



# Programación de bases de datos con SQL

16-1

Trabajar con Secuencias



# Objetivos

En esta lección se abordan los siguientes objetivos:

- Enumerar al menos tres características útiles de una secuencia
- Escribir y ejecutar una sentencia SQL que crea una secuencia
- Consultar el diccionario de datos mediante `USER_SEQUENCES` para confirmar una definición de secuencia
- Aplicar las reglas para utilizar `NEXTVAL` a fin de generar números únicos secuenciales en una tabla
- Enumerar las ventajas y desventajas del almacenamiento en caché de los valores de secuencia
- Nombrar tres motivos por los que se pueden producir intervalos en una secuencia



# Objetivo

- ¿Puede imaginar lo tedioso que puede ser tener que introducir los nombres de las 30.000 personas que participan en el Maratón de Londres en una base de datos, asegurándose al mismo tiempo de que no se le ha dado el mismo número de identificación a nadie?
- ¿Qué sucedería si se fuera a almorzar y al volver, otra persona hubiera introducido algunas de las solicitudes de los corredores?
- ¿Cómo sabría dónde volver a empezar?

# Objetivo

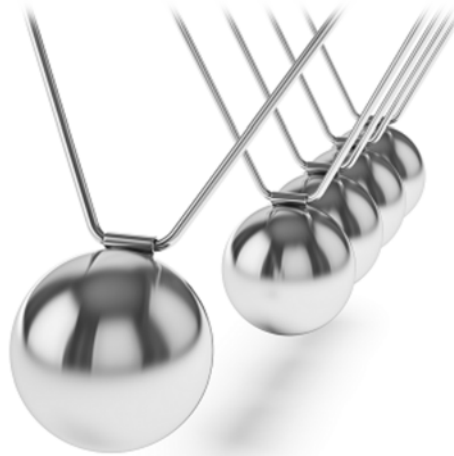
- Afortunadamente, SQL tiene un proceso para generar automáticamente números únicos que elimina la preocupación por los detalles de números duplicados.
- El proceso de numeración se maneja a través de un objeto de base de datos denominado SEQUENCE.

# Objeto Sequence

- Ya sabe cómo crear dos tipos de objetos de base de datos, TABLE y VIEW.
- Un tercer objeto de base de datos es SEQUENCE.
- SEQUENCE es un objeto que se puede compartir utilizado para generar automáticamente números únicos.
- Puesto que es un objeto que se puede, varios usuarios pueden acceder a él.
- Normalmente, se utilizan secuencias para crear un valor de clave primaria

# Objeto Sequence

- Como recordará, claves primarias deben ser únicas para cada fila. La secuencia se genera y aumenta (o disminuye) mediante una rutina interna de Oracle.
- Este objeto ahorra tiempo porque reduce la cantidad de código que tiene que escribir.



# Objeto Sequence

- Los números de secuencia se almacenan y generan independientemente de las tablas.
- Por lo tanto, la misma secuencia se puede utilizar para varias tablas.
- Para crear SEQUENCE:

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



# Sintaxis de Sequence

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

secuencia	es el nombre del generador de secuencias (objeto).
INCREMENT BY n	especifica el intervalo entre los números de secuencia, donde n es un número entero (si esta cláusula se omite, la secuencia se va incrementando en 1).
START WITH n	especifica el primer número de secuencia que se va a generar (si se omite esta cláusula, la secuencia empieza con 1).

# Sintaxis de Sequence

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

MAXVALUE n	especifique el valor máximo que puede generar la secuencia.
NOMAXVALUE	especifica un valor máximo de $10^{27}$ para una secuencia ascendente y $-1$ para una secuencia descendente (valor por defecto).
MINVALUE n	especifica el valor mínimo de secuencia.

# Sintaxis de Sequence

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

NOMINVALUE	especifica un valor máximo de 1 para una secuencia ascendente y de $- (10^{26})$ para una secuencia descendente (por defecto).
CYCLE   NOCYCLE	especifica si la secuencia sigue generando valores después de alcanzar su valor máximo o mínimo (NOCYCLE es la opción por defecto).

# Sintaxis de Sequence

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

**CACHE n | NOCACHE**

**especifica cuántos valores asigna previamente y mantiene Oracle Server en la memoria. (Por defecto, Oracle Server almacena en caché 20 valores). Si el sistema falla, los valores se pierden.**

# Creación de Secuencias

- En la SEQUENCE creada para los corredores del Maratón de Londres, los números se irán incrementando en 1, empezando por el número 1.
- En este caso, empezar la secuencia con 1 es probablemente el mejor punto de partida.
- Es tradicional que el mejor corredor del grupo de élite lleve el número 1.

```
CREATE SEQUENCE runner_id_seq  
  INCREMENT BY 1  
  START WITH 1  
  MAXVALUE 50000  
  NOCACHE  
  NOCYCLE;
```

# Creación de Secuencias

- Para otras situaciones, como, por ejemplo, identificadores de departamentos y de empleados se le puede asignar el número de inicio de forma diferente.
- Puesto no será habrá por lo menos 30.000 corredores, el valor máximo de la secuencia se ha definido muy por encima del número esperado de corredores.

```
CREATE SEQUENCE runner_id_seq  
  INCREMENT BY 1  
  START WITH 1  
  MAXVALUE 50000  
  NOCACHE  
  NOCYCLE;
```

# Creación de Secuencias

- La opción NOCACHE evita que los valores de SEQUENCE se almacenen en la caché de la memoria, con lo que, en caso de fallo del sistema, se evitaría que se perdieran los números asignados previamente y retenidos en la memoria.

```
CREATE SEQUENCE runner_id_seq  
  INCREMENT BY 1  
  START WITH 1  
  MAXVALUE 50000  
  NOCACHE  
  NOCYCLE;
```

# Creación de Secuencias

- La opción NOCYCLE evita que la numeración vuelva a empezar en 1 si se excede el valor 50.000.
- No utilice la opción CYCLE si la secuencia se utiliza para generar valores de clave primaria, a menos que disponga de un mecanismo fiable que suprima las filas antiguas más rápido que lo que se agregan las nuevas.

```
CREATE SEQUENCE runner_id_seq  
  INCREMENT BY 1  
  START WITH 1  
  MAXVALUE 50000  
  NOCACHE  
  NOCYCLE;
```



# Confirmación de Secuencias

- Para verificar que la secuencia se ha creado, consulte el diccionario de datos USER\_OBJECTS.
- Para ver toda la configuración de SEQUENCE, consulte el diccionario de datos USER\_SEQUENCES como se muestra a continuación.
- Visualice los nombres de valor en la sentencia SELECT, como se muestra a continuación.

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

# Confirmación de Secuencias

- Si se especifica NOCACHE, la columna last\_number de la consulta muestra el siguiente número de secuencia disponible.
- Si se especifica CACHE, la columna last\_number muestra el siguiente número disponible de la secuencia que no se ha almacenado en caché en la memoria.

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

# Pseudocolumnas NEXTVAL y CURRVAL

- La pseudocolumna NEXTVAL se utiliza para extraer números de secuencia sucesivos de una secuencia especificada.
- Debe cualificar a NEXTVAL con el nombre de columna.
- Al hacer referencia a sequence.NEXTVAL, se genera un nuevo número de secuencia y el actual se sustituye en CURRVAL.

# Pseudocolumnas NEXTVAL y CURRVAL

- En el ejemplo siguiente se inserta un nuevo departamento en la tabla DEPARTMENTS.
- Utiliza la secuencia DEPARTMENTS\_SEQ para generar un nuevo número de departamento como se muestra a continuación:

```
INSERT INTO departments
      (department_id, department_name, location_id)
VALUES (departments_seq.NEXTVAL, 'Support', 2500);
```

# Pseudocolumnas NEXTVAL y CURRVAL

- Supongamos ahora que desea contratar empleados para el nuevo departamento.
- La sentencia INSERT que se debe ejecutar para los nuevos empleados puede incluir el siguiente código:

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

- Nota: en el ejemplo anterior, se asume que la secuencia denominada EMPLOYEE\_SEQ ya se ha creado para generar los números de los nuevos empleados.

# Pseudocolumnas NEXTVAL y CURRVAL

- La pseudocolumna CURRVAL del ejemplo siguiente se utiliza para hacer referencia al número de secuencia que acaba de generar el usuario actual.
- NEXTVAL se debe utilizar para generar un número de secuencia en la sesión del usuario actual antes de que se pueda hacer referencia a CURRVAL.
- Debe cualificar a CURRVAL con el nombre de secuencia.

# Pseudocolumnas NEXTVAL y CURRVAL

- Al hacer referencia a sequence.CURRVAL, se devuelve el último valor generado por el proceso de ese usuario.

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

# Uso de una Secuencia

- Después de crear una secuencia, se generan números secuenciales para utilizarlos en las tablas. Haga referencia a los valores de secuencia mediante las pseudocolumnas NEXTVAL y CURRVAL.
- Puede utilizar NEXTVAL y CURRVAL en los siguientes contextos:
  - La lista SELECT de una sentencia SELECT que no forme parte de una subconsulta
  - La lista SELECT de una subconsulta de la sentencia INSERT
  - La cláusula VALUES de la sentencia INSERT
  - La cláusula SET de la sentencia UPDATE



# Uso de una Secuencia

- No puede utilizar NEXTVAL y CURRVAL en los siguientes contextos:
  - La lista SELECT de una vista
  - Una sentencia SELECT con la palabra clave DISTINCT
  - Una sentencia SELECT con las cláusulas GROUP BY, HAVING o ORDER BY
  - Una subconsulta en una sentencia SELECT, DELETE o UPDATE
  - La expresión DEFAULT en una sentencia CREATE TABLE o ALTER TABLE

# Uso de una Secuencia

- Para continuar con nuestro ejemplo del Maratón de Londres, se ha creado una tabla para los corredores:

```
CREATE TABLE runners
(runner_id NUMBER(6,0) CONSTRAINT runners_id_pk PRIMARY KEY,
 first_name VARCHAR2(30),
 last_name VARCHAR2(30));
```

# Uso de una Secuencia

- A continuación, creamos la secuencia que generará valores para la columna de clave primaria runner\_id.

```
CREATE SEQUENCE runner_id_seq  
INCREMENT BY 1  
START WITH 1  
MAXVALUE 50000  
NOCACHE  
NOCYCLE;
```

# Uso de una Secuencia

- El uso de la siguiente sintaxis permitirá insertar nuevos participantes en la tabla de corredores.
- El número de identificación del corredor se generará mediante la recuperación del NEXTVAL de la secuencia.

```
INSERT INTO runners (runner_id, first_name, last_name)
VALUES
(runner_id_seq.NEXTVAL, 'Joanne', 'Everely');
```

```
INSERT INTO runners (runner_id, first_name, last_name)
VALUES
(runner_id_seq.NEXTVAL, 'Adam', 'Curtis');
```

# Uso de una Secuencia

- Para confirmar que la secuencia funciona correctamente, se consulta la tabla:

```
SELECT runner_id, first_name, last_name  
FROM runners;
```

RUNNER_ID	FIRST_NAME	LAST_NAME
1	Joanne	Everely
2	Adam	Curtis

# Uso de una Secuencia

- Para ver el valor actual de `runners_id_seq`, se utiliza `CURRVAL`.
- Observe el uso de la tabla `DUAL` en este ejemplo.
- Oracle Application Express no ejecutará esta consulta, pero debe comprender cómo funciona.

```
SELECT runner_id_seq.CURRVAL  
FROM dual;
```

# Uso de una Secuencia

- Las secuencias almacenadas en la caché de la memoria permiten un acceso más rápido a los valores de secuencia.
- La caché se rellena la primera vez que hace referencia a la secuencia.
- Las solicitudes del siguiente valor de secuencia se recuperan de la secuencia almacenada en caché.
- Después de utilizar el último valor de secuencia, la siguiente solicitud de la secuencia introduce otra caché de secuencias en la memoria.
- 20 es el número por defecto de los números de secuencia almacenados en caché.



# Números no Secuenciales

- Aunque los generadores de secuencias emiten números secuenciales sin intervalos, esta acción se realiza independientemente de que se realice una confirmación o un rollback en la base de datos.
- Los intervalos (números no secuenciales) se pueden generar:
  - Al realizar un rollback de una sentencia que contiene una secuencia, se pierde el número.
  - Un fallo del sistema. Si la secuencia almacena en la caché los valores en la memoria y el sistema falla, dichos valores se pierden.
  - El uso de la misma secuencia para varias tablas. Si lo hace así, cada tabla puede contener intervalos en los números secuenciales.



# Visualización del Siguiente Valor

- Si la secuencia se ha creado con NOCACHE, es posible ver el siguiente valor de secuencia disponible sin aumentarlo con la consulta de la tabla USER\_SEQUENCES.

```
SELECT sequence_name, min_value, max_value, last_number AS "Next number"  
FROM USER_SEQUENCES  
WHERE sequence_name = 'RUNNER_ID_SEQ';
```

SEQUENCE_NAME	MIN_VALUE	MAX_VALUE	Siguiente número
RUNNER_ID_SEQ	1	50000	3

# Modificación de una Secuencia

- Al igual que con otros objetos de base de datos que ha creado, SEQUENCE también se puede cambiar mediante la sentencia ALTER SEQUENCE.
- ¿Qué ocurre si el Maratón de Londres ha excedido los 50.000 registros de corredores y necesita agregar más números?
- La secuencia se puede cambiar para aumentar el valor MAXVALUE sin cambiar el orden de números existente.

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

# Modificación de una Secuencia

- Se realiza alguna validación al modificar una secuencia.
- Por ejemplo, no se puede ejecutar un nuevo valor MAXVALUE menor que el número de secuencia actual.

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

ERROR at line 1:

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

# Directrices de ALTER SEQUENCE

- Se aplican algunas directrices al ejecutar una sentencia ALTER SEQUENCE.
- Tienen las siguientes características:
  - Debe ser el propietario o tener el privilegio ALTER para la secuencia para poder modificarla.
  - Solo se ven afectados por la sentencia ALTER SEQUENCE los números de secuencia futuros.
  - La opción START WITH no se puede cambiar mediante ALTER SEQUENCE. La secuencia se debe borrar y volver a crear para reiniciar la secuencia en un número diferente.

# Eliminación de Secuencias

- Para eliminar una secuencia del diccionario de datos, utilice la sentencia DROP SEQUENCE.
- Debe ser el propietario de la secuencia o tener el privilegio DROP ANY SEQUENCE para eliminarla.
- Una vez eliminada, ya no se puede hacer referencia a la secuencia.

```
DROP SEQUENCE runner_id_seq;
```

# Terminología

Entre los términos clave utilizados en esta lección se incluyen:

- CACHE/ NOCACHE
- CREATE SEQUENCE
- CURRVAL
- CYCLE/ NOCYCLE
- INCREMENT BY
- MAXVALUE
- MINVALUE



# Terminología

Entre los términos clave utilizados en esta lección se incluyen:

- NEXTVAL
- NOMAXVALUE
- NOMINVALUE
- Secuencias
- START WITH

# Resumen

En esta lección, debe haber aprendido lo siguiente:

- Enumerar al menos tres características útiles de una secuencia
- Escribir y ejecutar una sentencia SQL que crea una secuencia
- Consultar el diccionario de datos mediante `USER_SEQUENCES` para confirmar una definición de secuencia
- Aplicar las reglas para utilizar `NEXTVAL` a fin de generar números únicos secuenciales en una tabla
- Enumerar las ventajas y desventajas del almacenamiento en caché de los valores de secuencia
- Nombrar tres motivos por los que se pueden producir intervalos en una secuencia





