

# **Base de Datos 2**

## **Obligatorio**



Guzmán López Orrego  
205873

M3A  
Fernando Martínez  
10 de julio de 2017

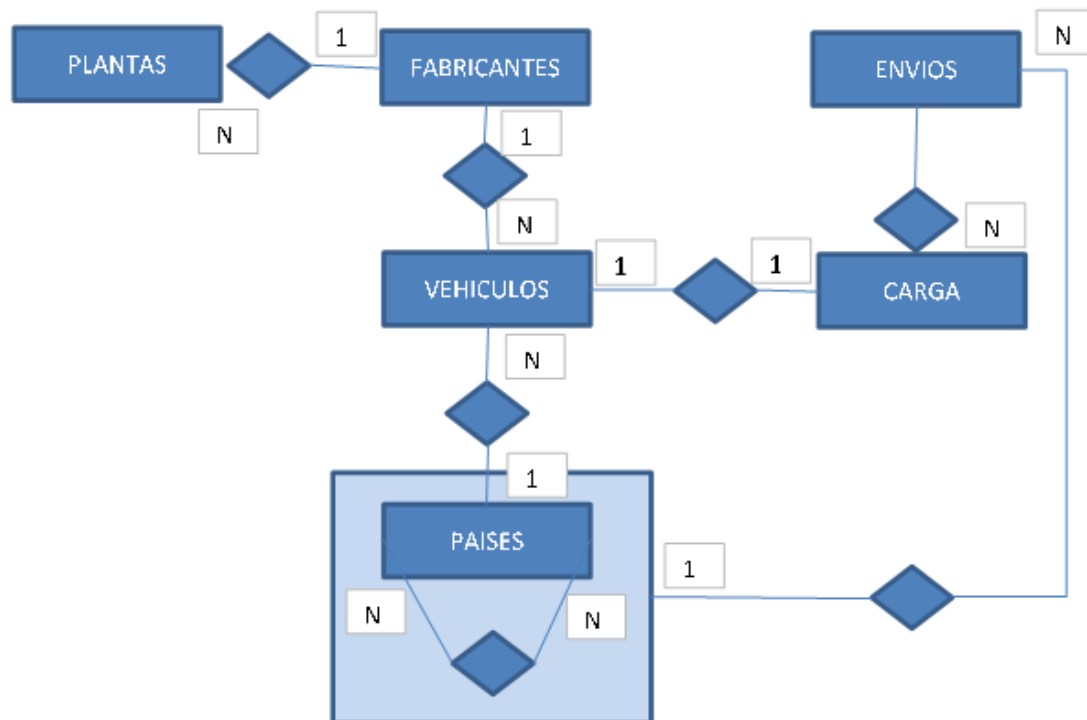
## Índice

Base de Datos 2.....	1
Obligatorio.....	1
Índice.....	3
Problema: empresa ROVIA INC.....	5
1) Crear las restricciones de integridad que surjan del análisis de la letra, sobre el script de creación de tablas proporcionado (mediante ALTER de las estructuras dadas).....	5
2) Creación de índices que considere puedan ser útiles para optimizar las consultas (según criterio establecido en el curso).....	8
3) Ingreso de un juego completo de datos de prueba (será más valorada la calidad de los datos más que la cantidad. El mismo debería incluir ejemplos que deban ser rechazados por no cumplir con las restricciones implementadas).....	9
Punto 4).....	31
4 a) Crear una función almacenada que reciba como parámetro un número de VIN completo y valide el dígito verificador, si dicho dígito es correcto la función debe retornar el texto 'OK', si es incorrecto, debe retornar todo el número de VIN pero incluyendo el dígito verificador correcto.....	31
4 b) Crear una función que reciba como parámetro un VIN y retorne el año del modelo de dicho vehículo.....	35
4 c) Dado un rango de fechas, retornar la cantidad de vehículos que recibió el país que más vehículos recibió en dicho rango.....	37
4 d) Mediante un procedimiento almacenado que reciba un rango de fechas de envío, cambiar el país de destino del envío a todos los vehículos cuyo país de fabricación del vehículo sea el mismo que el país de destino del envío, en ese caso crear un país de código '#' que tenga como descripción 'LOCAL' y poner ese dato como destino.....	38
4 e) Dado un rango de fechas, implementar una función que retorne el nombre del país al cual se le envió la menor cantidad de vehículos en dicho rango.....	39
4 f) Realizar un procedimiento almacenado que reciba un código de fabricante y retorne con parámetro de salida la cantidad de plantas que tiene y la cantidad de modelos de vehículos diferentes que produce.....	40
Punto 5).....	41
5 a) Crear un trigger que valide un número de VIN al ingresar un nuevo vehículo, debe utilizar la función creada en el punto 4a.....	41
5 b) Crear un trigger al ingresar una línea de carga, ponga el peso correcto de acuerdo al peso del vehículo mas el 5% de combustible y accesorios, también debe acumular el peso total del envío.....	45
5 c) Definir un trigger que al ingresar un envío, si el país de origen es igual al país de destino no procese la línea.....	47
5 d) Implementar un trigger que permita el borrado de uno o más envíos, para ello debe borrar en cascada todos los registros en las tablas relacionadas, se debe tener en cuenta los borrados múltiples.....	47
Punto 6).....	48

6 a) Mostrar el mayor peso, el menor peso, la mayor fecha y la menor fecha de los envíos de los años 2015 y 2016.....	48
6 b) Mostrar para cada fabricante su nombre, la cantidad de vehículos enviados y el peso total de los mismos en el año 2016 (ordenar los resultados de mayor peso a menor peso).....	49
6 c) Para todos los países que fueron destino de envíos, mostrar su nombre, su cantidad de envíos en los 20 primeros días de enero de 2016 y la fecha del último envío realizado, si algún país nunca fue destino de envíos igual debe mostrar su nombre.....	49
6 d) Mostrar los datos de los fabricantes que tengan envíos con más de 500 vehículos y que no tengan más de 3 envíos con menos de 100 vehículos.....	50
6 e) Obtener todos los datos de los vehículos que fueron enviados en la fecha más reciente.....	51
6 f) Mostrar los datos de los fabricantes que no han hecho envíos en el primer semestre del 2016, registraron algún envío en el 2017 y solo han hecho envíos con destino Holanda.....	51
6 g) Para cada vehículo con menos de 2.300 kg de peso, mostrar sus datos, la fecha del último envío donde participó, y el nombre del fabricante.....	52
6 h) Bajar el peso en un 5% a todos los vehículos que no nunca fueron enviados pasado 1 año de su fabricación (puede utilizar la función 4b).....	52
6 i) Obtener los datos de los fabricantes de Japón que en el 2016 hayan tenido en algún mes un envío de cantidad total superior a 100 vehículos y que también hayan tenido algún mes con una cantidad total inferior a 20.....	52
7) Realizar una vista que muestre País de Fabricación del vehículo y cantidad de vehículos enviados a Países diferentes a los de la fábrica.....	53
Anexo.....	54

## Problema: empresa ROVIA INC

La empresa de transporte marítimo de vehículos ROVIA INC tiene un sistema de gestión modelado de la siguiente manera:



**Países**(codPais, nomPais)

**Fabricantes**(codFab, nomFab, dirFab, mailFab)

**Plantas**(codPlan, codFab, nomPlan, dirPlan, mailPlan, *codPais*)

**Vehículos**(vin, modelo, color, peso, características, *codPais*, *codFab*)

**Envios**(idEnvio, fchEnvio, pesoEnvio, *oriEnvio*, *desEnvio*)

**Carga**(idEnvio, idCarga, *vin*, pesoCarga)

**1) Crear las restricciones de integridad que surjan del análisis de la letra, sobre el script de creación de tablas proporcionado (mediante ALTER de las estructuras dadas).**

### Script de creación de tablas dado:

```
/* Creacion de la BD */
CREATE DATABASE BD_VEHICULOS
go
USE BD_VEHICULOS
go
/* Creacion de tablas */
CREATE TABLE Paises(codPais character(1) not null,
                     nomPais varchar(30) not null)
go
CREATE TABLE Fabricantes(codFab character(2) not null,
                          nomFab varchar(30) not null,
                          dirFab varchar(30) not null,
                          mailFab varchar(50))
go
CREATE TABLE Plantas(codPlan numeric(6) not null,
                      codFab character(2) not null,
                      nomPlan varchar(30) not null,
                      dirPlan varchar(30) not null,
                      mailPlan varchar(50),
                      codPais character(1) not null)
go
CREATE TABLE Vehiculos(vin character(17) not null,
                        modelo varchar(30) not null,
                        color varchar(20) not null,
                        peso numeric(12,2),
                        características varchar(100),
                        codPais character(1) not null,
                        codFab character(2) not null)
go
CREATE TABLE Envios(idEnvio numeric(6) identity(1,1) not null,
                     fchEnvio datetime,
                     pesoEnvio numeric(12,2),
                     oriEnvio character(1) not null,
                     desEnvio character(1) not null)
go
CREATE TABLE Carga(idEnvio numeric(6) not null,
                    idCarga numeric(6) not null,
                    vin character(17) not null,
                    pesoCarga numeric(12,2))
go
```

### Restricciones de integridad:

```
/*
*****
* Script con las restricciones de integridad creadas sobre el script de
* creación de tablas, índices, y el ingreso de datos de prueba.
*****
*/
USE BD_VEHICULOS;
go
/*
*****
* Creación de Restricciones de Integridad
*****
*/
```

```

*/
-- Tabla Paises
ALTER TABLE Paises
ADD CONSTRAINT const_Paises_pk_codPais
PRIMARY KEY(codPais);
go
ALTER TABLE Paises
ADD CONSTRAINT const_Paises_uniq_nomPais
UNIQUE(nomPais);
go
-- Tabla Fabricantes
ALTER TABLE Fabricantes
ADD CONSTRAINT const_Fabricantes_pk_codFab
PRIMARY KEY(codFab);
go
ALTER TABLE Fabricantes
ADD CONSTRAINT const_Fabricantes_uniq_mailFab
UNIQUE(mailFab);
go
ALTER TABLE Fabricantes
ADD cantEmp INT NOT NULL;
go
ALTER TABLE Fabricantes
ADD CONSTRAINT const_Fabricantes_chk_cantEmp
CHECK (cantEmp > 0);
go
-- Tabla Plantas
ALTER TABLE Plantas
ADD CONSTRAINT const_Plantas_pk_codPlan_codFab
PRIMARY KEY(codPlan, codFab);
go
ALTER TABLE Plantas
ADD CONSTRAINT const_Plantas_fk_codFab
FOREIGN KEY(codFab) REFERENCES Fabricantes(codFab);
go
ALTER TABLE Plantas
ADD CONSTRAINT const_Plantas_fk_codPais
FOREIGN KEY(codPais) REFERENCES Paises(codPais);
go
ALTER TABLE Plantas
ADD CONSTRAINT const_Plantas_uniq_mailPlan
UNIQUE(mailPlan);
go
-- Tabla Vehiculos
ALTER TABLE Vehiculos
ADD CONSTRAINT const_Vehiculos_pk_vin
PRIMARY KEY(vin);
go
ALTER TABLE Vehiculos
ADD CONSTRAINT const_Vehiculos_fk_codPais
FOREIGN KEY(codPais) REFERENCES Paises(codPais);
go
ALTER TABLE Vehiculos
ADD CONSTRAINT const_Vehiculos_fk_codFab
FOREIGN KEY(codFab) REFERENCES Fabricantes(codFab);
go
-- Tabla Envios
ALTER TABLE Envios
ADD CONSTRAINT const_Envios_pk_idEnvio
PRIMARY KEY(idEnvio);

```

```

go
ALTER TABLE Envios
ADD CONSTRAINT const_Envios_fk_oriEnvio
FOREIGN KEY(oriEnvio) REFERENCES Países(codPaís);
go
ALTER TABLE Envios
ADD CONSTRAINT const_Envios_fk_desEnvio
FOREIGN KEY(desEnvio) REFERENCES Países(codPaís);
go
-- Tabla Carga
ALTER TABLE Carga
ADD CONSTRAINT const_Carga_pk_idEnvio_idCarga
PRIMARY KEY(idEnvio, idCarga);
go
ALTER TABLE Carga
ADD CONSTRAINT const_Carga_fk_idEnvio
FOREIGN KEY(idEnvio) REFERENCES Envios(idEnvio);
go
ALTER TABLE Carga
ADD CONSTRAINT const_Carga_fk_vin
FOREIGN KEY(vin) REFERENCES Vehículos(vin);
go

```

## 2) Creación de índices que considere puedan ser útiles para optimizar las consultas (según criterio establecido en el curso).

```

/*
*****
* Creación de Índices
*****
*/
-- Tabla Plantas
CREATE INDEX idx_Plantas_codFab
ON Plantas(codFab);
go
CREATE INDEX idx_Plantas_codPaís
ON Plantas(codPaís);
go
-- Tabla Vehículos
CREATE INDEX idx_Vehículos_codPaís
ON Vehículos(codPaís);
go
CREATE INDEX idx_Vehículos_codFab
ON Vehículos(codFab);
go
-- Tabla Envios
CREATE INDEX idx_Envios_oriEnvio
ON Envios(oriEnvio);
go
CREATE INDEX idx_Envios_desEnvio
ON Envios(desEnvio);
go
-- Tabla Carga
CREATE INDEX idx_Carga_vin
ON Carga(vin);
go

```



**3) Ingreso de un juego completo de datos de prueba (será más valorada la calidad de los datos más que la cantidad. El mismo debería incluir ejemplos que deban ser rechazados por no cumplir con las restricciones implementadas.**

**Todos los datos ingresados fueron previamente validados. En los casos que se ingresaron pesos, a estos se les incorporó el 5% y para los envíos se sumo el peso total. Los VIN ingresados también fueron previamente validados. Con el objetivo de no obtener consultas vacías en el punto 6 se ingresaron datos específicos. Existen pruebas sobre las restricciones donde se prueban las mismas y se esperan que las salidas sean de error.**

```
/*
*****
* Ingreso de datos de prueba: OK
*****
*/
-- *****
-- Tabla Paises
-- *****
go
INSERT INTO Paises(codPais, nomPais)
VALUES('1', 'Estados Unidos');
go
INSERT INTO Paises(codPais, nomPais)
VALUES('2', 'Canadá');
go
INSERT INTO Paises(codPais, nomPais)
VALUES('3', 'México');
go
INSERT INTO Paises(codPais, nomPais)
VALUES('9', 'Brasil');
go
INSERT INTO Paises(codPais, nomPais)
VALUES('J', 'Japón');
go
INSERT INTO Paises(codPais, nomPais)
VALUES('K', 'Corea');
go
INSERT INTO Paises(codPais, nomPais)
VALUES('S', 'Inglaterra');
go
INSERT INTO Paises(codPais, nomPais)
VALUES('W', 'Alemania');
go
INSERT INTO Paises(codPais, nomPais)
VALUES('Y', 'Suecia');
go
INSERT INTO Paises(codPais, nomPais)
VALUES('Z', 'Italia');
go
-- Otros países
INSERT INTO Paises(codPais, nomPais)
VALUES('4', 'Argentina');
```

```

go
INSERT INTO Paises(codPais, nomPais)
VALUES('5', 'Chile');
go
INSERT INTO Paises(codPais, nomPais)
VALUES('6', 'Uruguay');
go
-- *****
-- Tabla Fabricantes
-- *****
INSERT INTO Fabricantes(codFab, nomFab, dirFab, mailFab, cantEmp)
VALUES('AA', 'Audi', '12 The Horseshoes, IP31', 'audi_factory1@audi.com', 120);
go
INSERT INTO Fabricantes(codFab, nomFab, dirFab, mailFab, cantEmp)
VALUES('AB', 'Audi', '14 The Horseshoes, IP31', 'audi_factory2@audi.com', 120);
go
INSERT INTO Fabricantes(codFab, nomFab, dirFab, mailFab, cantEmp)
VALUES('BA', 'Bmw', '7 Copley St, Batley WF17 8JA', 'bmw_factory1@bmw.com',
100);
go
INSERT INTO Fabricantes(codFab, nomFab, dirFab, mailFab, cantEmp)
VALUES('BB', 'Bmw', '8 Copley St, Batley WF17 8JA', 'bmw_factory2@bmw.com',
100);
go
INSERT INTO Fabricantes(codFab, nomFab, dirFab, mailFab, cantEmp)
VALUES('CA', 'Chrysler', '36 Trewsbury Rd, 5DN',
'chrysler_factory1@chrysler.com', 140);
go
INSERT INTO Fabricantes(codFab, nomFab, dirFab, mailFab, cantEmp)
VALUES('CB', 'Chrysler', '38 Trewsbury Rd, 5DN',
'chrysler_factory2@chrysler.com', 140);
go
INSERT INTO Fabricantes(codFab, nomFab, dirFab, mailFab, cantEmp)
VALUES('DA', 'Mercedes Benz', '44 Jean Armour Dr, 2SD',
'mercedesbenz_factory1@mercedesbenz.com', 570);
go
INSERT INTO Fabricantes(codFab, nomFab, dirFab, mailFab, cantEmp)
VALUES('DB', 'Mercedes Benz', '46 Jean Armour Dr, 2SD',
'mercedesbenz_factory2@mercedesbenz.com', 570);
go
INSERT INTO Fabricantes(codFab, nomFab, dirFab, mailFab, cantEmp)
VALUES('EA', 'Dodge', '4 Lon Ger-Y-Coed, Ammanford 18',
'dodge_factory1@dodge.com', 80);
go
INSERT INTO Fabricantes(codFab, nomFab, dirFab, mailFab, cantEmp)
VALUES('EB', 'Dodge', '3 Lon Ger-Y-Coed, Ammanford 18',
'dodge_factory2@dodge.com', 80);
go
INSERT INTO Fabricantes(codFab, nomFab, dirFab, mailFab, cantEmp)
VALUES('FA', 'Ford', '2 Boston End, Thetford IP24', 'ford_factory1@ford.com',
160);
go
INSERT INTO Fabricantes(codFab, nomFab, dirFab, mailFab, cantEmp)
VALUES('FB', 'Ford', '4 Boston End, Thetford IP24', 'ford_factory2@ford.com',
260);
go
INSERT INTO Fabricantes(codFab, nomFab, dirFab, mailFab, cantEmp)
VALUES('GA', 'GM', '1-4 Smithy Row, NG1', 'gm_factory1@gm.com', 450);
go
INSERT INTO Fabricantes(codFab, nomFab, dirFab, mailFab, cantEmp)

```

```

VALUES('GB', 'GM', '1 Smithy Row, NG1', 'gm_factory2@gm.com', 450);
go
INSERT INTO Fabricantes(codFab, nomFab, dirFab, mailFab, cantEmp)
VALUES('HA', 'Honda', '26A Station St, 5HJ', 'honda_factory1@honda.com', 320);
go
INSERT INTO Fabricantes(codFab, nomFab, dirFab, mailFab, cantEmp)
VALUES('HB', 'Honda', '25 Station St, 5HJ', 'honda_factory2@honda.com', 300);
go
INSERT INTO Fabricantes(codFab, nomFab, dirFab, mailFab, cantEmp)
VALUES('JA', 'Jaguar', '15 Sutton Ave, 3JY', 'jaguar_factory1@jaguar.com',
620);
go
INSERT INTO Fabricantes(codFab, nomFab, dirFab, mailFab, cantEmp)
VALUES('JB', 'Jaguar', '20 Sutton Ave, 3K', 'jaguar_factory2@jaguar.com', 320);
go
INSERT INTO Fabricantes(codFab, nomFab, dirFab, mailFab, cantEmp)
VALUES('LA', 'Lincon', '9-15 Queen St, Cardiff CF10',
'lincon_factory1@lincon.com', 220);
go
INSERT INTO Fabricantes(codFab, nomFab, dirFab, mailFab, cantEmp)
VALUES('LB', 'Lincon', '20 Queen St, Cardiff CF1',
'lincon_factory2@lincon.com', 220);
go
INSERT INTO Fabricantes(codFab, nomFab, dirFab, mailFab, cantEmp)
VALUES('MA', 'Mercury', '375 Lonsdale Rd, 9PY', 'mercury_factory1@mercury.com',
460);
go
INSERT INTO Fabricantes(codFab, nomFab, dirFab, mailFab, cantEmp)
VALUES('MB', 'Mercury', '300 Lonsdale Rd, 9PY', 'mercury_factory2@mercury.com',
460);
go
INSERT INTO Fabricantes(codFab, nomFab, dirFab, mailFab, cantEmp)
VALUES('NA', 'Nissan', 'Saltcoats, 1YZ', 'nissan_factory1@nissan.com', 340);
go
INSERT INTO Fabricantes(codFab, nomFab, dirFab, mailFab, cantEmp)
VALUES('NB', 'Nissan', 'Saltcoats, 20YZ', 'nissan_factory2@nissan.com', 340);
go
INSERT INTO Fabricantes(codFab, nomFab, dirFab, mailFab, cantEmp)
VALUES('PA', 'Plymouth', '21 Station Rd Blvd, 5SN',
'plymouth_factory1@plymouth.com', 380);
go
INSERT INTO Fabricantes(codFab, nomFab, dirFab, mailFab, cantEmp)
VALUES('PB', 'Plymouth', '18 Station Rd Blvd, 5SN',
'plymouth_factory2@plymouth.com', 380);
go
INSERT INTO Fabricantes(codFab, nomFab, dirFab, mailFab, cantEmp)
VALUES('SA', 'Saab', 'A9, Bridge of Allan, FK9', 'saab_factory1@saab.com',
180);
go
INSERT INTO Fabricantes(codFab, nomFab, dirFab, mailFab, cantEmp)
VALUES('SB', 'Saab', 'Bridge of Allan, FK9', 'saab_factory2@saab.com', 180);
go
INSERT INTO Fabricantes(codFab, nomFab, dirFab, mailFab, cantEmp)
VALUES('TA', 'Toyota', '12 Tayside Dr, Edgward 8RD',
'toyota_factory1@toyota.com', 240);
go
INSERT INTO Fabricantes(codFab, nomFab, dirFab, mailFab, cantEmp)
VALUES('TB', 'Toyota', '10 Tayside Dr, Edgward 8RD',
'toyota_factory2@toyota.com', 240);
go

```

```

INSERT INTO Fabricantes(codFab, nomFab, dirFab, mailFab, cantEmp)
VALUES('VA', 'Volkswagen', '21 Lon Tyr Haul, 9SF',
'volkswagen_factory1@volkswagen.com', 570);
go
INSERT INTO Fabricantes(codFab, nomFab, dirFab, mailFab, cantEmp)
VALUES('VB', 'Volkswagen', '31 Lon Tyr Haul, 9',
'volkswagen_factory2@volkswagen.com', 570);
go
INSERT INTO Fabricantes(codFab, nomFab, dirFab, mailFab, cantEmp)
VALUES('2A', 'Pontiac', '9 Kingsgate St, 9PD', 'pontiac_factory1@pontiac.com',
250);
go
INSERT INTO Fabricantes(codFab, nomFab, dirFab, mailFab, cantEmp)
VALUES('2B', 'Pontiac', '10 Kingsgate St, 9PD', 'pontiac_factory2@pontiac.com',
250);
go
INSERT INTO Fabricantes(codFab, nomFab, dirFab, mailFab, cantEmp)
VALUES('3A', 'Oldsmobile', '38 Queens Rd, 2ET',
'oldsmobile_factory1@oldsmobile.com', 440);
go
INSERT INTO Fabricantes(codFab, nomFab, dirFab, mailFab, cantEmp)
VALUES('3B', 'Oldsmobile', '40 Queens Rd, 2ET',
'oldsmobile_factory2@oldsmobile.com', 440);
go
INSERT INTO Fabricantes(codFab, nomFab, dirFab, mailFab, cantEmp)
VALUES('4A', 'Buick', '13 Beckett Cl, London SW16 1AN',
'buick_factory1@buick.com', 80);
go
INSERT INTO Fabricantes(codFab, nomFab, dirFab, mailFab, cantEmp)
VALUES('4B', 'Buick', '20 Beckett Cl, London SW16 1AN',
'buick_factory2@buick.com', 80);
go
INSERT INTO Fabricantes(codFab, nomFab, dirFab, mailFab, cantEmp)
VALUES('6A', 'Cadillac', '145 Sandringham Rd, Cleeth 9BP',
'cadillac_factory1@cadillac.com', 60);
go
INSERT INTO Fabricantes(codFab, nomFab, dirFab, mailFab, cantEmp)
VALUES('6B', 'Cadillac', '3 Sandringham Rd, Cleeth 9BP',
'cadillac_factory2@cadillac.com', 60);
go
INSERT INTO Fabricantes(codFab, nomFab, dirFab, mailFab, cantEmp)
VALUES('7A', 'GM Canada', '40 Springfield Rd, 3DU', 'gmcanada_factory1@gm.com',
250);
go
INSERT INTO Fabricantes(codFab, nomFab, dirFab, mailFab, cantEmp)
VALUES('7B', 'GM Canada', '20 Springfield Rd, 3DU', 'gmcanada_factory2@gm.com',
250);
go
INSERT INTO Fabricantes(codFab, nomFab, dirFab, mailFab, cantEmp)
VALUES('8A', 'Saturn', '42 Church St, 7AS', 'saturn_factory1@saturn.com', 190);
go
INSERT INTO Fabricantes(codFab, nomFab, dirFab, mailFab, cantEmp)
VALUES('8B', 'Saturn', '32 Church St, 7AS', 'saturn_factory2@saturn.com', 190);
go
-- *****
-- Tabla Plantas
-- *****
INSERT INTO Plantas(codPlan, codFab, nomPlan, dirPlan, mailPlan, codPais)
VALUES('1', 'AA', 'Audi usa', '2335 Canton Hwy #6 Windsor', 'audi@usa.com',
'1');

```

```

go
INSERT INTO Plantas(codPlan, codFab, nomPlan, dirPlan, mailPlan, codPais)
VALUES('2', 'AA', 'Audi Inglaterra', '7564 N Academy Ave',
'audi@inglaterra.com', 'S');
go
INSERT INTO Plantas(codPlan, codFab, nomPlan, dirPlan, mailPlan, codPais)
VALUES('3', 'AB', 'Audi Brasil', '88 E Saint Elmo Rd', 'audi@brasil.com', '9');
go
INSERT INTO Plantas(codPlan, codFab, nomPlan, dirPlan, mailPlan, codPais)
VALUES('1', 'BA', 'BMW US Perfection Inc', '6 Arch St #9757 Alcida',
'bmw@usa.com', '1');
go
INSERT INTO Plantas(codPlan, codFab, nomPlan, dirPlan, mailPlan, codPais)
VALUES('2', 'BA', 'BMW IT Luxury Inc', '75 Westchester Ave', 'bmw@italia.com',
'Z');
go
INSERT INTO Plantas(codPlan, codFab, nomPlan, dirPlan, mailPlan, codPais)
VALUES('3', 'BB', 'BMW CH Luxury Inc', '80 Westchester Ave', 'bmw@chile.com',
'5');
go
INSERT INTO Plantas(codPlan, codFab, nomPlan, dirPlan, mailPlan, codPais)
VALUES('1', 'CA', 'Chrysler Canada Creations', '447 Commercial St Se Lile',
'chrysler@canada.com', '2');
go
INSERT INTO Plantas(codPlan, codFab, nomPlan, dirPlan, mailPlan, codPais)
VALUES('2', 'CA', 'Chrysler Corea Designs', '765 Rock Island Rd',
'chrysler@corea.com', 'K');
go
INSERT INTO Plantas(codPlan, codFab, nomPlan, dirPlan, mailPlan, codPais)
VALUES('3', 'CB', 'Chrysler México Designs', '8 Rock Island Rd',
'chrysler@mexico.com', '3');
go
INSERT INTO Plantas(codPlan, codFab, nomPlan, dirPlan, mailPlan, codPais)
VALUES('1', 'DA', 'Benz German Designs', '9 So Ilhung Island Rd',
'benz@alemania.com', 'W');
go
INSERT INTO Plantas(codPlan, codFab, nomPlan, dirPlan, mailPlan, codPais)
VALUES('2', 'DA', 'Benz Sweeden Designs', '9 So Ilhung Island Rd',
'benz@suecia.com', 'Y');
go
INSERT INTO Plantas(codPlan, codFab, nomPlan, dirPlan, mailPlan, codPais)
VALUES('3', 'DB', 'Benz Argentina Designs', '9 So Ilhung Island Rd',
'benz@argentina.com', '4');
go
INSERT INTO Plantas(codPlan, codFab, nomPlan, dirPlan, mailPlan, codPais)
VALUES('1', 'EA', 'Dodge USA Designs', '40 Us Highway 12e', 'dodge@usa.com',
'1');
go
INSERT INTO Plantas(codPlan, codFab, nomPlan, dirPlan, mailPlan, codPais)
VALUES('2', 'EB', 'Dodge Mexico Designs', '50 Us Highway 12e',
'dodge@mexico.com', '3');
go
INSERT INTO Plantas(codPlan, codFab, nomPlan, dirPlan, mailPlan, codPais)
VALUES('1', 'FA', 'Ford usa Valley Propane Inc', '3 Mill Rd Baker Brook',
'ford@usa.com', '1');
go
INSERT INTO Plantas(codPlan, codFab, nomPlan, dirPlan, mailPlan, codPais)
VALUES('2', 'FA', 'Ford German Valley Propane Inc', '6 Mill Rd Baker Brook',
'ford@alemania.com', 'W');
go

```

```

INSERT INTO Plantas(codPlan, codFab, nomPlan, dirPlan, mailPlan, codPais)
VALUES('3', 'FB', 'Ford MX Valley Propane Inc', '20 Mill Rd Baker Brook',
'ford@mexico.com', '3');
go
INSERT INTO Plantas(codPlan, codFab, nomPlan, dirPlan, mailPlan, codPais)
VALUES('1', 'GA', 'GM Usa Ep Systems', '37 Us Highway 12e', 'gm@usa.com', '1');
go
INSERT INTO Plantas(codPlan, codFab, nomPlan, dirPlan, mailPlan, codPais)
VALUES('2', 'GA', 'GM England Ep Systems', '40 Us Highway 12e',
'gm@inglaterra.com', 'S');
go
INSERT INTO Plantas(codPlan, codFab, nomPlan, dirPlan, mailPlan, codPais)
VALUES('3', 'GB', 'GM Brasil Ep Systems', '50 Us Highway 12e', 'gm@brasil.com',
'9');
go
INSERT INTO Plantas(codPlan, codFab, nomPlan, dirPlan, mailPlan, codPais)
VALUES('1', 'HA', 'Honda Japon Campbell Auto', '2859 Dorsett Rd',
'honda@japon.com', 'J');
go
INSERT INTO Plantas(codPlan, codFab, nomPlan, dirPlan, mailPlan, codPais)
VALUES('2', 'HA', 'Honda Corea Campbell Auto', '2859 Dorsett Rd',
'honda@corea.com', 'K');
go
INSERT INTO Plantas(codPlan, codFab, nomPlan, dirPlan, mailPlan, codPais)
VALUES('3', 'HB', 'Honda Chile Campbell Auto', '2859 Dorsett Rd',
'honda@chile.com', '5');
go
INSERT INTO Plantas(codPlan, codFab, nomPlan, dirPlan, mailPlan, codPais)
VALUES('1', 'JA', 'Jaguar Italy Auto', '3045 Dorsett Rd', 'jaguar@italia.com',
'Z');
go
INSERT INTO Plantas(codPlan, codFab, nomPlan, dirPlan, mailPlan, codPais)
VALUES('2', 'JA', 'Jaguar Suecia Auto', '30 Dorsett Rd', 'jaguar@suecia.com',
'Y');
go
INSERT INTO Plantas(codPlan, codFab, nomPlan, dirPlan, mailPlan, codPais)
VALUES('3', 'JB', 'Jaguar Brasil Auto', '3060 Dorsett Rd', 'jaguar@brasil.com',
'9');
go
INSERT INTO Plantas(codPlan, codFab, nomPlan, dirPlan, mailPlan, codPais)
VALUES('1', 'LA', 'Lincon England Auto', '3045 Belmont Rd',
'lincon@inglaterra.com', 'S');
go
INSERT INTO Plantas(codPlan, codFab, nomPlan, dirPlan, mailPlan, codPais)
VALUES('2', 'LB', 'Lincon Argentina Auto', '30 Belmont Rd',
'lincon@argentina.com', '4');
go
INSERT INTO Plantas(codPlan, codFab, nomPlan, dirPlan, mailPlan, codPais)
VALUES('1', 'MA', 'Mercury USA Designs', '1 Belmont Rd 3F', 'mercury@usa.com',
'1');
go
INSERT INTO Plantas(codPlan, codFab, nomPlan, dirPlan, mailPlan, codPais)
VALUES('2', 'MB', 'Mercury Chile Designs', '99 Belmont Rd 3F',
'mercury@chile.com', '5');
go
INSERT INTO Plantas(codPlan, codFab, nomPlan, dirPlan, mailPlan, codPais)
VALUES('1', 'NA', 'Nissan Corea Designs', '1 Pittsford Rd 3F',
'nissan@corea.com', 'K');
go
INSERT INTO Plantas(codPlan, codFab, nomPlan, dirPlan, mailPlan, codPais)

```

```

VALUES('2', 'NB', 'Nissan Chile Designs', '10 Pittsford Rd 3F',
'nissan@chile.com', '5');
go
INSERT INTO Plantas(codPlan, codFab, nomPlan, dirPlan, mailPlan, codPais)
VALUES('1', 'PA', 'Plymouth USA Designs', '1 Pittsford Rd 3B',
'plymouth@usa.com', '1');
go
INSERT INTO Plantas(codPlan, codFab, nomPlan, dirPlan, mailPlan, codPais)
VALUES('2', 'PB', 'Plymouth Uruguay Designs', '13 Pittsford Rd 3B',
'plymouth@uruguay.com', '6');
go
INSERT INTO Plantas(codPlan, codFab, nomPlan, dirPlan, mailPlan, codPais)
VALUES('1', 'SA', 'Saab Italy Designs', '01 Pittsford Rd 3Z',
'saab@italia.com', 'Z');
go
INSERT INTO Plantas(codPlan, codFab, nomPlan, dirPlan, mailPlan, codPais)
VALUES('2', 'SB', 'Saab Uruguay Designs', '19 Pittsford Rd 3Z9',
'saab@uruguay.com', '6');
go
INSERT INTO Plantas(codPlan, codFab, nomPlan, dirPlan, mailPlan, codPais)
VALUES('1', 'TA', 'Toyota Japon Designs', 'Pittsford Rd 39',
'toyota@japon.com', 'J');
go
INSERT INTO Plantas(codPlan, codFab, nomPlan, dirPlan, mailPlan, codPais)
VALUES('2', 'TB', 'Toyota Corea Designs', 'Pittsford Rd 59',
'toyota@corea.com', 'K');
go
INSERT INTO Plantas(codPlan, codFab, nomPlan, dirPlan, mailPlan, codPais)
VALUES('1', 'VA', 'Volkswagen German Designs', 'Pittsford Rd 129',
'vw@alemania.com', 'W');
go
INSERT INTO Plantas(codPlan, codFab, nomPlan, dirPlan, mailPlan, codPais)
VALUES('2', 'VB', 'Volkswagen Brasil Designs', 'Pittsford Rd 300',
'vw@brasil.com', '9');
go
INSERT INTO Plantas(codPlan, codFab, nomPlan, dirPlan, mailPlan, codPais)
VALUES('1', '2A', 'Pontiac USA Designs', 'Pittsford Rd 23B', 'pontiac@usa.com',
'1');
go
INSERT INTO Plantas(codPlan, codFab, nomPlan, dirPlan, mailPlan, codPais)
VALUES('2', '2B', 'Pontiac Chile Designs', 'Pittsford Rd 23C',
'pontiac@chile.com', '5');
go
INSERT INTO Plantas(codPlan, codFab, nomPlan, dirPlan, mailPlan, codPais)
VALUES('1', '3A', 'Oldsmobile Sweeden Designs', 'Belmont Rd 33B',
'oldsmobile@suecia.com', 'Y');
go
INSERT INTO Plantas(codPlan, codFab, nomPlan, dirPlan, mailPlan, codPais)
VALUES('1', '3B', 'Oldsmobile Argentina Designs', 'Belmont Rd 33C',
'oldsmobile@argentina.com', '4');
go
INSERT INTO Plantas(codPlan, codFab, nomPlan, dirPlan, mailPlan, codPais)
VALUES('1', '4A', 'Buick England Auto', '9547 Belmont Rd #21',
'buick@inglaterra.com', 'S');
go
INSERT INTO Plantas(codPlan, codFab, nomPlan, dirPlan, mailPlan, codPais)
VALUES('2', '4B', 'Buick Uruguay Auto', '9547 Belmont Rd #21',
'buick@uruguay.com', '6');
go
INSERT INTO Plantas(codPlan, codFab, nomPlan, dirPlan, mailPlan, codPais)

```

```

VALUES('1', '6A', 'Cadillac USA Consultants', '73 Pittsford Victor Rd',
'cadillac@usa.com', '1');
go
INSERT INTO Plantas(codPlan, codFab, nomPlan, dirPlan, mailPlan, codPais)
VALUES('2', '6B', 'Cadillac Brasil Consultants', '5174 Interstate 45 N',
'cadillac@brasil.com', '9');
go
INSERT INTO Plantas(codPlan, codFab, nomPlan, dirPlan, mailPlan, codPais)
VALUES('1', '7A', 'GM Canada Designs', '136 W Grand Ave #3',
'gmcanada@canada.com', '2');
go
INSERT INTO Plantas(codPlan, codFab, nomPlan, dirPlan, mailPlan, codPais)
VALUES('2', '7A', 'GM Canada Designs', '136 W Grand Ave #4',
'gmcanada2@canada.com', '2');
go
INSERT INTO Plantas(codPlan, codFab, nomPlan, dirPlan, mailPlan, codPais)
VALUES('3', '7B', 'GM Canada Designs', '136 W Grand Ave #5',
'gmcanada@mexico.com', '3');
go
INSERT INTO Plantas(codPlan, codFab, nomPlan, dirPlan, mailPlan, codPais)
VALUES('1', '8A', 'Saturn Canada Designs', '136 W Grand Ave #8',
'saturn@canada.com', '2');
go
INSERT INTO Plantas(codPlan, codFab, nomPlan, dirPlan, mailPlan, codPais)
VALUES('2', '8B', 'Saturn Uruguay Designs', '136 W Grand Ave #20',
'saturn@uruguay.com', '6');
go
-- *****
-- Tabla Vehiculos
-- *****
-- Audi
INSERT INTO Vehiculos(vin, modelo, color, peso, características, codPais,
codFab)
VALUES ('1AAFE3203FAA10190', 'A4', 'gris', 2500, 'Frenos ABS, Aire
Acondicionado y tapizado de cuero', '1', 'AA')
go
INSERT INTO Vehiculos(vin, modelo, color, peso, características, codPais,
codFab)
VALUES ('1AAFE3201GAA10190', 'A4', 'negro', 2500, 'Frenos ABS, Aire
Acondicionado y tapizado de cuero', '1', 'AA')
go
INSERT INTO Vehiculos(vin, modelo, color, peso, características, codPais,
codFab)
VALUES ('1AAAL320XHAA10190', 'A5', 'negro', 1500, 'Frenos ABS, Aire
Acondicionado y tapizado de cuero', '1', 'AA')
go
INSERT INTO Vehiculos(vin, modelo, color, peso, características, codPais,
codFab)
VALUES ('9ABAL320XHAB10190', 'A5', 'gris', 1500, 'Frenos ABS, Aire
Acondicionado y tapizado de cuero', '9', 'AB')
go
INSERT INTO Vehiculos(vin, modelo, color, peso, características, codPais,
codFab)
VALUES ('SAAAL3207HAA10190', 'A5', 'gris', 1500, 'Frenos ABS, Aire
Acondicionado y tapizado de cuero', 'S', 'AA')
go
-- BMW
INSERT INTO Vehiculos(vin, modelo, color, peso, características, codPais,
codFab)

```



```

VALUES ('1BAAL3150FBA10190', 'Z3', 'negro', 1000, 'Frenos ABS, Aire
Acondicionado y tapizado de cuero', '1', 'BA')
go
INSERT INTO Vehiculos(vin, modelo, color, peso, características, codPais,
codFab)
VALUES ('1BAAL3159GBA10190', 'Z3', 'negro', 1000, 'Frenos ABS, Aire
Acondicionado y tapizado de cuero', '1', 'BA')
go
INSERT INTO Vehiculos(vin, modelo, color, peso, características, codPais,
codFab)
VALUES ('ZBAFE3201HBA10190', 'Z4', 'gris', 1800, 'Frenos ABS, Aire
Acondicionado y tapizado de cuero', 'Z', 'BA')
go
INSERT INTO Vehiculos(vin, modelo, color, peso, características, codPais,
codFab)
VALUES ('5BBFE3204HBB10190', 'Z4', 'negro', 1800, 'Frenos ABS, Aire
Acondicionado y tapizado de cuero', '5', 'BB')
go
-- Buick
INSERT INTO Vehiculos(vin, modelo, color, peso, características, codPais,
codFab)
VALUES ('S4AFE3201F4A10190', 'LaCrosse', 'negro', 2800, 'Frenos ABS, Aire
Acondicionado y tapizado de cuero', 'S', '4A')
go
INSERT INTO Vehiculos(vin, modelo, color, peso, características, codPais,
codFab)
VALUES ('S4AFE320XG4A10190', 'LaCrosse', 'negro', 2800, 'Frenos ABS, Aire
Acondicionado y tapizado de cuero', 'S', '4A')
go
INSERT INTO Vehiculos(vin, modelo, color, peso, características, codPais,
codFab)
VALUES ('S4AFE3208H4A10190', 'LaCrosse', 'gris', 2800, 'Frenos ABS, Aire
Acondicionado y tapizado de cuero', 'S', '4A')
go
INSERT INTO Vehiculos(vin, modelo, color, peso, características, codPais,
codFab)
VALUES ('64BFE3154G4B10190', 'LaCrosse', 'gris', 2800, 'Frenos ABS, Aire
Acondicionado y tapizado de cuero', '6', '4B')
go
-- Cadillac
INSERT INTO Vehiculos(vin, modelo, color, peso, características, codPais,
codFab)
VALUES ('16AFE3201F6A10190', 'XTS', 'gris', 3800, 'Frenos ABS, Aire
Acondicionado y tapizado de cuero', '1', '6A')
go
INSERT INTO Vehiculos(vin, modelo, color, peso, características, codPais,
codFab)
VALUES ('16AFE320XG6A10190', 'XTS', 'gris', 3800, 'Frenos ABS, Aire
Acondicionado y tapizado de cuero', '1', '6A')
go
INSERT INTO Vehiculos(vin, modelo, color, peso, características, codPais,
codFab)
VALUES ('16AFE3208H6A10190', 'XTS', 'negro', 3800, 'Frenos ABS, Aire
Acondicionado y tapizado de cuero', '1', '6A')
go
INSERT INTO Vehiculos(vin, modelo, color, peso, características, codPais,
codFab)
VALUES ('96BFE3208H6B10190', 'XTS', 'negro', 3800, 'Frenos ABS, Aire
Acondicionado y tapizado de cuero', '9', '6B')
go

```

```

-- Chrysler
INSERT INTO Vehiculos(vin, modelo, color, peso, características, codPais,
codFab)
VALUES ('2CAFE3208FCA10190', 'Pacifica', 'negro', 2800, 'Frenos ABS, Aire
Acondicionado y tapizado de cuero', '2', 'CA')
go
INSERT INTO Vehiculos(vin, modelo, color, peso, características, codPais,
codFab)
VALUES ('2CAFE3206GCA10190', 'Pacifica', 'negro', 2800, 'Frenos ABS, Aire
Acondicionado y tapizado de cuero', '2', 'CA')
go
INSERT INTO Vehiculos(vin, modelo, color, peso, características, codPais,
codFab)
VALUES ('2CAFE3204HCA10190', 'Pacifica', 'gris', 2800, 'Frenos ABS, Aire
Acondicionado y tapizado de cuero', '2', 'CA')
go
INSERT INTO Vehiculos(vin, modelo, color, peso, características, codPais,
codFab)
VALUES ('KCAFE3208FCA10190', 'Pacifica', 'verde', 2800, 'Frenos ABS, Aire
Acondicionado y tapizado de cuero', 'K', 'CA')
go
INSERT INTO Vehiculos(vin, modelo, color, peso, características, codPais,
codFab)
VALUES ('KCAFE3206GCA10190', 'Pacifica', 'verde', 2800, 'Frenos ABS, Aire
Acondicionado y tapizado de cuero', 'K', 'CA')
go
INSERT INTO Vehiculos(vin, modelo, color, peso, características, codPais,
codFab)
VALUES ('KCAFE3204HCA10190', 'Pacifica', 'verde', 2800, 'Frenos ABS, Aire
Acondicionado y tapizado de cuero', 'K', 'CA')
go
INSERT INTO Vehiculos(vin, modelo, color, peso, características, codPais,
codFab)
VALUES ('3CBFE3203HCB10190', 'Pacifica', 'rojo', 2800, 'Frenos ABS, Aire
Acondicionado y tapizado de cuero', '3', 'CB')
go
-- Dodge
INSERT INTO Vehiculos(vin, modelo, color, peso, características, codPais,
codFab)
VALUES ('1EAFE3208FEA10190', 'Challenger', 'negro', 3000, 'Frenos ABS, Aire
Acondicionado y tapizado de cuero', '1', 'EA')
go
INSERT INTO Vehiculos(vin, modelo, color, peso, características, codPais,
codFab)
VALUES ('1EAFE3206GEA10190', 'Challenger', 'negro', 3000, 'Frenos ABS, Aire
Acondicionado y tapizado de cuero', '1', 'EA')
go
INSERT INTO Vehiculos(vin, modelo, color, peso, características, codPais,
codFab)
VALUES ('1EAFE3204HEA10190', 'Challenger', 'negro', 3000, 'Frenos ABS, Aire
Acondicionado y tapizado de cuero', '1', 'EA')
go
INSERT INTO Vehiculos(vin, modelo, color, peso, características, codPais,
codFab)
VALUES ('3EBFE3200HEB10190', 'Challenger', 'negro', 3000, 'Frenos ABS, Aire
Acondicionado y tapizado de cuero', '3', 'EB')
go
-- Ford
INSERT INTO Vehiculos(vin, modelo, color, peso, características, codPais,
codFab)

```

```

VALUES ('1FAFE320XGFA10190', 'Mustang', 'rojo', 3200, 'Frenos ABS, Aire
Acondicionado y tapizado de cuero', '1', 'FA')
go
INSERT INTO Vehiculos(vin, modelo, color, peso, características, codPais,
codFab)
VALUES ('1FAFE3208HFA10190', 'Mustang', 'rojo', 3200, 'Frenos ABS, Aire
Acondicionado y tapizado de cuero', '1', 'FA')
go
INSERT INTO Vehiculos(vin, modelo, color, peso, características, codPais,
codFab)
VALUES ('WFAFE3204HFA10190', 'Mustang', 'negro', 3200, 'Frenos ABS, Aire
Acondicionado y tapizado de cuero', 'W', 'FA')
go
INSERT INTO Vehiculos(vin, modelo, color, peso, características, codPais,
codFab)
VALUES ('3FBFE3204HFB10190', 'Mustang', 'gris', 3200, 'Frenos ABS, Aire
Acondicionado y tapizado de cuero', '3', 'FB')
go
-- GM
INSERT INTO Vehiculos(vin, modelo, color, peso, características, codPais,
codFab)
VALUES ('1GAFE3205FGA10190', 'Camaro', 'negro', 5200, 'Frenos ABS, Aire
Acondicionado y tapizado de cuero', '1', 'GA')
go
INSERT INTO Vehiculos(vin, modelo, color, peso, características, codPais,
codFab)
VALUES ('1GAFE3203GGA10190', 'Camaro', 'gris', 5200, 'Frenos ABS, Aire
Acondicionado y tapizado de cuero', '1', 'GA')
go
INSERT INTO Vehiculos(vin, modelo, color, peso, características, codPais,
codFab)
VALUES ('SGAFE3202FGA10190', 'Camaro', 'gris', 5200, 'Frenos ABS, Aire
Acondicionado y tapizado de cuero', 'S', 'GA')
go
INSERT INTO Vehiculos(vin, modelo, color, peso, características, codPais,
codFab)
VALUES ('SGAFE3200GGA10190', 'Camaro', 'negro', 5200, 'Frenos ABS, Aire
Acondicionado y tapizado de cuero', 'S', 'GA')
go
INSERT INTO Vehiculos(vin, modelo, color, peso, características, codPais,
codFab)
VALUES ('9GBFE3203GGB10190', 'Camaro', 'negro', 5200, 'Frenos ABS, Aire
Acondicionado y tapizado de cuero', '9', 'GB')
go
-- GM Canada
INSERT INTO Vehiculos(vin, modelo, color, peso, características, codPais,
codFab)
VALUES ('27AFE3202F7A10190', 'Camaro', 'negro', 5200, 'Frenos ABS, Aire
Acondicionado y tapizado de cuero', '2', '7A')
go
INSERT INTO Vehiculos(vin, modelo, color, peso, características, codPais,
codFab)
VALUES ('27AFE3200G7A10190', 'Camaro', 'gris', 5200, 'Frenos ABS, Aire
Acondicionado y tapizado de cuero', '2', '7A')
go
INSERT INTO Vehiculos(vin, modelo, color, peso, características, codPais,
codFab)
VALUES ('27AFE3209H7A10190', 'Camaro', 'gris', 5200, 'Frenos ABS, Aire
Acondicionado y tapizado de cuero', '2', '7A')
go

```

```

INSERT INTO Vehiculos(vin, modelo, color, peso, características, codPais,
codFab)
VALUES ('37BFE320XG7B10190', 'Camaro', 'negro', 5200, 'Frenos ABS, Aire
Acondicionado y tapizado de cuero', '3', '7B')
go
INSERT INTO Vehiculos(vin, modelo, color, peso, características, codPais,
codFab)
VALUES ('37BFE3208H7B10190', 'Camaro', 'negro', 5200, 'Frenos ABS, Aire
Acondicionado y tapizado de cuero', '3', '7B')
go
-- Honda
INSERT INTO Vehiculos(vin, modelo, color, peso, características, codPais,
codFab)
VALUES ('JHAFE3207GHA10190', 'Civic', 'gris', 4000, 'Frenos ABS, Aire
Acondicionado y tapizado de cuero', 'J', 'HA')
go
INSERT INTO Vehiculos(vin, modelo, color, peso, características, codPais,
codFab)
VALUES ('JHAFE3205HHA10190', 'Civic', 'negro', 4000, 'Frenos ABS, Aire
Acondicionado y tapizado de cuero', 'J', 'HA')
go
INSERT INTO Vehiculos(vin, modelo, color, peso, características, codPais,
codFab)
VALUES ('KHAFE3206FHA10190', 'Civic', 'gris', 4000, 'Frenos ABS, Aire
Acondicionado y tapizado de cuero', 'K', 'HA')
go
INSERT INTO Vehiculos(vin, modelo, color, peso, características, codPais,
codFab)
VALUES ('KHAFE3204GHA10190', 'Civic', 'negro', 4000, 'Frenos ABS, Aire
Acondicionado y tapizado de cuero', 'K', 'HA')
go
INSERT INTO Vehiculos(vin, modelo, color, peso, características, codPais,
codFab)
VALUES ('5HBFE3208GHB10190', 'Civic', 'gris', 4000, 'Frenos ABS, Aire
Acondicionado y tapizado de cuero', '5', 'HB')
go
-- Jaguar
INSERT INTO Vehiculos(vin, modelo, color, peso, características, codPais,
codFab)
VALUES ('YJAFE3202GJA10190', 'IPACE CONCEPT', 'gris', 3500, 'Frenos ABS, Aire
Acondicionado y tapizado de cuero', 'Y', 'JA')
go
INSERT INTO Vehiculos(vin, modelo, color, peso, características, codPais,
codFab)
VALUES ('YJAFE3200HJA10190', 'IPACE CONCEPT', 'negro', 3500, 'Frenos ABS, Aire
Acondicionado y tapizado de cuero', 'Y', 'JA')
go
INSERT INTO Vehiculos(vin, modelo, color, peso, características, codPais,
codFab)
VALUES ('ZJAFE320XGJA10190', 'IPACE CONCEPT', 'gris', 3500, 'Frenos ABS, Aire
Acondicionado y tapizado de cuero', 'Z', 'JA')
go
INSERT INTO Vehiculos(vin, modelo, color, peso, características, codPais,
codFab)
VALUES ('ZJAFE3208HJA10190', 'IPACE CONCEPT', 'negro', 3500, 'Frenos ABS, Aire
Acondicionado y tapizado de cuero', 'Z', 'JA')
go
INSERT INTO Vehiculos(vin, modelo, color, peso, características, codPais,
codFab)

```

```

VALUES ('9JBFE320XHJB10190', 'IPACE CONCEPT', 'gris', 3500, 'Frenos ABS, Aire
Acondicionado y tapizado de cuero', '9', 'JB')
go
--Mercedes Benz
INSERT INTO Vehiculos(vin, modelo, color, peso, características, codPais,
codFab)
VALUES ('WDAFE3209GDA10190', 'Hatchback', 'blanco', 3500, 'Frenos ABS, Aire
Acondicionado y tapizado de cuero', 'W', 'DA')
go
INSERT INTO Vehiculos(vin, modelo, color, peso, características, codPais,
codFab)
VALUES ('WDAFE3150HDA10190', 'Spyder', 'negro', 2100, 'Frenos ABS, Aire
Acondicionado y tapizado de cuero', 'W', 'DA')
go
--Pontiac
INSERT INTO Vehiculos(vin, modelo, color, peso, características, codPais,
codFab)
VALUES ('12AFE3204G2A10190', 'Sunbird', 'blanco', 3800, 'Frenos ABS, Aire
Acondicionado y tapizado de cuero', '1', '2A')
go
-- *****
-- Tabla Envios
-- *****
INSERT INTO Envios(fchEnvio, pesoEnvio, oriEnvio, desEnvio)
VALUES ('20150315', 3990, '1', 'Z');
go
INSERT INTO Envios(fchEnvio, pesoEnvio, oriEnvio, desEnvio)
VALUES ('20170415', 3990, '1', 'Z');
go
INSERT INTO Envios(fchEnvio, pesoEnvio, oriEnvio, desEnvio)
VALUES ('20160915', 3990, '1', 'Y');
go
INSERT INTO Envios(fchEnvio, pesoEnvio, oriEnvio, desEnvio)
VALUES ('20170115', 1575, '1', 'Y');
go
INSERT INTO Envios(fchEnvio, pesoEnvio, oriEnvio, desEnvio)
VALUES ('20160115', 2625, '1', '2');
go
INSERT INTO Envios(fchEnvio, pesoEnvio, oriEnvio, desEnvio)
VALUES ('20151015', 2625, '1', '2');
go
INSERT INTO Envios(fchEnvio, pesoEnvio, oriEnvio, desEnvio)
VALUES ('20151115', 1050, '1', '3');
go
INSERT INTO Envios(fchEnvio, pesoEnvio, oriEnvio, desEnvio)
VALUES ('20160815', 1050, '1', '5');
go
INSERT INTO Envios(fchEnvio, pesoEnvio, oriEnvio, desEnvio)
VALUES ('20170415', 3150, '1', '5');
go
INSERT INTO Envios(fchEnvio, pesoEnvio, oriEnvio, desEnvio)
VALUES ('20160415', 3150, '1', '5');
go
INSERT INTO Envios(fchEnvio, pesoEnvio, oriEnvio, desEnvio)
VALUES ('20150415', 3150, '1', '6');
go
INSERT INTO Envios(fchEnvio, pesoEnvio, oriEnvio, desEnvio)
VALUES ('20170215', 3360, '1', '6');
go
INSERT INTO Envios(fchEnvio, pesoEnvio, oriEnvio, desEnvio)

```

```

VALUES ('20160615', 3360, '1', '9');
go
INSERT INTO Envios(fchEnvio, pesoEnvio, oriEnvio, desEnvio)
VALUES ('20160715', 5460, '1', '9');
go
INSERT INTO Envios(fchEnvio, pesoEnvio, oriEnvio, desEnvio)
VALUES ('20150715', 5460, '1', 'Z');
go
INSERT INTO Envios(fchEnvio, pesoEnvio, oriEnvio, desEnvio)
VALUES ('20160815', 5460, '2', 'Z');
go
INSERT INTO Envios(fchEnvio, pesoEnvio, oriEnvio, desEnvio)
VALUES ('20150915', 5460, '2', 'Z');
go
INSERT INTO Envios(fchEnvio, pesoEnvio, oriEnvio, desEnvio)
VALUES ('20170315', 5460, '2', 'J');
go
INSERT INTO Envios(fchEnvio, pesoEnvio, oriEnvio, desEnvio)
VALUES ('20170115', 2940, '2', 'Z');
go
INSERT INTO Envios(fchEnvio, pesoEnvio, oriEnvio, desEnvio)
VALUES ('20160115', 2940, '2', 'Z');
go
INSERT INTO Envios(fchEnvio, pesoEnvio, oriEnvio, desEnvio)
VALUES ('20150315', 2940, '2', 'J');
go
INSERT INTO Envios(fchEnvio, pesoEnvio, oriEnvio, desEnvio)
VALUES ('20170120', 5460, '3', 'J');
go
INSERT INTO Envios(fchEnvio, pesoEnvio, oriEnvio, desEnvio)
VALUES ('20160111', 5460, '3', '6');
go
INSERT INTO Envios(fchEnvio, pesoEnvio, oriEnvio, desEnvio)
VALUES ('20170207', 2940, '3', '6');
go
INSERT INTO Envios(fchEnvio, pesoEnvio, oriEnvio, desEnvio)
VALUES ('20170224', 3150, '3', 'K');
go
INSERT INTO Envios(fchEnvio, pesoEnvio, oriEnvio, desEnvio)
VALUES ('20170305', 3360, '3', 'K');
go
INSERT INTO Envios(fchEnvio, pesoEnvio, oriEnvio, desEnvio)
VALUES ('20170407', 1890, '5', '6');
go
INSERT INTO Envios(fchEnvio, pesoEnvio, oriEnvio, desEnvio)
VALUES ('20161120', 4200, '5', '6');
go
INSERT INTO Envios(fchEnvio, pesoEnvio, oriEnvio, desEnvio)
VALUES ('20160809', 2940, '6', '5');
go
INSERT INTO Envios(fchEnvio, pesoEnvio, oriEnvio, desEnvio)
VALUES ('20170115', 3990, '9', '1');
go
INSERT INTO Envios(fchEnvio, pesoEnvio, oriEnvio, desEnvio)
VALUES ('20170620', 1575, '9', '1');
go
INSERT INTO Envios(fchEnvio, pesoEnvio, oriEnvio, desEnvio)
VALUES ('20161005', 5460, '9', '2');
go
INSERT INTO Envios(fchEnvio, pesoEnvio, oriEnvio, desEnvio)

```

```

VALUES ('20170314', 3675, '9', '2');
go
INSERT INTO Envios(fchEnvio, pesoEnvio, oriEnvio, desEnvio)
VALUES ('20170318', 4200, 'J', 'Z');
go
INSERT INTO Envios(fchEnvio, pesoEnvio, oriEnvio, desEnvio)
VALUES ('20161204', 4200, 'J', 'Z');
go
INSERT INTO Envios(fchEnvio, pesoEnvio, oriEnvio, desEnvio)
VALUES ('20170103', 2940, 'K', 'Z');
go
INSERT INTO Envios(fchEnvio, pesoEnvio, oriEnvio, desEnvio)
VALUES ('20160828', 2940, 'K', 'J');
go
INSERT INTO Envios(fchEnvio, pesoEnvio, oriEnvio, desEnvio)
VALUES ('20150120', 2940, 'K', 'J');
go
INSERT INTO Envios(fchEnvio, pesoEnvio, oriEnvio, desEnvio)
VALUES ('20160214', 4200, 'K', 'S');
go
INSERT INTO Envios(fchEnvio, pesoEnvio, oriEnvio, desEnvio)
VALUES ('20150226', 4200, 'K', 'S');
go
INSERT INTO Envios(fchEnvio, pesoEnvio, oriEnvio, desEnvio)
VALUES ('20151010', 2940, 'S', '9');
go
INSERT INTO Envios(fchEnvio, pesoEnvio, oriEnvio, desEnvio)
VALUES ('20170102', 2940, 'S', '9');
go
INSERT INTO Envios(fchEnvio, pesoEnvio, oriEnvio, desEnvio)
VALUES ('20160505', 2940, 'S', '6');
go
INSERT INTO Envios(fchEnvio, pesoEnvio, oriEnvio, desEnvio)
VALUES ('20170213', 1575, 'S', '6');
go
INSERT INTO Envios(fchEnvio, pesoEnvio, oriEnvio, desEnvio)
VALUES ('20161220', 5460, 'S', '6');
go
INSERT INTO Envios(fchEnvio, pesoEnvio, oriEnvio, desEnvio)
VALUES ('20150606', 5460, 'S', '2');
go
INSERT INTO Envios(fchEnvio, pesoEnvio, oriEnvio, desEnvio)
VALUES ('20170420', 3360, 'W', 'J');
go
INSERT INTO Envios(fchEnvio, pesoEnvio, oriEnvio, desEnvio)
VALUES ('20170501', 3675, 'Y', 'J');
go
INSERT INTO Envios(fchEnvio, pesoEnvio, oriEnvio, desEnvio)
VALUES ('20160820', 3675, 'Y', 'K');
go
INSERT INTO Envios(fchEnvio, pesoEnvio, oriEnvio, desEnvio)
VALUES ('20170123', 1890, 'Z', '2');
go
INSERT INTO Envios(fchEnvio, pesoEnvio, oriEnvio, desEnvio)
VALUES ('20170111', 3675, 'Z', '1');
go
INSERT INTO Envios(fchEnvio, pesoEnvio, oriEnvio, desEnvio)
VALUES ('20161008', 3675, 'Z', 'K');
go
-- *****

```

```

-- Tabla Cargas
-- *****
INSERT INTO Carga(idEnvio, idCarga, vin, pesoCarga)
VALUES(1, 1, '16AFE3201F6A10190', 3990)
go
INSERT INTO Carga(idEnvio, idCarga, vin, pesoCarga)
VALUES(2, 1, '16AFE3208H6A10190', 3990)
go
INSERT INTO Carga(idEnvio, idCarga, vin, pesoCarga)
VALUES(3, 1, '16AFE320XG6A10190', 3990)
go
INSERT INTO Carga(idEnvio, idCarga, vin, pesoCarga)
VALUES(4, 1, '1AAAL320XHAA10190', 1575)
go
INSERT INTO Carga(idEnvio, idCarga, vin, pesoCarga)
VALUES(5, 1, '1AAFE3201GAA10190', 2625)
go
INSERT INTO Carga(idEnvio, idCarga, vin, pesoCarga)
VALUES(6, 1, '1AAFE3203FAA10190', 2625)
go
INSERT INTO Carga(idEnvio, idCarga, vin, pesoCarga)
VALUES(7, 1, '1BAAL3150FBA10190', 1050)
go
INSERT INTO Carga(idEnvio, idCarga, vin, pesoCarga)
VALUES(8, 1, '1BAAL3159GBA10190', 1050)
go
INSERT INTO Carga(idEnvio, idCarga, vin, pesoCarga)
VALUES(9, 1, '1EAFE3204HEA10190', 3150)
go
INSERT INTO Carga(idEnvio, idCarga, vin, pesoCarga)
VALUES(10, 1, '1EAFE3206GEA10190', 3150)
go
INSERT INTO Carga(idEnvio, idCarga, vin, pesoCarga)
VALUES(11, 1, '1EAFE3208FEA10190', 3150)
go
INSERT INTO Carga(idEnvio, idCarga, vin, pesoCarga)
VALUES(12, 1, '1FAFE3208HFA10190', 3360)
go
INSERT INTO Carga(idEnvio, idCarga, vin, pesoCarga)
VALUES(13, 1, '1FAFE320XGFA10190', 3360)
go
INSERT INTO Carga(idEnvio, idCarga, vin, pesoCarga)
VALUES(14, 1, '1GAFE3203GGA10190', 5460)
go
INSERT INTO Carga(idEnvio, idCarga, vin, pesoCarga)
VALUES(15, 1, '1GAFE3205FGA10190', 5460)
go
INSERT INTO Carga(idEnvio, idCarga, vin, pesoCarga)
VALUES(16, 1, '27AFE3200G7A10190', 5460)
go
INSERT INTO Carga(idEnvio, idCarga, vin, pesoCarga)
VALUES(17, 1, '27AFE3202F7A10190', 5460)
go
INSERT INTO Carga(idEnvio, idCarga, vin, pesoCarga)
VALUES(18, 1, '27AFE3209H7A10190', 5460)
go
INSERT INTO Carga(idEnvio, idCarga, vin, pesoCarga)
VALUES(19, 1, '2CAFE3204HCA10190', 2940)
go
INSERT INTO Carga(idEnvio, idCarga, vin, pesoCarga)

```



```

VALUES(20, 1, '2CAFE3206GCA10190', 2940)
go
INSERT INTO Carga(idEnvio, idCarga, vin, pesoCarga)
VALUES(21, 1, '2CAFE3208FCA10190', 2940)
go
INSERT INTO Carga(idEnvio, idCarga, vin, pesoCarga)
VALUES(22, 1, '37BFE3208H7B10190', 5460)
go
INSERT INTO Carga(idEnvio, idCarga, vin, pesoCarga)
VALUES(23, 1, '37BFE320XG7B10190', 5460)
go
INSERT INTO Carga(idEnvio, idCarga, vin, pesoCarga)
VALUES(24, 1, '3CBFE3203HCB10190', 2940)
go
INSERT INTO Carga(idEnvio, idCarga, vin, pesoCarga)
VALUES(25, 1, '3EBFE3200HEB10190', 3150)
go
INSERT INTO Carga(idEnvio, idCarga, vin, pesoCarga)
VALUES(26, 1, '3FBFE3204HFB10190', 3360)
go
INSERT INTO Carga(idEnvio, idCarga, vin, pesoCarga)
VALUES(27, 1, '5BBFE3204HBB10190', 1890)
go
INSERT INTO Carga(idEnvio, idCarga, vin, pesoCarga)
VALUES(28, 1, '5HBFE3208GHB10190', 4200)
go
INSERT INTO Carga(idEnvio, idCarga, vin, pesoCarga)
VALUES(29, 1, '64BFE3154G4B10190', 2940)
go
INSERT INTO Carga(idEnvio, idCarga, vin, pesoCarga)
VALUES(30, 1, '96BFE3208H6B10190', 3990)
go
INSERT INTO Carga(idEnvio, idCarga, vin, pesoCarga)
VALUES(31, 1, '9ABAL320XHAB10190', 1575)
go
INSERT INTO Carga(idEnvio, idCarga, vin, pesoCarga)
VALUES(32, 1, '9GBFE3203GGB10190', 5460)
go
INSERT INTO Carga(idEnvio, idCarga, vin, pesoCarga)
VALUES(33, 1, '9JBFE320XHJB10190', 3675)
go
INSERT INTO Carga(idEnvio, idCarga, vin, pesoCarga)
VALUES(34, 1, 'JHAFE3205HHA10190', 4200)
go
INSERT INTO Carga(idEnvio, idCarga, vin, pesoCarga)
VALUES(35, 1, 'JHAFE3207GHA10190', 4200)
go
INSERT INTO Carga(idEnvio, idCarga, vin, pesoCarga)
VALUES(36, 1, 'KCAFE3204HCA10190', 2940)
go
INSERT INTO Carga(idEnvio, idCarga, vin, pesoCarga)
VALUES(37, 1, 'KCAFE3206GCA10190', 2940)
go
INSERT INTO Carga(idEnvio, idCarga, vin, pesoCarga)
VALUES(38, 1, 'KCAFE3208FCA10190', 2940)
go
INSERT INTO Carga(idEnvio, idCarga, vin, pesoCarga)
VALUES(39, 1, 'KHA FE3204GHA10190', 4200)
go
INSERT INTO Carga(idEnvio, idCarga, vin, pesoCarga)

```

```

VALUES(40, 1, 'KHAFF3206FHA10190', 4200)
go
INSERT INTO Carga(idEnvio, idCarga, vin, pesoCarga)
VALUES(41, 1, 'S4AFE3201F4A10190', 2940)
go
INSERT INTO Carga(idEnvio, idCarga, vin, pesoCarga)
VALUES(42, 1, 'S4AFE3208H4A10190', 2940)
go
INSERT INTO Carga(idEnvio, idCarga, vin, pesoCarga)
VALUES(43, 1, 'S4AFE320XG4A10190', 2940)
go
INSERT INTO Carga(idEnvio, idCarga, vin, pesoCarga)
VALUES(44, 1, 'SAAAL3207HAA10190', 1575)
go
INSERT INTO Carga(idEnvio, idCarga, vin, pesoCarga)
VALUES(45, 1, 'SGAFE3200GGA10190', 5460)
go
INSERT INTO Carga(idEnvio, idCarga, vin, pesoCarga)
VALUES(46, 1, 'SGAFE3202FGA10190', 5460)
go
INSERT INTO Carga(idEnvio, idCarga, vin, pesoCarga)
VALUES(47, 1, 'WFAFE3204HFA10190', 3360)
go
INSERT INTO Carga(idEnvio, idCarga, vin, pesoCarga)
VALUES(48, 1, 'YJAFE3200HJA10190', 3675)
go
INSERT INTO Carga(idEnvio, idCarga, vin, pesoCarga)
VALUES(49, 1, 'YJAFE3202GJA10190', 3675)
go
INSERT INTO Carga(idEnvio, idCarga, vin, pesoCarga)
VALUES(50, 1, 'ZBAFE3201HBA10190', 1890)
go
INSERT INTO Carga(idEnvio, idCarga, vin, pesoCarga)
VALUES(51, 1, 'ZJAFE3208HJA10190', 3675)
go
INSERT INTO Carga(idEnvio, idCarga, vin, pesoCarga)
VALUES(52, 1, 'ZJAFE320XGJA10190', 3675)
go
/*
*****
* Test restricciones de integridad: ERROR
*****
*/
-- *****
-- Tabla Paises
-- *****
PRINT('Test PRIMARY KEY: se espera ERROR por repetir codPais')
INSERT INTO Paises(codPais, nomPais)
VALUES('1', 'Uruguay');
go
PRINT('Test PRIMARY KEY: se espera ERROR porque codPais no puede ser nulo')
INSERT INTO Paises(nomPais)
VALUES('Uruguay');
go
PRINT('Test UNIQUE nomPais: se espera ERROR porque se repite nomPais')
INSERT INTO Paises(codPais, nomPais)
VALUES('H', 'Alemania');
go
PRINT('Test UNIQUE nomPais: se espera ERROR porque se repite nomPais')
INSERT INTO Paises(codPais, nomPais)

```

```

VALUES('H', 'Suecia');
go
-- *****
-- Tabla Fabricantes
-- *****
PRINT('Test PRIMARY KEY: se espera ERROR por repetir codFab')
INSERT INTO Fabricantes(codFab, nomFab, dirFab, mailFab, cantEmp)
VALUES('AA', 'Audi', '7 John Knox Ln, 9QW', 'audi_factory1@audi.com', 570);
go
PRINT('Test PRIMARY KEY: se espera ERROR porque codFab no puede ser nulo')
INSERT INTO Fabricantes(nomFab, dirFab, mailFab, cantEmp)
VALUES('Renault', '7 John Knox Ln, 9QW', 'renault_factory1@renault.com', 570);
go
PRINT('Test UNIQUE mailFab: se espera ERROR porque se repite mailFab')
INSERT INTO Fabricantes(codFab, nomFab, dirFab, mailFab, cantEmp)
VALUES('XA', 'Renault', '7 John Knox Ln, 9QW', 'audi_factory1@audi.com', 570);
GO
PRINT('Test CHECK cantEmp > 0: se espera ERROR porque cantEmp tiene que ser
mayor a cero')
INSERT INTO Fabricantes(codFab, nomFab, dirFab, mailFab, cantEmp)
VALUES('XA', 'Renault', '7 John Knox Ln, 9QW', 'renault_factory1@renault.com',
0);
go
-- *****
-- Tabla Plantas
-- *****
PRINT('Test PRIMARY KEY: se espera ERROR por repetir codPlan y codFab')
INSERT INTO Plantas(codPlan, codFab, nomPlan, dirPlan, mailPlan, codPais)
VALUES('1', 'AA', 'Audi USA Motors', '38 Douglas Rd', 'audiusa@motors.com',
'4');
go
PRINT('Test PRIMARY KEY: se espera ERROR porque codFab no puede ser nulo')
INSERT INTO Plantas(codPlan, nomPlan, dirPlan, mailPlan, codPais)
VALUES('5', 'Audi USA Motors', '38 Douglas Rd', 'audiusa@motors.com', '4');
go
PRINT('Test PRIMARY KEY: se espera ERROR porque codPlan no puede ser nulo')
INSERT INTO Plantas(codFab, nomPlan, dirPlan, mailPlan, codPais)
VALUES('5', 'Audi USA Motors', '38 Douglas Rd', 'audiusa@motors.com', '4');
go
PRINT('Test FOREIGN KEY: se espera ERROR porque no existe el codFab')
INSERT INTO Plantas(codPlan, codFab, nomPlan, dirPlan, mailPlan, codPais)
VALUES('6', 'AC', 'Audi USA Motors', '38 Douglas Rd', 'audiusa@motors.com',
'4');
go
PRINT('Test FOREIGN KEY: se espera ERROR porque no existe el codPais')
INSERT INTO Plantas(codPlan, codFab, nomPlan, dirPlan, mailPlan, codPais)
VALUES('6', 'AA', 'Audi USA Motors', '38 Douglas Rd', 'audiusa@motors.com',
'X');
go
PRINT('Test UNIQUE mailPlan: se espera ERROR porque se repite mailPlan')
INSERT INTO Plantas(codPlan, codFab, nomPlan, dirPlan, mailPlan, codPais)
VALUES('6', 'AA', 'Audi USA Motors', '38 Douglas Rd', 'audi@usa.com', '4');
go
-- *****
-- Tabla Vehiculos
-- *****
PRINT('Test PRIMARY KEY: se espera ERROR por repetir vin')
INSERT INTO Vehiculos(vin, modelo, color, peso, características, codPais,
codFab)

```

```

VALUES ('9JBFE320XHJB10190', 'IPACE CONCEPT', 'gris', 3500, 'Frenos ABS, Aire
Acondicionado y tapizado de cuero', '9', 'JB')
go
PRINT('Test PRIMARY KEY: se espera ERROR porque vin no puede ser nulo')
INSERT INTO Vehiculos(modelo, color, peso, características, codPais, codFab)
VALUES ('IPACE CONCEPT', 'gris', 3500, 'Frenos ABS, Aire Acondicionado y
tapizado de cuero', '9', 'JB')
go
PRINT('Test FOREIGN KEY: se espera ERROR porque no existe el codPais')
INSERT INTO Vehiculos(vin, modelo, color, peso, características, codPais,
codFab)
VALUES ('9JBFE320XHJB10191', 'IPACE CONCEPT', 'gris', 3500, 'Frenos ABS, Aire
Acondicionado y tapizado de cuero', 'L', 'JB')
go
PRINT('Test FOREIGN KEY: se espera ERROR porque no existe el codFab')
INSERT INTO Vehiculos(vin, modelo, color, peso, características, codPais,
codFab)
VALUES ('9JBFE320XHJB10191', 'IPACE CONCEPT', 'gris', 3500, 'Frenos ABS, Aire
Acondicionado y tapizado de cuero', '9', 'JC')
go
-- *****
-- Tabla Envios
-- *****
PRINT('Test IDENTITY: se espera ERROR porque no se puede insertar en un campo
AUTONUMERADO como idEnvio')
INSERT INTO Envios(idEnvio, fchEnvio, pesoEnvio, oriEnvio, desEnvio)
VALUES (1, '20161008', 3675, 'Z', 'K');
go
PRINT('Test FOREIGN KEY: se espera ERROR porque no existe el oriEnvio')
INSERT INTO Envios(fchEnvio, pesoEnvio, oriEnvio, desEnvio)
VALUES ('20161008', 3675, 'X', 'K');
go
PRINT('Test FOREIGN KEY: se espera ERROR porque no existe el desEnvio')
INSERT INTO Envios(fchEnvio, pesoEnvio, oriEnvio, desEnvio)
VALUES ('20161008', 3675, 'Z', 'X');
go
-- *****
-- Tabla Carga
-- *****
PRINT('Test PRIMARY KEY: se espera ERROR por repetir idEnvio e idCarga')
INSERT INTO Carga(idEnvio, idCarga, vin, pesoCarga)
VALUES(50, 1, 'ZJAFE320XGJA10190', 3675)
go
PRINT('Test PRIMARY KEY: se espera ERROR porque idEnvio no puede ser nulo')
INSERT INTO Carga(idCarga, vin, pesoCarga)
VALUES(53, 'ZJAFE320XGJA10190', 3675)
go
PRINT('Test PRIMARY KEY: se espera ERROR porque idCarga no puede ser nulo')
INSERT INTO Carga(idEnvio, vin, pesoCarga)
VALUES(52, 'ZJAFE320XGJA10190', 3675)
go
PRINT('Test FOREIGN KEY: se espera ERROR porque no existe el vin')
INSERT INTO Carga(idEnvio, idCarga, vin, pesoCarga)
VALUES(52, 53, 'ZJAFE320XGJA10191', 3675)
go
-- *****
-- Insertar datos de cargas para que la consulta D) no sea vacía
-- *****
CREATE PROCEDURE sp_insertar_datos_query_d
@idEnvio INT,

```

```

@desdeidCarga INT,
@vin CHARACTER(17),
@pesoCarga NUMERIC(12,2),
@cantCargas INT
AS
BEGIN
DECLARE @ite INT
SET @ite = @desdeidCarga
WHILE (@ite <= @cantCargas)
BEGIN
INSERT INTO Carga(idEnvio, idCarga, vin, pesoCarga)
VALUES(@idEnvio, @ite, @vin, @pesoCarga)
SET @ite = @ite + 1
END
END;
go
-- Crear envío
DECLARE @pesoTotal NUMERIC(12,2)
SELECT @pesoTotal = peso*1.05*600 FROM Vehiculos WHERE vin =
'12AFE3204G2A10190'
INSERT INTO Envios(fchEnvio, pesoEnvio, oriEnvio, desEnvio)
VALUES ('20170408', @pesoTotal, '1', '2');
go
-- Insertar cargas al envío creado
DECLARE @idE INT
DECLARE @peso NUMERIC(12,2)
SELECT @idE = idEnvio FROM Envios WHERE fchEnvio = '20170408'
SELECT @peso = peso*1.05 FROM Vehiculos WHERE vin = '12AFE3204G2A10190'
EXEC sp_insertar_datos_query_d @idE, 1, '12AFE3204G2A10190', @peso, 600
go
-- *****
-- Insertar datos para que la consulta F) no sea vacía
-- *****
-- Crear país Holanda en Tabla Países
INSERT INTO Países(codPaís, nomPaís)
VALUES('H', 'Holanda');
go
-- Crear un envío
DECLARE @pesoTotal NUMERIC(12,2)
SELECT @pesoTotal = peso*1.05*20 FROM Vehiculos WHERE vin = 'WDAFE3209GDA10190'
INSERT INTO Envios(fchEnvio, pesoEnvio, oriEnvio, desEnvio)
VALUES ('20170308', @pesoTotal, '1', 'H');
go
-- Crear cargas (20) para el envío creado
DECLARE @idE INT
DECLARE @peso NUMERIC(12,2)
SELECT @idE = idEnvio FROM Envios WHERE fchEnvio = '20170308'
SELECT @peso = peso*1.05 FROM Vehiculos WHERE vin = 'WDAFE3209GDA10190'
EXEC sp_insertar_datos_query_d @idE, 1, 'WDAFE3209GDA10190', @peso, 20
go
-- *****
-- Insertar datos para que la consulta H) no sea vacía
-- *****
-- Vehículo sin envío
INSERT INTO Vehiculos(vin, modelo, color, peso, características, codPaís,
codFab)
VALUES ('WDAFE3200FDA10190', 'E500', 'gris', 3300, 'Frenos ABS, Aire
Acondicionado y tapizado de cuero', 'W', 'DA')
go
-- Vehículo enviado (dos años después de su fabricación)

```

```

INSERT INTO Vehiculos(vin, modelo, color, peso, características, codPais,
codFab)
VALUES ('WDAFE3202EDA10190', 'E500', 'gris', 3300, 'Frenos ABS, Aire
Acondicionado y tapizado de cuero', 'W', 'DA')
go
-- Crear un envío
DECLARE @pesoTotal NUMERIC(12,2)
SELECT @pesoTotal = peso*1.05*1 FROM Vehiculos WHERE vin = 'WDAFE3202EDA10190'
INSERT INTO Envios(fchEnvio, pesoEnvio, oriEnvio, desEnvio)
VALUES ('20160910', @pesoTotal, 'W', '9');
go
-- Crear cargas para el envío creado
DECLARE @idE INT
DECLARE @peso NUMERIC(12,2)
SELECT @idE = idEnvio FROM Envios WHERE fchEnvio = '20160910'
SELECT @peso = peso*1.05 FROM Vehiculos WHERE vin = 'WDAFE3202EDA10190'
EXEC sp_insertar_datos_query_d @idE, 1, 'WDAFE3202EDA10190', @peso, 1
go
-- *****
-- Insertar datos para que la consulta I) no sea vacía
-- *****
-- Insertar Vehículo
INSERT INTO Vehiculos(vin, modelo, color, peso, características, codPais,
codFab)
VALUES ('JHAFE3150GHA10190', 'Avancier', 'rojo', 4300, 'Frenos ABS, Aire
Acondicionado y tapizado de cuero', 'J', 'HA')
go
-- Crear envío para mes 04
DECLARE @pesoTotal NUMERIC(12,2)
SELECT @pesoTotal = peso*1.05*150 FROM Vehiculos WHERE vin =
'JHAFE3150GHA10190'
INSERT INTO Envios(fchEnvio, pesoEnvio, oriEnvio, desEnvio)
VALUES ('20160420', @pesoTotal, 'J', 'W');
-- Insertar cargas al envío creado (150 autos)
DECLARE @idE INT
DECLARE @peso NUMERIC(12,2)
SELECT @idE = idEnvio FROM Envios WHERE fchEnvio = '20160420'
SELECT @peso = peso*1.05 FROM Vehiculos WHERE vin = 'JHAFE3150GHA10190'
EXEC sp_insertar_datos_query_d @idE, 1, 'JHAFE3150GHA10190', @peso, 150
go
-- Crear envío para mes 08
DECLARE @pesoTotal NUMERIC(12,2)
SELECT @pesoTotal = peso*1.05*150 FROM Vehiculos WHERE vin =
'JHAFE3150GHA10190'
INSERT INTO Envios(fchEnvio, pesoEnvio, oriEnvio, desEnvio)
VALUES ('20160825', @pesoTotal, 'J', 'W');
go
-- Insertar cargas al envío creado (150 autos)
DECLARE @idE INT
DECLARE @peso NUMERIC(12,2)
SELECT @idE = idEnvio FROM Envios WHERE fchEnvio = '20160825'
SELECT @peso = peso*1.05 FROM Vehiculos WHERE vin = 'JHAFE3150GHA10190'
EXEC sp_insertar_datos_query_d @idE, 1, 'JHAFE3150GHA10190', @peso, 110
go
-- Crear envío para mes 12
DECLARE @pesoTotal NUMERIC(12,2)
SELECT @pesoTotal = peso*1.05*15 FROM Vehiculos WHERE vin = 'JHAFE3150GHA10190'
INSERT INTO Envios(fchEnvio, pesoEnvio, oriEnvio, desEnvio)
VALUES ('20161221', @pesoTotal, 'J', 'W');
go

```

```
-- Insertar cargas al envío creado (menos de 20 autos = 15)
DECLARE @idE INT
DECLARE @peso NUMERIC(12,2)
SELECT @idE = idEnvio FROM Envios WHERE fchEnvio = '20161221'
SELECT @peso = peso*1.05 FROM Vehiculos WHERE vin = 'JHAFE3150GHA10190'
EXEC sp_insertar_datos_query_d @idE, 1, 'JHAFE3150GHA10190', @peso, 15
go
```

## Punto 4)

Para cada solución del punto 4 , se ingresaron pruebas para su verificación .

**4 a) Crear una función almacenada que reciba como parámetro un número de VIN completo y valide el dígito verificador, si dicho dígito es correcto la función debe retornar el texto 'OK', si es incorrecto, debe retornar todo el número de VIN pero incluyendo el dígito verificador correcto.**

```
-- Función auxiliar: retorna el valor numérico asociado
CREATE FUNCTION funct_valor_num_asociado_vin
(@letravin CHARACTER(1))
RETURNS INT
AS
BEGIN
DECLARE @ret INT

IF(@letravin = 'A')
BEGIN
SET @ret = 1
END
ELSE IF(@letravin = 'B')
BEGIN
SET @ret = 2
END
ELSE IF(@letravin = 'C')
BEGIN
SET @ret = 3
END
ELSE IF(@letravin = 'D')
BEGIN
SET @ret = 4
END
ELSE IF(@letravin = 'E')
BEGIN
SET @ret = 5
END
ELSE IF(@letravin = 'F')
BEGIN
SET @ret = 6
END
ELSE IF(@letravin = 'G')
BEGIN
SET @ret = 7
END
```

```

ELSE IF(@letravin = 'H')
    BEGIN
        SET @ret = 8
    END
ELSE IF(@letravin = 'J')
    BEGIN
        SET @ret = 1
    END
ELSE IF(@letravin = 'K')
    BEGIN
        SET @ret = 2
    END
ELSE IF(@letravin = 'L')
    BEGIN
        SET @ret = 3
    END
ELSE IF(@letravin = 'M')
    BEGIN
        SET @ret = 4
    END
ELSE IF(@letravin = 'N')
    BEGIN
        SET @ret = 5
    END
ELSE IF(@letravin = 'P')
    BEGIN
        SET @ret = 7
    END
ELSE IF(@letravin = 'R')
    BEGIN
        SET @ret = 9
    END
ELSE IF(@letravin = 'S')
    BEGIN
        SET @ret = 2
    END
ELSE IF(@letravin = 'T')
    BEGIN
        SET @ret = 3
    END
ELSE IF(@letravin = 'U')
    BEGIN
        SET @ret = 4
    END
ELSE IF(@letravin = 'V')
    BEGIN
        SET @ret = 5
    END
ELSE IF(@letravin = 'W')
    BEGIN
        SET @ret = 6
    END
ELSE IF(@letravin = 'X')
    BEGIN
        SET @ret = 7
    END
ELSE IF(@letravin = 'Y')
    BEGIN
        SET @ret = 8
    END

```



```

ELSE IF(@letravin = 'Z')
    BEGIN
        SET @ret = 9
    END
ELSE
    BEGIN
        SET @ret = @letravin
    END
RETURN @ret
END;

-- Función auxiliar: retorna el factor multiplicador
CREATE FUNCTION funct_factor_multiplicador_vin
(@posicionvin INT)
RETURNS INT
AS
BEGIN
    DECLARE @ret INT
    IF(@posicionvin IN (1,11))
        BEGIN
            SET @ret = 8
        END
    ELSE IF(@posicionvin IN (2,12))
        BEGIN
            SET @ret = 7
        END
    ELSE IF(@posicionvin IN (3,13))
        BEGIN
            SET @ret = 6
        END
    ELSE IF(@posicionvin IN (4,14))
        BEGIN
            SET @ret = 5
        END
    ELSE IF(@posicionvin IN (5,15))
        BEGIN
            SET @ret = 4
        END
    ELSE IF(@posicionvin IN (6,16))
        BEGIN
            SET @ret = 3
        END
    ELSE IF(@posicionvin IN (7,17))
        BEGIN
            SET @ret = 2
        END
    ELSE IF(@posicionvin = 8)
        BEGIN
            SET @ret = 10
        END
    ELSE IF(@posicionvin = 10)
        BEGIN
            SET @ret = 9
        END
    RETURN @ret
END;

-- Función principal: valida el dígito verificador
CREATE FUNCTION funct_validar_digitoverificador_vin
(@vin CHARACTER(17))

```

```

RETURNS CHARACTER(17)
AS
BEGIN
DECLARE @ret CHARACTER(17)
DECLARE @valAsociado INT
DECLARE @factMultiplicador INT
DECLARE @producto INT
DECLARE @sumaProductos INT
DECLARE @resto VARCHAR(2)
DECLARE @dv CHAR(1)
SET @sumaProductos = 0
SET @dv = SUBSTRING(@vin,9,1)
SET @ret = 'OK'

-- Recorrer VIN
DECLARE @ite INT
SET @ite = 1
WHILE (@ite < 18)
    BEGIN
        IF(@ite <> 9)
            BEGIN
                SET @valAsociado =
dbo.funct_valor_num_asociado_vin(SUBSTRING(@vin,@ite,1))
                SET @factMultiplicador = dbo.funct_factor_multiplicador_vin(@ite)
                SET @producto = @valAsociado * @factMultiplicador
                SET @sumaProductos = @sumaProductos + @producto
            END
        SET @ite = @ite + 1
    END

-- Obtener dígito verificador
SET @resto = CAST(@sumaProductos % 11 AS VARCHAR(2))

-- Test
IF(@resto = '10' AND @dv <> 'X')
    BEGIN
        SET @ret = SUBSTRING(@vin,1,8) + 'X' + SUBSTRING(@vin,10,8)
    END
IF (@resto <> @dv AND @resto <> '10')
    BEGIN
        SET @ret = SUBSTRING(@vin,1,8) + @resto + SUBSTRING(@vin,10,8)
    END
RETURN @ret
END;

```

### Tests:

```

-- Test OK
DECLARE @output CHARACTER(17)
SET @output = dbo.funct_validar_digitoverificador_vin('1AAFE3203FAA10190');
PRINT @output

```

OUTPUT: OK

```

-- Test devuelve VIN corregido
DECLARE @output CHARACTER(17)
SET @output = dbo.funct_validar_digitoverificador_vin('1AAFE3209FAA10190');
PRINT @output

```

OUTPUT: 1AAFE3203FAA10190

**4 b) Crear una función que reciba como parámetro un VIN y retorne el año del modelo de dicho vehículo.**

**Nota: se asume que los autos transportados por la empresa ROVIA INC fueron modelos de autos que se hicieron a partir del año 2010.**

```
CREATE FUNCTION funct_aniodelmodelo_vin
(@vin CHARACTER(17))
RETURNS INT
AS
BEGIN
DECLARE @ret INT
DECLARE @digitoAnio CHARACTER(1)
SET @digitoAnio = SUBSTRING(@vin,10,1)
IF(@digitoAnio = 'A')
BEGIN
SET @ret = 2010
END
IF(@digitoAnio = 'B')
BEGIN
SET @ret = 2011
END
IF(@digitoAnio = 'C')
BEGIN
SET @ret = 2012
END
IF(@digitoAnio = 'D')
BEGIN
SET @ret = 2013
END
IF(@digitoAnio = 'E')
BEGIN
SET @ret = 2014
END
IF(@digitoAnio = 'F')
BEGIN
SET @ret = 2015
END
IF(@digitoAnio = 'G')
BEGIN
SET @ret = 2016
END
IF(@digitoAnio = 'H')
BEGIN
SET @ret = 2017
END
IF(@digitoAnio = 'J')
BEGIN
SET @ret = 2018
END
IF(@digitoAnio = 'K')
BEGIN
SET @ret = 2019
```

```

        END
    IF(@digitoAnio = 'L')
        BEGIN
            SET @ret = 2020
        END
    IF(@digitoAnio = 'M')
        BEGIN
            SET @ret = 2021
        END
    IF(@digitoAnio = 'N')
        BEGIN
            SET @ret = 2022
        END
    IF(@digitoAnio = 'P')
        BEGIN
            SET @ret = 2023
        END
    IF(@digitoAnio = 'R')
        BEGIN
            SET @ret = 2024
        END
    IF(@digitoAnio = 'S')
        BEGIN
            SET @ret = 2025
        END
    IF(@digitoAnio = 'T')
        BEGIN
            SET @ret = 2026
        END
    IF(@digitoAnio = 'V')
        BEGIN
            SET @ret = 2027
        END
    IF(@digitoAnio = 'W')
        BEGIN
            SET @ret = 2028
        END
    IF(@digitoAnio = 'X')
        BEGIN
            SET @ret = 2029
        END
    IF(@digitoAnio = 'Y')
        BEGIN
            SET @ret = 2030
        END
    IF(@digitoAnio = '1')
        BEGIN
            SET @ret = 2031
        END
    IF(@digitoAnio = '2')
        BEGIN
            SET @ret = 2032
        END
    IF(@digitoAnio = '3')
        BEGIN
            SET @ret = 2033
        END
    IF(@digitoAnio = '4')
        BEGIN
            SET @ret = 2034

```

```

        END
    IF(@digitoAnio = '5')
        BEGIN
            SET @ret = 2035
        END
    IF(@digitoAnio = '6')
        BEGIN
            SET @ret = 2036
        END
    IF(@digitoAnio = '7')
        BEGIN
            SET @ret = 2037
        END
    IF(@digitoAnio = '8')
        BEGIN
            SET @ret = 2038
        END
    IF(@digitoAnio = '9')
        BEGIN
            SET @ret = 2039
        END
    RETURN @ret
END;

```

### Tests:

```

-- Test 2015
DECLARE @output INT
SET @output = dbo.funct_anioDelmodelo_vin('1AAFE3203FAA10190');
PRINT @output

```

OUTPUT: 2015

```

-- Test 2016
DECLARE @output INT
SET @output = dbo.funct_anioDelmodelo_vin('1AAFE3201GAA10190');
PRINT @output

```

OUTPUT: 2016

### 4 c) Dado un rango de fechas, retornar la cantidad de vehículos que recibió el país que más vehículos recibió en dicho rango.

```

CREATE FUNCTION funct_fechas_cantvehiculos_pais_maxcant
(@fchIni DATE, @fchFin DATE)
RETURNS INT
AS
BEGIN
    DECLARE @ret INT
    SELECT @ret = COUNT(V.vin)
    FROM Vehiculos V, Envios E, Carga C
    WHERE V.vin = C.vin
    AND E.idEnvio = C.idEnvio
    AND E.fchEnvio BETWEEN @fchIni AND @fchFin
    GROUP BY E.desEnvio
    HAVING COUNT(V.vin) >= ALL(SELECT COUNT(V2.vin)
                                FROM Vehiculos V2, Envios E2, Carga C2
                                WHERE V2.vin = C2.vin

```

```

AND E2.idEnvio = C2.idEnvio
AND E2.fchEnvio BETWEEN @fchIni AND @fchFin
GROUP BY E2.desEnvio)

RETURN @ret
END;

```

## Test

```

-- Test
DECLARE @fchIni DATE
DECLARE @fchFin DATE
DECLARE @output INT
SET @fchIni = '20150101'
SET @fchFin = '20170620'
SET @output = dbo.funct_fechas_cantvehiculos_pais_maxcant(@fchIni, @fchFin)
PRINT @output

```

OUTPUT: 606

**4 d) Mediante un procedimiento almacenado que reciba un rango de fechas de envío, cambiar el país de destino del envío a todos los vehículos cuyo país de fabricación del vehículo sea el mismo que el país de destino del envío, en ese caso crear un país de código '#' que tenga como descripción 'LOCAL' y poner ese dato como destino.**

```

CREATE PROCEDURE sp_fechasenvio_cambiapaisdestino
@fchIni DATE,
@fchFin DATE
AS
BEGIN
UPDATE Envios
SET desEnvio = '#'
WHERE fchEnvio BETWEEN @fchIni AND @fchFin
AND desEnvio IN (SELECT E.desEnvio
                  FROM Vehiculos V, Carga C, Envios E
                  WHERE V.vin = C.vin
                  AND C.idEnvio = E.idEnvio
                  AND E.fchEnvio BETWEEN @fchIni AND @fchFin
                  AND V.codPais = E.desEnvio)
END;

```

## Tests

```

-- Test procedure
-- Insertar país local en la tabla Paises
INSERT INTO Paises(codPais, nomPais)
VALUES('#', 'LOCAL');

```

OUTPUT: (1 row(s) affected)

```

-- Insertar un envío cuyo destino final sea el país que lo fabrico
INSERT INTO Envios(fchEnvio, pesoEnvio, oriEnvio, desEnvio)
VALUES ('20151008', 0, '2', '1');

```

OUTPUT: (1 row(s) affected)

```
-- Insertar Carga (1 auto para el envío creado)
INSERT INTO Carga(idEnvio, idCarga, vin, pesoCarga)
VALUES((SELECT idEnvio FROM Envios
        WHERE fchEnvio = '20151008'
        AND oriEnvio = '2'
        AND desEnvio = '1'), 1, '16AFE3201F6A10190', 3800)
```

OUTPUT: (1 row(s) affected)

```
-- Ejecutar procedure sin cambios por no coincidir fechas
EXEC sp_fechasenvio_cambiapaisdestino '20170101', '20170105'
```

OUTPUT: (0 row(s) affected)

```
-- Ejecutar procedure con cambios
EXEC sp_fechasenvio_cambiapaisdestino '20150101', '20151231'
```

OUTPUT: (1 row(s) affected)

```
SELECT * FROM Envios E WHERE E.desEnvio = '#';
```

OUTPUT:

idEnvio	fchEnvio	pesoEnvio	oriEnvio	desEnvio
61	2015-10-08 00:00:00.000	0.00	2	#

**4 e) Dado un rango de fechas, implementar una función que retorne el nombre del país al cual se le envió la menor cantidad de vehículos en dicho rango.**

```
CREATE FUNCTION funct_fechas_nompais_menorcantvehic
(@fchIni DATE, @fchFin DATE)
RETURNS VARCHAR(30)
AS
BEGIN
    DECLARE @ret VARCHAR(30)
    SELECT @ret = P.nomPais
    FROM Países P, Envios E
    WHERE P.codPais = E.desEnvio
    AND E.fchEnvio BETWEEN @fchIni AND @fchFin
    GROUP BY P.nomPais
    HAVING COUNT(E.idEnvio) <= ALL(SELECT COUNT(E2.idEnvio)
                                   FROM Países P2, Envios E2
                                   WHERE P2.codPais = E2.desEnvio
                                   AND E2.fchEnvio BETWEEN @fchIni AND @fchFin
                                   GROUP BY P2.nomPais)

    RETURN @ret
END;
```

## Tests

```
-- Test
DECLARE @output VARCHAR(30)
SET @output = dbo.funct_fechas_nompais_menorcantvehic('20100101', GETDATE())
PRINT @output
```

OUTPUT: Holanda

```
-- Durante 2015
DECLARE @output VARCHAR(30)
SET @output = dbo.funct_fechas_nompais_menorcantvehic('20150101','20151231')
PRINT @output
```

OUTPUT: Inglaterra

```
-- Durante 2016
DECLARE @output VARCHAR(30)
SET @output = dbo.funct_fechas_nompais_menorcantvehic('20160101','20161231')
PRINT @output
```

OUTPUT: Suecia

**4 f) Realizar un procedimiento almacenado que reciba un código de fabricante y retorne con parámetro de salida la cantidad de plantas que tiene y la cantidad de modelos de vehículos diferentes que produce.**

```
-- Con el codFab
CREATE PROCEDURE sp_codfab_cantplantas_cantmodelosvehic
@codFab CHARACTER(2),
@cantPlantas INT OUTPUT,
@cantModelosVehic INT OUTPUT
AS
BEGIN
SELECT @cantPlantas = COUNT(DISTINCT(P.codPlan)), @cantModelosVehic =
COUNT(DISTINCT(V.modelo))
FROM Fabricantes F, Plantas P, Vehiculos V
WHERE F.codFab = V.codFab
AND F.codFab = P.codFab
AND F.codFab = @codFab
GROUP BY F.codFab
END;
```

## Tests

```
-- Test para codFab = AA
DECLARE @out1 INT
DECLARE @out2 INT
EXEC sp_codfab_cantplantas_cantmodelosvehic 'AA', @out1 OUTPUT, @out2 OUTPUT
PRINT 'Cant_plantas: ' + CAST(@out1 AS CHARACTER(2)) + '|' Cant_modelos: ' +
CAST(@out2 AS CHARACTER(2))
```

OUTPUT: Cant\_plantas: 2 | Cant\_modelos: 2

```
-- Test para codFab = JA
DECLARE @out1 INT
DECLARE @out2 INT
EXEC sp_codfab_cantplantas_cantmodelosvehic 'JA', @out1 OUTPUT, @out2 OUTPUT
PRINT 'Cant_plantas: ' + CAST(@out1 AS CHARACTER(2)) + '|' Cant_modelos: ' +
CAST(@out2 AS CHARACTER(2))
```

OUTPUT: Cant\_plantas: 2 | Cant\_modelos: 1



**Misma función pero ingresando el nombre de un fabricante sin distinguir entre sus divisiones, para retornar la cantidad de plantas y la cantidad de modelos de la marca "Audi" por ejemplo:**

```
-- Con el codFab
CREATE PROCEDURE sp_nomfab_cantplantas_cantmodelosvehic
@nomFab VARCHAR(30),
@cantPlantas INT OUTPUT,
@cantModelosVehic INT OUTPUT
AS
BEGIN
SELECT @cantPlantas = COUNT(DISTINCT(P.codPlan)), @cantModelosVehic =
COUNT(DISTINCT(V.modelo))
FROM Fabricantes F, Plantas P, Vehiculos V
WHERE F.codFab = V.codFab
AND F.codFab = P.codFab
AND F.nomFab = @nomFab
GROUP BY F.nomFab
END;

-- Test para 'Audi'
DECLARE @out1 INT
DECLARE @out2 INT
EXEC sp_nomfab_cantplantas_cantmodelosvehic 'Audi', @out1 OUTPUT, @out2 OUTPUT
PRINT 'Cant_plantas: ' + CAST(@out1 AS CHARACTER(2)) + '|' Cant_modelos: ' +
CAST(@out2 AS CHARACTER(2))
```

OUTPUT: Cant\_plantas: 3 | Cant\_modelos: 2

## Punto 5)

**Para las soluciones del punto 5 se ingresaron pruebas al final que validan la solución planteada.**

**5 a) Crear un trigger que valide un número de VIN al ingresar un nuevo vehículo, debe utilizar la función creada en el punto 4a.**

```
-- Función auxiliar: impide VIN con caracteres I, O, Q o Ñ
CREATE FUNCTION funct_impedir_caracteres_vin
(@vin CHARACTER(17))
RETURNS BIT
AS
BEGIN
DECLARE @ret BIT
SET @ret = 1
-- Recorrer VIN
DECLARE @ite INT
SET @ite = 1
WHILE (@ite < 18 AND @ret = 1)
BEGIN
IF(SUBSTRING(@vin,@ite,1) IN ('I','O','Q','Ñ'))
BEGIN
SET @ret = 0

```

```

        END
SET @ite = @ite + 1
    END
RETURN @ret
END;

-- Trigger
CREATE TRIGGER trigger_validar_vin_insert_vehiculo
ON Vehiculos
INSTEAD OF INSERT
AS
BEGIN
    DECLARE @vin CHARACTER(17)
    DECLARE @codPais CHARACTER(1)
    DECLARE @codFab CHARACTER(2)
    DECLARE @codFabPais CHARACTER(2)
    DECLARE @caracteres BIT
    SET @codPais = ''
    SET @codFab = ''
    SET @codFabPais = ''
    SET @caracteres = 1
    -- Validar caracteres no permitidos
    SELECT @caracteres = dbo.funct_impedir_caracteres_vin(I.vin)
    FROM inserted I
    IF (@caracteres = 1)
    BEGIN
        SELECT @vin = dbo.funct_validar_digitoverificador_vin(I.vin)
        FROM inserted I
        -- Check codPais
        SELECT @codPais = (I.codPais)
        FROM inserted I, Países P
        WHERE I.codPais = P.codPais
        -- Check codFab
        SELECT @codFab = (I.codFab)
        FROM inserted I, Fabricantes F
        WHERE I.codFab = F.codFab
        -- Check codPais y codFab
        SELECT @codFabPais = (I.codFab)
        FROM inserted I, Plantas P
        WHERE I.codFab = P.codFab
        AND I.codPais = P.codPais
        IF (@codPais = '')
        BEGIN
            PRINT 'No existe el [codPais] para el vehículo que se quiere
ingresar'
        END
        ELSE IF (@codFab = '')
        BEGIN
            PRINT 'No existe el [codFab] para el vehículo que se quiere
ingresar'
        END
        ELSE IF (@codFabPais = '')
        BEGIN
            PRINT 'No existe la planta para el vehículo que se quiere
ingresar'
        END
        ELSE IF (@vin <> 'OK')
        BEGIN
            INSERT INTO Vehiculos

```

```

                SELECT @vin, I.modelo, I.color, I.peso, I.caracteristicas,
I.codPais, I.codFab
                FROM INSERTED I
                PRINT 'VEHICULO INGRESADO (VIN CORREGIDO!)'
                END
            ELSE IF (@vin = 'OK')
            BEGIN
                INSERT INTO Vehiculos
                SELECT I.vin, I.modelo, I.color, I.peso, I.caracteristicas,
I.codPais, I.codFab
                FROM INSERTED I
                PRINT 'VEHICULO INGRESADO (VIN OK)'
                END
            END
        ELSE
        BEGIN
            PRINT 'CARACTERES EN vin NO PERMITIDOS'
        END
    END;

```

## Tests

```

-- Test OK, vin OK
INSERT INTO Vehiculos(vin, modelo, color, peso, caracteristicas, codPais,
codFab)
VALUES ('1AAAL3205EAA10190', 'A4', 'gris', 2500, 'Frenos ABS, Aire
Acondicionado y tapizado de cuero', '1', 'AA')

```

OUTPUT:

```

(1 row(s) affected)
VEHICULO INGRESADO (VIN OK)

```

```

(1 row(s) affected)

```

```

-- Test OK, vin CORREGIDO
INSERT INTO Vehiculos(vin, modelo, color, peso, caracteristicas, codPais,
codFab)
VALUES ('1AAAL3205EAA10190', 'A4', 'gris', 2500, 'Frenos ABS, Aire
Acondicionado y tapizado de cuero', '1', 'AA')

```

OUTPUT:

```

(1 row(s) affected)
VEHICULO INGRESADO (VIN CORREGIDO!)

```

```

(1 row(s) affected)

```

```

-- Test ERROR, caracter no permitido I
INSERT INTO Vehiculos(vin, modelo, color, peso, caracteristicas, codPais,
codFab)
VALUES ('1AIAL3205EAA10190', 'A4', 'gris', 2500, 'Frenos ABS, Aire
Acondicionado y tapizado de cuero', '1', 'AA')

```

OUTPUT:

```

CARACTERES EN vin NO PERMITIDOS

```

(1 row(s) affected)

```
-- Test ERROR, caracter no permitido 0
INSERT INTO Vehiculos(vin, modelo, color, peso, características, codPais,
codFab)
VALUES ('1AOAL3205EAA10190', 'A4', 'gris', 2500, 'Frenos ABS, Aire
Acondicionado y tapizado de cuero', '1', 'AA')
```

OUTPUT:

CARACTERES EN vin NO PERMITIDOS

(1 row(s) affected)

```
-- Test ERROR, caracter no permitido Q
INSERT INTO Vehiculos(vin, modelo, color, peso, características, codPais,
codFab)
VALUES ('1AQAL3205EAA10190', 'A4', 'gris', 2500, 'Frenos ABS, Aire
Acondicionado y tapizado de cuero', '1', 'AA')
```

OUTPUT:

CARACTERES EN vin NO PERMITIDOS

(1 row(s) affected)

```
-- Test ERROR, caracter no permitido Ñ
INSERT INTO Vehiculos(vin, modelo, color, peso, características, codPais,
codFab)
VALUES ('1AÑAL3205EAA10190', 'A4', 'gris', 2500, 'Frenos ABS, Aire
Acondicionado y tapizado de cuero', '1', 'AA')
```

OUTPUT:

CARACTERES EN vin NO PERMITIDOS

(1 row(s) affected)

```
-- Test ERROR, no existe codPais
INSERT INTO Vehiculos(vin, modelo, color, peso, características, codPais,
codFab)
VALUES ('XAAAL3204EAA10190', 'A4', 'verde', 2500, 'Frenos ABS, Aire
Acondicionado y tapizado de cuero', 'X', 'AA')
```

OUTPUT:

No existe el [codPais] para el vehículo que se quiere ingresar

(1 row(s) affected)

```
-- Test ERROR, no existe codFab
INSERT INTO Vehiculos(vin, modelo, color, peso, características, codPais,
codFab)
VALUES ('XAAAL3204EAA10190', 'A4', 'verde', 2500, 'Frenos ABS, Aire
Acondicionado y tapizado de cuero', '1', 'AC')
```

OUTPUT:

No existe el [codFab] para el vehículo que se quiere ingresar

(1 row(s) affected)

```
-- Test ERROR, no existe codFab para el codPais (no existe la planta)
INSERT INTO Vehiculos(vin, modelo, color, peso, características, codPais,
codFab)
VALUES ('XAAAL3204EAA10190', 'A4', 'verde', 2500, 'Frenos ABS, Aire
Acondicionado y tapizado de cuero', '2', 'AA')
```

OUTPUT:

No existe la planta para el vehículo que se quiere ingresar

(1 row(s) affected)

**5 b) Crear un trigger al ingresar una línea de carga, ponga el peso correcto de acuerdo al peso del vehículo mas el 5% de combustible y accesorios, también debe acumular el peso total del envío.**

```
CREATE TRIGGER trigger_peso_insert_carga
ON Carga
INSTEAD OF INSERT
AS
BEGIN
UPDATE Envios
SET pesoEnvio = pesoEnvio + (SELECT SUM(V.peso * 1.05)
                             FROM inserted I, Vehiculos V, Envios E
                             WHERE I.vin = V.vin
                             AND I.idEnvio = E.idEnvio)
FROM inserted I2, Envios E2
WHERE E2.idEnvio = I2.idEnvio;
INSERT INTO Carga(idEnvio, idCarga, vin, pesoCarga)
SELECT I3.idEnvio, I3.idCarga, I3.vin, (SELECT SUM(V2.peso * 1.05)
                                         FROM inserted I3, Vehiculos V2
                                         WHERE I3.vin = V2.vin)
FROM inserted I3, Vehiculos V2
WHERE I3.vin = V2.vin
END;
```

## Tests

```
-- Test OK
INSERT INTO Carga(idEnvio, idCarga, vin)
VALUES(1, 2, '16AFE3201F6A10190')
```

OUTPUT:

(1 row(s) affected)

(1 row(s) affected)

(1 row(s) affected)

```
INSERT INTO Carga(idEnvio, idCarga, vin)
VALUES(2, 2, '16AFE3208H6A10190')
```

OUTPUT:

(1 row(s) affected)

(1 row(s) affected)

(1 row(s) affected)

-- Test OK, ingresar 3 cargas más al idEnvio = 52

```
INSERT INTO Carga(idEnvio, idCarga, vin)
VALUES(52, 2, 'ZJAFE320XGJA10190')
```

OUTPUT:

(1 row(s) affected)

(1 row(s) affected)

(1 row(s) affected)

```
INSERT INTO Carga(idEnvio, idCarga, vin)
VALUES(52, 3, 'ZJAFE320XGJA10190')
```

OUTPUT:

(1 row(s) affected)

(1 row(s) affected)

(1 row(s) affected)

```
INSERT INTO Carga(idEnvio, idCarga, vin)
VALUES(52, 4, 'ZJAFE320XGJA10190')
```

OUTPUT:

(1 row(s) affected)

(1 row(s) affected)

(1 row(s) affected)

-- Ver cargas del idEnvio = 52

```
SELECT * FROM Carga WHERE idEnvio = 52;
```

OUTPUT:

idEnvio	idCarga	vin	pesoCarga
52	1	ZJAFE320XGJA10190	3675.00
52	2	ZJAFE320XGJA10190	3675.00
52	3	ZJAFE320XGJA10190	3675.00

**5 c) Definir un trigger que al ingresar un envío, si el país de origen es igual al país de destino no procese la línea.**

```
CREATE TRIGGER trigger_validar_origendestino_insert_envio
ON Envios
INSTEAD OF INSERT
AS
BEGIN
INSERT INTO Envios
SELECT I.fchEnvio, I.pesoEnvio, I.oriEnvio, I.desEnvio
FROM inserted I, inserted I2
WHERE I.idEnvio = I2.idEnvio
AND I.oriEnvio <> I2.desEnvio
END;
```

**Tests**

```
-- Test OK
INSERT INTO Envios(fchEnvio, pesoEnvio, oriEnvio, desEnvio)
VALUES ('20150315', 3990, '1', 'Z');
```

OUTPUT:

(1 row(s) affected)

(1 row(s) affected)

```
-- Test no procesa línea
INSERT INTO Envios(fchEnvio, pesoEnvio, oriEnvio, desEnvio)
VALUES ('20150315', 3990, '1', '1');
```

OUTPUT:

(0 row(s) affected)

(1 row(s) affected)

```
-- Ver tabla (el idEnvio = 62 corresponde a la inserción Test OK)
SELECT * FROM Envios WHERE fchEnvio = '20150315';
```

OUTPUT:

idEnvio	fchEnvio	pesoEnvio	oriEnvio	desEnvio
1	2015-03-15 00:00:00.000	7980.00	1	Z
21	2015-03-15 00:00:00.000	2940.00	2	J
62	2015-03-15 00:00:00.000	3990.00	1	Z

**5 d) Implementar un trigger que permita el borrado de uno o más envíos, para ello debe borrar en cascada todos los registros en las tablas relacionadas, se debe tener en cuenta los borrados múltiples.**

```
CREATE TRIGGER trigger_delete_envio
ON Envios
INSTEAD OF DELETE
AS
```

```

BEGIN
-- Borrar de tabla Carga
DELETE FROM Carga
WHERE idEnvio IN (SELECT D.idEnvio FROM deleted D)
-- Borrar de tabla Envio
DELETE FROM Envios
WHERE idEnvio IN (SELECT D2.idEnvio FROM deleted D2)
END;

```

## Tests

```

-- Test OK
-- Múltiple (borra las cuatro cargas del idEnvio = 52 en la tabla Carga y el
propio idEnvio = 52 en la tabla Envios)
DELETE FROM Envios
WHERE idEnvio = 52;

```

OUTPUT:

(4 row(s) affected)

(1 row(s) affected)

(1 row(s) affected)

```

-- Ver cargas del idEnvio = 52
SELECT *
FROM Carga
WHERE idEnvio = 52;

```

OUTPUT: (vacío)

```

-- Ver idEnvio = 52
SELECT *
FROM Envios
WHERE idEnvio = 52;

```

OUTPUT: (vacío)

## Punto 6)

**6 a) Mostrar el mayor peso, el menor peso, la mayor fecha y la menor fecha de los envíos de los años 2015 y 2016.**

```

SELECT YEAR(E.fchEnvio) AS Anio,
       MAX(E.pesoEnvio) AS Mayor_peso,
       MIN(E.pesoEnvio) AS Menor_peso,
       MAX(E.fchEnvio) AS Mayor_fecha,
       MIN(E.fchEnvio) AS Menor_fecha
FROM Envios E
WHERE YEAR(E.fchEnvio) = 2015 OR YEAR(E.fchEnvio) = 2016
GROUP BY YEAR(E.fchEnvio);

```



OUTPUT:

Anio	Mayor_peso	Menor_peso	Mayor_fecha	Menor_fecha
2015	7980.00	0.00	2015-11-15	2015-01-20
2016	677250.00	1050.00	2016-12-21	2016-01-11

**6 b) Mostrar para cada fabricante su nombre, la cantidad de vehículos enviados y el peso total de los mismos en el año 2016 (ordenar los resultados de mayor peso a menor peso).**

**Se devuelve la suma del peso de los vehículos (incluye el peso de accesorios y combustible). Se devuelve para cada nombre de fabricante no distinguiendo por su división de fábrica**

```
SELECT F.nomFab AS Nombre_Fabricante,
       COUNT(V.vin) AS Cant_vehiculos_env,
       SUM(V.peso*1.05) AS Peso_Total
FROM Fabricantes F, Vehiculos V, Carga C, Envios E
WHERE F.codFab = V.codFab
AND V.vin = C.vin
AND C.idEnvio = E.idEnvio
AND YEAR(E.fchEnvio) = 2016
GROUP BY F.nomFab
ORDER BY Peso_Total DESC;
```

OUTPUT:

Nombre_Fabricante	Cant_vehiculos_env	Peso_Total
Honda	278	1254225.0000
GM	3	16380.0000
GM Canada	2	10920.0000
Chrysler	2	5880.0000
Buick	2	5880.0000
Cadillac	1	3990.0000
Jaguar	1	3675.0000
Mercedes Benz	1	3465.0000
Ford	1	3360.0000
Dodge	1	3150.0000
Audi	1	2625.0000
Bmw	1	1050.0000

**6 c) Para todos los países que fueron destino de envíos, mostrar su nombre, su cantidad de envíos en los 20 primeros días de enero de 2016 y la fecha del último envío realizado, si algún país nunca fue destino de envíos igual debe mostrar su nombre.**

**En relación a la fecha, la consulta devuelve la última fecha de envío para los envíos en los primeros 20 días del mes de Enero de 2016. Si se deseara obtener la última fecha de envío para ese país (independiente de los días, el mes y el año) se debe modificar MAX(E2.fchEnvio) por MAX(E.fchEnvio)**

```

SELECT P.nomPais AS Nombre_de_pais,
       COUNT(DISTINCT(E2.idEnvio)) AS Cantidad_de_envios,
       MAX(E2.fchEnvio) AS Fecha_del_ultimo_envio
FROM Países P
LEFT JOIN Envios E
ON P.codPais = E.desEnvio
LEFT JOIN Envios E2
ON P.codPais = E2.desEnvio
AND E2.fchEnvio BETWEEN '20160101' AND '20160120'
GROUP BY P.nomPais;

```

OUTPUT:

Nombre_de_pais	Cantidad_de_envios	Fecha_del_ultimo_envio
Alemania	0	NULL
Argentina	0	NULL
Brasil	0	NULL
Canadá	1	2016-01-15 00:00:00.000
Chile	0	NULL
Corea	0	NULL
Estados Unidos	0	NULL
Holanda	0	NULL
Inglaterra	0	NULL
Italia	1	2016-01-15 00:00:00.000
Japón	0	NULL
LOCAL	0	NULL
México	0	NULL
Suecia	0	NULL
Uruguay	1	2016-01-11 00:00:00.000

**6 d) Mostrar los datos de los fabricantes que tengan envíos con más de 500 vehículos y que no tengan más de 3 envíos con menos de 100 vehículos.**

```

SELECT F.codFab, F.nomFab, F.dirFab, F.mailFab
FROM Fabricantes F
WHERE F.codFab IN (SELECT F2.codFab
                   FROM Fabricantes F2, Vehiculos V2, Carga C2, Envios E2
                   WHERE F2.codFab = V2.codFab
                   AND V2.vin = C2.vin
                   AND C2.idEnvio = E2.idEnvio
                   GROUP BY F2.codFab, E2.idEnvio
                   HAVING COUNT(C2.idCarga) > 500)
AND F.codFab NOT IN (SELECT F3.codFab
                    FROM Fabricantes F3, Vehiculos V3, Carga C3, Envios E3
                    WHERE F3.codFab = V3.codFab
                    AND V3.vin = C3.vin
                    AND C3.idEnvio = E3.idEnvio
                    AND F3.codFab IN (SELECT F4.codFab
                                     FROM Fabricantes F4, Vehiculos V4, Carga
C4, Envios E4
                                     WHERE F4.codFab = V4.codFab
                                     AND V4.vin = C4.vin
                                     AND C4.idEnvio = E4.idEnvio
                                     GROUP BY F4.codFab, E4.idEnvio
                                     HAVING COUNT(C4.idCarga) < 100)
                    GROUP BY F3.codFab, E3.idEnvio
                    HAVING COUNT(F3.codFab) > 3);

```

OUTPUT:

codFab	nomFab	dirFab	mailFab
2A	Pontiac	9 Kingsgate St, 9PD	pontiac_factory1@pontiac.com

**6 e) Obtener todos los datos de los vehículos que fueron enviados en la fecha más reciente.**

```
SELECT V.vin, V.modelo, V.color, V.peso, V.caracteristicas, V.codPais, V.codFab
FROM Vehiculos V, Carga C, Envios E
WHERE V.vin = C.vin
AND E.idEnvio = C.idEnvio
AND E.fchEnvio >= ALL(SELECT E2.fchEnvio
                      FROM Vehiculos V2, Carga C2, Envios E2
                      WHERE V2.vin = C2.vin
                      AND E2.idEnvio = C2.idEnvio);
```

OUTPUT:

vin	modelo	color	peso	caracteristicas	codPais	codFab
9ABAL320XHAB10190	A5	gris	1500.00	Frenos ABS, Aire		
Acondicionado y tapizado de cuero				9	AB	

**6 f) Mostrar los datos de los fabricantes que no han hecho envíos en el primer semestre del 2016, registraron algún envío en el 2017 y solo han hecho envíos con destino Holanda.**

```
SELECT F.codFab, F.nomFab, F.dirFab, F.mailFab
FROM Fabricantes F
WHERE F.codFab NOT IN (SELECT F2.codFab
                      FROM Fabricantes F2, Vehiculos V2, Carga C2, Envios E2
                      WHERE F2.codFab = V2.codFab
                      AND V2.vin = C2.vin
                      AND C2.idEnvio = E2.idEnvio
                      AND E2.fchEnvio BETWEEN '20160101' AND '20160630')
AND F.codFab IN (SELECT F3.codFab
                FROM Fabricantes F3, Vehiculos V3, Carga C3, Envios E3
                WHERE F3.codFab = V3.codFab
                AND V3.vin = C3.vin
                AND C3.idEnvio = E3.idEnvio
                AND YEAR(E3.fchEnvio) = '2017')
AND F.codFab IN (SELECT F4.codFab
                FROM Fabricantes F4, Vehiculos V4, Carga C4, Envios E4
                WHERE F4.codFab = V4.codFab
                AND V4.vin = C4.vin
                AND C4.idEnvio = E4.idEnvio
                AND E4.desEnvio = 'H');
```

OUTPUT:

codFab	nomFab	dirFab	mailFab
DA	Mercedes Benz	44 Jean Armour Dr, 2SD	mercedesbenz_factory1@mercedesbenz.com

**6 g) Para cada vehículo con menos de 2.300 kg de peso, mostrar sus datos, la fecha del último envío donde participó, y el nombre del fabricante.**

```
SELECT V.vin,
       V.modelo,
       V.peso,
       MAX(E.fchEnvio) AS Fecha_ultimo_envio,
       F.nomFab AS Nombre_fabricante
FROM Vehiculos V, Carga C, Envios E, Fabricantes F
WHERE V.vin = C.vin
AND E.idEnvio = C.idEnvio
AND F.codFab = V.codFab
AND V.peso < 2300
GROUP BY V.vin, V.modelo, V.peso, F.nomFab;
```

OUTPUT:

vin	modelo	peso	Fecha_ultimo_envio	Nombre_fabricante
1AAAL320XHAA10190	A5	1500.00	2017-01-15 00:00:00.000	Audi
1BAAL3150FBA10190	Z3	1000.00	2015-11-15 00:00:00.000	Bmw
1BAAL3159GBA10190	Z3	1000.00	2016-08-15 00:00:00.000	Bmw
5BBFE3204HBB10190	Z4	1800.00	2017-04-07 00:00:00.000	Bmw
9ABAL320XHAB10190	A5	1500.00	2017-06-20 00:00:00.000	Audi
SAAAL3207HAA10190	A5	1500.00	2017-02-13 00:00:00.000	Audi
ZBAFE3201HBA10190	Z4	1800.00	2017-01-23 00:00:00.000	Bmw

**6 h) Bajar el peso en un 5% a todos los vehículos que no nunca fueron enviados pasado 1 año de su fabricación (puede utilizar la función 4b).**

```
UPDATE Vehiculos
SET peso = peso - peso * 0.05
WHERE vin NOT IN (SELECT V.vin
                  FROM Vehiculos V, Carga C, Envios E
                  WHERE V.vin = C.vin
                  AND E.idEnvio = C.idEnvio
                  AND YEAR(E.fchEnvio) - dbo.funct_anioDelmodelo_vin(V.vin) <
1)
```

OUTPUT:

(8 row(s) affected)

**6 i) Obtener los datos de los fabricantes de Japón que en el 2016 hayan tenido en algún mes un envío de cantidad total superior a 100 vehículos y que también hayan tenido algún mes con una cantidad total inferior a 20.**

```
SELECT F.codFab, F.nomFab, F.dirFab, F.mailFab
FROM Fabricantes F, Plantas P
WHERE F.codFab = P.codFab
AND F.codFab IN (SELECT F2.codFab
                 FROM Fabricantes F2, Vehiculos V2, Carga C2, Envios E2
                 WHERE F2.codFab = V2.codFab
                 AND V2.vin = C2.vin
                 AND C2.idEnvio = E2.idEnvio
```

```

        AND YEAR(E2.fchEnvio) = '2016'
        GROUP BY MONTH(E2.fchEnvio), F2.codFab, E2.idEnvio
        HAVING COUNT(C2.idCarga) > 100)
AND F.codFab IN (SELECT F3.codFab
                FROM Fabricantes F3, Vehiculos V3, Carga C3, Envios E3
                WHERE F3.codFab = V3.codFab
                AND V3.vin = C3.vin
                AND C3.idEnvio = E3.idEnvio
                AND YEAR(E3.fchEnvio) = '2016'
                GROUP BY MONTH(E3.fchEnvio), F3.codFab, E3.idEnvio
                HAVING COUNT(C3.idCarga) < 20)
AND P.codPais = 'J';

```

OUTPUT:

codFab	nomFab	dirFab	mailFab
HA	Honda	26A Station St, 5HJ	honda_factory1@honda.com

## 7) Realizar una vista que muestre País de Fabricación del vehículo y cantidad de vehículos enviados a Países diferentes a los de la fábrica.

```

CREATE VIEW vista_codPais_cantVehiculos AS
SELECT P.nomPais AS Pais_de_fabricacion,
       COUNT(C.vin) AS Cant_vehiculos_env
FROM Vehiculos V, Carga C, Envios E, Países P
WHERE V.codPais = P.codPais
AND V.vin = C.vin
AND V.codPais <> E.desEnvio
AND E.idEnvio = C.idEnvio
GROUP BY P.nomPais;

```

### Test

```

-- Test
SELECT * FROM vista_codPais_cantVehiculos
ORDER BY Cant_vehiculos_env DESC;

```

OUTPUT:

Pais_de_fabricacion	Cant_vehiculos_env
Estados Unidos	618
Japón	277
Alemania	22
Canadá	6
Inglaterra	6
Corea	5
México	5
Brasil	4
Chile	2
Suecia	2
Italia	2
Uruguay	1

## Anexo

En el anexo se incluyen funciones accesorias que se utilizaron para facilitar el ingreso de datos de prueba, ejemplo una función que construye VINs.

```
/*
*****
* Script auxiliar con funciones para facilitar el ingreso de datos de prueba y
tests
*****
*/

USE BD_VEHICULOS;

/*
*****
* Dado un nombre de país retornar su código
*****
*/

CREATE FUNCTION funct_aux_nombre_codPais
(@nombre VARCHAR(30))
RETURNS CHARACTER(1)
BEGIN
DECLARE @ret CHARACTER(1)
IF(@nombre = 'Estados Unidos')
BEGIN
SET @ret = '1'
END
ELSE IF(@nombre = 'Canadá')
BEGIN
SET @ret = '2'
END
ELSE IF(@nombre = 'México')
BEGIN
SET @ret = '3'
END
ELSE IF(@nombre = 'Argentina')
BEGIN
SET @ret = '4'
END
ELSE IF(@nombre = 'Chile')
BEGIN
SET @ret = '5'
END
ELSE IF(@nombre = 'Uruguay')
BEGIN
SET @ret = '6'
END
ELSE IF(@nombre = 'Brasil')
BEGIN
SET @ret = '9'
END
ELSE IF(@nombre = 'Japón')
BEGIN
SET @ret = 'J'
END
END
```

```

ELSE IF(@nombre = 'Korea')
    BEGIN
        SET @ret = 'K'
    END
ELSE IF(@nombre = 'Inglaterra')
    BEGIN
        SET @ret = 'S'
    END
ELSE IF(@nombre = 'Alemania')
    BEGIN
        SET @ret = 'W'
    END
ELSE IF(@nombre = 'Suecia')
    BEGIN
        SET @ret = 'Y'
    END
ELSE IF(@nombre = 'Italia')
    BEGIN
        SET @ret = 'Z'
    END
ELSE IF(@nombre = 'Holanda')
    BEGIN
        SET @ret = 'H'
    END
ELSE
    BEGIN
        SET @ret = '-'
    END

RETURN @ret
END;

-- Test
DECLARE @output CHARACTER(1)
SET @output = dbo.funcnt_aux_nombre_codPais('Alemania')
PRINT @output

DECLARE @output CHARACTER(1)
SET @output = dbo.funcnt_aux_nombre_codPais('Panamá')
PRINT @output

/*
*****
* Dado un código de fabricante retornar su nombre
*****
*/

CREATE FUNCTION funcnt_aux_codFab_nombre
(@codFab CHARACTER(2))
RETURNS VARCHAR(30)
BEGIN
    DECLARE @ret VARCHAR(30)
    DECLARE @codFabSinDiv CHARACTER(1)

    SET @codFabSinDiv = SUBSTRING(@codFab,1,1)

    IF(@codFabSinDiv = 'A')
        BEGIN
            SET @ret = 'Audi'
        END

```

```

IF(@codFabSinDiv = 'B')
    BEGIN
        SET @ret = 'Bmw'
    END
IF(@codFabSinDiv = 'C')
    BEGIN
        SET @ret = 'Chrysler'
    END
IF(@codFabSinDiv = 'D')
    BEGIN
        SET @ret = 'Mercedes Benz'
    END
IF(@codFabSinDiv = 'E')
    BEGIN
        SET @ret = 'Dodge'
    END
IF(@codFabSinDiv = 'F')
    BEGIN
        SET @ret = 'Ford'
    END
IF(@codFabSinDiv = 'G')
    BEGIN
        SET @ret = 'GM'
    END
IF(@codFabSinDiv = 'H')
    BEGIN
        SET @ret = 'Honda'
    END
IF(@codFabSinDiv = 'J')
    BEGIN
        SET @ret = 'Jaguar'
    END
IF(@codFabSinDiv = 'L')
    BEGIN
        SET @ret = 'Lincon'
    END
IF(@codFabSinDiv = 'M')
    BEGIN
        SET @ret = 'Mercury'
    END
IF(@codFabSinDiv = 'N')
    BEGIN
        SET @ret = 'Nissan'
    END
IF(@codFabSinDiv = 'P')
    BEGIN
        SET @ret = 'Plymounth'
    END
IF(@codFabSinDiv = 'S')
    BEGIN
        SET @ret = 'Saab'
    END
IF(@codFabSinDiv = 'T')
    BEGIN
        SET @ret = 'Toyota'
    END
IF(@codFabSinDiv = 'V')
    BEGIN
        SET @ret = 'Volkswagen'
    END

```



```

IF(@codFabSinDiv = '2' OR @codFabSinDiv = '5')
    BEGIN
        SET @ret = 'Pontiac'
    END
IF(@codFabSinDiv = '3')
    BEGIN
        SET @ret = 'Oldsmobile'
    END
IF(@codFabSinDiv = '4')
    BEGIN
        SET @ret = 'Buick'
    END
IF(@codFabSinDiv = '6')
    BEGIN
        SET @ret = 'Cadillac'
    END
IF(@codFabSinDiv = '7')
    BEGIN
        SET @ret = 'GM Canada'
    END
IF(@codFabSinDiv = '8')
    BEGIN
        SET @ret = 'Saturn'
    END
RETURN @ret
END

-- Test OK
DECLARE @output VARCHAR(30)
SET @output = dbo.funcnt_aux_codFab_nombre('A')
PRINT @output

DECLARE @output VARCHAR(30)
SET @output = dbo.funcnt_aux_codFab_nombre('5')
PRINT @output

/*
*****
* Dado un nombre de fabricante retornar su código
*****
*/

CREATE FUNCTION funcnt_aux_nombre_codFab
(@nomFab VARCHAR(30))
RETURNS CHARACTER(1)
BEGIN
    DECLARE @ret CHARACTER(1)

    IF(@nomFab = 'Audi')
        BEGIN
            SET @ret = 'A'
        END
    IF(@nomFab = 'Bmw')
        BEGIN
            SET @ret = 'B'
        END
    IF(@nomFab = 'Chrysler')
        BEGIN
            SET @ret = 'C'
        END

```

```

        END
    IF(@nomFab = 'Mercedes Benz')
        BEGIN
            SET @ret = 'D'
        END
    IF(@nomFab = 'Dodge')
        BEGIN
            SET @ret = 'E'
        END
    IF(@nomFab = 'Ford')
        BEGIN
            SET @ret = 'F'
        END
    IF(@nomFab = 'GM')
        BEGIN
            SET @ret = 'G'
        END
    IF(@nomFab = 'Honda')
        BEGIN
            SET @ret = 'H'
        END
    IF(@nomFab = 'Jaguar')
        BEGIN
            SET @ret = 'J'
        END
    IF(@nomFab = 'Lincon')
        BEGIN
            SET @ret = 'L'
        END
    IF(@nomFab = 'Mercury')
        BEGIN
            SET @ret = 'M'
        END
    IF(@nomFab = 'Nissan')
        BEGIN
            SET @ret = 'N'
        END
    IF(@nomFab = 'Plymounth')
        BEGIN
            SET @ret = 'P'
        END
    IF(@nomFab = 'Renault')
        BEGIN
            SET @ret = 'R'
        END
    IF(@nomFab = 'Saab')
        BEGIN
            SET @ret = 'S'
        END
    IF(@nomFab = 'Toyota')
        BEGIN
            SET @ret = 'T'
        END
    IF(@nomFab = 'Volkswagen')
        BEGIN
            SET @ret = 'V'
        END
    IF(@nomFab = 'Pontiac')
        BEGIN
            SET @ret = '2'
        END

```

```

        END
    IF(@nomFab = 'Oldsmobile')
        BEGIN
            SET @ret = '3'
        END
    IF(@nomFab = 'Buick')
        BEGIN
            SET @ret = '4'
        END
    IF(@nomFab = 'Cadillac')
        BEGIN
            SET @ret = '6'
        END
    IF(@nomFab = 'GM Canada')
        BEGIN
            SET @ret = '7'
        END
    IF(@nomFab = 'Saturn')
        BEGIN
            SET @ret = '8'
        END
    RETURN @ret
END

-- Test OK
DECLARE @output CHARACTER(1)
SET @output = dbo.funcnt_aux_nombre_codFab('Audi')
PRINT @output

DECLARE @output CHARACTER(1)
SET @output = dbo.funcnt_aux_codFab_nombre('Saturn')
PRINT @output

/*
*****
* Dado un año retornar código
*****
*/

CREATE FUNCTION funcnt_anio_cod
(@anio INT)
RETURNS CHAR(1)
AS
BEGIN
    DECLARE @ret CHAR(1)

    IF(@anio = 2010)
        BEGIN
            SET @ret = 'A'
        END
    IF(@anio = 2011)
        BEGIN
            SET @ret = 'B'
        END
    IF(@anio = 2012)
        BEGIN
            SET @ret = 'C'
        END
    IF(@anio = 2013)
        BEGIN

```

```

        SET @ret = 'D'
    END
IF(@anio = 2014)
    BEGIN
        SET @ret = 'E'
    END
IF(@anio = 2015)
    BEGIN
        SET @ret = 'F'
    END
IF(@anio = 2016)
    BEGIN
        SET @ret = 'G'
    END
IF(@anio = 2017)
    BEGIN
        SET @ret = 'H'
    END
IF(@anio = 2018)
    BEGIN
        SET @ret = 'J'
    END
IF(@anio = 2019)
    BEGIN
        SET @ret = 'K'
    END
IF(@anio = 2020)
    BEGIN
        SET @ret = 'L'
    END
IF(@anio = 2021)
    BEGIN
        SET @ret = 'M'
    END
IF(@anio = 2022)
    BEGIN
        SET @ret = 'N'
    END
IF(@anio = 2023)
    BEGIN
        SET @ret = 'P'
    END
IF(@anio = 2024)
    BEGIN
        SET @ret = 'R'
    END
IF(@anio = 2025)
    BEGIN
        SET @ret = 'S'
    END
IF(@anio = 2026)
    BEGIN
        SET @ret = 'T'
    END
IF(@anio = 2027)
    BEGIN
        SET @ret = 'V'
    END
IF(@anio = 2028)
    BEGIN

```

```

        SET @ret = 'W'
    END
IF(@anio = 2029)
    BEGIN
        SET @ret = 'X'
    END
IF(@anio = 2030)
    BEGIN
        SET @ret = 'Y'
    END
IF(@anio = 2031)
    BEGIN
        SET @ret = '1'
    END
IF(@anio = 2032)
    BEGIN
        SET @ret = '2'
    END
IF(@anio = 2033)
    BEGIN
        SET @ret = '3'
    END
IF(@anio = 2034)
    BEGIN
        SET @ret = '4'
    END
IF(@anio = 2035)
    BEGIN
        SET @ret = '5'
    END
IF(@anio = 2036)
    BEGIN
        SET @ret = '6'
    END
IF(@anio = 2037)
    BEGIN
        SET @ret = '7'
    END
IF(@anio = 2038)
    BEGIN
        SET @ret = '8'
    END
IF(@anio = 2039)
    BEGIN
        SET @ret = '9'
    END
RETURN @ret
END;

-- Test
DECLARE @output CHARACTER(1)
SET @output = dbo.funct_anio_cod(2010)
PRINT @output

DECLARE @output CHARACTER(1)
SET @output = dbo.funct_anio_cod(2023)
PRINT @output

/*
*****

```

```

* Generador de VINs dado un codPais, codFab, tipoChasis, tipoMotor, Año
*****
*/

-- Funciones auxiliares a la función crear VIN

-- Crear WMI
CREATE FUNCTION funct_aux_crear_WMI
(@codPais CHARACTER(1), @codFab CHARACTER(2))
RETURNS CHARACTER(3)
BEGIN
DECLARE @ret CHARACTER(3)
SET @ret = @codPais + @codFab
RETURN @ret
END

-- Test
DECLARE @output CHARACTER(3)
SET @output = dbo.funct_aux_crear_WMI('1', 'AA')
PRINT @output

-- Crear VDS
CREATE FUNCTION funct_aux_crear_VDS
(@tipoChasis VARCHAR(30), @modeloMotor VARCHAR(30))
RETURNS CHARACTER(6)
BEGIN
DECLARE @ret CHARACTER(6)
DECLARE @codChasis CHARACTER(2)
DECLARE @codMotor CHARACTER(2)
DECLARE @rand1 CHARACTER(1)
DECLARE @rand2 CHARACTER(1)
IF(@tipoChasis = 'Aluminio')
BEGIN
SET @codChasis = 'AL'
END
IF(@tipoChasis = 'Hierro')
BEGIN
SET @codChasis = 'FE'
END
IF(@modeloMotor = '1.0')
BEGIN
SET @codMotor = '10'
END
IF(@modeloMotor = '1.5')
BEGIN
SET @codMotor = '15'
END
IF(@modeloMotor = '2.0')
BEGIN
SET @codMotor = '20'
END
SET @rand1 = '3'
SET @rand2 = '4'
SET @ret = @codChasis + @rand1 + @codMotor + @rand2
RETURN @ret
END;

-- Test
DECLARE @output CHARACTER(6)
SET @output = dbo.funct_aux_crear_VDS('Aluminio', '2.0')

```

```

PRINT @output

DECLARE @output CHARACTER(6)
SET @output = dbo.funct_aux_crear_VDS('Hierro','1.0')
PRINT @output

-- Crear VIS
CREATE FUNCTION funct_aux_crear_VIS
(@anio INT, @codFab CHARACTER(2))
RETURNS CHARACTER(8)
BEGIN
DECLARE @ret CHARACTER(8)
DECLARE @codAnio CHARACTER(1)
DECLARE @codCifras CHARACTER(4)
DECLARE @numArbit CHARACTER(1)
SET @codAnio = dbo.funct_anio_cod(@anio)
SET @numArbit = 0
IF(@anio BETWEEN 2010 AND 2019)
    BEGIN
        SET @codCifras = '1019'
    END
IF(@anio BETWEEN 2019 AND 2020)
    BEGIN
        SET @codCifras = '1920'
    END
IF(@anio BETWEEN 2020 AND 2029)
    BEGIN
        SET @codCifras = '2029'
    END
IF(@anio BETWEEN 2029 AND 2039)
    BEGIN
        SET @codCifras = '2939'
    END
ELSE IF(@anio < 2010 OR @anio > 2039)
    BEGIN
        SET @codCifras = '5555'
    END
SET @ret = @codAnio + @codFab + @codCifras + @numArbit
RETURN @ret
END;

-- Test
DECLARE @output CHARACTER(8)
SET @output = dbo.funct_aux_crear_VIS(2010,'AA')
PRINT @output

DECLARE @output CHARACTER(8)
SET @output = dbo.funct_aux_crear_VIS(2019,'AB')
PRINT @output

DECLARE @output CHARACTER(8)
SET @output = dbo.funct_aux_crear_VIS(2025,'BS')
PRINT @output

-- Crear VIN generator (sin verificar el dígito verificador en la posición 9)
CREATE FUNCTION funct_aux_VIN_generator
(@codPais CHARACTER(1), @codFab CHARACTER(2), @tipoChasis VARCHAR(30),
@tipoMotor VARCHAR(30), @Anio INT)
RETURNS CHARACTER(17)
BEGIN

```

```

DECLARE @ret CHARACTER(17)
DECLARE @WMI CHARACTER(3)
DECLARE @VDS CHARACTER(6)
DECLARE @VIS CHARACTER(8)
SET @WMI = dbo.funct_aux_crear_WMI(@codPais, @codFab)
SET @VDS = dbo.funct_aux_crear_VDS(@tipoChasis, @tipoMotor)
SET @VIS = dbo.funct_aux_crear_VIS(@Anio, @codFab)
SET @ret = @WMI + @VDS + @VIS
RETURN @ret
END;

-- Test
DECLARE @output CHARACTER(17)
SET @output = dbo.funct_aux_VIN_generator('1','AA','Aluminio','2.0',2017)
PRINT @output

DECLARE @output CHARACTER(17)
SET @output = dbo.funct_aux_VIN_generator('1','BS','Hierro','1.0',2015)
PRINT @output

DECLARE @output CHARACTER(17)
SET @output = dbo.funct_aux_VIN_generator('W','DA','Hierro','2.0',2016)
PRINT @output

-- Verificar VIN
DECLARE @output CHARACTER(17)
SET @output = dbo.funct_validar_digitoverificador_vin('WDAFE3204GDA10190');
PRINT @output

```