

Criando Outros Objetos

Sequences

- Sintaxe

- Opções

- Exemplos

- Verificando sua Existência

- Comando INSERT Utilizando “nextval”

- Obtendo o Valor Corrente da Sequence com “currval”

- Visualizando a Descrição da Sequence

Sinônimos

- Sinônimo Privado

- Sinônimo Público

- Exemplo

- Uso de Sinônimos com Database Links

1 Sequences

Quando uma aplicação necessita utilizar valores numéricos seqüenciais em uma tabela, ela faz uso de um objeto chamado **sequence**. Com ele os valores serão gerados automaticamente pelo Oracle.

Ela é um objeto que os usuários podem utilizar para geração de números sem haver repetição de valores. Geralmente seu uso está associado a geração de valores para chaves primárias.

Quando um número é gerado, a sequence é incrementada, independente da transação realizar commit ou rollback.

Se dois usuários concorrentes solicitam valores a mesma sequence, os valores gerados serão obrigatoriamente diferentes. A princípio um usuário nunca conseguirá acessar o mesmo número que foi gerado para outro usuário.

1 Sequences

Os números de uma sequence são gerados independentemente das tabelas, portanto uma mesma sequence pode ser utilizada para múltiplas tabelas.

É possível ter a impressão de que os números gerados possam estar sofrendo “pulos”. Isto ocorre por causa do acesso concorrente dos usuários.

Após uma sequence ser criada, seus valores podem ser solicitados em comandos SQL, utilizando-se de recursos especiais chamados “pseudo-colunas”:

- CURRVAL: retorna o valor corrente;
- NEXTVAL: incrementa/decrementa a sequence e retorna um novo valor.

2 Sintaxe

Create sequence <nome_da_sequence>
[increment by n]
[start with n]
[maxvalue n | nomaxvalue] or [minvalue n | nominvalue]
[cycle | nocycle]
[cache n | nocache];

Pré requisitos:

Para criar uma sequence em seu “ schema”, deve-se ter o privilégio
CREATE SEQUENCE

Para criar uma sequence em qualquer “schema”, deve-se ter o privilégio
CREATE ANY SEQUENCE

3 Opções

Opção	Descrição
nome_da_sequence	Nome do objeto, não podendo ser o mesmo de uma tabela ou view.
Increment by n	Especifica de quanto será o incremento ou decremento. O padrão é 1.
Start with n	Especifica o primeiro número a ser gerado. O padrão é 1.
Maxvalue n	Especifica o valor máximo que a sequence pode gerar. O padrão é nomaxvalue, indo até 10^{27}
Minvalue n	Especifica o valor mínimo para a sequence. É mutuamente exclusiva ao maxvalue.
Cycle nocycle	Indica que, ao atingir o valor máximo, a numeração continuará a partir do valor inicial. O default é nocycle.
Cache n nocache	Especifica quantos valores o Oracle pré-aloca e mantém em memória para efeitos de performance. O padrão é 20.

4 Exemplos

Criando...

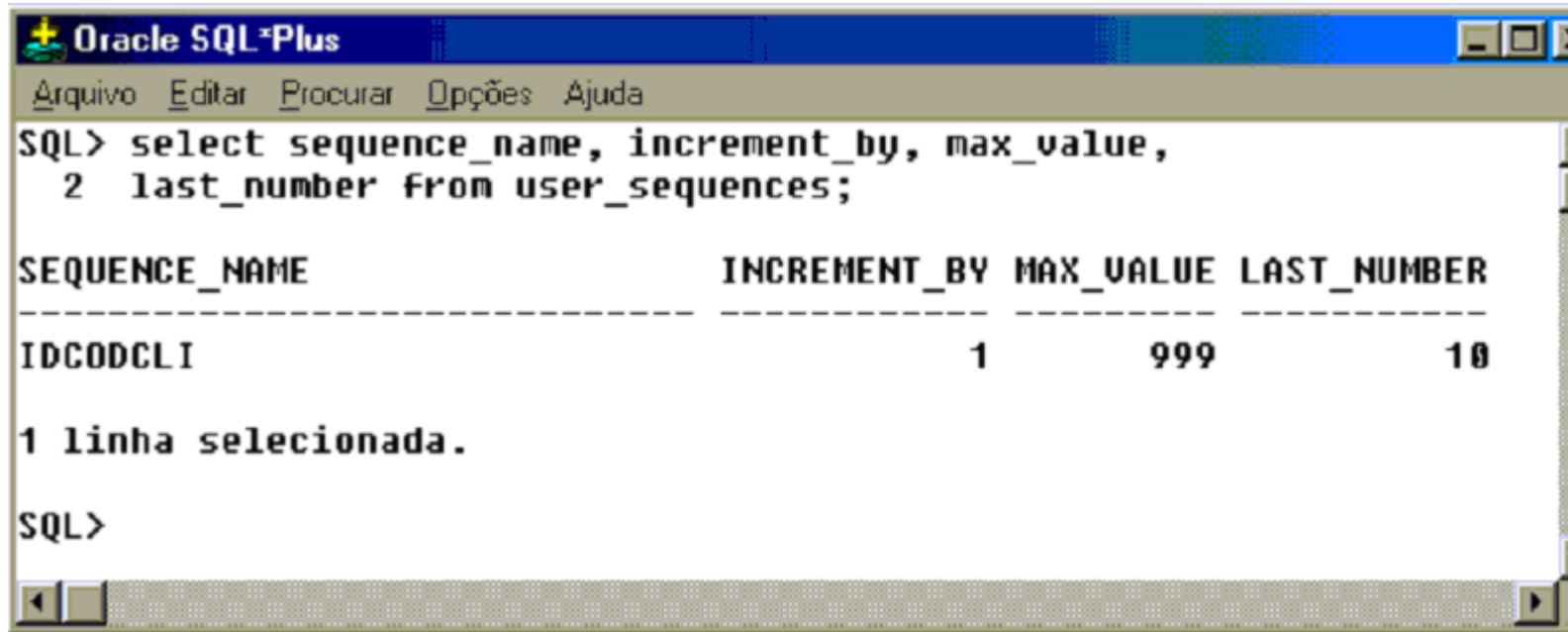
```
create sequence idcodcli  
increment by 1  
start with 10  
maxvalue 999  
nocycle;
```

```
create sequence idcodrnd  
minvalue 0  
maxvalue 100000  
cycle  
increment by -2;
```

Removendo...

```
drop sequence idcodcli;  
drop sequence idcodrnd;
```

5 Verificando sua Existência



The screenshot shows the Oracle SQL*Plus application window. The title bar reads "Oracle SQL*Plus". The menu bar includes "Arquivo", "Editar", "Procurar", "Opções", and "Ajuda". The command prompt shows the following SQL query and its result:

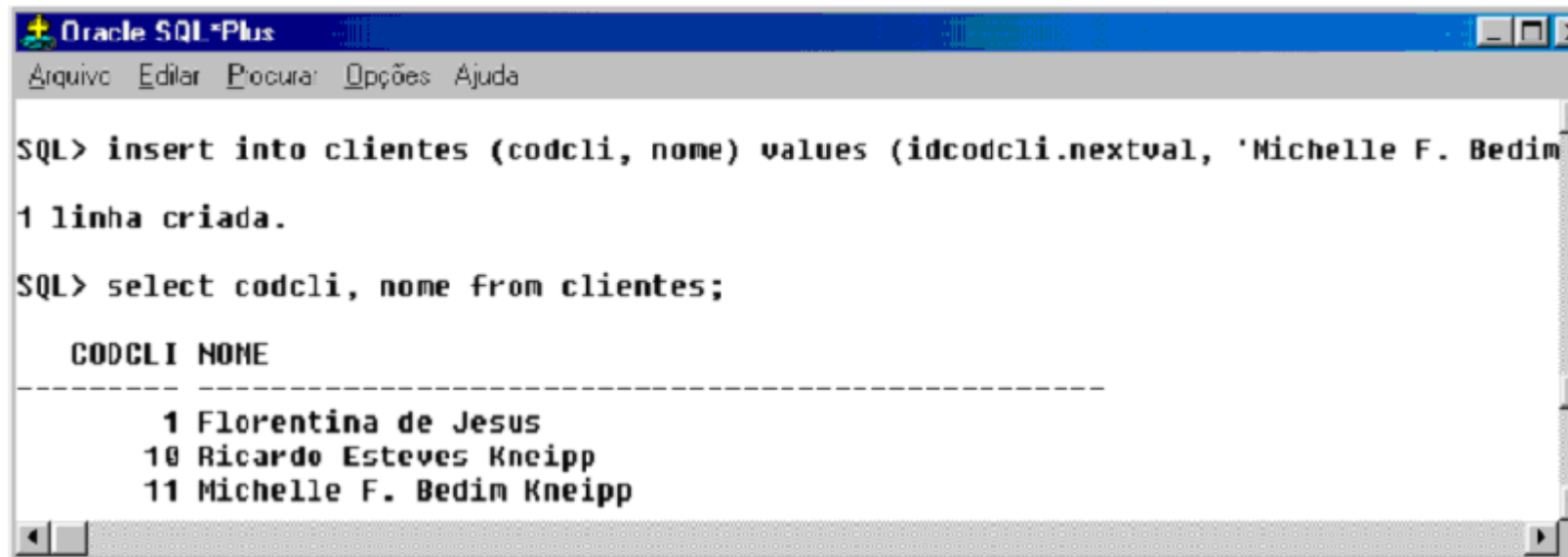
```
SQL> select sequence_name, increment_by, max_value,  
2 last_number from user_sequences;
```

SEQUENCE_NAME	INCREMENT_BY	MAX_VALUE	LAST_NUMBER
IDCODCLI	1	999	10

1 linha selecionada.

SQL>

6 Comando INSERT Utilizando “nextval”



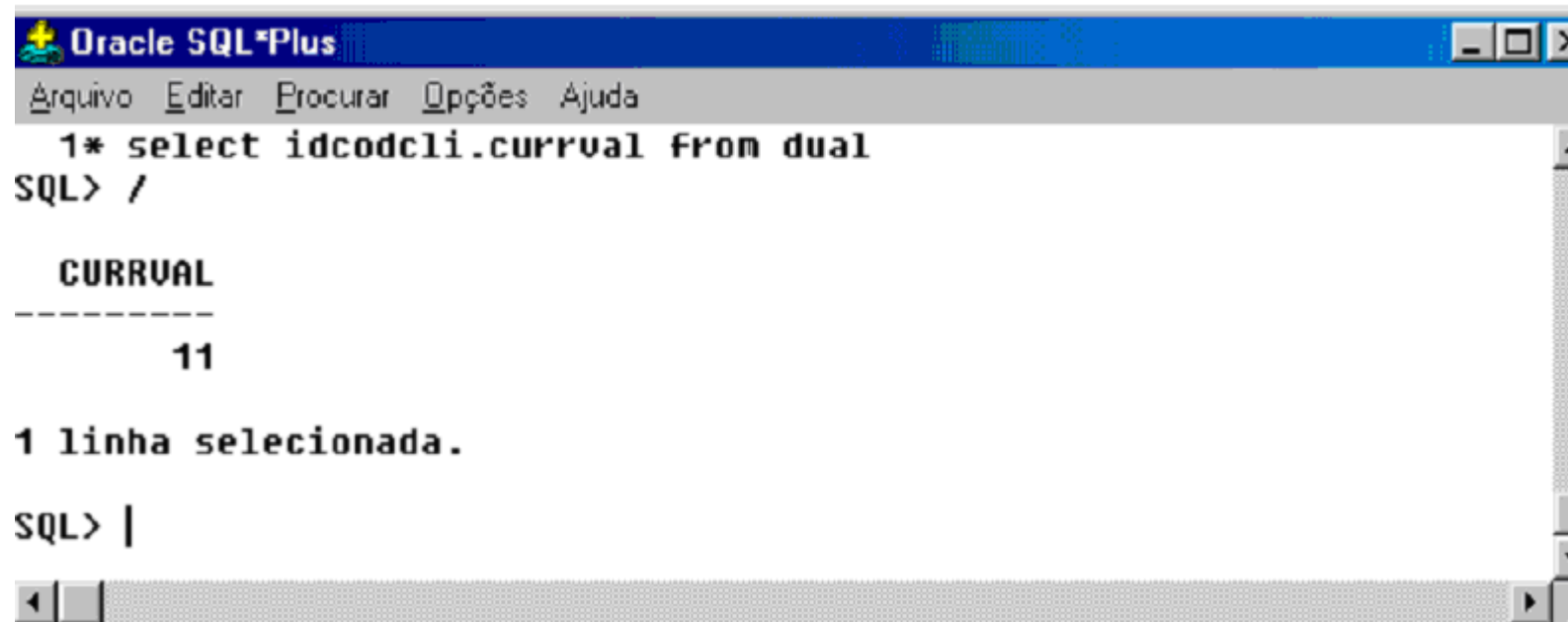
The screenshot shows a terminal window titled "Oracle SQL*Plus". The menu bar includes "Arquivo", "Editar", "Procurar", "Opções", and "Ajuda". The command prompt shows the following sequence of commands and output:

```
SQL> insert into clientes (codcli, nome) values (idcodcli.nextval, 'Michelle F. Bedim')  
1 linha criada.  
SQL> select codcli, nome from clientes;
```

The output of the SELECT query is displayed as follows:

CODCLI	NOME
1	Florentina de Jesus
10	Ricardo Esteves Kneipp
11	Michelle F. Bedim Kneipp

7 Obtendo o Valor Corrente da Sequence com “currval”



```
Oracle SQL*Plus
Arquivo Editar Procurar Opções Ajuda
1* select idcodcli.currval from dual
SQL> /

  CURRVAL
-----
        11

1 linha selecionada.
SQL> |
```


9 Sinônimos

São apelido (“alias”) que podem ser concedidos a um objeto.

Um sinônimo pode ser criado para os seguintes objetos:

- Tabela;
- View;
- Sequence;
- Procedure;
- Função;
- Package.

Sinônimos podem ser utilizados em conjunto com “database links”, permitindo referenciar objetos em outro banco de dados.

10 Sinônimo Privado

São aqueles que são criados e utilizados dentro do próprio SCHEMA do usuário.

Criando um Sinônimo Privado:

```
CREATE OR REPLACE SYNONYM <synonym_name> FOR <object_name>;
```

Dropando um Sinônimo Privado:

```
DROP SYNONYM <synonym_name>;
```

11 Sinônimo Público

- Usuários com privilégio de DBA podem criá-los automaticamente;
- Deve-se possuir o privilégio “CREATE PUBLIC SYNONYM”;
- Depois de criada, fica disponível para todos os usuários do banco de dados.

Criando um Sinônimo Público:

```
CREATE OR REPLACE PUBLIC SYNONYM <synonym_name> FOR <object_name>;
```

Dropando um Sinônimo Público:

```
DROP PUBLIC SYNONYM <synonym_name>;
```

12 Exemplo

Ambiente:

- Usuário (schema) ADMIN que contém a tabela Cliente;
- Usuário (schema) RICARDO quer realizar SELECT nesta tabela.

Execução:

- Ele deverá ter recebido previamente o privilégio de executar SELECT na tabela Cliente, que pertence ao usuário ADMIN;
- Deverá ser criado um sinônimo privado para o usuário RICARDO.

12 Exemplo

Passo 1:

Conectar com um usuário apropriado (SYS, SYSTEM, DBA, etc.) e conceder o privilégio de criação de sinônimos ao usuário RICARDO.

```
SQL> GRANT CREATE SYNONYM TO RICARDO;
```

Grant succeeded.

12 Exemplo

Passo 2:

Conectar com o usuário ADMIN e conceder o privilegio de SELECT na tabela Cliente ao usuário RICARDO.

```
SQL> GRANT SELECT ON CLIENTE TO RICARDO;
```

Grant succeeded.

12 Exemplo

Passo 3:

Verificar se o acesso a tabela Cliente já é possível, sem a utilização de sinônimos.

```
SQL> SELECT * FROM ADMIN.CLIENTE;
```

ID	NAME
1	JOSE
2	MARIA
3	JOANA
4	JOAO

É importante reparar que a finalidade do uso de sinônimos é o de evitar a referência ao objeto desejado tendo que indicar seu SCHEMA (Usuario.Tabela). O objetivo é esconder a complexidade.

12 Exemplo

Passo 4:

Conectar com usuario RICARDO e criar o sinônimo para a tabela Cliente.

```
SQL> CREATE OR REPLACE SYNONYM CLIENTE FOR ADMIN.CLIENTE;
```

Synonym created.

12 Exemplo

Passo 5:

Testar o uso do sinônimo.

```
SQL> SELECT * FROM CLIENTE;
```

ID	NAME
1	JOSE
2	MARIA
3	JOANA
4	JOAO

13 Uso de Sinônimos com Database Links

Database link (link de banco de dados) é uma referência para um banco de dados remoto, que pode estar localizado em um local completamente diferente do banco de dados original.

Sintaxe:

```
CREATE <PUBLIC> DATABASE LINK <nome_do_dblink>  
  CONNECT TO <nome_do_usuario>  
  IDENTIFIED BY <senha_do_usuario>  
  USING <string_do_sqlnet>;
```

Exemplo:

```
CREATE DATABASE LINK filial  
CONNECT TO scott  
IDENTIFIED BY tiger  
USING 'CNOVA02';
```

13 Uso de Sinônimos com Database Links

Uso de Database Links:

```
UPDATE cliente@filial SET tipocliente = 1 WHERE codcliente = 2;
```

Criação do Sinônimo:

```
CREATE SYNONYM <nome> FOR <referência>;
```

Exemplo: CREATE SYNONYM cliente_filial FOR cliente@filial;

Uso do Sinônimo:

```
SELECT * FROM cliente_filial;
```