¡Perfecto, Bryan! Te explico toda esta teoría en **español fácil y claro**, y te doy una **guía práctica** para que puedas usar CREATE SEQUENCE en tus ejercicios de Oracle paso a paso. 💪

**🧠 ¿Qué es una SEQUENCE en Oracle?**

Una **secuencia** (SEQUENCE) sirve para **generar números únicos y consecutivos** automáticamente, por ejemplo, para un campo ID o DEPTNO.

Es como una **fábrica de números**: cada vez que llamas a NEXTVAL, te da el siguiente número.

**📚 Sintaxis básica de CREATE SEQUENCE**

CREATE SEQUENCE nombre\_secuencia

INCREMENT BY 1 -- cuánto aumenta cada vez

START WITH 1 -- por qué número empieza

MAXVALUE 99999 -- valor máximo permitido

MINVALUE 1 -- valor mínimo permitido

NOCYCLE -- que no vuelva a empezar

CACHE 20 -- números que guarda en memoria

NOORDER; -- no garantiza orden exacto

**🧩 ¿Qué significa cada parte?**

| **Opción** | **¿Para qué sirve?** |
| --- | --- |
| INCREMENT BY | Número que se suma cada vez. Si es -1, cuenta hacia atrás. |
| START WITH | El primer valor que va a entregar. |
| MAXVALUE | El máximo que puede alcanzar. |
| MINVALUE | El mínimo que puede entregar. |
| CYCLE | Si lo activas, vuelve a empezar cuando llega al límite. |
| NOCYCLE | (por defecto) Detiene la secuencia cuando llega al máximo o mínimo. |
| CACHE | Guarda en memoria los siguientes valores (mejora el rendimiento). |
| ORDER | Asegura que los valores salgan en orden (para sistemas distribuidos, clústeres). |
| NOORDER | (por defecto) No asegura orden si hay mucha concurrencia. |

**🔧 Cómo usar la secuencia**

**1. Crear una secuencia**

CREATE SEQUENCE seq\_dept

INCREMENT BY 1

START WITH 50

MINVALUE 1

MAXVALUE 9999

NOCYCLE

CACHE 10;

**2. Usarla al insertar**

INSERT INTO dept (deptno, dname, loc)

VALUES (seq\_dept.NEXTVAL, 'MARKETING', 'SEVILLA');

🔹 seq\_dept.NEXTVAL te da el siguiente número.

**3. Ver el siguiente valor**

SELECT seq\_dept.NEXTVAL FROM dual;

**4. Ver el valor actual (solo después de haber usado NEXTVAL)**

SELECT seq\_dept.CURRVAL FROM dual;

**🧪 EJEMPLO COMPLETO**

**Crear tabla**

CREATE TABLE empleados (

id\_empleado NUMBER PRIMARY KEY,

nombre VARCHAR2(50)

);

**Crear secuencia**

CREATE SEQUENCE seq\_empleado

START WITH 1

INCREMENT BY 1

NOCYCLE

CACHE 5;

**Insertar datos con la secuencia**

INSERT INTO empleados (id\_empleado, nombre)

VALUES (seq\_empleado.NEXTVAL, 'Bryan');

INSERT INTO empleados (id\_empleado, nombre)

VALUES (seq\_empleado.NEXTVAL, 'Lucía');

**Consultar datos**

SELECT \* FROM empleados;

**🧭 GUÍA PARA TUS EJERCICIOS**

1. **Crea la tabla** que va a tener la columna con números automáticos.
2. **Crea la secuencia** con CREATE SEQUENCE.
3. En cada INSERT, usa nombre\_secuencia.NEXTVAL como valor para esa columna.
4. Usa SELECT nombre\_secuencia.NEXTVAL FROM dual; si solo quieres ver el número que sigue.

¿Te gustaría que hagamos juntos un ejercicio real con una tabla departamentos y una secuencia paso a paso?

¡Perfecto, Bryan! Vamos a complementar la explicación sobre las columnas IDENTITY en Oracle 12c, basándonos en la información de la página que mencionaste: citeturn0search0.

**🆕 ¿Qué es una columna IDENTITY en Oracle 12c?**

A partir de Oracle 12c, puedes definir una columna numérica que se autoincremente automáticamente, sin necesidad de crear una secuencia ni un trigger manualmente. Es similar al AUTO\_INCREMENT de MySQL o al IDENTITY de SQL Server.

**🧱 Sintaxis básica**

GENERATED [ ALWAYS | BY DEFAULT [ ON NULL ] ] AS IDENTITY

[ ( opciones\_de\_secuencia ) ]

**🔍 Tipos de generación**

| **Tipo** | **Comportamiento** |
| --- | --- |
| GENERATED ALWAYS | Oracle siempre genera el valor. Si intentas insertar un valor manualmente, se produce un error. |
| GENERATED BY DEFAULT | Oracle genera el valor solo si no proporcionas uno. Puedes insertar valores manualmente. |
| GENERATED BY DEFAULT ON NULL | Oracle genera el valor si insertas NULL o no proporcionas ningún valor. |

**⚙️ Opciones de secuencia**

Puedes personalizar el comportamiento de la generación automática:

* START WITH: Valor inicial.
* INCREMENT BY: Incremento entre valores.
* MAXVALUE / MINVALUE: Valores máximos y mínimos permitidos.
* CYCLE / NOCYCLE: Si se reinicia al alcanzar el límite.
* CACHE: Número de valores que Oracle prealmacena para mejorar el rendimiento.

**🧪 Ejemplo práctico**

CREATE TABLE departamentos (

deptno NUMBER GENERATED ALWAYS AS IDENTITY

(START WITH 10 INCREMENT BY 5) PRIMARY KEY,

dname VARCHAR2(50),

loc VARCHAR2(50)

);

En este ejemplo:

* deptno comienza en 10 y se incrementa en 5: 10, 15, 20, etc.
* Oracle crea automáticamente una secuencia interna para manejar estos valores.

Al insertar datos:

INSERT INTO departamentos (dname, loc)

VALUES ('INVESTIGACIÓN', 'MADRID');

Oracle asigna automáticamente el siguiente valor de deptno.

**🔄 Comparación: IDENTITY vs. SEQUENCE manual**

| **Característica** | **IDENTITY** | **SEQUENCE manual + NEXTVAL** |
| --- | --- | --- |
| ¿Necesita secuencia manual? | No | Sí |
| ¿Necesita trigger? | No | A veces |
| ¿Fácil de usar? | Sí | Requiere más configuración |
| ¿Control total? | Menos flexible | Más control sobre la secuencia |
| ¿Nombre de secuencia? | Automático (no visible) | Definido por el usuario |

**✅ ¿Cuándo usar IDENTITY?**

* Cuando deseas una columna numérica que se autoincremente automáticamente.
* Para claves primarias técnicas (surrogate keys).
* Cuando no necesitas un control detallado sobre la secuencia.

Si necesitas más ejemplos o tienes dudas específicas sobre cómo implementar columnas IDENTITY en tus tablas, ¡aquí estoy para ayudarte!