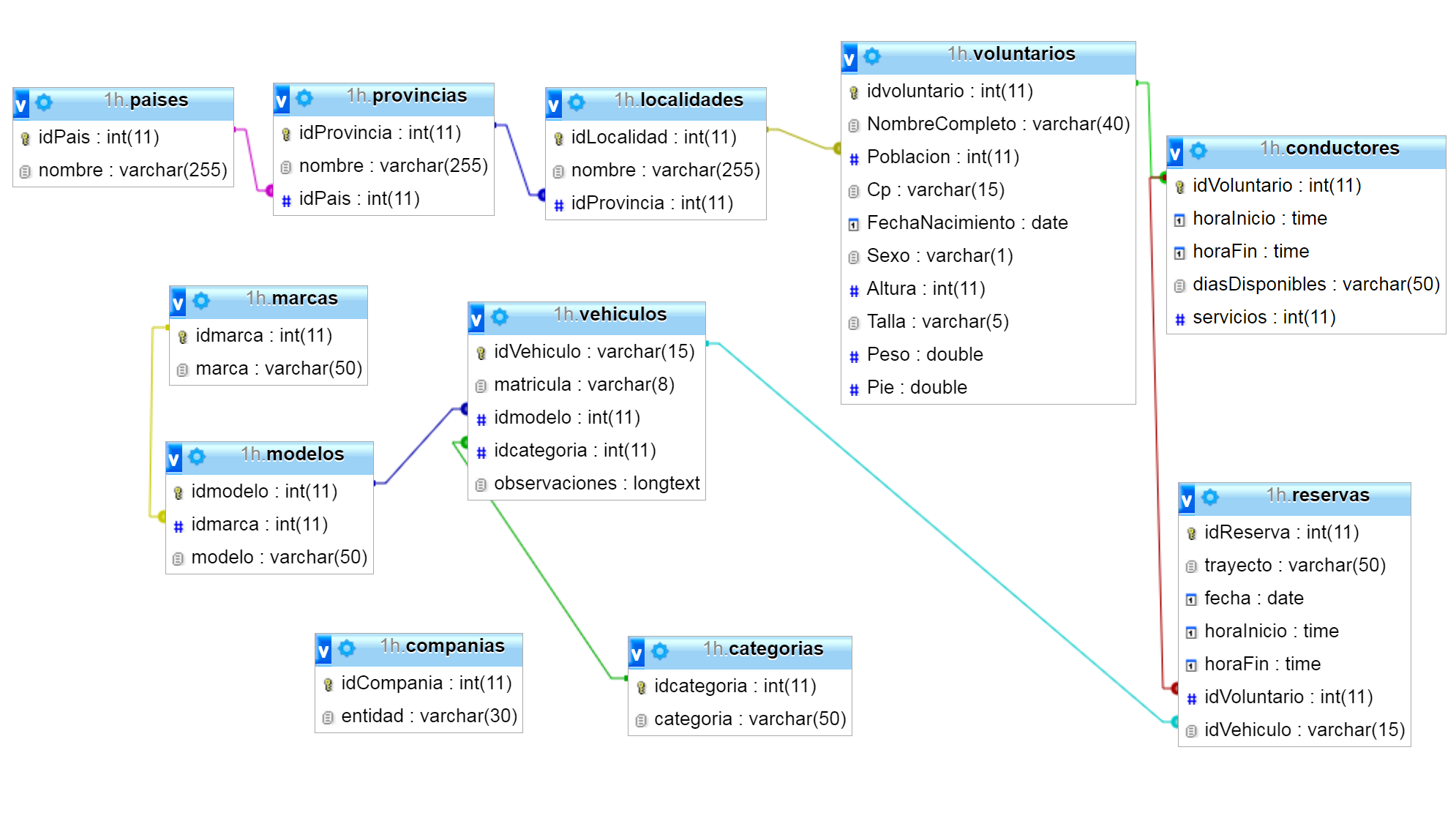
|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Curso: | **2016/2017** | Fecha: | **10/03/2017** | Grupo:**1ºH** |  |  | Etapa:GS |  |
| Asignatura/Módulo: | | **Bases de Datos** | | | | | | |
| Profesor: | | **Mercedes López Fatás** | | | | | | |

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:



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

-UPDATE vehiculos AS v, marcas AS ma, modelos AS mo SET v.idcompañia=2 WHERE v.idmodelo=mo.idmodelo AND mo.idmarca=ma.idmarca AND ma.marca="NISSAN"

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

-UPDATE vehiculos AS v, marcas AS ma, modelos AS mo SET v.idcompañia=1 WHERE v.idmodelo=mo.idmodelo AND mo.idmarca=ma.idmarca AND ma.marca="FORD"

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

-UPDATE vehiculos AS v, marcas AS ma, modelos AS mo SET v.idcompañia=3 WHERE v.idmodelo=mo.idmodelo AND mo.idmarca=ma.idmarca AND ma.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 v.matricula, ma.marca FROM vehiculos AS v, marcas AS ma, modelos AS mo WHERE v.idmodelo=mo.idmodelo AND mo.idmarca=ma.idmarca AND v.idvehiculo NOT IN (SELECT r.idvehiculo FROM reservas AS r)

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 vo.nombreCompleto, vo.fechaNacimiento, pr.nombre FROM voluntarios AS vo, localidades AS lo, provincias AS pr WHERE vo.poblacion=lo.idLocalidad AND lo.idProvincia=pr.idProvincia AND (pr.nombre="Zaragoza" OR pr.nombre="Huesca" OR pr.nombre="Teruel") ORDER BY vo.fechaNacimiento DESC

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

- SELECT ca.categoria, COUNT(v.idvehiculo) as 'TOTAL VEHÍCULOS' FROM vehiculos as v, categorias as ca WHERE v.idcategoria = ca.idcategoria GROUP BY ca.categoria

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 as c WHERE c.horarioInicio <= ’14:00:00’ AND horaFin >= ’18:00:00’

AND 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 v.matricula, mo.modelo FROM vehiculos AS v, modelos AS mo WHERE v.idmodelo=mo.idmodelo AND v.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 pro.nombre, COUNT(vo.idvoluntario) AS 'TOTAL VOLUNTARIOS' FROM voluntarios AS vo, provincias AS pro WHERE vo.poblacion=pro.idprovincia GROUP BY pro.nombre HAVING COUNT (vo.idvoluntario) < 5

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

-SELECT v.NombreCompleto, v.FechaNacimiento, p.nombre FROM conductores as c, voluntarios as v, localidades as l WHERE(c.idvoluntario=v.idvoluntario AND v.poblacion =l.idLocalidad AND l.idProvincia = p.idprovincia)

AND TIMESTAMPDIFF(year, v.FechaNacimiento, CURDATE()) <= 50 AND TIMESTAMPDIFF(year, v.FechaNacimiento, CURDATE()) >=40

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 numtrayectos = numtrayectos+1;

UPDATE conductores SET servicios = servicios+1;

INSERT INTO eventos (operacion,clave,fecha) VALUES ('ALTA RESERVAS', new.idreserva,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

BEFORE DELETE ON reservas

For each row begin

UPDATE vehiculos SET numtrayectos = numtrayectos-1;

UPDATE conductores SET servicios = servicios-1;

INSERT INTO eventos(operacion,clave,fecha) VALUES (‘BAJA RESERVAS’,old.idreserva,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** contenga BAJA VOLUNTARIO, la **clave** el idVoluntario que se va a cambiar y **fecha** que contenga la **fecha y hora actual**.

-- DROP TRIGGER if EXISTS T2 //

CREATE TRIGGER T2

BEFORE DELETE ON reservas

For each row begin

INSERT INTO eventos(operacion,clave,fecha) VALUES (‘BAJA RESERVAS’,old.idreserva,NOW());

DELETE conductores WHERE idVoluntario = old.idVoluntario;

DELETE reservas WHERE idVoluntario = old.idVoluntario;

end

//