

Base de Datos

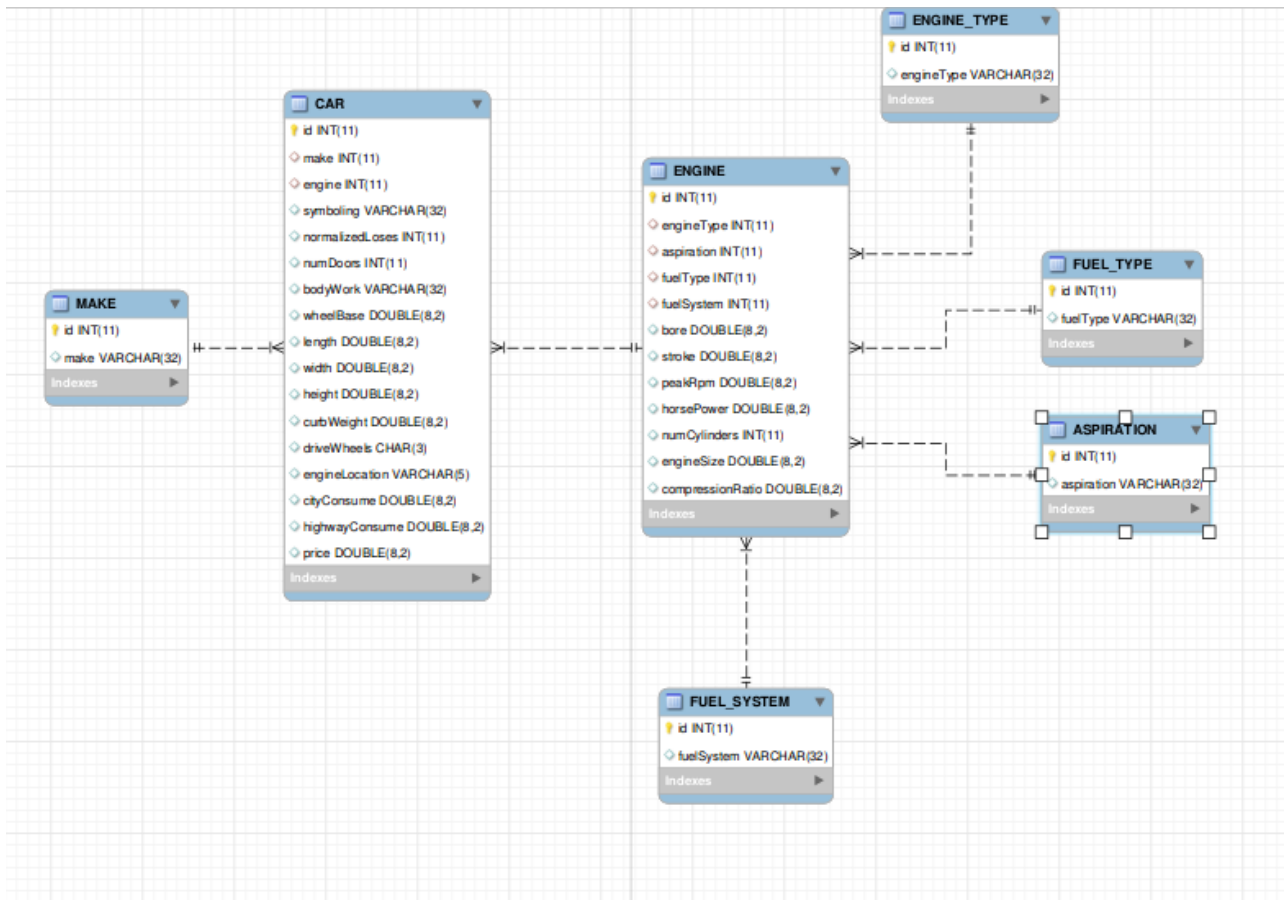
Automobile integration procedures

Sergio López Pérez

Índice

1. Data model.....	3
1.1 Justification of the data model..	4
2. Description of the procedures and fucntions.....	4
2.1 Procedures.....	4
2.2 Functions.....	6
3. Query tests.....	8
4. Anotaciones.....	10

1. Data model



1.1 Justification of the data model

He escogido este modelo a partir de unos cuantos base, en el que incluía una tabla de carrocería, con todas las medidas de coche, además del estilo de la carrocería (`bodyStyle`). También tenía la tabla de motor sin tablas extra.

A partir de este modelo me di cuenta de que la tabla de motor quedaba demasiado espesa como para poder manejarla a gusto, por tanto hacia falta dividirla en subtablas. Para ello elegí las tablas que tenían valores por defecto, y con menos filas para hacer las subtablas, haciendo motor mas manejable.

Luego me deshice de la tabla de carrocería, ya que no era del todo claro, porque realmente los coches pueden variar en sus medidas según el lugar, por lo que podría dar problemas a la hora de manejar los datos y las foreign key, por tanto opte por agrandar la tabla principal.

2. Description of the procedures and fucntions.

2.1 Procedures

load_make(): Este procedure carga la tabla de marca, después de borrarla si existe. El cursor esta preparado para solo usar la información que no esta ya en la tabla de MAKE.

load_fuelSystem(): Este procedure carga los sistemas de combustible en FUEL_SYSTEM, después de borrarlo si es que existe.
El cursor está preparado para solo usar la información que no este ya en la tabla FUEL_SYSTEM.

load_engineType(): Este carga los tipos de motor en ENGINE_TYPE, después de comprobar si existe ya.
El cursor esta preparado para que solo se use la información que no esta ya en ENGINE_TYPE.

load_aspiration(): Este procedure carga la información de la aspiración del motor en la tabla ASPIRATION, después de borrarlo si ya existe.

load_fuelType(): Este procedure carga la información de los tipos de combustible a la tabla FUEL_TYPE, después de borrarlo si ya existe.

El cursor esta preparado para solo usar la información de la tabla temporal que no esta ya en FUEL_TYPE.

load_engine(): Este procedure carga la información de los motores, después de borrarlo si ya existe, a partir de la tabla temporal y las subtablas de motor. En este caso el cursor coge siempre la información de toda la tabla temporal respecto al motor, y sus subtablas, por lo que si se llama dos veces el procedure, se duplicaría la información.

load_car(): Este procedure carga la información en CAR, después de borrarlo si ya existe, a partir de la tabla temporal, MAKE y ENGINE.

El cursor coge toda la información de todas las tablas, por lo que si se llama dos veces al procedure, se duplicará la información.

Cabe decir también, que debido a los valores nulos de la tabla temporal, ha sido necesario hacer varias comparaciones entre ellos, ya que si no lo hacía, el cursor mostraba 6 coches menos.

load_database(): Finalmente este procedure carga el resto de procedures, para llenar la base de datos de coches.

Para cargar la base basta con usar:

```
call load_database;
```

2.2 Functions

```
delimiter $$
create function fiabilidad (sym int) returns varchar(32)
begin

    declare v_sym int;
    declare v_fiab varchar(32);

    set v_sym = sym;

    if v_sym = -3 then
        set v_fiab = 'Excelente';
    elseif v_sym = -2 then
        set v_fiab = 'Muy fiable';
    elseif v_sym = -1 then
        set v_fiab = 'Fiable';
    elseif v_sym = 0 then
        set v_fiab = 'Mejorable';
    elseif v_sym = 1 then
        set v_fiab = 'Baja calidad';
    elseif v_sym = 2 then
        set v_fiab = 'Mala calidad';
    elseif v_sym = 3 then
        set v_fiab = 'Peligroso';
    end if;

    return v_fiab;

end;
$$
delimiter ;
```

Esta función recoge los valores de symboling, la fiabilidad del coche, y los convierte de un valor numérico a una cadena de texto más comprensible.

```

drop function if exists inchesToCm;
delimiter $$
create function inchesToCm (inches double) returns double(8, 2)
begin
    declare v_inch double;
    declare v_m double;

    set v_inch = inches;

    set v_m = v_inch/0.39370;

    return round(v_m, 2);
end;
$$
delimiter ;

```

Esta función cambia los valores de pulgadas del sistema imperial al centímetros del sistema internacional.

```

delimiter $$
create function poundsToKg (pound int) returns double(8, 2)
begin
    declare v_pound int;
    declare v_kg double;

    set v_pound = pound;

    set v_kg = v_pound/2.2046;

    return round(v_kg, 2);
end;
$$
delimiter ;

```

Esta función cambia los valores de libras (peso) a kilogramos del sistema internacional.

```

delimiter $$
create function cubicInchToCm (inch int) returns double(8, 2)
begin
    declare v_inch int;
    declare v_cm double;

    set v_inch = inch;

    set v_cm = v_inch / 0.061024;

    return round(v_cm, 2);

end;
$$
delimiter ;

```

Esta función cambia los valores de las pulgadas cubicas a centímetros cúbicos del sistemas internacional.

```

delimiter $$
create function mpgToL (mpg int) returns double(8, 2)
begin
    declare v_millas int;
    declare v_litros double;

    set v_millas = mpg;

    set v_litros = 235.21 / v_millas;

    return round(v_litros, 2);

end;
$$
delimiter ;

```

Esta función cambia los valores de consumo del vehículo de millas por galón a litros por cada 100KM.


```

delimiter $$
create function dolarToEuro (dolar int) returns double(8, 2)
]begin
    declare v_dolar int;
    declare v_euro double;

    set v_dolar = dolar;

    set v_euro = v_dolar / 1.231838;

    return round(v_euro, 2);
end;
$$
delimiter ;

```

Esta función cambia de dólares americanos a euros. El valor de esta función puede cambiar con el tiempo por el cambio de divisas.

3. Query tests

1 -- Show every car and its engine of alfa-romeo
2 • select
3 CAR.id, MAKE.make, ENGINE_TYPE.engineType, ASPIRATION.aspiration, FUEL_TYPE.fuelType, FUEL_SYSTEM.fuelSystem,
4 peakRpm, horsepower, numCylinders, engineSize, curbWeight, length, width, height
5
6 from
7 ENGINE,
8 CAR,
9 MAKE,
10 FUEL_SYSTEM,
11 FUEL_TYPE,
12 ASPIRATION,
13 ENGINE_TYPE
14 where
15 ENGINE.id = CAR.engine and
16 CAR.make = MAKE.id and
17 ENGINE.fuelSystem = FUEL_SYSTEM.id and
18 ENGINE.fuelType = FUEL_TYPE.id and
19 ENGINE.aspiration = ASPIRATION.id and
20 ENGINE.engineType = ENGINE_TYPE.id and
21 MAKE.make = 'alfa-romeo';
22
23

Result Grid Filter Rows: Export: Wrap Cell Content:

#	id	make	engineType	aspiration	fuelType	fuelSystem	peakRpm	horsePower	numCylinders	engineSize	curbWeight	length	width	height
1	1	alfa-romeo	dohc	std	gas	mpfi	5000.00	111.00	4	2130.31	1155.77	428.75	162.81	123.95
2	2	alfa-romeo	dohc	std	gas	mpfi	5000.00	111.00	4	2130.31	1155.77	428.75	162.81	123.95
3	3	alfa-romeo	ohcv	std	gas	mpfi	5000.00	154.00	6	2490.82	1280.50	434.85	166.37	133.10

Query que devuelve información acerca de todos los coches de la marca alfa romeo.

```

26 • select
27     CAR.id, MAKE.make, ENGINE_TYPE.engineType, ASPIRATION.aspiration, FUEL_TYPE.fuelType, FUEL_SYSTEM.fuelSystem,
28     peakRpm, horsepower, numCylinders, engineSize, curbWeight, length, width, height
29
30 from
31     ENGINE,
32     CAR,
33     MAKE,
34     FUEL_SYSTEM,
35     FUEL_TYPE,
36     ASPIRATION,
37     ENGINE_TYPE
38 where
39     ENGINE.id = CAR.engine and
40     CAR.make = MAKE.id and
41     ENGINE.fuelSystem = FUEL_SYSTEM.id and
42     ENGINE.fuelType = FUEL_TYPE.id and
43     ENGINE.aspiration = ASPIRATION.id and
44     ENGINE.engineType = ENGINE_TYPE.id and
45     FUEL_TYPE.fuelType = 'diesel';

```

#	id	make	engineType	aspiration	fuelType	fuelSystem	peakRpm	horsePower	numCylinders	engineSize	curbWeight	length	width	height
1	64	mazda	ohc	std	diesel	idi	4650.00	64.00	4	1999.21	1108.14	451.61	168.91	140.97
2	67	mazda	ohc	std	diesel	idi	4200.00	72.00	4	2195.86	1224.71	444.50	167.89	138.18
3	91	nissan	ohc	std	diesel	idi	4800.00	55.00	4	1687.86	914.91	419.86	162.05	138.43
4	159	toyota	ohc	std	diesel	idi	4500.00	56.00	4	1802.57	1031.93	422.40	163.58	134.62
5	160	toyota	ohc	std	diesel	idi	4500.00	56.00	4	1802.57	1031.93	422.40	163.58	134.11
6	183	volkswagen	ohc	std	diesel	idi	4800.00	52.00	4	1589.54	1025.58	436.12	166.37	141.48
7	185	volkswagen	ohc	std	diesel	idi	4800.00	52.00	4	1589.54	1026.94	436.12	166.37	141.48
8	68	mercedes-benz	ohc	turbo	diesel	idi	4350.00	123.00	5	2998.82	1594.39	484.89	178.56	143.51
9	69	mercedes-benz	ohc	turbo	diesel	idi	4350.00	123.00	5	2998.82	1700.99	484.89	178.56	149.10
10	70	mercedes-benz	ohc	turbo	diesel	idi	4350.00	123.00	5	2998.82	1585.32	476.25	178.56	139.45
11	71	mercedes-benz	ohc	turbo	diesel	idi	4350.00	123.00	5	2998.82	1710.06	514.61	182.12	143.00

Query que devuelve información de los coches diesel.

```

47 -- show the cars consuming more than 13 liters
48
49 • select
50     CAR.id, MAKE.make, ENGINE_TYPE.engineType, ASPIRATION.aspiration, FUEL_TYPE.fuelType, FUEL_SYSTEM.fuelSystem,
51     peakRpm, horsepower, numCylinders, engineSize, curbWeight, length, width, height
52
53 from
54     ENGINE,
55     CAR,
56     MAKE,
57     FUEL_SYSTEM,
58     FUEL_TYPE,
59     ASPIRATION,
60     ENGINE_TYPE
61 where
62     ENGINE.id = CAR.engine and
63     CAR.make = MAKE.id and
64     ENGINE.fuelSystem = FUEL_SYSTEM.id and
65     ENGINE.fuelType = FUEL_TYPE.id and
66     ENGINE.aspiration = ASPIRATION.id and
67     ENGINE.engineType = ENGINE_TYPE.id and
68     highwayConsume > 13;
69

```

#	id	make	engineType	aspiration	fuelType	fuelSystem	peakRpm	horsePower	numCylinders	engineSize	curbWeight	length	width	height
1	50	jaguar	ohcv	std	gas	mpfi	5000.00	262.00	12	5342.16	1791.71	486.92	179.32	121.41
2	72	mercedes-benz	ohcv	std	gas	mpfi	4750.00	155.00	8	3834.56	1696.45	514.61	182.12	143.51
3	73	mercedes-benz	ohcv	std	gas	mpfi	4750.00	155.00	8	3834.56	1671.51	457.96	179.07	129.03
4	74	mercedes-benz	ohcv	std	gas	mpfi	4500.00	184.00	8	5047.19	1769.03	528.58	182.12	144.02
5	75	mercedes-benz	ohcv	std	gas	mpfi	4500.00	184.00	8	4981.65	1685.11	505.97	182.88	140.72

Query que devuelve información de los coches que consumen mas de 13 litros por 100KM en carretera.

4. Anotaciones

Los nulls de la tabla temporal eran originalmente -1 (excepto los de la columna de symboling) importados directamente del .csv, los cuales he cambiado con varios updates de la tabla temporal.

He modificado la tabla temporal, añadiéndole una columna de id como primary key, para poder borrarla fila a fila en el procedure load_car().