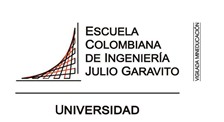
****

**Escuela Colombiana de Ingeniería Julio Garavito**

**Modelos y bases de datos 2024-1**

**Autoestudio IV**

**Cristian Santiago Pedraza Rodríguez**

**Andersson David Sánchez Méndez**

**17 de marzo de 2024**

# 

## MODELOS Y BASES DE DATOS

**PL/ SQL Básico 2024-1**

## 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

### INVESTIGACIÓN

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

Acciones que va a tomar la base de datos cuando un registro que está siendo referenciado en otra tabla sea actualizado o eliminado.

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

**SET NULL:** Si se elimina un registro de la tabla referenciada (TABLA2) cuyo valor existe en la tabla principal (TABLA1), dicho registro se elimina y los valores coincidentes en la tabla principal se establecen a NULL.

**CASCADE:** Si se elimina un registro de la tabla referenciada en una FOREIGN KEY (TABLA2), los registros coincidentes en la tabla principal (TABLA1) también se eliminan.

**NO ACTION:** Si se intenta eliminar un registro de la tabla referenciada por una FOREIGN KEY, Oracle no lo permita y muestre un mensaje de error. Se establece omitiendo la cláusula ON DELETE al establecer la restricción

### PL/SQL

* 1. ¿Qué es PL/SQL?

PL/SQL (Procedural Language/Structured Query Language) es un lenguaje de procedimiento diseñado para abarcar sentencias SQL dentro de su sintaxis. Se compilan las unidades del programa con el servidor de Oracle DB almacenadas en una base de datos.

**CREACIÓN Y GESTIÓN DE UNIDADES DE PROGRAMA**

* Creación con bloques
* Control del flujo de ejecución
* Paquete
* Gestión de errores
* The Data Dictionary

**SQL**

* Trabajar con cursores.
* Procesamiento masivo con BULK COLLECT y FORALL.
  1. ¿Qué motores lo soportan?

PL/SQL es exclusivamente el lenguaje de programación para la Base de datos de Oracle. Se puede crear Funciones, Paquetes, Procedimientos y Triggers.

T-SQL sería su homólogo para bases de datos SQL Server (Microsoft)

Lo único standard entre los motores de base de datos es el lenguaje de consulta ANSI-SQL.

Algunos motores de base de datos como MariaDB se puede configurar para ser compatibles.

### Datos e instrucciones en PL/SQL

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

**Trabajar con cadenas en PL/SQL**, **trabajar con cifras en PL/SQL**, t**rabajar con fechas en PL/SQL**, **trabajar con registros**, **gestión de errores** y **trabajar con recopilaciones**

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

Para definir constantes:

Se hace añadiendo la palabra **CONSTANT,** asignándole un valor a esa constante.

**EJEMPLO:**

Interes NUMBER(5,3);

Descripcion VARCHAR2(50) :**=** 'inicial';

Fecha\_max DATE;

Contabilizado BOOLEAN :**=** **TRUE**;

PI CONSTANT REAL :**=** 3.14159;

Para definir variables:

Se hace con la palabra **DECLARE**, con la siguiente sintaxis (si alguna variable no se inicializa, por defecto pondrá **NULL**):

Nombre\_variable [CONSTANT] TIPO [**NOT** **NULL**] [:**=**inicialización];

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

El tipo de una variable se puede declarar tomando de referencia otro identificador, se usa el atributo **%TYPE** y se declara con un tipo estructurado con el atributo **%ROWTYPE**(tomar valores de una columna). El atributo %TYPE nos ayuda a usarlo sin necesidad de conocer el tipo exacto, y si cambiamos ese tipo, automáticamente varía el tiempo de ejecución.

* 1. ¿Cuál es la forma de los diferentes tipos de asignación? (Son tres)
* **Asignación directa**: operador := para asignar un valor a una variable.

SQL

v\_variable := 'Hello, World!';

* **Asignación a través de una función**: asignación del valor devuelto por una función a una variable.

SQL

v\_variable := funcion();

* **Asignación a través de una consulta SELECT INTO**: asignación del resultado de una consulta SQL a una variable local.

SQL

SELECT columna

INTO v\_variable

FROM tabla

WHERE condicion;

### Cursores

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

Este tipo de cursores se utiliza para operaciones SELECT INTO. Se usan cuando la consulta devuelve un único registro.

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

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

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

**CURSOR IMPLÍCITO**

* Los cursores implícitos no deben ser declarados.
* Con cada cursor debe existir la palabra clave **INTO**.
* Las variables con el mismo tipo de dato.
* Devuelven una única fila. En caso de que no pase eso, entonces generará algunos errores como:
* NO\_DATA\_FOUND: Sentencia **SELECT** intenta recuperar datos, pero ninguna fila satisface sus condiciones (no hay datos).
* TOO\_MANY\_ROWS: Detecta la existencia de más de una fila.

**CURSOR EXPLÍCITO**

* **Declarar el cursor:** Nombre que tendrá el cursor y qué consulta SELECT ejecutará.

La sintaxis básica es:

**CURSOR** nombre\_cursor **IS** instrucción\_SELECT

**CURSOR** nombre\_cursor(param1 tipo1, ..., paramN tipoN) **IS** instrucción\_SELECT

* **Abrir el cursor en el servidor:** Se ejecuta la sentencia SELECT asociada y el resultado se guarda en el servidor en un área de memoria interna (tablas temporales).

La sintaxis de apertura es:

**OPEN** nombre\_cursor;

**OPEN** nombre\_cursor(valor1, valor2, ..., valorN);

* **Recuperar cada una de sus filas (bucle):** Se accederá a una única fila.

La sintaxis de recuperación de fila es:

**FETCH** nombre\_cursor **INTO** variables;

* **Cerrar el cursor:** Cuando se recuperen todas las filas del cursor, hay que cerrarlo para liberar de la memoria los objetos temporales creados.

La sintaxis es:

**CLOSE** numbre\_cursor;

### Modularidad

* 1. ¿Cuál es la estructura general de un bloque PL/SQL?
* **Sección de declaración:** Es opcional. Se declaran todas las constantes, variables, cursores, excepciones, subprogramas, instrucciones, pragma y colecciones que se necesitan en el bloque.
* **Sección de ejecución:** Parte principal y obligatoria que ejecuta el código escrito en su interior, tanto código PL/SQL como código SQL.
* **Sección de manejo de excepciones:** Opcional de los bloques PL/SQL. Se manejan las excepciones que ocurren durante la ejecución del bloque.
  1. ¿Para qué sirven las diferentes estructuras modulares? (bloque anónimo, procedimiento, función y disparador).
* **BLOQUE ANÓNIMO:** Puede implementar la lógica de procedimiento en un lenguaje de scripts. En contextos PL/SQL, esta sentencia la puede compilar y ejecutar el servidor de datos DB2 con sentencias de control PL/SQL y sentencias SQL.
* **PROCEDIMIENTO:** Objeto de base de datos que contiene lógica de procedimiento PL/SQL y sentencias SQL invocados en sentencias o RECALL como:
* **Sentencia CREATE PROCEDURE**: procedimiento almacenado en la base de datos.
* **Referencias de procedimiento:** Invocación a procedimientos PL/SQL en contextos PL/SQL.
* **Soporte de sintaxis de invocación de función**: Aplicado a una sentencia de asignación PL/SQL.

# **FUNCIONES:** compilación y ejecución de funciones PL/SQL escalares (contexto con expresiones válidas) e interconectadas (invocar en cláusula FROM de sentencias SELECT para calcular una tabla de una fila a la vez).

* **Sentencia CREATE FUNCTION:** Define una función escalar o de interconexión que se almacena en la base de datos.
* **Referencias de función**: referencias escalares siempre y cuando se admita una expresión.

# **DISPARADOR:** Objeto de base de datos con nombre que encapsula y define un conjunto de acciones que se van a realizar en respuesta a una operación de inserción, actualización o supresión en una tabla.

* **Tipos de desencadenantes:** Desencadenantes a nivel de fila y a nivel de sentencia dentro de un contexto PL/SQL.
* **Variables de desencadenante:** NEW y OLD son variables especiales que puede utilizar con desencadenantes PL/SQL sin definirlos explícitamente.
* **Predicados de suceso desencadenante:** UPDATING, DELETING e INSERTING solo se pueden utilizar en un desencadenante para identificar el suceso que ha activado el desencadenante.
* **Transacciones y excepciones:** Un desencadenante siempre se ejecuta como parte de la misma transacción en la que se ejecuta la sentencia desencadenante.
* **Sentencia CREATE TRIGGER**: Define un desencadenante PL/SQL en la base de datos.
* **Descarte de desencadenantes:** Puede eliminar un desencadenante de la base de datos utilizando la sentencia DROP TRIGGER.

**REFERENCIAS**

### [https://gavilanch.wordpress.com/2018/05/06/entity-framework-core-2-acciones-referenciales/#:~:text=Las%20acciones%20referenciales%20son%20las%20que%20nos%20dicen%20qué%20acción,otro%20registro%20de%20otra%20tabla](https://gavilanch.wordpress.com/2018/05/06/entity-framework-core-2-acciones-referenciales/#:~:text=Las%20acciones%20referenciales%20son%20las%20que%20nos%20dicen%20qu%C3%A9%20acci%C3%B3n,otro%20registro%20de%20otra%20tabla).

* <https://www.tutorialesprogramacionya.com/oracleya/temarios/descripcion.php?cod=220>
* <https://www.oracle.com/co/database/technologies/appdev/plsql.html>
* [https://es.stackoverflow.com/questions/440291/qué-motores-de-base-de-datos-pueden-hacer-uso-de-pl-sql#:~:text=1%20respuesta&text=PL%2FSQL%20es%20exclusivamente%20el,lenguaje%20y%20sus%20propias%20características](https://es.stackoverflow.com/questions/440291/qu%C3%A9-motores-de-base-de-datos-pueden-hacer-uso-de-pl-sql#:~:text=1%20respuesta&text=PL%2FSQL%20es%20exclusivamente%20el,lenguaje%20y%20sus%20propias%20caracter%C3%ADsticas).
* [https://elbauldelprogramador.com/plsql-declaracion-de-variables/#:~:text=Índice,por%20defecto%20el%20valor%20NULL%20](https://elbauldelprogramador.com/plsql-declaracion-de-variables/#:~:text=%C3%8Dndice,por%20defecto%20el%20valor%20NULL%20).
* [https://elbauldelprogramador.com/plsql-cursores/#:~:text=Los%20cursores%20implícitos%20se%20utilizan,existir%20la%20palabra%20clave%20INTO%20](https://elbauldelprogramador.com/plsql-cursores/#:~:text=Los%20cursores%20impl%C3%ADcitos%20se%20utilizan,existir%20la%20palabra%20clave%20INTO%20).
* <https://www.ibm.com/docs/es/db2/11.1?topic=support-blocks-plsql>

### PRACTICANDO.

En este autoestudio vamos a implementar un caso de uso de la base de datos

[**GuestHouse**](https://sqlzoo.net/wiki/Guest_House)

|  |  |
| --- | --- |
|  | **TExtra**  Entero(11) Positivo inicia en 9  **TDetail**  Cadena(50) No vacia  **TPrice**  Entero(9) Positivo |
|  | **Register an Extra(Ad,Co,El) Ad**  El id se genera automáticamente: 9AAAAMMDDmmss El descuento no puede ser mayor al 50% del precio.  Si tiene un precio mayor a 100000 y no se indica descuento, se aplica un descuento del 10% y el primer detalle es ‘CON 10% de descuento automático’.  **Mo**  Sólo se pueden adicionar detalles a la descripción, no modificar ni eliminar. **Co**  Consulta todos los datos del extra  **El**  Si no tiene descripción, se puede eliminar |

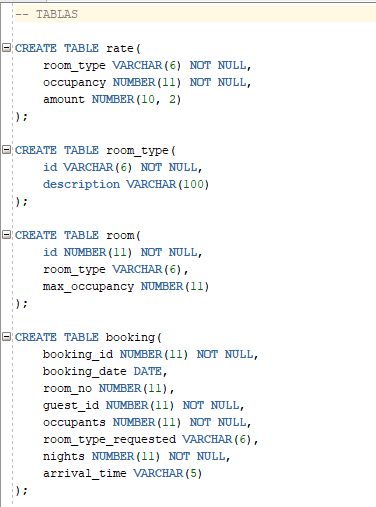
**Consultar las guías de diseño que se presenta al final del autoestudio.**

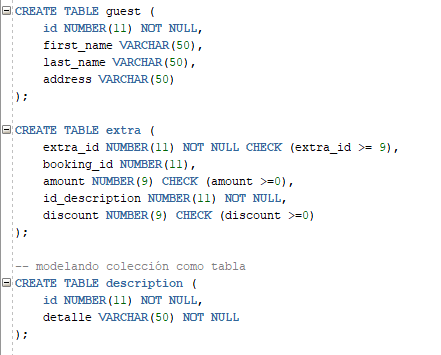
**Consultar los** [**REQUISITOS DE ENTREGA**](https://campusvirtual.escuelaing.edu.co/moodle/mod/wiki/view.php?id=59242)en la sección Proyecto del Curso correspondiente a: **Restricciones Declarativas y Restricciones Declarativas, Procedimentales y Automatización.**

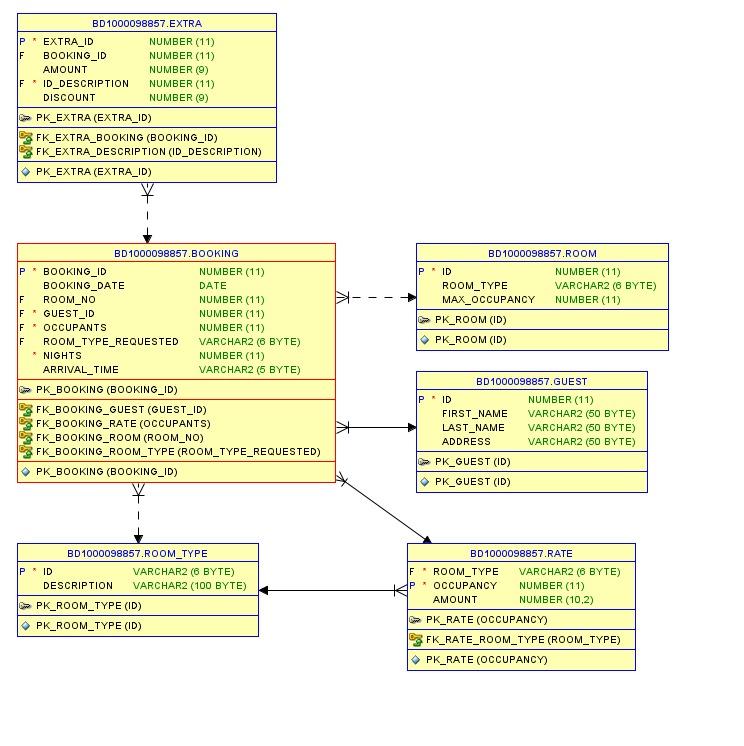
# Adicionandorestriccionesdeclarativas.

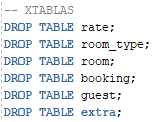
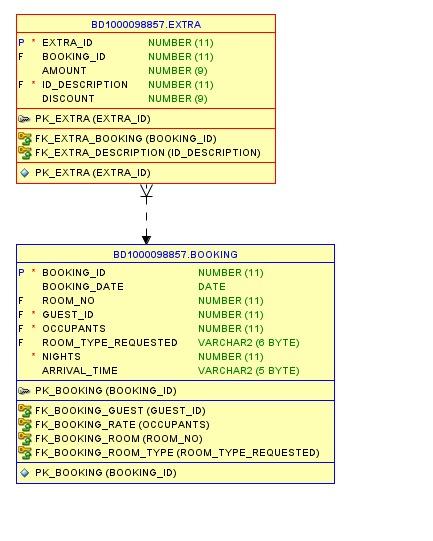
Para proteger la base de datos vamos a implementar las restricciones que se pueden garantizar usando restricciones declarativas.

1. Cree la nueva tabla. (Tablas, XTablas)



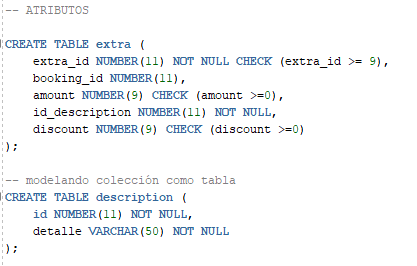




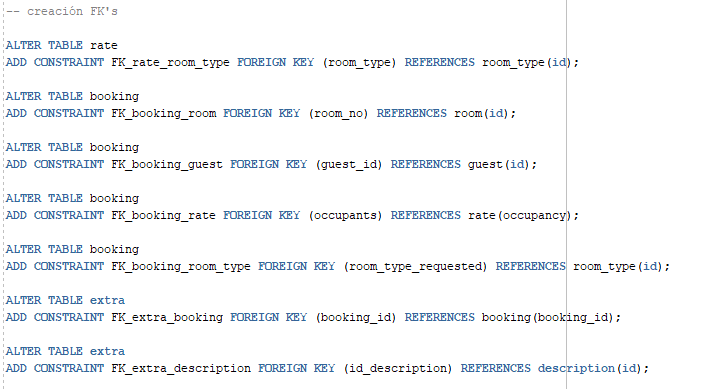


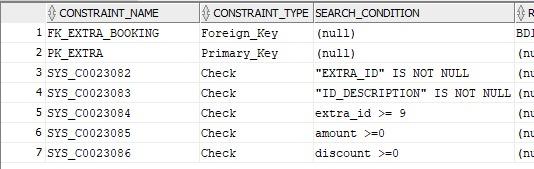
1. Definir las restricciones declarativas

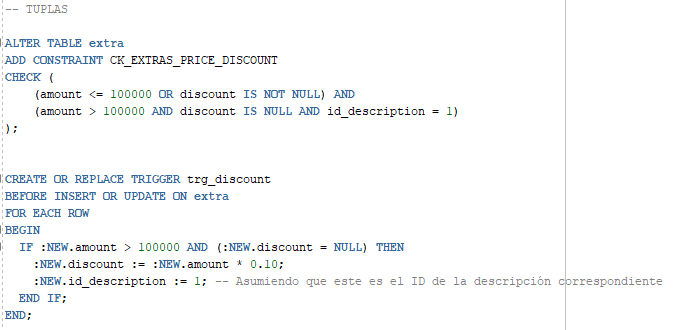
(Atributos, Tuplas)





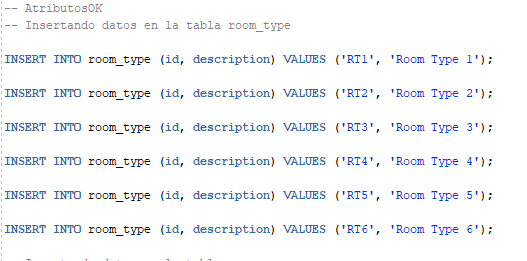




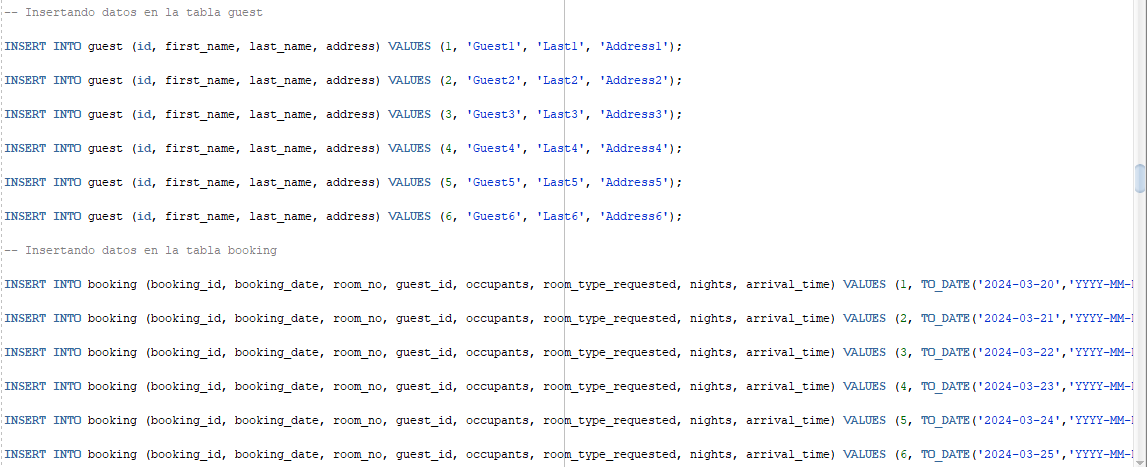


1. Validar con casos significativos la protección de la base de datos, adicione comentarios que expliquen la condición que están validando.

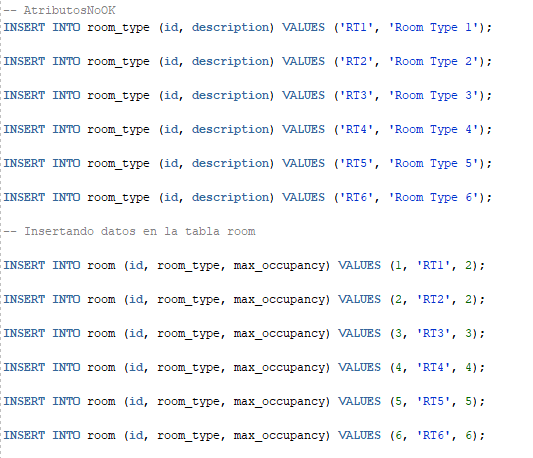
(AtributosOK, AtributosNoOK, TuplasOK, TuplasNoOK)





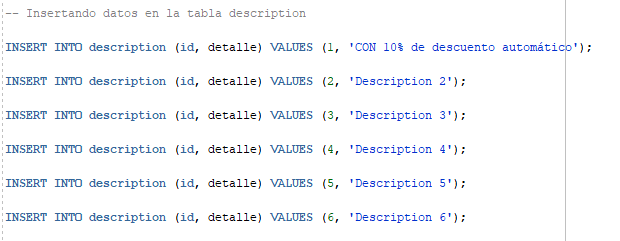












NO DEBERÍA ACEPTAR DATOS REPETIDOS, POR ESO, SON ATRIBUTOS NO OK





amount valor muy grande

# Adicionandoaccionesdereferencia

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

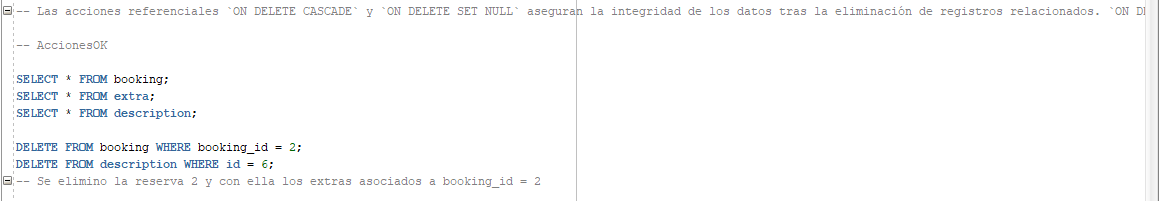
(Deben eliminar y volver a crear las FK afectadas. Justificar la decisión)

(Acciones)



1. Validar con casos significativos las condiciones definidas

(AccionesOK)

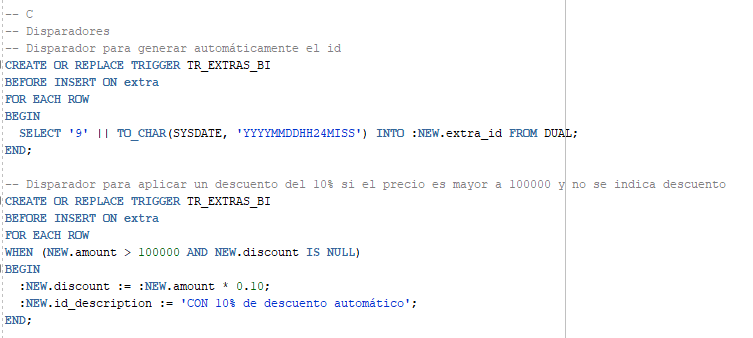


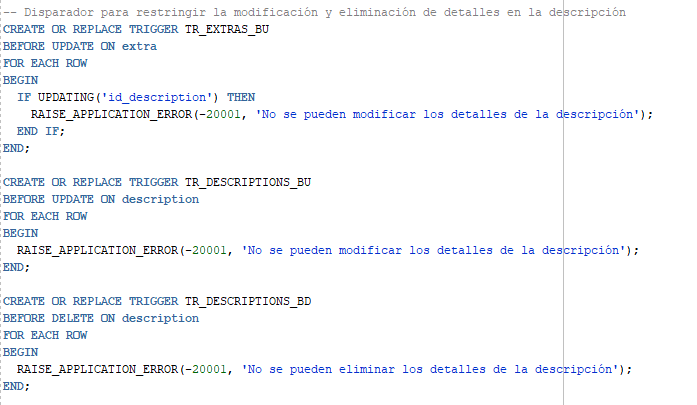
# Adicionandodisparadores

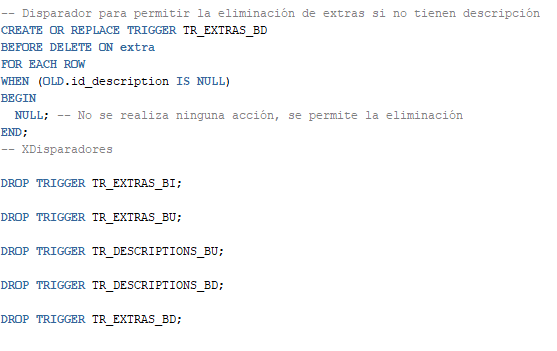
Para preparar las acciones asociadas a los diferentes casos de uso vamos a implementar algunos disparadores (CRUD).

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

( Disparadores – XDisparadores )



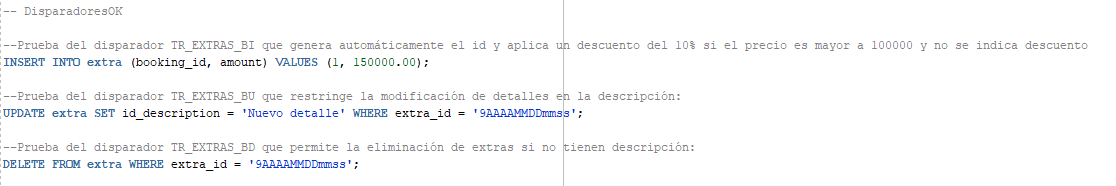




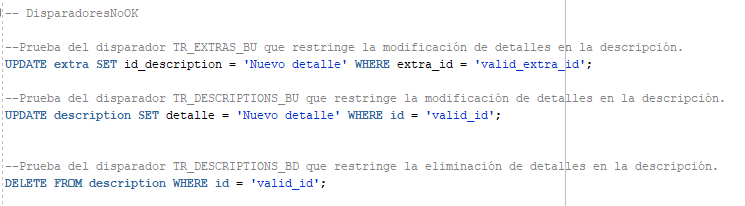


1. Escriba 3 instrucciones que permitan probar la actualización de la base de datos haciendo uso de cada disparador, cuando sea pertinente.

( 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.

|  |
| --- |
| **TExtra**  Entero(11) Positivo, inicia en 9  **CK\_EXTRAS\_ID**  **TDetail**  Cadena(50) No vacia  **CK\_DESCRIPTIONS\_DETAIL**  **TPrice**  Entero(9) Positivo **CK\_EXTRAS\_PRICE CK\_EXTRAS\_DISCOUNT** |
| **PK\_EXTRAS, FK\_EXTRSA\_BOOKING PK\_DESCRIPTIONS, FK\_DESCRIPTIONS\_EXTRA** |
|  |
| **Register an Extra(Ad,Mo, Co,El) Ad**  El id se genera automáticamente: 9AAAAMMDDmmss  **TR\_EXTRAS\_BI (disparador.automatizar)**  El descuento no puede ser mayor al 50% del precio.  **CK\_EXTRAS\_PRICE\_DISCOUNT (tupla.restringir)**  Si tiene un precio mayor a 100000 y no se indica descuento, se aplica un descuento del 10% y el primer detalle es ‘CON 10% de descuento automático’.  **TR\_EXTRAS\_BI (disparador.automatizar) Mo**  Sólo se pueden adicionar detalles a la descripción, no modificar ni eliminar.  **TR\_EXTRAS\_BU (disparador.restringir)**  **TR\_DESCRIPTIONS\_BU, TR\_DESCRIPTIONS\_BD (disparador.restringir) Co**  Consulta todos los datos del extra  **El**  Si no tiene descripcion, se puede eliminar **TR\_EXTRAS\_BD (disparador.restringir) FK\_DESCRIPTIONS\_EXTRAS (accion.automatizar)** |