



	CONSTRUCCIÓN DEL SISTEMA DE INFORMACIÓN (CSI)	BASES DE DATOS
 Escuela Superior de Ingeniería y Tecnología Universidad de La Laguna	PROYECTO: <u>Base de datos para la empresa Cheap-Car.</u>	Generación de código
	Autor: Yeray Expósito García y Sergio Ravelo Vegino	alu0100951844 alu0100902341
Versión: 1	Tiempo invertido: 3 días	Fecha : 05/07/19

GENERACIÓN DE CÓDIGO DE COMPONENTES Y PROCEDIMIENTOS (CSI-2)

CÓDIGO DE COMPONENTES
<p>Script en Drive -> https://drive.google.com/open?id=1cUCxsEydOPczbDapOJ2SHPnPA9p7212X</p> <p>-- CREANDO TIPO DE DE DATO</p> <pre>CREATE TYPE DATOS_COCHE AS (Modelo VARCHAR(64), Antigüedad INT, Kilometros INT, Estado VARCHAR(264), Precio FLOAT);</pre> <p>-----</p> <p>-- Table Tiendas</p> <p>-----</p> <pre>CREATE TABLE IF NOT EXISTS Tiendas (Id_tienda INT NOT NULL, Localizacion VARCHAR(64) NOT NULL, Stock INT NOT NULL CHECK(Stock >= 0), Aforo INT NOT NULL CHECK(Aforo > 0), PRIMARY KEY (Id_tienda));</pre> <p>-----</p> <p>-- Table Trabajadores</p> <p>-----</p> <pre>CREATE TABLE IF NOT EXISTS Trabajadores (DNI VARCHAR(9) NOT NULL, Puesto VARCHAR(64) NOT NULL, Nombre VARCHAR(64) NOT NULL, Apellidos VARCHAR(64) NOT NULL, Sueldo FLOAT NOT NULL CHECK(Sueldo > 0), Horario VARCHAR(264) NOT NULL, Telefono INT NOT NULL, Id_tienda INT NOT NULL, PRIMARY KEY (DNI), CONSTRAINT fk_Trabajadores_Tiendas FOREIGN KEY (Id_tienda) REFERENCES Tiendas (Id_tienda) ON DELETE NO ACTION ON UPDATE CASCADE);</pre>

	CONSTRUCCIÓN DEL SISTEMA DE INFORMACIÓN (CSI)	BASES DE DATOS
 Escuela Superior de Ingeniería y Tecnología Universidad de La Laguna	PROYECTO: <u>Base de datos para la empresa Cheap-Car.</u>	Generación de código
	Autor: Yeray Expósito García y Sergio Ravelo Vegino	alu0100951844 alu0100902341
Versión: 1	Tiempo invertido: 3 días	Fecha : 05/07/19

```
CREATE INDEX fk_Trabajadores_Tiendas_idx ON Trabajadores(Id_tienda);
```

```
-----  
-- Table Clientes_Compra  
-----
```

```
CREATE TABLE IF NOT EXISTS Clientes_Compra (
  DNI VARCHAR(9) NOT NULL,
  Nombre VARCHAR(64) NOT NULL,
  Apellidos VARCHAR(64) NULL,
  Telefono INT NOT NULL,
  PRIMARY KEY (DNI));
```


```
-----  
-- Table Ventas  
-----
```

```
CREATE TABLE IF NOT EXISTS Ventas (
  Id_venta INT NOT NULL,
  Importe FLOAT NOT NULL CHECK(Importe > 0),
  Fecha DATE NOT NULL,
  DNI_trabajador VARCHAR(9) NOT NULL,
  DNI_comprador VARCHAR(9) NOT NULL,
  PRIMARY KEY (Id_venta),
  CONSTRAINT fk_Ventas_Trabajadores1
    FOREIGN KEY (DNI_trabajador)
    REFERENCES Trabajadores (DNI)
    ON DELETE NO ACTION
    ON UPDATE CASCADE,
  CONSTRAINT fk_Ventas_Clientes_Compra1
    FOREIGN KEY (DNI_comprador)
    REFERENCES Clientes_Compra (DNI)
    ON DELETE NO ACTION
    ON UPDATE CASCADE);

CREATE INDEX fk_Ventas_Trabajadores1_idx ON Ventas(DNI_trabajador);
CREATE INDEX fk_Ventas_Clientes_Compra1_idx ON Ventas(DNI_comprador);
```

```
-----  
-- Table Clientes_Venta  
-----
```

```
CREATE TABLE IF NOT EXISTS Clientes_Venta (
  DNI VARCHAR(9) NOT NULL,
```

	CONSTRUCCIÓN DEL SISTEMA DE INFORMACIÓN (CSI)	BASES DE DATOS
 Escuela Superior de Ingeniería y Tecnología Universidad de La Laguna	PROYECTO: <u>Base de datos para la empresa Cheap-Car.</u>	Generación de código
	Autor: Yeray Expósito García y Sergio Ravelo Vegino	alu0100951844 alu0100902341
Versión: 1		Fecha : 05/07/19

Nombre VARCHAR(64) NOT NULL,
Apellidos VARCHAR(64) NULL,
Telefono INT NOT NULL,
PRIMARY KEY (DNI));


-- Table Vehiculos

```
CREATE TABLE IF NOT EXISTS Vehiculos (
  Matricula VARCHAR(7) NOT NULL,
  Num_rotaciones INT NOT NULL CHECK(Num_rotaciones >= 0),
  Datos_coche DATOS_COCHE NOT NULL,
  Disponibilidad VARCHAR(45) NOT NULL,
  Id_tienda INT,
  PRIMARY KEY (Matricula),
  CONSTRAINT fk_Vehiculos_Tiendas1
  FOREIGN KEY (Id_tienda)
  REFERENCES Tiendas (Id_tienda)
  ON DELETE NO ACTION
  ON UPDATE CASCADE);

CREATE INDEX fk_Vehiculos_Tiendas1_idx ON Vehiculos(Id_tienda);
```

-- Table Venta_Vehiculos

```
CREATE TABLE IF NOT EXISTS Venta_Vehiculos (
  Fin_garantia DATE NULL,
  Incidentes VARCHAR(264) NULL,
  Num_incidentes INT NOT NULL CHECK(Num_incidentes >= 0),
  Id_venta INT NOT NULL,
  Matricula VARCHAR(7) NOT NULL,
  PRIMARY KEY (Id_venta, Matricula),
  CONSTRAINT fk_Venta_Vehiculos_Ventas1
  FOREIGN KEY (Id_venta)
  REFERENCES Ventas (Id_venta)
  ON DELETE NO ACTION
  ON UPDATE CASCADE,
  CONSTRAINT fk_Venta_Vehiculos_Vehiculos1
  FOREIGN KEY (Matricula)
  REFERENCES Vehiculos (Matricula)
  ON DELETE NO ACTION
  ON UPDATE CASCADE);
```

	CONSTRUCCIÓN DEL SISTEMA DE INFORMACIÓN (CSI)	BASES DE DATOS
 Escuela Superior de Ingeniería y Tecnología Universidad de La Laguna	PROYECTO: <u>Base de datos para la empresa Cheap-Car.</u>	Generación de código
	Autor: Yeray Expósito García y Sergio Ravelo Vegino	alu0100951844 alu0100902341
Versión: 1	Tiempo invertido: 3 días	Fecha : 05/07/19

```
CREATE INDEX fk_Venta_Vehiculos_Ventas1_idx ON Venta_Vehiculos(Id_venta);
CREATE INDEX fk_Venta_Vehiculos_Vehiculos1_idx ON Venta_Vehiculos(Matricula);
```


```
-- Table Vehiculos_Acquisidos
```

```
CREATE TABLE IF NOT EXISTS Vehiculos_Acquisidos (
  Fecha_adquisicion DATE NULL,
  Precio_adquisicion FLOAT NOT NULL CHECK(Precio_adquisicion > 0),
  Matricula VARCHAR(7) NOT NULL,
  DNI_vendedor VARCHAR(9) NOT NULL,
  PRIMARY KEY (Matricula),
  CONSTRAINT fk_Vehiculos_Acquisidos_Vehiculos1
    FOREIGN KEY (Matricula)
    REFERENCES Vehiculos (Matricula)
    ON DELETE NO ACTION
    ON UPDATE CASCADE,
  CONSTRAINT fk_Vehiculos_Acquisidos_Clientes_Venta1
    FOREIGN KEY (DNI_vendedor)
    REFERENCES Clientes_Venta (DNI)
    ON DELETE NO ACTION
    ON UPDATE CASCADE);
```

```
CREATE INDEX fk_Vehiculos_Acquisidos_Vehiculos1_idx ON
Vehiculos_Acquisidos(Matricula);
CREATE INDEX fk_Vehiculos_Acquisidos_Clientes_Venta1_idx ON
Vehiculos_Acquisidos(DNI_vendedor);
```

```
-- Table Vehiculos_Entregados
```

```
CREATE TABLE IF NOT EXISTS Vehiculos_Entregados (
  Fecha_entrega DATE NULL,
  Matricula VARCHAR(7) NOT NULL,
  DNI_cliente VARCHAR(9) NOT NULL,
  PRIMARY KEY (Matricula),
  CONSTRAINT fk_Vehiculos_Entregados_Vehiculos1
    FOREIGN KEY (Matricula)
    REFERENCES Vehiculos (Matricula)
    ON DELETE NO ACTION
    ON UPDATE CASCADE,
```

	CONSTRUCCIÓN DEL SISTEMA DE INFORMACIÓN (CSI)	BASES DE DATOS
 Escuela Superior de Ingeniería y Tecnología Universidad de La Laguna	PROYECTO: <u>Base de datos para la empresa Cheap-Car.</u>	Generación de código
	Autor: Yeray Expósito García y Sergio Ravelo Vegino	alu0100951844 alu0100902341
Versión: 1	Tiempo invertido: 3 días	Fecha : 05/07/19

```

CONSTRAINT fk_Vehiculos_Entregados_Clientes_Compra1
FOREIGN KEY (DNI_cliente)
REFERENCES Clientes_Compra (DNI)
ON DELETE NO ACTION
ON UPDATE CASCADE);


CREATE INDEX fk_Vehiculos_Entregados_Vehiculos1_idx ON
Vehiculos_Entregados(Matricula);
CREATE INDEX fk_Vehiculos_Entregados_Clientes_Compra1_idx ON
Vehiculos_Entregados(DNI_cliente);

-----
-- Table Rotacion
-----

CREATE TABLE IF NOT EXISTS Rotacion (
  Id_rotacion INT NOT NULL,
  Fecha DATE NOT NULL,
  Tienda_origen INT NOT NULL,
  Tienda_destino INT NOT NULL,
  Matricula VARCHAR(7) NOT NULL,
  PRIMARY KEY (Id_rotacion, Matricula),
  CONSTRAINT fk_Rotacion_Tiendas1
  FOREIGN KEY (Tienda_origen)
  REFERENCES Tiendas (Id_tienda)
  ON DELETE NO ACTION
  ON UPDATE CASCADE,
  CONSTRAINT fk_Rotacion_Tiendas2
  FOREIGN KEY (Tienda_destino)
  REFERENCES Tiendas (Id_tienda)
  ON DELETE NO ACTION
  ON UPDATE CASCADE,
  CONSTRAINT fk_Rotacion_Vehiculos1
  FOREIGN KEY (Matricula)
  REFERENCES Vehiculos (Matricula)
  ON DELETE NO ACTION
  ON UPDATE CASCADE);

CREATE INDEX fk_Rotacion_Tiendas1_idx ON Rotacion(Tienda_origen);
CREATE INDEX fk_Rotacion_Tiendas2_idx ON Rotacion(Tienda_destino);
CREATE INDEX fk_Rotacion_Vehiculos1_idx ON Rotacion(Matricula);

```

	CONSTRUCCIÓN DEL SISTEMA DE INFORMACIÓN (CSI)	BASES DE DATOS
 Escuela Superior de Ingeniería y Tecnología Universidad de La Laguna	PROYECTO: <u>Base de datos para la empresa Cheap-Car.</u>	Generación de código
	Autor: Yeray Expósito García y Sergio Ravelo Vegino	alu0100951844 alu0100902341
Versión: 1	Tiempo invertido: 3 días	Fecha : 05/07/19

CÓDIGO DE PROCEDIMIENTOS DE OPERACIÓN Y SEGURIDAD

Script en Drive -> <https://drive.google.com/open?id=1Y2yn1vja-dpF2Bb-ywTTLkpT7L7tGqG7>

1.- Trigger “venta_realizada”.

Este trigger ejecuta la función “nueva_venta()” después de una operación INSERT sobre la tabla Venta_Vehiculos. Dicha función, se encarga de actualizar ciertos atributos de otras tablas que están relacionados con la venta que se haya llevado a cabo. En primer lugar reduce en una unidad el atributo Stock de la tienda que posea el vehículo, a continuación, en la tabla Vehículos se cambia el valor del atributo Disponibilidad, asociado al coche, a “VENDIDO” y el campo Id_tienda pasa a ser NULL, ya que el automóvil no se encuentra en ninguna tienda.

```
CREATE OR REPLACE FUNCTION nueva_venta() RETURNS TRIGGER AS $$
BEGIN
    UPDATE Tiendas
    SET Stock = Stock - 1
    WHERE Tiendas.Id_tienda = (SELECT Id_tienda
                                FROM Vehiculos
                                WHERE NEW.Matricula = Vehiculos.Matricula);


    UPDATE Vehiculos
    SET Disponibilidad = 'VENDIDO', Id_tienda = NULL
    WHERE NEW.Matricula = Vehiculos.Matricula;

    RAISE NOTICE 'DATOS RELACIONADOS CON LA VENTA ACTUALIZADOS';
    RETURN NULL;
END;
$$ LANGUAGE plpgsql;

CREATE TRIGGER venta_realizada AFTER INSERT
ON Venta_Vehiculos FOR EACH ROW
EXECUTE PROCEDURE nueva_venta();
```

2.- Trigger “verificar_rotacion”.

Este trigger ejecuta la función “comprobar_rotacion” antes de que se lleve a cabo una operación INSERT sobre la tabla Rotación. Dicha función, se encarga de comprobar, antes de registrar una rotación, que la tienda de origen dispone de algún vehículo, es decir, si Stock=0 no se podría llevar a cabo la rotación. También se comprueba si la tienda destino dispone de espacio para ubicar el nuevo vehículo, por lo que si Stock=Aforo no se podrá realizar dicha rotación.

	CONSTRUCCIÓN DEL SISTEMA DE INFORMACIÓN (CSI)	BASES DE DATOS
 Escuela Superior de Ingeniería y Tecnología Universidad de La Laguna	PROYECTO: <u>Base de datos para la empresa Cheap-Car.</u>	Generación de código
	Autor: Yeray Expósito García y Sergio Ravelo Vegino	alu0100951844 alu0100902341
Versión: 1	Tiempo invertido: 3 días	Fecha : 05/07/19

```

CREATE OR REPLACE FUNCTION comprobar_rotacion() RETURNS TRIGGER AS $$
BEGIN
    IF EXISTS (SELECT * FROM Tiendas WHERE Id_tienda=NEW.Tienda_origen AND Stock=0) THEN
        RAISE EXCEPTION 'LA TIENDA DE ORIGEN NO DISPONE DE VEHÍCULOS';
    END IF;

    IF EXISTS (SELECT * FROM Tiendas WHERE Id_tienda=NEW.Tienda_destino AND Stock=Aforo) THEN
        RAISE EXCEPTION 'LA TIENDA DE DESTINO NO DISPONE DE ESPACIO SUFICIENTE';
    END IF;

    RAISE NOTICE 'ROTACIÓN CORRECTA';
    RETURN NEW;
END
$$ LANGUAGE plpgsql;

CREATE TRIGGER verificar_rotacion BEFORE INSERT
ON Rotacion FOR EACH ROW
EXECUTE PROCEDURE comprobar_rotacion();

```

3.- Trigger “rotacion_realizada”.

Este trigger ejecuta la función “nueva_rotacion” después de que se lleve a cabo una operación INSERT sobre la tabla Rotación. Dicha función se encarga de actualizar los atributos de otras tablas relacionados con la rotación en cuestión. En este caso, será necesario reducir en una unidad el Stock de la tienda origen e incrementar el Stock de la tienda de destino. Además, se deberá actualizar el atributo Id_tienda de la tabla Vehículo, con el fin de que refleje la nueva ubicación, por último, se incrementará en una unidad el atributo Num_rotaciones.

```

CREATE OR REPLACE FUNCTION nueva_rotacion() RETURNS TRIGGER AS $$
BEGIN
    UPDATE Tiendas
    SET Stock = Stock - 1
    WHERE Id_tienda = NEW.Tienda_origen;


    UPDATE Tiendas
    SET Stock = Stock + 1
    WHERE Id_tienda = NEW.Tienda_destino;

    UPDATE Vehiculos
    SET Id_tienda = NEW.Tienda_destino, Num_rotaciones = Num_rotaciones + 1
    WHERE Matricula=NEW.Matricula;

    RAISE NOTICE 'ROTACIÓN REALIZADA CORRECTAMENTE';
    RETURN NULL;
END
$$ LANGUAGE plpgsql;

CREATE TRIGGER rotacion_realizada AFTER INSERT
ON Rotacion FOR EACH ROW
EXECUTE PROCEDURE nueva_rotacion();

```

	CONSTRUCCIÓN DEL SISTEMA DE INFORMACIÓN (CSI)	BASES DE DATOS
 Escuela Superior de Ingeniería y Tecnología Universidad de La Laguna	PROYECTO: <u>Base de datos para la empresa Cheap-Car.</u>	Generación de código
	Autor: Yeray Expósito García y Sergio Ravelo Vegino	alu0100951844 alu0100902341
Versión: 1	Tiempo invertido: 3 días	Fecha : 05/07/19

4.- Trigger “verificar_ubucacion”.

Este trigger ejecuta la función “comprobar_ubicacion” antes de que se lleve a cabo una operación INSERT sobre la tabla Vehículos. Dicha función comprueba si la tienda en la que se desea colocar el vehículo dispone de espacio suficiente, de ser así, incrementará su Stock en la tabla Tiendas y permitirá que se realice la inserción.

```
CREATE OR REPLACE FUNCTION comprobar_ubicacion() RETURNS TRIGGER AS $$
BEGIN
    IF EXISTS (SELECT * FROM Tiendas WHERE Id_tienda=NEW.Id_tienda AND Stock=Aforo) THEN
        RAISE EXCEPTION 'NO HAY ESPACIO EN LA TIENDA ESPECIFICADA';
    END IF;

    UPDATE Tiendas
    SET Stock = Stock + 1
    WHERE Id_tienda=NEW.Id_tienda;

    RAISE NOTICE 'UBICACIÓN CORRECTA';
    RETURN NEW;
END
$$ LANGUAGE plpgsql;

CREATE TRIGGER verificar_ubicacion BEFORE INSERT
ON Vehiculos FOR EACH ROW
EXECUTE PROCEDURE comprobar_ubicacion();
```