





Visões – Sequencias – Sinônimos – Índices

(Parte 2)



Objetivos

Ao concluir esta lição, você será capaz de:

- Criar views simples e complexas
- Recuperar dados de views

Objetivos

Esta lição apresenta os objetos view, sequência, sinônimo e índice. Você conhecerá os princípios básicos de como criar e usar views.

Objetivos

Ao concluir esta lição, você será capaz de:

- Criar, manter e usar sequências
- Criar e manter índices
- Criar sinônimos privados e públicos

Objetivos

Esta lição apresenta os objetos view, sequência, sinônimo e índice. Você conhecerá os princípios básicos de como criar e usar views, sequências e índices.

Sequências

Objeto	Descrição
Tabela	Unidade básica de armazenamento; composta de linhas
View	Representa logicamente subconjuntos de dados de uma ou mais tabelas
Sequência	Gera valores numéricos
Índice	Melhora o desempenho de algumas consultas
Sinônimo	Fornece nomes alternativos a objetos

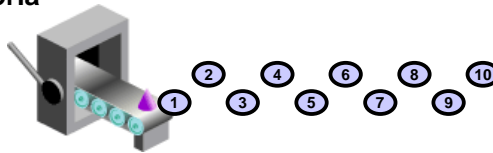
Sequências

Uma sequência é um objeto de banco de dados que cria valores inteiros. Você pode criar sequências e utilizá-las para gerar números.

Sequências

Uma sequência:

- Pode gerar números exclusivos automaticamente
- É um objeto compartilhável
- Pode ser usada para criar um valor de chave primária
- Substitui o código da aplicação
- Aumenta a eficiência do acesso a valores de sequência quando armazenados no cache da memória



1-6

Sequências

Uma sequência é um objeto de banco de dados criado por um usuário que pode ser compartilhado por vários usuários para gerar inteiros.

Você pode definir uma sequência para gerar valores exclusivos ou para reciclar e reutilizar os mesmos números.

As sequências são muito usadas para criar um valor de chave primária, que precisa ser exclusivo para cada linha. A sequência é gerada e incrementada (ou decrementada) por uma rotina interna do Oracle. Esse objeto pode poupar tempo reduzindo o tamanho do código da aplicação necessário para criar uma rotina de geração de sequência.

Os números de sequência são armazenados e gerados independentemente de tabelas. Portanto, é possível usar a mesma sequência para várias tabelas.

Instrução CREATE SEQUENCE: Sintaxe

Defina uma sequência para gerar números sequenciais automaticamente:

```
CREATE SEQUENCE sequence
  [INCREMENT BY n]
  [START WITH n]
  [{MAXVALUE n | NOMAXVALUE}]
  [{MINVALUE n | NOMINVALUE}]
  [{CYCLE | NOCYCLE}]
  [{CACHE n | NOCACHE}];
```

1-7

Criando uma Sequência

Gere automaticamente números sequenciais usando a instrução CREATE SEQUENCE.

Na sintaxe:

sequence

INCREMENT BY *n*

esta cláusula

incrementada em 1.)

START WITH *n*

sequência

MAXVALUE *n*

NOMAXVALUE

uma

a opção

MINVALUE *n*

NOMINVALUE

é o nome do gerador de sequência

especifica o intervalo entre os números da sequência, em que *n* é um inteiro (Se

for omitida, a sequência será

especifica o primeiro número da sequência a ser gerado (Se esta cláusula for omitida, a

começará com 1.)

especifica o valor máximo que a sequência pode gerar

especifica o valor máximo 10^{27} para uma sequência em ordem crescente e -1 para

sequência em ordem decrescente (Esta é

default.)

especifica o valor mínimo da sequência

especifica o valor mínimo 1 para uma sequência

uma
(Esta é a opção
em ordem crescente e $-(10^{26})$ para
sequência em ordem decrescente
default.)

+ Criando uma Sequência

- Crie uma sequência denominada DEPT_DEPTID_SEQ a ser usada para a chave primária da tabela DEPARTMENTS.
- Não use a opção CYCLE.

```
CREATE SEQUENCE dept_deptid_seq  
    INCREMENT BY 10  
    START WITH 120  
    MAXVALUE 9999  
    NOCACHE  
    NOCYCLE;
```

Sequence created.

1-8

Criando uma Sequência (continuação)

CYCLE | NOCYCLE
a gerar

máximo ou

default.)

CACHE *n* | NOCACHE
Oracle

(Por default, o

no cache.)

especifica se a sequência continuará

valores depois de atingir seu valor

mínimo (NOCYCLE é a opção

especifica quantos valores o servidor

pré-aloca e mantém na memória

servidor Oracle armazena 20 valores

O exemplo do slide cria uma sequência denominada DEPT_DEPTID_SEQ a ser usada para a coluna DEPARTMENT_ID da tabela DEPARTMENTS. A sequência começa em 120, não permite armazenamento em cache e não percorre um ciclo.

Não use a opção CYCLE se a sequência for usada para gerar valores de chave primária, a menos que você tenha um mecanismo confiável que expurgue as linhas antigas mais rápido que os ciclos da sequência.

Para obter mais informações, consulte "CREATE SEQUENCE" no manual *Oracle SQL Reference*.

Observação: A sequência não é vinculada a uma tabela. Em geral, nomeie a sequência de acordo com o uso pretendido. No entanto, é possível usar a sequência em qualquer local, independentemente de seu nome.

Pseudocolunas NEXTVAL e CURRVAL

- **NEXTVAL** retorna o próximo valor disponível da sequência. Quando referenciada, mesmo que por usuários distintos, ela retorna um valor exclusivo.
- **CURRVAL** obtém o valor atual da sequência.
- É necessário executar **NEXTVAL** para a sequência antes que **CURRVAL** contenha um valor.

Pseudocolunas NEXTVAL e CURRVAL

Depois que você cria uma sequência, ela gera números sequenciais para uso nas suas tabelas. Faça referência aos valores da sequência usando as pseudocolunas **NEXTVAL** e **CURRVAL**.

A pseudocoluna **NEXTVAL** é usada para extrair números sucessivos de uma sequência especificada. Qualifique **NEXTVAL** com o nome da sequência. Quando você fizer referência a *sequência.NEXTVAL*, um novo número da sequência será gerado e o número atual da sequência será inserido em **CURRVAL**.

A pseudocoluna **CURRVAL** é usada para fazer referência a um número da sequência que o usuário atual acabou de gerar. É preciso usar **NEXTVAL** para gerar um número de sequência na sessão do usuário atual antes que seja possível fazer referência a **CURRVAL**. Qualifique **CURRVAL** com o nome da sequência. Quando você fizer referência a *sequência.CURRVAL*, o último valor retornado para o processo desse usuário será exibido.

Pseudocolunas NEXTVAL e CURRVAL (continuação)

Regras de Uso de NEXTVAL e CURRVAL

Você pode usar NEXTVAL e CURRVAL nos seguintes contextos:

- Lista SELECT de uma instrução SELECT que não faz parte de uma subconsulta

- Lista SELECT de uma subconsulta em uma instrução INSERT

- Cláusula VALUES de uma instrução INSERT

- Cláusula SET de uma instrução UPDATE

Você não pode usar NEXTVAL e CURRVAL nos seguintes contextos:

- Lista SELECT de uma view

- Instrução SELECT com a palavra-chave DISTINCT

- Instrução SELECT com as cláusulas GROUP BY, HAVING ou ORDER BY

- Subconsulta em uma instrução SELECT, DELETE ou UPDATE

- Expressão DEFAULT em uma instrução CREATE TABLE ou ALTER TABLE

Para obter mais informações, consulte "Pseudocolumns" e "CREATE SEQUENCE" no manual *Oracle SQL Reference*.

+ Usando uma Sequência

- Insira um novo departamento denominado "Support" no ID de local 2500:

```
INSERT INTO departments (department_id,
                        department_name, location_id)
VALUES (dept_deptid_seq.NEXTVAL,
        'Support', 2500);
1 row created.
```

- Exiba o valor atual da sequência DEPT_DEPTID_SEQ:

```
SELECT dept_deptid_seq.CURRVAL
FROM dual;
```

1-11

Usando uma Sequência

O exemplo do slide insere um novo departamento na tabela DEPARTMENTS. Ele usa a sequência DEPT_DEPTID_SEQ para gerar um novo número de departamento como mostrado a seguir.

Você pode exibir o valor atual da sequência:

```
SELECT dept_deptid_seq.CURRVAL
FROM dual;
```

CURRVAL

120

funcionários pode incluir o seguinte código.

```
INSERT INTO employees (employee_id, department_id,
...)
VALUES (employees_seq.NEXTVAL, dept_deptid_seq
.CURRVAL, ...);
```

Observação: O exemplo anterior supõe que uma sequência denominada EMPLOYEE_SEQ já tenha sido criada para gerar novos números de funcionários.

Armazenando Valores de Sequência no Cache

- O armazenamento de valores da sequência no cache da memória permite um acesso mais rápido a esses valores.
- Os intervalos em valores de sequência ocorrem quando:
 - É efetuado um rollback
 - Ocorre uma falha do sistema
 - Uma sequência é usada em outra tabela

1-12

Armazenando Valores de Sequência no Cache

É possível armazenar sequências no cache da memória para permitir o acesso mais rápido aos valores da sequência. O cache é preenchido na primeira vez em que você faz referência à sequência. Cada solicitação do próximo valor da sequência é recuperada da sequência armazenada no cache. Depois que o último valor da sequência é usado, a próxima solicitação dessa sequência armazena outro cache de sequências na memória.

Intervalos da Sequência

Embora os geradores de sequência gerem números sequenciais sem intervalos, essa ação ocorre independentemente de um commit ou rollback. Portanto, se você executar rollback de uma instrução que contenha uma sequência, o número será perdido.

Outro evento que pode causar intervalos na sequência é uma falha do sistema. Se a sequência armazenar valores no cache da memória, esses valores serão perdidos caso ocorra uma falha do sistema.

Como as sequências não estão diretamente vinculadas a tabelas, é possível usar a mesma sequência para várias tabelas. Se você fizer isso, cada tabela poderá conter intervalos nos números sequenciais.

Modificando uma Sequência

Altere o valor de incremento, o valor máximo, o valor mínimo, a opção de ciclo ou a opção de cache:

```
ALTER SEQUENCE dept_deptid_seq
    INCREMENT BY 20
    MAXVALUE 999999
    NOCACHE
    NOCYCLE;
Sequence altered.
```

1-13

Modificando uma Sequência

Se o limite `MAXVALUE` definido para a sequência for atingido, não serão alocados outros valores da sequência e você receberá um erro indicando que essa sequência excede `MAXVALUE`. Para continuar a usar a sequência, você poderá modificá-la usando a instrução `ALTER SEQUENCE`.

Sintaxe

```
ALTER SEQUENCE sequence
    [INCREMENT BY n]
    [{MAXVALUE n | NOMAXVALUE}]
    [{MINVALUE n | NOMINVALUE}]
    [{CYCLE | NOCYCLE}]
    [{CACHE n | NOCACHE}];
```

Na sintaxe, *sequence* é o nome do gerador de sequência.

Para obter mais informações, consulte "ALTER SEQUENCE" no manual *Oracle SQL Reference*.

Diretrizes para Modificar uma Sequência

- Você precisa ser o proprietário da sequência ou ter o privilégio **ALTER**.
- Somente os futuros números da sequência são afetados.
- É necessário eliminar e recriar a sequência para reiniciá-la em um número diferente.
- É executada uma validação.
- Para remover uma sequência, use a instrução **DROP**:

```
DROP SEQUENCE dept_deptid_seq;  
Sequence dropped.
```

1-14

Diretrizes para Modificar uma Sequência

Você precisa ser o proprietário da sequência ou ter o privilégio **ALTER** para modificá-la. Você precisa ser o proprietário da sequência ou ter o privilégio **DROP ANY SEQUENCE** para removê-la.

Somente os futuros números da sequência serão afetados pela instrução **ALTER SEQUENCE**.

Não é possível alterar a opção **START WITH** usando **ALTER SEQUENCE**. É necessário eliminar e recriar a sequência para reiniciá-la em um número diferente.

É executada uma validação. Por exemplo, não é possível impor um novo **MAXVALUE** menor que o número atual da sequência.

```
ALTER SEQUENCE dept_deptid_seq  
INCREMENT BY 20  
MAXVALUE 90  
NOCACHE  
NOCYCLE;
```

```
ALTER SEQUENCE dept_deptid_seq
```

*

ERROR at line 1:

ORA-04009: MAXVALUE cannot be made to be less than

the current value

Índices

Objeto	Descrição
Tabela	Unidade básica de armazenamento; composta de linhas
View	Representa logicamente subconjuntos de dados de uma ou mais tabelas
Sequência	Gera valores numéricos
Índice	Melhora o desempenho de algumas consultas
Sinônimo	Fornece nomes alternativos a objetos

Índices

Os índices são objetos de banco de dados que você pode criar para melhorar o desempenho de algumas consultas. Eles também podem ser criados automaticamente pelo servidor quando você cria uma constraint exclusiva ou de chave primária.

Índices

Um índice:

- É um objeto de esquema
- É usado pelo servidor Oracle para acelerar a recuperação de linhas usando um ponteiro
- Pode reduzir a entrada/saída de disco por um método de acesso de caminho rápido para localizar dados rapidamente
- É independente da tabela que indexa
- É usado e mantido automaticamente pelo servidor Oracle



1-16

Índices (continuação)

Um índice do servidor Oracle é um objeto de esquema que pode acelerar a recuperação de linhas usando um ponteiro. É possível criar índices de forma explícita ou automática. Se você não tiver um índice na coluna, ocorrerá uma varredura integral da tabela.

Um índice permite acesso direto e rápido a linhas de uma tabela. Seu objetivo é reduzir a necessidade de entrada/saída de disco usando um caminho indexado para localizar os dados com rapidez. O servidor Oracle usa e mantém automaticamente o índice. Depois que um índice é criado, não é exigida qualquer atividade direta do usuário.

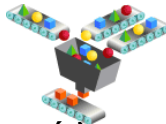
Os índices são independentes do ponto de vista lógico e físico da tabela que indexam. Isso significa que é possível criá-los ou eliminá-los a qualquer momento sem causar efeitos nas tabelas-base ou em outros índices.

Observação: Quando você elimina uma tabela, os índices correspondentes também são eliminados.

Para obter mais informações, consulte "Schema Objects: Indexes" em *Database Concepts*.

Como Criar Índices?

- **Automaticamente:** Um índice exclusivo é criado automaticamente quando você define uma constraint de chave `PRIMARY` ou `UNIQUE` em uma definição de tabela.



- **Manualmente:** Os usuários podem criar índices não exclusivos em colunas para acelerar o acesso às linhas.



1-17

Tipos de Índice

É possível criar dois tipos de índice.

Índice exclusivo: O servidor Oracle cria automaticamente este índice quando você define uma constraint de chave `PRIMARY KEY` ou `UNIQUE` para a coluna de uma tabela. O nome do índice é aquele fornecido à constraint.

Índice não exclusivo: É um índice que um usuário pode criar. Por exemplo, você pode criar um índice de coluna `FOREIGN KEY` para uma join em uma consulta a fim de aumentar a velocidade de recuperação.

Observação: Você pode criar um índice exclusivo de forma manual, mas é recomendável criar uma constraint exclusiva, que gera implicitamente um índice exclusivo.

Criando um Índice

- Crie um índice em uma ou mais colunas:

```
CREATE INDEX index
ON table (column[, column]...);
```

- Aumente a velocidade de acesso da consulta à coluna `LAST_NAME` da tabela `EMPLOYEES`:

```
CREATE INDEX emp_last_name_idx
ON employees(last_name);
Index created.
```

1-18

Criando um Índice

Crie um índice em uma ou mais colunas executando a instrução `CREATE INDEX`.

Na sintaxe:

<i>index</i>	é o nome do índice
<i>table</i>	é o nome da tabela
<i>column</i>	é o nome da coluna da tabela a ser indexada

Para obter mais informações, consulte "CREATE INDEX" no manual *Oracle SQL Reference*.

Diretrizes para a Criação de um Índice

Crie um índice quando:	
<input checked="" type="checkbox"/>	Uma coluna conter uma grande faixa de valores
<input checked="" type="checkbox"/>	Uma coluna conter um grande número de valores nulos
<input checked="" type="checkbox"/>	Uma ou mais colunas forem usadas em conjunto com frequência em uma cláusula <code>WHERE</code> ou em uma condição de
<input checked="" type="checkbox"/>	A tabela for grande e for esperado que a maioria das consultas recupere menos de 2% a 4% das linhas da tabela
Não crie um índice quando:	
<input checked="" type="checkbox"/>	As colunas não forem usadas com frequência como uma condição na consulta
<input checked="" type="checkbox"/>	A tabela for pequena ou for esperado que a maioria das consultas recupere mais de 2% a 4% das linhas da tabela
<input checked="" type="checkbox"/>	A tabela for atualizada com frequência
<input checked="" type="checkbox"/>	As colunas indexadas forem referenciadas como parte de uma expressão

1-19

Mais Não Significa Melhor

Um número maior de índices em uma tabela não resulta em consultas mais rápidas. Para cada operação DML submetida a commit em uma tabela com índices, é necessário atualizar os índices. Quanto mais índices estiverem associados a uma tabela, maior será o esforço do servidor Oracle para atualizar todos os índices após uma operação DML.

Quando Criar um Índice

Portanto, você só deverá criar índices se:

- A coluna conter uma grande faixa de valores

- A coluna conter um grande número de valores nulos

- Uma ou mais colunas forem usadas em conjunto com frequência em uma cláusula `WHERE` ou em uma condição de join

- A tabela for grande e for esperado que a maioria das consultas recupere menos de 2% a 4% das linhas

Lembre-se de que, para impor a exclusividade, você deverá definir uma constraint exclusiva na definição da tabela. Depois, um índice exclusivo será criado automaticamente.

Removendo um Índice

- Para remover um índice do dicionário de dados, use o comando `DROP INDEX`:

```
DROP INDEX index;
```

- Remova o índice `UPPER_LAST_NAME_IDX` do dicionário de dados:

```
DROP INDEX emp_last_name_idx;  
Index dropped.
```

- Para eliminar um índice, você precisa ser o proprietário dele ou ter o privilégio `DROP ANY INDEX`.

Removendo um Índice

Não é possível modificar índices. Para alterar um índice, elimine-o e, depois, recrie-o.

Remova uma definição de índice do dicionário de dados executando a instrução `DROP INDEX`. Para eliminar um índice, você precisa ser o proprietário dele ou ter o privilégio `DROP ANY INDEX`.

Na sintaxe, *index* é o nome do índice.

Observação: Se você eliminar uma tabela, os índices e as constraints serão eliminados automaticamente, mas as views e as sequências permanecerão.

Sinônimos

Objeto	Descrição
Tabela	Unidade básica de armazenamento; composta de linhas
View	Representa logicamente subconjuntos de dados de uma ou mais tabelas
Sequência	Gera valores numéricos
Índice	Melhora o desempenho de algumas consultas
Sinônimo	Fornece nomes alternativos a objetos

Sinônimos

Os sinônimos são objetos de banco de dados que permitem chamar uma tabela usando outro nome. É possível criar sinônimos para atribuir um nome alternativo a uma tabela.

Sinônimos

Simplifique o acesso a objetos criando um sinônimo (outro nome para um objeto). Com sinônimos, você pode:

- Criar uma referência mais fácil a uma tabela pertencente a outro usuário
- Reduzir nomes longos de objetos

```
CREATE [PUBLIC] SYNONYM synonym  
FOR object;
```

1-22

Criando um Sinônimo para um Objeto

Para fazer referência a uma tabela pertencente a outro usuário, é preciso inserir o nome do autor dessa tabela seguido de um ponto como prefixo do nome da tabela. A criação de um sinônimo elimina a necessidade de qualificar o nome do objeto com o esquema e fornece um nome alternativo para uma tabela, uma view, uma sequência, um procedure ou outros objetos. Esse método pode ser útil especialmente com nomes longos de objetos, como views.

Na sintaxe:

<code>PUBLIC</code>	cria um sinônimo acessível a todos os usuários
<code><i>synonym</i></code>	é o nome do sinônimo a ser criado
<code><i>object</i></code>	identifica o objeto para o qual o sinônimo é criado

Diretrizes

O objeto não pode estar contido em um package.

O nome de um sinônimo privado deve ser diferente do de outros objetos pertencentes ao mesmo usuário.

Para obter mais informações, consulte "CREATE SYNONYM" no manual *Oracle SQL Reference*.

Criando e Removendo Sinônimos

- Crie um nome abreviado para a view DEPT_SUM_VU:

```
CREATE SYNONYM d_sum  
FOR dept_sum_vu;  
Synonym Created.
```

- Elimine um sinônimo:

```
DROP SYNONYM d_sum;  
Synonym dropped.
```

1-23

Criando um Sinônimo

O exemplo do slide cria um sinônimo relativo à view DEPT_SUM_VU para agilizar a referência.

O administrador do banco de dados pode criar um sinônimo público acessível a todos os usuários. Este exemplo cria um sinônimo público denominado DEPT para a tabela DEPARTMENTS da usuária Alice:

```
CREATE PUBLIC SYNONYM dept  
FOR alice.departments;  
Synonym created.
```

Removendo um Sinônimo

Para remover um sinônimo, use a instrução DROP SYNONYM. Somente o administrador do banco de dados pode eliminar um sinônimo público.

```
DROP PUBLIC SYNONYM dept;  
Synonym dropped.
```

Para obter mais informações, consulte "DROP SYNONYM" no manual *Oracle SQL Reference*.

Sumário

Nesta lição, você aprendeu a:

- Gerar automaticamente números sequenciais usando um gerador de sequência
- Criar índices para aumentar a velocidade de recuperação de consultas
- Usar sinônimos para fornecer nomes alternativos a objetos

Exercício: Visão Geral da Parte 2

• • • • •
• • • • •
• + •
+ •

Este exercício aborda os seguintes tópicos:

- Criando sequências
- Usando sequências
- Criando índices não exclusivos
- Criando sinônimos

|
+

1-25

•
• +
■ □ •
• •
•
• + • •
• • •

Exercício : Visão Geral da Parte 2

A Parte 2 do exercício desta lição contém várias atividades que permitem criar e usar uma sequência, um índice e um sinônimo.

Faça as questões de 7 a 10 no final desta lição.

Exercício

Parte 2

7. Você precisa de uma sequência que possa ser usada com a coluna de chave primária da tabela `DEPT`. A sequência deve começar com o valor 200 e ter o valor máximo 1.000. Incremente a sequência em 10. Nomeie-a como `DEPT_ID_SEQ`.
8. Para testar a sequência, crie um script para inserir duas linhas na tabela `DEPT`. Nomeie o script como `lab_08.sql`. Certifique-se de utilizar a sequência criada para a coluna `ID`. Adicione dois departamentos: Education e Administration. Confirme as adições. Execute os comandos no script.
9. Crie um índice não exclusivo na coluna `DEPT_ID` da tabela `DEPT`.
10. Crie um sinônimo para a tabela `EMPLOYEES`. Nomeie o sinônimo como `EMP`.

OBRIGADO



profalexandre.barcelos@fiap.com.br



<https://www.linkedin.com/in/alexandrebarcelos>

FIAP

Copyright © 2022 | Professor Me. Alexandre Barcelos
Todos os direitos reservados. Reprodução ou divulgação total ou parcial deste documento, é expressamente
proibido sem consentimento formal, por escrito, do professor/autor.

