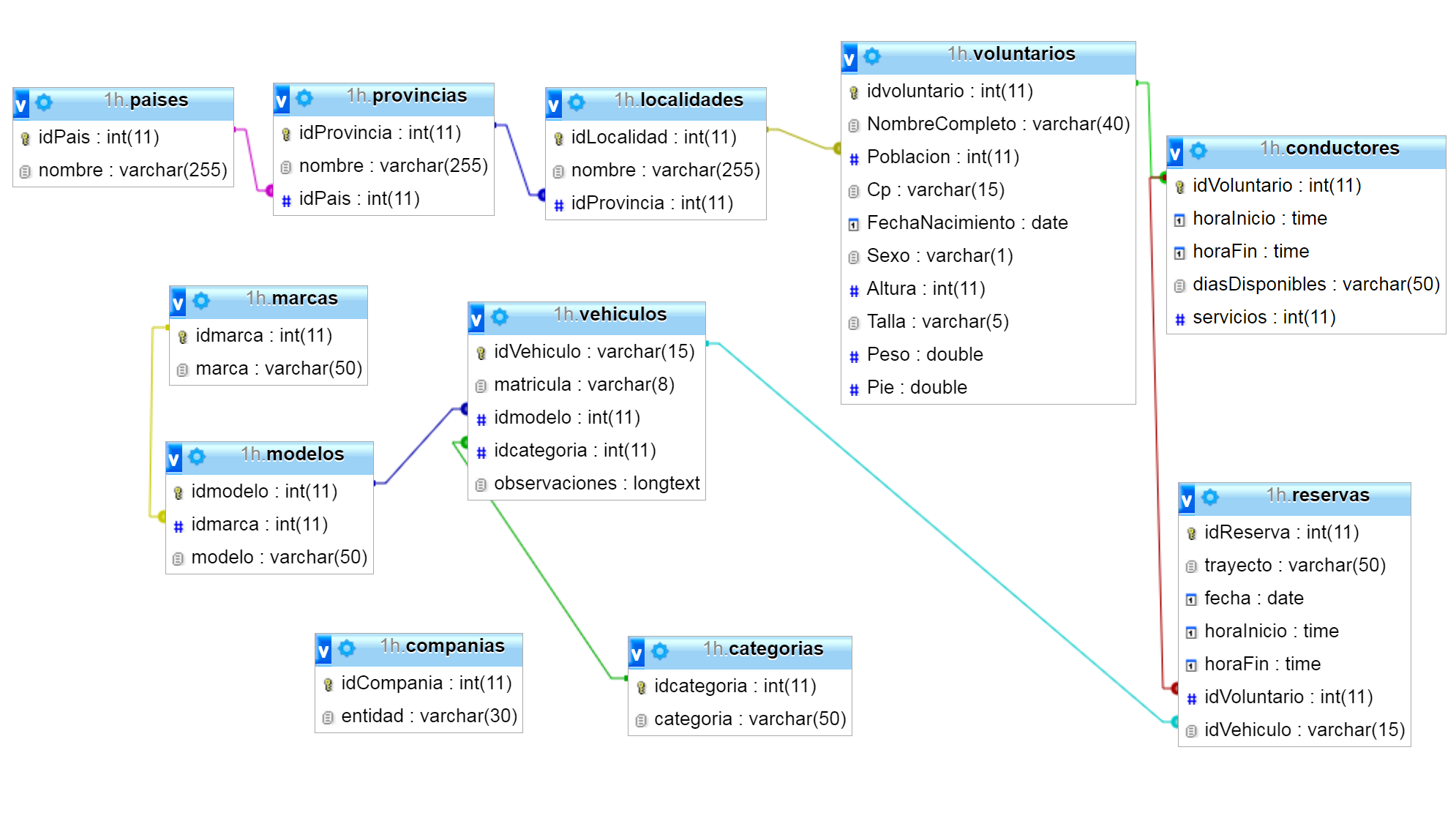
|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Curso: | **2016/2017** | Fecha: | **10/03/2017** | Grupo:**1ºH** |  |  | Etapa:GS |  |
| Asignatura/Módulo: | | **Bases de Datos** | | | | | | |
| Profesor: | | **Mercedes López Fatás** | | | | | | |

Crea la base de datos**examen**, importa el fichero **examen.sql**. Fíjate en el siguiente diseño que muestra las talas que contiene y observarás que la tabla de **companias** está sin relacionar. Teniendo en cuenta que de la misma Compañía puede haber varios vehículos alquilados y que un vehículo está vinculado a una sola Compañía de Alquiler, **modifica la estructura de la tabla** que consideres oportuna e indica la relación que ha surgido:



La tabla de companias no está unida a vehículos, cuando debería haber constancia de que un vehículo pertenece a una compañía. Un vehículo pertenece a una compañía, pero esta puede tener varios vehículos. Por eso se establece una relación N:1, teniéndose que añadir a vehículos la clave foránea de idCompania.

*Escribe una consulta para que los vehículos de la marca NISSAN lleven el valor 2 correspondiente a la Compañía AVIS.*

UPDATE vehiculos, modelos, marcas

SET vehiculos.idCompania = 2

WHERE vehiculos.idmodelo = modelos.idmodelo AND modelos.idmarca = marcas.idmarca AND marcas.marca = "NISSAN"

*Escribe una consulta para que los vehículos de la marca FORD lleven el valor 1 correspondiente a la Compañía Hertz*

UPDATE vehiculos, modelos, marcas

SET vehiculos.idCompania = 1

WHERE vehiculos.idmodelo = modelos.idmodelo AND modelos.idmarca = marcas.idmarca AND marcas.marca = "FORD"

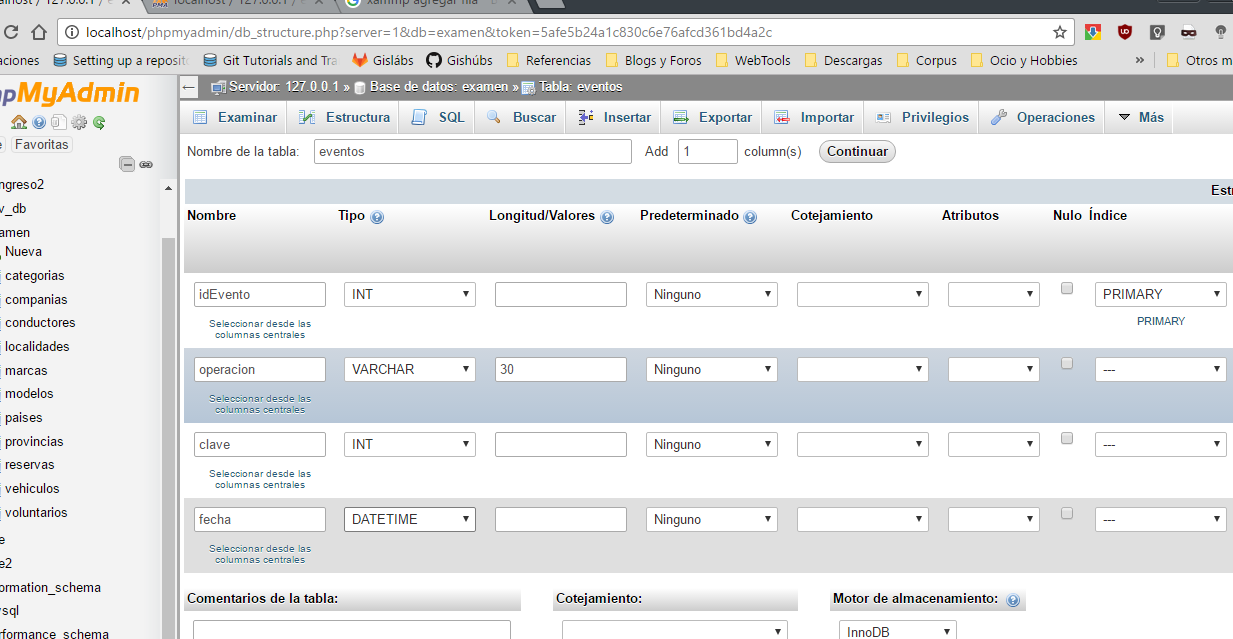
*Escribe una consulta para que los vehículos de la marca CITROEN lleven el valor 3 correspondiente a la Compañía Europcar*

UPDATE vehiculos, modelos, marcas

SET vehiculos.idCompania = 3

WHERE vehiculos.idmodelo = modelos.idmodelo AND modelos.idmarca = marcas.idmarca AND marcas.marca = "CITROEN"

*Crea una tabla llamada* ***eventos*** *con los campos* ***idevento*** *INT PRIMARY KEY Auto Increment,* ***operacion****VARCHAR 30,* ***clave*** *INT,* ***fecha*** *DATETIME*

**

*Añade un campo nuevo en la tabla* ***vehiculos*** *llamado* ***numtrayectos*** *de tipo INT*



1. Escribe una consulta para mostrar la **matricula** y **marca** de aquellos vehiculos que no hayan sido nunca reservados.

SELECT vehiculos.matricula, marcas.marca

FROM vehiculos, marcas, modelos

WHERE vehiculos.idmodelo = modelos.idmodelo AND modelos.idmarca = marcas.idmarca AND vehiculos.idVehiculo NOT IN

(

SELECT reservas.idVehiculo

FROM reservas

)

1. Escribe una consulta para mostrar **nombreCompleto, fechaNacimiento** y nombre de la provincia de los **conductores** de Aragón ordenados descendentemente por la fecha de Nacimiento

SELECT voluntarios.NombreCompleto, voluntarios.FechaNacimiento, provincias.nombre

FROM voluntarios, localidades, provincias

WHERE voluntarios.Poblacion = localidades.idLocalidad AND localidades.idProvincia = provincias.idProvincia AND (provincias.nombre = "Zaragoza" OR provincias.nombre = "Huesca" OR provincias.nombre = "Teruel") AND voluntarios.idvoluntario IN

(

SELECT conductores.idVoluntario

FROM conductores

)

ORDER BY voluntarios.fechaNacimiento DESC

1. Escribe una consulta para mostrar el **modelo** y la cantidad de vehiculos con el alias TOTAL VEHICULOS que tenemos **de cadamodelo**.

SELECT modelos.modelo, COUNT(vehiculos.idVehiculo) AS `TOTAL VEHICULOS`

FROM vehiculos, modelos

WHERE vehiculos.idmodelo = modelos.idmodelo

GROUP BY modelos.modelo

1. Mostrar todos los campos de conductores que tengan disponibles los días 24 y 25 y su disponibilidad horaria contemple el siguiente tramo horario 14:00 a 18:00.

P.e. un conductor con **horaInicio** 8:00 y **horaFin** 19:00 y **diasDisponibles** 17,24,25 sería seleccionado, pero un conductor con **horainicio** 17:00 y **horaFin** 20:00 y **diasDisponibles** 17,24,25 no sería seleccionado por empezar a las 17:00 en lugar de las 14:00 o antes.

SELECT \*

FROM conductores

WHERE conductores.horaInicio <= '14:00:00' AND conductores.horaFin >='18:00:00'

AND conductores.diasDisponibles LIKE "%,24,25"

1. Escribe una consulta que nos muestre la matricula, el nombre de modelo de aquellos vehículos en cuya matrícula tenga, en la parte de las letras, una D como primer carácter.

SELECT vehiculos.matricula, modelos.modelo

FROM vehiculos, modelos

WHERE vehiculos.idmodelo = modelos.idmodelo AND vehiculos.matricula LIKE "\_\_\_\_\_D%"

1. Escribe una consulta que muestre el **nombre** de provincia y la**cantidad de voluntarios** que tenemos **de cada**provincia, pero sólo de aquellas provincias cuya cantidad de voluntarios sea inferior a 5.

SELECT provincias.nombre, COUNT(voluntarios.idvoluntario)

FROM provincias, localidades, voluntarios

WHERE provincias.idProvincia = localidades.idLocalidad AND localidades.idLocalidad = voluntarios.Poblacion

GROUP BY provincias.nombre

HAVING COUNT(voluntarios.idvoluntario < 5)

1. Mostrar **nombreCompleto,FechaNacimiento**, **nombre** de provincia de conductores entre 40 y 50 años

SELECT voluntarios.NombreCompleto, voluntarios.FechaNacimiento, provincias.nombre

FROM voluntarios, provincias, localidades

WHERE (voluntarios.Poblacion = localidades.idLocalidad AND localidades.idProvincia = provincias.idProvincia) AND TIMESTAMPDIFF(year,voluntarios.FechaNacimiento, CURDATE()) < 50 AND TIMESTAMPDIFF(year,voluntarios.FechaNacimiento, CURDATE()) > 40 AND voluntarios.idvoluntario IN

(

SELECT conductores.idVoluntario

FROM conductores

)

T1 **Evento**: DESPUES DE INSERTAR en reservas

**Acciones**:

Modificar el campo **numtrayectos** de vehiculos aumentado en 1 su valor

Modificar el campo **servicios** de conductores aumentado en 1 su valor

Insertar en la tabla **eventos** un registro donde el campo de **operacion**contenga ALTA RESERVAS, la **clave** el idReserva añadido y **fecha** que contenga la **fecha y hora actual**.

DROP TRIGGER IF EXISTS T1

//

CREATE TRIGGER T1

AFTER INSERT ON reservas

FOR EACH ROW

BEGIN

UPDATE vehiculos SET vehiculos.numTrayectos = vehiculos.numTrayectos + 1

WHERE reservas.idVehiculo = vehiculos.idVehiculo;

UPDATE conductores SET conductores.servicios = conductores.servicios + 1

WHERE reservas.idVoluntario = conductores.idVoluntario;

INSERT INTO eventos (eventos.idEvento, eventos.operacion, eventos.fecha) VALUES (new.idReserva, "ALTA RESERVAS", now());

END

//

T2 **Evento**: ANTES DE ELIMINAR en reservas

**Acciones**:

Modificar el campo **numtrayectos** de vehiculos disminuyendo en 1 su valor

Modificar el campo **servicios** de conductores disminuyendo en 1 su valor

Insertar en la tabla **eventos** un registro donde el campo de **operacion**contenga BAJA RESERVAS, la **clave** el idReserva que va a borrarse y **fecha** que contenga la **fecha y hora actual**.

DROP TRIGGER IF EXISTS T2

//

CREATE TRIGGER T2

AFTER DELETE ON reservas

FOR EACH ROW

BEGIN

UPDATE vehiculos SET vehiculos.numTrayectos = vehiculos.numTrayectos - 1

WHERE reservas.idVehiculo = vehiculos.idVehiculo;

UPDATE conductores SET conductores.servicios = conductores.servicios - 1

WHERE reservas.idVoluntario = conductores.idVoluntario;

INSERT INTO eventos (eventos.idEvento, eventos.operacion, eventos.fecha) VALUES (old.idReserva, "BAJA RESERVAS", now());

END

//

T3 **Evento**: ANTES DE ELIMINAR en voluntarios

**Acciones**:

Prepara las acciones que consideres que tendrían que realizarse cuando eliminemos un voluntario para mantener la **integridad referencial de todas las tablas**

Insertar en la tabla **eventos** un registro donde el campo de **operacion**contengaBAJA VOLUNTARIO, la **clave** el idVoluntario que se va a cambiar y **fecha** que contenga la **fecha y hora actual**.

DROP TRIGGER IF EXISTS T3

//

CREATE TRIGGER T3

AFTER DELETE ON voluntarios

FOR EACH ROW

BEGIN

INSERT INTO eventos (eventos.idEvento, eventos.operacion, eventos.fecha) VALUES (old.idVoluntario, "BAJA VOLUNTARIO", now());

DELETE FROM conductores WHERE conductores.idVoluntario = old.idVoluntario;

DELETE FROM reservas WHERE reservas.idVoluntario = old.idVoluntario;

END

//