

Banco de Dados Orientado a Objetos

Banco de Dados Objeto- Relacional

História da SQL

- 1986:
 - SQL-86 – Primeira versão padronizada da SQL
 - Inicialmente desenvolvida no âmbito da ANSI (*American National Standards Institute* – Instituto Nacional Americano de Padrões)
- 1987:
 - Aprovada pela ISO (*International Organization for Standardization* – Organização Internacional para Padronização)
- 1989:
 - SQL-89 – Extensão do padrão SQL-86

História da SQL

- 1992:
 - SQL-92 – Aprovada pela ISO
- 1999:
 - SQL-99 ou SQL3 – Aprovada pela ISO
 - Definidos os usos de *triggers*, *stored procedures*, consultas recursivas, entre outros
 - Definiu regras para os SGBDOR (Sistema de Gerenciamento de Bancos de Dados Objeto-Relacional)

História da SQL

- 2003:
 - SQL-2003
 - Introduzindo características relacionadas ao XML (*eXtensible Markup Language* – Linguagem de Marcação Extensiva), sequências padronizadas e colunas com valores de auto-generalização.
- 2008:
 - SQL-2008
 - Trouxe nas especificações formas para a SQL poder ser usada em conjunto com XML

História da SQL

- 2003:
 - SQL-2003
 - Introduzindo características relacionadas ao XML (*eXtensible Markup Language* - Linguagem de Marcação Extensiva), sequências padronizadas e colunas com valores de auto-generalização.
- 2008:
 - SQL-2008
 - Trouxe nas especificações formas para a SQL poder ser usada em conjunto com XML

Tipo Composto

- Também conhecido como tipos definidos pelo usuário (*UDT – User-Defined Types*)
- Podem ser definidos tipos complexos
- Sintaxe:

```
CREATE TYPE nome_tipo AS (  
    atributo tipo_dado [ {, atributo tipo_dado} ]  
);
```

Tipo Composto

- Exemplo:

```
CREATE TYPE obd.tipo_endereco AS (  
    logradouro VARCHAR(45),  
    municipio VARCHAR(25),  
    uf CHAR(2),  
    cep CHAR(10)  
);  
  
CREATE TYPE obd.tipo_empregado AS (  
    nome VARCHAR(40),  
    dt_nasc DATE,  
    endereco obd.tipo_endereco  
);
```

Tipo Composto

- Usando:

```
CREATE TABLE obd.empregado (  
    id INTEGER NOT NULL PRIMARY KEY,  
    tipo obd.tipo_empregado  
);  
  
INSERT INTO obd.empregado (  
    id, tipo.nome, tipo.dt_nasc,  
    tipo.endereco.logradouro, tipo.endereco.municipio,  
    tipo.endereco.uf, tipo.endereco.cep  
) VALUES (  
    1, 'Tadeu', '01/01/1981',  
    'Rua Padre Cicero', 'Juazeiro do Norte', 'CE', '63.010-010'  
);
```

Tipo Composto

- Usando:

```
SELECT * FROM obd.empregado;
```

```
SELECT
    id, (tipo).nome, (tipo).dt_nasc,
    (tipo).endereco.logradouro, (tipo).endereco.municipio,
    (tipo).endereco.uf, (tipo).endereco.cep
FROM obd.empregado;
```

```
UPDATE obd.empregado
SET tipo.endereco.cep = '63.010-020'
WHERE id = 1;
```

Tipo Multivalorado

- Sintaxe:

```
CREATE TABLE nome_tabela (
    atributo tipo_dado ARRAY[n], [ {, atributo tipo_dado} ] );
CREATE TABLE nome_tabela (
    atributo tipo_dado[ ], [ {, atributo tipo_dado} ] );
```

- Exemplo:

```
CREATE TABLE obd.pessoa (
    id INTEGER NOT NULL PRIMARY KEY,
    nome VARCHAR(40),
    -- telefones VARCHAR(15) ARRAY[5]
    telefones VARCHAR(15) []
);
```

Tipo Multivalorado

- Usando:

```
INSERT INTO obd.pessoa (id, nome, telefones)
VALUES (1, 'José', ARRAY['1111-1111', '2222-2222']);
```

```
SELECT * FROM obd.pessoa;
```

```
SELECT id, nome, telefones[1], telefones[2] FROM
obd.pessoa;
```

```
UPDATE obd.pessoa SET telefones[3] = '3333-3333' WHERE
id = 1;
```

```
UPDATE obd.pessoa SET telefones[7] = '7777-7777' WHERE
id = 1;
```

Herança de Tabelas

- Sintaxe:

```
CREATE TABLE tabela_filha (
    atributo tipo_dado [ {, atributo tipo_dado} ]
) INHERITS (tabela_pai);
```

- Exemplo:

```
CREATE TABLE obd.cidade (
    nome VARCHAR(30) NOT NULL PRIMARY KEY,
    populacao INTEGER, altitude INTEGER
);
```

```
CREATE TABLE obd.capital (
    uf CHAR(2),
    PRIMARY KEY(nome)
) INHERITS (obd.cidade);
```

Herança de Tabelas

- Usando:

```
INSERT INTO obd.cidade (nome, populacao, altitude)
VALUES ('Crato', 150000, 275),
VALUES ('JUAZEIRO DO NORTE', 250000, 270);
```

```
INSERT INTO obd.capital (nome, populacao, altitude, uf)
VALUES ('Natal', 750000, 0, 'RN');
```

```
SELECT * FROM obd.capital;
```

```
SELECT * FROM obd.cidade;
```

```
SELECT * FROM ONLY obd.cidade;
```

Herança de Tabelas

- Usando:

```
UPDATE obd.cidade SET populacao = populacao * 1.1;
```

```
UPDATE ONLY obd.cidade SET populacao = populacao * 1.1;
```

```
UPDATE obd.capital SET populacao = populacao * 1.5;
```

```
SELECT tableoid, * FROM obd.cidade;
```

Herança de Tipo Composto

- Sintaxe:

```
CREATE TABLE tabela_filha OF tipo_pai;
```

- Exemplo:

```
CREATE TABLE obd.endereco OF obd.tipo_endereco;
ALTER TABLE obd.endereco ADD PRIMARY KEY(cep);
```

- Usando:

```
INSERT INTO obd.endereco (logradouro, municipio, uf, cep)
VALUES ('Rua Padre Cicero', 'Juazeiro do Norte', 'CE', '63.000-000');
```

```
UPDATE obd.endereco SET cep = '63.050-050'
WHERE cep = '63000-000';
```