**PL/ SQL Básico**

**Guia autoestudio 4/6**

## INVESTIGACIÓN

### Acciones referenciales

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

Son las acciones que va a tomar la base de datos cuando un registro que está siendo referenciado en otra tabla sea actualizado o eliminado en registros que están relacionados entre sí a través de claves foráneas y primarias. Su principal función es garantizar que las relaciones entre tablas permanezcan coherentes y que no existan referencias inválidas o huérfanas

* 1. ¿Qué acciones soporta ORACLE? ¿Qué permite hacer cada una de ellas?
* CASCADE: Borra los registros de la tabla dependiente cuando se borra el registro de la tabla principal (en una sentencia DELETE), o actualiza el valor de la clave secundaria cuando se actualiza el valor de la clave referenciada (en una sentencia UPDATE)
* SET NULL: Establece a NULL el valor de la clave secundaria cuando se elimina el registro en la tabla principal o se modifica el valor del campo referenciado.
* SET DEFAULT: Esta acción establece que, al eliminar o actualizar una fila en la tabla padre, las columnas de clave foránea en la tabla hija se establecerán a un valor predeterminado previamente definido.
* NO ACTION: Especifica que no se realizará ninguna acción automática en la tabla hija cuando se elimine o actualice una fila en la tabla padre. Si existen filas en la tabla hija que referencian la fila modificada en la tabla padre, la operación en la tabla padre será rechazada y se generará un error, manteniendo así la integridad referencial.
* RESTRICT: Es el comportamiento por defecto, que impide realizar modificaciones que atentan contra la integridad referencial.

## PL/SQL

* 1. ¿Qué es PL/SQL?

Procedural Language/Structured Query Language Es un lenguaje de procedimiento diseñado específicamente para abarcar sentencias SQL dentro de su sintaxis con estructuras de programación tradicionales

* 1. ¿Qué motores lo soportan?
* Oracle Database

### Datos e instrucciones en PL/SQL

* 1. ¿Cuáles son los tipos de datos que ofrece?
     + BINARY DOUBLE, BINARY FLOAT
     + CHAR, NCHAR
     + CLOB, NCLOB
     + DATE
     + FLOAT
     + NUMBER, NUMBER (p,s),
     + NVARCHAR2, VARCHAR2
     + ROWID
     + TIMESTAMP, TIMESTAMP WITH LOCAL TIMEZONE, TIMESTAMP WITH TIMEZONE
  2. ¿Cuál es la forma de definir constantes y variables?
* Variable

nombre tipo [NOT NULL] [:= valor];

* := y DEFAULT son equivalentes.
* Si escribe NOT NULL es obligatorio inicializar la variable
* Constante

nombre CONSTANT tipo := valor;

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

Se utiliza el atributo %TYPE

nombre tabla.columna%TYPE;

* 1. ¿Cuál es la forma de los diferentes tipos de asignación? (Son tres)
* simple o directa:

nombre := valor;

* mediante consulta SELECT INTO
* por defecto en la declaración

### Cursores

### Un cursor son estructuras que permiten manejar y controlar el conjunto de registros que resultan de una consulta SQL, es el nombre para un área de memoria privada que contiene información procedente de la ejecución de una sentencia SELECT. Cada cursor tiene atributos que nos devuelven información útil sobre el estado del cursor en la ejecución de la sentencia SQL. Cuando un cursor está abierto y los datos referenciados por la consulta SELECT cambian, estos cambios no son recogidos por el cursor.

* 1. ¿Qué es un cursor implícito?

Este tipo de cursores se utiliza para operaciones SELECT INTO. Se usan cuando la consulta devuelve un único registro. Los cursores implícitos son útiles cuando solo se espera un conjunto de resultados simple.

¿Para qué sirve?

* Facilita la ejecución de consultas sin necesidad de definir manualmente un cursor.
* Se usa cuando la consulta devuelve una sola fila.
* Se accede a información sobre el cursor usando variables del sistema (SQL%FOUND, SQL%NOTFOUND, SQL%ROWCOUNT, SQL%ISOPEN).
  1. ¿Qué es un cursor explícito?

son declarados y controlados por el programador. Se utilizan cuando la consulta devuelve un conjunto de registros. Ocasionalmente también se utilizan en consultas que devuelven un único registro por razones de eficiencia. Son más rápidos.

¿Para qué sirve?

* Permite procesar varias filas de una consulta de manera controlada.
* Se usa cuando se necesita recorrer un conjunto de registros uno por uno.
* Ofrece mayor control sobre la ejecución y procesamiento de los datos.
  1. ¿Cuáles son las excepciones propias de uso de estos cursores?

Algunas de estas excepciones son:

ACCESS\_INTO\_NULL: Se dispara cuando se intenta acceder a los atributos de un objeto no inicializado.

CURSOR\_ALREADY\_OPEN: Se dispara cuando intentamos abrir un cursor que ya se encuentra abierto.

ROWTYPE\_MISMATCH: Se dispara cuando la variable del cursor del host y la variable del cursor PL/SQL pertenecen a tipos incompatibles.

SUBSCRIPT\_OUTSIDE\_LIMIT: Se dispara cuando se intenta acceder a una tabla anidada o un array con un valor del índice ilegal.

STORAGE\_ERROR: Se dispara cuando el bloque PL/SQL se ejecuta fuera de memoria.

### Modularidad

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

La estructura general de un bloque PL/SQL consta de tres partes: la sección declarativa, la sección de ejecución y la sección de excepciones. La cual se representa con la siguiente sintaxis:

[DECLARE Declaración de variables]

/*Parte declarativa para constantes y variables a usar*/

BEGIN

Sentencias SQL y PL/SQL

/*Parte de ejecución con las sentencias ejecutables*/

[EXCEPTION Manejadores de excepciones]

/*Parte de excepciones para el manejo de errores*/

END;

* 1. ¿Para qué sirven las diferentes estructuras modulares? (bloque anónimo, procedimiento, función y disparador)

Un bloque anónimo es Código PL/SQL sin nombre que se ejecuta inmediatamente, usualmente se usa para pruebas rápidas y ejecución única de scripts

Los procedimientos almacenados es un bloque de código reutilizable que puede recibir parámetros, pero no siempre devuelve un valor, su uso se suele encontrar en la automatización de tareas repetitivas y actualización de datos en masa.

Las funciones son similares a los procedimientos, pero siempre devuelven un valor y se suelen usar en cálculos y transformaciones de datos dentro de consultas SQL.

Los disparadores o también llamados triggers se ejecutan automáticamente en respuesta a eventos en la base de datos (INSERT, UPDATE) y se usan para auditoría, validación de datos y sincronización de tablas.

## PRACTICANDO.

En este auto-estudio vamos a implementar un caso de uso simplificado de la base de datos [**Adventure Works**](https://www.sqlzoo.net/wiki/AdventureWorks)

|  |  |
| --- | --- |
|  | **T10Natural** Integer(10) Positivo **TPMName**  String(10) Mínimo dos palabras separadas por espacio o por guion medio (-)  **TCulture**  String(6)  [en, ar, fr, th, he, zh-cht, co] |
|  | **Ad**  El número de identificación del modelo se genera automáticamente: es un consecutivo  El modelo se crea con una descripción en inglés que corresponde a la descripción del catálogo o al nombre, si no se especifica descripción.  **Mo**  No se pueden modificar ninguno de los datos dados.  Es posible adicionar y eliminar descripciones para diferentes culturas (no se pueden repetir las culturas).  **El**  Se puede eliminar si no tiene productos asociados.  **Co**  Consulta todos los datos del modelo (identificador, nombre, descripción de catalogo) y la descripción del producto para cada una de las culturas.  NOTA:  El número de identificación de la descripción es un número aleatorio generado automáticamente. |

**Consultar la guía de diseño que se presenta al final del auto-estudio.**

**Consultar los** [**REQUISITOS DE ENTREGA**](https://campusvirtual.escuelaing.edu.co/moodle/mod/wiki/view.php?id=59242)en la sección Proyecto del Curso correspondiente a: **Restricciones Declarativas y Restricciones Declarativas, Procedimentales y Automatización.**

# Adicionando restricciones declarativas.

Para proteger la base de datos vamos a implementar las restricciones que se pueden garantizar usando restricciones declarativas.

* 1. Cree la nueva tabla. (Tablas,XTablas)
  2. Definir las restricciones declarativas

(Atributos, Tuplas)

* 1. Validar con casos significativos la protección de la base de datos, adicione comentarios que expliquen la condición que están validando.

(AtributosOK, AtributosNoOK, TuplasOK, TuplasNoOK)

# Adicionando acciones de referencia

Para proteger la base de datos vamos a implementar las acciones de referencia necesarias para el caso de uso.

* 1. Definir las acciones de referencia

(Deben eliminar y volver a crear las FK afectadas.)

(Acciones)

* 1. Validar con casos significativos las condiciones definidas

(AccionesOK)

# Adicionando disparadores

Para preparar los escenarios asociados a los diferentes casos de uso vamos a implementar los disparadores (CRUD).

* 1. Escriba las instrucciones necesarias para crear y eliminar los disparadores definidos.

( Disparadores – XDisparadores )

* 1. Escriba 3 instrucciones que permitan probar la actualización de la base de datos haciendo uso de cada disparador, cuando sea pertinente.

( DisparadoresOK )

* 1. Escriban 3 instrucciones para validar que protege la base de datos impidiendo cambios, cuando sea pertinente. (DisparadoresNoOK

**Diseño**

En este diseño se presenta el mecanismo recomendado para implementar cada una de las condiciones o reglas.

|  |
| --- |
| **T10Natural** Integer(10) Positivo **CK\_PMODEL\_ID CK\_PDESCRIPTION\_ID**  **TPMName**  String(10) Mínimo dos palabras separadas por espacio o por guion medio (-)  **CK\_PMODEL\_NAME**  **TCulture**  String(6)  [en, ar, fr, th, he, zh-cht, co]  **CK\_PMODELPRODUCT\_CULTURE** |
| **CLAVES**  **PK\_PMOEL, UK\_PMOEL\_NAME FK\_PMODELPRODUCT\_PMODEL, FK\_PMODELPRODUCT\_PDESCRIPTION** |
|  |
| **Ad**  El número de identificación del modelo se genera automáticamente: es un consecutivo  **TR\_PMODEL\_BI (disparador.automatizar)**  El modelo se crea con una descripción en inglés que corresponde a la descripción del catálogo o al nombre, si no se especifica descripción.  **TR\_PMODEL\_AI (disparador.automatizar) Mo**  No se pueden modificar ninguno de los datos dados. **TR\_PMODEL\_BU (disparador.restringir) TR\_PMODELPRODUCT\_BU (disparador.restringir) TR\_PDESCRIPTION\_BU (disparador.restringir)**  Es posible adicionar y eliminar descripciones para diferentes culturas (no se pueden repetir las culturas).  **UK\_PMODELPRODUCT\_ID\_CULTURE**  **El**  Se puede eliminar si no tiene productos asociados.  **FK\_PMODELPRODUCT\_PMODEL (accion.automatizar) Co**  Consulta todos los datos del modelo (identificador, nombre, descripción de catalogo) y la descripción del producto para cada una de las culturas.  NOTA:  El número de identificación de la descripción es un número aleatorio generado automáticamente.  **TR\_PMODELPRODUCT\_BI (disparador.automatizar)** |