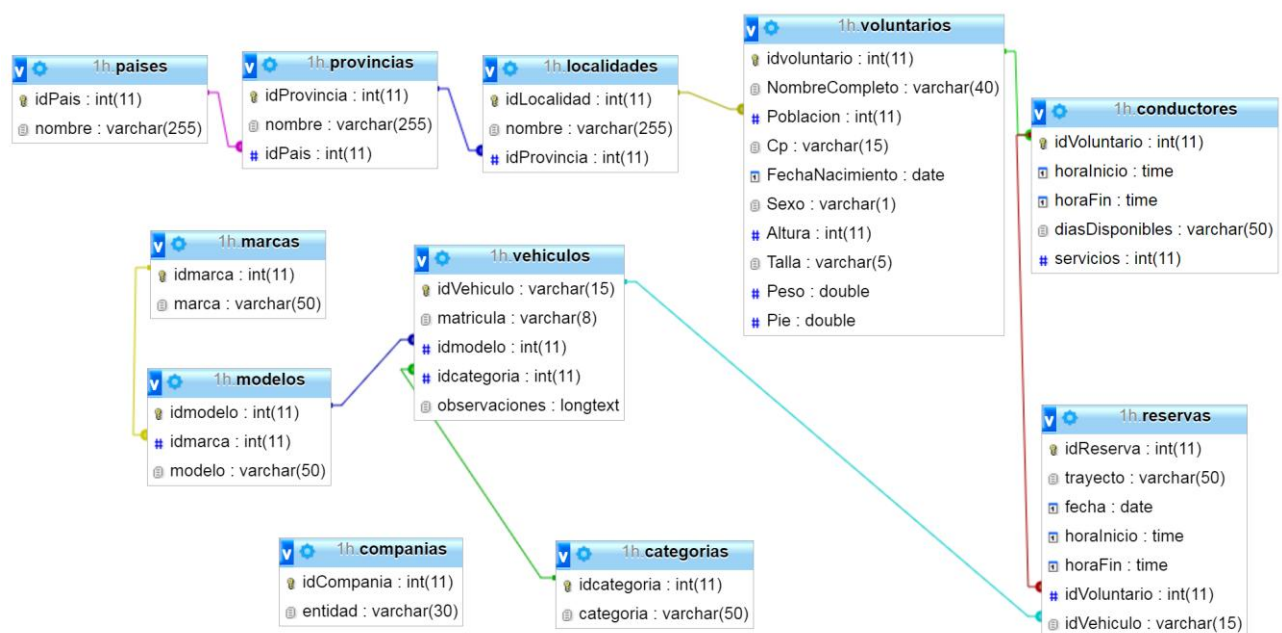


Apellidos:González Tello				Nombre: Javier				Nº. 11			
Curso:	2016/2017		Fecha:	10/03/2017		Grupo:1ºH			Etapa:GS		
Asignatura/Módulo:			Bases de Datos								
Profesor:			Mercedes López Fatás								

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.



Apellidos:	Nombre:	Nº
------------	---------	----

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

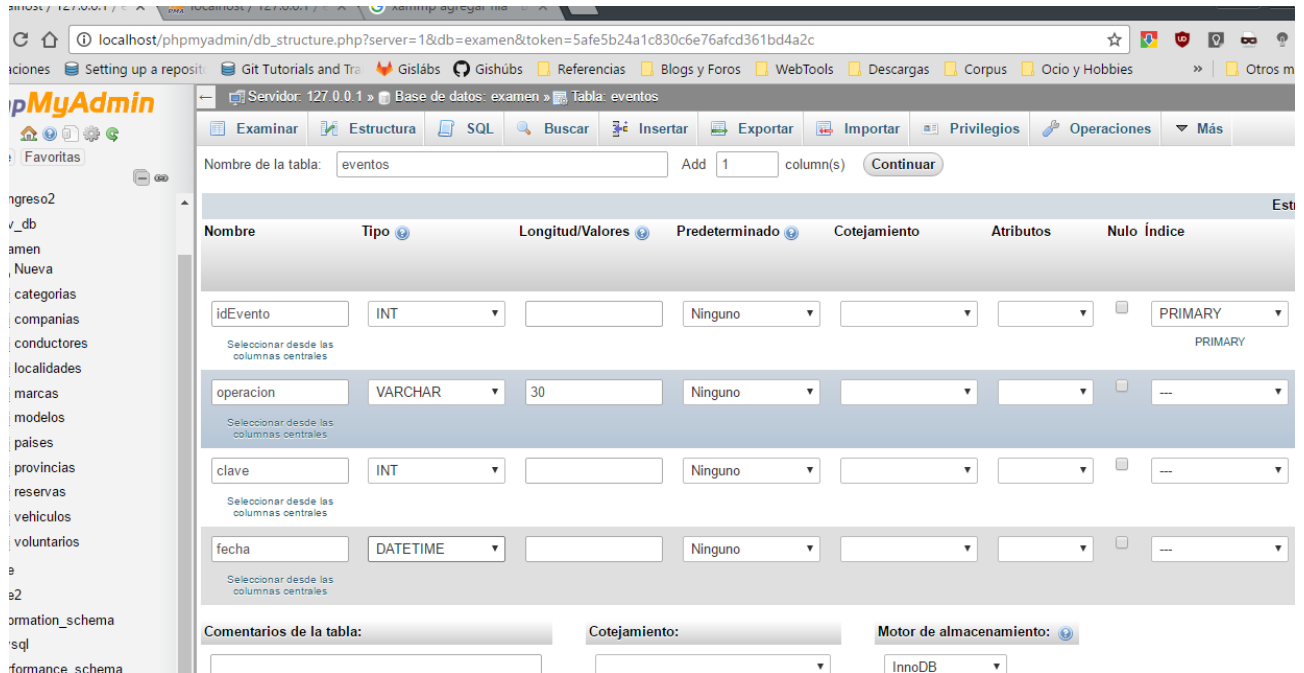
```
UPDATE vehiculos, modelos, marcas
SET vehiculos.idCompania = 1
WHERE vehiculos.idmodelo = modelos.idmodelo AND
modelos.idmarca = marcas.idmarca AND marcas.marca
= "FORD"
```

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"
```

Apellidos:	Nombre:	Nº
------------	---------	----

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



Nombre de la tabla: eventos Add 1 column(s) Continuar

Nombre	Tipo	Longitud/Valores	Predeterminado	Cotejamiento	Atributos	Nulo	Índice
idEvento	INT		Ninguno			PRIMARY	PRIMARY
operacion	VARCHAR	30	Ninguno				
clave	INT		Ninguno				
fecha	DATETIME		Ninguno				

Comentarios de la tabla: Cotejamiento: Motor de almacenamiento: InnoDB

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



#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Extra	Acción
1	idVehiculo	varchar(15)			No	Ninguna		Cambiar Eliminar Primaria Único Índice Espacial Texto completo Más
2	matricula	varchar(8)			Si	NULL		Cambiar Eliminar Primaria Único Índice Espacial Texto completo Más
3	idmodelo	int(11)			No	Ninguna		Cambiar Eliminar Primaria Único Índice Espacial Texto completo Más
4	idcategoria	int(11)			No	Ninguna		Cambiar Eliminar Primaria Único Índice Espacial Texto completo Más
5	observaciones	longtext			Si	NULL		Cambiar Eliminar Primaria Único Índice Espacial Texto completo Más
6	idCompania	int(11)			No	Ninguna		Cambiar Eliminar Primaria Único Índice Espacial Texto completo Más
7	numTrayectos	int(11)			No	Ninguna		Cambiar Eliminar Primaria Único Índice Espacial Texto completo Más

Seleccionar todo Para los elementos que están marcados: Examinar Cambiar Eliminar Primaria Único Índice Espacial Texto completo
Agregar a columnas centrales Eliminar de las columnas centrales



Apellidos:	Nombre:	Nº
------------	---------	----

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
)
```



Apellidos:	Nombre:	Nº
------------	---------	----

2. Escribe una consulta para mostrar **nombreCompleto**, **fechaNacimiento** y nombre de la provincia de los **conductores** de Aragón ordenados descendientemente 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
```



salesianos

COLEGIO NTRA. SRA. DEL PILAR ZARAGOZA

Apellidos:	Nombre:	Nº
------------	---------	----

3. 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
```



Apellidos:	Nombre:	Nº
------------	---------	----

4. 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"
```



salesianos

COLEGIO NTRA. SRA. DEL PILAR ZARAGOZA

Apellidos:	Nombre:	Nº
------------	---------	----

5. Escribe una consulta que nos muestre la matrícula, 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%"
```




salesianos

COLEGIO NTRA. SRA. DEL PILAR ZARAGOZA

Apellidos:	Nombre:	Nº
------------	---------	----

6. 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)
```



Apellidos:	Nombre:	Nº
------------	---------	----

7. 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
)
```



Apellidos:	Nombre:	Nº
------------	---------	----

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
//
```



Apellidos:	Nombre:	Nº
------------	---------	----

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
```

```
//
```



Apellidos:	Nombre:	Nº
------------	---------	----

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 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
```

```
//
```