

## SQL - Structured Query Language : DDL – Data Definition Language

### Introdução

- SQL (Structured Query Language);
- Usa uma combinação da álgebra relacional e construções de cálculo relacional;
- Foi desenvolvida pela IBM no início dos anos 70 e mais tarde se tornou um padrão ANSI chamada SQL-1 em 1986;
- Se estabeleceu como a linguagem padrão para banco de dados relacional;
- Embora seja chamada de “linguagem de consulta” ela contém outras capacidades além de consultas a banco de dados;
- Inclui recursos para definição de estruturas, modificação e restrições de dados.

## Partes da Linguagem

- **Linguagem de Definição de Dados (DDL):** A DDL fornece comandos para definição, remoção e modificação de esquemas de relações (tabelas), criação de índices e chaves, etc.;
- **Linguagem Interativa de Manipulação de Dados (DML):** A DML inclui uma linguagem de consulta baseada na álgebra relacional e no cálculo relacional de tuplas. Contém também comandos para inserção, exclusão e modificação de tuplas no banco de dados;
- **DML Embutida:** É a forma embutida da SQL que foi projetada para utilização em linguagens de programação de uso geral como Cobol, C, Pascal;

## Partes da Linguagem (cont.)

- **Definição de Visões:** É a parte da DDL que inclui comandos para definição de visões;
- **Autenticação:** É a parte da DDL que inclui comandos para especificação de autorização de acesso a relações e visões;
- **Integridade:** Possui comandos para a especificação de regras de integridade;
- **Controle de Transações:** Fornece comandos para especificação de início e fim de transações, e bloqueios de dados para controle de concorrência.

## Definição de Dados em SQL

- Usado para CRIAR, EXCLUIR, e ALTERAR a descrição das tabelas (relações) de um banco de dados

## CREATE TABLE

- Especifica uma nova relação, dando-lhe um nome e especificando cada um de seus atributos e seus tipos de dados (INTEGER, FLOAT, DECIMAL(i,j), CHAR(n), VARCHAR(n))
- Uma restrição NOT NULL pode ser especificada para um atributo que não aceita valor nulo

```
CREATE TABLE DEPARTAMENTO (  
    DNAME          VARCHAR(10)      NOT NULL,  
    DNUMERO        INTEGER          NOT NULL,  
    GERSSN         CHAR(9),  
    GERDATAINICIO  CHAR(9) );
```

## CREATE TABLE

- Em SQL2, o comando CREATE TABLE pode ser usado para especificar a chave primária, a chave secundária e restrições de integridade referencial (chaves estrangeiras).
- Atributos chave podem ser especificados através dos comandos PRIMARY KEY e UNIQUE

```
CREATE TABLE DEPT (  
    DNAME          VARCHAR(10)    NOT NULL,  
    DNUMERO        INTEGER        NOT NULL,  
    GERSSN         CHAR(9),  
    GERDATAINICIO  CHAR(9),  
    PRIMARY KEY (DNUMERO),  
    UNIQUE (DNAME),  
    FOREIGN KEY (GERSSN) REFERENCES  
    EMP(SSN) );
```

Copyright © 2007 Ramez Elmasri and Shamkant B. Navathe

7

## DROP TABLE

- Usada para excluir elementos de um esquema, como relações (tabelas), domínios ou restrições
- A relação não poderá mais ser utilizada em consultas, atualizações ou qualquer outro comando visto que ela já não existe mais
- Exemplo:

```
DROP TABLE  DEPENDENTE;
```

Copyright © 2007 Ramez Elmasri and Shamkant B. Navathe

8

## ALTER TABLE

- Usado para alterar elementos do esquema.
- As ações de alteração podem ser: adicionar / eliminar um atributo, alterar um atributo ou adicionar / eliminar restrições
  - O novo atributo terá NULLs em todas as tuplas da relação logo após a adição; consequentemente, a restrição NOT NULL não é permitida
- Exemplo:  
`ALTER TABLE EMPREGADO ADD FUNCAO VARCHAR(12);`
- O usuário deve informar um valor para o novo atributo FUNCAO para cada tupla EMPREGADO.
  - Isso pode ser feito usando o comando UPDATE

Copyright © 2007 Ramez Elmasri and Shamkant B. Navathe

9

## Características adicionadas pela SQL-2 (ou SQL-92)

- Criação de esquema
- Opções de integridade referencial

Copyright © 2007 Ramez Elmasri and Shamkant B. Navathe

10

## CREATE SCHEMA

- Especifica um novo esquema de banco de dados, dando-lhe um nome

- Em SQL padrão:

```
CREATE SCHEMA EMPRESA
AUTHORIZATION Silva;
```

- Em PostgreSQL:

```
CREATE DATABASE postgres
WITH OWNER = postgres
ENCODING = 'UTF8';
```

## OPÇÕES BÁSICAS DE INTEGRIDADE

- Restrições de Atributos

- Em SQL é permitido NULL como valor de atributos
- Uma restrição NOT NULL implica que o valor do atributo não pode ser vazio (NULL)

```
CREATE TABLE DEPT (
    DNOME          VARCHAR(10)    NOT NULL,
    DNUMERO        INTEGER        NOT NULL,
    ...
);
```

## OPÇÕES BÁSICAS DE INTEGRIDADE

### ■ Restrições de Padrão (*default*) de Atributos

- Em SQL é também possível definir um valor default para um atributo
- Adicionando a cláusula `DEFAULT <valor>` na definição do atributo

```
CREATE TABLE DEPT (  
    ...  
    GERSSN          CHAR(9)  DEFAULT 555,  
    GERDATAINICIO   CHAR(9)  DEFAULT '8/01/2009',  
    ...  
);
```

Copyright © 2007 Ramez Elmasri and Shamkant B. Navathe

13

## OPÇÕES BÁSICAS DE INTEGRIDADE

### ■ Restrições de Verificação de Atributos

- Em SQL restrições podem limitar os valores de atributos ou de domínios pelo uso da cláusula `CHECK`
- Adicionando a cláusula `CHECK <restrição>` na definição do atributo

```
CREATE TABLE DEPT (  
    DNAME           VARCHAR(10)  NOT NULL,  
    DNUMERO         INTEGER       NOT NULL  
                    CHECK (DNUMERO > 0),  
    ...  
);
```

Copyright © 2007 Ramez Elmasri and Shamkant B. Navathe

14

## OPÇÕES BÁSICAS DE INTEGRIDADE

### ■ Restrições de Unicidade de Atributos

- A restrição de unicidade garante que os dados contidos na coluna é único em relação a todas as outras linhas da tabela

```
CREATE TABLE DEPT (  
    DNOME          VARCHAR(10)    NOT NULL,  
    DNUMERO        INTEGER        NOT NULL,  
    SSN            INTEGER        UNIQUE  
    ...  
);
```

Copyright © 2007 Ramez Elmasri and Shamkant B. Navathe

15

## OPÇÕES DE INTEGRIDADE REFERENCIAL

- Pode-se especificar restrições de integridade referencial tais como: RESTRICT, CASCADE, SET NULL or SET DEFAULT

```
CREATE TABLE DEPT (  
    DNOME          VARCHAR(10)    NOT NULL,  
    DNUMERO        INTEGER        NOT NULL,  
    GERSSN         CHAR(9),  
    GERDATAINICIO CHAR(9),  
    PRIMARY KEY (DNUMERO),  
    UNIQUE (DNOME),  
    FOREIGN KEY (GERSSN) REFERENCES EMP  
    ON DELETE SET DEFAULT ON UPDATE  
    CASCADE);
```

Copyright © 2007 Ramez Elmasri and Shamkant B. Navathe

16



## OPÇÕES DE INTEGRIDADE REFERENCIAL (cont)

```
CREATE TABLE EMP(  
    ENOME      VARCHAR(30)    NOT NULL,  
    ESSN       CHAR(9),  
    BDATA      DATE,  
    DNO        INTEGER  DEFAULT 1,  
    SUPERSSEN  CHAR(9),  
    PRIMARY KEY (ESSN),  
    FOREIGN KEY (DNO) REFERENCES DEPT  
        ON DELETE SET DEFAULT ON UPDATE  
        CASCADE,  
    FOREIGN KEY (SUPERSSEN) REFERENCES EMP  
        ON DELETE SET NULL ON UPDATE CASCADE);
```

Copyright © 2007 Ramez Elmasri and Shamkant B. Navathe

17

## Tipos de Dados de Atributos e Domínios em SQL

- **Numérico:** Englobam os números inteiros de vários tamanhos (INT ou SMALLINT) e os ponto flutuantes (FLOAT ou REAL)
- **Cadeia de Caracteres:** As cadeias de caracteres podem ter tamanho fixo – CHAR(n) – ou ter tamanho variável – VARCHAR(n)
- **Booleano:** O tipo de dado booleano possui os valores verdadeiro (TRUE) e falso (FALSE). Em virtude da presença de valores NULLs, é usado um terceiro valor para o tipo booleano, o desconhecido (UNKNOWN)

Copyright © 2007 Ramez Elmasri and Shamkant B. Navathe

18

## Tipos de Dados Adicionais da SQL2 e SQL-99 (cont.)

- **DATE:**
  - Formata o ano-mês-dia como aaaa-mm-dd
- **TIME:**
  - Formada a hora:minuto:segundo como hh:mm:ss
- **TIME(i):**
  - Formada a hora:minuto:segundo mais i dígitos adicionais especificando frações de um segundo
  - O formato é hh:mm:ss:ii...i

Copyright © 2007 Ramez Elmasri and Shamkant B. Navathe

19

## Tipos de Dados Adicionais da SQL2 e SQL-99 (cont.)

- **TIMESTAMP:**
  - Tem ambos os componentes DATE e TIME
- **INTERVAL:**
  - Especifica um valor relativo ao invés de um valor absoluto
  - Pode ser intervalos de DAY/TIME ou intervalos YEAR/MONTH
  - Pode ser positivo ou negativo quando adicionado a ou subtraído de um valor absoluto, o resultado é um valor absoluto

Copyright © 2007 Ramez Elmasri and Shamkant B. Navathe

20

## Tipos de Dados do PostgreSQL - Numéricos

Nome	Tamanho de armazenamento	Descrição	Faixa de valores
<code>smallint</code>	2 bytes	inteiro com faixa pequena	-32768 a +32767
<code>integer</code>	4 bytes	escolha usual para inteiro	-2147483648 a +2147483647
<code>bigint</code>	8 bytes	inteiro com faixa larga	-9223372036854775808 a 9223372036854775807
<code>decimal</code>	variável	precisão especificada pelo usuário, exato	sem limite
<code>numeric</code>	variável	precisão especificada pelo usuário, exato	sem limite
<code>real</code>	4 bytes	precisão variável, inexacto	precisão de 6 dígitos decimais
<code>double precision</code>	8 bytes	precisão variável, inexacto	precisão de 15 dígitos decimais
<code>serial</code>	4 bytes	inteiro com auto-incremento	1 a 2147483647
<code>bigserial</code>	8 bytes	inteiro grande com auto-incremento	1 a 9223372036854775807

Copyright © 2007 Ramez Elmasri and Shamkant B. Navathe

21

## Tipos de Dados do PostgreSQL - Monetário

- O tipo monetário está em desuso. Em seu lugar deve ser utilizado o tipo `numeric` ou `decimal`

Nome	Tamanho de Armazenamento	Descrição	Faixa
<code>money</code>	4 bytes	quantia monetária	-21474836.48 a +21474836.47

Copyright © 2007 Ramez Elmasri and Shamkant B. Navathe

22

## Tipos de Dados do PostgreSQL – Cadeias de Caracteres

Nome	Descrição
<code>character varying(n)</code> , <code>varchar(n)</code>	comprimento variável com limite
<code>character(n)</code> , <code>char(n)</code>	comprimento fixo, completado com brancos
<code>text</code>	comprimento variável não limitado

- Onde `n` é um número inteiro positivo e indica que pode ser armazenada uma cadeia de caracteres com comprimento de até `n` caracteres.

## Tipos de Dados do PostgreSQL – Dado Binário

- O tipo de dado `bytea` permite o armazenamento de cadeias binárias, ou seja, sequências de octetos

Nome	Tamanho de Armazenamento	Descrição
<code>bytea</code>	4 bytes mais a cadeia binária	Cadeia binária de comprimento variável

## Tipos de Dados do PostgreSQL – Data e Hora

Nome	Tamanho de Armazenamento	Descrição	Menor valor	Maior valor	Resolução
timestamp [ (p) ] [ without time zone ]	8 bytes	tanto data quanto hora	4713 AC	5874897 DC	1 microssegundo / 14 dígitos
timestamp [ (p) ] with time zone	8 bytes	tanto data quanto hora, com zona horária	4713 AC	5874897 DC	1 microssegundo / 14 dígitos
interval [ (p) ]	12 bytes	intervalo de tempo	-178000000 anos	178000000 anos	1 microssegundo / 14 dígitos
date	4 bytes	somente data	4713 AC	32767 DC	1 dia
time [ (p) ] [ without time zone ]	8 bytes	somente a hora do dia	00:00:00.00	23:59:59.99	1 microssegundo / 14 dígitos
time [ (p) ] with time zone	12 bytes	somente a hora do dia, com zona horária	00:00:00.00+12	23:59:59.99-12	1 microssegundo / 14 dígitos

Copyright © 2007 Ramez Elmasri and Shamkant B. Navathe

25

## Tipos de Dados do PostgreSQL – Formato de Data e Hora

- Utilizando o comando `set datastyle` o formato da saída para os tipos data e hora pode ser definido

Especificação de estilo	Descrição	Exemplo
ISO	ISO 8601/padrão SQL	2005-04-21 18:39:28.283566-03
SQL	estilo tradicional	04/21/2005 18:39:28.283566 BRT
POSTGRES	estilo original	Thu Apr 21 18:39:28.283566 2005 BRT
German	estilo regional	21.04.2005 18:39:28.283566 BRT

Copyright © 2007 Ramez Elmasri and Shamkant B. Navathe

26

## Tipos de Dados do PostgreSQL – Endereço de Rede

- O PostgreSQL disponibiliza tipos de dados para armazenar endereços IPv4, IPv6 e MAC
- É preferível utilizar estes tipos em vez dos tipos texto puro, porque possuem verificação de erro

Nome	Tamanho de Armazenamento	Descrição
<code>cidr</code>	12 ou 24 bytes	redes IPv4 e IPv6
<code>inet</code>	12 ou 24 bytes	hospedeiros e redes IPv4 e IPv6
<code>macaddr</code>	6 bytes	endereço MAC

Copyright © 2007 Ramez Elmasri and Shamkant B. Navathe

27

## Tipos de Dados do PostgreSQL – Matrizes

- O PostgreSQL permite que colunas de uma tabela sejam definidas como matrizes (arrays) multidimensionais de comprimento variável.
- Podem ser criadas matrizes de qualquer tipo de dado, nativo ou definido pelo usuário
- Entretanto, ainda não são suportadas matrizes de tipos compostos ou domínios (BDOR)

Copyright © 2007 Ramez Elmasri and Shamkant B. Navathe

28