



Outros Objetos do Banco de Dados



Curso de Introdução a Oracle 10g: SQL
Prof.: Marlon Mendes Minussi
marlonminussi@gmail.br

Objetos do Banco de Dados

Objeto

Table

View

Sequence

Index

Synonym

Descrição

Unidade básica de armazenamento; composta de linhas e colunas.

Representação lógica de um subconjunto dos dados de uma ou mais tabelas.

Gera valores para chaves primárias.

Aumenta a performance das consultas.

Fornece nomes alternativos para objetos.



O que é uma Sequence?

- Um gerador de seqüências pode ser utilizado para gerar números seqüenciais automaticamente para linhas em tabelas. Uma sequence é um objeto de banco de dados criado por um usuário podendo ser compartilhado por múltiplos usuários.
- Um uso típico para sequence é criar um valor para uma chave primária, que deve ser único para cada linha.
- A sequence é gerada e incrementada por uma rotina interna do Oracle10g. Este objeto pode reduzir o tempo de desenvolvimento, uma vez que reduz a quantidade de código de aplicação necessária para gerar uma rotina que implemente uma seqüência.
- Os números das sequences são armazenados e gerados independentemente das tabelas. Portanto, a mesma sequence pode ser utilizada em tabelas diferentes.



CREATE SEQUENCE

- Sintaxe:

CREATE SEQUENCE sequence

[INCREMENT BY n]

[START WITH n]

[{MAXVALUE n | NOMAXVALUE}]

[{MINVALUE n | NOMINVALUE}]

[{CYCLE | NOCYCLE}]

[{CACHE n | NOCACHE}];

- Automaticamente gera números seqüenciais utilizando o comando CREATE SEQUENCE.



CREATE SEQUENCE

- *Sequence*: é o nome do gerador da seqüência.
- INCREMENT BY n: especifica o intervalo entre os números da seqüência onde n é um inteiro. Se esta cláusula for omitida, a seqüência será incrementada por 1.
- START WITH n: especifica o primeiro numero da seqüência a ser gerado. Se esta clausula for omitida, a seqüência começara em 1.
- MAXVALUE n: especifica o valor de Maximo que a seqüência pode gerar.
- NOMAXVALUE: especifica um valor de Maximo de -10^{27} para uma sequence ascendente e -1 para uma sequence descendente. Esta é a opção default.
- MINVALUE n: especifica o valor mínimo da seqüência.
- NOMINVALUE: especifica um valor mínimo de 1 para uma sequence ascendente e $-(10^{26})$ para uma sequence descendente. Esta é a opção default.
- CYCLE\NOCYCLE: especifica que a sequence continua gerando valores depois de atingir seu valor máximo ou mínimo ou que ela não deve gerar valores adicionais. NOCYCLE é a opção default.
- CACHE n\NOCACHE: especificam quantos valores o Servidor Oracle deve alocar e manter em memória. Por default, o Servidor Oracle mantém 20 valores em memória (cache).

Criando uma Sequence

Ex.:

```
CREATE SEQUENCE seq_cod_pac  
INCREMENT BY 1  
START WITH 201  
MAXVALUE 300  
NOCACHE  
NOCYCLE;
```

Sequence created.

- O exemplo acima cria uma sequence chamada SEQ_COD_PAC para ser utilizada para a coluna ID da tabela TCLIENTES. A sequência começa em 201, não permite cachê e não permite que ela seja cíclica.
- Não utilize a opção CYCLE se a sequence for utilizada para gerar valores de chave primária a menos que você possua um mecanismo que remova linhas antigas mais rapidamente que os ciclos da sequence.



Confirmando Sequences

- Ex.:

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

- Uma vez criada, a sequence é documentada no dicionário de dados. Considerando-se que uma sequence é um objeto do banco de dados, você pode identificá-la na tabela do dicionário de dados USER_OBJECTS.



Confirmando Sequences

- Você pode também confirmar as configurações da sequence selecionado a partir da tabela do dicionário de dados USER_SEQUENCES.

SEQUENCE_NAME	MIN_VALUE	MAX_VALUE	INCREMENT_BY	LAST_NUMBER
-----	-----	-----	-----	-----
SEQ_COD_PAC	1	300	1	201
SEQ_COD_ATENDIMENTO	1	1.0000E+27	1	1021

Pseudocolunas

NEXTVAL e CURRVAL

- Uma vez criada sequence, você pode utilizá-la para gerar números sequenciais para suas tabelas. Referenceie os valores da sequence utilizando as pseudocolunas NEXTVAL e CURRVAL.
- A pseudocoluna NEXTVAL é utilizada para extrair sucessivos números da sequence especificada. Você deve qualificar NEXTVAL com o nome da sequence. Quando você referencia sequence.NEXTVAL, um novo número da sequence é gerado e colocado em CURRVAL.
- A pseudocoluna CURRVAL é utilizada para se referenciar a um número da sequence que o usuário atual gerou anteriormente.
- NEXTVAL deve ser utilizada para gerar um número da sequence na sessão do usuário atual antes de CURRVAL poder ser referenciada. Você deve qualificar CURRVAL com o nome da sequence. Quando sequence.CURRVAL é referenciada, o ultimo valor retornado ao processo daquele usuário é exibido.



Regras para Utilizar NEXTVAL e CURRVAL

- Você pode utilizar NEXTVAL e CURRVAL no seguinte:
- Na lista da cláusula SELECT de um comando SELECT que não seja parte de uma sub-consulta;
- Na lista da cláusula SELECT de uma sub-consulta em um comando INSERT;
- Na cláusula VALUES de um comando INSERT;
- Na cláusula SET de um comando UPDATE;

Regras para Utilizar NEXTVAL e CURRVAL

- Você não pode utilizar NEXTVAL e CURRVAL no seguinte:
- Na lista da cláusula SELECT de uma visão;
- Em um comando SELECT com a palavra chave DISTINCT;
- Em um comando SELECT com as cláusulas GROUP BY, HAVING ou ORDER BY;
- Em uma sub-consulta no comando SELECT, DELETE ou UPDATE;
- Na expressão da opção DEFAULT em um comando CREATE TABLE ou ALTER TABLE.



Utilizando uma Sequence

- Ex.:

```
insert into atendimento
```

```
values(seq_cod_atendimento.nextval,1,3,sysdate,',75.5,7.5);
```

- O exemplo acima insere um novo atendimento na tabela ATENDIMENTOS, utilizando a sequence SEQ_ATENDIMENTO.NEXTVAL para gerar um novo número de cliente.
- Você pode visualizar o valor atual da sequence:

```
select seq_cod_atendimento.currval from dual;
```



Modificando uma Sequence

- EX.:

```
ALTER SEQUENCE seq_cod_pac  
INCREMENT BY 1  
MAXVALUE 999999  
NOCACHE  
NOCYCLE;
```

- Se você atingir o limite MAXVALUE para uma sequence, não serão alocados valores adicionais da sequence e você receberá um erro que indica o valor de MAXVALUE foi excedido. Para continuar utilizando a sequence, você pode modificá-la utilizando o comando acima.

**MV**

Diretrizes para Modificar uma Sequence

- Você deve ser o dono ou possuir o privilégio ALTER para a sequence para poder modificá-la;
- Somente os próximos números da sequence serão afetados pelo comando ALTER SEQUENCE;
- A opção START WITH não pode ser modificada utilizando o comando ALTER SEQUENCE. A sequence deve ser removida e recriada para reiniciar em um número diferente;
- Algumas validações são executadas. Por exemplo, um novo MAXVALUE não pode ser menos que o número atual da sequence.



Diretrizes para Modificar uma Sequence

Ex.:

```
ALTER SEQUENCE seq_cod_pac  
    INCREMENT BY 1  
    MAXVALUE 90  
    NOCACHE  
    NOCYCLE;  
ALTER SEQUENCE seq_cod_pac
```

*

ERROR at line 1:

ORA-04009: MAXVALUE cannot be made to be less than the current value





Removendo uma Sequence

- Ex.:

```
DROP SEQUENCE seq_cod_pac
```

- Para remover uma sequence do dicionário de dados, utilize o comando **DROP SEQUENCE**. Você deve ser o dono da sequence ou ter o privilégio **DROP ANY SEQUENCE** para removê-la.

- Sintaxe:

```
DROP SEQUENCE sequence;
```



O que é um Índice?

- Um índice do Servidor Oracle é um objeto de um schema que pode acelerar a recuperação das linhas utilizando um ponteiro. Podem ser criados explicitamente ou automaticamente. Se você não possuir um índice em uma coluna, então um método de acesso chamado full table scan ocorrerá.
- Um índice fornece acesso direto e rápido para as linhas de uma tabela. Seu propósito é reduzir a necessidade de I/O de disco utilizando um caminho indexado para localizar os dados rapidamente. O índice é automaticamente utilizado e mantido pelo Servidor Oracle.
- Índices são lógica e fisicamente independentes da tabela que indexam. Isto significa que eles podem ser criados ou removidos a qualquer momento e não causam nenhum efeito nas tabelas ou outros índices.
- Nota: quando você remove uma tabela, os índices correspondentes também são removidos.



Como os Índices são Criados?

- Podem ser criados dois tipos de índices. Um tipo são os índices únicos. O Servidor Oracle cria este tipo de índice automaticamente quando você define para uma coluna em uma tabela uma constraint PRIMARY KEY ou UNIQUE KEY. O nome do índice é o nome fornecido para a constraint.
- O outro tipo de índice que um usuário pode criar é um índice não único (nonunique). Por exemplo, você pode criar um índice em uma coluna definida como FOREIGN KEY para melhorar a performance de um JOIN em uma consulta.



Criando um Índice

- Crie um índice em uma ou mais colunas:
`CREATE INDEX índice ON table (column[, column]...);`
- Aumente a velocidade de acesso da consulta na coluna NOME da tabela PACIENTES:
`CREATE INDEX nome_pac_idx
ON paciente(nome);`
- Crie um índice em uma ou mais colunas executando o comando `CREATE INDEX`.

Sintaxe:

index: é o nome do índice

table: é o nome da tabela

column: é o nome da coluna da tabela a ser indexada

**MV**

Diretrizes para a Criação de Índices

- **Mais nem sempre é Melhor**
- Muitos índices em uma tabela nem sempre significam melhora na velocidade das consultas.
- Para cada operação DML, que é confirmada em uma tabela os índices devem ser atualizados.
- Quanto mais índices você associou à uma tabela, mais esforço o Servidor Oracle terá que realizar para atualizar todos os índices após uma operação DML.



Quando Criar um Índice

- A coluna é freqüentemente utilizada na cláusula WHERE ou em uma condição de join.
- A coluna possui uma grande faixa de valores.
- A coluna possui um grande número de valores nulos.
- Duas ou mais colunas são freqüentemente utilizadas juntas em uma cláusula WHERE ou condição de join.
- Lembre-se que se você quiser garantir unicidade, você deve definir uma constraint do tipo UNIQUE na definição da tabela. Então, um índice único será criado automaticamente.



Quando não Criar um Índice

- A tabela for pequena;
- As colunas não são utilizadas freqüentemente como uma condição da consulta;
- A maioria das consultas recuperam mais que 2% a 4% das linhas;
- A tabela é freqüentemente atualizada. Se você possui um ou mais índices em uma tabela, os comandos DML que acessam a tabela utilizam relativamente mais tempo para sua execução devido a manutenção dos índices.



Confirmando Índices

- Ex.:

```
SELECT ic.index_name, ic.column_name, ic.column_position  
       col_pos, ix.uniqueness  
FROM   user_indexes ix, user_ind_columns ic  
WHERE  ic.index_name = ix.index_name  
AND    ic.table_name = 'PACIENTE';
```

- Confirme a existência de índices a partir da visão do dicionário de dados USER_INDEXES. Você também pode conferir as colunas envolvidas em um índice examinando a visão USER_IND_COLUMNS.
- O exemplo acima exibe todos os índices previamente criados, nomes das colunas afetadas e a unicidade para a tabela PACIENTE.



Removendo um Índice

- Remova um índice do dicionário de dados:
- Sintaxe:
`DROP INDEX índice;`
- Ex.:
`DROP INDEX nome_pac_idx;`
- Você não pode modificar índices. Para alterar um índice, você deve removê-lo então criá-lo novamente. Remova a definição de um índice do dicionário de dados executando o comando `DROP INDEX`. Para remover um índice, você deve ser o dono do índice ou possuir o privilégio `DROP ANY INDEX`.

- Sintaxe:
`CREATE SYNONYM synonym FOR object;`
- Para referenciar uma tabela criada por outro usuário, você deve prefixar o nome da tabela com o nome do usuário que a criou seguido por um ponto.
- Criando um sinônimo você elimina a necessidade de qualificar o nome do objeto com o schema e o provê um nome alternativo para uma tabela, visão, sequence, procedure ou outros objetos.
- Este método pode ser especialmente útil com nomes de objetos longos, como visões.



Sinônimos

- Sintaxe:
 - *PUBLIC*: cria um sinônimo acessível a todos os usuários.
 - *synonym*: é o nome do sinônimo a ser criado.
 - *object*: identifica o objeto para o qual o sinônimo deve ser criado.

- O objeto não pode estar contido em uma package;
- Um nome de sinônimo privado deve ser distinto de todos os outros objetos criados pelo mesmo usuário.

Criando ou Removendo Sinônimos

- Ex.:

```
CREATE SYNONYM med FOR medico;
```

- O exemplo acima cria um sinônimo para a tabela medico para uma referencia mais rápida.
- O DBA pode criar um sinônimo publico acessível para todos os usuários. O exemplo abaixo cria um sinônimo público chamado PAC para a tabela PACIENTE do usuário MARIA:

- Ex.:

```
CREATE PUBLIC SYNONYM paciente  
FOR Maria.paciente;
```



Removendo um Sinônimo

- Para remover um sinônimo, utilize o comando DROP SYNONYM. Somente um DBA pode remover um sinônimo publico.
- Ex.:
`DROP SYNONYM paciente;`