

**MODELOS Y BASES DE DATOS**  
**PL/ SQL Básico**  
**2023-2**  
**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**

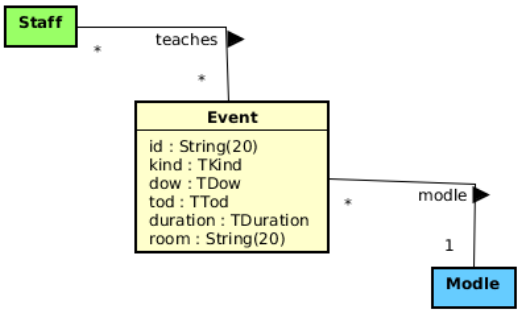
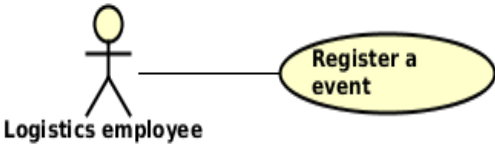
1. ¿Qué es un cursor implícito? ¿Para qué sirve?
2. ¿Qué es un cursor explícito? ¿Para qué sirve?
3. ¿Cuáles son las excepciones propias de uso de estos cursores?

**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 **neeps**.

	<p><b>TKind</b> String(1) 'L' o 'T'</p> <p><b>TDow</b> String(15) Dias de laborales en inglés</p> <p><b>TTod</b> String(5) De 8:00 a 20:00</p> <p><b>TDuration</b> Integer(1) Únicamente 1 o 2.</p> <p>No vamos a considerar occurs</p>
	<p><b>Register an Event (Ad, Mo, Co, El)</b></p> <p><b>Ad</b> El id se genera automáticamente: id del módulo, un punto, el tipo del evento y dos dígitos aleatorio. Si está programado a las 20:00 la duración debe ser una hora.</p> <p><b>Mo</b> El único dato a modificar es el salón. Se pueden adicionar profesores; pero no modificar ni eliminar.</p> <p><b>Co</b> Consulta todos los datos del evento, incluyendo sus profesores.</p> <p><b>El</b> Siempre se puede eliminar</p>

**Consultar las guías 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. Si es necesario, modifique los tipos base.

1. Modificar los tipos base, si es necesario.

(Tablas)

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.

### **TKind**

String(1) 'L' o 'T'

[CK\\_EVENT\\_KIND](#)

### **TDow**

String(15) Dias de laborales en inglés

[CK\\_EVENT\\_DOW](#)

### **TTod**

String(5) De 8:00 a 20:00

[CK\\_EVENT\\_TOD](#)

### **TDuration**

Integer(1) Únicamente 1 o 2.

[CK\\_EVENT\\_DURATION](#)

[PK\\_EVENT](#), [FK\\_EVENT\\_MODLE](#)

[PK\\_TEACHES](#), [FK\\_TEACHES\\_EVENT](#), [FK\\_TEACHES\\_STAFF](#)

### **Register an Event (Ad,Mo,Co,El)**

#### **Ad**

El id se genera automáticamente: id del módulo, un punto, el tipo del evento y dos dígitos aleatorio.

[TR\\_EVENT\\_BI](#) ([disparador.automatizar](#))

Si está programado a las 20:00 la duración debe ser una hora.

[CK\\_EVENT\\_TOD\\_DURATION](#) ([tupla.restringir](#))

#### **Mo**

El único dato a modificar es el salón.

[TR\\_EVENT\\_BU](#) ([disparador.restringir](#))

Se pueden adicionar profesores; pero no modificar ni eliminar.

[TR\\_TEACHES\\_BU](#), [TR\\_TEACHES\\_BD](#) ([disparador.restringir](#))

#### **Co**

Consulta todos los datos del evento, incluyendo sus profesores.

#### **El**

Siempre se puede eliminar

[FK\\_TEACHES\\_EVENT](#) ([accion.automatizar](#))