

Oracle SQL

Comandos

- [Introdução](#)
- [Restrições de Integridade](#)
- [Comandos](#)
 - [DDL](#)
 - Tabela
 - [CREATE TABLE](#)
 - [ALTER TABLE](#)
 - [DROP TABLE](#)
 - Utilizador
 - CREATE USER
 - [ALTER USER](#)
 - DROP USER
 - [DML](#)
 - [INSERT](#)
 - [UPDATE](#)
 - [DELETE](#)

■ Comandos SQL

■ Categorias

- DDL -- Data Definition Language
- DML -- Data Manipulation Language
- DCL -- Data Control Language

■ Comandos DDL - Definição de Dados

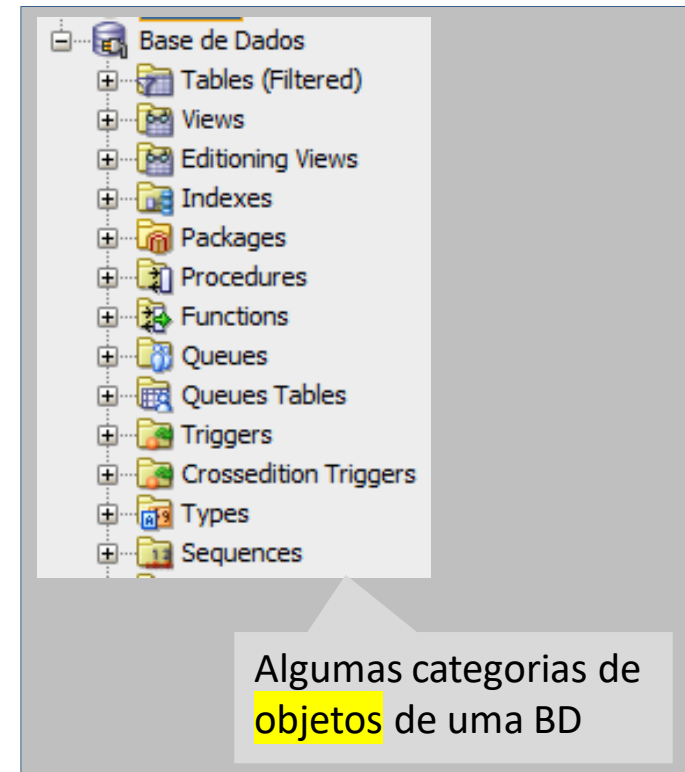
- CREATE -- Criar
 - DROP -- Eliminar
 - ALTER -- Alterar
- BD
• **Objetos** de BD

■ Comandos DML - Manipulação de Dados

- INSERT -- Inserir
 - DELETE -- Eliminar
 - UPDATE -- Atualizar
 - SELECT -- Consultar
- Dados da BD

■ Comandos DCL - Controlo de Dados

- CREATE USER -- Definir perfis de utilizadores
- GRANT -- Fornecer acesso ou privilégios a utilizadores sobre objetos da BD.
- REVOKE -- Remover acesso ou privilégios a utilizadores sobre objetos da BD.
- Etc.



Comandos DDL

Introdução

Restrições de Integridade

CREATE TABLE

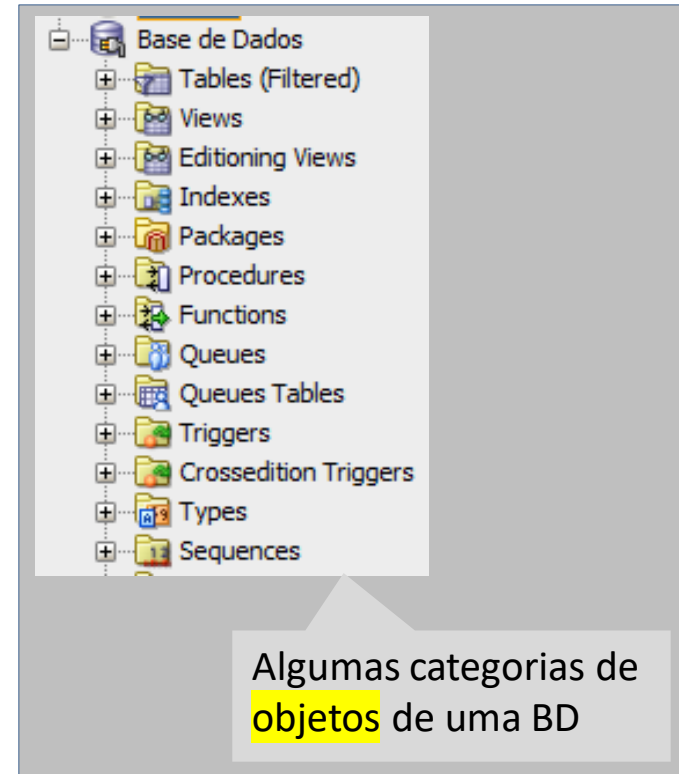
ALTER TABLE

DROP TABLE

ALTER USER

Interesse

- Criar
Eliminar
Alterar
- BD
• **Objetos** de BD
- Exemplos
 - Criar
Eliminar
Alterar
 - Tabela**
 - Criar
Eliminar
Alterar
 - Vista**



Principais Comandos DDL

- CREATE -- para criar objetos de BD.
- DROP -- para eliminar objetos de BD e partes de objetos (ex: restrições de integridade).
- ALTER -- para alterar objetos de BD e partes de objetos.

Sintaxe

- Cada comando ... tem uma sintaxe própria.

Comandos DDL

Introdução

Restrições de Integridade

CREATE TABLE

ALTER TABLE

DROP TABLE

ALTER USER

- [Introdução](#)
- [Tipos de Restrições de Integridade](#)
 - [NOT NULL](#)
 - [UNIQUE](#)
 - [PRIMARY KEY](#)
 - [FOREIGN KEY](#)
 - [CHECK](#)

- Interesse Comum a Diversos Comandos DDL
 - Impor regras que restringem valores na BD.
- Exemplo
 - Tabela ALUNO

NUMERO	NOME	TELEFONE
1010100	João Guedes	214457696
1020223	Ana Semedo	(null)
1030452	Rita Gomes	962534827

Proibir:

- Valores NULL

Impor:

- Números interiores com 9 dígitos

Impor:

- Números inteiros com 7 dígitos.

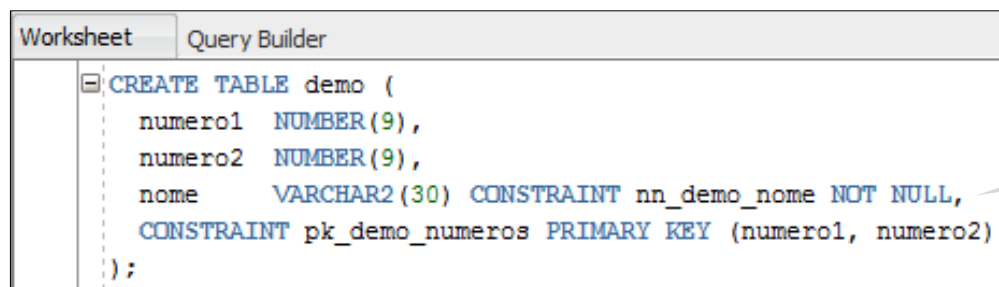
Proibir:

- Números repetidos.
- Valor NULL.

Tipo de Restrição	Descrição
NOT NULL	Proíbe o valor NULL numa coluna de uma tabela.
UNIQUE	<p>Proíbe valores repetidos (exceto o valor NULL):</p> <ul style="list-style-type: none"> ▪ Na mesma coluna de uma tabela; ▪ Numa combinação de múltiplas colunas de uma tabela. <p>Permite alguns valores NULL.</p>
PRIMARY KEY	<p>Combina, numa simples declaração, duas restrições:</p> <ul style="list-style-type: none"> ▪ NOT NULL ▪ UNIQUE <p>Proíbe múltiplas linhas de terem o mesmo valor na:</p> <ul style="list-style-type: none"> ▪ Mesma coluna de uma tabela; ▪ Combinação de múltiplas colunas de uma tabela. <p>Proíbe valores NULL.</p>
FOREIGN KEY	Requer que valores de uma tabela correspondam a valores noutra tabela.
CHECK	Requer que valores de uma tabela satisfaçam uma condição específica.

Formas

- *Inline* -- especificação de restrição ... **incluída** na definição de uma **coluna** individual.
- *Out-of-line* -- especificação de restrição ... **separada** da definição de qualquer **coluna**.



```
CREATE TABLE demo (  
    numero1  NUMBER(9),  
    numero2  NUMBER(9),  
    nome     VARCHAR2(30) CONSTRAINT nn_demo_nome NOT NULL,  
    CONSTRAINT pk_demo_numeros PRIMARY KEY (numero1, numero2)  
);
```

Restrição **inline**

Restrição **out-of-line**

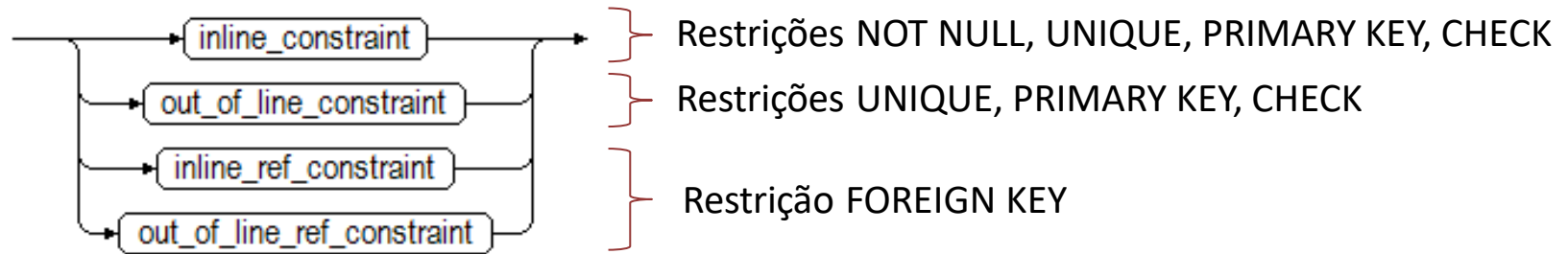
Restrição **NOT NULL**:

- **Única** restrição que **só** pode ser especificada *inline*.

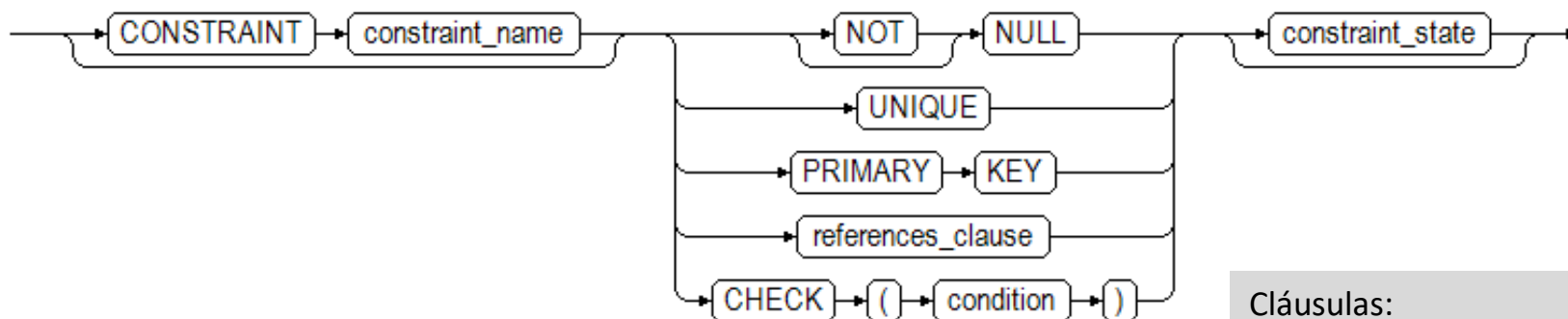
Feita em Comandos DDL

- [CREATE TABLE](#)
- [ALTER TABLE](#)

Tipos de Sintaxe



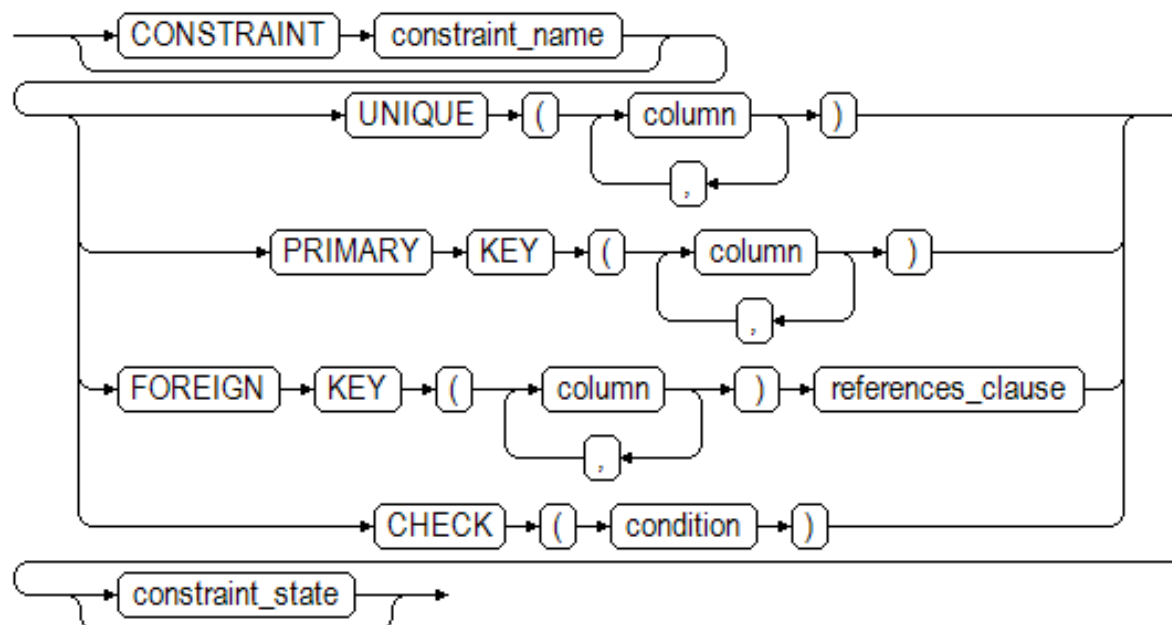
inline_constraint



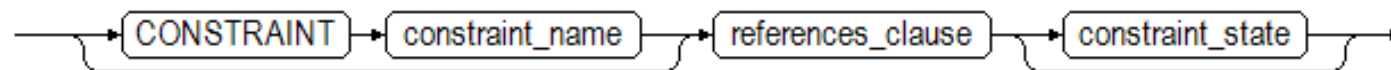
Cláusulas:

- [NOT NULL](#)
- [UNIQUE](#)
- [PRIMARY KEY](#)
- [CHECK](#)
- [references_clause](#)
- [constraint_state](#)

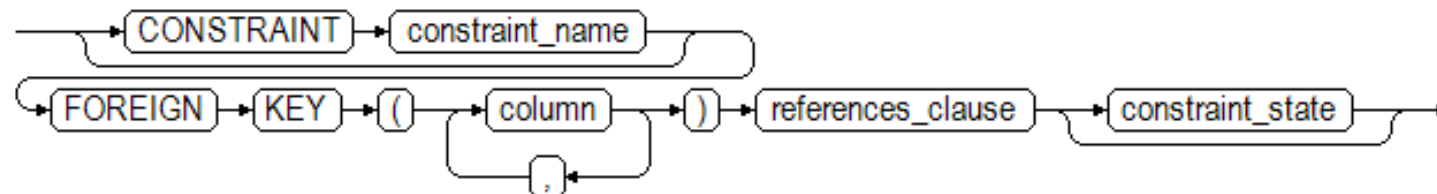
out_of_line_constraint



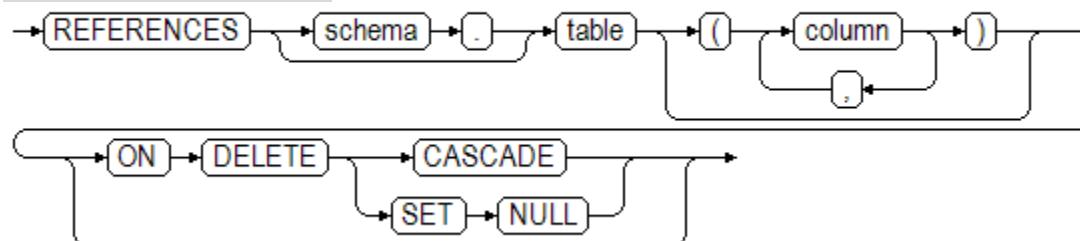
inline_ref_constraint



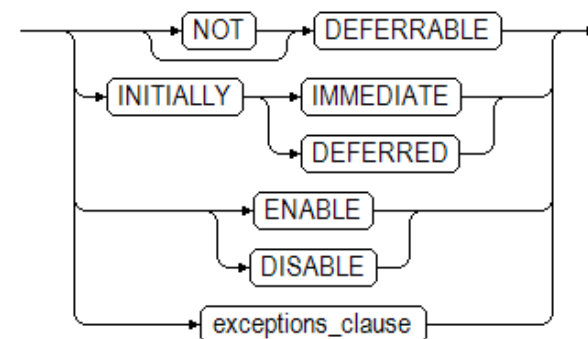
out_of_line_ref_constraint



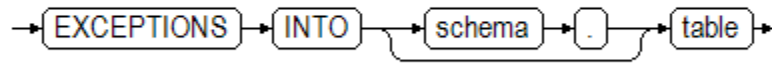
references_clause



constraint_state

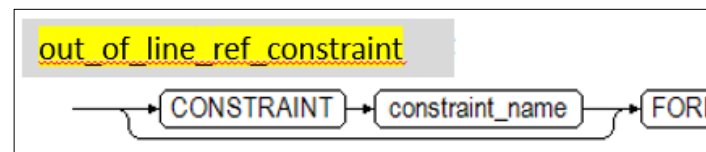
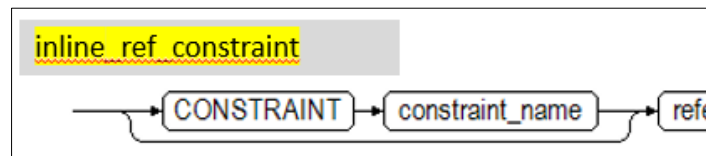
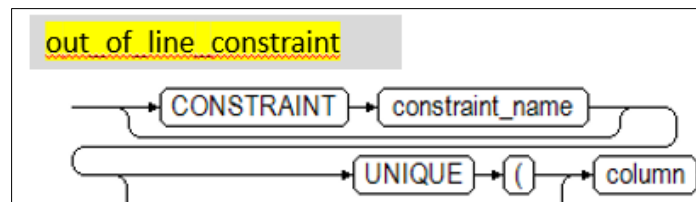
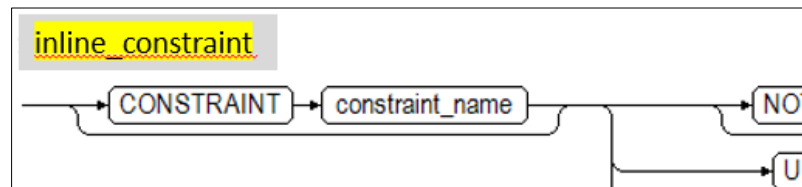


exception_clause



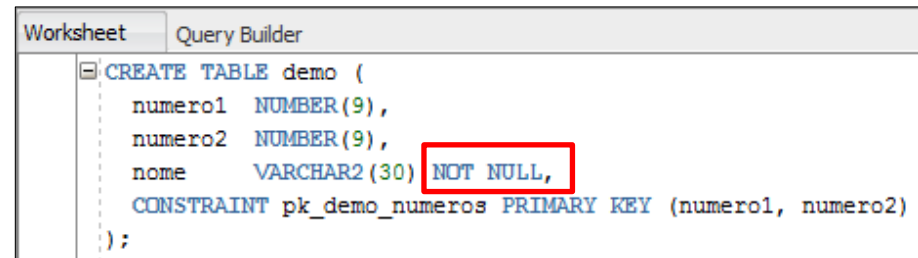
- Especificada na Cláusula **CONSTRAINT**

- **CONSTRAINT constraint_name** -- início da especificação de uma restrição.



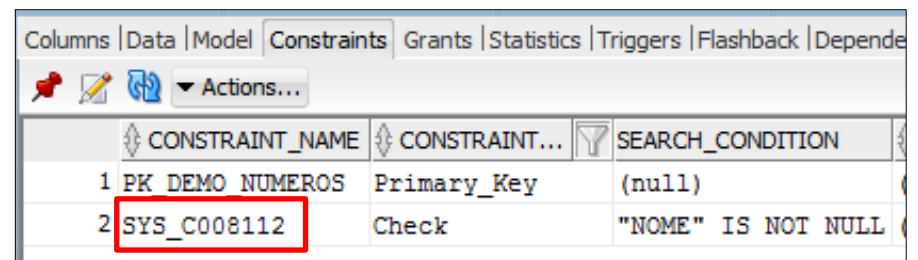
▪ Especificação

- Opcional
- Nome por omissão
 - Formato
 - SYS_Cn -- n = número
 - Exemplo
 - Restrição NOT NULL
 - Especificada **sem** nome:



```
CREATE TABLE demo (  
  numero1  NUMBER(9),  
  numero2  NUMBER(9),  
  nome     VARCHAR2(30) NOT NULL,  
  CONSTRAINT pk_demo_numeros PRIMARY KEY (numero1, numero2)  
);
```

- Nome atribuído pela BD:



	CONSTRAINT_NAME	CONSTRAINT...	SEARCH_CONDITION
1	PK DEMO NUMEROS	Primary_Key	(null)
2	SYS_C008112	Check	"NOME" IS NOT NULL

▪ **Formato**

▪ Genérico

- TipoDeRestrição_nomeTabela_nomeColuna -- comprimento máximo = 30 caracteres.

▪ TipoDeRestrição

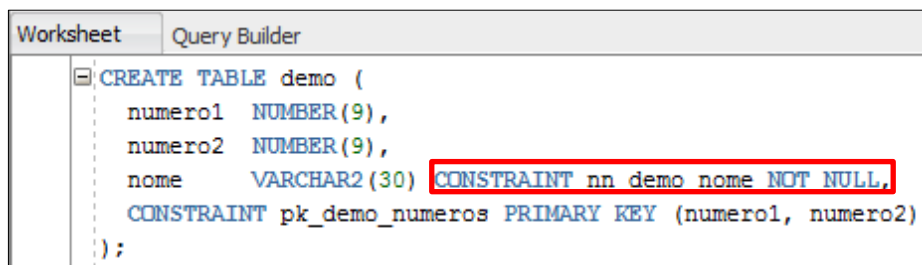
- nn -- NOT NULL
- uk -- UNIQUE
- ck -- CHECK
- pk -- PRIMARY_KEY
- fk -- FOREIGN KEY

▪ Exemplo

Worksheet	Query Builder
<pre>CREATE TABLE empregado (id_empregado NUMBER(6,0) CONSTRAINT pk_emp_id_empregado PRIMARY KEY, primeiro_nome VARCHAR2(20) CONSTRAINT nn_emp_primeiro_nome NOT NULL, ultimo_nome VARCHAR2(25) CONSTRAINT nn_emp_ultimo_nome NOT NULL, email VARCHAR2(25) CONSTRAINT nn_emp_email NOT NULL, telefone NUMBER(9) CONSTRAINT ck_emp_telefone CHECK (REGEXP_LIKE(telefone,'[29]\d{8}')), data_contratacao DATE CONSTRAINT nn_emp_data_contratacao NOT NULL, id_funcao VARCHAR2(10) CONSTRAINT nn_emp_id_funcao NOT NULL, salario NUMBER(8,2), perc_comissao NUMBER(2,2), id_diretor NUMBER(6,0), id_departamento NUMBER(4,0), CONSTRAINT ck_emp_salario_min CHECK (salario > 0), CONSTRAINT uk_emp_email UNIQUE (email), CONSTRAINT fk_emp_id_departamento FOREIGN KEY (id_departamento) REFERENCES departamento(id_departamento), CONSTRAINT fk_emp_id_funcao FOREIGN KEY (id_funcao) REFERENCES funcao(id_funcao), CONSTRAINT fk_emp_id_diretor FOREIGN KEY (id_diretor) REFERENCES empregado(id_empregado));</pre>	

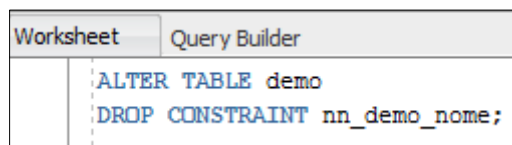
▪ Interesse (1/2)

- Referenciar uma restrição num comando SQL.
 - Exemplo
 - Restrição NOT NULL da tabela demo.



```
Worksheet Query Builder
CREATE TABLE demo (
  numero1 NUMBER(9),
  numero2 NUMBER(9),
  nome VARCHAR2(30) CONSTRAINT nn_demo_nome NOT NULL,
  CONSTRAINT pk_demo_numeros PRIMARY KEY (numero1, numero2)
);
```

▪ Eliminar restrição



```
Worksheet Query Builder
ALTER TABLE demo
DROP CONSTRAINT nn_demo_nome;
```

Nome da restrição:

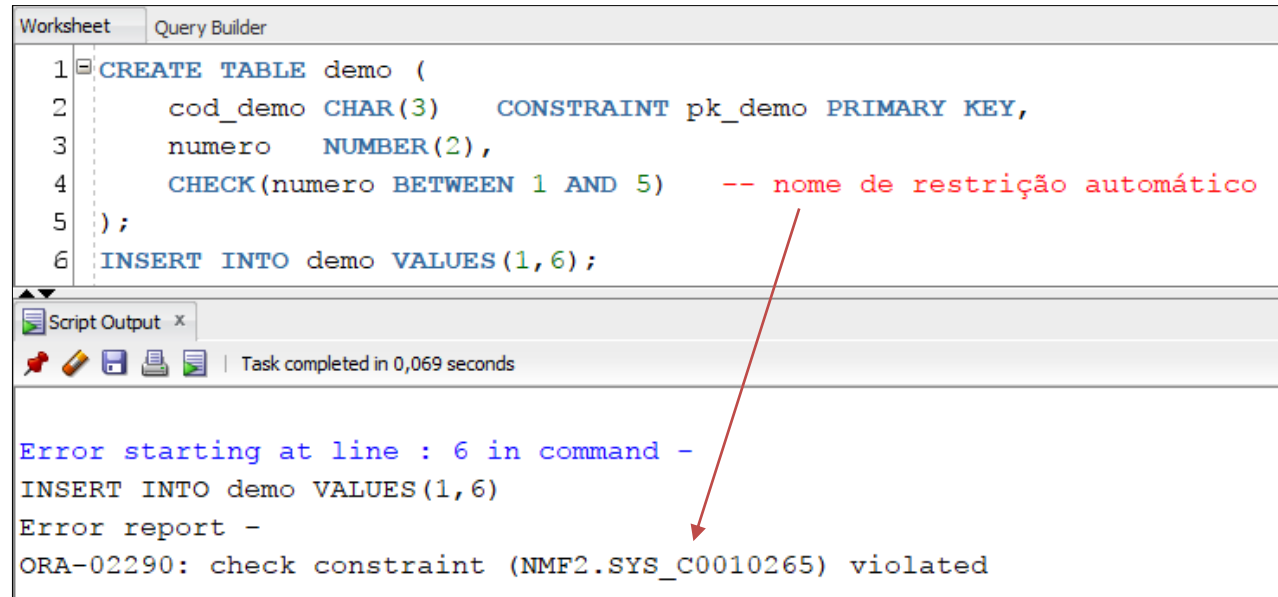
- Referência da restrição a eliminar.

▪ Identificar a ...

▪ Interesse (2/2)

▪ Facilitar o *debug*.

▪ Exemplo:



Worksheet Query Builder

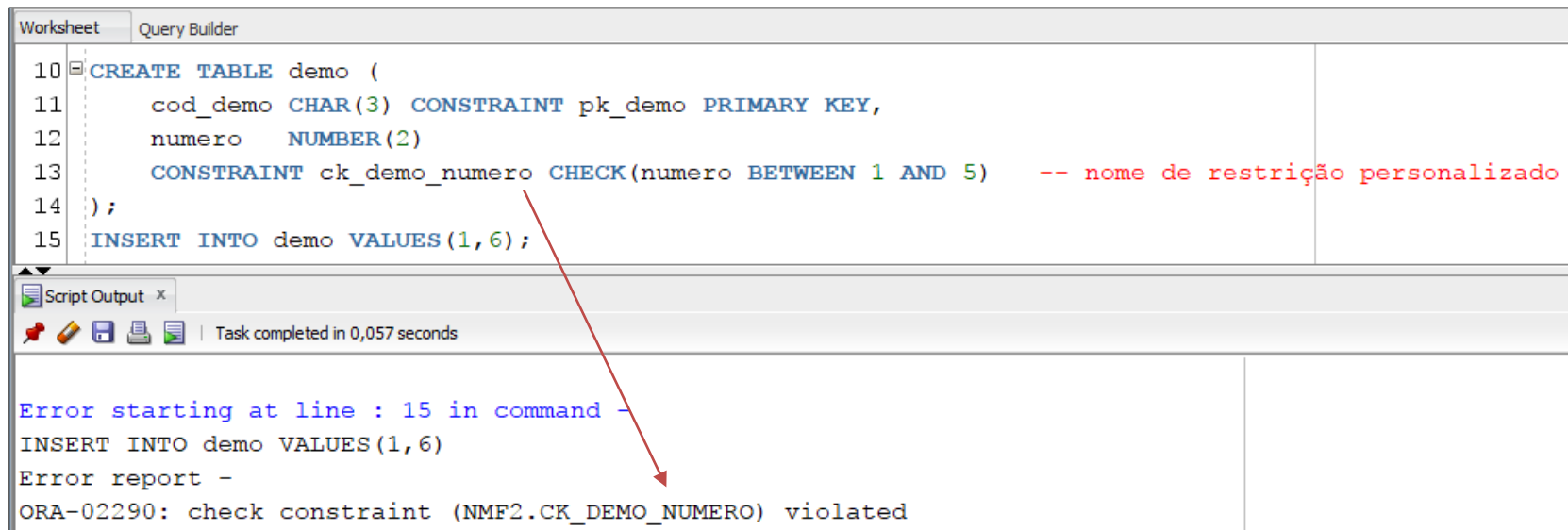
```
1 CREATE TABLE demo (  
2     cod_demo CHAR(3) CONSTRAINT pk_demo PRIMARY KEY,  
3     numero  NUMBER(2),  
4     CHECK(numero BETWEEN 1 AND 5)  -- nome de restrição automático  
5 );  
6 INSERT INTO demo VALUES (1,6);
```

Script Output x

Task completed in 0,069 seconds

Error starting at line : 6 in command -
INSERT INTO demo VALUES (1,6)
Error report -
ORA-02290: check constraint (NMF2.SYS_C0010265) violated

A red arrow points from the comment "-- nome de restrição automático" in the SQL code to the error message "ORA-02290: check constraint (NMF2.SYS_C0010265) violated".



Worksheet Query Builder

```
10 CREATE TABLE demo (  
11     cod_demo CHAR(3) CONSTRAINT pk_demo PRIMARY KEY,  
12     numero  NUMBER(2)  
13     CONSTRAINT ck_demo_numero CHECK(numero BETWEEN 1 AND 5)  -- nome de restrição personalizado  
14 );  
15 INSERT INTO demo VALUES (1,6);
```

Script Output x

Task completed in 0,057 seconds

Error starting at line : 15 in command -
INSERT INTO demo VALUES (1,6)
Error report -
ORA-02290: check constraint (NMF2.CK_DEMO_NUMERO) violated

A red arrow points from the comment "-- nome de restrição personalizado" in the SQL code to the error message "ORA-02290: check constraint (NMF2.CK_DEMO_NUMERO) violated".

- Interesse

- Proibir coluna de uma tabela de ter valores NULL.

- NOT

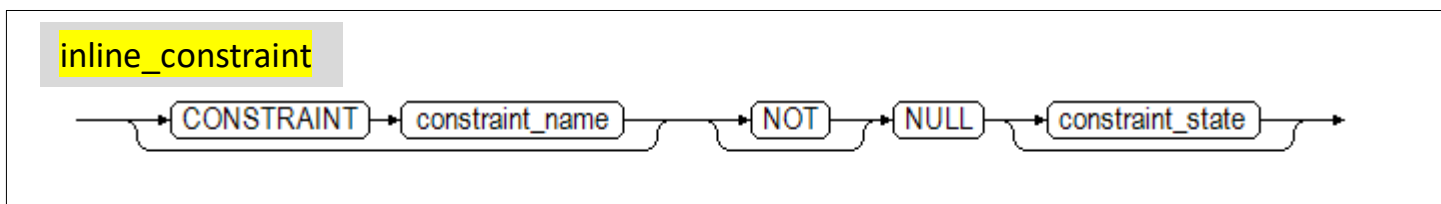
- Opcional

- Restrição por Omissão

- NULL
 - Especifica que coluna de uma tabela pode ter valores NULL.

- Forma de Especificação

- *Inline*
 - Única forma permitida.
 - Sintaxe



■ Exemplos

Worksheet	Query Builder
	<pre>CREATE TABLE empregado (id_empregado NUMBER(6,0) CONSTRAINT pk_emp_id_empregado PRIMARY KEY, primeiro_nome VARCHAR2(20) CONSTRAINT nn_emp_primeiro_nome NOT NULL, ultimo_nome VARCHAR2(25) CONSTRAINT nn_emp_ultimo_nome NOT NULL, email VARCHAR2(25) CONSTRAINT nn_emp_email NOT NULL, telefone NUMBER(9) CONSTRAINT ck_emp_telefone CHECK (REGEXP_LIKE(telefone, '[29]\d{8}')), data_contratacao DATE CONSTRAINT nn_emp_data_contratacao NOT NULL, id_funcao VARCHAR2(10) CONSTRAINT nn_emp_id_funcao NOT NULL, salario NUMBER(8,2), perc_comissao NUMBER(2,2), id_diretor NUMBER(6,0), id_departamento NUMBER(4,0), CONSTRAINT ck_emp_salario_min CHECK (salario > 0), CONSTRAINT uk_emp_email UNIQUE (email), CONSTRAINT fk_emp_id_departamento FOREIGN KEY (id_departamento) REFERENCES departamento(id_departamento), CONSTRAINT fk_emp_id_funcao FOREIGN KEY (id_funcao) REFERENCES funcao(id_funcao), CONSTRAINT fk_emp_id_diretor FOREIGN KEY (id_diretor) REFERENCES empregado(id_empregado));</pre>

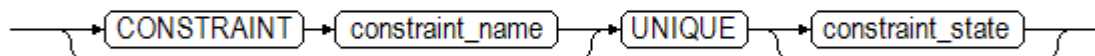
- Interesse

- Impor uma chave UNIQUE: -- ≠ chave primária
 - Simples
 - Composta

▪ Chave UNIQUE Simples

- Garante que numa **coluna** (UNIQUE):
 - Não existem valores repetidos. -- ⇒ designação de **chave**.
 - Podem existir valores NULL.
- Formas de especificação

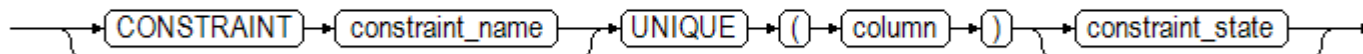
inline_constraint



Worksheet Query Builder

```
CREATE TABLE pessoa (  
  id    NUMBER CONSTRAINT pk_pessoa_id PRIMARY KEY,  
  nome  VARCHAR2(30) CONSTRAINT nn_pessoa_nome NOT NULL,  
  email VARCHAR2(25) CONSTRAINT uk_pessoa_email UNIQUE  
);
```

out-of-line constraint

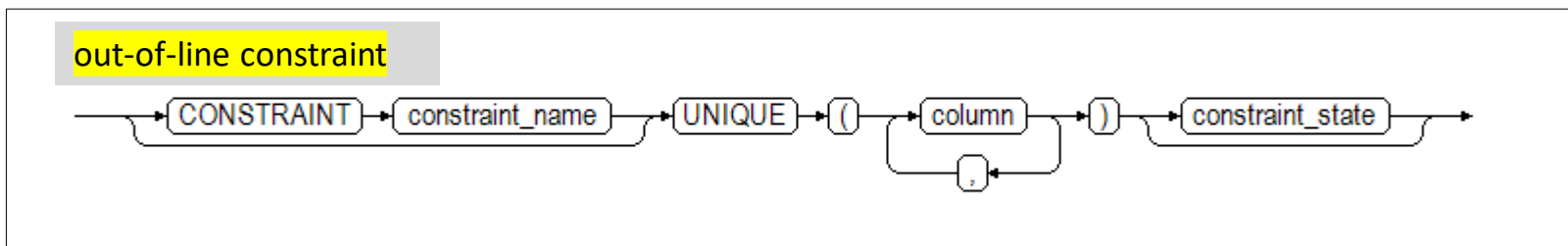


Worksheet Query Builder

```
CREATE TABLE pessoa (  
  id    NUMBER CONSTRAINT pk_pessoa_id PRIMARY KEY,  
  nome  VARCHAR2(30) CONSTRAINT nn_pessoa_nome NOT NULL,  
  email VARCHAR2(25),  
  CONSTRAINT uk_pessoa_email UNIQUE (email)  
);
```

Chave UNIQUE Composta

- Garante que numa **combinação de colunas** de uma tabela (colunas UNIQUE):
 - Não existem duas linhas ... com a mesma combinação de valores.
 - Podem existir valores NULL.
- Forma de especificação



- Exemplo
 - Restrição UNIQUE impõe:
 - Combinação de valores das colunas **nome** e **id_tipo_projeto** ... única.

```

Worksheet  Query Builder
CREATE TABLE projeto (
  id_projeto      NUMBER(3) CONSTRAINT pk_proj_id_projeto PRIMARY KEY,
  nome            VARCHAR(20) CONSTRAINT nn_proj_nome NOT NULL,
  descricao       VARCHAR(50) CONSTRAINT nn_proj_descricao NOT NULL,
  id_tipo_projeto NUMBER CONSTRAINT nn_proj_id_tipo_projeto NOT NULL,
  CONSTRAINT uk_proj_nome_tipo UNIQUE (nome, id_tipo_projeto),
  CONSTRAINT fk_proj_tipo FOREIGN KEY (id_tipo_projeto) REFERENCES tipo_projeto(id_tipo_projeto)
);
    
```


- **Limitações**

- Illegal:
 - Em colunas TIMESTAMP WITH TIME ZONE.
- Chave UNIQUE composta:
 - Nº colunas ≤ 32
- Mesma coluna/combinação de colunas:
 - Não pode ser simultaneamente:
 - UNIQUE
 - PRIMARY KEY

- Interesse

- Impor uma chave primária:
 - Simples
 - Composta

- Combinação das Restrições

- NOT NULL
- UNIQUE

PRIMARY KEY = UNIQUE + NOT NULL

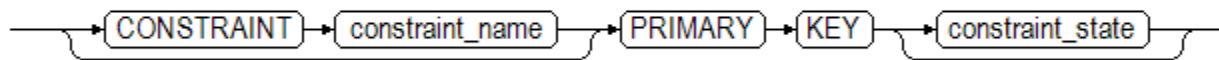
- Declaração numa Tabela

- *Inline* -- chave primária simples
- *Out-of-line* -- chave primária composta

▪ Chave Primária Simples

- Garante que numa **coluna**:
 - Não existem valores repetidos. -- ⇒ designação de **chave**.
 - Não existem valores NULL.
- Formas de especificação

inline_constraint

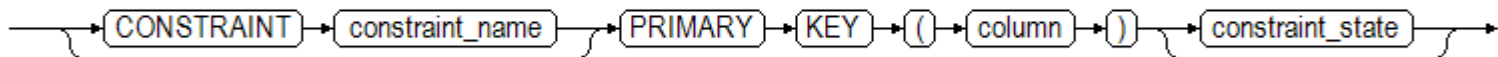


Worksheet

Query Builder

```
CREATE TABLE demo (  
  numero NUMBER(9) CONSTRAINT pk_demo_numero PRIMARY KEY,  
  nome VARCHAR2(30) CONSTRAINT nn_demo_nome NOT NULL  
);
```

out-of-line constraint



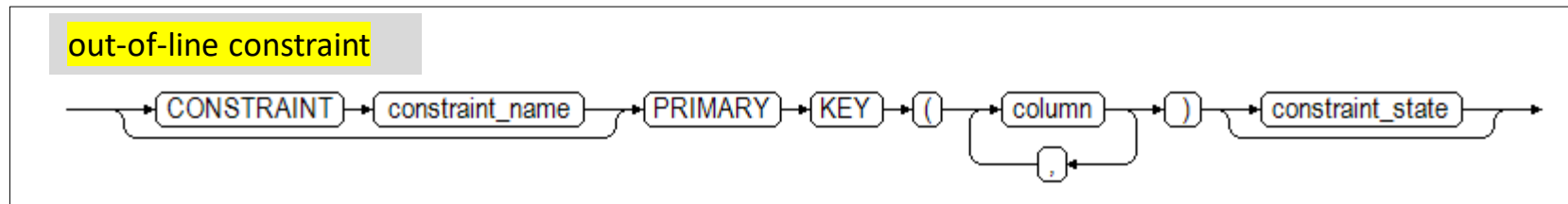
Worksheet

Query Builder

```
CREATE TABLE demo (  
  numero NUMBER(9),  
  nome VARCHAR2(30) CONSTRAINT nn_demo_nome NOT NULL,  
  CONSTRAINT pk_demo_numero PRIMARY KEY (numero)  
);
```

Chave Primária Composta

- Garante que numa **combinação de colunas** de uma tabela:
 - Não existem duas linhas ... com a mesma combinação de valores.
 - Não existem valores NULL.
- Forma de especificação



- Exemplo
 - Restrição PRIMARY KEY impõe:
 - Combinação de valores das colunas **id_disciplina** e **num_aluno** ... única.

```
Worksheet  Query Builder
CREATE TABLE inscricao (
  id_disciplina NUMBER(2),
  num_aluno      NUMBER(7),
  CONSTRAINT pk_inscricao_id_disc_num_aluno PRIMARY KEY (id_disciplina, num_aluno),
  CONSTRAINT fk_inscricao_id_disciplina FOREIGN KEY (id_disciplina) REFERENCES disciplina(id_disciplina),
  CONSTRAINT fk_inscricao_num_aluno FOREIGN KEY (num_aluno) REFERENCES aluno(num_aluno)
);
```

▪ Limitações

- Uma Tabela:
 - Apenas pode ter **uma** chave primária. -- ao contrário de UNIQUE ... que pode ter várias.
- Illegal:
 - Em colunas TIMESTAMP WITH TIME ZONE.
- Chave primária composta:
 - Nº colunas ≤ 32
- Mesma coluna/combinação de colunas:
 - Não pode ser simultaneamente:
 - UNIQUE
 - PRIMARY KEY

- Interesse

- Impor uma chave estrangeira:
 - Simples
 - Composta

Uma chave estrangeira referencia:

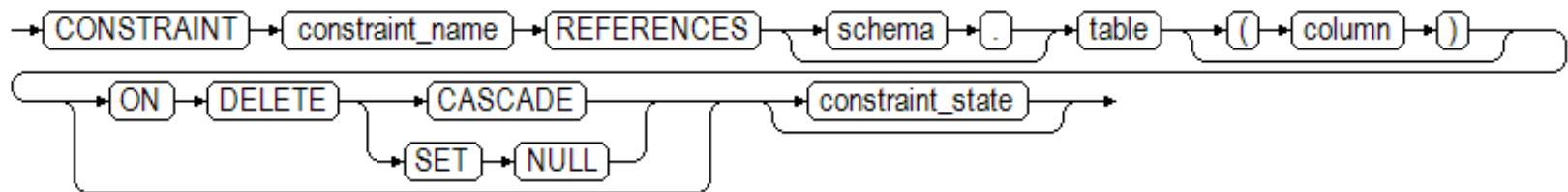
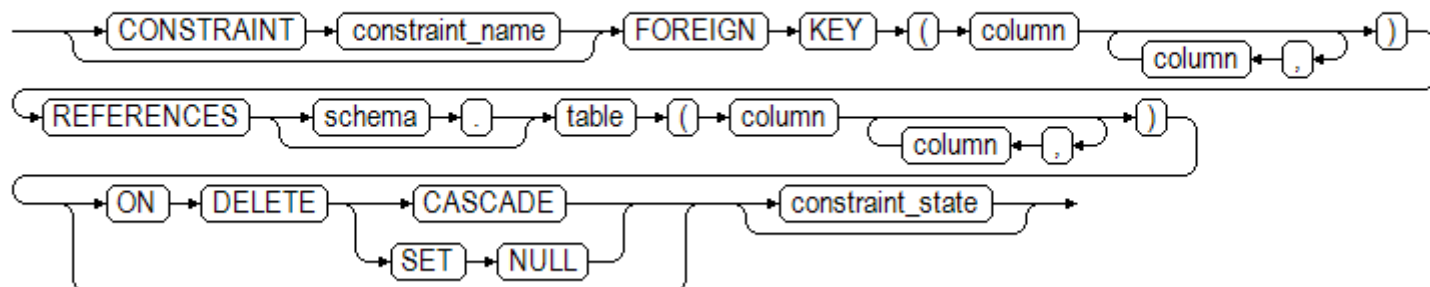
- Chave **primária**
- Chave **unique**

▪ Chave Estrangeira Simples (1/3)

- Garante que **uma** coluna:
 - Contém valores **correspondentes** de outra coluna (chave primária/*unique*) pertencente a:
 - Outra tabela.
 - Mesma tabela.
- Formas de especificação:

Restrição FOREIGN KEY:

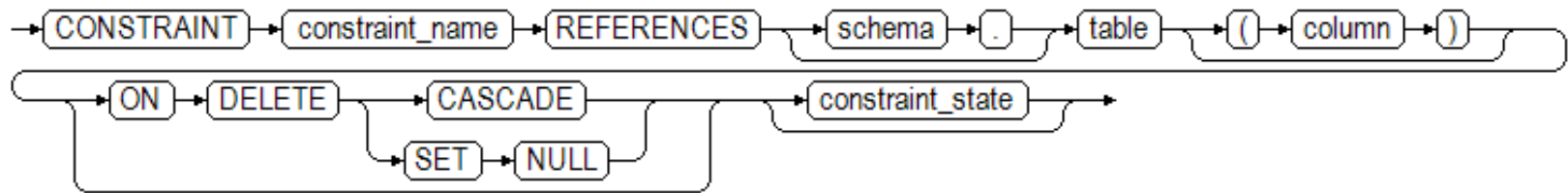
- Permite o valor NULL.

inline_ref_constraint**out_of_line_ref_constraint**

Chave Estrangeira Simples (2/3)

Especificação *inline*

inline_ref_constraint



Worksheet Query Builder

```

CREATE TABLE aluno(
  num_aluno NUMBER(7) CONSTRAINT pk_aluno_num_aluno PRIMARY KEY,
  nome VARCHAR2(25) CONSTRAINT nn_aluno_nome NOT NULL,
  id_pais NUMBER(3) CONSTRAINT fk_aluno_pais REFERENCES pais(id_pais),
  CONSTRAINT ck_aluno_num_aluno CHECK(REGEXP_LIKE(num_aluno, '\d{7}'))
);
  
```

País do aluno

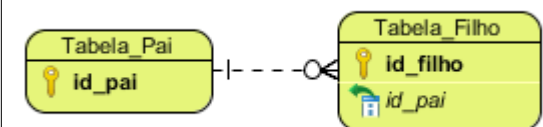
Worksheet Query Builder

```

CREATE TABLE aluno(
  num_aluno NUMBER(7) CONSTRAINT pk_aluno_num_aluno PRIMARY KEY,
  nome VARCHAR2(25) CONSTRAINT nn_aluno_nome NOT NULL,
  id_pais NUMBER(3) CONSTRAINT fk_aluno_pais REFERENCES pais,
  CONSTRAINT ck_aluno_num_aluno CHECK(REGEXP_LIKE(num_aluno, '\d{7}'))
);
  
```

Nome de coluna da tabela pai:

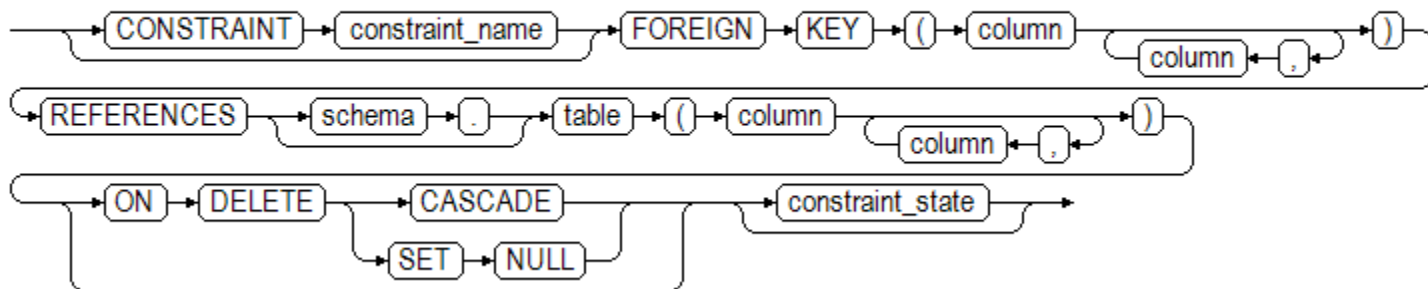
- Opcional
- nome = nome col. definida.



Chave Estrangeira Simples (3/3)

Especificação *out-of-line*

out_of_line_ref_constraint



Worksheet Query Builder

```

CREATE TABLE aluno(
  num_aluno NUMBER(7) CONSTRAINT pk_aluno_num_aluno PRIMARY KEY,
  nome VARCHAR2(25) CONSTRAINT nn_aluno_nome NOT NULL,
  id_pais NUMBER(3),
  CONSTRAINT ck_aluno_num_aluno CHECK(REGEXP LIKE(num_aluno,'d{7}')),
  CONSTRAINT fk_aluno_pais FOREIGN KEY (id_pais) REFERENCES pais(id_pais)
);
    
```

Worksheet Query Builder

```

CREATE TABLE aluno(
  num_aluno NUMBER(7) CONSTRAINT pk_aluno_num_aluno PRIMARY KEY,
  nome VARCHAR2(25) CONSTRAINT nn_aluno_nome NOT NULL,
  id_pais NUMBER(3),
  CONSTRAINT ck_aluno_num_aluno CHECK(REGEXP LIKE(num_aluno,'d{7}')),
  CONSTRAINT fk_aluno_pais FOREIGN KEY (id_pais) REFERENCES pais
);
    
```

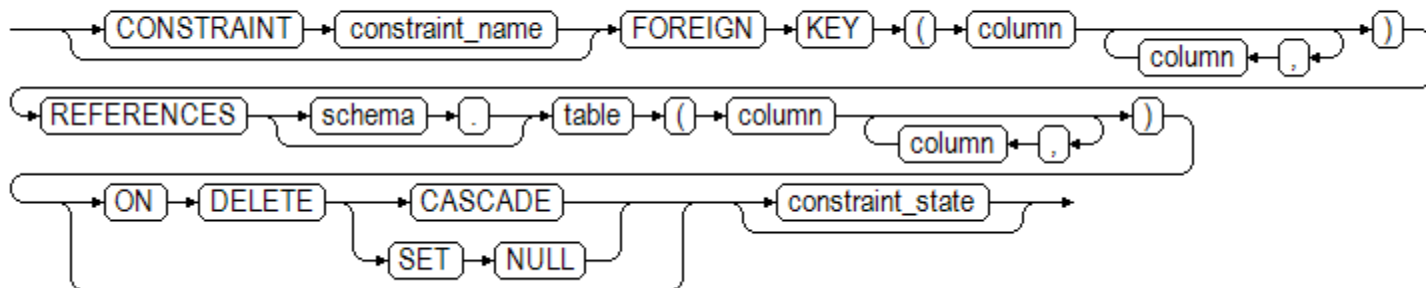
Nome de coluna da tabela pai:

- Opcional
- nome = nome col. definida.

Chave Estrangeira Composta

- Garante que numa **combinação de colunas** de uma tabela:
 - Não existem duas linhas ... com a mesma combinação de valores.
 - Não existem valores NULL.
- Forma de especificação

out_of_line_ref_constraint

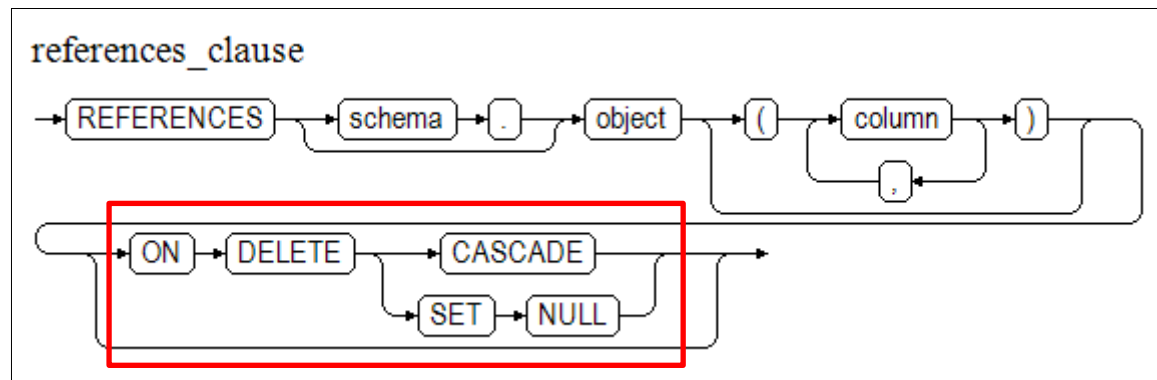
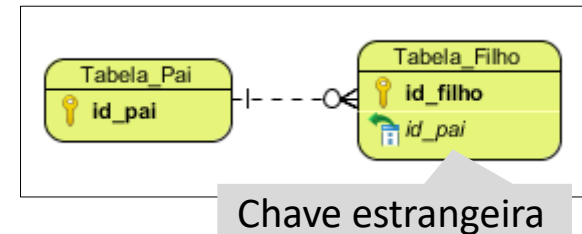


```
Worksheet Query Builder
CREATE TABLE demo (
  id_demo    NUMBER CONSTRAINT pk_demo_id PRIMARY KEY,
  pk1 NUMBER,
  pk2 NUMBER,
  CONSTRAINT fk_demo_pks FOREIGN KEY (pk1, pk2) REFERENCES tabela1(pk1,pk2)
);

CREATE TABLE tabela1 (
  pk1 NUMBER,
  pk2 NUMBER,
  CONSTRAINT pk_tabela1_pks PRIMARY KEY (pk1, pk2)
);
```

Cláusula ON DELETE

- Opcional
- Quando é omitida:
 - Proibido eliminar valores de chave primária/*unique* que:
 - Têm linhas **dependentes** em tabelas filho.
- Especifica:
 - Forma de manter, automaticamente, a **integridade referencial** quando:
 - É eliminado um valor de chave primária/*unique* referenciado em tabelas filho.
 - Formas alternativas:
 - Cláusula **CASCADE** -- para **eliminar linhas** das chaves estrangeiras dependentes.
 - Cláusula **SET NULL** /* para **converter** os **valores** das chaves estrangeiras dependentes para **NULL**. */



▪ Limitações

- Coluna da chave estrangeira:
 - Não pode ser do tipo `TIMESTAMP WITH TIME ZONE`.
- Chave primária/*unique* de uma tabela pai:
 - Só pode ser referenciada se estiver definida.
- Chave composta:
 - nº colunas ≤ 32
- Cláusula *AS subquery*:
 - Impede a definição deste tipo de restrição. -- p.ex., em `CREATE TABLE ... AS subquery;`
 - Contudo é possível:
 1. Criar tabela sem a restrição chave estrangeira;
 2. Adicionar restrição com comando `ALTER TABLE`.

- Interesse

- Impor uma condição que cada linha numa tabela deve satisfazer.

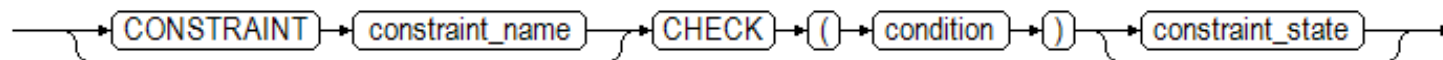
Valores alternativos de uma condição:

- TRUE
- FALSE
- UNKNOWN -- devido a NULL.

■ Formas de Especificação

- *Inline* -- para referenciar a coluna correspondente.
- *Out-of-line* -- permite referenciar múltiplas colunas da própria tabela.

inline_constraint / out-of-line constraint

[condition](#)

```
Worksheet Query Builder
CREATE TABLE encomenda_detalhe (
  id_encomenda NUMBER CONSTRAINT fk_enc_det_id_enc REFERENCES encomenda(id_encomenda),
  id_produto NUMBER CONSTRAINT fk enc det id prod REFERENCES produto(id_produto),
  quantidade NUMBER CONSTRAINT ck_enc_det_qtd CHECK(quantidade>0),
  custo NUMBER CONSTRAINT ck enc det custo CHECK(custo>0),
  CONSTRAINT pk_enc_det PRIMARY KEY (id_encomenda,id_produto)
);
```

inline

```
Worksheet Query Builder
CREATE TABLE encomenda_detalhe (
  id_encomenda NUMBER CONSTRAINT fk_enc_det_id_enc REFERENCES encomenda(id_encomenda),
  id_produto NUMBER CONSTRAINT fk_enc_det_id_prod REFERENCES produto(id_produto),
  quantidade NUMBER,
  custo NUMBER,
  CONSTRAINT pk enc det PRIMARY KEY (id encomenda,id produto),
  CONSTRAINT ck enc det qtd cust CHECK(quantidade>0 AND custo>0)
);
```

out-of-line

▪ Limitações

▪ Condição da restrição CHECK:

- Só pode referenciar **colunas** da **mesma tabela**.

Illegal referenciar colunas de outras tabelas.

- Não pode incluir:

- Expressões de **subqueries**.

- Invocação de funções que não são determinísticas:

- CURRENT_DATE
 - CURRENT_TIMESTAMP
 - DBTIMEZONE
 - LOCALTIMESTAMP
 - SESSIONTIMEZONE
 - SYSDATE
 - SYSTIMESTAMP
 - USER

- **Múltiplas Restrições CHECK**

- Oracle não verifica:
 - Se restrições são mutuamente exclusivas.
- Para evitar conflitos:
 - A especificação deve ser cuidada.
 - Não assumir qualquer ordem de avaliação das condições.

Comandos DDL

Introdução

Restrições de Integridade

CREATE TABLE

ALTER TABLE

DROP TABLE

ALTER USER

- [Introdução](#)
 - [Interesse](#)
 - [Alterar uma Tabela](#)
- [Declaração](#)
 - [Geral \(Simplificada\)](#)
 - [Detalhada](#)
 - [CREATE TABLE](#)
 - [Tabela com Dados](#)
 - [Coluna](#)
 - [Coluna Identidade \(Auto-Incremento\)](#)

■ Criar Tabela Relacional

- Sem dados
- Com dados -- baseada numa *subquery*.

■ Exemplos

- Sem dados

Declaração de **coluna** da tabela:

- **Sintaxe**: nome tipoDeDados restrição

```
Worksheet Query Builder
CREATE TABLE aluno (
  num_aluno NUMBER(7) CONSTRAINT pk_aluno_num PRIMARY KEY,
  nome VARCHAR2(50) CONSTRAINT nn_aluno_nome NOT NULL,
  id_pais NUMBER CONSTRAINT fk_aluno_pais REFERENCES pais(id_pais),
  CONSTRAINT ck_aluno_num CHECK( REGEXP_LIKE(num_aluno, '\d{7}') ),
  CONSTRAINT ck_aluno_nome CHECK( REGEXP_LIKE(nome, '[a-zA-Z][a-zA-Z\s]+' ) )
);
```

Especificação **inline** de restrição CHAVE PRIMÁRIA

Especificação **out-of-line** de restrição CHECK

Query Result x

SQL | All Rows Fetched: 2 in 0,001 se

	NUM_ALUNO	NOME	ID_PAIS
1	1050999	NELO SEMEDO	1
2	1999999	JONAS GULOSO	2

Dados adicionados posteriormente.

- Com dados -- comando CREATE TABLE com **cláusula AS subquery**

```
Worksheet Query Builder
CREATE TABLE aluno_pt AS (
  SELECT a.num_aluno, a.nome
  FROM aluno a, pais p
  WHERE a.id_pais=p.id_pais AND UPPER(p.nome)='PORTUGAL'
);
```

subquery

Query Result x

SQL | All Rows Fetched: 1 in 0,00

	NUM_ALUNO	NOME
1	1050999	NELO SEMEDO

▪ Adicionar:

- Linha -- com comando **INSERT** ...
- Coluna -- com comando **ALTER** TABLE ... **ADD** COLUMN ...
- Restrição de integridade -- com comando **ALTER** TABLE ... **ADD** CONSTRAINT

▪ Eliminar:

- Linha -- com comando **DELETE** ...
- Coluna -- com comando **ALTER** TABLE ... **DROP** COLUMN ...
- Restrição de integridade -- com comando **ALTER** TABLE ... **DROP** CONSTRAINT ...

▪ Modificar:

- Definição de uma coluna -- com comando **ALTER** TABLE ... **MODIFY** ...

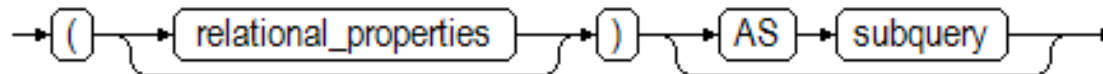
Cláusulas do comando ALTER:

- ADD
- DROP
- MODIFY

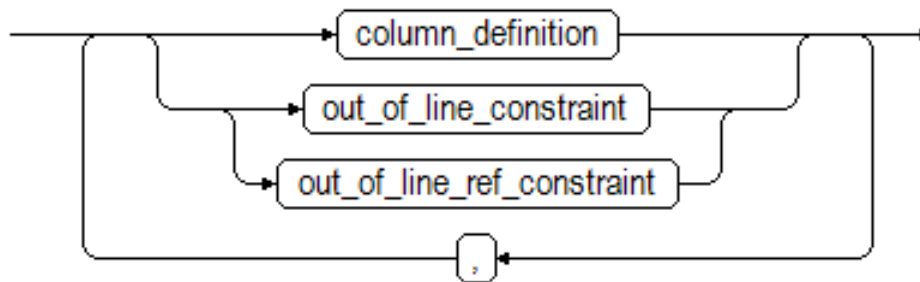
CREATE TABLE



relational table



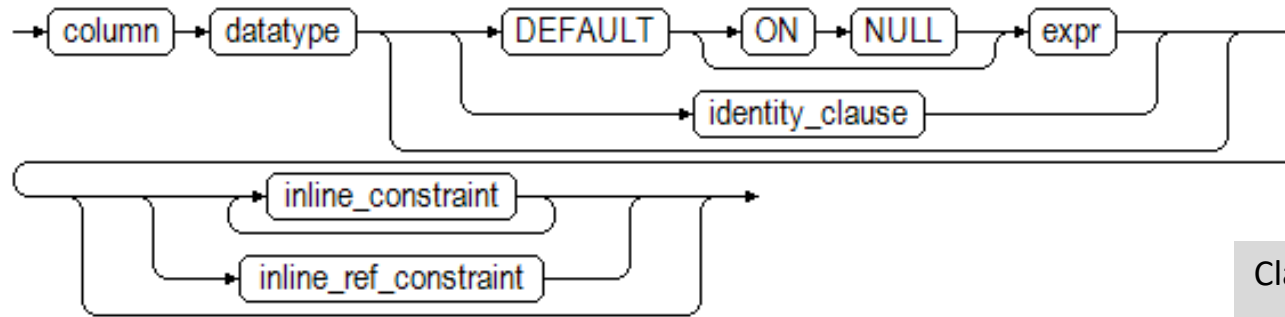
relational_properties



Cláusulas:

- [column definition](#)
- [out of line constraint](#)
- [out of line ref constraint](#)

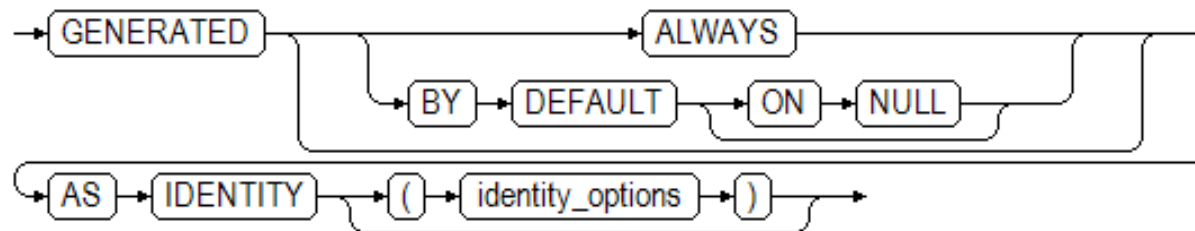
column definition



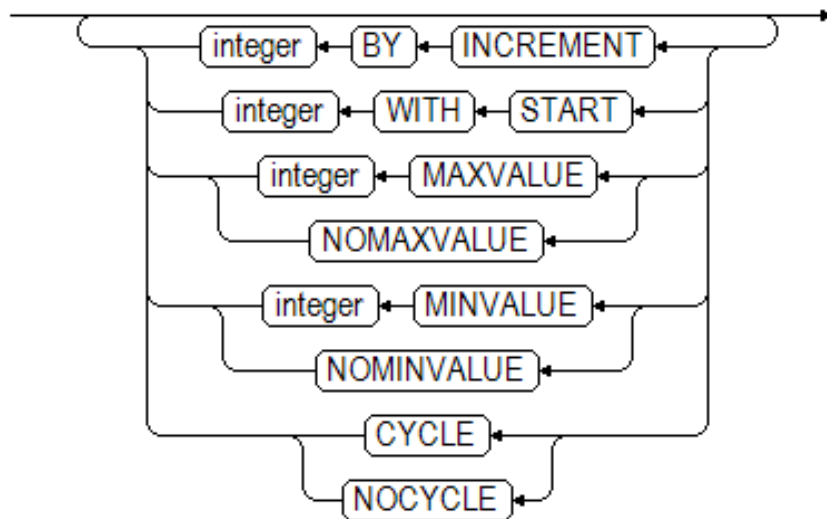
Cláusulas:

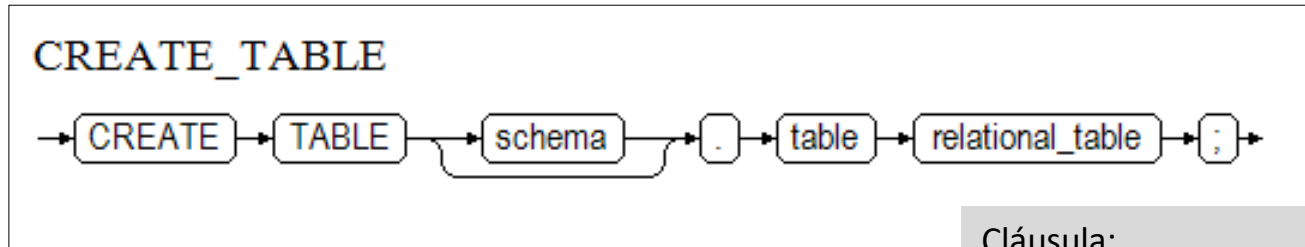
- [identity clause](#)
- [inline_constraint](#)
- [inline_ref_constraint](#)

identity_clause



identity_options





Cláusula:

- [relational_table](#)

▪ **schema**

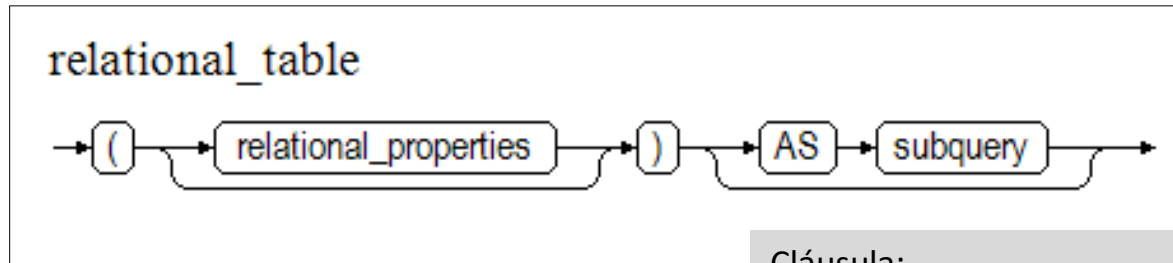
- Opcional
- Especifica:
 - Nome do esquema de BD para armazenar a tabela criada.
- Omitida:
 - Tabela é criada no esquema de BD da sessão.

▪ **table**

- Especifica:
 - Nome da tabela a criar.
- Nome da tabela:
 - [Regras](#) de definição.

▪ **Cláusula relational_table**

- Especifica:
 - Componentes da tabela relacional.

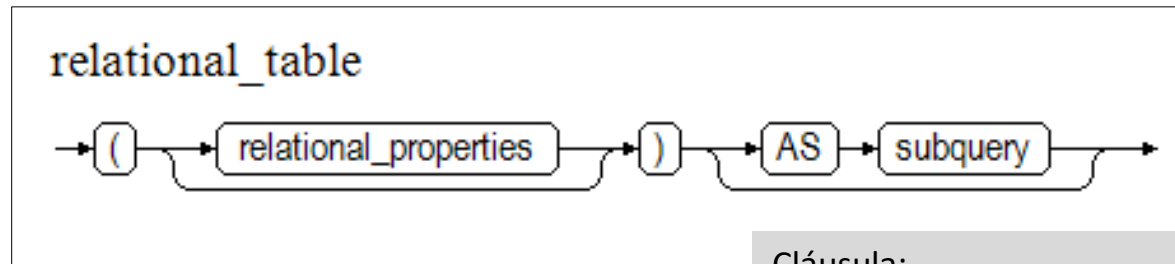


Cláusula:

- [relational_properties](#)

- **Cláusula `relational_properties`**

- Opcional
- Especifica:
 - Componentes da tabela relacional.



Cláusula:

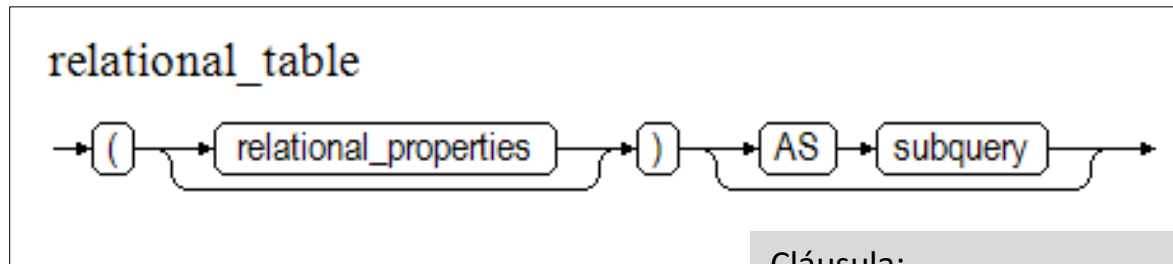
- [relational_properties](#)

- **Opcional**
- **Especifica uma Subquery**
 - Para determinar o conteúdo da tabela criada.
 - Linhas retornadas:
 - Inseridas na tabela depois de criada.

▪ Tabela Criada

- Tipos de dados e espaços de armazenamento:
 - Determinados a partir da subquery.
- Restrições de integridade
 - Não são transferidas para a tabela criada:
 - Restrições PRIMARY KEY, FOREIGN KEY e UNIQUE.
 - Restrições CHECK.
 - Valores por omissão de colunas (DEFAULT).
 - Restrições NOT NULL (criadas implicitamente).

Worksheet	Query Builder
	<pre> CREATE TABLE aluno_pt AS (SELECT a.num_aluno, a.nome FROM aluno a, pais p WHERE a.id_pais=p.id_pais AND UPPER(p.nome)='PORTUGAL'); </pre>



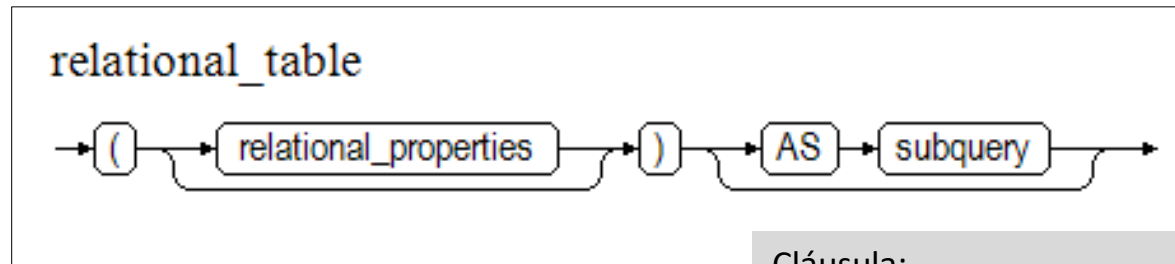
Cláusula:

- [relational_properties](#)

▪ Nomes das Colunas

- Podem ser omitidas:
 - Da definição da tabela (na cláusula relational_properties).
 - Se todas as expressões na subquery forem colunas. -- expressões na cláusula SELECT.

Worksheet	Query Builder
	<pre>CREATE TABLE aluno_pt AS (SELECT a.num_aluno, a.nome FROM aluno a, pais p WHERE a.id_pais=p.id_pais AND UPPER(p.nome)='PORTUGAL');</pre>



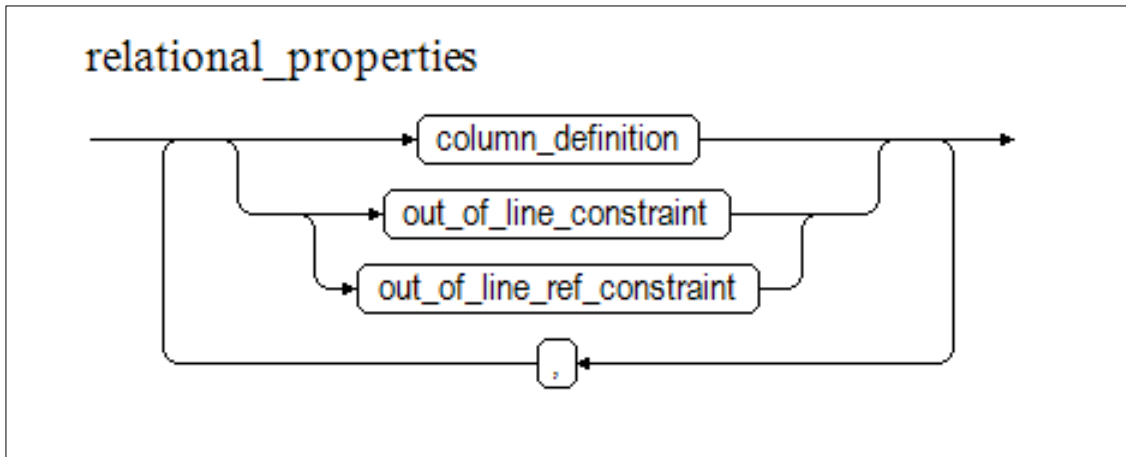
Cláusula:

- [relational_properties](#)

▪ Restrições na Definição da *Subquery*:

- Nº de colunas da tabela ... especificadas na cláusula *relational_properties*:
 - Deve ser igual ao nº de expressões na *subquery*.
- Definição de colunas:
 - Pode especificar apenas:
 - Nomes de colunas.
 - Valores por omissão (DEFAULT).
 - Restrições de integridade.
 - Não pode especificar:
 - Tipos de dados.
- Restrição FOREIGN KEY
 - Ilegal (em geral) CREATE TABLE com *AS subquery*
 - Alternativa:
 - Criar tabela sem esta restrição ... e depois adicioná-la com comando ALTER TABLE.

Worksheet	Query Builder
	<pre> CREATE TABLE aluno_pt AS (SELECT a.num_aluno, a.nome FROM aluno a, pais p WHERE a.id_pais=p.id_pais AND UPPER(p.nome)='PORTUGAL'); </pre>



Cláusulas:

- [column_definition](#)
- [out of line constraint](#)
- [out of line ref constraint](#)

▪ Cláusula **column_definition**

- Especifica:
 - Características de coluna da tabela relacional.

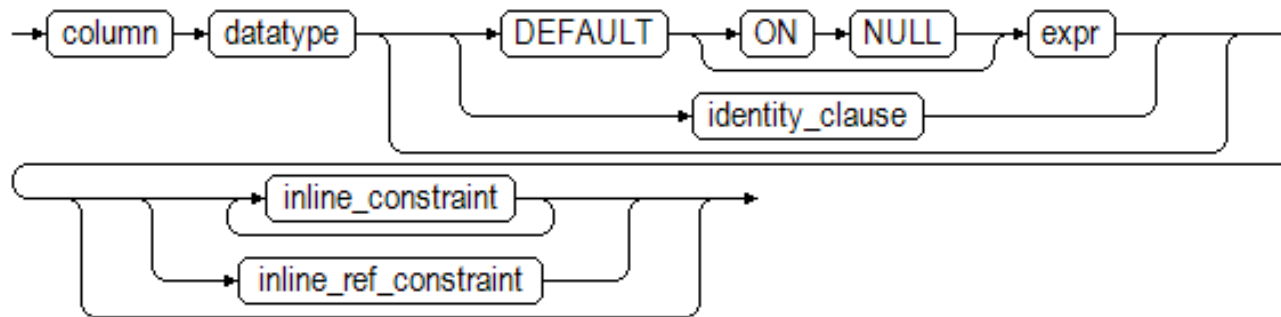
▪ Cláusula **out_of_line_constraint**

- Especifica:
 - Restrições *out_of_line*.

▪ Cláusula **out_of_line_ref_constraint**

- Especifica:
 - Restrições FOREIGN KEY *out_of_line*.

column_definition



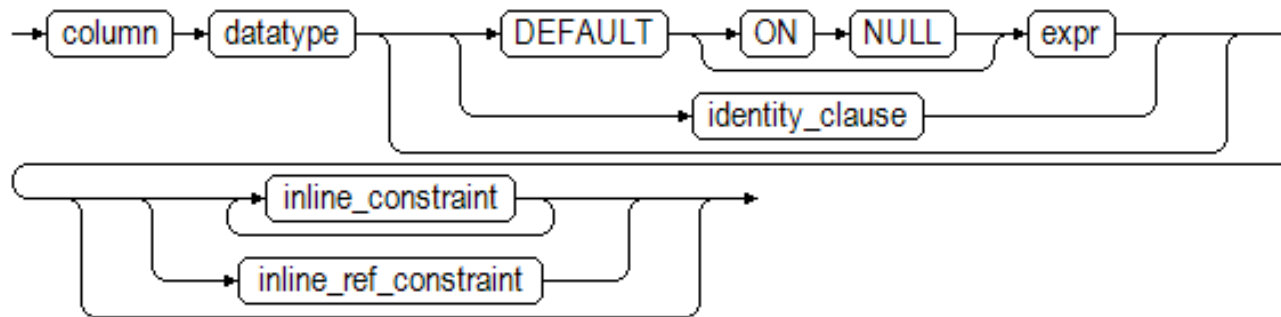
Cláusulas:

- [identity clause](#)
- [inline constraint](#)
- [inline ref constraint](#)

▪ column

- Especifica:
 - Nome de uma coluna da tabela.
- Pode ser omitido:
 - Se especificado *AS subquery*.
- Nº máximo de colunas numa tabela:
 - 1000

column_definition



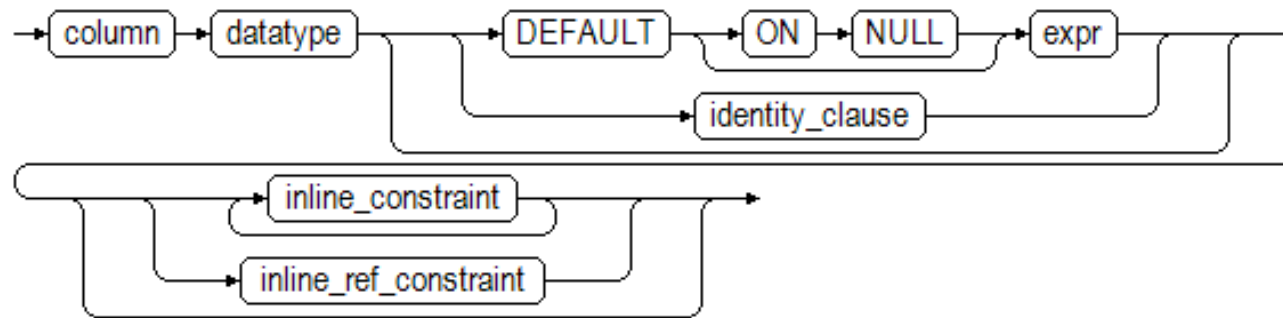
Cláusulas:

- [identity clause](#)
- [inline constraint](#)
- [inline ref constraint](#)

▪ datatype

- Especifica:
 - Tipo de dados (domínio) de uma coluna.
- Pode ser omitido em coluna ... que faz parte de chave estrangeira:
 - Atribuído (automaticamente):
 - Tipo de dados da coluna correspondente da chave referenciada na restrição de integridade referencial.
- Tem de ser omitido:
 - Se especificado AS *subquery*.

column_definition



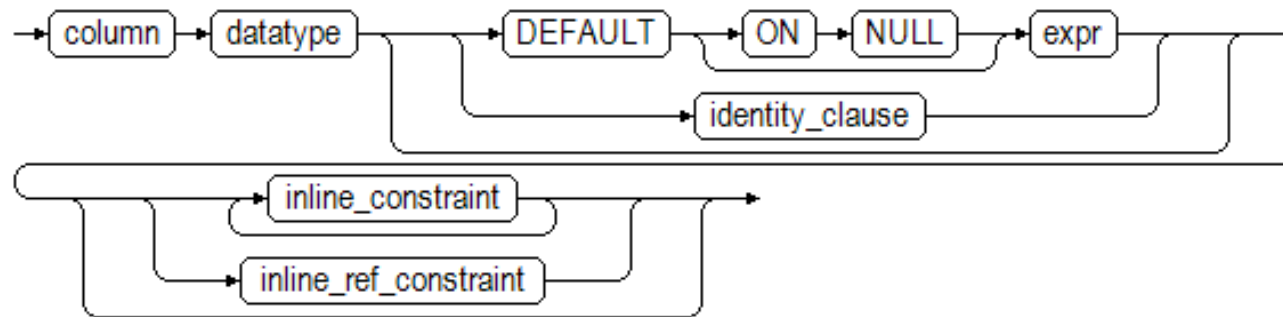
Cláusulas:

- [identity clause](#)
- [inline_constraint](#)
- [inline ref constraint](#)

▪ Cláusula **DEFAULT** (1/4)

- Opcional
- Especifica:
 - O valor atribuído à coluna ... quando o comando INSERT omite um valor para a coluna.
 - Tipo de dados = datatype.

column_definition



Cláusulas:

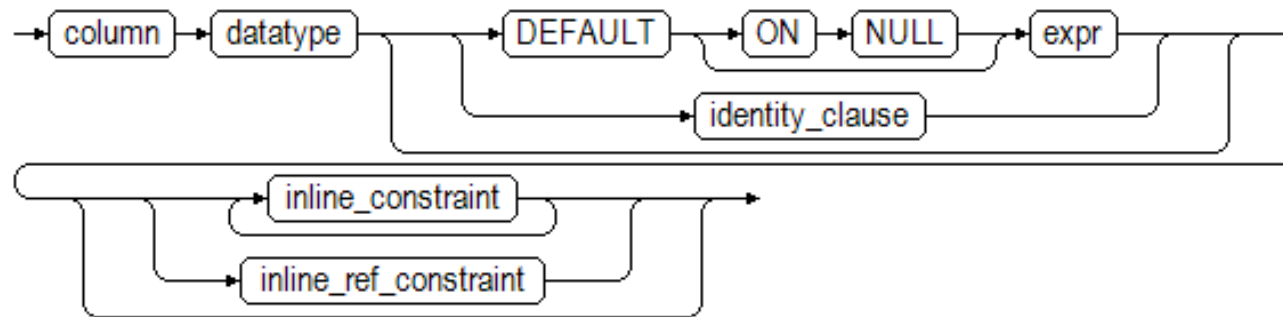
- [identity clause](#)
- [inline constraint](#)
- [inline ref constraint](#)

▪ Cláusula **DEFAULT** (2/4)

▪ Cláusula **ON NULL**:

- Opcional.
- Especifica:
 - O valor por omissão atribuído à coluna ... quando o comando INSERT tenta inserir na coluna o valor NULL.
- Quando é especificada:
 - Origina a especificação implícita da restrição NOT NULL.
 - Se for especificada explicitamente uma restrição NOT NULL que gera um conflito com a referida no ponto anterior, é gerado um erro.

column_definition



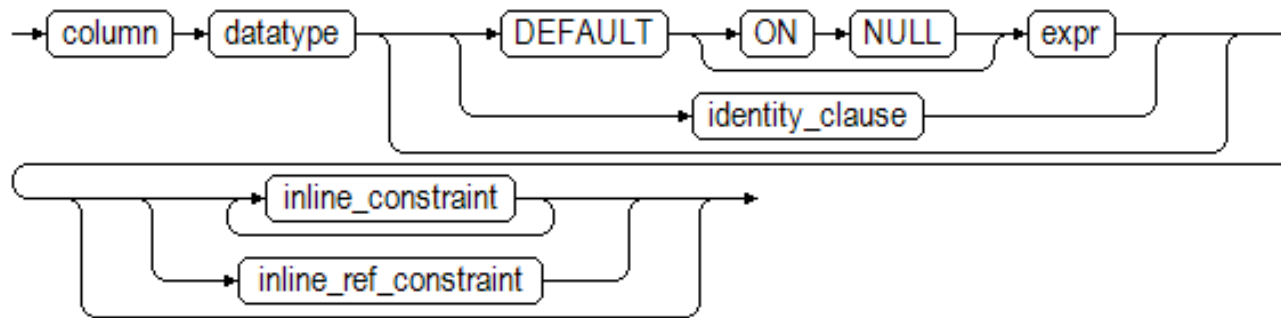
Cláusulas:

- [identity clause](#)
- [inline constraint](#)
- [inline ref constraint](#)

▪ Cláusula **DEFAULT** (3/4)

- Expressão **expr**:
 - Pode incluir **funções** SQL.
 - **Não** pode ser:
 - Uma expressão **subquery** escalar.
 - *Subquery* que retorna exatamente o valor de uma coluna de uma linha.

column_definition



Cláusulas:

- [identity clause](#)
- [inline constraint](#)
- [inline ref constraint](#)

▪ Cláusula **DEFAULT** (4/4)

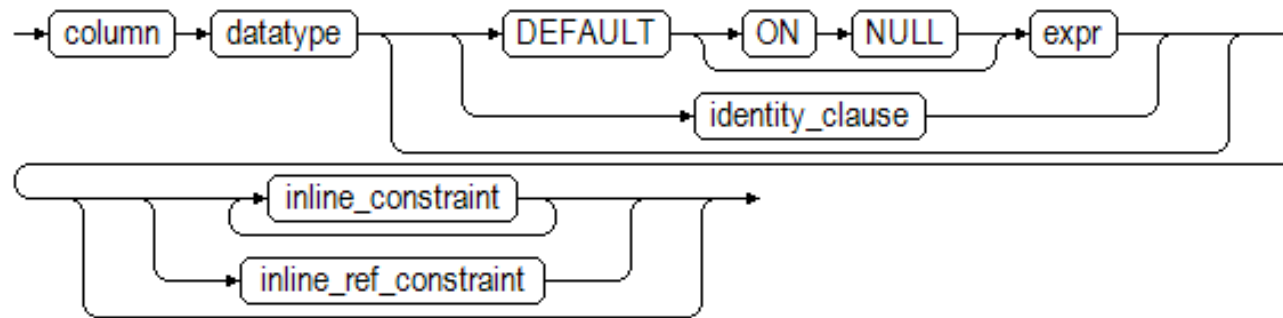
▪ Exemplo:

```

Worksheet  Query Builder
CREATE TABLE empregado (
  id_empregado    NUMBER CONSTRAINT pk_empregado_id PRIMARY KEY,
  nome            VARCHAR2(25) CONSTRAINT nn_empregado_nome NOT NULL,
  id_departamento NUMBER DEFAULT ON NULL 50 NOT NULL
)
    
```

Atribuído 50 em vez de NULL.

column_definition

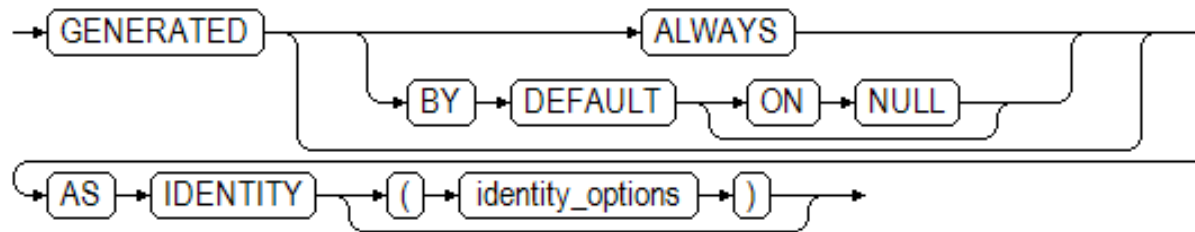


Cláusulas:

- [identity clause](#)
- [inline constraint](#)
- [inline ref constraint](#)

- **Cláusula `identity_clause`**
 - Opcional
 - Especifica:
 - Uma coluna identidade (id). -- com números inteiros únicos gerados automaticamente.
- **Cláusula `inline_constraint`**
 - Opcional
 - Especifica:
 - Restrição de integridade *inline*.
- **Cláusula `inline_ref_constraint`**
 - Opcional
 - Especifica:
 - Restrição de integridade referencial (chave estrangeira) *inline*.

identity_clause



Cláusula:

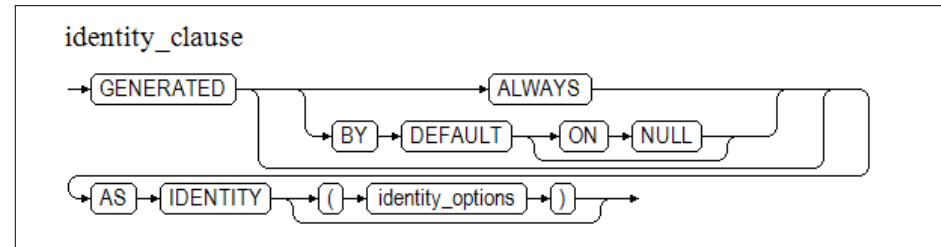
- [identity_options](#)

▪ identity_clause

- Especifica:
 - Coluna identidade (id). -- com números inteiros únicos gerados automaticamente.

Coluna identidade

- Tipo:
 - Auto-incremento.
- Contém:
 - Números inteiros:
 - Atribuídos **automaticamente** pelo comando INSERT.
 - Gerados por uma sequência.



Sequência:

- É um objeto da BD.
- Gera números inteiros únicos.
- Após gerar um número:
 - Sequência é incrementada / decrementada conforme configuração predefinida.

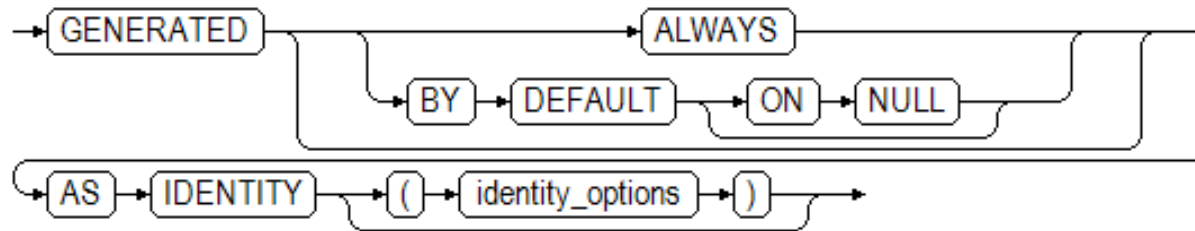
Exemplo:

```

Worksheet  Query Builder
CREATE TABLE empregado (
  id_empregado NUMBER GENERATED BY DEFAULT AS IDENTITY (START WITH 100 INCREMENT BY 10),
  nome VARCHAR2(30),
  CONSTRAINT pk_empregado_id PRIMARY KEY (id_empregado)
);
    
```

Cláusula introduzida na **versão 12c** da BD Oracle.
Ilegal na versão 11g.

identity_clause

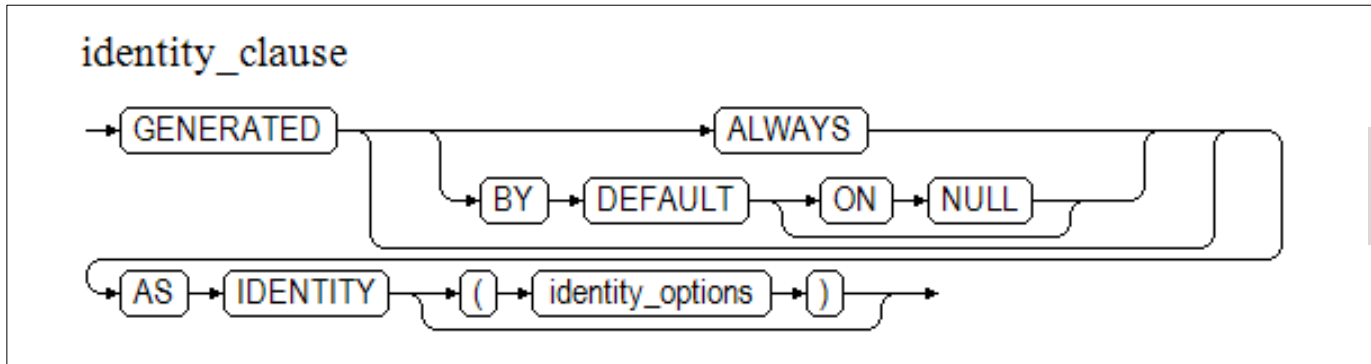


Cláusula:

- [identity_options](#)

▪ Limitações da Coluna Identidade (id)

- Permitida apenas **uma** coluna id por tabela.
- Tipo de dados da coluna:
 - Tem de ser numérico.
- Cláusula DEFAULT da coluna:
 - É ilegal.
- Restrição NOT NULL da coluna:
 - Especificada implicitamente.



Cláusula:

- [identity_options](#)

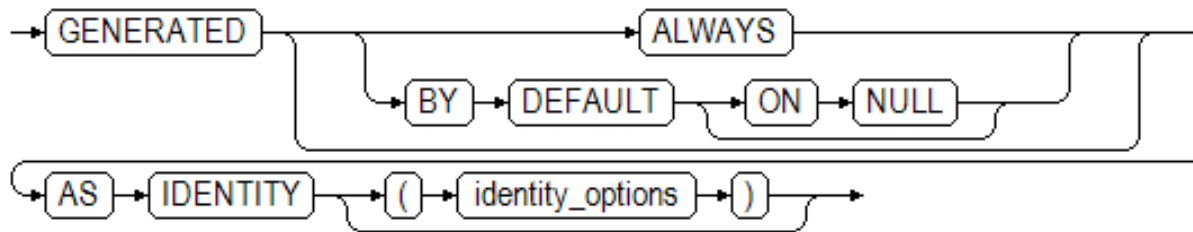
▪ Cláusula **ALWAYS**:

- Opcional
- Especifica:
 - Valor atribuído à coluna:
 - Feito **sempre** pelo gerador de sequência. -- funcionamento por omissão.
 - Feito por INSERT / UPDATE retorna um erro.

▪ Cláusula **BY DEFAULT**:

- Opcional
- Semelhante a **ALWAYS**, excepto:
 - Possível atribuir (ex: INSERT/UPDATE) também, explicitamente, um valor específico à coluna.
- Com **ON NULL**:
 - Usado gerador de sequência para atribuir um valor à coluna ... quando INSERT tenta atribuir um valor NULL.

identity_clause



Cláusula:

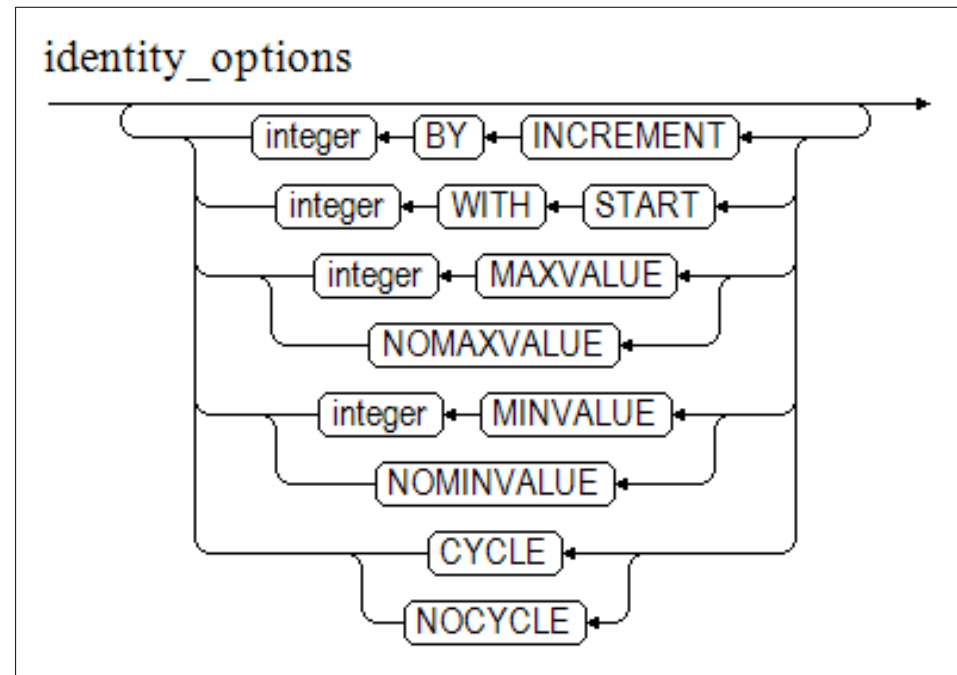
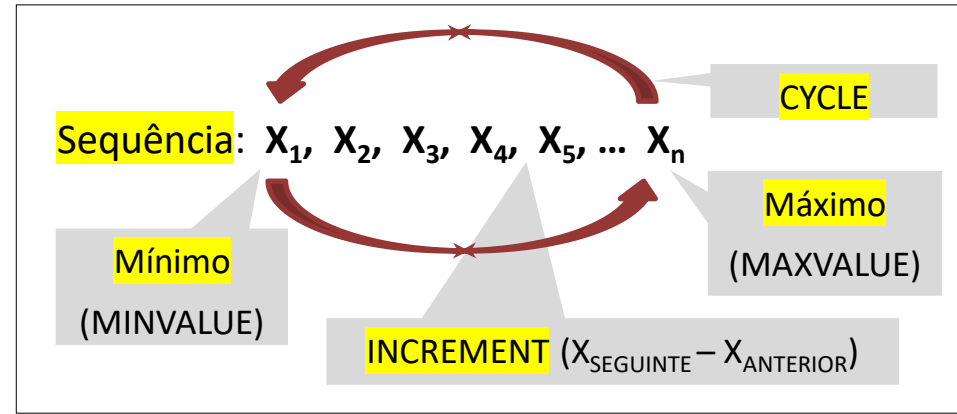
- [identity_options](#)

▪ Cláusula **identity_options**

- Opcional
- Especifica:
 - Configuração do gerador de sequência.
- Quando é omitida:
 - Criada sequência:
 - Crescente
 - 1º valor: 1
 - Incremento: 1

identity_options (simplificado) (1/6)

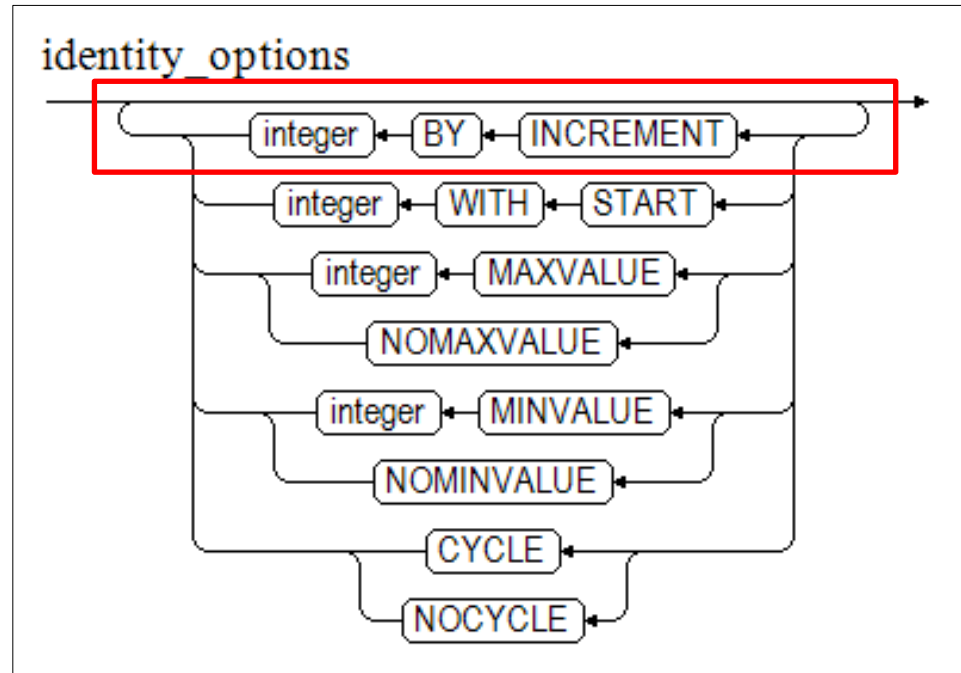
- **Especifica:**
 - Parâmetros de configuração do gerador de sequência.
- **Parâmetros de configuração**
 - Iguais aos parâmetros de:
 - Comando CREATE SEQUENCE.
 - Podem ser especificados:
 - Múltiplos parâmetros.
 - Todos os parâmetros:
 - **Opcionais.**
 - Por omissão:
 - INCREMENT BY 1
 - NOMAXVALUE
 - NOMINVALUE
 - NOCYCLE
 - NOCACHE



identity_options (simplificado) (2/6)

Parâmetro INCREMENT BY integer

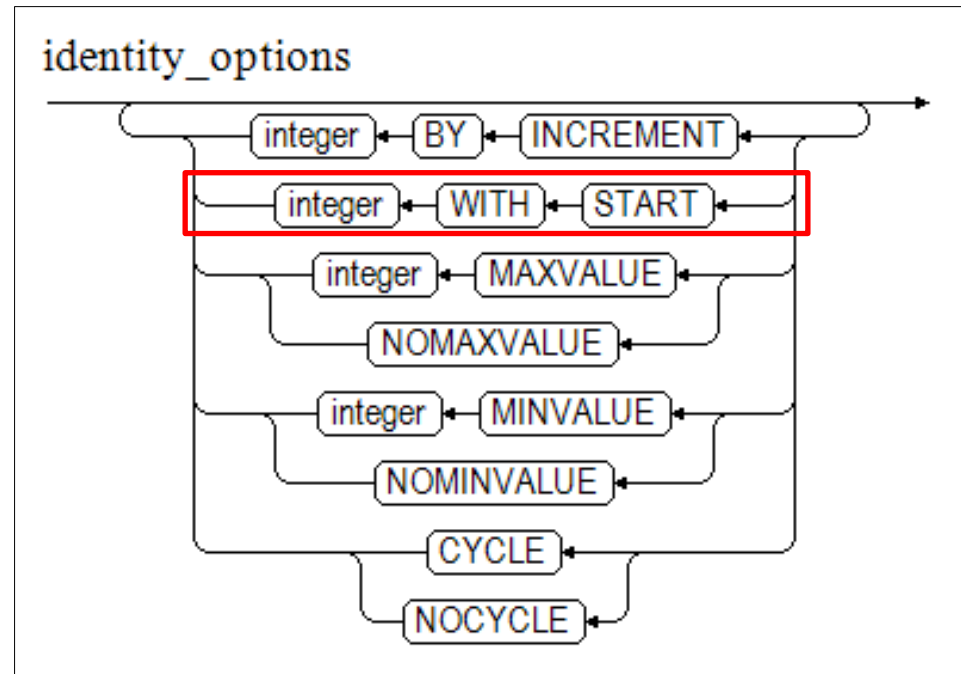
- Especifica:
 - Intervalo entre números consecutivos da sequência.
- Quando omitido:
 - Intervalo por omissão: 1
- integer:
 - Nº inteiro > 0:
 - Nº dígitos < 28.
 - Sequência crescente.
 - Nº inteiro < 0:
 - Nº dígitos < 27
 - Sequência decrescente.
 - ilegal 0.
 - Valor absoluto:
 - < valor_absoluto(MAXVALUE – MINVALUE)
- Exemplo:
 - INCREMENT BY -1
 - Especifica sequência: decrescente ; 1º valor: -1 ; “sem” limite inferior.



identity_options (simplificado) (3/6)

Parâmetro START WITH integer

- Especifica:
 - 1º valor da sequência gerado.
- Quando omitido:
 - Número inicial:
 - Sequência ascendente:
 - Nº mínimo.
 - Sequência descendente:
 - Nº máximo.
- integer:
 - Nº inteiro >0:
 - Nº dígitos < 28.
 - Nº inteiro <0:
 - Nº dígitos < 27
 - Nº inteiro:
 - >= MINVALUE
 - <= MAXVALUE
- Em sequências que se repetem (CYCLE):
 - integer pode não ser o número após alcançado o número máximo/mínimo.



identity_options (simplificado) (4/6)

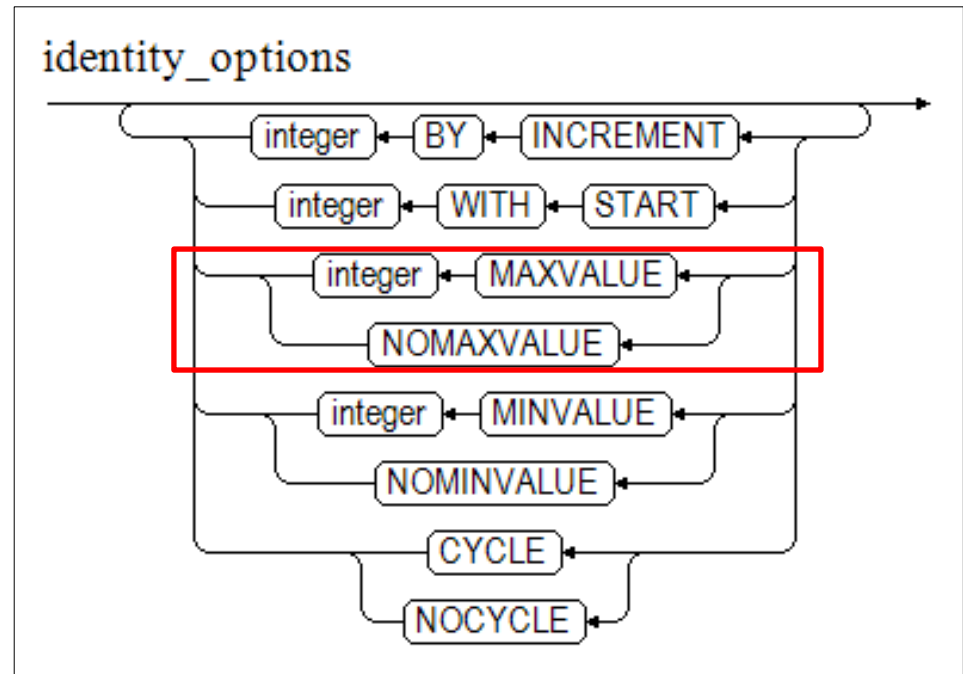
Parâmetro MAXVALUE integer

Especifica:

- Número máximo que a sequência pode gerar.

integer:

- Nº inteiro >0:
 - Nº dígitos < 28.
- Nº inteiro <0:
 - Nº dígitos < 27
- Nº inteiro:
 - >= START WITH
 - > MINVALUE



Parâmetro NOMAXVALUE

Especifica:

- Número máximo:
 - $10^{28}-1$ sequência crescente.
 - 1 sequência decrescente.

Parâmetro por omissão:

- NOMAXVALUE

identity_options (simplificado) (5/6)

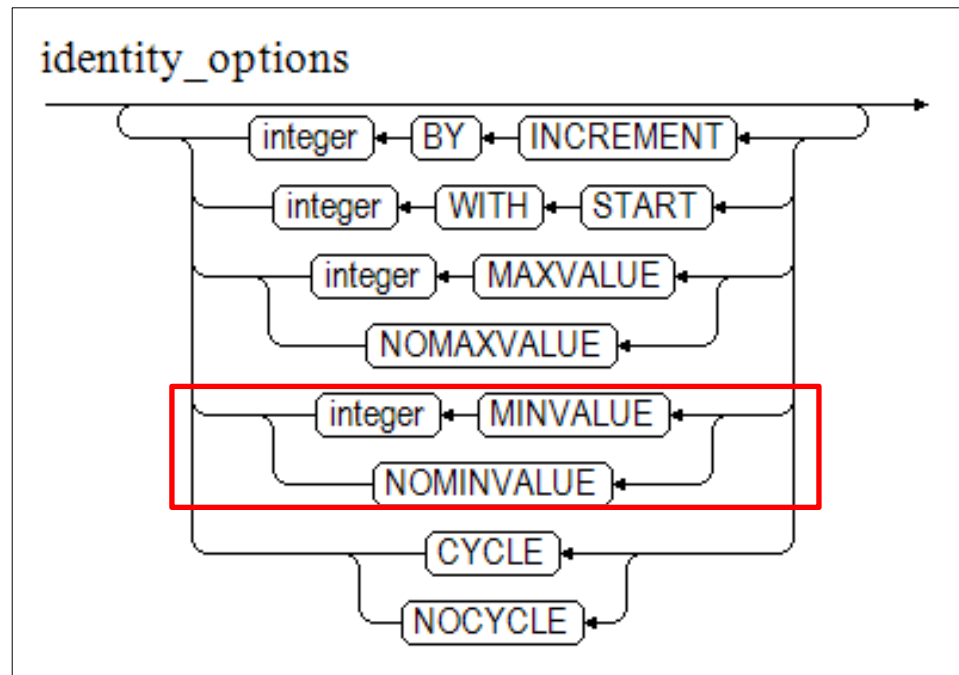
Parâmetro MINVALUE integer

Especifica:

- Número mínimo que a sequência pode gerar.

integer:

- Nº inteiro >0:
 - Nº dígitos < 28.
- Nº inteiro <0:
 - Nº dígitos < 27
- Nº inteiro:
 - <= START WITH
 - < MAXVALUE



Parâmetro NOMINVALUE

Especifica:

Número mínimo:

- 1 sequência crescente.
- $-(10^{28}-1)$ sequência decrescente.

Parâmetro por omissão:

- NOMINVALUE

identity_options (simplificado) (6/6)

Parâmetro CYCLE

Especifica:

- Sequência continua a gerar números depois de alcançar o número máximo/mínimo.
- Sequência ascendente
 - Após atingir nº máximo:
 - Gera nº mínimo.
- Sequência descendente
 - Após atingir nº mínimo:
 - Gera nº máximo.

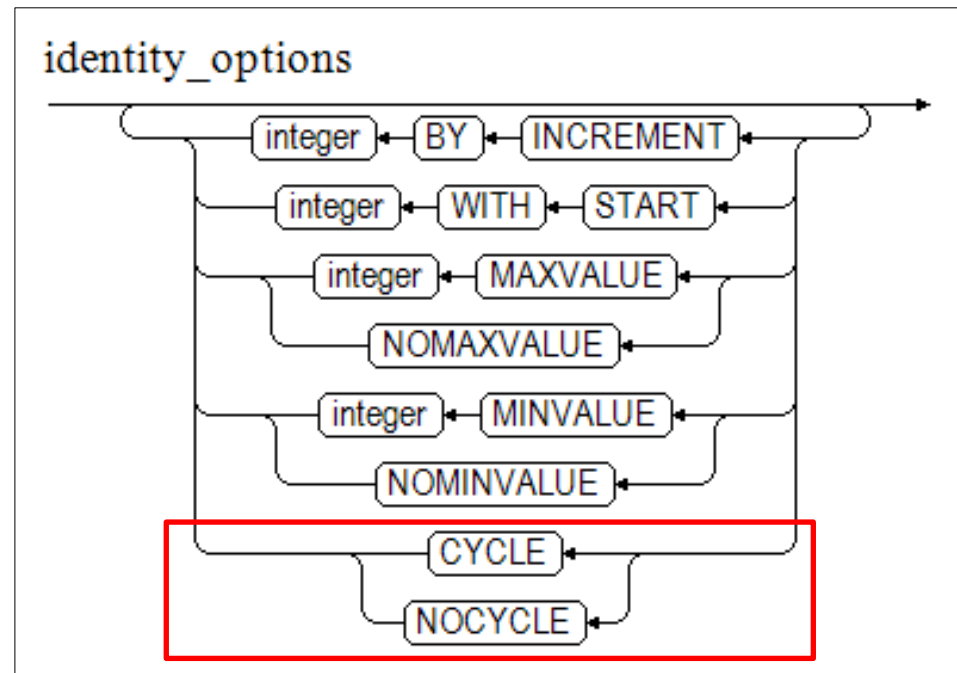
Parâmetro NOCYCLE

Especifica:

- Sequência não pode gerar mais números após alcançar o número máximo/mínimo.

Parâmetro por omissão:

- NOCYCLE



Exemplos

Worksheet Query Builder

```
CREATE TABLE pessoa (  
  id_pessoa NUMBER GENERATED AS IDENTITY,  
  nome VARCHAR2(25) CONSTRAINT nn_pessoa_nome NOT NULL,  
  CONSTRAINT pk_pessoa_id PRIMARY KEY (id_pessoa)  
);
```

Gerador de sequência:

- Atribui sempre à coluna id_pessoa um valor inteiro incrementado.
- Valor inicial = 1.
- Incremento = 1.

Worksheet Query Builder

```
CREATE TABLE tabela (  
  id NUMBER GENERATED BY DEFAULT AS IDENTITY (START WITH 100 INCREMENT BY 10)  
);
```

Gerador de sequência:

- Atribui sempre à coluna id um valor inteiro incrementado.
- Valor inicial = 100.
- Incremento = 10.

Comandos DDL

Introdução

Restrições de Integridade

CREATE TABLE

ALTER TABLE

DROP TABLE

ALTER USER

- [Introdução](#)
- [Sintaxe](#)

- **Interesse**

- Alterar a **especificação** de uma tabela.

- **Exemplos**

- Alterar **Colunas**:

- Adicionar

```
ALTER TABLE alunos  
ADD email VARCHAR(100) ;
```

- Alterar tipo de dados

```
ALTER TABLE alunos  
MODIFY email VARCHAR(75) ;
```

- Eliminar

```
ALTER TABLE alunos  
DROP COLUMN email ;
```

- **Restrições**:

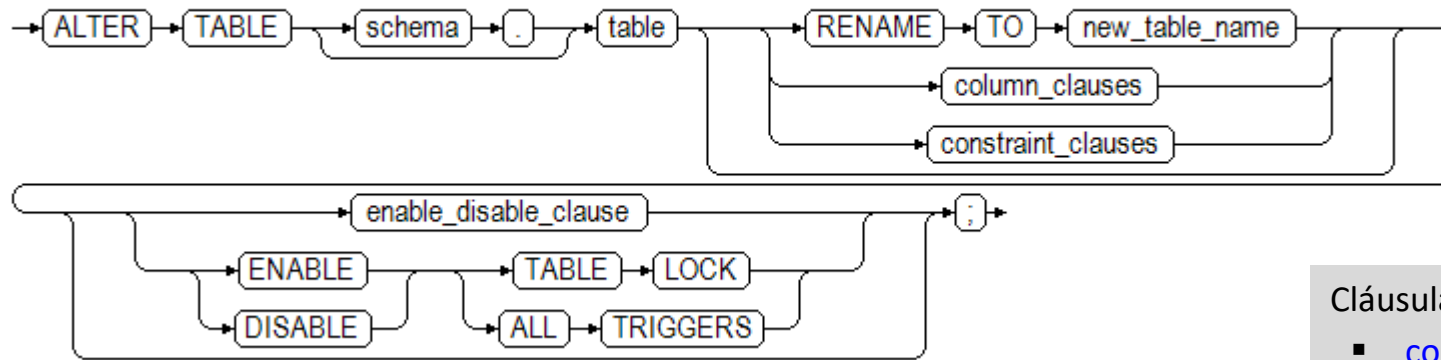
- Adicionar

```
ALTER TABLE disciplinas  
ADD CONSTRAINT fk_disciplinas_id_aluno  
          FOREIGN KEY (id_aluno) REFERENCES alunos(id_aluno);
```

- Eliminar

```
ALTER TABLE disciplinas  
DROP CONSTRAINT fk_disciplinas_id_alunos;
```

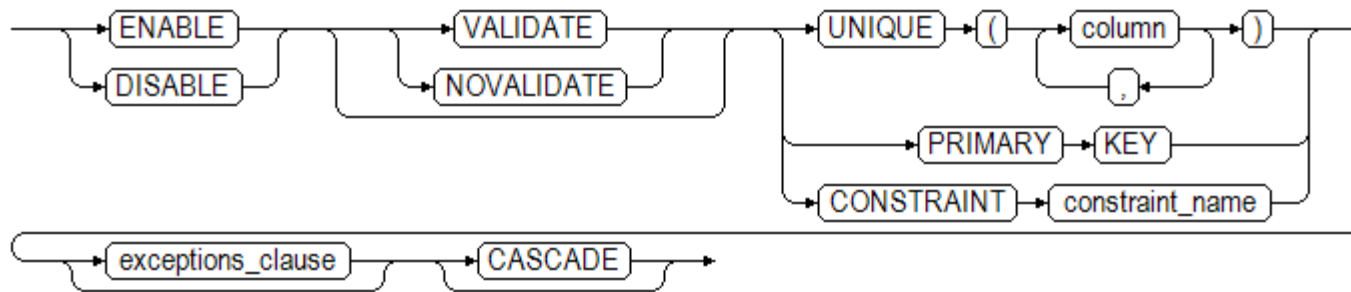
ALTER_TABLE



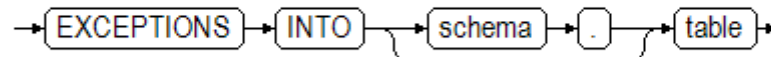
Cláusulas:

- [column clauses](#)
- [constraint clauses](#)
- [enable disable clause](#)

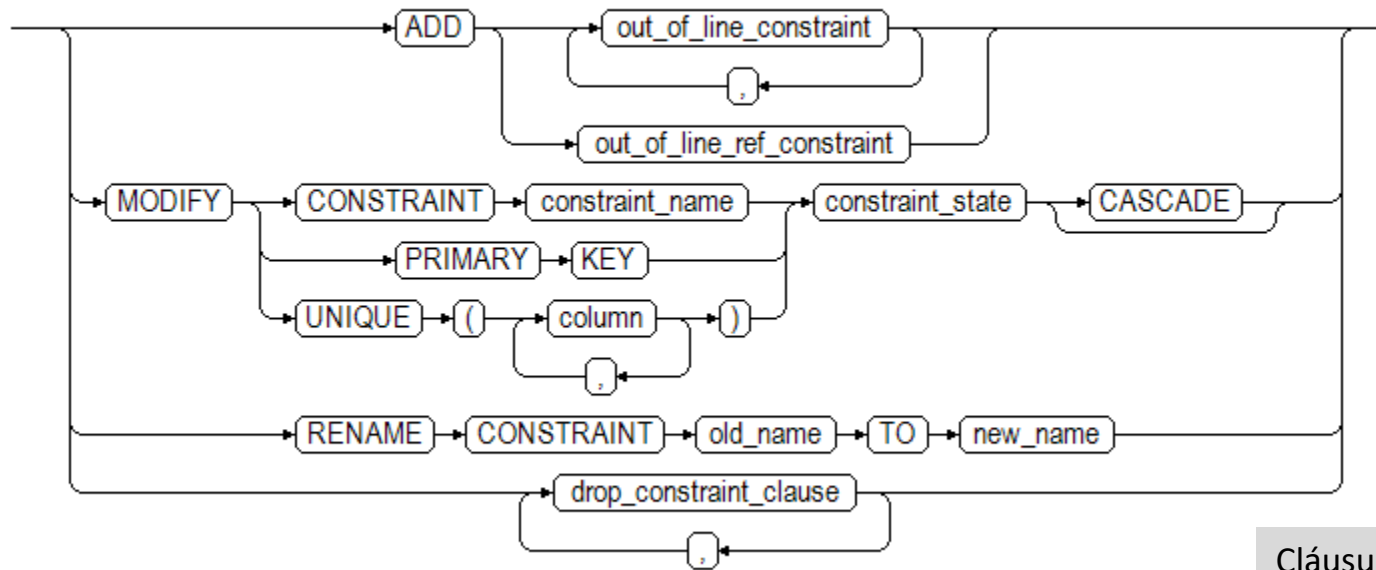
enable_disable_clause



exception_clause



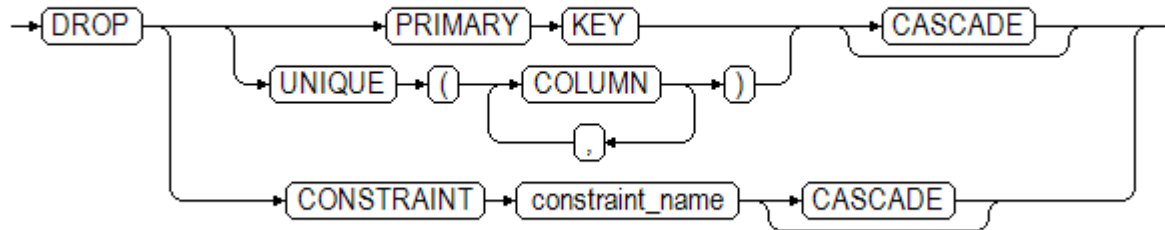
constraint_clauses



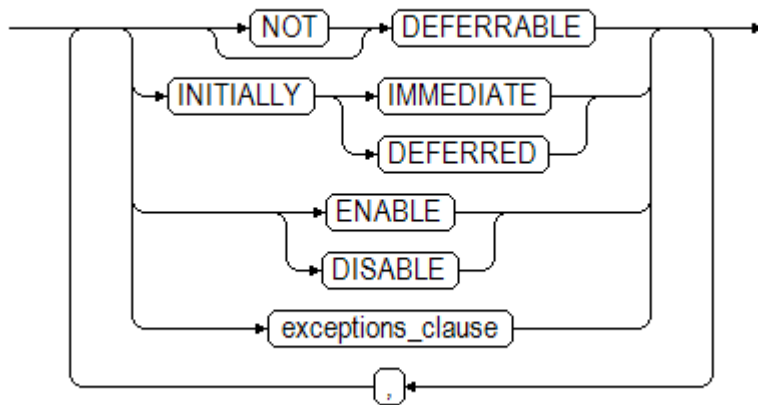
Cláusulas:

- [out of line constraint](#)
- [out of line ref constraint](#)

drop_constraint_clause



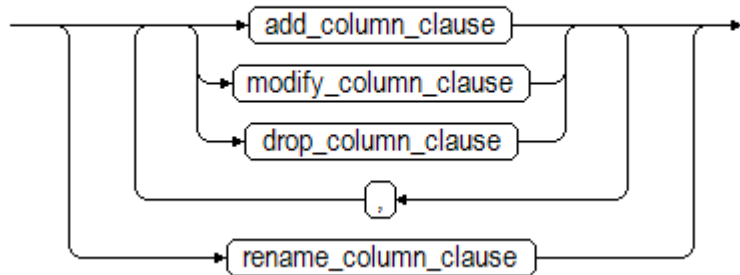
constraint_state



Cláusula:

- [exceptions clause](#)

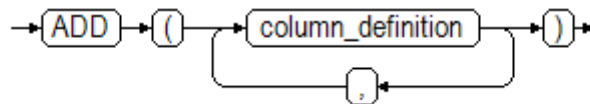
column_clauses



Cláusula:

- [modify column clause](#)
- [drop column clause](#)
- [rename column clause](#)

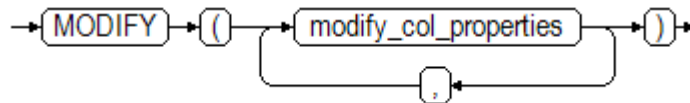
add_column_clause



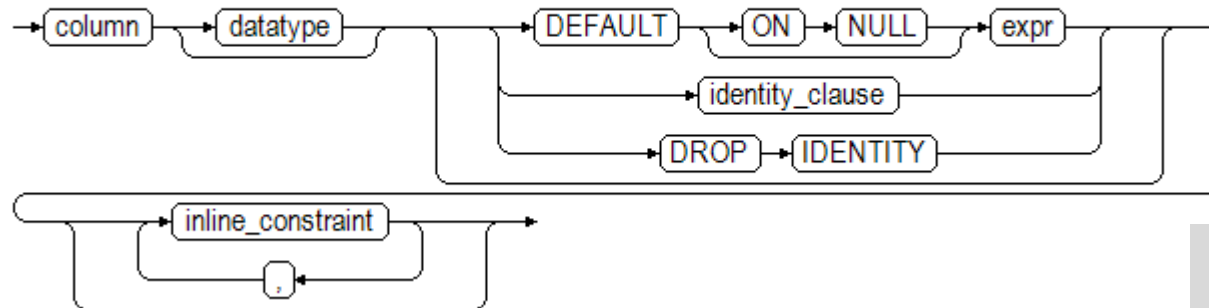
Cláusula:

- [column definition](#)

modify_column_clauses



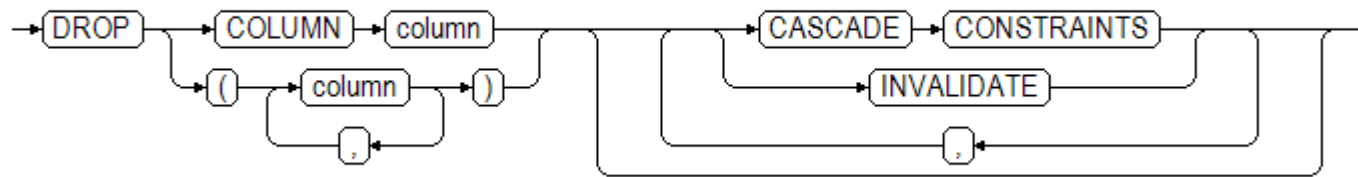
modify_col_properties



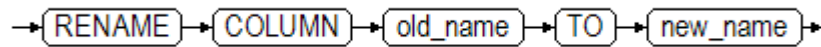
Cláusula:

- [identity_clause](#)

drop_column_clause



rename_column_clause



Comandos DDL

Introdução

Restrições de Integridade

CREATE TABLE

ALTER TABLE

DROP TABLE

ALTER USER

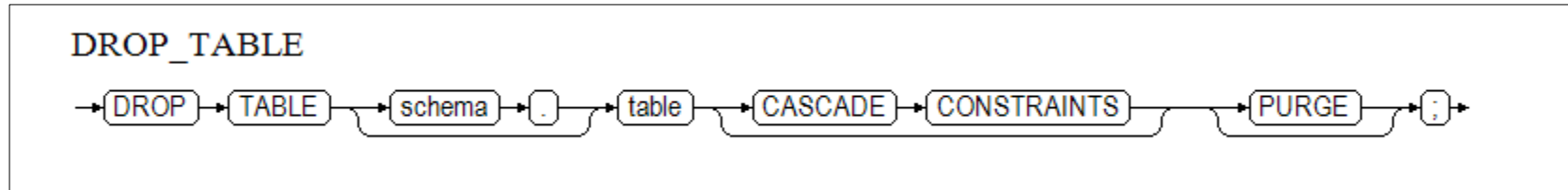
- [Introdução](#)
- Sintaxe
 - [Geral](#)
 - [Cláusula CASCADE CONSTRAINTS](#)
 - [Cláusula PURGE](#)

- Interesse

- Eliminar uma tabela:
 - Inclui a eliminação de:
 - Todas as linhas.
 - Todos os *triggers* definidos sobre a tabela.

- **Objetos Dependentes da Tabela Eliminada**

- Exemplos:
 - Vistas
 - *Stored procedure*
 - Funções
 - *Package*
- São apenas invalidados.
 - Não são eliminados.
- Só poderão voltar a ser usados se:
 - Tabela eliminada for recriada;
ou
 - Forem recriados.
 - Passarem a ser independentes da tabela eliminada.



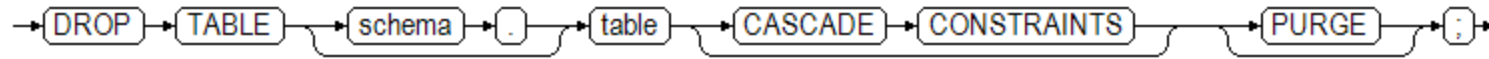
▪ schema

- Opcional
- Especifica:
 - Nome do esquema de BD que contém a tabela a eliminar.
- Quando é omitido:
 - Assumida tabela no esquema de BD da sessão.

▪ table

- Especifica:
 - Nome da tabela a eliminar.

DROP_TABLE



Cláusula CASCADE CONSTRAINTS

- Opcional
- Especifica:
 - Eliminar **todas** as **restrições de integridade referencial** (chaves estrangeiras) que referem as chaves primárias e *unique* da tabela eliminada.

Exemplo:

Worksheet	Query Builder
	<pre> CREATE TABLE demo_pai (id_pai NUMBER CONSTRAINT pk_demo_pai_id PRIMARY KEY); CREATE TABLE demo_filho (id_filho NUMBER CONSTRAINT pk_demo_filho_id PRIMARY KEY, id_pai NUMBER CONSTRAINT fk_demo_filho_pai REFERENCES demo_pai(id_pai)); </pre>

Worksheet	Query Builder
	<pre> DROP TABLE demo_pai CASCADE CONSTRAINTS; </pre>

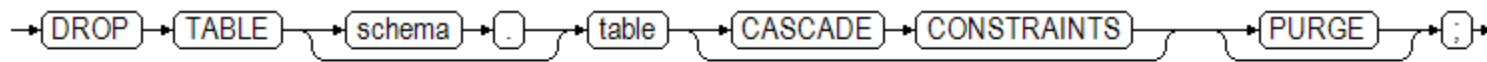
Definição
equivalente
da tabela
atualizada.

Worksheet	Query Builder
	<pre> CREATE TABLE demo_filho (id_filho NUMBER CONSTRAINT pk_demo_filho_id PRIMARY KEY, id_pai NUMBER); </pre>

Na tabela **demo_filho**:

- restrição **fk_demo_filho** é **eliminada**.
- Valores da coluna **id_pai** mantêm-se.

DROP_TABLE



Cláusula CASCADE CONSTRAINTS

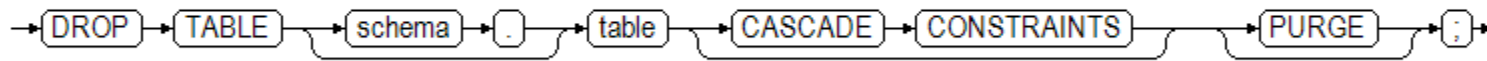
Quando é omitida:

- Se existirem as restrições de integridade referencial referidas anteriormente, a BD:
 - Retorna um erro.
 - Não elimina a tabela.

Worksheet	Query Builder
	<pre> CREATE TABLE demo_pai(id_pai NUMBER CONSTRAINT pk_demo_pai_id PRIMARY KEY); CREATE TABLE demo_filho (id_filho NUMBER CONSTRAINT pk_demo_filho_id PRIMARY KEY, id_pai NUMBER CONSTRAINT fk_demo_filho_pai REFERENCES demo_pai(id_pai)); </pre>

Worksheet	Query Builder
	<pre>DROP TABLE demo_pai;</pre>
<p>Script Output x Query Result x</p> <p>Task completed in 0,015 seconds</p> <p>Error starting at line : 47 in command - DROP TABLE demo_pai Error report - SQL Error: ORA-02449: unique/primary keys in table referenced by foreign keys 02449. 00000 - "unique/primary keys in table referenced by foreign keys"</p>	

DROP_TABLE



Cláusula PURGE

- Opcional
- Quando é omitida:
 - Tabela e objetos dependentes são colocados no *recycle bin* da BD.
- Especifica:
 - Libertar todo o espaço de armazenamento usado pela tabela eliminada, num único passo.
 - Tabela e seus objetos dependentes não são colocados no *recycle bin* da BD.
 - Equivalente a dois passos:
 1. Eliminar tabela.
 2. Eliminar a tabela do *recycle bin* da BD.
- Quando é usada:
 - Tabela eliminada é irre recuperável. -- impossível o *roll back* do comando DROP TABLE.

Worksheet	Query Builder
	<pre>DROP TABLE demo_pai; DROP TABLE demo_pai CASCADE CONSTRAINTS; DROP TABLE demo_pai PURGE; DROP TABLE demo_pai CASCADE CONSTRAINTS PURGE;</pre>

Comandos DDL

Introdução

Restrições de Integridade

CREATE TABLE

ALTER TABLE

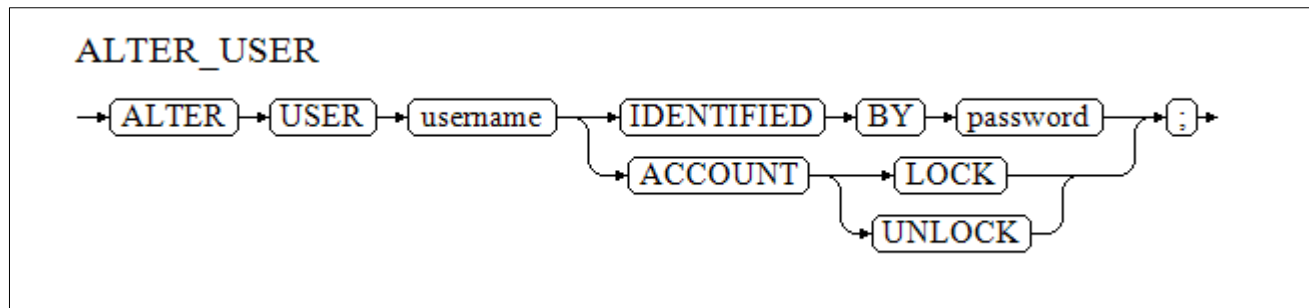
DROP TABLE

ALTER USER

▪ Interesse

- Alterar a **autenticação / características** de um utilizador da BD.

▪ Sintaxe Simplificada



▪ Exemplos

- Alterar **Password**:

Worksheet	Query Builder
1	ALTER USER BDDAD_DA1
2	IDENTIFIED BY Bd2017;

- **Acesso** à BD

- **Bloquear**:

Worksheet	Query Builder
6	ALTER USER BDDAD_DA1
7	ACCOUNT LOCK;

- **Desbloquear**:

Worksheet	Query Builder
11	ALTER USER BDDAD_DA1
12	ACCOUNT UNLOCK;

Requisito:

- Privilégio de administração

Comandos DML

Introdução

INSERT

UPDATE

DELETE

- **Interesse**

- Manipular dados das tabelas
 - Inserir
 - Alterar
 - Eliminar

- **Mais importantes**

- INSERT
- UPDATE
- DELETE
- SELECT -- em construção

Comandos DML

Introdução

INSERT

UPDATE

DELETE

- Inserir numa Tabela
 - Uma linha
 - Um conjunto de linhas de outra tabela

▪ Sintaxe

▪ Formato Longo

```
INSERT INTO nome_tabela (nome_coluna1, ..., nome_colunaN)  
VALUES (valor1, ..., valorN)
```

- Permite inserir valores:
 - Em **algumas colunas** de uma tabela.
- Colunas não especificadas:
 - Preenchidas com valor NULL.
 - Exceto **coluna identidade**.

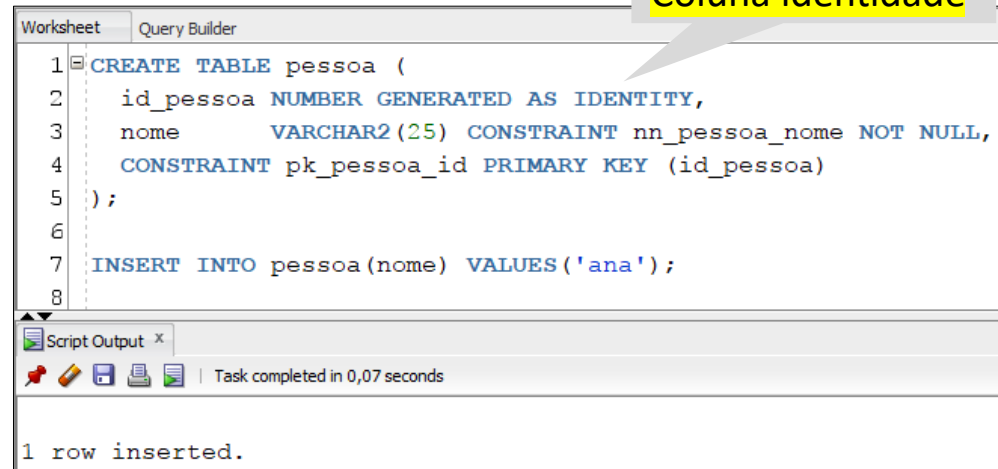
▪ Formato Abreviado

```
INSERT INTO nome_tabela  
VALUES (valor1, ..., valorN)
```

- Permite inserir valores:
 - Em **todas as colunas** de uma tabela. -- preciso fornecer um valor para cada coluna.
 - Pela **ordem de criação** na tabela.

▪ Falha

- Quando **valores violam as restrições** na tabela/coluna correspondente.



Coluna identidade

```
1 CREATE TABLE pessoa (  
2   id_pessoa NUMBER GENERATED AS IDENTITY,  
3   nome      VARCHAR2(25) CONSTRAINT nn_pessoa_nome NOT NULL,  
4   CONSTRAINT pk_pessoa_id PRIMARY KEY (id_pessoa)  
5 );  
6  
7 INSERT INTO pessoa (nome) VALUES ('ana');  
8
```

Script Output x
Task completed in 0,07 seconds
1 row inserted.

- **INSERT Combinado com SELECT**
 - SELECT não pode incluir a tabela onde são inseridas as linhas.

- **Sintaxe**

- **Formato longo**

```
INSERT INTO nome_tabela (nome_coluna1, ..., nome_colunaN)  
SELECT ...
```

- **Formato abreviado**

```
INSERT INTO nome_tabela  
SELECT ...
```


- Inserir Datas

- Depende do SGBD
- No Oracle:
 - Usa-se a função **TO_DATE** (data_string, formato_string)
 - Exemplo:

```
INSERT INTO tabela (numero, nome, data_nasc)  
VALUES (1, 'Ana', TO_DATE('02/07/1961', 'dd/mm/yyyy') );
```

Comandos DML

Introdução

INSERT

UPDATE

DELETE

- Interesse

- Atualizar valores de uma tabela

- Sintaxe

```
UPDATE nome_tabela
SET  coluna1 = { expressão1 | query1 },
     coluna2 = { expressão2 | query2 }
...
[WHERE condição];
```

- Cláusula WHERE
 - Opcional
 - Especifica as linhas da tabela que o comando deve atualizar.

- Exemplo

```
UPDATE empregados
SET vencimento = vencimento * 1.05
WHERE avaliacao = 5;
```

Aumentar em 5% os
vencimentos dos empregados
cuja avaliação é 5.

Comandos DML

Introdução

INSERT

UPDATE

DELETE

- Eliminar **Linhas** de uma Tabela

- Uma linha
- Um conjunto de linhas

Para **eliminar** uma **tabela**:

- Comando DROP TABLE.

Worksheet	Query Builder
	<pre>DROP TABLE demo_pai; DROP TABLE demo_pai CASCADE CONSTRAINTS; DROP TABLE demo_pai PURGE; DROP TABLE demo_pai CASCADE CONSTRAINTS PURGE;</pre>

```
DELETE FROM nome_tabela  
[WHERE condição];
```

- Cláusula **WHERE**
 - **Opcional**
 - Especifica as linhas da tabela que o comando deve eliminar.
- Exemplos

```
DELETE FROM pessoas  
WHERE idade < 18;
```

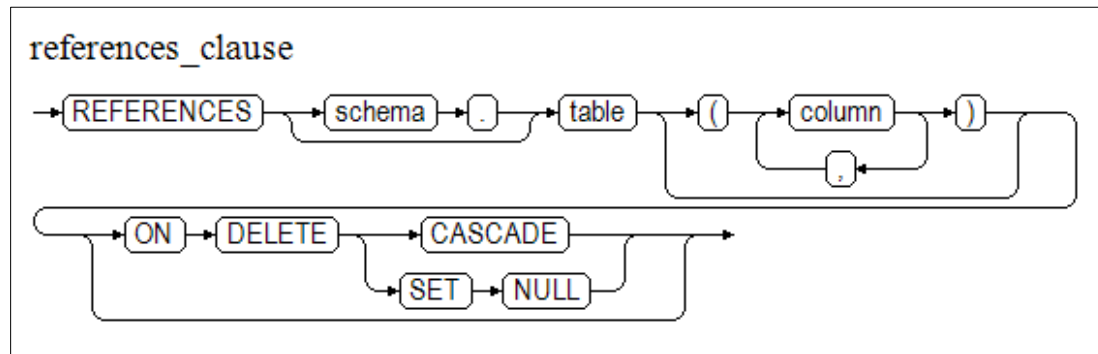
Para **tentar** eliminar o **conjunto de linhas** da tabela **pessoas** cuja coluna **idade** é inferior a 18 anos.

```
DELETE FROM pessoas;
```

Para **tentar** eliminar **todas** as linhas da tabela **pessoas**.

- **Depende**

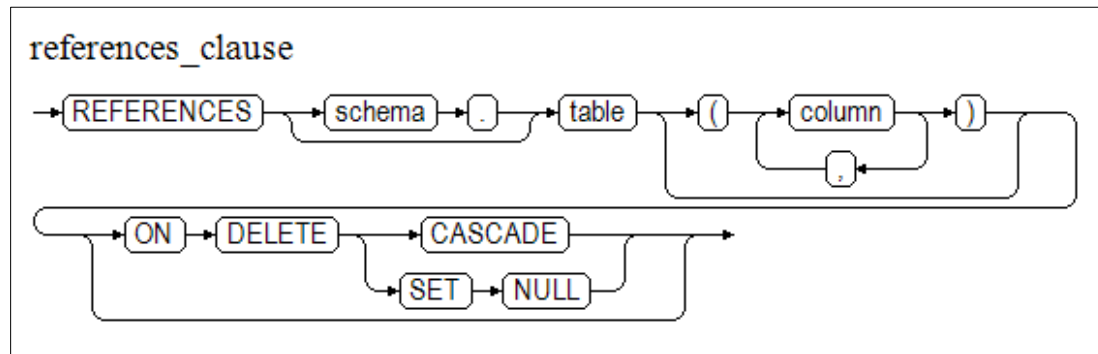
- Da **definição** da **restrição FOREIGN KEY** da tabela:



- **Sem** Cláusula **ON DELETE**:
 - Linhas **eliminadas apenas** se não forem referenciadas noutras tabelas.
 - **Com** Cláusula **ON DELETE**:
 - Linhas **eliminadas sempre**.
 - Linhas de **outras tabelas** dependentes (com referências) das linhas eliminadas:
 - Com Cláusula **CASCADE**: também são eliminadas.
 - Com Cláusula **SET NULL**: somente as referências passam a NULL.

- **Depende**

- Da **definição** da **restrição FOREIGN KEY** da tabela:



- **Sem** Cláusula **ON DELETE**:
 - Linhas **eliminadas apenas** se não forem referenciadas noutras tabelas.
- **Com** Cláusula **ON DELETE**:
 - Linhas **eliminadas sempre**.
 - Linhas de **outras tabelas** dependentes (com referências) das linhas eliminadas:
 - Com Cláusula **CASCADE**: também são eliminadas.
 - Com Cláusula **SET NULL**: somente as referências passam a NULL.