**O Guia Completo para Sequências Oracle**

por [Ben](https://www.databasestar.com/author/ben/) | 13 de novembro de 2017 | [1 comentário](https://www.databasestar.com/oracle-sequence/#respond)

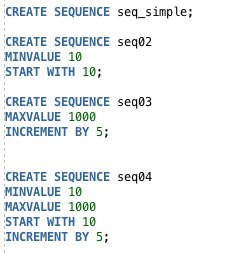
Neste artigo, explicarei o que é uma sequência Oracle, como criar uma e como usá-las.

**O que é uma sequência no Oracle?**

Uma sequência é um objeto de banco de dados usado para gerar números exclusivos.

Eles costumam ser usados ​​para gerar chaves primárias, pois a [chave primária](https://www.databasestar.com/database-keys/) precisa ser única e não precisa ser um valor relevante para seu aplicativo ou negócio.

Gerar um número exclusivo com uma sequência é fácil depois que ele é criado.



**Por que usar uma sequência?**

Você pode estar se perguntando, por que eu gostaria de usar uma sequência?

Eu não poderia simplesmente usar uma tabela que armazena a Chave Primária e a incrementa toda vez? Ou defina uma coluna como INCREMENTO AUTOMÁTICO?

Bem, uma sequência é uma ótima opção por alguns motivos.

Se você estiver executando o Oracle 12c, poderá usar um novo recurso que permite criar uma coluna de incremento automático. Você pode [ler mais sobre isso aqui](https://www.databasestar.com/auto_increment-oracle-sql/) . No entanto, isso foi introduzido apenas no Oracle 12c; portanto, se você estiver com 11g ou 10g ou anterior, não poderá usar esse recurso.

A outra opção seria usar uma tabela para armazenar um valor de chave primária. Isso pode ser feito algo como isto:

1. [Crie uma nova tabela](https://www.databasestar.com/sql-create-table/) para armazenar um único valor
2. Sempre que você inserir um novo valor na tabela desejada, use esse valor na nova tabela
3. Atualize a nova tabela aumentando-a em um
4. Essa lógica pode entrar em um [procedimento armazenado,](https://www.databasestar.com/sql-stored-procedures/) se necessário.

No entanto, existem vários problemas com essa abordagem:

* Se sua [transação](https://www.databasestar.com/sql-transactions/) falhar no meio do caminho, você poderá acabar com dados duplicados (os dados inseridos têm um ID, mas a nova tabela não foi incrementada). Isso pode ser parcialmente evitado incrementando primeiro e depois inserindo, mas ainda existe o risco de falha na metade.
* A atualização de uma tabela colocará um bloqueio nela, o que pode causar problemas para várias transações ou usuários. Isso pode ser bom se houver um usuário, mas se seu aplicativo tiver dois ou mais, isso poderá causar problemas.
* É mais trabalho criar e manter uma tabela e o código para fazer isso.

Uma sequência é uma abordagem muito melhor porque:

* É mais simples de criar, como veremos em breve.
* É mais simples de usar
* É seguro para transações - sem risco de bloquear e causar problemas em outras transações.

Vamos dar uma olhada em como você pode criar uma sequência.Parte inferior do formulário

**Criar sequência no Oracle**

Para criar uma sequência no Oracle, usamos o comando **CREATE SEQUENCE** .

CREATE SEQUENCE [schema.]sequence\_name

[MINVALUE minval|NOMINVALUE]

[MAXVALUE maxval|NOMAXVALUE]

[START WITH startval]

[INCREMENT BY incrementval]

[CYCLE|NOCYCLE]

[CACHE cacheval|NOCACHE]

[ORDER|NOORDER];

Existem muitos parâmetros disponíveis, mas a maioria deles é opcional. Esses parâmetros são:

* **esquema** : o nome do [esquema em](https://www.databasestar.com/database-schema/) que a sequência será armazenada. Se omitido, a sequência é criada em seu próprio esquema.
* **sequence\_name** : o nome da sequência a ser criada. Este é o único parâmetro necessário.
* **MINVALUE minval:**  este é o valor mínimo gerado pela sequência. Deve ser um número inteiro de até 28 dígitos. Ele deve ser menor ou igual a INICIAR COM e deve ser menor que MAXVALUE.
* **NOMINVAL** : Indica que não existe um valor mínimo especificado para a sequência, mas usa -1026 como o mínimo para sequências descendentes e 1 para sequências ascendentes. Isso é especificado em vez de MINVALUE e é o valor padrão (em oposição a MINVALUE).
* **MAXVALUE maxval** : este é o valor máximo gerado pela sequência. Deve ser um número inteiro de até 28 dígitos. Ele deve ser maior ou igual a INICIAR COM e deve ser maior que MINVALUE.
* **NOMAXVALUE** : indica que não há um valor máximo especificado para a sequência, mas usa 1027 como o máximo para sequências ascendentes e -1 para sequências descendentes. Isso é especificado em vez de MAXVALUE e é o valor padrão (em oposição a MAXVALUE).
* **START WITH startval** : este é o primeiro número a ser gerado a partir da sequência. Para sequências ascendentes, o padrão é o valor mínimo e, para sequências descendentes, o padrão é o valor máximo. Pode ter até 28 dígitos.
* **INCREMENT BY incrementval** : este é o intervalo entre os números de sequência. Ele deve ser um número inteiro, pode ser positivo ou negativo, mas não pode ser 0. Pode ter até 28 dígitos. O valor absoluto deve ser menor que a diferença entre MAXVALUE e MINVALUE. Se o valor do incremento for negativo, a sequência será decrescente. Se for positivo, será executado em ordem crescente. Se isso for omitido, o valor padrão será 1.
* **CICLO** : Indica que a sequência continua gerando valores depois que MAXVALUE ou MINVALUE é atingido. Se a sequência estiver em ascensão e atingir o MAXVALUE, ela gera o MINVALUE. Se a sequência estiver descendente e atingir o MINVALUE, ela gera o MAXVALUE.
* **NOCYCLE** : Indica que a sequência não pode gerar mais valores depois que o máximo é atingido. Esse é o padrão.
* **CACHE cacheval** : são quantos valores da sequência que o banco de dados aloca e mantém na memória para poder acessá-lo mais rapidamente. Ele deve ser maior que 2 e pode ter até 28 dígitos. Se ocorrer uma falha no sistema, os valores em cache serão perdidos. Isso geralmente é bom, como se elas fossem usadas para chaves primárias, elas não precisam ser sequenciais. O valor CACHE precisa ser menor que o número de sequências em um ciclo, se a sequência estiver configurada para ciclo. O máximo para o valor CACHE é:

(CEIL (MAXVALUE - MINVALUE)) / ABS (INCREMENT)

* **NOCACHE** : Isso indica que nenhum valor de sequência é alocado e armazenado em cache. Se você omitir CACHE e NOCACHE, a configuração padrão é armazenar em cache 20 números de sequência.
* **PEDIDO** : Indica que os números de sequência são gerados na ordem em que são solicitados. Isso pode ser útil se você estiver usando os números de sequência para registros de data e hora.
* **NOORDER** : Indica que não é garantido que os números de sequência estejam na ordem em que foram solicitados. Este é o valor padrão.

Esta tabela indica alguns cenários usando os diferentes parâmetros para sequências:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **MINVALUE** | **VALOR MÁXIMO** | **COMEÇAR COM** | **INCREMENTO** | **CICLO** | **Descrição** |
| n / D | n / D | 1 | 1 | Não | Padrão. Valores de 1, 2, 3, 4, 5. |
| 10 | n / D | 10 | 1 | Não | Valores de 10, 11, 12, 13, 14. |
| n / D | 1000 | 1 | 5 | Não | Valores de 1, 6, 11, 16, 21. Erro causado quando o valor acima do máximo foi atingido. |
| 10 | 1000 | 10 | 5 | Não | Valores de 10, 15, 20, 25, 30. Erro causado quando o valor acima do máximo foi atingido. |
| 10 | 1000 | 10 | 50. | sim | Valores de 10, 60, 110, 160. Quando atingido o máximo, os valores de 10, 60, 110 são gerados novamente. |
| 10 | 500 | 500 | -1 | Não | Valores de 500, 499, 498, 497. Erro causado quando o valor abaixo do mínimo foi atingido. |
| 10 | 500 | 500 | -10 | sim | Valores de 500, 490, 480, 470. Quando atingido o mínimo, os valores de 500, 490, 480 são gerados novamente. |

**Criar exemplos de sequência**

Vamos ver alguns exemplos de criação de sequências.

**Exemplo 1: Simples**

Este é um exemplo de criação de uma sequência simples.

CREATE SEQUENCE seq\_simple;

A sequência é criada com todos os valores padrão e pode ser usada.

**Exemplo 2: MINVALUE e START WITH**

Este exemplo usa os parâmetros MINVALUE e START WITH.

CREATE SEQUENCE seq02

MINVALUE 10

START WITH 10;

Essa sequência começará no valor de 10 e gerará valores de 11, 12, 13 e assim por diante.

**Exemplo 3: MAXVALUE e INCREMENT**

Este exemplo usa os parâmetros MAXVALUE e INCREMENT.

CREATE SEQUENCE seq03

MAXVALUE 1000

INCREMENT BY 5;

Essa sequência começará no valor de 1 e aumentará em 5 (gerando valores de 6, 11, 16 e assim por diante), até atingir um valor acima de 1000. Isso causará um erro.

**Exemplo 4: MINVALUE, MAXVALUE, START WITH e INCREMENT**

Este exemplo usa todos os quatro desses parâmetros.

CREATE SEQUENCE seq04

MINVALUE 10

MAXVALUE 1000

START WITH 10

INCREMENT BY 5;

Isso criará uma sequência que gera valores de 10, 15, 20, 25 e assim por diante. Depois que um valor acima do máximo for gerado, ocorrerá um erro.

**Exemplo 5: CICLO**

Este exemplo usa o parâmetro CYCLE.

CREATE SEQUENCE seq05

MINVALUE 10

MAXVALUE 1000

START WITH 10

INCREMENT BY 50

CYCLE;

Essa sequência gerará valores de 10, 60, 110, 160 e assim por diante. Quando o máximo for atingido, em vez de gerar um erro, ele será reiniciado no valor mínimo de 10 e continuará com 60, 110 e assim por diante.

**Exemplo 6: NOCYCLE**

Essa sequência especifica o parâmetro NOCYCLE, que é o parâmetro padrão.

CREATE SEQUENCE seq06

MINVALUE 10

MAXVALUE 1000

START WITH 10

INCREMENT BY 50

NOCYCLE;

Essa sequência gerará valores de 10, 60, 110, 160 e assim por diante. Uma vez atingido o máximo, ocorrerá um erro.

**Exemplo 7: Incremento negativo**

Esta sequência especifica um valor negativo para o incremento.

CREATE SEQUENCE seq07

MINVALUE 10

MAXVALUE 500

START WITH 500

INCREMENT -1;

Essa sequência começará a gerar números de 500, 499, 498 e assim por diante. Quando ficar abaixo do mínimo, ocorrerá um erro.

**Exemplo 8: Incremento e ciclo negativos**

Este exemplo usa um valor de incremento negativo e um ciclo.

CREATE SEQUENCE seq08

MINVALUE 10

MAXVALUE 500

START WITH 500

INCREMENT BY -10;

Essa sequência começará a gerar números de 500, 490, 480 e assim por diante. Quando ficar abaixo do mínimo, retornará para 500, 490, 480 e assim por diante.

**Exemplo 9: CACHE**

Este exemplo usa o parâmetro CACHE.

CREATE SEQUENCE seq09

CACHE 50;

Essa sequência gerará valores de 1, 2, 3 e assim por diante, mas armazenará em cache os primeiros 50 valores.

**Exemplo 10: PEDIDO**

Este exemplo usa o parâmetro ORDER.

CREATE SEQUENCE seq10

ORDER;

Isso garantirá que os valores de sequência sejam gerados em ordem, o que se tornará aparente quando vários usuários estiverem usando o sistema.

**Usando sequências: Como gerar valores de sequência**

Então, agora criamos uma sequência para gerar os valores que queremos.

Como podemos gerar esses valores?

Podemos usar uma das duas pseudocolunas disponíveis para uma sequência:

* **CURRVAL** : obtém o valor atual da sequência.
* **NEXTVAL** : obtém o próximo valor da sequência e incrementa a sequência pelo seu valor de incremento.

Portanto, para obter o próximo valor da sequência (por exemplo, para usar em uma instrução INSERT para gerar uma [chave primária](https://www.databasestar.com/database-keys/) ), você precisará usar a pseudocoluna NEXTVAL.

INSERT INTO customer(customer\_id, full\_name)

VALUES (seq\_simple.NEXTVAL, ‘John Smith’);

Você pode consultar a tabela para ver qual valor foi inserido.

SELECT customer\_id, full\_name

FROM customer;

|  |  |
| --- | --- |
| **IDENTIFICAÇÃO DO CLIENTE** | **NOME COMPLETO** |
| 1 | John Smith |

É importante lembrar que, ao chamar a pseudocoluna NEXTVAL, ele retorna um novo valor para cada linha de dados que está sendo selecionada ou inserida.

Por exemplo, podemos executar esta consulta para obter o NEXTVAL da sequência:

SELECT seq\_simple.NEXTVAL

FROM dual;

|  |
| --- |
| **NEXTVAL** |
| 2 |

Então, se eu executar esta consulta com vários comandos NEXTVAL:

SELECT seq\_simple.NEXTVAL AS nv1,

seq\_simple.NEXTVAL AS nv2,

seq\_simple.NEXTVAL AS nv3

FROM dual;

|  |  |  |
| --- | --- | --- |
| **NV1** | **NV2** | **NV3** |
| 3 | 3 | 3 |

Isso mostra que o NEXTVAL foi chamado apenas uma vez para este registro e o valor é o mesmo sempre.

Para encontrar o valor atual da sequência, sem aumentar o número, use CURRVAL.

SELECT seq\_simple.CURRVAL

FROM dual;

|  |
| --- |
| **CURRVAL** |
| 3 |

Essas duas pseudocolunas podem ser usadas em vários lugares:

* A cláusula VALUES de uma instrução INSERT.
* A lista SELECT de uma instrução SELECT
* A cláusula UPDATE de uma instrução UPDATE

Eles não podem ser usados ​​nesses locais:

* A condição de uma restrição CHECK
* Uma consulta de visualização
* Uma consulta de [exibição materializada](https://www.databasestar.com/sql-views/)
* Uma subconsulta
* Uma instrução SELECT com a palavra-chave [DISTINCT](https://www.databasestar.com/sql-distinct/) , a cláusula GROUP BY ou a cláusula ORDER BY
* Uma instrução SELECT combinada com outra instrução SELECT usando o operador [UNION](https://www.databasestar.com/sql-set-operators/) , INTERSECT ou MINUS
* A cláusula WHERE de uma instrução SELECT
* O valor DEFAULT de uma coluna em uma instrução CREATE TABLE ou ALTER TABLE

Este último ponto significa que você não pode definir uma coluna para aumentar automaticamente usando uma sequência usando a palavra-chave DEFAULT. Existe uma maneira de criar colunas de incremento automático e usar um [gatilho](https://www.databasestar.com/oracle-triggers/) e uma sequência, sobre as quais [escrevi aqui](https://www.databasestar.com/auto_increment-oracle-sql/) .

No Oracle 12c, um novo recurso foi adicionado para permitir a adição automática de colunas de incremento automático, sobre as quais você [pode ler aqui](https://www.databasestar.com/oracle-12c-new-features-for-developers/) .

**Alterar sequência**

Para alterar uma sequência após a criação, use o comando ALTER SEQUENCE.

Você pode alterar a maioria das propriedades de uma sequência depois que ela é criada. A única coisa que você não pode alterar é o número inicial da sequência e o valor atual da sequência. Para alterá-las, você precisará soltar e recriar a sequência.

Para usar o comando ALTER SEQUENCE:

ALTER SEQUENCE seq02

INCREMENT BY 20

MAXVALUE 100000

CYCLE;

Isso fará os ajustes acima na sequência seq02 existente.

**Sequência de descarte**

Para descartar uma sequência, ela deve estar em seu esquema ou você deve ter o privilégio DROP ANY SEQUENCE para descartar seqüências em outros esquemas.

Você pode soltar uma sequência usando o comando DROP SEQUENCE. Se parece com isso:

DROP SEQUENCE seq05;

Isso eliminará a sequência seq05 do banco de dados.

Então, é assim que você pode criar e usar sequências. Eles são objetos úteis de banco de dados e têm muitas vantagens sobre as alternativas.

Por fim, se você gosta das informações e conselhos de carreira que eu venho fornecendo, assine meu boletim abaixo para manter-se atualizado sobre meus artigos. Você também receberá um bônus fantástico. Obrigado!