Modelo Físico

Banco de Dados I

Modelo Físico (Projeto BD)

- Até o momento, trabalhamos com:
  - Modelo Conceitual
  - Modelo Lógico

- O que é o **modelo físico**?

Modelo Físico (SQL)

- O que é SQL (Structured Query Language)?

  É um conjunto de comandos para manipulação de banco de dados que abrange desde a **criação/alteração e exclusão da estrutura do banco de dados**, bem como a **manipulação (inclusão, alteração, exclusão e recuperação) dos dados**

Modelo Físico (SQL)

- Histórico
  - 1970 – E. F. Codd, pesquisador do Laboratório de Pesquisa da IBM, publicou um artigo em que era considerado um modelo relacional
  - 1974 – Desenvolvimento pela IBM da SEQUEL (Structured English Language). Baseado no conceito proposto por E. F. Codd
  - 1977 – Foi oficializada como SQL (Structured Query Language)
  - 1979 – A Relational Software Inc. (Atual Oracle), lançou a primeira versão comercial da linguagem SQL

Modelo Físico (SQL)

- Histórico
  - É considerada uma linguagem padrão para manipulação de dados
  - 1986 – (SQL-86 ou SQL-87). Padronizada pela ANSI/ISO
  - 1987 – A IBM Lança seu próprio padrão SAA-SQL
  - 1989 – (SQL-89 ou FIPS 127-1). Alterações para Banco de Dados atuais
  - **1992 – (SQL-92 ou FIPS 127-2). Padrão utilizado atualmente**

Modelo Físico (SQL)

- Histórico
  - 1999 – (SQL:1999 ou SQL3). Recursividade, ***triggers***, características de orientação a objetos e domínios
  - 2003 – (SQL:2003). Introdução do padrão XML, padronização de ***sequences***, colunas com valores de auto incremento
  - 2006 – (SQL:2006). Importação, exportação e armazenamento em dados XML em Banco de Dados Relacionais, **W3C**
  - 2008 – (SQL:2008). Definição da cláusula "**order by**"

Modelo Físico (SQL)

- A SQL está dividida em:
  - **DDL (Data Definition Language)**
    - **Criação do MODELO FÍSICO**
  - DML (Data Manipulation Language)
    - Manipulação de dados
    - Inclusão, alteração e exclusão
  - DQL (Data Query Language)
    - Recuperação/extração de dados
  - DCL (Data Control Language)
    - Fornece a segurança interna dos dados e o próprio modelo de dados

Modelo Físico (SQL)

- Antes de desenvolvermos o modelo físico, observamos que, do **modelo lógico** para o **modelo físico** bastaria agregarmos o **domínio** de cada atributo bem como algumas **regras**

Modelo Físico (SQL)

- Tipos de dados (domínio)
  - Numéricos exatos:
    - INTEGER (INT) e SMALLINT para representar inteiros
    - NUMERIC(p,s): tem uma precisão e uma escala(número de dígitos na parte fracionária). A escala não pode ser maior que a precisão. Muito usado para representar dinheiro
    - DECIMAL: também tem precisão e escala. A precisão é fornecida pela implementação (SGBD)
  - Numéricos aproximados:
    - REAL: ponto flutuante de precisão simples
    - DOUBLE: ponto flutuante com precisão dupla
    - FLOAT(p): permite especificar a precisão que se quer. Usado para portabilidade em aplicações

Modelo Físico (SQL)

- Character
  - CHARACTER(x) (CHAR): representa um string de tamanho x. Se x for omitido então é equivalente a CHAR(1). Se um string a ser armazenado é menor do que x, então o restante é preenchido com brancos
  - CHARACTER VARYING(x) (VARCHAR): representa um string de tamanho x. Armazena exatamente o tamanho do string (tam <= x) sem preencher o resto com brancos. Neste caso x é obrigatório. O Banco de Dados Oracle utiliza o VARCHAR2 que possui o mesmo comportamento
  - CHARACTER LARGE OBJECT (CLOB): armazena strings longos. Usado para armazenar documentos
  - OBS.: Existem os *National character data types*: NCHAR, NVARCHAR, NCLOB que permitem implementar internacionalização

Modelo Físico (SQL)

- Bit string e Binary Strings (BLOB)
  - BIT(X): permite armazenar uma quantidade “x” de bits
  - BIT VARING(X) (VARBIT): permite armazenar uma quantidade variável de bits até o tamanho X
  - BINARY LARGE OBJECT (BLOB): para armazenar grandes quantidades de bytes como fotos, vídeo, áudio, gráficos, mapas, etc.

Modelo Físico (SQL)

- DATETIMES
  - DATE: armazena ano (4 dígitos), mês (2 dígitos) e dia(2 dígitos)
  - TIME: armazena hora (2 dígitos), minuto (2 dígitos) e segundo (2 dígitos, podendo ter frações 0 a 61.9999)
  - TIMESTAMP: DATE + TIME
  - TIME WITH TIME ZONE: igual a time + UTC offset
  - TIMESTAMP WITH TIME ZONE: igual a TIMESTAMP + UTC offset

- Existem outras definições de domínio (boolean, intervals, collections, tipos definidos pelo usuário, referências, ...)

Modelo Físico (DDL)

- Os comandos SQL para definição de dados são:
  - **Create**
  - **Drop**
  - **Alter**

- Para as seguintes estruturas:
  - Schema, domain, **table**, view, index, assertion, **database**

Modelo Físico (DDL)

- Sintaxe para criação de tabelas

CREATE [{LOCAL TEMPORARY|GLOBAL TEMPORARY}] **TABLE nome_tabela**

(nome_coluna tipo_dado atributos [,…])

[CONSTRAINT [nome_constraint] tipo_constraint [,…]]

Exemplo básico:

```
CREATE TABLE Pessoa
(Matricula integer,
Nome varchar(35),
CPF varchar(11))
```

Modelo Físico (DDL)

- Constraints (restrições) mais comuns:
  - Chave primária (Primary key)
  - Chave estrangeira (Foreign Key)
  - Check
  - Unique

  - As duas primeiras já conhecemos!

Exemplo:

```
CREATE TABLE Pessoa
(Matricula integer PRIMARY KEY,
Nome varchar(35),
CPF varchar(11))
```

Modelo Físico (DDL)

- Constraints (restrições) mais comuns:

  Banco de Dados implementam as restrições em alguns casos diferente do padrão.

  Então, poderá ser definido uma restrição do tipo "check", tanto ao lado da coluna, como ao final do bloco de criação da tabela!

Exemplo:

```
CREATE TABLE Pessoa
(Matricula integer NOT NULL,
Nome varchar(35),
CPF varchar(11),
CONSTRAINT PessoaPK PRIMARY KEY
    (Matricula))
```

Modelo Físico (DDL)

- Atributos podem ser:
  - **NOT NULL**
  - **DEFAULT**
  - **UNIQUE**

Modelo Físico (DDL)

- Default
  - Atribuir um conteúdo padrão a uma coluna da tabela, sempre que for incluída uma nova linha na tabela

Exemplo:

```
CREATE TABLE Pessoa
(Matricula integer PRIMARY KEY,
Nome varchar(35) DEFAULT ‘Maria’,
CPF varchar(11))
```

Modelo Físico (DDL)

- Not Null
  - Indica que o conteúdo de uma coluna não pode ser Nulo

Exemplo:

```
CREATE TABLE Pessoa
(Matricula integer PRIMARY KEY,
Nome varchar(35) NOT NULL DEFAULT
     'Maria',
CPF varchar(11))
```

  - O padrão ao criar um campo é null

Modelo Físico (DDL)

- Unique

  Mesmo conceito que a chave primária, ou seja, não poderá estar repetido o valor do atributo

Exemplo:

```
CREATE TABLE Pessoa
(Matricula integer PRIMARY KEY,
Nome varchar(35) NOT NULL DEFAULT
    'Maria',
CPF varchar(11) UNIQUE)
```

Modelo Físico (DDL)

- Check

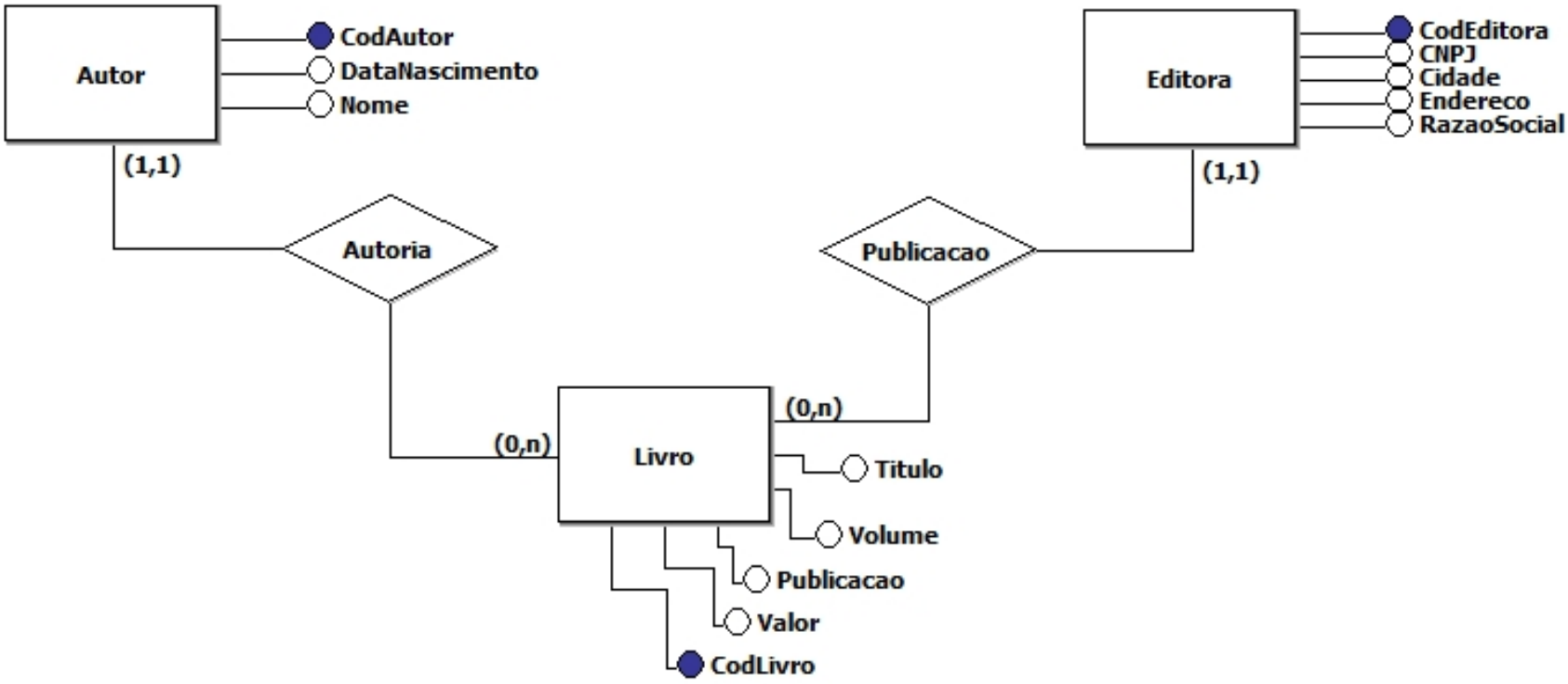
  Permite validar um determinado conteúdo de entrada na coluna

Exemplo:

```
CREATE TABLE Pessoa
(Matricula integer PRIMARY KEY,
Nome varchar(35) NOT NULL DEFAULT
     'Maria',
Sexo char(1) CHECK (Sexo = 'M' or Sexo = 'F'),
CPF varchar(11) UNIQUE)
```

Modelo Físico (DDL)

- Utilizaremos o seguinte exemplo para entender a criação do modelo físico



Modelo Físico (DDL)

- Abordagem relacional resumida, teríamos:

AUTOR (codautor, nome, datanascimento)

EDITORA (codeditora, razaosocial, cnpj, endereco, cidade)

LIVRO (codlivro, codautor, codeditora, titulo, publicacao, volume, valor)

codautor referencia autor

codeditora referencia editora

Modelo Físico (DDL)

- Convertendo modelo lógico para físico

AUTOR (codautor, nome, datanascimento)

```
CREATE TABLE Autor
(codautor integer NOT NULL ,
nome varchar(30) NOT NULL,
datanascimento date NULL,
PRIMARY KEY (codautor))
```

Modelo Físico (DDL)

- Convertendo modelo lógico para físico

EDITORA (codeditora, razaosocial, cnpj, endereco, cidade)

```
CREATE TABLE Editora
(codeditora integer NOT NULL,
 razaosocial varchar(40) NOT NULL,
 cnpj varchar (14) NOT NULL UNIQUE,
 endereco varchar(40) NOT NULL,
 cidade varchar(25) DEFAULT “São Leopoldo”,
 PRIMARY KEY (codeditora))
```

Modelo Físico (DDL)

- Convertendo modelo lógico para físico

LIVRO (codlivro, codautor, codeditora, titulo, publicacao, volume, valor)

codautor referencia autor

codeditora referencia editora

```
CREATE TABLE Livro
(codlivro integer,
codautor integer NOT NULL,
codeditora integer NOT NULL,
titulo varchar(60) NOT NULL,
publicacao date NOT NULL,
volume integer DEFAULT 1,
valor double NOT NULL,
PRIMARY KEY (codlivro),
FOREIGN KEY (codautor) REFERENCES Autor (codautor),
FOREIGN KEY (codeditora) REFERENCES Editora
(codeditora))
```

Modelo Físico (DDL)

- **ALTER TABLE**
  - Permite alterar ou adicionar atributos de uma determinada tabela.
  - Os novos atributos terão valores nulos em todas as linhas
  - Utilizado na alteração ou exclusão das restrições da tabela

Modelo Físico (DDL)

- **ALTER TABLE**

Sintaxe

ALTER TABLE nome_tabela
    ADD [COLUMN] nome_coluna tipo_dado

Para modificar uma coluna de uma tabela

ALTER TABLE nome_tabela
    MODIFY [COLUMN] nome_coluna
     SET valor-default
     ou
    DROP DEFAULT

Obs. 1: em alguns BDs, [COLUMN] não existe.

Obs. 2: se a tabela já possui tuplas, é possível incluir uma nova coluna somente “null”.

Modelo Físico (DDL)

- **ALTER TABLE**

Para remover uma coluna de uma tabela:

ALTER TABLE nome_tabela

    DROP [COLUMN] nome_coluna

Para adicionar uma restrição a uma tabela:

ALTER TABLE nome_tabela

    ADD CONSTRAINT nome_restrição e seus parâmetros

Para remover uma restrição de um tabela

ALTER TABLE nome_tabela

    DROP CONSTRAINT nome_restrição

Obs.: em alguns Bancos de Dados não utiliza-se a palavra CONSTRAINT (ao excluir é opcional).

Modelo Físico (DDL)

- **Exemplos**

ALTER TABLE Livro

ADD idioma varchar(15)

(Adiciona a coluna idioma com 15 espaços na tabela Livro)

ALTER TABLE Livro

MODIFY idioma varchar(30)

(Altera a coluna idioma para 30 espaços)

ALTER TABLE Livro

DROP idioma

(Exclui a coluna idioma)

Modelo Físico (DDL)

- **Exemplos**

ALTER TABLE Livro

DROP PRIMARY KEY

(Exclui a chave primária da tabela Livro)

ALTER TABLE Livro

ADD CONSTRAINT PRIMARY KEY (codlivro)

(Adiciona novamente a chave primária)

Modelo Físico (DDL)

- **DROP TABLE**

  Remove uma tabela do banco de dados

  Sintaxe:

  DROP TABLE nome_tabela

  Exemplo:

  DROP TABLE Livro

Referências Bibliográficas

- Material originalmente elaborado por Prof. Gilberto Irajá Müller. Material autorizado e cedido pelo autor. Revisado e atualizado por Prof. João Tavares.
- HEUSER, Carlos Alberto. **Projeto de Banco de Dados**. 6. ed. Bookman Companhia Ed, 2009.
- DATE, C. J. **Introdução aos Sistemas de Bancos de Dados**. Rio de Janeiro: Campus, 2004.
- SILBERSCHATZ, Abraham; KORTH, Henry F.; SUDARSHAN, S. **Sistemas de Banco de Dados**. 3ª. Ed. São Paulo: Makron Books, 2010.