



Fundação Educacional Comunitária
de São Sebastião do Paraíso



Sistemas de
Informação

SQL – Linguagem de Consulta Estruturada – Introdução

Disciplina: Bando de Dados I

Prof. Me. Fernando Roberto Proença

SQL – Definição

2

- A SQL (*structured query language*) – **Linguagem de Consulta Estruturada**, é uma linguagem padrão oferecida pelos SGBDs relacionais para manipular e requisitar informações do Banco de Dados.
- Foi criada originariamente pela IBM em meados dos anos 70, mas adotada pelos Bancos de Dados Relacionais.

Alguns SGBDs que utilizam SQL




Microsoft®
SQL Server®



PostgreSQL



SQL – Versões

Ano	Versão	Apelido	Comentários
1986	SQL-86	SQL-87	Publicada pela ANSI. Ratificada pela ISO em 1987.
1989	SQL-89		Pequenas alterações
1992	SQL-92	SQL2	Incluídas: integridade referencial, valores NULL, DEFAULT, entre outros. Versão ainda muito usada por SGBDs.
1999	SQL:1999	SQL3	Novos tipos de dados, consultas recursivas, gatilhos e procedimentos armazenados, etc.
2003	SQL:2003		Recursos para XML, Atributos com valores gerados automaticamente, etc.

SQL – Tipos de dados – Inteiros

5

Tipo	Bytes	Limites	Descrição
TINYINT	1 byte	-128 a 127	Número inteiro
SMALLINT	2 bytes	-32.768 a 32.767	Número inteiro
MEDIUMINT	3 bytes	-8.388.608 a 8.388.607	Número inteiro
INT (ou INTEGER)	4 bytes	-2.147.483.648 a 2.147.483.647	Número inteiro
BIGINT	8 bytes	-922.372.036.854.775.808 a 922.372.036.854.775.808	Número inteiro

SQL – Tipos de dados – Ponto Flutuante

6

Tipo	Bytes	Limites	Descrição
FLOAT	4 bytes	1.175×10^{-38} a 3.402×10^{38}	Número real com precisão de 7 dígitos
DOUBLE	8 bytes	1.175×10^{-308} a 3.402×10^{306}	Número real com precisão de 15 dígitos
DECIMAL(n,m)	n bytes	2.225×10^{-308} a 1.797×10^{306}	Número com precisão definida. N = quantidade de caracteres e M = casas decimais
NUMERIC(n,m)	n bytes	2.225×10^{-308} a 1.797×10^{306}	Número com precisão definida. N = quantidade de caracteres e M = casas decimais

SQL – Tipos de dados – Caracteres

7

Tipo	Bytes	Limites	Descrição
CHAR(n)	n bytes	0 a 255 caracteres	Tamanho fixo de Caracteres
VARCHAR(n)	n bytes	0 a 255 caracteres	Tamanho variável de Caracteres
TINYTEXT	Longitude + 1 byte	0 a 255 caracteres	Texto
TEXT	Longitude + 2 bytes	0 a 65.535 caracteres	Texto
MEDIUMTEXT	Longitude + 3 bytes	0 a 16.777.215 caracteres	Texto
LONGTEXT	Longitude + 4 bytes	0 a 4.294.967.297 caracteres	Texto

SQL – Tipos de dados – Data e Hora

8

Tipo	Bytes	Limites	Descrição
DATE	3 bytes	01/01/1000 a 31/12/9999	Data no formato YYYY-MM-DD
DATETIME	8 bytes	01/01/1000 00:00:00 a 31/12/9999 23:59:59	Data e hora no formato YYYY-MM-DD HH:MM:SS
TIME	3 bytes	00:00:00 a 23:59:59	Hora no formato HH:MM:SS
TIMESTAMP	4 bytes	01/01/1000 00:00:00 a 31/12/9999 23:59:59	Data e hora atual no formato YYYY-MM-DD HH:MM:SS
YEAR	1 byte	1901 a 2155	Ano

SQL – Tipos de dados – Binário

9

Tipo	Bytes	Limites	Descrição
TINYBLOB	Longitude + 1 byte	0 a 255 bytes	Binário
BLOB	Longitude+ 2 bytes	0 a 65.535 bytes	Binário
MEDIUMBLOB	Longitude + 3 bytes	0 a 16.777.215 bytes	Binário
LOB	Longitude + 4 bytes	0 a 4.294.967.297 bytes	Binário

SQL – Restrições de chave e integridade

10

Índices	Descrição
UNSIGNED	Sem sinal (somente números positivos, dobra o limite dos números).
NOT NULL	O valor de um registro de uma coluna da tabela é obrigatório (não pode ser nulo).
AUTO_INCREMENT (AI)	O valor do registro de uma coluna da tabela será gerado, incrementado e armazenado automaticamente.
UNIQUE	O valor do registro de uma coluna tem que ser único.
PRIMARY KEY (PK)	Chave primária: O valor do registro da(s) coluna(s) que compõe(m) a chave primária deve ser único e não pode ser nulos.
FOREIGN KEY (FK)	Chave estrangeira: relacionamento entre tabelas de dados do banco de dados.

SQL – Definição

11

- A SQL é composta por “sub-linguagens”:
 - ▣ DDL (*Data Definition Language*) – Linguagem de Definição de Dados
 - ▣ DML (*Data Manipulation Language*) – Linguagem de Manipulação de Dados
 - DQL (*Data Query Language*) – Linguagem de Consulta de Dados
 - ▣ DCL (*Data Control Language*) – Linguagem de Controle de Dados

SQL – DDL (Linguagem de Definição de Dados)

12

- Basicamente são os comandos/instruções de criação, alteração, remoção de:
 - ▣ Banco de Dados e Esquemas
 - ▣ Tabelas, atributos e seus tipos
 - ▣ Chaves,
 - ▣ Índices,
 - ▣ Restrições,
 - ▣ Procedimentos Armazenados,
 - ▣ Gatilhos...

SQL – Definição do Banco de Dados

13

- Exibição, criação, uso e exclusão de banco de dados

- ▣ Mostra Bancos existentes na conexão

```
SHOW DATABASES;
```

- ▣ Cria um novo banco de dados

```
CREATE DATABASE [nome_banco];
```

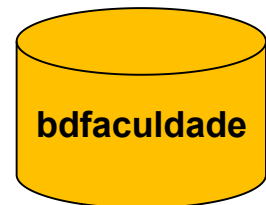
- ▣ Coloca um banco de dados em utilização

```
USE [nome_banco];
```

- ▣ Exclui um banco de dados

```
DROP DATABASE [nome_banco];
```

*Exemplo de
Banco de Dados*



SQL – DDL – Exemplo

14

- A partir do Dicionário de Dados seguinte crie um novo banco de dados chamado "**bdfaculdade**"

- ▣ Cria um novo banco de dados

```
CREATE DATABASE bdfaculdade;
```

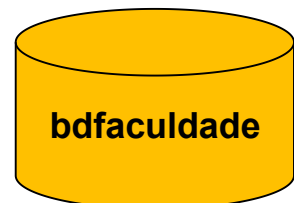
- ▣ Coloca um banco de dados em utilização

```
USE bdfaculdade;
```

- ▣ Exclui um banco de dados

```
DROP DATABASE bdfaculdade;
```

*Exemplo de
Banco de Dados*



SQL – Definição da Tabela – CREATE TABLE

15

- Criação de tabelas (relações) – Sintaxe básica

```
CREATE TABLE nome_tabela (
  coluna1 <TIPO> [NOT NULL] [DEFAULT 'VALOR'],
  coluna2 <TIPO> [NOT NULL] [DEFAULT 'VALOR'],
  ...
  PRIMARY KEY (coluna1),

  CONSTRAINT nome_chave_estr FOREIGN KEY (coluna2)
  REFERENCES outratabela(chave_primaria_outra_tabela)
);
```

SQL – Definição da Tabela – CREATE TABLE

16

- Criação de tabelas (relações) – Sintaxe básica

```
CREATE TABLE nome_tabela (
  coluna1 <TIPO> [NOT NULL] [DEFAULT 'VALOR'],
  coluna2 <TIPO> [NOT NULL] [DEFAULT 'VALOR'],
  ...
  PRIMARY KEY (coluna1),

  CONSTRAINT nome_chave_estr FOREIGN KEY (coluna2)
  REFERENCES outratabela(chave_primaria_outra_tabela)
);
```

Entre <> configuração obrigatória

Entre [] configuração opcional

SQL – Parâmetros do CREATE TABLE

17

- ❑ **NOT NULL** – Atributo não pode ser nulo em nenhuma linha
- ❑ **DEFAULT 'VALOR'** – Caso não seja preenchida, a coluna assumirá o valor definido como padrão
- ❑ **PRIMARY KEY (coluna1)** – Escolhe atributo como chave primária
- ❑ **FOREIGN KEY (coluna1) REFERENCES outra_tabela(coluna2)**
Escolhe "coluna1" como chave estrangeira que faz referência à "coluna2" da tabela "outra_tabela".

SQL – DDL – Exemplo

18

- ❑ Dicionário de Dados da tabela "**Pessoas**"

Nome da Tabela: Pessoas					
Definição: Armazena dados das pessoas.					
Nome da Coluna	Tipo de Dados	Tamanho	Restrições	Valor Padrão	Descrição
cpf	Caracteres	11 bytes	PK	N/D	CPF do aluno, chave primária
nome	Caracteres	50 bytes	NOT NULL	N/D	Nome completo
datanasc	Data	10 bytes	NOT NULL	N/D	Data de nascimento
telefone	Caracteres	20 bytes		N/D	Telefone
sexo	Caractere	1 byte		N/D	[M–Masc. F–Fem. O–Outro]

SQL – CREATE TABLE – Exemplo

19

- Comandos SQL para criar a tabela "**Pessoas**" no BD – Exemplo

```
CREATE TABLE pessoas (
    cpf          VARCHAR(11)    NOT NULL,
    nome         VARCHAR(50)    NOT NULL,
    datanasc     DATE           NOT NULL,
    telefone     VARCHAR(20),
    sexo         CHAR(1),
    PRIMARY KEY (cpf)
);
```

Pessoas

#cpf
Nome
DataNasc
Telefone
Sexo

SQL – DDL – Exemplo

20

- Dicionário de Dados da tabela "**Cursos**"

Nome da Tabela: Cursos					
Definição: Armazena os dados dos cursos.					
Nome da Coluna	Tipo de Dados	Tamanho	Restrições	Valor Padrão	Descrição
<u>codcurso</u>	Inteiro	4 bytes	PK, AI	N/D	Código do curso, chave primária, gerada automaticamente
nome	Caracteres	50 bytes	NOT NULL	N/D	Nome do curso
datacriacao	Data	10 bytes	NOT NULL	N/D	Data de criação do curso
duracao	Inteiro	4 bytes		4	Tempo de duração em anos
valor	Decimal	8 bytes		N/D	Valor da mensalidade

SQL – CREATE TABLE – Exemplo

21

Comandos SQL para criar a tabela "**Cursos**" no BD – Exemplo

```
CREATE TABLE cursos (
  codcurso      INT          AUTO_INCREMENT,
  nome          VARCHAR(50)  NOT NULL,
  datacriacao   DATE        NOT NULL,
  duracao       INT          DEFAULT 4,
  valor         FLOAT,
  PRIMARY KEY (codcurso)
);
```

Cursos

#CodCurso
Nome
DataCriacao
Duracao
Valor

SQL – DDL – Exemplo

22

- Dicionário de Dados da tabela "**Disciplinas**"

Nome da Tabela: Disciplinas					
Definição: Armazena os dados das disciplinas dos cursos.					
Nome da Coluna	Tipo de Dados	Tamanho	Restrições	Valor Padrão	Descrição
<u>coddisc</u>	Inteiro	4 bytes	PK, AI	N/D	Código da disciplina, chave primária, gerada automaticamente
nome	Caracteres	50 bytes	NOT NULL	N/D	Nome da disciplina
horas	Inteiro	4 bytes		80	Quantidade de horas da disciplina
curso	Inteiro	4 bytes	NN, FK		Código do curso, chave estrangeira para referenciar a tabela "Cursos"

SQL – CREATE TABLE – Exemplo

23

Comandos SQL para criar a tabela "**Disciplinas**" no BD – Exemplo

```
CREATE TABLE disciplinas (
  coddisc      INT          AUTO_INCREMENT,
  nome         VARCHAR(50)  NOT NULL,
  horas       INT          DEFAULT 80,
  curso        INT          NOT NULL,
  PRIMARY KEY (coddisc),
  CONSTRAINT FK_disciplinas_cursos FOREIGN KEY (curso)
    REFERENCES cursos(codcurso)
);
```

Disciplinas

#CodDisc
Nome
Horas
Curso (FK)

SQL – RENAME TABLE – Exemplo

24

- Alteração do nome da tabela – Sintaxe Básica

```
RENAME TABLE nome_tabela TO novo_nome_tabela;
```

- Alteração do nome da tabela – Exemplo

```
RENAME TABLE pessoas TO alunos;
```

Alunos

#cpf
Nome
DataNasc
Telefone
Sexo

SQL – ALTER TABLE – Exemplo

25

- Alteração de colunas da tabela – Sintaxe Básica

```
ALTER TABLE nome_tabela
  ADD coluna <TIPO> [NOT NULL] [DEFAULT 'VALOR'],
  DROP coluna,

  ADD CONSTRAINT nome_chave FOREIGN KEY (chave_estr)
    REFERENCES outra_tabela(atributo_outra_tabela),
  DROP FOREIGN KEY nome_chave,

  ADD PRIMARY KEY (coluna),
  DROP PRIMARY KEY;
```

SQL – ALTER TABLE – Exemplo

26

- Alteração de colunas da tabela – Sintaxe Básica

```
ALTER TABLE nome_tabela
  ADD coluna <TIPO> [NOT NULL] [DEFAULT 'VALOR'],
  DROP coluna,

  ADD CONSTRAINT nome_chave FOREIGN KEY (chave_estr)
    REFERENCES outra_tabela(atributo_outra_tabela),
  DROP FOREIGN KEY nome_chave,

  ADD PRIMARY KEY (coluna),
  DROP PRIMARY KEY;
```

Adiciona uma nova coluna

Exclui uma coluna existente

Adiciona uma relação entre duas tabelas (Chave estrangeira)

Exclui relação entre duas tabelas

Adiciona e remove uma chave primária

Última alteração encerra o comando com ponto-e-vírgula

SQL – ALTER TABLE – Exemplo

27

- Alteração de colunas da tabela – Exemplo

ALTER TABLE disciplinas

CHANGE horas **TO** ha **INT NOT NULL,**

ADD semestre **INT NOT NULL,**

DROP semestre,

ADD CONSTRAINT FOREIGN KEY (curso)

REFERENCES cursos (codcurso) ;

Disciplinas

#CodDisc

Nome

Horas

Curso (FK)

SQL – ALTER TABLE – Exemplo

28

- Alteração de colunas da tabela – Exemplo

ALTER TABLE disciplinas

CHANGE horas **TO** ha **INT NOT NULL,**

ADD semestre **INT NOT NULL,**

DROP semestre,

ADD CONSTRAINT FOREIGN KEY (curso)

REFERENCES cursos (codcurso) ;

Disciplinas

#CodDisc

Nome

ha

Curso (FK)

Semestre

SQL – ALTER TABLE – Exemplo

29

- Alteração de colunas da tabela – Exemplo

ALTER TABLE disciplinas

CHANGE horas **TO** ha **INT NOT NULL,**

ADD semestre **INT NOT NULL,**

DROP semestre,

ADD CONSTRAINT FOREIGN KEY (curso)

REFERENCES cursos (codcurso) ;

Disciplinas

#CodDisc
Nome
ha
Curso (FK)

Semestre

SQL – ALTER TABLE – Exemplo

30

- Alteração de colunas da tabela – Exemplo

ALTER TABLE disciplinas

CHANGE horas **TO** ha **INT NOT NULL,**

ADD semestre **INT NOT NULL,**

DROP semestre,

ADD CONSTRAINT FOREIGN KEY (curso)

REFERENCES cursos (codcurso) ;

Disciplinas

#CodDisc
Nome
ha
Curso (FK)

Cursos

#CodCurso
Nome
Datacriacao
Area
Duracao

SQL – DROP TABLE – Exemplo

31

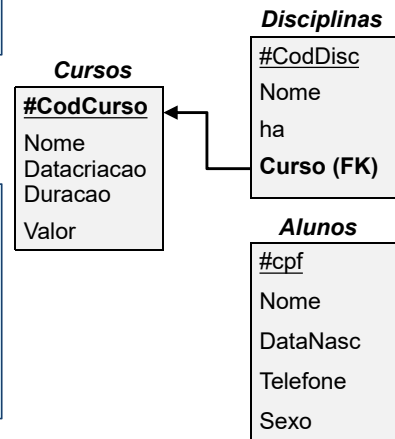
- Exclusão de relações (tabelas) – Exemplo

```
DROP TABLE alunos;
```

- Não é permitido deletar tabela que seja referenciada por outra tabela. Exemplo:

```
mysql> DROP TABLE cursos;
```

ERROR 1217: Cannot delete or update a parent row: a foreign constraint fails



SQL – Manipulando Tabelas

32

- Manutenção Geral de Tabelas

- Mostra Tabelas existentes no banco

```
SHOW TABLES;
```

- Mostra a Descrição das colunas e restrições da Tabela

```
DESC (ou DESCRIBE) [nome_da_tabela];
```

- Mostra as Chaves de uma Tabela

```
SHOW KEYS FROM [nome_da_tabela];
```

- Apaga todos os dados de uma Tabela

```
TRUNCATE TABLE [nome_da_tabela];
```


SQL – DDL – Resumindo

33

- A linguagem de definição de dados (DDL) é um conjunto de comandos da SQL para definir tabelas, chaves, procedimentos, entre outros.
- **CREATE / DROP DATABASE** – permite fazer manutenção do banco de dados, seus arquivos, senha, conjuntos de caracteres.
- **CREATE / ALTER / DROP TABLE** – permite criar, fazer manutenção das tabelas, definir seus atributos, suas chaves primárias e estrangeiras, bem como deletar tabelas.

Dúvidas?

34



Prof. Me. Fernando Roberto Proença

fernandorroberto@gmail.com

Referências

35

- DATE, C. J. **Introdução a Sistemas de Bancos de Dados**. 8. ed. Editora Campus, 2004.
- ELMASRI, R.; NAVATHE, S. **Sistemas de banco de dados**. 6ª ed. São Paulo: Pearson, 2011. Capítulo 4.
- ROB, P.; CORONEL, C. **Sistemas de banco de dados** : Projeto, implementação e gerenciamento. 8ª ed. São Paulo: Cengage Learning, 2014. Capítulo 7.
- Agradeço à professora **Renata de Oliveira Rodrigues**, que cedeu o material base para montar esta apresentação.