



Universidade Federal de Mato Grosso-UFMT
Sistemas de Informação
Laboratório de Banco de Dados
Prof. Dr. Clóvis Júnior

Regras de Integridade

(Constraints)



Constraints

- Regras para impor restrições em nível de tabelas.
- Previnem exclusão de dados caso haja dependências.
- Tipos de constraint:
 - NOT NULL
 - UNIQUE
 - PRIMARY KEY
 - FOREIGN KEY
 - CHECK

Constraints – Abreviações Populares

- Primary Key pk
- Foreign Key fk
- Unique uk
- Check ck
- Not Null nn

Definindo Constraints

- Podem ser criadas em conjunto com a tabela, ou adicionadas posteriormente
- São criadas em dois níveis:
 - Column level.
 - Table level.

Tabelas - Regras e Convenções

- São objetos armazenados no banco de dados;
- A criação de uma tabela consiste:
 1. Nome da da tabela,
 2. Nomes para cada coluna,
 3. Tipo para cada coluna,
 4. Tamanho para as colunas (quando necessário).

Tabelas - Regras e Convenções

- Diferentes tipos de Constraints:
 - Especificação de chave primária,
 - Definição de chave estrangeira,
 - Regras para validação de dados em colunas,
 - Valores nulos (NULL) permissíveis,
 - Especificação de valores únicos.

Criação de Tabelas

```
CREATE TABLE [schema.] tablename  
  
(column1 datatype [CONSTRAINT constraint_name] constraint_type . . .,  
  
(column2 datatype [CONSTRAINT constraint_name] constraint_type,  
  
[CONSTRAINT constraint_name] constraint_type (column, . . . ), . . . );
```

Criação de Tabelas

```
CREATE TABLE Estados
```

```
( CodigoEstado NUMBER (4),
```

```
NomeEstado VARCHAR2 (30) ) ;
```

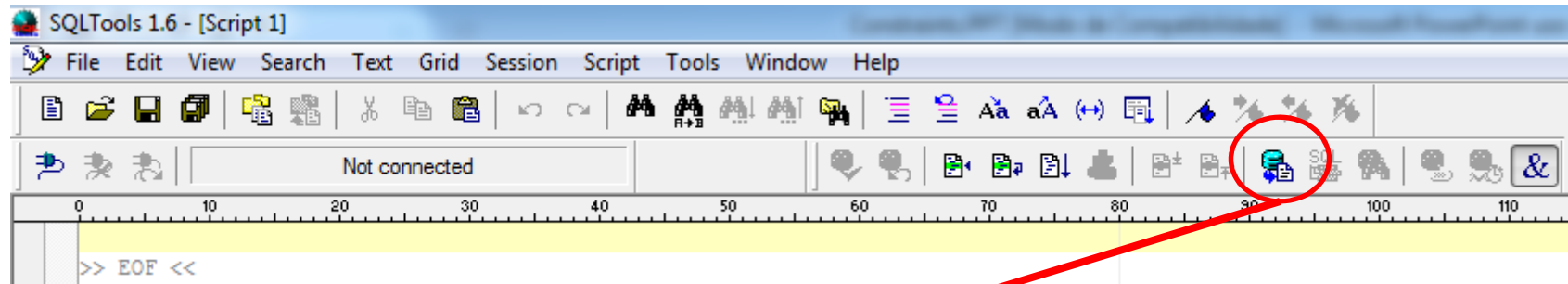

Visualização da Estrutura de Tabelas

- SQL*Plus -> **DESCRIBE**

SQL> DESCRIBE Estados;

SQL> DESC Estados;

Visualização da Estrutura de Tabelas



Objects List (Alt+3)

AMBIENTAL

Types Type Bodies Sequences Views Synonyms DB Links Clusters Grantee Recyclebin

Tables PKs UKs FKs CHks Indexes Triggers Procedures Packages Package Bodies

Name	Type	Tablespace	Initial	Next	Increase	Max Ext	Free	Used
Insira o texto aqui	Insira o ...	Insira o ...	Insira o ...	Insira o ...	Insira o ...	Insira o ...	Insira o ...	Insira o ...
AREAS_CAUSAS	REGULAR	SYSTEM	65536	0	0	2147483645	10	40
AREAS_DEGRADACOES	REGULAR	SYSTEM	65536	0	0	2147483645	10	40
AREAS_DEGRADADAS	REGULAR	SYSTEM	65536	0	0	2147483645	10	40
AREAS_EFEITOS	REGULAR	SYSTEM	65536	0	0	2147483645	10	40
BACIAHIDROGRAFICA	REGULAR	SYSTEM	65536	0	0	2147483645	10	40
CIDADES	REGULAR	SYSTEM	65536	0	0	2147483645	10	40
CLIMA	REGULAR	SYSTEM	65536	0	0	2147483645	10	40
ESTADOS	REGULAR	SYSTEM	65536	0	0	2147483645	10	40
FUNCOES_RH	REGULAR	SYSTEM	65536	0	0	2147483645	10	40
INTERVENCOES_AMEACAS	REGULAR	SYSTEM	65536	0	0	2147483645	10	40
INTERVENCOES_PRESSOES	REGULAR	SYSTEM	65536	0	0	2147483645	10	40
MATERIAIS_CONSUMO	REGULAR	SYSTEM	65536	0	0	2147483645	10	40
MATERIAIS_PERMANENTE	REGULAR	SYSTEM	65536	0	0	2147483645	10	40
ORCAMENTO_AMEACA_CONS...	REGULAR	SYSTEM	65536	0	0	2147483645	10	40
ORCAMENTO_AMEACA_MAO...	REGULAR	SYSTEM	65536	0	0	2147483645	10	40
ORCAMENTO_AMEACA_PERM...	REGULAR	SYSTEM	65536	0	0	2147483645	10	40
ORCAMENTO_PRESSOES_CON...	REGULAR	SYSTEM	65536	0	0	2147483645	10	40
ORCAMENTO_PRESSOES_MAO...	REGULAR	SYSTEM	65536	0	0	2147483645	10	40
ORCAMENTO_PRESSOES_PER...	REGULAR	SYSTEM	65536	0	0	2147483645	10	40

Selected 1 of 31

Tabelas – Adicionando Novas Colunas

- Sintaxe geral:

ALTER TABLE nome-da-tabela ADD nome-da-coluna datatype;

```
SQL> ALTER TABLE Estados ADD Sigla Varchar2(2) ;  
Table altered.  
SQL>
```

Tabelas – Modificando Colunas

- Sintaxe Geral:

**ALTER TABLE nome-da-tabela MODIFY nome-da-coluna
newdatatype;**

newdatatype representa o novo tipo ou alteração de tamanho da coluna.

```
SQL> ALTER TABLE Estados MODIFY  
      Sigla Varchar2(2) ;  
Table altered.  
SQL>
```

Tabelas – Excluindo Colunas

- Sintaxe geral:

ALTER TABLE nome-da-tabela DROP COLUMN nome-da-coluna;

Excluindo Tabelas

- Sintaxe geral:
DROP TABLE nome-da-tabela [CASCADE CONSTRAINTS];
- For example,
DROP TABLE Estados;

Table dropped

- SQL>

Criação de Constraints

- Nomes de constraints seguem as iniciais `SYS_Cn`.
- Quando são criadas as constraints:
 - Ao mesmo tempo que as tabelas são criadas
 - Depois que as tabelas são criadas
- Podem ser definidas a nível de colunas.

Constraints – Coluna

- Aplicado a uma única coluna;
- Constraints podem ser definidas em qualquer coluna exceto FOREIGN KEY e composições de chave primária.

Column datatype [CONSTRAINT constraint_name] constraint_type

Example:

Nome VARCHAR2(7) CONSTRAINT Regra_Nome NOT NULL

Constraints – Tabela

- São aplicadas a uma ou mais colunas e criadas separadamente das definições das colunas;
- Usualmente são definidas após a criação das colunas;
- All constraints can be defined at the table level except for the NOT NULL constraint.

[CONSTRAINT constraint_name] constraint_typ (Column, . . .),

Exemplo:

CONSTRAINT ChavePrimaria_pk PRIMARY KEY(Matricula)

Constraints – Chave Primária

- Também conhecida como **entity integrity constraint**;

- Nível de coluna:

Matricula NUMBER (2) CONSTRAINT Matricula_pk PRIMARY KEY,

- Nível de tabela:

CONSTRAINT Matricula_pk PRIMARY KEY(Matricula),

Constraints – Chave Estangeira

Constraint FOREIGN KEY

- FOREIGN KEY também chamada de **referential integrity constraint**.
- Define uma coluna como referência a outra tabela, estabelecendo relacionamentos.
- Pode ser definida para definir auto-relacionamento;

Constraints – Chave Estangeira

- Não é obrigatório o mesmo nome entre as colunas, mas é preciso ter o mesmo conjunto de valores;

```
CONSTRAINT Curso_fk FOREIGN KEY(CodigoCurso)  
REFERENCES Cursos (CodigoCurso),
```

Constraints – Nulos – NOT NULL

- Aplicado a colunas com obrigatoriedade de preenchimento;
- Espaços ou zeros são valores válidos;
- Somente pode ser definida a nível de colunas:

Nome VARCHAR2(40) CONSTRAINT Nome_Aluno_nn NOT NULL,

Constraints – Valores Únicos

- Determina valores únicos para determinada coluna;

- Definição a nível de tabela:

CONSTRAINT NomeCurso_uk UNIQUE(NomeCurso),

- Definição a nível de coluna:

NomeCurso VARCHAR2(25) CONSTRAINT NomeCurso_uk UNIQUE,

Constraints – CHECK

- Define um conjunto de valores para uma coluna específica;
- Definição a nível de coluna:
**CodigoCurso NUMBER(2) CONSTRAINT CodigoCurso_cc
CHECK((CodigoCurso >= 10) and (CodigoCurso <= 99)),**
- Definição a nível de tabela:
**CONSTRAINT CodigoCurso_cc
CHECK((CodigoCurso >= 10) and (CodigoCurso <= 99)),**

Constraint NOT NULL

Definição em nível de coluna

```
CREATE TABLE Estados  
  
  (CodigoEstado NUMBER (4) NOT NULL ,  
  
   NomeEstado VARCHAR2 (30) ) ;
```


Constraint PRIMARY KEY

Definição em nível de tabela ou coluna

```
CREATE TABLE Estados
```

```
( CodigoEstado NUMBER (4) NOT NULL ,
```

```
NomeEstado VARCHAR2 (30) ,
```

```
ADD CONSTRAINT ChavePrimaria_PK
```

```
PRIMARY KEY ( CodigoEstado ) );
```

Constraint FOREIGN KEY

Definição em nível de tabela ou coluna.

```
CREATE TABLE Cidades
```

```
( CodigoCidade NUMBER (7) NOT NULL ,
```

```
NomeCidade VARCHAR2 (40) ,
```

```
CodigoEstado NUMBER (4) NOT NULL,
```

```
primary key(CodigoCidade));
```

```
Alter table Cidades
```

```
ADD CONSTRAINT Cidades_Estados_FK FOREIGN KEY
```

```
( CodigoEstado) REFERENCES Estados
```

```
( CodigoEstado);
```

Constraint CHECK

- Define condições para cada linha da tabela
- Não é utilizada em expressões com:
 - CURRVAL, NEXTVAL e ROWNUM ou outras pseudocolunas
 - Chamadas a SYSDATE, USER e USERENV funções equivalentes
 - Consultas referenciando valores em outras linhas da tabela

```
CREATE TABLE Clientes
```

```
( Codigo NUMBER (8) PRIMARY KEY ,
```

```
Nome VARCHAR2 (40) ,
```

```
Salario NUMBER (12,3), CONSTRAINT MinimoSalario_ck
```

```
CHECK (Salario>0));
```

Eliminando Constraints

- Removendo constraint CHECK

```
ALTER TABLE Clientes DROP  
CONSTRAINT MinimoSalario_ck;
```

- Removendo chave primária

```
ALTER TABLE Cidades DROP PRIMARY KEY CASCADE;
```

Desabilitando Constraints

- Não elimina a constraint.
- A opção CASCADE desabilita a integridade de dependentes.

```
ALTER TABLE clientes DISABLE CONSTRAINT  
ChavePrimaria_pk CASCADE;
```

Habilitando Constraints

```
ALTER TABLE clientes ENABLE CONSTRAINT  
ChavePrimaria_pk CASCADE;
```

Exercícios: Criação de modelo

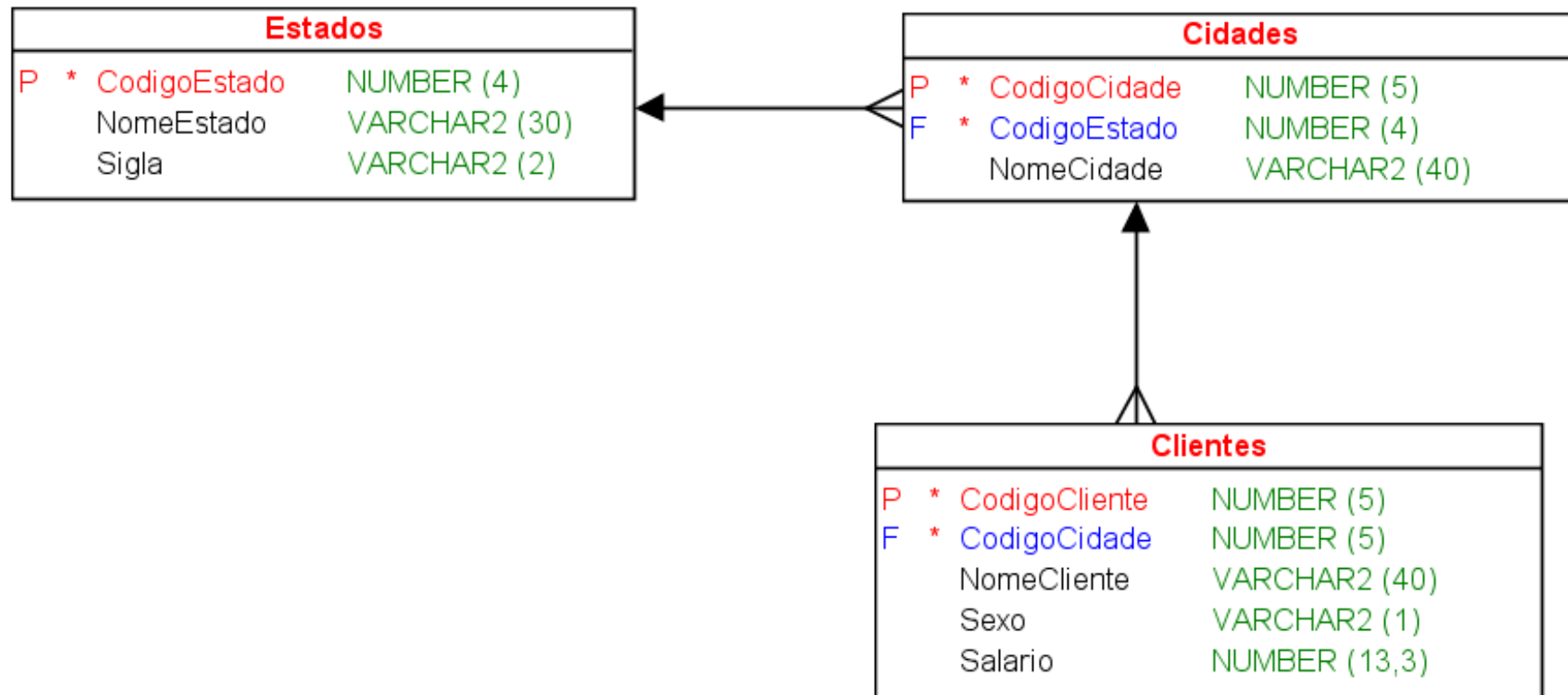
Usuário: CadPrincipal.

CREATE USER CadPrincipal IDENTIFIED BY abc

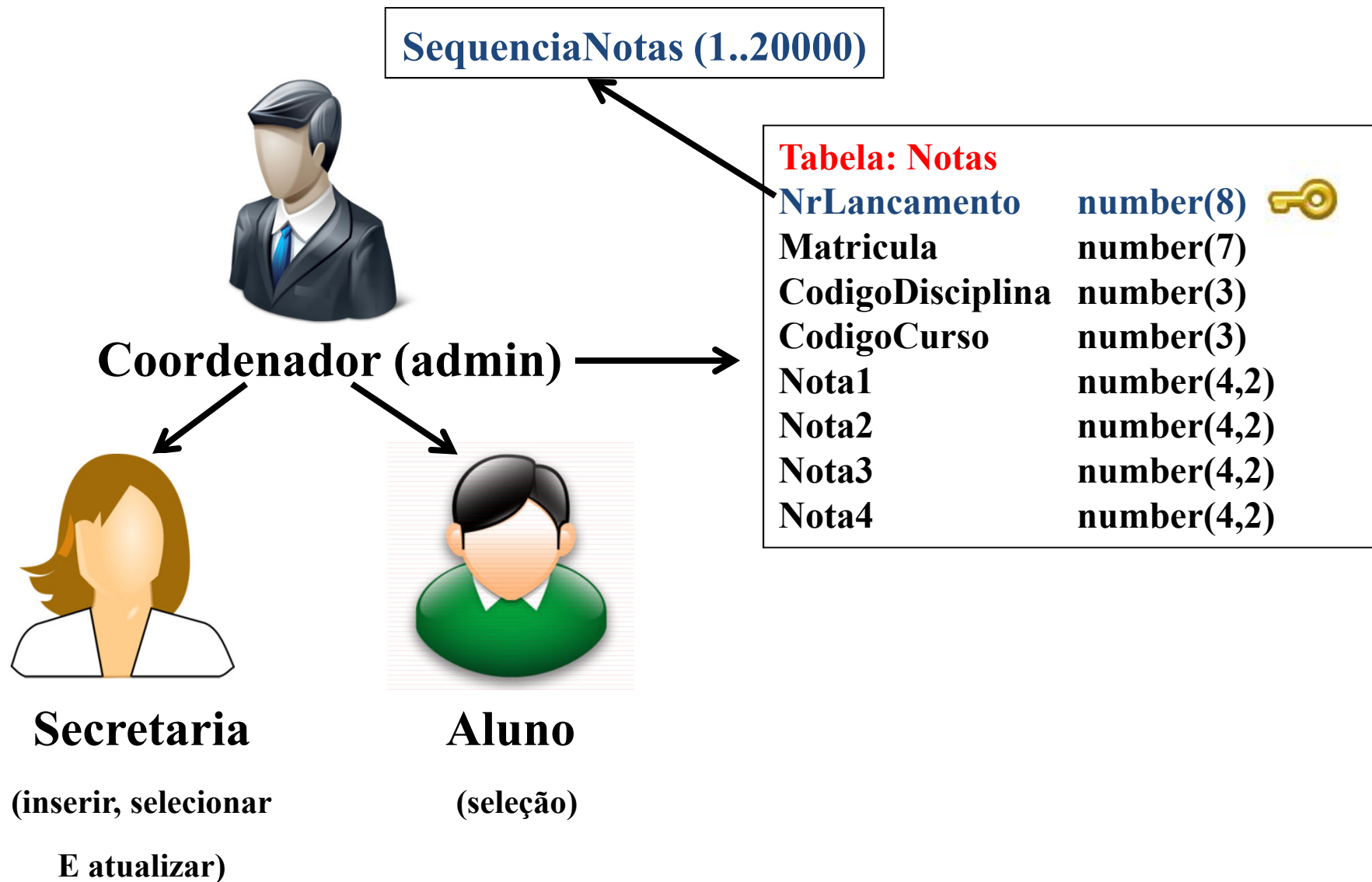
DEFAULT TABLESPACE Dados QUOTA

UNLIMITED ON Dados;

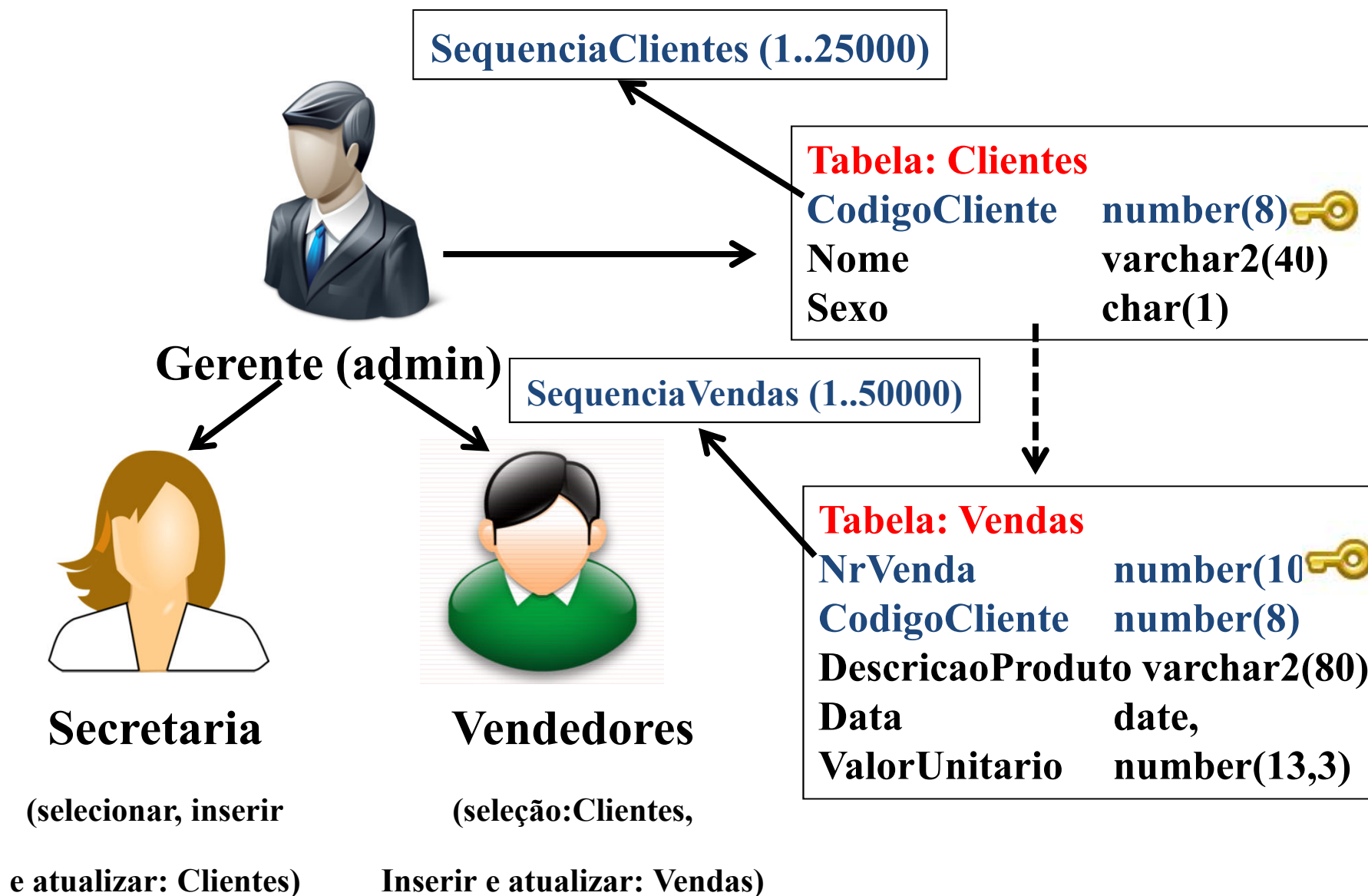
GRANT DBA TO CadPrincipal;



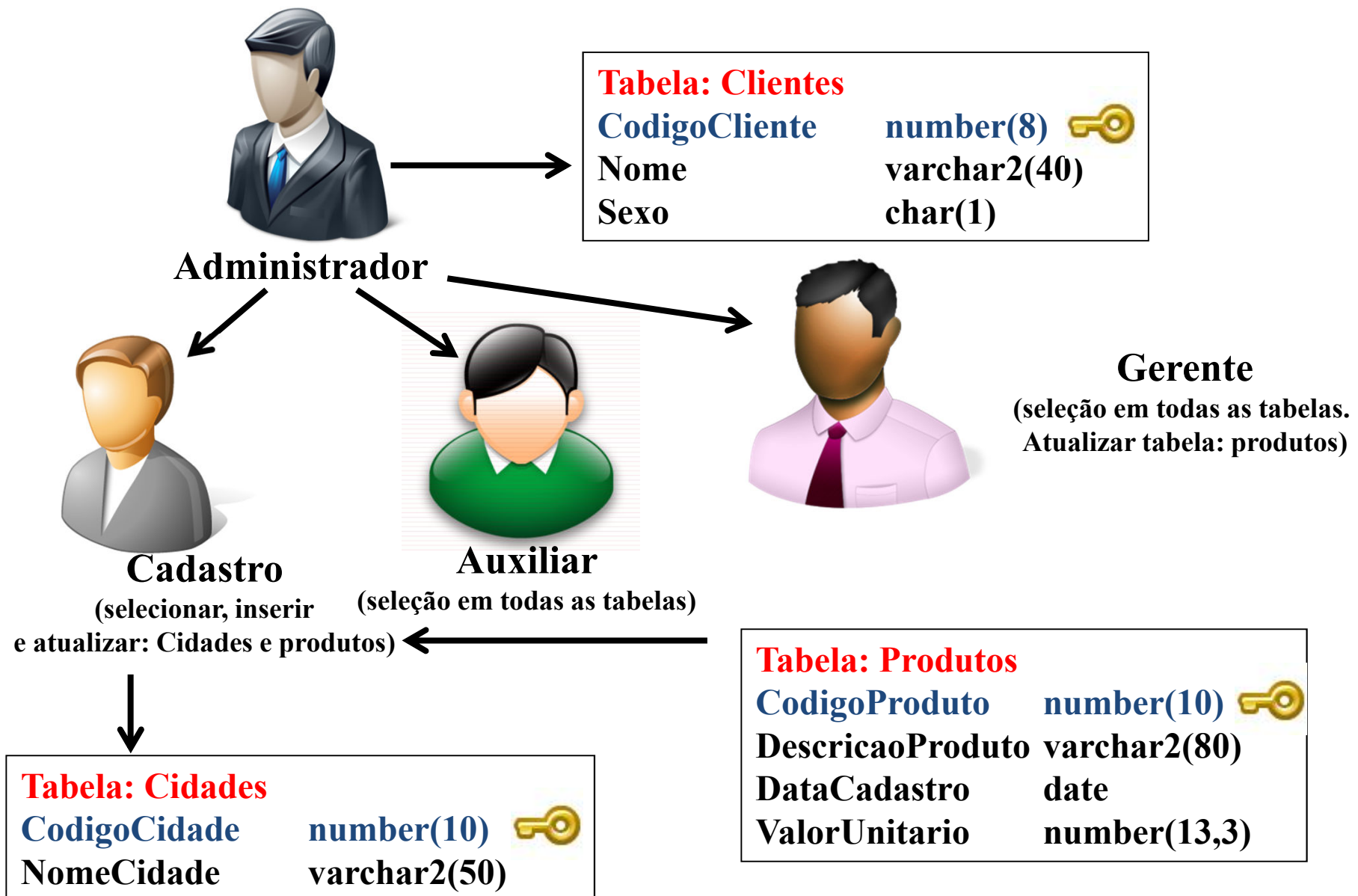
Exercícios: Privilégios para Usuários - Notas



Exercícios: Privilégios para Usuários - Vendas



Exercícios: Privilégios para Usuários - Estoque



Exercícios: Privilégios para Usuários - Universidade

