1. Implementa un trigger para que cuando se modifique el atributo duración en la tabla Vuelo, se actualice automáticamente las horas de vuelo del piloto asociado a dicho vuelo (base de datos aeropuerto).

DELIMITER //

**CREATE** **TRIGGER** vuelo\_AU\_TRIGGER

**BEFORE** **UPDATE** **ON** vuelo

**FOR** **EACH** **ROW**

**BEGIN**

**UPDATE** piloto

**set** piloto.horas\_vuelo=piloto.horas\_vuelo+(**new**.duracion-**old**.duracion)

**where** cod\_piloto=**new**.cod\_piloto;

**END/**/

DELIMITER ;

1. Crea un trigger que permita actualizar las horas de vuelo del piloto tras la inserción de un nuevo vuelo con un piloto específico (base de datos aeropuerto).

DELIMITER //

**CREATE** **TRIGGER** vuelo\_AI\_TRIGGER

**AFTER** **INSERT** **ON** vuelo

**FOR** **EACH** **ROW**

**BEGIN**

**UPDATE** piloto

**set** piloto.horas\_vuelo=piloto.horas\_vuelo+**new**.duracion

**where** cod\_piloto=**new**.cod\_piloto;

**END/**/

DELIMITER ;

1. Implementa un trigger que tras la modificación del id\_concierto en la tabla interpreta actualice automáticamente el campo num\_obras de la tabla concierto (base de datos CursosVeranoPiano).

DELIMITER //

**CREATE** **TRIGGER** interpreta\_BU\_TRIGGER

**BEFORE** **UPDATE** **ON** interpreta

**FOR** **EACH** **ROW**

**BEGIN no funciona**

**UPDATE** concierto

**set** **old**.num\_obras=**new**.num\_obras-1 **and** **new**.num\_obras=**old**.num\_obras-1

**where** interpreta.id\_concierto=id\_concierto;

**END/**/

DELIMITER ;

1. Crea un trigger que permita actualizar el número de obras del concierto tras la inserción en interpreta (base de datos cursosVeranoPiano).

DELIMITER //

**CREATE** **TRIGGER** interpreta\_BI\_TRIGGER

**BEFORE** **INSERT** **ON** interpreta

**FOR** **EACH** **ROW**

**BEGIN**

**UPDATE** concierto

**set** num\_obras=num\_obras+1

**where** interpreta.id\_concierto=id\_concierto;

**END/**/

DELIMITER ;

1. Diseña un disparador (trigger) para actualizar el número de pilotos ante una inserción en la relación participa (base de datos campeonato).

DELIMITER //

**CREATE** **TRIGGER** participa\_BI\_TRIGGER

**BEFORE** **INSERT** **ON** participa

**FOR** **EACH** **ROW**

**BEGIN**

**UPDATE** prueba

**set**

**where** ;

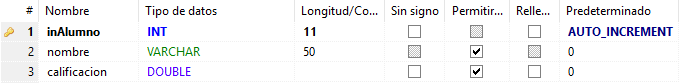
**END/**/

DELIMITER ;

1. Realiza lo siguiente:
   1. Crear una base de datos ‘centro’ que contenga 3 tablas:

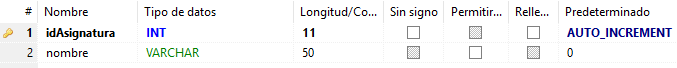
alumno (idAlumno, nombre, calificacion)

Clave primaria: {idAlumno}



asignatura (idAsignatura, nombre)

Clave primaria: {idAsignatura}

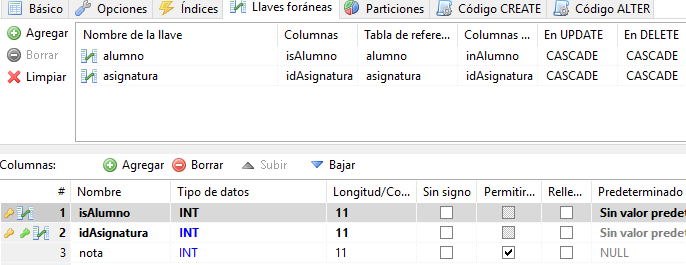


nota (idAlumno, idAsignatura, nota:integer)

Clave primaria: {idAlumno, idAsignatura}

Clave ajena: {idAlumno}

Clave ajena: {idAlumno}



* 1. Introduce los siguientes registros:
     1. Tabla alumno: (1,Pepe,NULL), (2,Manolo,NULL)

**use** centro;

**insert** **into** alumno **values** (1,'Pepe',**NULL**),

(2,'Manolo',**NULL**)

* + 1. Tabla asignatura: (1,BBDD), (2,ED), (3,PRG)

**use** centro;

**insert** **into** asignatura **values** (1,BBDD),

(2,ED),

(3,PRG)

1. Realiza un trigger que cuando se inserte un nuevo registro de nota de un alumno para una asignatura, automáticamente se deberá añadir el comentario a la tabla alumno. Se deberá calcular la nota media y según dicha nota media aparecerá el mensaje de insuficiente, suficiente, bien, notable o excelente.

DELIMITER //

**CREATE** **TRIGGER** nota\_BI\_TRIGGER

**BEFORE** **INSERT** **ON** nota

**FOR** **EACH** **ROW**

**BEGIN**

**UPDATE** alumno

**set**

**where** ;

**END/**/

DELIMITER ;