

MODELOS Y BASES DE DATOS
PL/ SQL Básico
2024-2
Guia autoestudio 4/6

OBJETIVO

1. Conocer herramientas que facilitan el trabajo del desarrollador de una base de datos específicamente la herramienta SQL Developer
2. Desarrollar competencias para definir e implementar restricciones de integridad con mecanismos declarativos y procedimentales.

TÓPICOS OBJETIVO 2

1. Acciones referenciales
2. Disparadores
3. Constantes y variables
4. Instrucciones básicas: asignación
5. Cursores: implícitos y explícitos

ENTREGA

Publicar las respuestas en el espacio correspondiente en un archivo **.zip** , el nombre de este archivo debe ser la concatenación en orden alfabético de los primeros apellidos de cada uno de los miembros.

INVESTIGACIÓN

A. Acciones referenciales

1. ¿Para qué sirven las acciones referenciales?
2. ¿Qué acciones soporta ORACLE? ¿Qué permite hacer cada una de ellas?

B. PL/SQL

1. ¿Qué es PL/SQL?
2. ¿Qué motores lo soportan?

C. Datos e instrucciones en PL/SQL

1. ¿Cuáles son los tipos de datos que ofrece?
2. ¿Cuál es la forma de definir constantes y variables?
3. ¿Cómo se define una variable con un tipo tomado de la base de datos?
4. ¿Cuál es la forma de los diferentes tipos de asignación? (Son tres)

D. Cursores

1. ¿Qué es un cursor implícito? ¿Para qué sirve?
2. ¿Qué es un cursor explícito? ¿Para qué sirve?
3. ¿Cuáles son las excepciones propias de uso de estos cursores?

E. Modularidad

1. ¿Cuál es la estructura general de un bloque PL/SQL?
2. ¿Para qué sirven las diferentes estructuras modulares? (bloque anónimo, procedimiento, función y disparador)

PRACTICANDO.

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

<pre> classDiagram class Band { <</>> band_no: TConsecutive <<U>> band_name: TName band_type: TType b_date: Date[0..1] } class Place class Musician Band "1" -- "*" Place : band_home Band "1" -- "*" Musician : band_contact Band "*" -- "*" Musician : Plays in </pre>	<p>TConsecutive Entero(9) Positivo consecutivo</p> <p>TName Cadena(20) Mínimo una palabra</p> <p>TType Cadeba(10) [rock, classical, jazz, blues, pop, soul]</p>
<pre> useCaseDiagram actor BandContact as Band Contact usecase MantainBand as Mantain band BandContact -- MantainBand </pre>	<p>Ad El número se genera automáticamente: es un consecutivo El nombre de la banda debe tener mínimo una palabra Si no se indica el tipo de banda, se asume que es 'rock' La fecha debe ser menor a la del día de hoy, si se indica.</p> <p>Mo De los datos generales, únicamente se puede modificar la fecha, si no se había dado. Es posible adicionar y eliminar músicos, no modificarlos. Máximo pueden existir 10 músicos en una banda</p> <p>El Siempre se pueden eliminar</p> <p>Co Consulta todos los datos de la banda (número, nombre, tipo, fecha, casa, y nombre del contacto y de cada uno de los músicos que tocan en ella)</p>

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

Consultar los **REQUISITOS DE ENTREGA** en la sección Proyecto del Curso correspondiente a: **Restricciones Declarativas** y **Restricciones Declarativas, Procedimentales y Automatización.**

A. 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)

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

(AtributosOK, AtributosNoOK, TuplasOK, TuplasNoOK)

B. 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. Justificar la decisión)

(Acciones)

2. Validar con casos significativos las condiciones definidas

(AccionesOK)

C. Adicionando disparadores

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

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

(Disparadores – XDisparadores)

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

(DisparadoresOK)

3. 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.

TIPOS

TConsecutive

Entero(9) Positivo consecutivo

[CK_BAND_BAND_NO](#)

TName

Cadena(20) Mínimo una palabra

[CK_BAND_BAND_NAME](#)

TType

Cadeba(10)

[rock, classical, jazz, blues, pop, soul]

[CK_BAND_BAND_TYPE](#)

CLAVES

[PK_BAND](#), [UK_BAND_BAND_NAME](#)

[FK_BAND_PLACE](#), [FK_BAND_MUSICIAN](#)

[FK_PLAYS_IN_MUSICIAN](#), [FK_PLAYS_IN_BAND](#)

Mantain band(ad, mo, el, co)

Ad

El número se genera automáticamente: es un consecutivo

[TR_BAND_BI](#) (disparador.automatizar)

El nombre de la banda debe tener mínimo una palabra

[TR_BAND_BI](#) (disparador.restringir)

Si no se indica el tipo de banda, se asume que es 'rock'

[TR_BAND_BI](#) (disparador.automatizar)

La fecha debe ser menor a la del día de hoy, si se indica.

[TR_BAND_BI](#) (disparador.restringir)

Mo

De los datos generales, únicamente se puede modificar la fecha, si no se había dado.

[TR_BAND_BU](#) (disparador.restringir)

Es posible adicionar y eliminar músicos, no modificarlos.

[TR_PLAYS_IN_BU](#) (disparador.restringir)

Máximo pueden existir 10 músicos en una banda

[TR_PLAYS_IN_BI](#) (disparador.restringir)

El

Siempre se pueden eliminar

[FK_PLAYS_IN_BAND](#) (accion.automatizar)

Co

Consulta todos los datos de la banda (número, nombre, tipo, fecha, casa, y nombre del contacto y de cada uno de los músicos que tocan en ella)