

11

## Dicionário de Dados

## Objetivos deste Capítulo

- Ao concluir este capítulo, você poderá:
  - Usar as visões do dicionário de dados para obter informações sobre todos os objetos do banco de dados
  - Consultar diversas visões de dicionário de dados

11-2

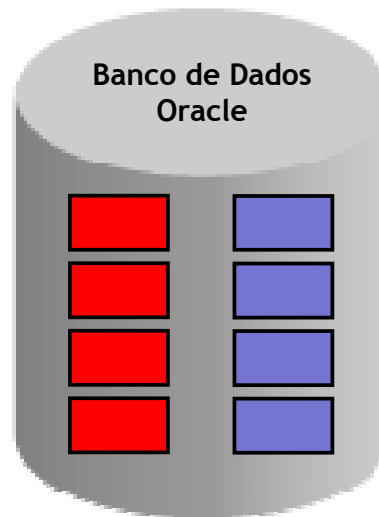
### Objetivos deste Capítulo

Este capítulo apresenta as visões do dicionário de dados. Você aprenderá que é possível usar as visões de dicionário para recuperar metadados e criar relatórios sobre objetos armazenados no banco de dados.

# O Dicionário de Dados

Tabelas com dados sobre a empresa:

**FUNCIONARIO**  
**DEPARTAMENTO**  
**LOCALIDADE**  
**HISTORICO\_CARGO**  
...



Visões de dicionário de dados:

**DICTIONARY**  
**USER\_OBJECTS**  
**USER\_TABLES**  
**USER\_TAB\_COLUMNS**  
...

11-3

## O Dicionário de Dados

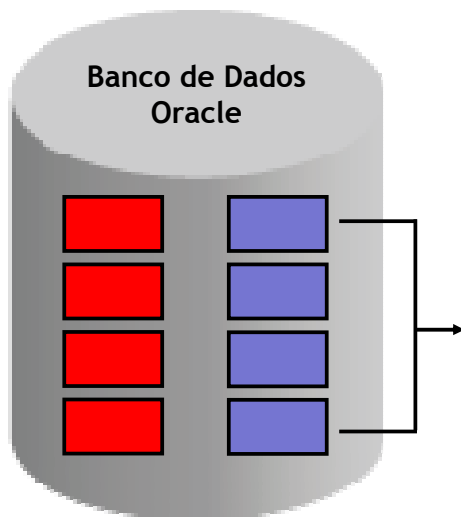
As tabelas de um sistema são criadas pelo usuário e contêm dados sobre a empresa, como `FUNCIONARIO`. Existe um outro conjunto de tabelas e visões no banco de dados Oracle conhecido como *dicionário de dados*. Esse conjunto é criado e mantido pelo Oracle e contém informações sobre o banco de dados. O dicionário de dados é estruturado em tabelas e visões, da mesma forma que em outros dados do banco de dados. Além de ser essencial para todos os bancos de dados Oracle, o dicionário de dados é uma importante ferramenta tanto para usuários finais quanto desenvolvedores e administradores de bancos de dados.

Para acessar o dicionário de dados, use comandos SQL. Como o dicionário de dados é somente para leitura, é possível executar apenas consultas.

Você pode consultar as visões de dicionário baseadas nas tabelas de dicionário para localizar informações como:

- Definições de todos os objetos de esquema do banco de dados (tabelas, visões, índices, sinônimos, sequências, procedimentos, funções, pacotes, triggers e outros)
- Valores padrão de colunas
- Informações sobre constraints de integridade
- Nomes dos usuários do banco de dados e seus privilégios e atribuições
- Outras informações gerais sobre o banco de dados

# Estrutura do Dicionário de Dados



- Consiste em:
  - Tabelas-base
  - Visões acessíveis aos usuários

11-4

## Estrutura do Dicionário de Dados

As tabelas-base armazenam informações sobre o banco de dados. Apenas o Oracle pode executar operações de gravação e leitura nessas tabelas. Você raramente terá acesso direto a elas.

Várias visões resumem e exibem as informações armazenadas nas tabelas-base do dicionário de dados. Essas visões decodificam os dados das tabelas-base e os apresentam como informações úteis (como nomes de tabelas ou usuários) usando joins e cláusulas `WHERE` para simplificar as informações. A maioria dos usuários obtém acesso às visões, e não às tabelas-base.

O usuário Oracle `SYS` é proprietário de todas as tabelas-base e visões acessíveis do dicionário de dados. Nenhum usuário Oracle poderá alterar (`UPDATE`, `DELETE` ou `INSERT`) linhas ou objetos contidos no esquema `SYS`, pois essa atividade poderá comprometer a integridade do banco de dados.

# Estrutura do Dicionário de Dados

Prefixo da Visão	Objetivo
<b>USER</b>	Exibe informações sobre objetos pertencentes ao próprio usuário
<b>ALL</b>	Exibe informações sobre objetos aos quais o usuário tem permissão de acessar
<b>DBA</b>	Exibe informações sobre todos os objetos do banco de dados
<b>V\$</b>	Exibe informações de desempenho

11-5

## Estrutura do Dicionário de Dados (continuação)

O dicionário de dados consiste em conjuntos de visões. Em vários casos, um conjunto é formado por três visões com informações semelhantes e diferenciadas por prefixos. Por exemplo, há uma visão denominada `USER_OBJECTS`, outra denominada `ALL_OBJECTS` e uma terceira denominada `DBA_OBJECTS`.

Essas três visões contêm informações semelhantes sobre objetos do banco de dados, com exceção do escopo, que é diferente. `USER_OBJECTS` contém informações sobre objetos do próprio usuário. `ALL_OBJECTS` contém informações sobre todos os objetos aos quais o usuário tem acesso. `DBA_OBJECTS` contém informações sobre todos os objetos pertencentes a todos os usuários. Em geral, as visões com o prefixo `ALL` ou `DBA` contêm uma coluna adicional, denominada `OWNER`, que identifica o dono do objeto.

Há também um conjunto de visões com o prefixo `V$`. Essas visões são dinâmicas por natureza e contêm informações sobre desempenho. Essas visões são usadas pelos administradores de banco de dados e normalmente não são liberadas para acesso aos outros usuários.

# Como Usar as Visões de Dicionário

- A visão `DICTIONARY` contém os nomes e as descrições das tabelas e visões do dicionário de dados.

```
DESCRIBE DICTIONARY
```

Table	Column	Tipo De Dados	Tamanho	Precisão	Escala	Chave Primária	Anulável
DICTIONARY	TABLE_NAME	Varchar2	30	-	-	-	✓
	COMMENTS	Varchar2	4000	-	-	-	✓

```
SELECT *  
FROM dictionary  
WHERE table_name = 'USER_OBJECTS';
```

TABLE_NAME	COMMENTS
USER_OBJECTS	Objects owned by the user

11-6

## Como Usar as Visões de Dicionário

Para se familiarizar com as visões de dicionário, você pode usar a visão de dicionário `DICTIONARY`. Ela contém o nome e a descrição resumida de cada visão de dicionário à qual você tem acesso.

Você pode criar consultas para procurar informações sobre um nome de visão específico ou pesquisar uma palavra ou expressão na coluna `COMMENTS`. No exemplo mostrado, a visão `DICTIONARY` é descrita. Ela tem duas colunas. O comando `SELECT` recupera informações sobre a visão de dicionário `USER_OBJECTS`. Essa visão contém informações sobre todos os objetos pertencentes a você.

É possível criar consultas para pesquisar uma palavra ou expressão na coluna de comentários. Por exemplo, a consulta a seguir retorna os nomes de todas as visões às quais você tem acesso que contêm a palavra *columns* na coluna `COMMENT`:

```
SELECT table_name  
FROM dictionary  
WHERE LOWER(comments) LIKE '%columns';
```

Os nomes no dicionário de dados são exibidos em letras maiúsculas.

## Visão USER\_OBJECTS

- Descreve todos os objetos pertencentes a você
- Usada para obter uma listagem de todos os nomes e tipos de objetos do seu esquema, além das seguintes informações:
  - Data de criação
  - Data da última modificação
  - Status (válido ou inválido)
- Consultar `ALL_OBJECTS` para exibir todos os objetos aos quais você tem acesso

11-7

### Visão USER\_OBJECTS

Você pode consultar a visão `USER_OBJECTS` para ver os nomes e os tipos de todos os objetos do seu esquema. Essa visão contém várias colunas:

- **OBJECT\_NAME**: Nome do objeto
- **OBJECT\_ID**: Número do objeto de dicionário de dados
- **OBJECT\_TYPE**: Tipo de objeto (como `TABLE`, `VIEW`, `INDEX`, `SEQUENCE`)
- **CREATED**: Data/hora (timestamp) da criação do objeto
- **LAST\_DDL\_TIME**: Data/hora (timestamp) da última modificação do objeto resultante de um comando DDL
- **STATUS**: Status do objeto (`VALID`, `INVALID` ou `N/A`)
- **GENERATED**: Se o nome deste objeto foi gerado pelo sistema? (Y | N)

Nem todas as colunas foram relacionadas nessa listagem.

Você também pode consultar a visão `ALL_OBJECTS` para obter uma listagem de todos os objetos aos quais tem acesso.

# Visão USER\_OBJECTS

```
SELECT object_name, object_type, created, status
FROM   user_objects
ORDER BY object_type;
```

OBJECT_NAME	OBJECT_TYPE	CREATED	STATUS
DEPARTAMENTO_PK	INDEX	12/05/09	VALID
REGIAO_PK	INDEX	12/05/09	VALID
LOCALIDADE_PAIS_IX	INDEX	12/05/09	VALID
LOCALIDADE_ESTADO_IX	INDEX	12/05/09	VALID
PAIS	TABLE	12/05/09	VALID
REGIAO	TABLE	12/05/09	VALID
CALCULO	TABLE	10/05/09	VALID
CARGO	TABLE	12/05/09	VALID
FUNCIONARIO	TABLE	12/05/09	VALID
HISTORICO_CARGO	TABLE	12/05/09	VALID
FAIXA_SALARIO	TABLE	12/05/09	VALID
DETALHE_FUNCIONARIO	VIEW	12/05/09	VALID

11-8

## Visão USER\_OBJECTS (continuação)

O exemplo mostra os nomes, os tipos, as datas de criação e o status de todos os objetos pertencentes ao usuário.

A coluna **OBJECT\_TYPE** contém o valor **TABLE**, **VIEW**, **SEQUENCE**, **INDEX**, **PROCEDURE**, **FUNCTION**, **PACKAGE** ou **TRIGGER**.

A coluna **STATUS** contém o valor **VALID**, **INVALID** ou **N/A**. Embora as tabelas sejam sempre válidas, as visões, os procedimentos, as funções, os pacotes e as triggers podem ser inválidos.

### A Visão CAT

Para consultas e informações com resultado simples, é possível consultar a visão **CAT**. Essa visão contém apenas duas colunas: **TABLE\_NAME** e **TABLE\_TYPE**. Ela fornece os nomes de todos os objetos **INDEX**, **TABLE**, **CLUSTER**, **VIEW**, **SYNONYM**, **SEQUENCE** ou **UNDEFINED**.



# Visão USER\_TABLES

```
DESCRIBE user_tables
```

Table	Column	Tipo De Dados	Tamanho	Precisão	Escala	Chave Primária	Anulável
USER TABLES	TABLE_NAME	Varchar2	30	-	-	-	-
	TABLESPACE_NAME	Varchar2	30	-	-	-	✓
	CLUSTER_NAME	Varchar2	30	-	-	-	✓
	IOT_NAME	Varchar2	30	-	-	-	✓
	STATUS	Varchar2	9	-	-	-	✓

```
SELECT table_name  
FROM user_tables;
```

TABLE_NAME
CALCULO
REGIAO
LOCALIDADE
DEPARTAMENTO
CARGO
FUNCONARIO

11-9

## Visão USER\_TABLES

Você pode usar a visão USER\_TABLES para obter os nomes e várias informações sobre todas as suas tabelas. Além do nome da tabela, ela contém informações detalhadas sobre o local de armazenamento.

Você também pode consultar as visões ALL\_TABLES e TABS para exibir uma listagem de todas as tabelas às quais tem acesso:

```
SELECT table_name  
FROM tabs;
```

# Informações sobre Colunas

- Visão `USER_TAB_COLUMNS`:

```
DESCRIBE user_tab_columns
```

Table	Column	Tipo De Dados	Tamanho	Precisão	Escala	Chave Primária	Anulável
<code>USER_TAB_COLUMNS</code>	<code>TABLE_NAME</code>	Varchar2	30	-	-	-	-
	<code>COLUMN_NAME</code>	Varchar2	30	-	-	-	-
	<code>DATA_TYPE</code>	Varchar2	106	-	-	-	✓
	<code>DATA_TYPE_MOD</code>	Varchar2	3	-	-	-	✓
	<code>DATA_TYPE_OWNER</code>	Varchar2	30	-	-	-	✓
	<code>DATA_LENGTH</code>	Number	-	-	-	-	-
	<code>DATA_PRECISION</code>	Number	-	-	-	-	✓
	<code>DATA_SCALE</code>	Number	-	-	-	-	✓
	<code>NULLABLE</code>	Varchar2	1	-	-	-	✓
	<code>COLUMN_ID</code>	Number	-	-	-	-	✓

11-10

## Informações sobre Colunas

Você pode consultar a visão `USER_TAB_COLUMNS` para obter informações detalhadas sobre as colunas das suas tabelas. Enquanto a visão `USER_TABLES` fornece informações sobre o nome e o local de armazenamento das tabelas, a visão `USER_TAB_COLUMNS` contém informações detalhadas sobre as colunas das tabelas.

A visão contém as seguintes informações:

- Nomes de colunas
- Tipos de dados da coluna
- Tamanho dos tipos de dados
- Precisão e escala das colunas `NUMBER`
- Se são permitidos valores nulos (Há uma constraint `NOT NULL` na coluna?)
- Valor padrão

# Informações sobre Colunas

```
SELECT column_name, data_type, data_length,  
       data_precision, data_scale, nullable  
FROM   user_tab_columns  
WHERE  table_name = 'FUNCIONARIO';
```

COLUMN_NAME	DATA_TYPE	DATA_LENGTH	DATA_PRECISION	DATA_SCALE	NULLABLE
COD_FUNCIONARIO	NUMBER	22	6	0	N
NOME	VARCHAR2	20	-	-	Y
SOBRENOME	VARCHAR2	25	-	-	N
EMAIL	VARCHAR2	25	-	-	N
TELEFONE	VARCHAR2	20	-	-	Y
DATA_ADMISSAO	DATE	7	-	-	N
COD_CARGO	VARCHAR2	10	-	-	N
SALARIO	NUMBER	22	8	2	Y
COMISSAO	NUMBER	22	2	2	Y
COD_GERENTE	NUMBER	22	6	0	Y
COD_DEPARTAMENTO	NUMBER	22	4	0	Y

11-11

## Informações sobre Colunas (continuação)

A consulta à tabela `USER_TAB_COLUMNS` permite localizar detalhes sobre colunas, como nomes, tipos de dados, tamanhos dos tipos de dados, constraints nulas e valores padrão.

O exemplo mostra as colunas, os tipos de dados, os tamanhos dos dados e as constraints NOT NULL da tabela `FUNCIONARIO`. Observe que essas informações são semelhantes à saída do comando `SQL*Plus DESCRIBE`.

# Informações sobre Constraints

- `USER_CONSTRAINTS` descreve as definições de constraints nas tabelas.
- `USER_CONS_COLUMNS` descreve as colunas pertencentes a você e especificadas nas constraints.

```
DESCRIBE user_constraints
```

Table	Column	Tipo De Dados	Tamanho	Precisão	Escala	Chave Primária	Anulável
USER_CONSTRAINTS	OWNER	Varchar2	30	-	-	-	-
	CONSTRAINT_NAME	Varchar2	30	-	-	-	-
	CONSTRAINT_TYPE	Varchar2	1	-	-	-	✓
	TABLE_NAME	Varchar2	30	-	-	-	-
	SEARCH_CONDITION	Long	0	-	-	-	✓
	R_OWNER	Varchar2	30	-	-	-	✓
	R_CONSTRAINT_NAME	Varchar2	30	-	-	-	✓
	DELETE_RULE	Varchar2	9	-	-	-	✓
	STATUS	Varchar2	8	-	-	-	✓

11-12

## Informações sobre Constraints

Você pode obter os nomes das constraints, o tipo de constraint, o nome da tabela à qual se aplica a constraint, a condição das constraints de `CHECK`, informações sobre constraints de chave estrangeira, a regra de deleção das constraints de chave estrangeira, o status e vários outros tipos de informações sobre constraints.

# Informações sobre Constraints

```
SELECT constraint_name, constraint_type,  
       search_condition, r_constraint_name,  
       delete_rule, status  
FROM   user_constraints  
WHERE  table_name = 'FUNCIONARIO';
```

CONSTRAINT_NAME	CONSTRAINT_TYPE	SEARCH_CONDITION	R_CONSTRAINT_NAME	DELETE_RULE	STATUS
FUNCIONARIO_SOBRENOME_NN	C	"SOBRENOME" IS NOT NULL	-	-	ENABLED
FUNCIONARIO_EMAIL_NN	C	"EMAIL" IS NOT NULL	-	-	ENABLED
FUNCIONARIO_DATA_CONTRAT_NN	C	"DATA_ADMISSAO" IS NOT NULL	-	-	ENABLED
FUNCIONARIO_COD_CARGO_NN	C	"COD_CARGO" IS NOT NULL	-	-	ENABLED
FUNCIONARIO_MIN_SALARIO	C	salario > 0	-	-	ENABLED
FUNCIONARIO_EMAIL_UK	U	-	-	-	ENABLED
FUNCIONARIO_PK	P	-	-	-	ENABLED
FUNCIONARIO_DEPARTAMENTO_FK	R	-	DEPARTAMENTO_PK	NO ACTION	ENABLED
FUNCIONARIO_CARGO_FK	R	-	CARGO_PK	NO ACTION	ENABLED
FUNCIONARIO_GERENTE_FK	R	-	FUNCIONARIO_PK	NO ACTION	ENABLED

11-13

## Informações sobre Constraints (continuação)

No exemplo mostrado, a visão `USER_CONSTRAINTS` é consultada para localizar os nomes, os tipos, as condições de verificação, o nome da constraint `UNIQUE` ou `PRIMARY KEY` à qual a chave estrangeira faz referência, a regra de deleção de uma chave estrangeira e o status das constraints na tabela `FUNCIONARIO`.

`CONSTRAINT_TYPE` pode ser:

- C (constraint de `CHECK`)
- P (chave primária)
- U (chave exclusiva - `UNIQUE`)
- R (chave estrangeira)
- V (com opção de verificação, em uma visão)
- O (somente para leitura, em uma visão)

`DELETE_RULE` pode ser:

- **CASCADE:** Se o registro pai for deletado, os registros filhos também serão.
- **NO ACTION:** Um registro pai só é deletado se não houver registros filhos.

`STATUS` pode ser:

- **ENABLED:** A constraint está ativa.
- **DISABLED:** A constraint é desativada.

# Informações sobre Constraints

```
DESCRIBE user_cons_columns
```

Table	Column	Tipo De Dados	Tamanho	Precisão	Escala	Chave Primária	Anulável
USER_CONS_COLUMNS	OWNER	Varchar2	30	-	-	-	-
	CONSTRAINT_NAME	Varchar2	30	-	-	-	-
	TABLE_NAME	Varchar2	30	-	-	-	-
	COLUMN_NAME	Varchar2	4000	-	-	-	✓
	POSITION	Number	-	-	-	-	✓

```
SELECT constraint_name, column_name  
FROM user_cons_columns  
WHERE table_name = 'FUNCIONARIO';
```

CONSTRAINT_NAME	COLUMN_NAME
FUNCIONARIO_EMAIL_UK	EMAIL
FUNCIONARIO_MN_SALARIO	SALARIO
FUNCIONARIO_COD_CARGO_NN	COD_CARGO
FUNCIONARIO_DATA_CONTRAT_NN	DATA_ADMISSAO
FUNCIONARIO_EMAIL_NN	EMAIL
FUNCIONARIO_SOBRENOME_NN	SOBRENOME
FUNCIONARIO_FK	COD_FUNCIONARIO
FUNCIONARIO_GERENTE_FK	COD_GERENTE
FUNCIONARIO_CARGO_FK	COD_CARGO
FUNCIONARIO_DEPARTAMENTO_FK	COD_DEPARTAMENTO

11-14

## Informações sobre Constraints (continuação)

Consulte a visão de dicionário `USER_CONS_COLUMNS` para obter os nomes das colunas às quais se aplica uma constraint. Essa visão informa o nome do dono de uma constraint, o nome da constraint, a tabela que contém a constraint, o nome da coluna da constraint e a posição original da coluna ou do atributo na definição do objeto.

Você também pode criar um join entre `USER_CONSTRAINTS` e `USER_CONS_COLUMNS` para gerar um relatório personalizado com informações das duas visões.

# Informações sobre Visões

1 **DESCRIBE user\_views**

Table	Column	Tipo De Dados	Tamanho	Precisão	Escala	Chave Primária	Anulável
USER_VIEWS	VIEW_NAME	Varchar2	30	-	-	-	-
	TEXT_LENGTH	Number	-	-	-	-	✓
	TEXT	Long	0	-	-	-	✓

2 **SELECT DISTINCT view\_name FROM user\_views;**

VIEW_NAME
DETALHE_FUNCIONARIO

3 **SELECT text FROM user\_views  
WHERE view\_name = 'DETALHE\_FUNCIONARIO';**

**TEXT**

```
SELECT e.cod_funcionario, e.cod_cargo, e.cod_gerente, e.cod_departamento, d.cod_localidade, l.cod_pais, e.nome, e.sobrenome, e.salario, e.comissao, d.nome_departamento, j.nome_cargo, l.cidade, l.estado, c.nome_pais, r.nome_regiao FROM funcionario e, departamento d, cargo j, localidade l, pais c, regiao r WHERE e.cod_departamento = d.cod_departamento AND d.cod_localidade = l.cod_localidade AND l.cod_pais = c.cod_pais AND c.cod_regiao = r.cod_regiao AND j.cod_cargo = e.cod_cargo WITH READ ONLY
```

11-15

## Informações sobre Visões

Depois que uma visão for criada por um usuário, você poderá consultar `USER_VIEWS` para verificar o nome e a definição dessa visão. O texto do comando `SELECT` que constitui a visão é armazenado em uma coluna de tipo de dados `LONG`. A coluna `LENGTH` representa o número de caracteres no comando `SELECT`. Por padrão, quando você seleciona um valor em uma coluna `LONG`, são exibidos apenas os 80 primeiros caracteres do valor da coluna. Para exibir mais do que 80 caracteres, use o comando *SQL\*Plus* `SET LONG:`

```
SET LONG 1000
```

Nos exemplos do slide:

1. Uma listagem das colunas da visão `USER_VIEWS` são exibidas. Observe que, no slide, aparece apenas uma parte da listagem.
2. Os nomes das visões são recuperados.
3. O comando `SELECT` referente a `DETALHE_FUNCIONARIO` é exibida a partir do dicionário.

# Informações sobre Sequências

- `USER_SEQUENCES` descreve as sequências do usuário

```
DESCRIBE user_sequences
```

Table	Column	Tipo De Dados	Tamanho	Precisão	Escala	Chave Primária	Anulável
<u>USER_SEQUENCES</u>	<u>SEQUENCE_NAME</u>	Varchar2	30	-	-	-	-
	<u>MIN_VALUE</u>	Number	-	-	-	-	✓
	<u>MAX_VALUE</u>	Number	-	-	-	-	✓
	<u>INCREMENT_BY</u>	Number	-	-	-	-	-
	<u>CYCLE_FLAG</u>	Varchar2	1	-	-	-	✓
	<u>ORDER_FLAG</u>	Varchar2	1	-	-	-	✓
	<u>CACHE_SIZE</u>	Number	-	-	-	-	-
	<u>LAST_NUMBER</u>	Number	-	-	-	-	-

11-16

## Informações sobre Sequências

A visão `USER_SEQUENCES` descreve todas as sequências que pertencem a você. Quando você cria uma sequência, você especifica critérios que são armazenados na visão `USER_SEQUENCES`. As colunas dessa visão são:

- `SEQUENCE_NAME`: Nome da sequência
- `MIN_VALUE`: Valor mínimo da sequência
- `MAX_VALUE`: Valor máximo da sequência
- `INCREMENT_BY`: Valor de incremento da sequência
- `CYCLE_FLAG`: A sequência é reiniciada quando o limite é atingido?
- `ORDER_FLAG`: Os números da sequência são gerados em ordem?
- `CACHE_SIZE`: Quantidade de números de sequência armazenados no cache de memória
- `LAST_NUMBER`: Último número de sequência gravado no disco. Se uma sequência usar armazenamento em cache, o número gravado no disco será o último número incluído no cache de memória. Esse número provavelmente será maior que o último número de sequência usado.



# Informações sobre Sequências

- Verifique os valores de uma sequência na visão `USER_SEQUENCES` do dicionário de dados.

SEQUENCE	INCREMENT	START	MAX
...	...	...	...

- ```
SELECT  sequence_name, min_value, max_value,
        increment_by, last_number
FROM    user_sequences;
```

[illegible]

- A coluna `LAST_NUMBER` exibirá o próximo número disponível da sequência se `NOCACHE` for especificado.

|       |
|-------|
| 11-17 |
|-------|

## Informações sobre Sequências (continuação)

Após a criação da sequência, ela fica documentada no dicionário de dados. Como uma sequência é um objeto de banco de dados, você pode identificá-la na visão `USER OBJECTS` do dicionário de dados.

Para confirmar as definições da sequência, selecione-a na visão `USER SEQUENCES` do dicionário de dados.

Se a sequência tiver sido criada com `NOCACHE`, você poderá exibir o próximo valor disponível dessa sequência sem incrementá-lo consultando a coluna `LAST NUMBER` da `visão USER SEQUENCES`.

# Informações sobre Sinônimos

- `USER_SYNONYMS` descreve os sinônimos do usuário

```
DESCRIBE user_synonyms
```

| Table         | Column       | Tipo De Dados | Tamanho | Precisão | Escala | Chave Primária | Anulável |
|---------------|--------------|---------------|---------|----------|--------|----------------|----------|
| USER_SYNONYMS | SYNONYM_NAME | Varchar2      | 30      | -        | -      | -              | -        |
|               | TABLE_OWNER  | Varchar2      | 30      | -        | -      | -              | ✓        |
|               | TABLE_NAME   | Varchar2      | 30      | -        | -      | -              | -        |
|               | DB_LINK      | Varchar2      | 128     | -        | -      | -              | ✓        |

```
SELECT *  
FROM user_synonyms ;
```

| SYNONYM_NAME | TABLE_OWNER | TABLE_NAME  | DB_LINK |
|--------------|-------------|-------------|---------|
| FUNC         | USUARIOS2   | FUNCIONARIO | -       |

11-18

## Informações sobre Sinônimos

A visão de dicionário `USER_SYNONYMS` descreve sinônimos privados (sinônimos que pertencem a você).

Você pode consultar essa visão para localizar seus sinônimos. Também pode consultar `ALL_SYNONYMS` para obter o nome de todos os sinônimos disponíveis para você e os objetos aos quais esses sinônimos se aplicam.

As colunas dessa visão são:

- `SYNONYM_NAME`: Nome do sinônimo
- `TABLE_OWNER`: Proprietário do objeto referenciado pelo sinônimo
- `TABLE_NAME`: Nome da tabela ou visão referenciada pelo sinônimo
- `DB_LINK`: Nome da referência do vínculo do banco de dados (se houver)

# Comentários em uma Tabela

- Você pode adicionar comentários a uma tabela ou coluna usando o comando `COMMENT`:

```
COMMENT ON TABLE funcionario
IS 'Informações sobre os funcionários';
Comentário criado.
```

- É possível exibir os comentários do banco de dados usando as seguintes visões de dicionário de dados:
  - `ALL_COL_COMMENTS`
  - `USER_COL_COMMENTS`
  - `ALL_TAB_COMMENTS`
  - `USER_TAB_COMMENTS`

11-19

## Comentários em uma Tabela

Você pode adicionar um comentário de até 2.000 bytes sobre uma coluna, uma tabela, uma visão ou uma visão materializada usando o comando `COMMENT`. O comentário é armazenado no dicionário de dados e é possível exibi-lo em uma das seguintes visões do dicionário de dados na coluna `COMMENTS`:

- `ALL_COL_COMMENTS`
- `USER_COL_COMMENTS`
- `ALL_TAB_COMMENTS`
- `USER_TAB_COMMENTS`

### Sintaxe

```
COMMENT ON TABLE tabela | COLUMN tabela.coluna
IS 'comentário';
```

### Na sintaxe:

|                   |                              |
|-------------------|------------------------------|
| <i>tabela</i>     | é o nome da tabela           |
| <i>coluna</i>     | é o nome da coluna da tabela |
| <i>comentário</i> | é o texto do comentário      |

Para eliminar um comentário do banco de dados, defina-o como uma string vazia (' '):

```
COMMENT ON TABLE funcionario IS ' ';
```

# Exercício 11

11-20

## Exercício 11

1. Para uma tabela especificada, crie um script que informe os nomes de colunas, os tipos de dados, os tamanhos dos tipos de dados e a possibilidade ou não de valores nulos. Solicite o nome da tabela ao usuário. Forneça apelidos apropriados às colunas `DATA_PRECISION` e `DATA_SCALE`. Salve esse script no arquivo `cap_11_01.sql`.

Por exemplo, se o usuário informar `DEPARTAMENTO`, este será o resultado:

| COLUMN_NAME       | DATA_TYPE | DATA_LENGTH | PRECISAO | ESCALA | NULLABLE |
|-------------------|-----------|-------------|----------|--------|----------|
| COD_DEPARTAMENTO  | NUMBER    | 22          | 4        | 0      | N        |
| NOME_DEPARTAMENTO | VARCHAR2  | 30          | -        | -      | N        |
| COD_GERENTE       | NUMBER    | 22          | 6        | 0      | Y        |
| COD_LOCALIDADE    | NUMBER    | 22          | 4        | 0      | Y        |

2. Crie um script que informe o nome da coluna, o nome da constraint, o tipo de constraint, a condição de pesquisa e o status para uma tabela específica. Una as visões `USER_CONSTRAINTS` e `USER_CONS_COLUMNS` para obter todas essas informações. Solicite o nome da tabela ao usuário. Salve o script no arquivo `cap_11_02.sql`.

Por exemplo, se o usuário informar `DEPARTAMENTO`, este será o resultado:

| COLUMN_NAME       | CONSTRAINT_NAME            | CONSTRAINT_TYPE | SEARCH_CONDITION                | STATUS  |
|-------------------|----------------------------|-----------------|---------------------------------|---------|
| NOME_DEPARTAMENTO | NOME_DAPARTAMENTO_NN       | C               | "NOME_DEPARTAMENTO" IS NOT NULL | ENABLED |
| COD_DEPARTAMENTO  | DEPARTAMENTO_PK            | P               | -                               | ENABLED |
| COD_LOCALIDADE    | DEPARTAMENTO_LOCALIDADE_FK | R               | -                               | ENABLED |
| COD_GERENTE       | DEPARTAMENTO_GERENTE_FK    | R               | -                               | ENABLED |

3. Adicione um comentário à tabela `DEPARTAMENTO`. Em seguida, consulte a visão `USER_TAB_COMMENTS` para verificar se o comentário está presente.

| COMMENTS                                                                            |
|-------------------------------------------------------------------------------------|
| Informações sobre os departamentos da empresa incluindo, nome, código e localidade. |

4. Localize os nomes de todos os sinônimos existentes no esquema.

| SYNONYM_NAME | TABLE_OWNER | TABLE_NAME  | DB_LINK |
|--------------|-------------|-------------|---------|
| FUNC         | USUARIOS2   | FUNCIONARIO | -       |

## Exercício 11

5. Você precisa determinar os nomes e as definições de todas as visões do seu esquema. Crie um relatório que recupere informações sobre visões: o nome e o texto das visões em `USER VIEWS`.

**Observação:** Para exibir mais conteúdo de uma coluna `LONG`, use o comando `SET LONG n` do *SQL\*Plus*, em que *n* é o valor do número de caracteres da coluna `LONG` que você deseja exibir.

| VIEW_NAME           | TEXT                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|---------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| DETALHE_FUNCIONARIO | SELECT e.cod_funcionario, e.cod_cargo, e.cod_gerente, e.cod_departamento, d.cod_localidade, l.cod_pais, e.nome, e.sobrenome, e.salario, e.comissao, d.nome_departamento, j.nome_cargo, l.cidade, l.estado, c.nome_pais, r.nome_regiao FROM funcionario e, departamento d, cargo j, localidade l, pais c, regiao r WHERE e.cod_departamento = d.cod_departamento AND d.cod_localidade = l.cod_localidade AND l.cod_pais = c.cod_pais AND c.cod_regiao = r.cod_regiao AND j.cod_cargo = e.cod_cargo WITH READ ONLY |

6. Obtenha os nomes das suas sequências. Crie uma consulta em um script para exibir as seguintes informações sobre suas sequências: nome, valor máximo, tamanho do incremento e último número. Salve o script como `cap_11_06.sql`. Execute o comando no script.

[illegible]