**Autoestudio 04**

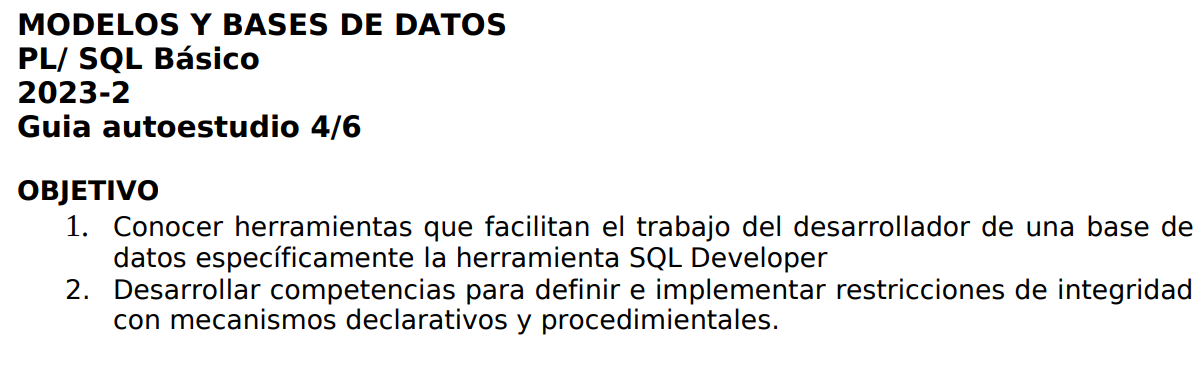
**Integrantes:**

**Jeimy Alejandra Yaya Martínez**

**Diego Alexander Cardenas Beltrán**

**Universidad Escuela Colombiana de Ingeniería Julio Garavito**  
**Modelos y bases de datos**  
 **María Irma Rozo**

**19/10/2023**



Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

### INVESTIGACIÓN

1. **Acciones referenciales**
   1. **¿Para qué sirven las acciones referenciales?**

Las acciones referenciales, a menudo se utilizan en el contexto de bases de datos relacionales, como Oracle, para garantizar la integridad referencial de los datos. Sirven para mantener la consistencia de la base de datos al establecer reglas que aseguran que las relaciones entre tablas se mantengan correctamente. Las acciones referenciales se utilizan para definir qué sucede cuando se intenta realizar una operación que podría violar la integridad referencial, como insertar, actualizar o eliminar registros en una tabla relacionada.

* 1. **¿Qué acciones soporta ORACLE? ¿Qué permite hacer cada una de ellas?**

Oracle Database admite acciones referenciales que se definen en la restricción de clave externa (FOREIGN KEY). Las acciones típicas que se pueden especificar en una restricción de clave externa en Oracle son:

* **NO ACTION:** No se permite realizar ninguna acción cuando se viola la integridad referencial. Se produce un error y se revierte la operación.
* **CASCADE:** Cuando se realiza una operación en la tabla principal (como eliminar un registro), las operaciones correspondientes se aplican a las tablas secundarias automáticamente.
* **SET NULL:** Se establecen los valores en NULL en las columnas de la tabla secundaria cuando se realiza una operación en la tabla principal que causa una violación de la integridad referencial.
* **SET DEFAULT:** Los valores en las columnas de la tabla secundaria se establecen en sus valores predeterminados cuando se realiza una operación en la tabla principal que causa una violación de la integridad referencial.

### PL/SQL

* 1. **¿Qué es PL/SQL?**

PL/SQL (Procedural Language/Structured Query Language) es un lenguaje de programación de bases de datos que extiende las capacidades del SQL estándar. Permite escribir procedimientos almacenados, funciones, disparadores y bloques anónimos que se ejecutan en una base de datos Oracle. PL/SQL combina SQL para la manipulación de datos y características de programación procedural como bucles, condicionales y manejo de excepciones.

* 1. **¿Qué motores lo soportan?**

PL/SQL está diseñado específicamente para Oracle Database y es compatible con este sistema de gestión de bases de datos. Otros motores de bases de datos pueden tener sus propios lenguajes de programación procedural, pero PL/SQL es exclusivo de Oracle.

### Datos e instrucciones en PL/SQL

* 1. **¿Cuáles son los tipos de datos que ofrece?**

PL/SQL ofrece varios tipos de datos, incluyendo tipos escalares (como VARCHAR2, NUMBER), tipos compuestos (como registros y tablas) y tipos de referencia (como cursores y objetos). Estos tipos se utilizan para declarar variables y parámetros en programas PL/SQL.

* 1. **¿Cuál es la forma de definir constantes y variables?**

Para definir constantes en PL/SQL, se utiliza la palabra clave CONSTANT. Por ejemplo:

**CONSTANT pi CONSTANT NUMBER := 3.14159;**

Para definir variables, se utiliza la palabra clave VARIABLE o DECLARE. Por ejemplo:

**VARIABLE nombre VARCHAR2(50);**

**DECLARE**

**edad NUMBER := 30;**

**END;**

* 1. **¿Cómo se define una variable con un tipo tomado de la base de datos?**

Para definir una variable en PL/SQL con un tipo tomado de la base de datos utilizando %TYPE. Por ejemplo:

**nombre\_empleado employees.first\_name%TYPE;**

* 1. **¿Cuál es la forma de los diferentes tipos de asignación? (Son tres)**
* Asignación simple (:=): Se utiliza para asignar un valor a una variable. Por ejemplo:

**salario := 5000;**

* Asignación SELECT INTO: Se utiliza para asignar valores de una consulta SELECT a una o más variables. Por ejemplo:

**SELECT first\_name, last\_name INTO nombre, apellido**

**FROM employees**

**WHERE employee\_id = 100;**

* Asignación PL/SQL RECORD: Se utiliza para asignar valores de un registro PL/SQL a otro. Por ejemplo:

**registro1 := registro2;**

### Cursores

* 1. **¿Qué es un cursor implícito? ¿Para qué sirve?**

Un cursor implícito es un cursor que se crea automáticamente cuando se ejecuta una sentencia SQL en un bloque PL/SQL sin necesidad de declarar explícitamente un cursor. Se utiliza para recuperar filas de un conjunto de resultados de una consulta. Los cursores implícitos son útiles cuando solo se espera un conjunto de resultados simple.

* 1. **¿Qué es un cursor explícito? ¿Para qué sirve?**

Un cursor explícito es un cursor que se declara y se utiliza de manera explícita en un bloque PL/SQL. Se utiliza para realizar operaciones más avanzadas, como recorrer filas en un conjunto de resultados, realizar consultas parametrizadas o realizar múltiples operaciones de base de datos en un procedimiento o función.

* 1. **¿Cuáles son las excepciones propias de uso de estos cursores?**

NO\_DATA\_FOUND: Se lanza cuando un cursor no encuentra filas en el conjunto de resultados.

TOO\_MANY\_ROWS: Se lanza cuando un cursor encuentra más de una fila en el conjunto de resultados cuando solo se esperaba una.

CURSOR\_ALREADY\_OPEN: Se lanza cuando se intenta abrir un cursor que ya está abierto.

CURSOR\_NOT\_FOUND: Se lanza cuando se intenta realizar una operación en un cursor que no se ha abierto previamente.

### Modularidad

* 1. **¿Cuál es la estructura general de un bloque PL/SQL?**

Un bloque PL/SQL consta de tres secciones principales: declaración, ejecución y excepción. La estructura general es la siguiente:

**DECLARE**

**-- Declaraciones de variables, cursores, constantes, etc.**

**BEGIN**

**-- Sentencias de ejecución**

**EXCEPTION**

**-- Manejo de excepciones**

**END;**

* 1. **¿Para qué sirven las diferentes estructuras modulares? (bloque anónimo, procedimiento, función y disparador)**
* **Bloque anónimo**: Se utiliza para escribir secuencias de comandos PL/SQL independientes que se ejecutan una vez. Pueden incluir declaraciones, sentencias SQL y manejo de excepciones.
* **Procedimiento:** Es un bloque PL/SQL con un nombre que se almacena en la base de datos y se puede invocar de forma repetida. Se utiliza para encapsular lógica de negocio y realizar tareas específicas.
* **Función:** Similar a un procedimiento, pero devuelve un valor. Puede ser invocada en expresiones SQL y en otras partes del código PL/SQL.
* **Disparador (Trigger):** Es un bloque PL/SQL que se ejecuta automáticamente en respuesta a un evento específico en la base de datos, como una inserción, actualización o eliminación de datos. Se utiliza para aplicar lógica antes o después de que ocurra el evento.

**PRACTICANDO.**

**En este auto-estudio vamos a implementar un caso de uso de la base de datos neeps.**

Diagrama

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Texto

Descripción generada automáticamente con confianza media

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente