constraint fk_emp foreign key (empregado) references empregados(matricula),
```

```
  constraint fk_depto foreign key (departamento) references departamentos(codigo)
```

```
);
```

ATUACOES		
Empregado	Departamento	Função
100	D1	Secretária
250	D1	Engenheiro
108	D2	Técnico
210	D2	Engenheiro
105	D1	Engenheiro

Descrição de uma tabela

DESC EMPREGADOS ;

TABLE EMPREGADOS

Column	Null?	Type
MATRICULA	NOT NULL	NUMBER(10,0)
NOME	NOT NULL	VARCHAR2(50)
ENDERECO	-	VARCHAR2(250)
FUNCAO	-	VARCHAR2(50)
SALARIO	-	NUMBER(10,2)

Criando uma tabela – prática

Criar as tabelas em um banco de dados relacional utilizando a linguagem SQL para as tabelas do sistema de Seguros

Pode ser:

- Banco de dados instalado na máquina
- Banco de dados na nuvem
- <https://livesql.oracle.com/>

Criando uma tabela - prática

Ferramentas gráficas

Create Table ✕

* Table Name Add a Trigger

Column Name	Data Type	Length	Nullable?	Primary Key?	Unique?
<input type="text" value="Column Name"/>	<input type="text" value="VARCHAR2"/>	<input type="text" value="Length"/>	<input type="text" value="Yes"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="text" value="Column Name"/>	<input type="text" value="VARCHAR2"/>	<input type="text" value="Length"/>	<input type="text" value="Yes"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="text" value="Column Name"/>	<input type="text" value="VARCHAR2"/>	<input type="text" value="Length"/>	<input type="text" value="Yes"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="text" value="Column Name"/>	<input type="text" value="VARCHAR2"/>	<input type="text" value="Length"/>	<input type="text" value="Yes"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="text" value="Column Name"/>	<input type="text" value="VARCHAR2"/>	<input type="text" value="Length"/>	<input type="text" value="Yes"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="text" value="Column Name"/>	<input type="text" value="VARCHAR2"/>	<input type="text" value="Length"/>	<input type="text" value="Yes"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="text" value="Column Name"/>	<input type="text" value="VARCHAR2"/>	<input type="text" value="Length"/>	<input type="text" value="Yes"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="text" value="Column Name"/>	<input type="text" value="VARCHAR2"/>	<input type="text" value="Length"/>	<input type="text" value="Yes"/>	<input type="checkbox"/>	<input type="checkbox"/>

[Add More Columns](#)

Alterando uma tabela – adição de coluna

```
ALTER TABLE ATUACOES
```

```
ADD
```

```
(
```

```
data_inicio date,
```

```
data_fim date,
```

```
temporario number(10)
```

```
);
```

ATUACOES					
<u>Empregado</u>	<u>Departamento</u>	Função	data_inicio	data_fim	temporario
100	D1	Secretária	01/01/2020		
250	D1	Engenheiro	01/01/2020	01/02/2021	
108	D2	Técnico	01/01/2022		
210	D2	Engenheiro	01/01/2023		
250	D3	Engenheiro	01/02/2021		

Alterando uma tabela – alterando coluna

ALTER TABLE ATUACOES

MODIFY

(
temporario **varchar2(50) DEFAULT 'teste' NOT NULL**
);

ATUACOES					
<u>Empregado</u>	<u>Departamento</u>	Função	Data início	Data fim	Temporario
100	D1	Secretária	01/01/2020		
250	D1	Engenheiro	01/01/2020	01/02/2021	
108	D2	Técnico	01/01/2022		
210	D2	Engenheiro	01/01/2023		
250	D3	Engenheiro	01/02/2021		

Alterando uma tabela – removendo coluna

ALTER TABLE ATUACOES

DROP COLUMN

temporario

;

ATUACOES					
<u>Empregado</u>	<u>Departamento</u>	Função	Data início	Data fim	Temporario
100	D1	Secretária	01/01/2020		
250	D1	Engenheiro	01/01/2020	01/02/2021	
108	D2	Técnico	01/01/2022		
210	D2	Engenheiro	01/01/2023		
250	D3	Engenheiro	01/02/2021		

Alterando uma tabela – alterando nome de coluna

ALTER TABLE ATUACOES

RENAME COLUMN

data_inicio TO data_admissao

;

ATUACOES				
<u>Empregado</u>	<u>Departamento</u>	Função	Data admissao	Data fim
100	D1	Secretária	01/01/2020	
250	D1	Engenheiro	01/01/2020	01/02/2021
108	D2	Técnico	01/01/2022	
210	D2	Engenheiro	01/01/2023	
250	D3	Engenheiro	01/02/2021	

Alterando uma tabela – adicionando restrição

ALTER TABLE ATUACOES

MODIFY

data_admissao **DEFAULT** SYSDATE **NOT NULL**

;

ATUACOES				
<u>Empregado</u>	<u>Departamento</u>	Função	Data admissao	Data fim
100	D1	Secretária	01/01/2020	
250	D1	Engenheiro	01/01/2020	01/02/2021
108	D2	Técnico	01/01/2022	
210	D2	Engenheiro	01/01/2023	
250	D3	Engenheiro	01/02/2021	

Apagando uma tabela

```
DROP TABLE ATUACOES
```

```
;
```

FOREIGN KEY – Regras de remoção/atualização

Uma cláusula **FOREIGN KEY** inclui regras de remoção/atualização:

FOREIGN KEY (coluna) REFERENCES tabela

[ON DELETE {RESTRICT|CASCADE|SET NULL| SET DEFAULT}]

[ON UPDATE {RESTRICT|CASCADE|SET NULL| SET DEFAULT}]

FOREIGN KEY – Regras de remoção/atualização

Supondo que T2 tem uma chave estrangeira para T1, vejamos as cláusulas ON DELETE e ON UPDATE

T2					
<u>Matrícula</u>	Nome	Endereço	Função	Salário	Departamento
100	Ana	R. Pedro I, 12, A. Branco	Secretária	500,00	D1
250	Pedro	R. J. Silva, 24, Liberdade	Engenheiro	1500,00	D1
108	André	R. Itália, 33, B. Nações	Técnico	950,00	D2
210	Paulo	R. Pará, 98, B. Estados	Engenheiro	1810,00	D2
105	Sônia	R. Oliveira, 76, A. Branco	Engenheiro	2500,00	D1

T1	
<u>Código</u>	Nome
D1	Compras
D2	Médico
D3	Educação

FOREIGN KEY – Regras de remoção/atualização

ON DELETE:

RESTRICT: (default) significa que uma tentativa de se remover uma linha de T1 falhará se alguma linha em T2 combina com a chave

CASCADE: remoção de uma linha de T1 implica em remoção de todas as linhas de T2 que combina com a chave de T1

SET NULL: remoção de T1 implica em colocar NULL em todos os atributos da chave estrangeira de cada linha de T2 que combina.

SET DEFAULT: remoção de linha em T1 implica em colocar valores DEFAULT nos atributos da chave estrangeira de cada linha de T2 que combina.

FOREIGN KEY – Regras de remoção/atualização

ON UPDATE:

RESTRICT: (default) update de um atributo de T1 falha se existem linhas em T2 combinando

CASCADE: update de atributo em T1 implica que linhas que combinam em T2 também serão atualizadas

SET NULL: update de T1 implica que valores da chave estrangeira em T2 nas linhas que combinam são postos par NULL.

SET DEFAULT: update de T1 implica que valores da chave estrangeira de T2 nas linhas que combinam terão valores default aplicados.

Alterando uma tabela - prática

Altere as tabelas incluindo uma coluna e depois removendo-a, utilizando a linguagem SQL para as tabelas do exercício anterior

Data Manipulation Language - DML

- Utilizada para a manipulação de dados dentro de uma estrutura definida
- As principais ações de manipulação são:
 - **INSERT**
 - **UPDATE**
 - **DELETE**
 - **SELECT**

Comando INSERT

- Usado para adicionar uma tupla (linha) em uma relação (tabela)
- Sintaxe 01
 - INSERT INTO tabela (lista colunas) VALUES (lista valores)
 - Insere apenas 1 registro
- Sintaxe 02
 - INSERT INTO tabela (lista colunas) **SELECT** ...
 - Insere vários registros resultado da consulta do SELECT

Comando INSERT

- **INSERT INTO** EMPREGADOS (matricula, nome, endereco, funcao, salario) **VALUES** (100, 'Ana','R. Pedro I, 12, A. Branco','Secretária', 500)

EMPREGADOS				
<u>Matrícula</u>	Nome	Endereço	Função	Salário
100	Ana	R. Pedro I, 12, A. Branco	Secretária	500,00

Comando INSERT

- INSERT INTO EMPREGADOS(matricula, nome, endereco, funcao, salario) values (250, 'Pedro','R. J. Silva, 24, Liberdade','Engenheiro', 1500);
- INSERT INTO EMPREGADOS(matricula, nome, endereco, funcao, salario) values (108, 'André','R. Itália, 33, B. Nações','Técnico', 950);
- INSERT INTO EMPREGADOS(matricula, nome, endereco, funcao, salario) values (210, 'Paulo','R. Pará, 98, B. Estados','Engenheiro', 1810);
- INSERT INTO EMPREGADOS(matricula, nome, endereco, funcao, salario) values (105, 'Sônia','R. Oliveira, 76, A. Branco','Engenheiro', 2500);

EMPREGADOS				
Matrícula	Nome	Endereço	Função	Salário
100	Ana	R. Pedro I, 12, A. Branco	Secretária	500,00
250	Pedro	R. J. Silva, 24, Liberdade	Engenheiro	1500,00
108	André	R. Itália, 33, B. Nações	Técnico	950,00
210	Paulo	R. Pará, 98, B. Estados	Engenheiro	1810,00
105	Sônia	R. Oliveira, 76, A. Branco	Engenheiro	2500,00

Comando INSERT

- Os valores devem estar na mesma ordem que as colunas
- As colunas que não forem definidas, serão gravadas como NULL no banco
- Se as colunas não forem informadas, então subentende-se que serão todas as colunas, na ordem que estão gravadas no banco
- Pensar bem sobre decisões de como gravar números que correspondem a documentos (exemplo: CPF)
- NULL é diferente de zero
- NULL é diferente de ""
- Não se pode realizar comparações com NULL
 - Apenas verificar se é ou não NULL
- Usar o valor DEFAULT como alternativa ao NULL

Comando UPDATE

- Modifica o valor de atributos de **um ou mais** registros (tuplas)
- **UPDATE** tabela
- **SET** coluna01=valor, coluna02=valor02,...
- [**WHERE** condição]
- Todos os registros que estiverem dentro dos critérios da **condição**, serão atualizados
- Omitir a cláusula WHERE implica que o UPDATE deve ser aplicado a todos os registros da tabela
- Para atualizar apenas 1 registro, sugere-se utilizar o valor de uma chave primária na condição

Comando UPDATE

- UPDATE EMPREGADOS SET salario=2500 WHERE funcao='Engenheiro'

EMPREGADOS				
<u>Matrícula</u>	Nome	Endereço	Função	Salário
100	Ana	R. Pedro I, 12, A. Branco	Secretária	500,00
250	Pedro	R. J. Silva, 24, Liberdade	Engenheiro	1500,00
108	André	R. Itália, 33, B. Nações	Técnico	950,00
210	Paulo	R. Pará, 98, B. Estados	Engenheiro	1810,00
105	Sônia	R. Oliveira, 76, A. Branco	Engenheiro	2500,00

Comando DELETE

- Apaga o valor de atributos de **um ou mais** registros (tuplas)
- **DELETE FROM** tabela
- [**WHERE** condição]
- Todos os registros que estiverem dentro dos critérios da **condição**, serão removidos
- Omitir a cláusula WHERE implica que o DELETE deve ser aplicado a todos os registros da tabela
- Para atualizar apenas 1 registro, sugere-se utilizar o valor de uma chave primária na condição

Comando DELETE

- DELETE FROM EMPREGADOS WHERE matricula=105

EMPREGADOS				
<u>Matrícula</u>	Nome	Endereço	Função	Salário
100	Ana	R. Pedro I, 12, A. Branco	Secretária	500,00
250	Pedro	R. J. Silva, 24, Liberdade	Engenheiro	2500,00
108	André	R. Itália, 33, B. Nações	Técnico	950,00
210	Paulo	R. Pará, 98, B. Estados	Engenheiro	2500,00
105	Sônia	R. Oliveira, 76, A. Branco	Engenheiro	2500,00

Manipulando dados - prática

- Crie script SQL para
 - Inserir dados nas tabelas do sistema de Seguro de carros
 - 5 clientes
 - 5 carros
 - 3 acidentes
 - Atualizar uma linha de uma das tabelas
 - Deletar uma linha também