Projeto Físico
SQL – Parte 1

Banco de Dados I

Projeto Físico

- Linguagem de banco de dados
  - Linguagem usada para implementar o mecanismos de persistência
    - Mecanismos de Persistência = Banco de dados Relacional
    - Linguagem de banco de dados = SQL
  - Os comandos SQL são executados via SGBD

SQL

- Padrão
  - SQL-86
  - SQL-89
  - SQL-92
  - SQL:1999
  - SQL:2003

DDL

- Permite a especificação das relações e informações sobre cada relação
  - Esquema para cada relação
    - Atributo e o domínio de cada atributo
  - Restrições de integridade
  - Índices a serem mantidos para cada relação
  - Informações de segurança e autorização para cada relação
  - Estrutura de armazenamento físico de cada relação no disco

DDL

- Comando create
  - Criar elementos do esquema do banco de dados
    - Banco de dados (databases)
      - Coleção de dados relacionados organizados em tabelas
    - Tabela
      - Coleção de linhas e colunas que representam os identificados no domínio da aplicação
- Comando drop
  - Remover elementos do esquema do banco de dados
    - Remover banco de dados
    - Remover tabelas

DDL

- Criar banco de dados
  - create database nomeDoBanco
  - create database empresa
- Remover banco de dados
  - drop database nomeDoBanco
  - drop database empresa

nomeDoBanco: identificador da base de dados

DDL

- Criar tabela

  create table ***nomeTabela*** **(**

  nomeColuna1 domínioDaColuna1,

  nomeColuna2 domínioDaColuna2,

  …

)

DDL

- Criar tabela agencia com os atributos codigo (5), nome (até 50), ativo (inteiro), cidade (até 50)

```
create table agencia
(codigo char(5),
nome varchar(50),
ativo int,
cidade varchar(50))
```

DDL – Tipos de dados

- Tipos (padrão SQL)
  - char(n): string de caracteres de tamanho fixo com tamanho n especificado pelo usuário
  - varchar(n): string de caracteres de tamanho variável com tamanho n máximo especificado pelo usuário
  - int: inteiro (um subconjunto finito de inteiros que é dependente da máquina)
  - smallint: inteiro pequeno (um subconjunto dependente da máquina do tipo de domínio inteiro)
  - numeric(p,d): número de ponto fixo, com precisão de p dígitos especificada pelo usuário, com n dígitos à direita do ponto decimal
  - real, double precision: números de ponto flutuante e ponto flutuante de precisão dupla com precisão dependente da máquina
  - float(n): número de ponto flutuante, com precisão de pelo menos n dígitos

DDL

- Criar tabela com restrições

  create table ***r*** **(*A_1* *D_1* restrição-de-integridade$_1$,**

  ***A_2* *D_2* restrição-de-integridade$_2$,**

  ...,

  ***A_n* *D_n*,**
  restrição-de-integridade$_3$,

  ...,
  restrição-de-integridade$_k$)

DDL – Restrição chave primária

- Toda tabela deve ter uma chave primária
  - Comando primary key (colunas_chave)

```
create table agencia
        (codigo char(5),
        nome varchar(50),
        ativo int,
        cidade varchar(50),
        primary key (codigo))
```

DDL – Restrição chave primária

- Comando alternativo para chave primária simples

```
create table agencia
        (codigo char(5) primary key,
        nome varchar(50),
        ativo int,
        cidade varchar(50))
```

DDL – Restrições

- Restrições de Vazio
  - Coluna obrigatória (Not Null)

```
create table cliente
(
    matricula char(5) primary key,
    nome varchar(50) not null
)
```

DDL - Restrições

- Chave candidata
  - Unique

```
create table cliente
(
    matricula char(5) primary key,
    identidade char(7) not null,
    nome varchar(50) not null,
    unique (identidade)
)
```

DDL - Restrições

- Chave candidata
  - Unique – Versão alternativa

```
create table cliente
(
    matricula char(5) primary key,
    identidade char(7) not null unique,
    nome varchar(50) not null
)
```

DDL - Restrições

- Chave primária e candidata compostas

```
create table cliente
(
    codigoLetra char(2) not null,
    codigoNum char(5) not null,
    identidade char(7) not null,
    nome varchar(50) not null,
    sobrenome varchar(100),
    primary key(codigoLetra, codigoNum),
    unique(nome,sobrenome)
)
```

DDL – Restrições

- Restrições de domínio
  - Valor padrão (Default)

```
create table cliente
(   matricula char(5) primary key,
    identidade char(7) not null,
    nome varchar(50) not null,
    email varchar(100) default 'desconhecido'
)
```

DDL – Restrições

- Restrições de domínio
  - ckeck

```
create table cliente
(   matricula char(5) primary key,
    identidade char(7) not null,
    nome varchar(50) not null,
    idade int not null,
    check (idade > 0)
)
```

DDL – Restrições

- Restrições de domínio
  - Ckeck
  - Alguns SGBD não contemplam (MYSQL)

```
create table cliente
(   matricula char(5) primary key ,
    identidade char(7) not null,
    nome varchar(50) not null,
    idade int not null check (idade > 0)
)
```

DDL – Restrições

- Restrições de domínio
  - Ckeck
  - Enumeração

```
create table cliente
(   matricula char(5) primary key ,
    identidade char(7) not null,
    nome varchar(50) not null,
    tipo enum('Especial', 'Ouro', 'Premium')
)
```

Uma coluna do tipo enum pode ser nula. Mas, se a coluna for not null e não for preenchida, é atribuído, automaticamente, o primeiro valor da lista da enumeração.

```
create table cliente
(   matricula char(5) primary key ,
    identidade char(7) not null,
    nome varchar(50) not null,
    tipo not null enum('Especial', 'Ouro',
'Premium')
)
```

DDL – Restrições

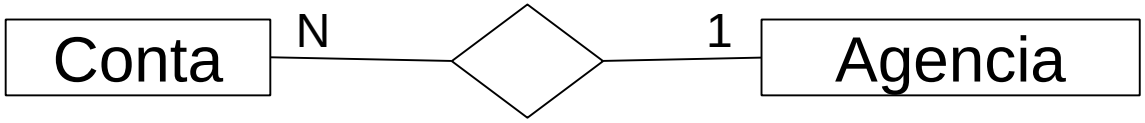
- Restrições de domínio
  - Ckeck
  - Enumeração (Postgres)

```
create table clienteT
(   matricula char(5) primary key ,
    identidade char(7) not null,
    nome varchar(50) not null,
    tipo varchar(10) not null,
    check   (tipo = 'Especial' OR tipo= 'Ouro' OR
tipo='Premium')
)
```

DDL – Restrições

- Restrições de domínio
  - ckeck
  - Tipo date no My SQL tem o formato YYYY-MM-DD

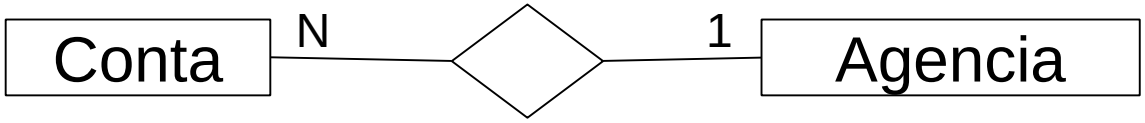
```
create table cliente
(   matricula char(5) primary key,
    nome varchar(50) not null,
    dataContratacao date,
    dataInicioGer date,
     check (dataContratacao > dataInicioGer)
)
```



DDL - Restrições

- **Integridade Referencial**
  - **Definir chave estrangeira**
  - **Cláusula foreign key**

```
create table conta
(
    numero char(5) primary key,
    saldo numeric(2,1) not null,
    nome_agencia char(15) not null,
    foreign key (nome_agencia) references agencia
(nome)
)
```

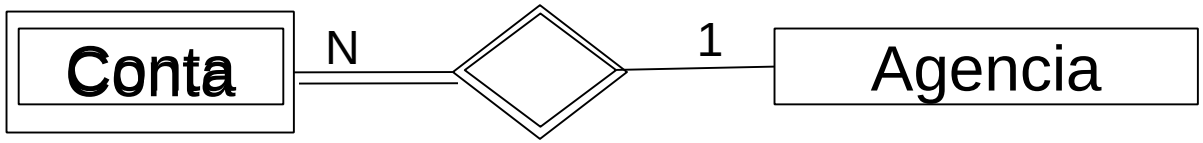


DDL - Restrições

- **Cláusula foreign key**

```
Create table agencia
(
    nome char(15) primary key,
    endereco  char(150) not null
)
```

```
Tabela conta                    CHAVE ESTRANGEIRA
(   ...
foreign key (nome_agencia) references
agencia(nome) )
```



Restrições em SQL

- **Integridade Referencial**
  - **foreign key com cascade**
  - **Delete cascade ou Update cascade**

```
Create table conta
(
    ....
foreign key (nome_agencia) references
agencia (nome) on delete cascade
)
```

DDL – Modificar tabela

- Modificar tabelas (alter table)
  - Modifica a estrutura da tabela
    - Adicionar ou remover atributo
    - Adicionar ou remover restrição
    - Alterar atributo

DDL – Modificar tabela

- Modificar tabelas (alter table)
  - **Adicionar atributo**
    - alter table nomeTabela add column nomeAtributo tipoAtributo
    - column é opcional

alter table agencia add telefone varchar(15)

DDL – Modificar tabela

- Modificar tabelas (alter table)
  - **Remover atributo**
    - alter table nomeTabela drop column nomeAtributo
    - Column é opcional

alter table agencia drop cidade

DDL – Modificar tabela

- Modificar tabelas (alter table)
- Pode variar muito em função do SGBD
  - **Alterar atributo**
    - Padrão SQL
    - alter table nomeTabela alter column nomeAtributo novaDefinição
    - Telefone era varchar(15) e deve ser alterado para varchar(30)

```
alter table agencia alter column telefone
varchar(30) not null
```

DDL – Modificar tabela

- Modificar tabelas (alter table)
  - **Alterar atributo (**MYSQL)
    - alter table nomeTabela modify column nomeAtributo novaDefinição
    - column é opcional

alter table agencia modify column telefone varchar(30) not null

DDL – Modificar tabela

- Modificar tabelas (alter table)
  - **Alterar atributo (**Postgresql)
    - Mudar o tipo do atributo
      - alter table nomeTabela alter nomeAtributo set data type novoTipo

```
alter table agencia alter column telefone
set data type varchar(30) not null
```

DDL – Modificar tabela

- Modificar tabelas (alter table)
  - **Alterar atributo (**Postgresql)
    - Criar ou remover restrição de not null
      - alter table nomeTabela alter nomeAtributo set /drop not null
      - Remover a restrição de not null do campo telefone

```
alter table agencia alter column telefone
drop not null
```

DDL – Modificar tabela

- Modificar restrições
  - Padrão SQL
  - Criar restrição

alter table nomeTabela add constraint nome da restrição RESTRIÇÃO

  - restrição pode ser de qualquer tipo
  - cláusula constraint e o nome da restrição são opcionais em alguns SGBD

alter table Conta add *constraint* conta_pkey primary key(numero)

DDL – Modificar tabela

- Modificar restrições
  - Chave canditada
    - alter table nomeTabela add constraint nomeConstraint unique(nomeColuna)
    - cláusula constraint e o nome da restrição são opcionais em alguns SGBD

DDL – Modificar tabela

- Exemplo: create table dependente (id int, nome varchar(50))

```
alter table dependente add constraint
    dependente_pkey primary key (id)
alter table dependente add constraint
    dependente_nome_unique unique
    (nome)
```

DDL – Modificar tabela

- Adicionar restrição
  - Check

```
create table Conta (
        numero char(5) primary key,
        saldo numeric(2,1) not null,
        nome_agencia char(15) not null)

    alter table Conta add constraint

    conta_saldo_check check(saldo >0)
```

DDL – Modificar tabela

- É possível executar mais de uma alteração na tabela com um alter table

alter table dependente add identidade varchar(15) unique

DDL – Modificar tabela

- Modificar restrições (apagar)
  - Padrão SQL
  - Apagar restrição

  alter table nomedaTabela drop constraint nomeRestrição

  alter table Conta drop constraint pkNum

DDL – Remover tabela

- Apagar (remover) tabela
  - drop table nomeTable
    - Remove a tabela do banco (apaga as tuplas e o esquema)

drop table agencia

Script de banco de dado

- Código necessário para executar uma operação no banco de dados
  - Alteração de esquema ou instância
- São descritos em arquivos com extensão sql
- Comentários : -- ou /* ... */

Script de banco de dado

- Criar um script a partir de um banco já existente (Postgresql)
  - Console
    - pg_dump dbname > outfile
  - Pgadmin
    - Backup
- Restaurar um banco a partir de um arquivo
  - Console
    - psql dbname < infile
  - Pgadmin
    - Restore

Exercícios – BD: empresa

- Crie um banco de dados empresa
- Criar as seguintes tabelas

departamento(codigo,nome)

funcionario(código,nome,identidade,cpf,email,coddept)

coddept referencia departamento(código)

projeto(código,nome,descrição,datainicial,datafim)

alocacao(codp,codf,datai)

codp referencia projeto (código)

codf referencia funcionário (código)

telefone(codf,numtel)

codf referencia funcionario(código)

Exercícios – BD: empresa

- Para o banco empresa
  - Os tipos das colunas devem ser definido pelos alunos(as)
  - Considere as seguintes restrições:
    - Todos os campos são obrigatórios com exceção da descrição do projeto
    - Um funcionário pode não ter telefone
    - O nome,identidade e cpf do funcionário são únicos
    - O nome do projeto e do departamento também são únicos
    - A data fim (datafim) do projeto deve ser maior que a data inicial (datainicial)

Exercícios – BD: empresa

- Para o banco empresa
  - Altere o esquema do banco de maneira que seja possível armazenar
    - A quantidade de horas que um funcionário trabalho em um projeto (obrigatório com valor >0)
    - Custo estimado de um projeto (valor obrigatório)
    - Gasto real de um projeto (não obrigatório)

Exercícios – BD: academico

- Crie um banco de dados academico
- Criar as seguintes tabelas

Aluno(mat, cpf, identidade, nome, email)

Professor(codigo,nome,cpf,email)

Disciplina(codigo,nome,ementa)

Turma(codigo,nome,disciplina,professor)

disciplina referencia Disciplina(codigo)

Professor referencia Professor(codigo)

Crie o banco de dados Empresa2

EMPREGADO

ENOME	SSN	DATANASC	ENDERECO	DNUMERO

chave primária (p.k.)

DEPARTAMENTO

DNOME	DNUMERO	DGERSSN

DEPT_LOCALIZACOES

DNUMERO	DLOCALIZACAO

PROJETO

PNOME	PNUMERO	PLOCALIZACAO	DNUM

TRABALHA_EM

SSN	PNUMERO	HORAS

Considere as seguintes restrições (além das restrições de integridade referencial):

- Todos as colunas sao obrigatórias (exceto DATANASC e DGERSSN)
- ENOME é chave candidata
- DATANASC deve ser maior que 01/01/1900
- HORAS > 0

Os campos com setas são chaves estrangeiras

Use letras minúsculas em todos os identificadores (nomes de tabelas, colunas e restrições)

EMPREGADO

ENOME	SSN	DATANASC	ENDERECO	DNUMERO

chave primária (p.k.)

DEPARTAMENTO

DNOME	DNUMERO	DGERSSN

Remover coluna enome de empregado

DEPT_LOCALIZACOES

DNUMERO	DLOCALIZACAO

Adicionar colunas pnome, mnome, unome em empregado
Todas as colunas são obrigatórias e cadeias de caracteres de tamanho variável, podendo atingir até 200 caracteres

PROJETO

PNOME	PNUMERO	PLOCALIZACAO	DNUM

TRABALHA_EM

SSN	PNUMERO	HORAS