

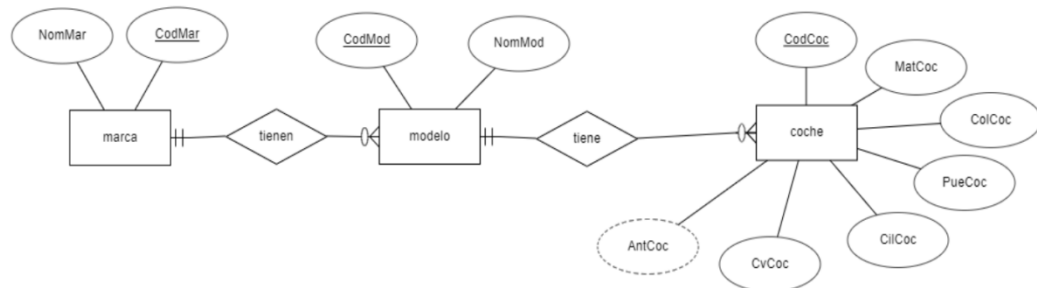
**Objetivo del ejercicio:**

1. Repasar los conceptos aprendidos en el ejercicio 1.
2. Aprender el concepto de Atributo derivado.
3. En SQL: Restricciones del CREATE TABLE (UNIQUE y DEFAULT). Tipos de DATOS TINYINT y SMALLINT. Funciones YEAR, SYSDATE y TIMESTAMPDIFF.

Diseñar una base de datos para almacenar y gestionar información de las marcas y de los modelos de vehículo de una empresa que vende coches de segunda mano. De las marcas se conoce el nombre, y de los modelos el nombre y la marca a la que pertenece. Además, el cliente desea registrar los coches que se venden en su tienda. De los coches quiere almacenar la matrícula, el color, el número de puertas, la antigüedad, los caballos de potencia y cilindrada. También quiere saber el modelo y la marca del mismo.

Se pide:

1. Modelar la base de datos. Para ello haremos:
  - a. Diseño Conceptual de Datos utilizando un Diagrama o Modelo Entidad-Relación. Lo hacemos en papel y lo pasamos a la Herramienta CASE ERD Plus.

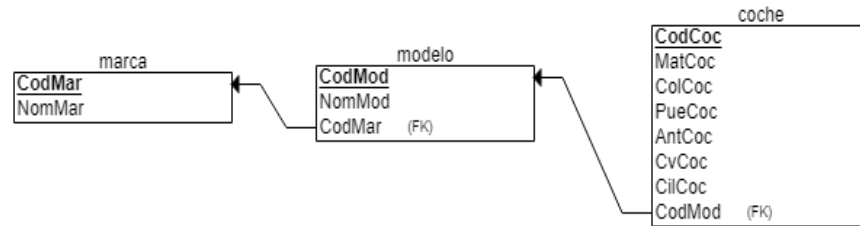


- b. Diseño Lógico de Datos utilizando un Diagrama de Estructura de datos (DED). Lo hacemos en papel y lo pasamos a la Herramienta CASE MySql Workbench. En este apartado también vamos a poner el Diagrama Referencial que genera ERD Plus a partir del Modelo Entidad-Relación.

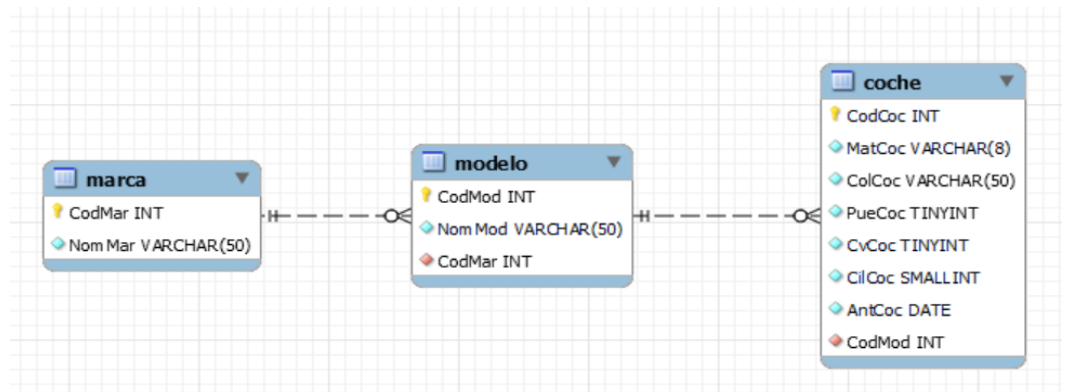
Recuerda que el Diseño Lógico de Datos es hacer el modelo relacional y para ello podemos hacer un DED o un Diagrama Referencial.

## EJERCICIO 2

**Diagrama referencial:** Observa que ERD Plus no ha puesto el atributo derivado en el modelo relacional. Se lo debes añadir tú.



**DED:**



- c. Diseño Físico de Datos. Creamos la base de datos y las tablas en SQL.

```
CREATE DATABASE ejercicio2;  
USE ejercicio2;
```

```
CREATE TABLE marca
```

```
(  
    CodMar INT NOT NULL,  
    NomMar VARCHAR(50) NOT NULL,  
    PRIMARY KEY (CodMar)  
);
```

```
CREATE TABLE modelo
```

```
(  
    CodMod INT NOT NULL,  
    NomMod VARCHAR(50) NOT NULL,  
    CodMar INT NOT NULL,  
    PRIMARY KEY (CodMod),  
    FOREIGN KEY (CodMar) REFERENCES marca(CodMar)  
);
```

## EJERCICIO 2

```
CREATE TABLE coche
(
  CodCoc INT NOT NULL,
  MatCoc VARCHAR(8) NOT NULL UNIQUE,
  ColCoc VARCHAR(50) NOT NULL DEFAULT 'blanco',
  PueCoc TINYINT NOT NULL,
  CvCoc TINYINT UNSIGNED NOT NULL,
  CilCoc SMALLINT NOT NULL,
  AntCoc DATE NOT NULL,
  CodMod INT NOT NULL,
  PRIMARY KEY (CodCoc),
  FOREIGN KEY (CodMod) REFERENCES modelo(CodMod)
);
```

Añadimos **UNIQUE** para impedir que se puedan repetir números de matrículas, aunque no sea una clave primaria (podría utilizarse como tal).

Añadimos **DEFAULT 'valor-por-defecto'** (en este caso 'blanco') para que el valor se autocomplete por defecto en caso de no añadir nada en el campo

### 2. Insertar datos desde phpmyadmin.

```
INSERT INTO marca (CodMar, NomMar)
VALUES (1, 'Audi'),
       (2, 'Citroën'),
       (3, 'Seat');
```

```
INSERT INTO modelo (CodMod, NomMod, CodMar)
VALUES (1, 'A3', 1),
       (2, 'A4', 1),
       (3, 'A6', 1),
       (4, 'C3', 2),
       (5, 'C4', 2),
       (6, 'C5', 2),
       (7, 'Ibiza', 3),
       (8, 'León', 3),
       (9, 'Arona', 3);
```

## EJERCICIO 2

```
INSERT INTO coche (CodCoc, MatCoc, ColCoc, PueCoc, AntCoc, CvCoc, CilCoc, CodMod)
VALUES (1, '1234AAA', 'rojo', 5, '1995-09-30', 110, 1500, 8),
      (2, '1234AAB', DEFAULT, 11, '1998/03/15', 90, 1400, 4),
      (3, '1234AAC', 'amarillo', 3, '1999/12/07', 80, 2200, 7),
      (4, '1234AAD', 'gris', 5, '2008/05/13', 210, 1600, 1);
```

Las fechas se pueden introducir con formato 'YYYY-MM-DD' o 'YYYY/MM/DD'.

El valor que quede como DEFAULT se autocompletará como 'blanco'.

### 3. Realizar las siguientes consultas en SQL:

- Muestra todas las filas y todos los campos de las tablas:

```
SELECT *
FROM marca
```

```
SELECT *
FROM modelo
```

- Muestra algunos campos de las tablas.

```
SELECT NomMar, CodMar
FROM marca
```

```
SELECT NomMod, CodMod
FROM modelo
```

- Muestra los modelos con una marca menor que 3

```
SELECT *
FROM modelo
WHERE CodMar<3
```

- Muestra el nombre de las marcas junto con el nombre de los modelos que tiene.

```
SELECT NomMar, NomMod
FROM marca ma INNER JOIN modelo mo ON (ma.CodMar=mo.CodMar)
```

- Muestra el nombre de los modelos junto con el nombre de su marca.

```
SELECT NomMod, NomMar
FROM marca ma INNER JOIN modelo mo ON (ma.CodMar=mo.CodMar)
```

## EJERCICIO 2

- Para la marca Audi, mostrar todos sus modelos.

```
SELECT NomMod
FROM marca ma INNER JOIN modelo mo ON (ma.CodMar=mo.CodMar)
WHERE NomMar='Audi'
```

- Para el modelo A6, indicar cuál es su marca.

```
SELECT NomMar
FROM marca ma INNER JOIN modelo mo ON (ma.CodMar=mo.CodMar)
WHERE NomMod='A6'
```

- Obtener los datos de todos los coches mostrando también el nombre del modelo y de la marca que tiene. Mostrar la antigüedad, no la fecha en la que se matriculó por primera vez.

```
SELECT      NomMar,      NomMod,      MatCoc,      ColCoc,      PueCoc,
YEAR(SYSDATE())-YEAR(AntCoc), CvCoc, CilCoc
FROM (marca ma INNER JOIN modelo mo ON (ma.CodMar=mo.CodMar))
INNER JOIN coche co ON (mo.CodMod=co.CodMod)
```

*La función SYSDATE() nos proporciona la fecha y hora actual. También podríamos haber utilizado la función CURDATE().*

*La función YEAR() obtiene el año a partir de una fecha (o fecha y hora).*

*Al unir YEAR(SYSDATE()) me da el año de la fecha actual.*

*YEAR(AntCoc) me da el año de la fecha de matriculación del coche.*

*La función TIMESTAMPDIFF, nos da la diferencia entre dos fechas con más exactitud. Por Ejemplo:*

```
SELECT      TIMESTAMPDIFF(YEAR,      '2022-10-21',      '2023-10-20')      AS
años_transcurridos;
```

*Esta sentencia me devolvería un 0, pues aún no ha hecho un año.*

```
SELECT      TIMESTAMPDIFF(YEAR,      '2022-10-21',      '2023-10-21')      AS
años_transcurridos;
```

*Esta otra me devolvería 1. Indicando que ha pasado un año entre las dos fechas.*