

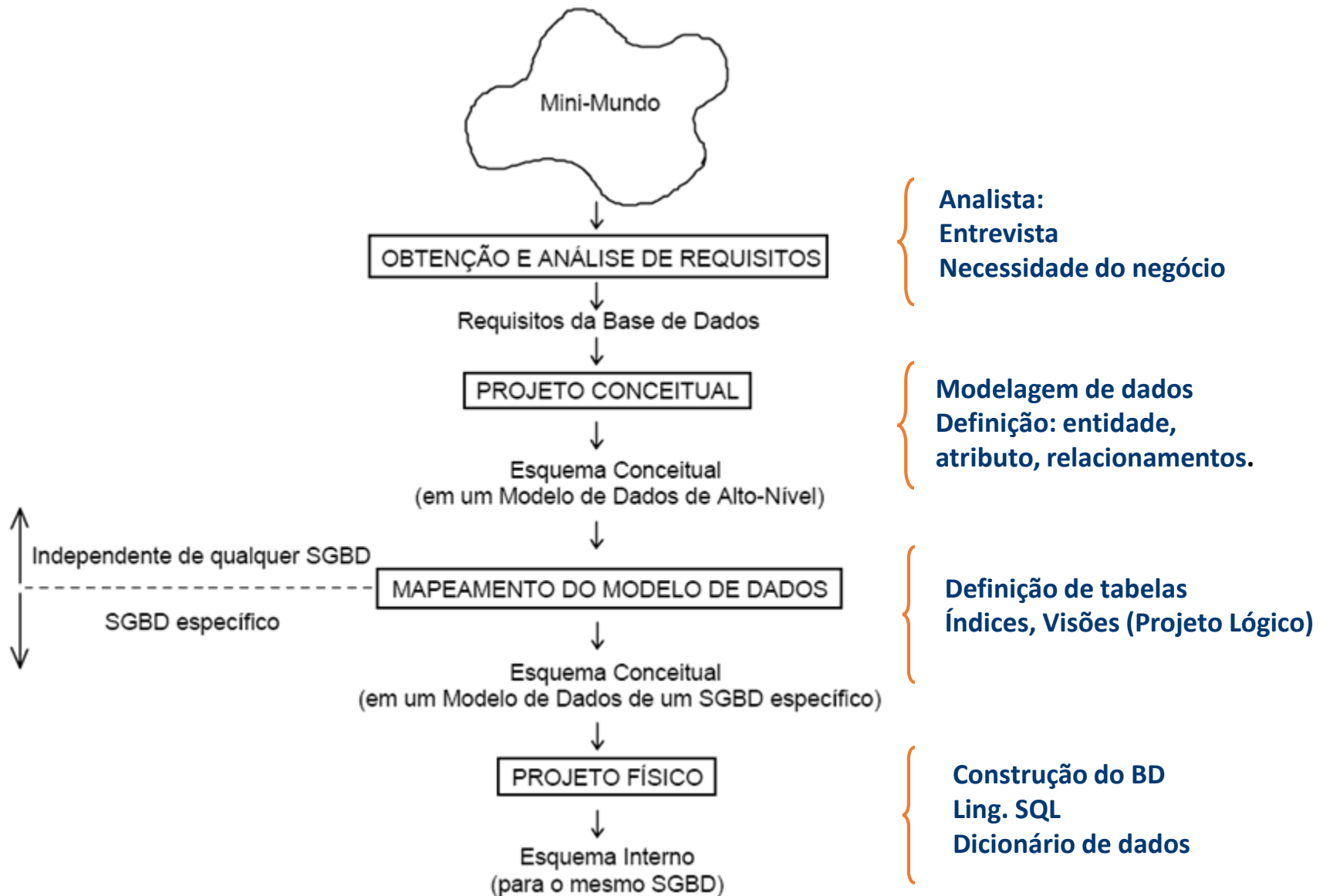


Estrutura e Banco de Dados

Projeto Físico

(Criando Tabelas e Estrutura de Dados)

Prof. Claudiney



Introdução

Tipos de dados

Criação de tabelas

Exemplo

Verificando a estrutura de uma tabela

Constraint

Primary key

Foreign Key

Not null

Check

Padrões

Exercício

Referências Bibliográficas

Introdução

Para as próximas aulas, iremos utilizar o software de banco de dados Oracle.

O Oracle Database 10g Express Edition (Oracle Database XE) é uma versão básica simplificada do banco de dados Oracle.

Preparei um tutorial a parte para que vocês possam realizar o “download” e instalar a ferramenta no qual iremos utilizar.

Acesse o tutorial “**Instalando o Oracle**”, instale o aplicativo para podermos praticar nossas próximas aulas.

Introdução

Nesta etapa é que detalhamos o modo de armazenamento interno [tamanho dos campos, índices, tipo de preenchimento dos campos, nomenclaturas, etc] através de uma linguagem denominada SQL.

Linguagens e notações variam de acordo com o produto (SGBD)

SQL significa “Structured Query Language”

(Linguagem Estruturada de Consulta);

A SQL foi desenvolvida originalmente na IBM Research no início da década de 1970;

Representa o padrão para linguagens de SGBD Relacionais, sendo padronizada pelo comitê ANSI/ISO.

Em 1999 foi publicado o atual padrão SQL/99 ou SQL3;

Introdução

Quando os Bancos de Dados Relacionais estavam sendo desenvolvidos, foram criadas linguagens destinadas à sua manipulação. O Departamento de Pesquisas da IBM, desenvolveu a SQL como forma de interface para o sistema de BD relacional denominado SYSTEM R, início dos anos 70.

Em 1986 o American National Standard Institute (ANSI), publicou um padrão SQL.

A SQL estabeleceu-se como linguagem padrão de Banco de Dados Relacional.

Introdução

SQL apresenta uma série de comandos que permitem a definição dos dados, chamada de DDL (Data Definition Language), composta entre outros pelos comandos Create, que é destinado a criação do Banco de Dados, das Tabelas que o compõe, além das relações existentes entre as tabelas. Como exemplo de comandos da classe DDL temos os comandos Create, Alter e Drop.

Introdução

Os comandos da série DML (Data Manipulation Language), destinados a consultas, inserções, exclusões e alterações em um ou mais registros de uma ou mais tabelas de maneira simultânea.

Como exemplo de comandos da classe DML temos os comandos Select, Insert, Update e Delete.

Uma subclasse de comandos DML, a DCL (Data Control Language), dispõe de comandos de controle como Grant e Revoke.

Introdução

A Linguagem SQL tem como grandes virtudes sua capacidade de gerenciar índices, sem a necessidade de controle individualizado de índice corrente, algo muito comum nas linguagens de manipulação de dados do tipo registro a registro. Outra característica muito importante disponível em SQL é sua capacidade de construção de visões, que são formas de visualizarmos os dados na forma de listagens independente das tabelas e organização lógica dos dados.

Introdução

Outra característica interessante na linguagem SQL é a capacidade que dispomos de cancelar uma série de atualizações ou de as gravarmos, depois de iniciarmos uma sequência de atualizações. Os comandos Commit e Rollback são responsáveis por estas facilidades.

Devemos notar que a linguagem SQL consegue implementar estas soluções, somente pelo fato de estar baseada em Banco de Dados, que garantem por si mesmo a integridade das relações existentes entre as tabelas e seus índices.

Regras de Nomeação

Os nomes das tabelas e colunas devem conter de 1-30 caracteres, sendo o 1º caractere alfabético.

Os nomes devem conter apenas caracteres de a-z, A-Z, 0-9, _, \$ e #.

Os nomes não podem ser iguais as palavras reservadas do Oracle.

Tipos de Dados

Ao se criar a estrutura de uma tabela é necessário que o usuário forneça, para cada coluna as seguintes informações:

Tipo de Dado

Tamanho

Restrições

| Tipo de Dado | Descrição |
|--------------|--|
| CHAR(n) | Campo fixo com tamanho máximo de 2000 bytes. |
| DATE | Permite data entre 1 de janeiro de 4712 AC até 31 de dezembro de 4712 DC |
| LONG | Caractere variável com tamanho de até 2 Gb |
| VARCHAR2(n) | Campo do tipo caractere com tamanho variável e limitado a 4000 bytes. |
| NUMBER(n,d) | Onde n é o número de dígitos e d o número de casas decimais. |

Restrições - CONSTRAINT

As restrições são regras básicas estabelecidas para o preenchimento de uma ou mais colunas da tabela, e são definidas ao final da especificação de cada coluna ou ao final do comando.

Entre as restrições encontram-se:

- Chaves primárias;

- Chaves únicas;

- Chaves estrangeiras;

- Identificadores de campos obrigatórios;

- Condições para valores permitidos para determinado campo.

Constraints de integridade de dados

| CONSTRAINT | DESCRIÇÃO |
|------------------------|---|
| NOT NULL | Especifica que esta coluna não pode conter valores nulos. |
| UNIQUE | Especifica uma coluna ou combinação de colunas que terão seus valores únicos na tabela. |
| PRIMARY KEY | Identifica a unicidade de cada linha na tabela. |
| FOREIGN KEY REFERENCES | Estabelece um relacionamento entre a chave estrangeira e a chave primária da tabela referenciada. |
| CHECK | Especifica uma condição que deve ser verdadeira obedecendo uma regra do negócio. |

Criação de Tabela

```
CREATE TABLE nome_tabela  
( nome_coluna tipo de dado | constraint_tabela ),  
( nome_coluna tipo de dado | constraint_tabela )
```

Onde:

Nome_tabela → é o nome da tabela

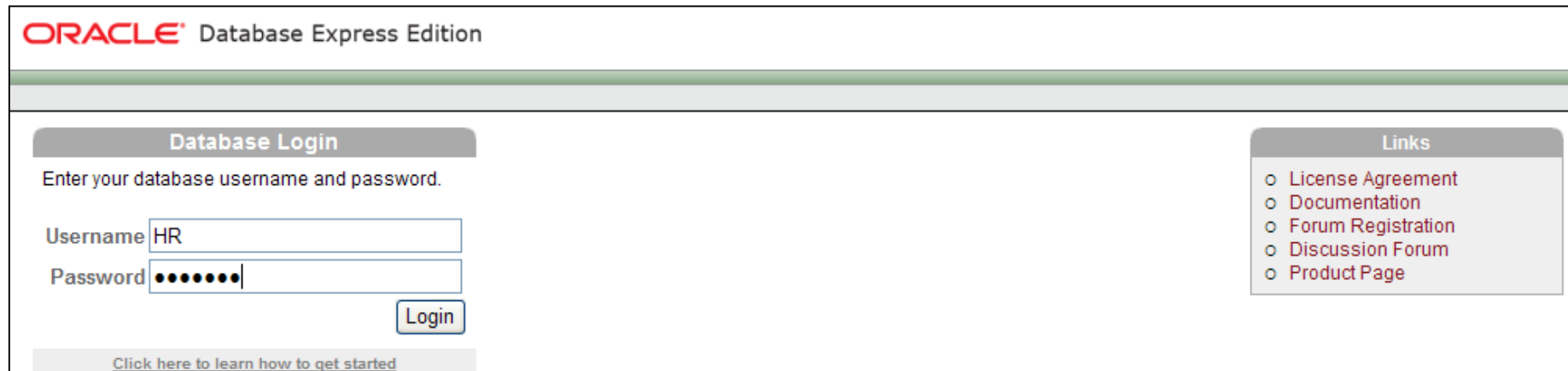
Nome_coluna → é o nome da coluna

Tipo de dado → é o tipo de dado da coluna

Constraint_tabela → é a “**constraint**” ou restrição para a coluna.

Pré requisito para este material

Se logar com o usuário HR e Selecionar SQL Commands



ORACLE® Database Express Edition

Database Login

Enter your database username and password.

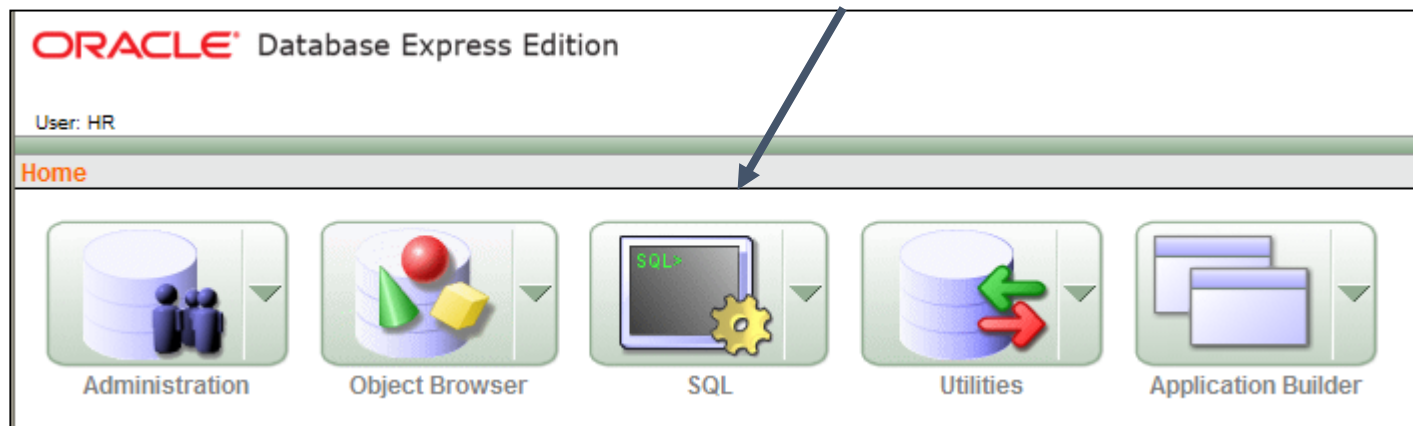
Username

Password

[Click here to learn how to get started](#)

Links

- [License Agreement](#)
- [Documentation](#)
- [Forum Registration](#)
- [Discussion Forum](#)
- [Product Page](#)



Exemplo

Nome da
tabela

Atributos

```
CREATE TABLE Fornecedor
( Cd_Fornecedor      Number (4),
  Nm_Fornecedor      Varchar2 (50),
  Ds_Endereco        Varchar2 (70),
  Nr_Fone             Number (12),
  Dt_Nascimento       Date,
  le_Sexo             Char (1),
  le_Fisica_Juridica  Char (1),
  Cd_Situacao         Number (3)
)
```

OBS.: Copiem e cole o comando acima no SQL Commands do Oracle

Exemplo

The screenshot displays the Oracle Database Express Edition SQL Commands interface within a Windows Internet Explorer browser. The browser's address bar shows the URL `http://127.0.0.1:8080/apex/f?p=4500:1003:613754123091805::NO::`. The Oracle interface includes a top navigation bar with links for Home, Logout, and Help. Below this, the user is identified as 'HR'. The main content area is titled 'SQL Commands' and features a 'Save' button and a 'Run' button. A yellow arrow points to the 'Run' button. The SQL command entered is:

```
CREATE TABLE Fornecedor
(
  Cd_Fornecedor      Number (4),
  Nm_Fornecedor      Varchar2 (50),
  Ds_Endereco        Varchar2 (70),
  Nr_Fone             Number (12),
  Dt_Nascimento       Date,
  Ie_Sexo             Char (1),
  Ie_Fisica_Juridica  Char (1),
  Cd_Situacao         Number (3)
)
```

Below the command area, the 'Results' tab is selected, showing the message 'Table created.' and the execution time '0.02 seconds'. The bottom status bar indicates 'Application Express 2.1.0.00.39' and 'Copyright © 1999, 2006, Oracle. All rights reserved.' The Windows taskbar at the bottom shows the 'Iniciar' button and several open applications, including '7 - Aula 7 - Projeto F...', 'SQL Commands - Win...', and '7 - Projetos Físico - C...'. The system clock shows the time as 23:54.

Trabalhando com Valores Default

Se após o tipo e tamanho do campo for colocada a palavra **DEFAULT** seguida de uma expressão, esta representará um valor padrão.

```
CREATE TABLE NOTA_FISCAL  
(  
...  
...  
Dt_Emissao Date Default Sysdate,  
...)
```

Verificando a Estrutura da Tabela

Para verificar a estrutura de uma tabela, utilizamos o comando desc.

OBSERVAÇÃO: Se logar no Oracle com o usuário HR e selecionar SQL Commands. (Veja tutorial de instalação do Oracle).

Exemplo: desc employees;

Results Explain Describe Saved SQL History

Object Type TABLE Object EMPLOYEES

| Table | Column | Data Type | Length | Precision | Scale | Primary Key | Nullable | Default | Comment |
|-----------|----------------|-----------|--------|-----------|-------|-------------|----------|---------|---|
| EMPLOYEES | EMPLOYEE_ID | Number | - | 6 | 0 | 1 | - | - | Primary key of employees table. |
| | FIRST_NAME | Varchar2 | 20 | - | - | - | ✓ | - | First name of the employee. A not null column. |
| | LAST_NAME | Varchar2 | 25 | - | - | - | - | - | Last name of the employee. A not null column. |
| | EMAIL | Varchar2 | 25 | - | - | - | - | - | Email id of the employee |
| | PHONE_NUMBER | Varchar2 | 20 | - | - | - | ✓ | - | Phone number of the employee; includes country code and area code |
| | HIRE_DATE | Date | 7 | - | - | - | - | - | Date when the employee started on this job. A not null column. |
| | JOB_ID | Varchar2 | 10 | - | - | - | - | - | Current job of the employee; foreign key to job_id column of the jobs table. A not null column. |
| | SALARY | Number | - | 8 | 2 | - | ✓ | - | Monthly salary of the employee. Must be greater than zero (enforced by constraint emp_salary_min) |
| | COMMISSION_PCT | Number | - | 2 | 2 | - | ✓ | - | Commission percentage of the employee; Only employees in sales department eligible for commission percentage |
| | MANAGER_ID | Number | - | 6 | 0 | - | ✓ | - | Manager id of the employee; has same domain as manager_id in departments table. Foreign key to employee_id column of employees table. (useful for reflexive joins and CONNECT BY query) |
| | DEPARTMENT_ID | Number | - | 4 | 0 | - | ✓ | - | Department id where employee works; foreign key to department_id column of the departments table |

1 - 11

Constraint Primary Key

Especifica uma ou mais colunas que compõem a chave primária de uma tabela.

```
CREATE TABLE Cliente
(cd_cliente          number (4) primary key,
 nm_cliente          varchar2 (50),
 ds_endereco         varchar2 (70),
 cd_municipio        number (5),
 sg_estado           char (2),
 nr_cep              varchar2 (8),
 nr_ddd              number (3),
 nr_fone             number(7),
 ie_sexo             char(1)
)
```

OBS: a expressão primary key declarada ao lado do campo, somente é válida para chaves primárias simples

Constraint Primary Key

OBSERVAÇÃO: Caso queira apagar a tabela para gerar novamente utilize o comando DROP TABLE Cliente

```
CREATE TABLE Cliente
(cd_cliente          number (4)
  constraint cliente_cd_cliente_pk primary key,
 nm_cliente          varchar2 (50),
 ds_endereco         varchar2 (70),
 cd_municipio        number (5),
 sg_estado           char (2),
 nr_cep              varchar2 (8),
 nr_ddd              number (3),
 nr_fone             number(7),
 ie_sexo             char(1)
)
```

Padronizando a restrição, seguiremos o seguinte padrão:

Constraint=> nome_tabela_nome_campo_tipodaconstraint

Constraint Primary Key

Uma forma mais organizada, onde após definirmos todos os campos da tabela definimos as restrições

```
CREATE TABLE Cliente
(cd_cliente          number (4),
 nm_cliente          varchar2 (50),
 ds_endereco         varchar2 (70),
 cd_municipio        number (5),
 sg_estado           char (2),
 nr_cep              varchar2 (8),
 nr_ddd              number (3),
 nr_fone             number(7),
 ie_sexo             char(1),
    constraint cliente_cd_cliente_pk primary key (cd_cliente)
)
```

Constraint Primary Key Composta

```
CREATE TABLE Historico
(cd_cliente          number (4),
 dt_compra          date,
 vl_compra          number (12,2),
 CONSTRAINT Historico_PK PRIMARY KEY (cd_cliente, Dt_Compra)
)
```

Aqui é apresentado um exemplo de uma PK (Primary Key) composta por 2 atributos: cd_cliente e dt_compra.

Constraint UNIQUE

```
CREATE TABLE Estado  
(Sg_Estado          char(2) primary key,  
  Nm_Estado         varchar2 (35),  
   constraint Estado_nm_Estado_UN UNIQUE (nm_Estado)  
)
```

Define uma ou mais colunas que não podem ter valor repetido em mais de uma linha da tabela.

Por exemplo, não existem duas pessoas com o mesmo CPF ou número do PIS , mas estes campos não serão colocados como chave primária.

Constraint FOREIGN KEY

Referencia um atributo que é chave primária de outra tabela com o propósito de implementar o relacionamento entre tabelas

```
CREATE TABLE Cliente
```

```
(cd_cliente          number (4),  
  nm_cliente         varchar2 (50),  
  ds_endereco        varchar2 (70),  
  cd_municipio        number (5),  
  sg_estado          char (2),  
  nr_cep              varchar2 (8),  
  nr_ddd              number (3),  
  nr_fone             number(7),  
  ie_sexo             char(1),
```

```
    constraint cliente_sg_estado_fk foreign key (sg_estado) references  
      Estado(sg_estado)
```

```
)
```

Constraint FOREIGN KEY

Neste tipo de constraint, relacionamentos que utilizem mais de uma coluna (chave composta) podem ser criados.

Para definição desta constraint utilizarmos o padrão

[nome_tabela/atributo/tipo_constraint]

REGRAS:

- Caso o tipo de dados da coluna na tabela inicial e na tabela referenciada sejam diferentes, será apresentado um erro;

- Caso a tabela referenciada não possua chave primária (a foreign key será estabelecida sobre a chave primária da tabela referenciada);

- O uso de chaves estrangeiras garante que não existirão linhas órfãs nas tabelas-filhas (tabelas que possuem dados que devem estar cadastrado previamente em outra tabela, denominada tabela mãe).

Constraint CHECK

Define um conjunto de valores permitidos ou condição para inserção de valores em uma determinada coluna.

```
CREATE TABLE Cliente
(cd_cliente          number (4),
 nm_cliente          varchar2 (50),
 ds_endereco        varchar2 (70),
 cd_municipio        number (5),
 sg_estado           char (2),
 nr_cep              varchar2 (8),
 nr_ddd              number (3),
 nr_fone             number(7),
 ie_sexo             char(1),
    constraint cliente_ie_sexo_ck
    check(ie_sexo in ('F', 'M'))
)
```

```
insert into cliente
values(1,'rafael','rua',555
55,'sp','08290370',011,
1234567,'f');
```

```
insert into cliente
values(1,'rafael','rua',555
55,'sp','08290370',011,
1234567,'F');
```

Constraint NOT NULL

Indica que **é obrigatória a inserção de algum valor** nessa coluna. Somente **pode ser declarado junto à coluna** e não recebe nome da restrição.

```
CREATE TABLE Cliente
```

```
(cd_cliente          number (4),  
  nm_cliente        varchar2 (50) not null,  
  ds_endereco       varchar2 (70) not null,  
  cd_municipio      number (5),  
  sg_estado         char (2),  
  nr_cep            varchar2 (8),  
  nr_ddd            number (3),  
  nr_fone           number(7),  
  ie_sexo           char(1)  
)
```

```
insert into cliente
```

```
values(2,'rafael','rua',55555,'s  
p','08290370',011,1234567,'  
m');
```

```
insert into cliente
```

```
values(2,null,null,55555,'sp','  
08290370',011,  
1234567,'m');
```

Padronização

A padronização em nomes de campos e restrições de tabelas é de suma importância para facilitar o entendimento e a velocidade de desenvolvimento de programadores e analistas.

Apresentaremos 2 técnicas de padrões utilizados em banco de dados:

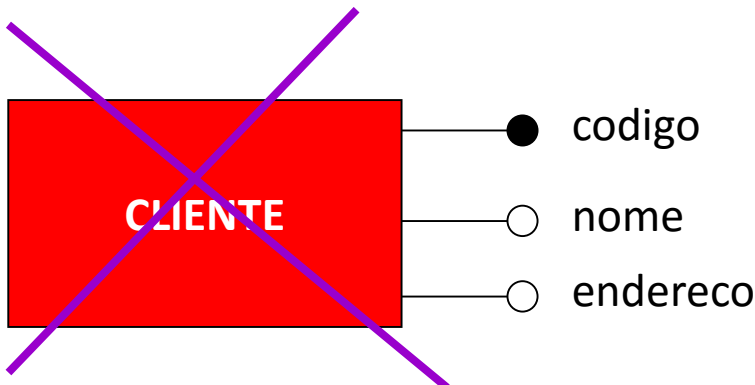
1) Trigramação

2) Qualificador_nomedescritivo

1) TRIGRAMAÇÃO

Um TRIGRAMA é uma cadeia de caracteres normalmente constituída **pelas três primeiras letras ou pelas três letras mais significativas de uma ENTIDADE**, que são utilizadas como prefixo dos Atributos ou Nomes de Campos de uma TABELA.

Para escolher-se o nome mais adequado de um Atributo, sugere-se a adoção de uma Notação Padronizada, utilizando a Técnica de TRIGRAMAÇÃO.



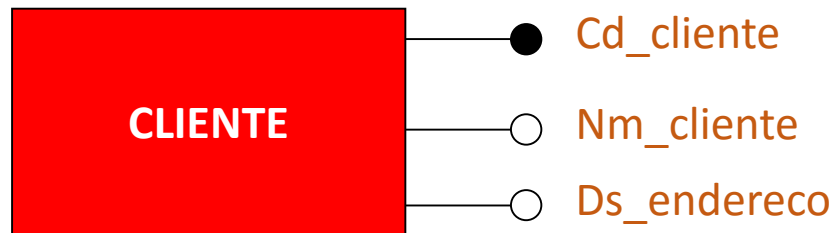
2) Qualificador_nome descritivo

Os qualificadores mais utilizados, com os seus devidos significados, são apresentados na tabela ao lado:

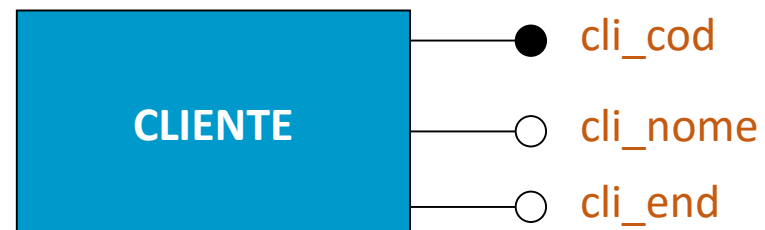
| Qualificador | Significado |
|--------------|--------------------|
| Cd | Código |
| Nm | Nome |
| Nr | Número |
| VI | Valor |
| Qt | Quantidade |
| Tx | Taxa ou percentual |
| Ds | Descrição |
| Sg | Sigla |
| Dt | Data |
| Hr | Hora |
| Ie ou Id | Identificador |
| Im | Imagem |

Padronizações

Qualificador_nome descritivo



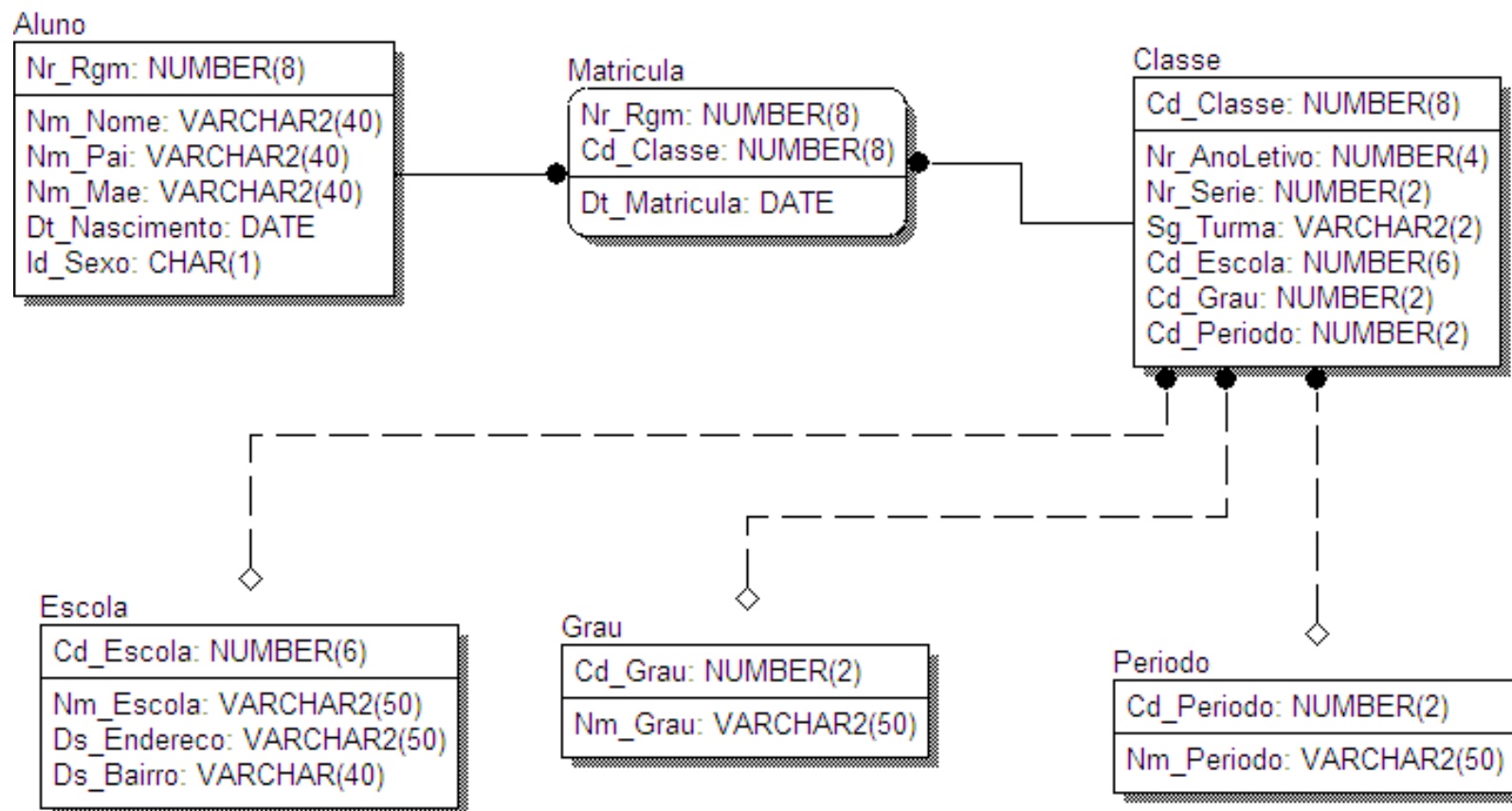
Trigramação



Exercício

Com base no DER abaixo e na estrutura da próxima página, crie as tabelas envolvidas.

Salve os comandos no arquivo Aula1.sql e envie via Link.



Exercício

Aluno

Nr_Rgm – Numérico(8) – Campo Chave
Nm_Nome - Varchar2(40) – Obrigatório
Nm_Pai – Varchar2(40) – Obrigatório
Nm_Mae – Varchar2(40) – Obrigatório
Dt_Nascimento – Date – Obrigatório
ID_Sexo – Char(1) – Obrigatório

Matricula

Cd_Classe - Numérico (8) – Campo Chave e (FK)
Nr_Rgm- Numérico(8) – Campo Chave e (FK)
Dt_Matricula – Date – Obrigatório

Escola

Cd_Escola - Numérico(6) – Campo Chave
Nm_Escola – Varchar2 (50) – Obrigatório
Ds_Endereco - Varchar2(50) – Obrigatório
Ds_Bairro - Varchar2 (40) – Obrigatório

Classe

Cd_Classe - Numérico(8) – Campo Chave
Nr_AnoLetivo - Numérico (4) – Obrigatório e maior que 2000
Cd_Escola - Numérico (6) – (FK)
Cd_Grau - Numérico (2) – (FK)
Nr_Serie - Numérico (2) – Obrigatório
Sg_Turma – Varchar2(2) – Obrigatório
Cd_Periodo - Numérico (2) - (FK)

Grau

Cd_Grau - Numérico(2) – Campo Chave
Nm_Grau – Varchar2 (50) – Obrigatório

Periodo

Cd_Periodo - Numérico(2) – Campo Chave
Nm_Periodo – Varchar2 (50) – Obrigatório

OBS.: Elaborar os relacionamentos e suas respectivas “constraint”.

Referências Bibliográficas

[1] Fanderuff, Damaris. Dominando o Oracle 9i: Modelagem e desenvolvimento. São Paulo: Pearson Education do Brasil, 2003.

[2] Costa, Rogério Luis de C., SQL : guia prático. 2. ed. Rio de Janeiro : Brasport, 2006.

[3] SILBERSCHATZ, A. Sistema de bancos de dados. São Paulo: Pearson Education do Brasil, 2004.

[4] Morelli, Eduardo M. Terra, 1996. Oracle 9i Fundamental: Sql, Pl/SQL e Administração. São Paulo: Érica, 2002.