

MODELOS Y SERVICIOS DE DATOS [MYSD-MBDA]
PL/ SQL Básico
2025-2
Guía 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 simplificado de la base de datos [help Desk](#)

<pre>classDiagram class Staff { staff_code: TCode first_name: TName last_name: TName } class Level { level_code: Integer(1) manager: TTrue[0..1] operator: TTrue[0..1] engineer: TTrue[0..1] } Staff "1" --> "*" Level : has_level</pre>	<p>TCode String(6) Compuesto por la primera letra del nombre más la segunda letra del apellido más un número positivo</p> <p>TName String(50) Máximo dos palabras. Únicamente letras mayúsculas.</p> <p>TTrue String(1) 'Y'</p>
<pre>graph LR Actor[Chief of staff] --- UseCase((Maintain Staff))</pre>	<p>Ad El código del empleado se genera automáticamente. El número del código del empleado es generado secuencialmente dentro de las iniciales. Por ejemplo: AB1, AB2, AE1. Si no se da el nivel, se asume el nivel 1. El nombre y el apellido debe quedar en mayúsculas. El nombre debe ser diferente al apellido.</p> <p>Mo El único dato que se puede modificar es el nivel. Únicamente puede aumentar.</p> <p>El Si se elimina un nivel, se eliminan todos los empleados de ese nivel.</p> <p>Co Consulta todos los datos del empleado: código, nombre, apellido y nivel.</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. Crear 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.)

(Acciones)

2. Validar con casos significativos las condiciones definidas

(AccionesOK)

C. Adicionando disparadores

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

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

(Disparadores - XDisparadores)

2. Escribir 3 instrucciones que permitan probar la actualización de la base de datos haciendo uso de los disparadores.

(DisparadoresOK)

3. Escribir 3 instrucciones para probar los disparadores protegen la base de datos impidiendo cambios no permitidos.

(DisparadoresNoOK)

Diseño

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

TCode

String(6) Compuesto por la primera letra del nombre más la segunda letra del apellido más un número positivo

CK_STAFF_CODE

TName

String(50)

Máximo dos palabras. Únicamente letras mayúsculas.

CK_STAFF_FIRST_NAME

CK_STAFF_LAST_NAME

TTrue

String(1)

'Y'

CK_STAFF_TRUE

CLAVES

PK_STAFF, PK_LEVEL

FK_STAFF_LEVEL

UK_LEVEL_MOE (No pueden existir dos niveles iguales con diferente código)

Ad

El código del empleado se genera automáticamente. El número del código del empleado es generado secuencialmente dentro de las iniciales. Por ejemplo: AB1, AB2, AE1.

TR_STAFF_BI (disparador.automatizar)

Si no se da el nivel, se asume el nivel 1.

TR_STAFF_BI (disparador.automatizar)

El nombre y el apellido debe quedar en mayúsculas.

TR_STAFF_BI (disparador.automatizar)

El nombre debe ser diferente al apellido.

CK_STAFF_FIRST_LAST

Mo

El único dato que se puede modificar es el nivel. Únicamente puede aumentar.

TR_STAFF_BU (disparador.restringir)

El

Si se elimina un nivel, se eliminan todos los empleados de ese nivel.

FK_STAFF_LEVEL (accion.automatizar)

Co

Consulta todos los datos del empleado: código, nombre, apellido y nivel.

NOTA:

El número del código del empleado es generado secuencialmente dentro de las iniciales. Por ejemplo: AB1, AB2, AE1