# INVESTIGACIÓN A. Acciones referenciales

1. ¿Para qué sirven las acciones referenciales?

**R/:** Sirven principalmente para controlar los datos que se pueden almacenar en una tabla con al menos una FK, además de controlar los cambios que se realizan en los PK de la tabla a la que se referencia.

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

**R/:**

* **NO ACTION:** Genera un error y revierte la acción de actualizar o eliminar datos de una tabla
* **CASCADE:** Si se actualiza o elimina una fila, todas las filas correspondientes que hagan referencia a ella también se verán actualizadas o eliminadas según el caso.
* **SET NULL:** Todas aquellas FK que estén presentes en la fila serán establecidos a un valor NULL
* **SET DEFAULT:** Todas aquellas FK que estén presentes en la fila serán establecidos a un valor predeterminado que debe de ser definido momentos antes en la base de datos.

# B. PL/SQL

1. ¿Qué es PL/SQL?

**R/:** Procedural Language/Structured Query Language es un lenguaje de procesamiento procedimental implementado por Oracle que dispone de estructuras de programación similares a las de la mayoría de lenguajes de programación (manejo de variables, estructuras modulares, de control de flujo y toma de decisiones).

1. ¿Qué motores lo soportan?

**R/:** Lo soportan SQL developer y todas aquellas herramientas desarrolladas por Oracle.

# C. Datos e instrucciones en PL/SQL

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

**R/:**

* **decimal( p,e )**
* **float**
* **integer**
* **numeric( p,e )**
* **char( tamaño )**
* **varchar2( tamaño )**
* **long**
* **raw**
* **timestamp**
* **interval**
* **urowid**
* **boolean**

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

**R/:**

**Variables:**

Nombre\_variable tipo [**NOT** **NULL**] [:= valor | **DEFAULT** valor]

En donde := y DEFAULT son lo mismo, por lo que solo es necesario colocar una de las dos.

Ex:

num\_dep number(2) **NOT** **NULL** :=20

nom\_emple varchar2(15) **default** ‘Pedro’

**Constantes:**

nombre\_constante CONSTANT tipo\_de\_dato := valor

Ex:

Imp\_iva constant number(3,1) := 19,5

1. ¿Cómo se define una variable con un tipo tomado de la base de datos?
2. ¿Cuál es la forma de los diferentes tipos de asignación? (Son tres)

**R/:**

* :=
* DEFAULT

# D. Cursores

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

Se utilizan para realizar consultas SELECT que devuelven un único registro.

Se debe tener en cuenta que:

* Con cada cursos implícito debe existir la palabra clave INTO
* Las variables que reciben los datos devueltos por el cursor tienen que contener el mismo tipo de dato que las columnas de la tabla.
* Los cursores implícitos solo pueden devolver una únic fila. En caso de que se devuelva más de una fila o ninguna fila, se producirá una excepción.

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

Se emplean para realizar consultas SELECT que pueden

devolver cero filas o más de una fila. Para trabajar con cursores explícitos es necesario realizar las

siguientes tareas:

* Declarar el cursor
* Declarar los posibles parámetros que requiera el cursor
* Abrir el cursor con la instrucción OPEN
* Leer los datos del cursor con la instrucción FETCH
* Cerrar el cursor y liberar los recursos con las instrucciones CLOSE

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

* INVALID\_CURSOR: El programa intentó efectuar una operación no válida sobre un cursor 1001.
* NO\_DATA\_FOUND: Una sentencia SELECT INTO no devolvió valores o el programa referenció un elemento no inicializado en una tabla indexada 100.
* TO\_MANY\_ROWS:Una sentencia SELECT INTO devuelve más de una fila 1422.
* PROGRAM\_ERROR PL/SQL: Tiene un problema interno 6501.

# E. Modularidad

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

* Declaraciones: Es una zona opcional donde se declaran los objetos locales, ya sean variables, constantes, cursores, excepciones definidas por el usuario, entre otros.
* Instrucciones: Es una zona obligatoria donde se especifica el cuerpo del subprograma
* Tratamiento de excepciones: Es una zona opcional donde son agregadas todas las excepciones del programa el signo ‘/’ es utilizado para verificar la sintaxis del bloque antes de ser ejecutado

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

* BLOQUES ANÓNIMOS Se construyen de forma dinámica y se ejecutan una sola vez tras ser escritos. No se almacenan en la base de datos.
* PROCEDIMIENTOS: Se almacenan en el DD y son invocados. Pueden recibir y devolver múltiples parámetros.
* FUNCIONES: Se almacenan en el DD y son invocadas. Pueden recibir parámetros y devuelven un valor.
* DISPARADORES: Se almacenan en el DD y se ejecutan automáticamente cuando ocurre algún evento.

# PRACTICANDO. Musicians

En este auto-estudio vamos a implementar dos casos de uso.

|  |
| --- |
| **CASOS DE USO**  **Mantener músico(Ad,Mo,Co,El)**  **Ad**  El dato mínimo para adicionar un músico es su nombre. El nombre debe tener mínimo dos palabras: primer nombre y primer apellido. El número se genera automáticamente de forma consecutiva.  La fecha de nacimiento debe ser menor de la fecha de muerte.  La ciudad en la que vive por omisión es la ciudad de nacimiento.  Los tipos de música puede tomar los siguientes valores ser: jazz, classical, rock, blues o not known. Los compositores sólo pueden componer un tipo de música y los interpretes pueden tocar máximo tres instrumentos.  Si no se indica el tipo de música de un compositor, se supone que el tipo de música que  sabe interpretar (si solamente es uno)  **El**  Los músicos siempre se pueden eliminar, si no tienen composiciones o no figuran como organizadores de conciertos.  **Mo**  No es posible modificar ninguno de los datos que ya tienen valor salvo el sitio donde vive el músico y la fecha de muerte. |

A. Adicionando restricciones declarativas.

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

1. Definir las restricciones declarativas **Hecho**

(Atributos, Tuplas) (Restricciones.sql, Tuplas.sql)

1. Validar con casos significativos la protección de la base de datos, adicione comentarios. **Hecho**

(AtributosOK, AtributosNoOK, TuplasOK, TuplasNoOK)

# B. Adicionando acciones de referencia

Para proteger la base de datos vamos a implementar las acciones de referencia necesarias para tareas pedidas. 1. Definir las acciones de referencia

(Deben eliminar y volver a crear las FK afectadas) **Hecho**

(Acciones)

2. Validar con casos significativos las condiciones definidas **Hecho**

(AccionesOK)

# C. Adicionando disparadores

Para preparar las acciones asociadas a los diferentes casos de uso vamos a implementar algunos disparadores (CRUD).

Use los siguientes nombres para los disparadores que necesite.

Disparadores : Ad\_MUSICIAN, Mo\_MUSICIAN, El\_MUSICIAN

Ad\_COMPOSER,Mo\_COMPOSER,El\_COMPOSER

Ad\_PERFORMER, Mo\_PERFORMER, El\_PERFORMER Para cada uno de ellos:

1. Escriba las instrucciones necesarias para crear y eliminar el disparador

( Disparadores – Xdisparadores ) **HECHO**

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

( DisparadoresOK ) **HECHO Revisar**

1. Escriban 3 instrucciones por disparador para validar que protege la base dedatos impidiendo que se actualice, cuando sea pertinente. **HECHO**

(DisparadoresNoOK )

|  |  |
| --- | --- |
| **COMPONENTE** | **CONTENIDO** |
| Tuplas | Implementación de las estricciones declarativas de más de un atributo |
| TuplasOK | Pruebas de las restricciones en que la actualización es aceptada |
| TuplasNoOK | Pruebas de las restricciones en las que la actualización es rechazada |
| Acciones | Implementación de las acciones de referencia  Incluya las instrucciones para eliminar la FK anterior e incluir la nueva definición |
| AccionesOK | Pruebas para verificar el comportamiento de las acciones de referencia |
| Disparadores | Implementación de los disparadores diseñados |
| DisparadoresOK | Pruebas de los disparadores en los que las actualizaciones son aceptadas |
| DisparadoresNoOK | Pruebas de los disparadores en los que las actualizaciones son rechazadas |
| XDisparadores | Eliminación de disparadores |

Inventario

|  |  |
| --- | --- |
| Tablas  Atributos  Primarias  Unicas  Foraneas  XTablas | PoblarOK  PoblarNoOK  XPoblar |
| Tuplas  Acciones  Disparadores  XDisparadores | TuplasOK  TuplasNoOK  AccionesOK  DisparadoresOK  DisparadoresNoOK |