



# ÍNDICE

<b>1. PROYECTO BASE DE DATOS</b>	<b>2</b>
1.1 INTRODUCCIÓN	2
1.2 IDENTIFICACIÓN DE ENTIDADES	2
1.3 SCRIPT MODELO FÍSICO	5
<b>2.0 MODELADO DE DATOS</b>	<b>29</b>
2.1 MODELO LÓGICO	29
2.1.2 MODELO RELACIONAL	29
<b>2.2 DE SQL DATA MODELER A SQL</b>	<b>30</b>
2.2.1 DATA MODELER	30
2.2.3 SQL SERVER	32
<b>2.3 DE SQL DATA MODELER A SQL</b>	<b>33</b>

# 1. PROYECTO BASE DE DATOS

[Volver al Índice →](#)

## 1.1 INTRODUCCIÓN

[Volver al Índice →](#)

La elaboración de este proyecto tiene como finalidad simular el diseño o modelado así como su implementación en una base de datos de acuerdo al modelo relacional.

Para el modelado de la base de datos se utilizará SQL Developer Data Modeler. Una vez creada la base de datos será exportada extrayendo el código SQL para su creación a SQL SERVER desde el cual será realizada la administración y mantenimiento de la base de datos.

El tema del proyecto se refiere a una empresa de venta de repuestos y accesorios para motos. Orientados a la calidad y satisfacción de nuestros clientes.

## 1.2 IDENTIFICACIÓN DE ENTIDADES

[Volver al Índice →](#)

Teniendo en cuenta nuestro modelo de negocio, se han elaborado las siguientes entidades con la finalidad de poder almacenar de una manera ordenada y eficaz cada uno de los datos.

**CLIENTE:** DNI, Nombre, Apellido, Telefono , Email.

**ARTÍCULO:** SKU, Nombre, Descripción, Imagen.

**PROVEEDOR :** Nombre, Dirección, Página Web, Teléfono, Email.

**PAÍS:** ISO , Nombre

**LOCALIDAD :** ID\_Localidad, Nombre

**MUNICIPIO :** ID\_Municipio, Nombre

En cuanto a las motos y repuestos se refiere, hemos decidido crear las siguientes entidades para llevar un mejor control facilitando la búsqueda por ( Fabricante, Modelo, Categoría, Cilindrada ), permitiéndonos utilizar dicha información que anclada a nuestros clientes, nos permitirá tener un mayor control y conocimiento acerca de los mismos.

**FABRICANTE :** ID\_Fabricante, Nombre, Logo

**MODELO :** ID\_Modelo, Nombre, Fecha de Fabricación, Imagen

**CATEGORÍA :** ID\_Categoria, Nombre

**CILINDRADA :** ID\_Cilindrada, Cilindrada

**COMPRA :** ID\_Compra, Fecha. A su vez será generada;

**DETALLE\_COMPRA :** ID\_DC, Cantidad, Precio

**VENTA :** ID\_Venta, Fecha, Total\_Venta

**DETALLE\_VENTA** : ID\_DV, Cantidad

**INVENTARIO**: ID\_Inventario, Cantidad, Fecha

**CARRITO DE COMPRA**: ID\_Carrito\_Web

**MODALIDAD DE PAGO**: En esta entidad almacenaremos las diversas opciones de pago que tendrá el cliente. Los campos serán los siguientes: ID\_Mod y dentro de ella tendremos la especialización con cada uno de los mismos, que serán los siguientes:

**PAYPAL**: Usuario, Email.

**TARJETA**: Fecha de caducidad, Número de Tarjeta, CCV ( COLUMNA ENCRIPATA )

**EFFECTIVO**: Cantidad

**PAGO**: ID\_Pago, Fecha.

**ENVÍO**: ID\_Envío, Fecha

**EMPLEADO**: DNI, Nombre, Apellido, FechaIngreso, Email, Teléfono.

**CARGO**: ID\_Cargo, Nombre, Descripcion\_Funciones.

**CONTRATO** : ID\_Contrato, ID\_Empleado. ( TABLA ENCRIPATA )

**DEPARTAMENTO** : ID\_Departamento, Nombre.

**COMISIONES**: En esta entidad almacenaremos las diversas comisiones que pueden tener los empleados. Los campos serán los siguientes: ID\_Comision y dentro de ella tendremos la especialización con cada uno de los mismos, que serán los siguientes: ID\_Comisión

**BAJO** : Porcentaje , Descripcion

**MEDIO** : Porcentaje , Descripcion

**ALTO** : Porcentaje , Descripcion

**CARRITO WEB:** ID\_Carrito

**ENVIO** : ID\_Envio, Fecha

**PAIS:** ISO, Nombre

**LOCALIDAD:** ID\_Localidad, Nombre

**MUNICIPIO:** ID\_Municipio, Nombre

[Volver al Indice →](#)

## 1.3 SCRIPT MODELO FÍSICO

[Volver al Índice →](#)

```
DROP DATABASE IF EXISTS STASHMOTOR
GO
```

```
CREATE DATABASE STASHMOTOR
GO
```

```
CREATE TABLE Alto
(
    ID_Comision INTEGER NOT NULL ,
    Porcentaje INTEGER NOT NULL ,
    Descripcion TEXT NOT NULL
)
GO
```

```
ALTER TABLE Alto ADD CONSTRAINT Alto_PK PRIMARY KEY CLUSTERED
(ID_Comision)
WITH (
    ALLOW_PAGE_LOCKS = ON ,
    ALLOW_ROW_LOCKS = ON )
GO
```

```
ALTER TABLE Alto ADD CONSTRAINT Alto_PKv1 UNIQUE NONCLUSTERED
(Porcentaje)
GO
```

```
CREATE TABLE Articulo
(
    SKU VARCHAR (20) NOT NULL ,
    Nombre VARCHAR (20) NOT NULL ,
    Descripcion VARCHAR (50) ,
    PVP DECIMAL (28) NOT NULL ,
    Inventario_ID_Inventario INTEGER NOT NULL ,
    Detalle_Venta_ID_DV INTEGER NOT NULL ,
    Detalle_Compra_ID_DC INTEGER NOT NULL ,
    Modelo_ID_Modelo INTEGER NOT NULL
)
GO
```

```
ALTER TABLE Articulo ADD CONSTRAINT Articulo_PK PRIMARY KEY CLUSTERED
(SKU)
```

```
        WITH (
            ALLOW_PAGE_LOCKS = ON ,
            ALLOW_ROW_LOCKS = ON )
GO

CREATE TABLE Bajo
(
    ID_Comision INTEGER NOT NULL ,
    Porcentaje INTEGER NOT NULL ,
    Descripcion TEXT NOT NULL
)
GO

ALTER TABLE Bajo ADD CONSTRAINT Bajov1_PK PRIMARY KEY CLUSTERED
(ID_Comision)
    WITH (
        ALLOW_PAGE_LOCKS = ON ,
        ALLOW_ROW_LOCKS = ON )
GO

ALTER TABLE Bajo ADD CONSTRAINT Bajo_PK UNIQUE NONCLUSTERED (Porcentaje)
GO

CREATE TABLE Cargo
(
    ID_Cargo INTEGER NOT NULL ,
    Nombre VARCHAR (20) NOT NULL ,
    Descripcion_Cargo VARCHAR (50)
)
GO

ALTER TABLE Cargo ADD CONSTRAINT Cargo_PK PRIMARY KEY CLUSTERED
(ID_Cargo)
    WITH (
        ALLOW_PAGE_LOCKS = ON ,
        ALLOW_ROW_LOCKS = ON )
GO

CREATE TABLE CarritoWeb
(
    ID_Carrito_Web INTEGER ,
```



```
    Cliente_DNI INTEGER NOT NULL ,
    Envio_ID_Envio INTEGER NOT NULL ,
    CarritoWeb_ID NUMERIC (28) NOT NULL IDENTITY NOT FOR REPLICATION
)
```

GO

```
CREATE UNIQUE NONCLUSTERED INDEX
    CarritoWeb__IDX ON CarritoWeb
(
    Envio_ID_Envio
)
```

GO

```
ALTER TABLE CarritoWeb ADD CONSTRAINT CarritoWeb_PK PRIMARY KEY
    CLUSTERED (CarritoWeb_ID)
    WITH (
        ALLOW_PAGE_LOCKS = ON ,
        ALLOW_ROW_LOCKS = ON )
```

GO

```
CREATE TABLE Categoria
(
    ID_Categoria INTEGER NOT NULL ,
    Nombre VARCHAR (20) NOT NULL
)
```

GO

```
ALTER TABLE Categoria ADD CONSTRAINT Categoria_PK PRIMARY KEY CLUSTERED
    (ID_Categoria)
    WITH (
        ALLOW_PAGE_LOCKS = ON ,
        ALLOW_ROW_LOCKS = ON )
```

GO

```
CREATE TABLE Cilindrada
(
    ID_Cilindrada INTEGER NOT NULL ,
    Cilindrada VARCHAR (10) NOT NULL ,
    Modelo_ID_Modelo INTEGER NOT NULL
)
```

GO

```
CREATE UNIQUE NONCLUSTERED INDEX
    Cilindrada__IDX ON Cilindrada
```

```
(
    Modelo_ID_Modelo
)
GO
```

```
ALTER TABLE Cilindrada ADD CONSTRAINT Cilindrada_PK PRIMARY KEY
CLUSTERED (ID_Cilindrada)
    WITH (
        ALLOW_PAGE_LOCKS = ON ,
        ALLOW_ROW_LOCKS = ON )
GO
```

```
CREATE TABLE Cliente
(
    DNI INTEGER NOT NULL ,
    Nombre VARCHAR (30) NOT NULL ,
    Apellido VARCHAR (30) ,
    Telefono VARCHAR (20) ,
    Direccion VARCHAR (50) NOT NULL ,
    Pais_ISO VARCHAR (50) NOT NULL
)
GO
```

```
ALTER TABLE Cliente ADD CONSTRAINT Cliente_PK PRIMARY KEY CLUSTERED
(DNI)
    WITH (
        ALLOW_PAGE_LOCKS = ON ,
        ALLOW_ROW_LOCKS = ON )
GO
```

```
CREATE TABLE Comisiones
(
    ID_Comision INTEGER NOT NULL ,
    Comisiones_TYPE VARCHAR (10) NOT NULL
)
GO
```

```
ALTER TABLE Comisiones
    ADD CONSTRAINT CH_INH_Comisiones
    CHECK ( Comisiones_TYPE IN ('Alto', 'Bajo', 'Comisiones', 'Medio') )
```

GO

```
ALTER TABLE Comisiones ADD CONSTRAINT Comisiones_PK PRIMARY KEY
CLUSTERED (ID_Comision)
    WITH (
        ALLOW_PAGE_LOCKS = ON ,
        ALLOW_ROW_LOCKS = ON )
```

GO

```
CREATE TABLE Compra
(
    ID_Compra INTEGER NOT NULL ,
    Fecha DATE NOT NULL ,
    Proovedor_ID_Proovedor INTEGER NOT NULL ,
    ID_Proovedor INTEGER NOT NULL
)
```

GO

```
ALTER TABLE Compra ADD CONSTRAINT Compra_PK PRIMARY KEY CLUSTERED
(ID_Compra)
    WITH (
        ALLOW_PAGE_LOCKS = ON ,
        ALLOW_ROW_LOCKS = ON )
```

GO

```
CREATE TABLE Contrato
(
    ID_Contrato INTEGER NOT NULL ,
    Documento VARBINARY ,
    Empleado_ID_Empleado INTEGER NOT NULL ,
    ID_Comision INTEGER NOT NULL
)
```

GO

```
CREATE UNIQUE NONCLUSTERED INDEX
    Contrato__IDX ON Contrato
(
```

```
Empleado_ID_Empleado  
)
```

GO

```
CREATE TABLE Departamento  
(  
    ID_Departamento INTEGER NOT NULL ,  
    Nombre VARCHAR (30) NOT NULL  
)
```

GO

```
ALTER TABLE Departamento ADD CONSTRAINT Departamento_PK PRIMARY KEY  
CLUSTERED (ID_Departamento)  
WITH (  
    ALLOW_PAGE_LOCKS = ON ,  
    ALLOW_ROW_LOCKS = ON )
```

GO

```
CREATE TABLE Detalle_Compra  
(  
    ID_DC INTEGER NOT NULL ,  
    Cantidad INTEGER NOT NULL ,  
    Precio MONEY NOT NULL ,  
    Compra_ID_Compra INTEGER NOT NULL  
)
```

GO

```
CREATE UNIQUE NONCLUSTERED INDEX  
Detalle_Compra__IDX ON Detalle_Compra  
(  
    Compra_ID_Compra  
)
```

GO

```
ALTER TABLE Detalle_Compra ADD CONSTRAINT Detalle_Compra_PK PRIMARY KEY  
CLUSTERED (ID_DC)  
WITH (  
    ALLOW_PAGE_LOCKS = ON ,  
    ALLOW_ROW_LOCKS = ON )
```

GO

```
CREATE TABLE Detalle_Venta  
(  
    ID_DV INTEGER NOT NULL ,
```

```
        Cantidad INTEGER NOT NULL ,
        Venta_ID_Venta INTEGER NOT NULL
    )
GO

CREATE UNIQUE NONCLUSTERED INDEX
    Detalle_Venta__IDX ON Detalle_Venta
    (
        Venta_ID_Venta
    )
GO

ALTER TABLE Detalle_Venta ADD CONSTRAINT Detalle_Venta_PK PRIMARY KEY
    CLUSTERED (ID_DV)
    WITH (
        ALLOW_PAGE_LOCKS = ON ,
        ALLOW_ROW_LOCKS = ON )
GO

CREATE TABLE Efectivo
    (
        ID_Modalidad_Pago INTEGER NOT NULL ,
        ID_Efectivo INTEGER NOT NULL ,
        Cantidad DECIMAL (28) NOT NULL
    )
GO

ALTER TABLE Efectivo ADD CONSTRAINT Efectivo_PK PRIMARY KEY CLUSTERED
    (ID_Modalidad_Pago)
    WITH (
        ALLOW_PAGE_LOCKS = ON ,
        ALLOW_ROW_LOCKS = ON )
GO

ALTER TABLE Efectivo ADD CONSTRAINT Efectivo_PKv1 UNIQUE NONCLUSTERED
    (ID_Efectivo)
GO

CREATE TABLE Empleado
    (
        ID_Empleado INTEGER NOT NULL ,
```

```
Nombre VARCHAR (20) NOT NULL ,
Apellido VARCHAR (20) NOT NULL ,
Fecha_Ingreso DATE NOT NULL ,
Email VARCHAR (30) ,
Telefono VARCHAR (20) ,
Cargo_ID_Cargo INTEGER NOT NULL ,
Departamento_ID_Departamento INTEGER NOT NULL ,
Comisiones_ID_Comision INTEGER NOT NULL ,
Comisiones_ID_Comision1 INTEGER NOT NULL
)
GO

CREATE UNIQUE NONCLUSTERED INDEX
    Empleado__IDX ON Empleado
    (
        Comisiones_ID_Comision
    )
GO

ALTER TABLE Empleado ADD CONSTRAINT Empleado_PK PRIMARY KEY CLUSTERED
(ID_Empleado)
    WITH (
        ALLOW_PAGE_LOCKS = ON ,
        ALLOW_ROW_LOCKS = ON )
GO

CREATE TABLE Envio
    (
        ID_Envio INTEGER NOT NULL ,
        Fecha DATE NOT NULL
    )
GO

ALTER TABLE Envio ADD CONSTRAINT Envio_PK PRIMARY KEY CLUSTERED
(ID_Envio)
    WITH (
        ALLOW_PAGE_LOCKS = ON ,
        ALLOW_ROW_LOCKS = ON )
GO

CREATE TABLE Fabricante
    (
        ID_Fabricante INTEGER NOT NULL ,
```

```
Nombre VARCHAR (30) NOT NULL ,  
Logo IMAGE ,  
Categoria_ID_Categoria INTEGER NOT NULL  
)  
GO
```

```
CREATE UNIQUE NONCLUSTERED INDEX  
Fabricante__IDX ON Fabricante  
(  
    Categoria_ID_Categoria  
)  
GO
```

```
ALTER TABLE Fabricante ADD CONSTRAINT Fabricante_PK PRIMARY KEY  
CLUSTERED (ID_Fabricante)  
WITH (  
    ALLOW_PAGE_LOCKS = ON ,  
    ALLOW_ROW_LOCKS = ON )  
GO
```

```
CREATE TABLE Inventario  
(  
    ID_Inventario INTEGER NOT NULL ,  
    Cantidad INTEGER NOT NULL ,  
    Fecha DATE NOT NULL  
)  
GO
```

```
ALTER TABLE Inventario ADD CONSTRAINT Inventario_PK PRIMARY KEY  
CLUSTERED (ID_Inventario)  
WITH (  
    ALLOW_PAGE_LOCKS = ON ,  
    ALLOW_ROW_LOCKS = ON )  
GO
```

```
CREATE TABLE Localidad  
(  
    ID_Localidad INTEGER NOT NULL ,  
    Nombre VARCHAR (30) NOT NULL ,  
    Pais_ISO VARCHAR (50) NOT NULL  
)  
GO
```

```
ALTER TABLE Localidad ADD CONSTRAINT Localidad_PK PRIMARY KEY CLUSTERED
```

```
(ID_Localidad)
    WITH (
        ALLOW_PAGE_LOCKS = ON ,
        ALLOW_ROW_LOCKS = ON )
GO
```

```
CREATE TABLE Medio
(
    ID_Comision INTEGER NOT NULL ,
    Porcentaje INTEGER NOT NULL ,
    Descripcion TEXT NOT NULL
)
GO
```

```
ALTER TABLE Medio ADD CONSTRAINT Medio_PK PRIMARY KEY CLUSTERED
(ID_Comision)
    WITH (
        ALLOW_PAGE_LOCKS = ON ,
        ALLOW_ROW_LOCKS = ON )
GO
```

```
ALTER TABLE Medio ADD CONSTRAINT Medio_PKv1 UNIQUE NONCLUSTERED
(Porcentaje)
GO
```

```
CREATE TABLE Mod_Pago
(
    ID_Mod INTEGER NOT NULL ,
    Mod_Pago_TYPE VARCHAR (8) NOT NULL
)
GO
```

```
ALTER TABLE Mod_Pago
    ADD CONSTRAINT CH_INH_Mod_Pago
    CHECK ( Mod_Pago_TYPE IN ('Efectivo', 'Mod_Pago', 'Paypal',
'Tarjeta') )
GO
```

```
ALTER TABLE Mod_Pago ADD CONSTRAINT Mod_Pago_PK PRIMARY KEY CLUSTERED
(ID_Mod)
    WITH (
```



```
    ALLOW_PAGE_LOCKS = ON ,  
    ALLOW_ROW_LOCKS = ON )  
GO
```

```
CREATE TABLE Modelo  
(  
    ID_Modelo INTEGER NOT NULL ,  
    Nombre VARCHAR (30) NOT NULL ,  
    Fecha_Fabricacion DATE ,  
    Fabricante_ID_Fabricante INTEGER NOT NULL ,  
    Categoria_ID_Categoria INTEGER NOT NULL  
)  
GO
```

```
ALTER TABLE Modelo ADD CONSTRAINT Modelo_PK PRIMARY KEY CLUSTERED  
(ID_Modelo)  
    WITH (  
        ALLOW_PAGE_LOCKS = ON ,  
        ALLOW_ROW_LOCKS = ON )  
GO
```

```
CREATE TABLE Municipio  
(  
    ID_Municipio INTEGER NOT NULL ,  
    Nombre VARCHAR (30) NOT NULL ,  
    Localidad_ID_Localidad INTEGER NOT NULL  
)  
GO
```

```
ALTER TABLE Municipio ADD CONSTRAINT Municipio_PK PRIMARY KEY CLUSTERED  
(ID_Municipio)  
    WITH (  
        ALLOW_PAGE_LOCKS = ON ,  
        ALLOW_ROW_LOCKS = ON )  
GO
```

```
CREATE TABLE Pago  
(  
    ID_Pago INTEGER NOT NULL ,  
    Fecha DATE NOT NULL ,  
    Modalidad_Pago_ID_Mod INTEGER NOT NULL ,  
    CarritoWeb_CarritoWeb_ID NUMERIC (28) NOT NULL ,  
    Mod_Pago_ID_Mod INTEGER NOT NULL  
)
```

GO

```
CREATE UNIQUE NONCLUSTERED INDEX
    Pago__IDX ON Pago
    (
        CarritoWeb_CarritoWeb_ID
    )
```

GO

```
ALTER TABLE Pago ADD CONSTRAINT Pago_PK PRIMARY KEY CLUSTERED
    (Mod_Pago_ID_Mod)
    WITH (
        ALLOW_PAGE_LOCKS = ON ,
        ALLOW_ROW_LOCKS = ON )
```

GO

```
CREATE TABLE Pais
    (
        ISO VARCHAR (50) NOT NULL ,
        Nombre VARCHAR (30) NOT NULL
    )
```

GO

```
ALTER TABLE Pais ADD CONSTRAINT Pais_PK PRIMARY KEY CLUSTERED (ISO)
    WITH (
        ALLOW_PAGE_LOCKS = ON ,
        ALLOW_ROW_LOCKS = ON )
```

GO

```
CREATE TABLE Paypal
    (
        ID_Modalidad_Pago INTEGER NOT NULL ,
        Email VARCHAR (30) NOT NULL ,
        Usuario VARCHAR (20) NOT NULL
    )
```

GO

```
ALTER TABLE Paypal ADD CONSTRAINT Paypal_PK PRIMARY KEY CLUSTERED
    (ID_Modalidad_Pago)
    WITH (
        ALLOW_PAGE_LOCKS = ON ,
        ALLOW_ROW_LOCKS = ON )
```

GO

```
ALTER TABLE Paypal ADD CONSTRAINT Paypal_PKv1 UNIQUE NONCLUSTERED
```

(Email)

GO

CREATE TABLE Proovedor

```
(
  ID_Proovedor INTEGER NOT NULL ,
  Nombre VARCHAR (30) NOT NULL ,
  Direccion VARCHAR (50) ,
  Web VARCHAR (30) ,
  Telefono VARCHAR (20) ,
  Email VARCHAR (20) ,
  Pais_ISO VARCHAR (50) NOT NULL
)
```

GO

ALTER TABLE Proovedor ADD CONSTRAINT Proovedor\_PK PRIMARY KEY CLUSTERED  
(ID\_Proovedor)

```
WITH (
  ALLOW_PAGE_LOCKS = ON ,
  ALLOW_ROW_LOCKS = ON )
```

GO

CREATE TABLE Tarjeta

```
(
  ID_Modalidad_Pago INTEGER NOT NULL ,
  ID_Tarjeta INTEGER NOT NULL ,
  Numero_Tarjeta INTEGER NOT NULL ,
  CCV INTEGER NOT NULL
)
```

GO

ALTER TABLE Tarjeta ADD CONSTRAINT Tarjeta\_PK PRIMARY KEY CLUSTERED  
(ID\_Modalidad\_Pago)

```
WITH (
  ALLOW_PAGE_LOCKS = ON ,
  ALLOW_ROW_LOCKS = ON )
```

GO

ALTER TABLE Tarjeta ADD CONSTRAINT Tarjeta\_PKv1 UNIQUE NONCLUSTERED  
(ID\_Tarjeta)

GO

CREATE TABLE Venta

```
(
  ID_Venta INTEGER NOT NULL ,
```

```
        Fecha DATE NOT NULL ,
        Cliente_DNI INTEGER NOT NULL ,
        Total_Venta DECIMAL (28)
    )
GO

ALTER TABLE Venta ADD CONSTRAINT Venta_PK PRIMARY KEY CLUSTERED
(ID_Venta)
    WITH (
        ALLOW_PAGE_LOCKS = ON ,
        ALLOW_ROW_LOCKS = ON )
GO

ALTER TABLE Alto
    ADD CONSTRAINT Alto_Comisiones_FK FOREIGN KEY
    (
        ID_Comision
    )
    REFERENCES Comisiones
    (
        ID_Comision
    )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
GO

ALTER TABLE Articulo
    ADD CONSTRAINT Articulo_Detalle_Compra_FK FOREIGN KEY
    (
        Detalle_Compra_ID_DC
    )
    REFERENCES Detalle_Compra
    (
        ID_DC
    )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
GO

ALTER TABLE Articulo
    ADD CONSTRAINT Articulo_Detalle_Venta_FK FOREIGN KEY
    (
        Detalle_Venta_ID_DV
    )
```

```
REFERENCES Detalle_Venta
(
    ID_DV
)
ON DELETE NO ACTION
ON UPDATE NO ACTION
GO

ALTER TABLE Artículo
ADD CONSTRAINT Articulo_Inventario_FK FOREIGN KEY
(
    Inventario_ID_Inventario
)
REFERENCES Inventario
(
    ID_Inventario
)
ON DELETE NO ACTION
ON UPDATE NO ACTION
GO

ALTER TABLE Artículo
ADD CONSTRAINT Articulo_Modelo_FK FOREIGN KEY
(
    Modelo_ID_Modelo
)
REFERENCES Modelo
(
    ID_Modelo
)
ON DELETE NO ACTION
ON UPDATE NO ACTION
GO

ALTER TABLE Bajo
ADD CONSTRAINT Bajov1_Comisiones_FK FOREIGN KEY
(
    ID_Comision
)
REFERENCES Comisiones
(
    ID_Comision
)
ON DELETE NO ACTION
```

```
        ON UPDATE NO ACTION
GO
```

```
ALTER TABLE CarritoWeb
    ADD CONSTRAINT CarritoWeb_Cliente_FK FOREIGN KEY
    (
        Cliente_DNI
    )
    REFERENCES Cliente
    (
        DNI
    )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
GO
```

```
ALTER TABLE CarritoWeb
    ADD CONSTRAINT CarritoWeb_Envio_FK FOREIGN KEY
    (
        Envio_ID_Envio
    )
    REFERENCES Envio
    (
        ID_Envio
    )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
GO
```

```
ALTER TABLE Cilindrada
    ADD CONSTRAINT Cilindrada_Modelo_FK FOREIGN KEY
    (
        Modelo_ID_Modelo
    )
    REFERENCES Modelo
    (
        ID_Modelo
    )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
GO
```

```
ALTER TABLE Cliente
    ADD CONSTRAINT Cliente_Pais_FK FOREIGN KEY
```

```
(
    Pais_ISO
)
REFERENCES Pais
(
    ISO
)
ON DELETE NO ACTION
ON UPDATE NO ACTION
GO
```

```
ALTER TABLE Compra
    ADD CONSTRAINT Compra_Proovedor_FK FOREIGN KEY
    (
        Proovedor_ID_Proovedor
    )
    REFERENCES Proovedor
    (
        ID_Proovedor
    )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
GO
```

```
ALTER TABLE Contrato
    ADD CONSTRAINT Contrato_Empleado_FK FOREIGN KEY
    (
        Empleado_ID_Empleado
    )
    REFERENCES Empleado
    (
        ID_Empleado
    )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
GO
```

```
ALTER TABLE Detalle_Compra
    ADD CONSTRAINT Detalle_Compra_Compra_FK FOREIGN KEY
    (
        Compra_ID_Compra
    )
    REFERENCES Compra
    (
```

```
        ID_Compra
    )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
GO
```

```
ALTER TABLE Detalle_Venta
    ADD CONSTRAINT Detalle_Venta_Venta_FK FOREIGN KEY
    (
        Venta_ID_Venta
    )
    REFERENCES Venta
    (
        ID_Venta
    )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
GO
```

```
ALTER TABLE Efectivo
    ADD CONSTRAINT Efectivo_Modalidad_Pago_FK FOREIGN KEY
    (
        ID_Modalidad_Pago
    )
    REFERENCES Mod_Pago
    (
        ID_Mod
    )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
GO
```

```
ALTER TABLE Empleado
    ADD CONSTRAINT Empleado_Cargo_FK FOREIGN KEY
    (
        Cargo_ID_Cargo
    )
    REFERENCES Cargo
    (
        ID_Cargo
    )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
GO
```



```
ALTER TABLE Empleado
  ADD CONSTRAINT Empleado_Comisiones_FK FOREIGN KEY
  (
    Comisiones_ID_Comision
  )
  REFERENCES Comisiones
  (
    ID_Comision
  )
  ON DELETE NO ACTION
  ON UPDATE NO ACTION
GO
```

```
ALTER TABLE Empleado
  ADD CONSTRAINT Empleado_Comisiones_FKv1 FOREIGN KEY
  (
    Comisiones_ID_Comision1
  )
  REFERENCES Comisiones
  (
    ID_Comision
  )
  ON DELETE NO ACTION
  ON UPDATE NO ACTION
GO
```

```
ALTER TABLE Empleado
  ADD CONSTRAINT Empleado_Departamento_FK FOREIGN KEY
  (
    Departamento_ID_Departamento
  )
  REFERENCES Departamento
  (
    ID_Departamento
  )
  ON DELETE NO ACTION
  ON UPDATE NO ACTION
GO
```

```
ALTER TABLE Fabricante
  ADD CONSTRAINT Fabricante_Categoria_FK FOREIGN KEY
  (
    Categoria_ID_Categoria
  )
```

```
)  
REFERENCES Categoria  
(  
    ID_Categoria  
)  
ON DELETE NO ACTION  
ON UPDATE NO ACTION  
GO
```

```
ALTER TABLE Localidad  
    ADD CONSTRAINT Localidad_Pais_FK FOREIGN KEY  
    (  
        Pais_ISO  
    )  
    REFERENCES Pais  
    (  
        ISO  
    )  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION  
GO
```

```
ALTER TABLE Medio  
    ADD CONSTRAINT Medio_Comisiones_FK FOREIGN KEY  
    (  
        ID_Comision  
    )  
    REFERENCES Comisiones  
    (  
        ID_Comision  
    )  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION  
GO
```

```
ALTER TABLE Modelo  
    ADD CONSTRAINT Modelo_Categoria_FK FOREIGN KEY  
    (  
        Categoria_ID_Categoria  
    )  
    REFERENCES Categoria  
    (  
        ID_Categoria  
    )
```

```
ON DELETE NO ACTION
ON UPDATE NO ACTION
GO
```

```
ALTER TABLE Modelo
  ADD CONSTRAINT Modelo_Fabricante_FK FOREIGN KEY
  (
    Fabricante_ID_Fabricante
  )
  REFERENCES Fabricante
  (
    ID_Fabricante
  )
  ON DELETE NO ACTION
  ON UPDATE NO ACTION
GO
```

```
ALTER TABLE Municipio
  ADD CONSTRAINT Municipio_Localidad_FK FOREIGN KEY
  (
    Localidad_ID_Localidad
  )
  REFERENCES Localidad
  (
    ID_Localidad
  )
  ON DELETE NO ACTION
  ON UPDATE NO ACTION
GO
```

```
ALTER TABLE Pago
  ADD CONSTRAINT Pago_CarritoWeb_FK FOREIGN KEY
  (
    CarritoWeb_CarritoWeb_ID
  )
  REFERENCES CarritoWeb
  (
    CarritoWeb_ID
  )
  ON DELETE NO ACTION
  ON UPDATE NO ACTION
GO
```

```
ALTER TABLE Pago
```

```
ADD CONSTRAINT Pago_Mod_Pago_FK FOREIGN KEY
(
    Mod_Pago_ID_Mod
)
REFERENCES Mod_Pago
(
    ID_Mod
)
ON DELETE NO ACTION
ON UPDATE NO ACTION
GO

ALTER TABLE Pago
ADD CONSTRAINT Pago_Modalidad_Pago_FK FOREIGN KEY
(
    Modalidad_Pago_ID_Mod
)
REFERENCES Mod_Pago
(
    ID_Mod
)
ON DELETE NO ACTION
ON UPDATE NO ACTION
GO

ALTER TABLE Paypal
ADD CONSTRAINT Paypal_Modalidad_Pago_FK FOREIGN KEY
(
    ID_Modalidad_Pago
)
REFERENCES Mod_Pago
(
    ID_Mod
)
ON DELETE NO ACTION
ON UPDATE NO ACTION
GO

ALTER TABLE Proveedor
ADD CONSTRAINT Proveedor_Pais_FK FOREIGN KEY
(
    Pais_ISO
)
REFERENCES Pais
```

```
(
  ISO
)
ON DELETE NO ACTION
ON UPDATE NO ACTION
GO
```

```
ALTER TABLE Tarjeta
  ADD CONSTRAINT Tarjeta_Modalidad_Pago_FK FOREIGN KEY
  (
    ID_Modalidad_Pago
  )
  REFERENCES Mod_Pago
  (
    ID_Mod
  )
  ON DELETE NO ACTION
  ON UPDATE NO ACTION
GO
```

```
ALTER TABLE Venta
  ADD CONSTRAINT Venta_Cliente_FK FOREIGN KEY
  (
    Cliente_DNI
  )
  REFERENCES Cliente
  (
    DNI
  )
  ON DELETE NO ACTION
  ON UPDATE NO ACTION
GO
```

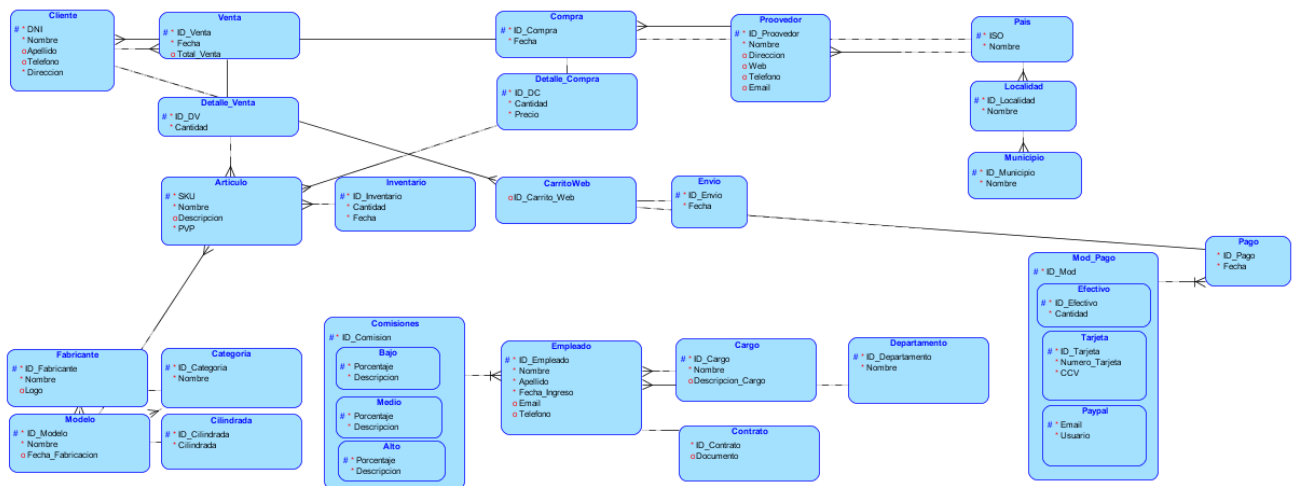
```
-- Informe de Resumen de Oracle SQL Developer Data Modeler:
--
-- CREATE TABLE                      30
-- CREATE INDEX                       8
-- ALTER TABLE                      69
-- CREATE VIEW                        0
-- ALTER VIEW                         0
-- CREATE PACKAGE                     0
-- CREATE PACKAGE BODY                 0
```

-- CREATE PROCEDURE	0
-- CREATE FUNCTION	0
-- CREATE TRIGGER	0
-- ALTER TRIGGER	0
-- CREATE DATABASE	0
-- CREATE DEFAULT	0
-- CREATE INDEX ON VIEW	0
-- CREATE ROLLBACK SEGMENT	0
-- CREATE ROLE	0
-- CREATE RULE	0
-- CREATE SCHEMA	0
-- CREATE SEQUENCE	0
-- CREATE PARTITION FUNCTION	0
-- CREATE PARTITION SCHEME	0
--	
-- DROP DATABASE	0
--	
-- ERRORS	0
-- WARNINGS	0

## 2.0 MODELADO DE DATOS

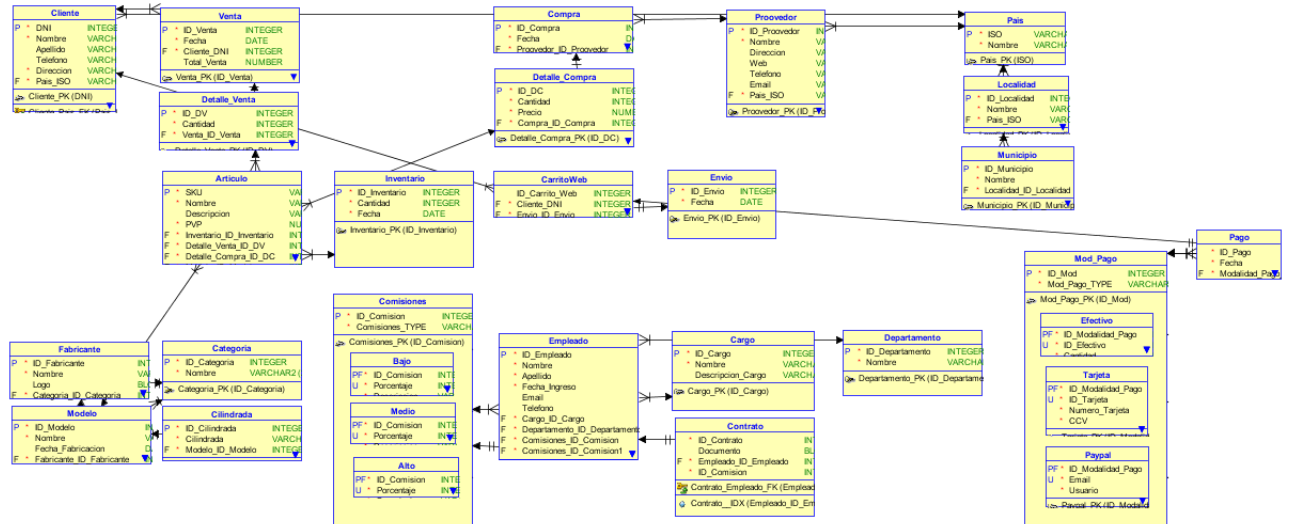
## 2.1 MODELO LÓGICO

[Volver al Indice →](#)



## 2.1.2 MODELO RELACIONAL

[Volver al Índice →](#)



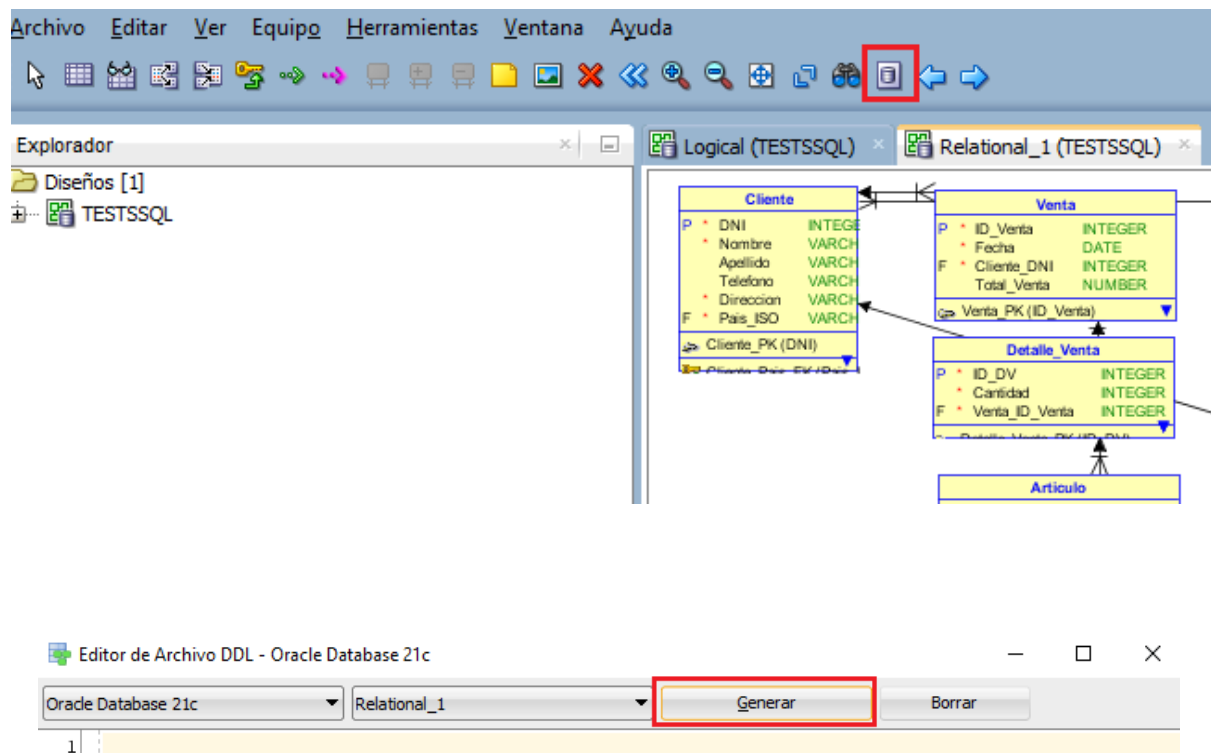


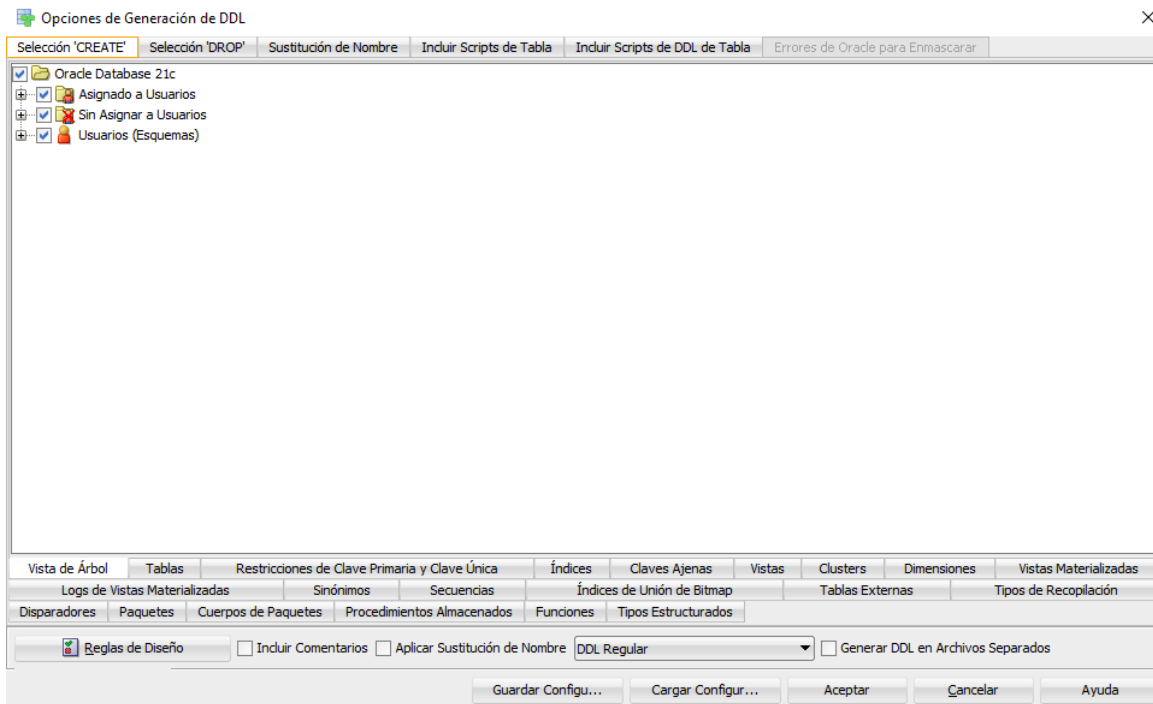
## 2.2 DE SQL DATA MODELER A SQL

[Volver al Índice →](#)

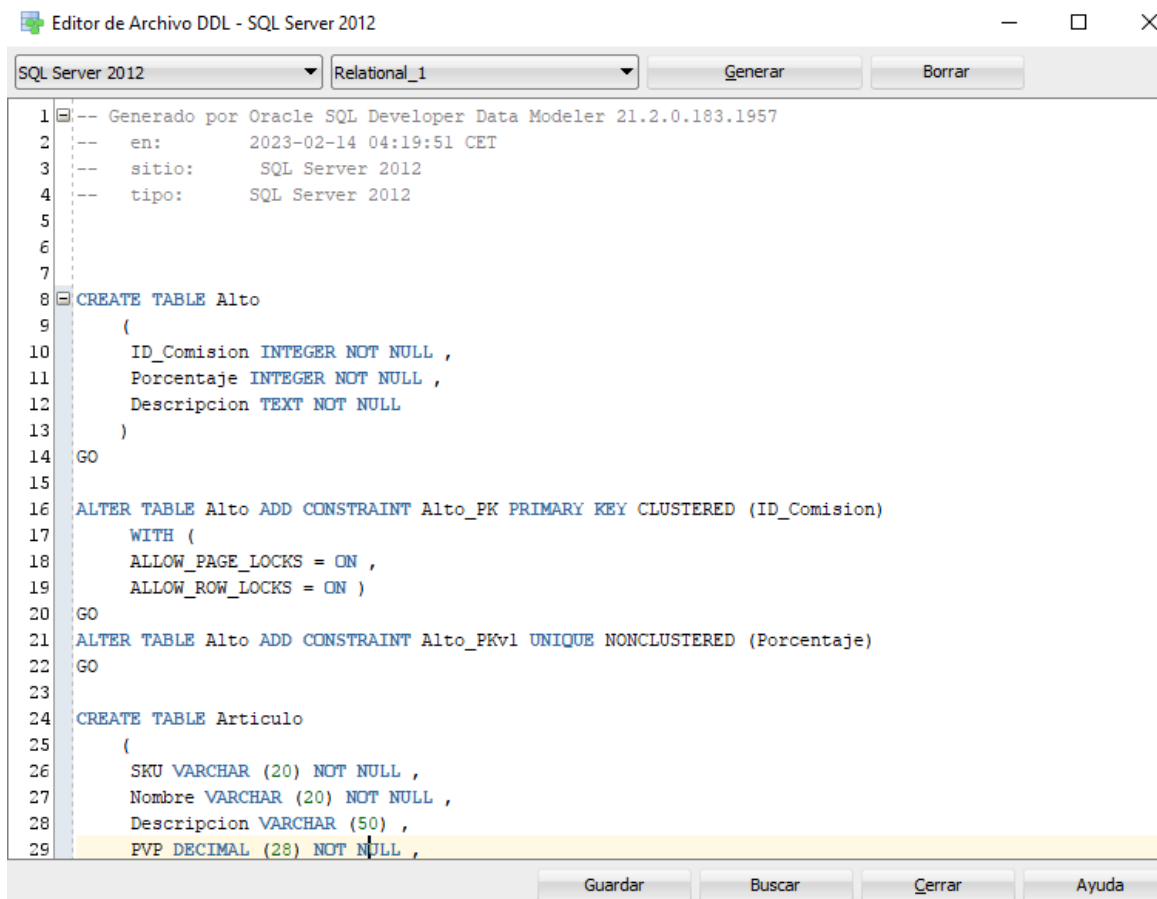
### 2.2.1 DATA MODELER

Una vez hemos creado nuestra base de datos, procedemos a realizar los siguientes pasos, los cuales generarán un script que nos permitirá transportar nuestro modelado a SQL server





Luego de generar dicho fichero e indicar cuales son los parámetros que deseamos asignar al ddl, tendremos un script de nuestra base de datos.

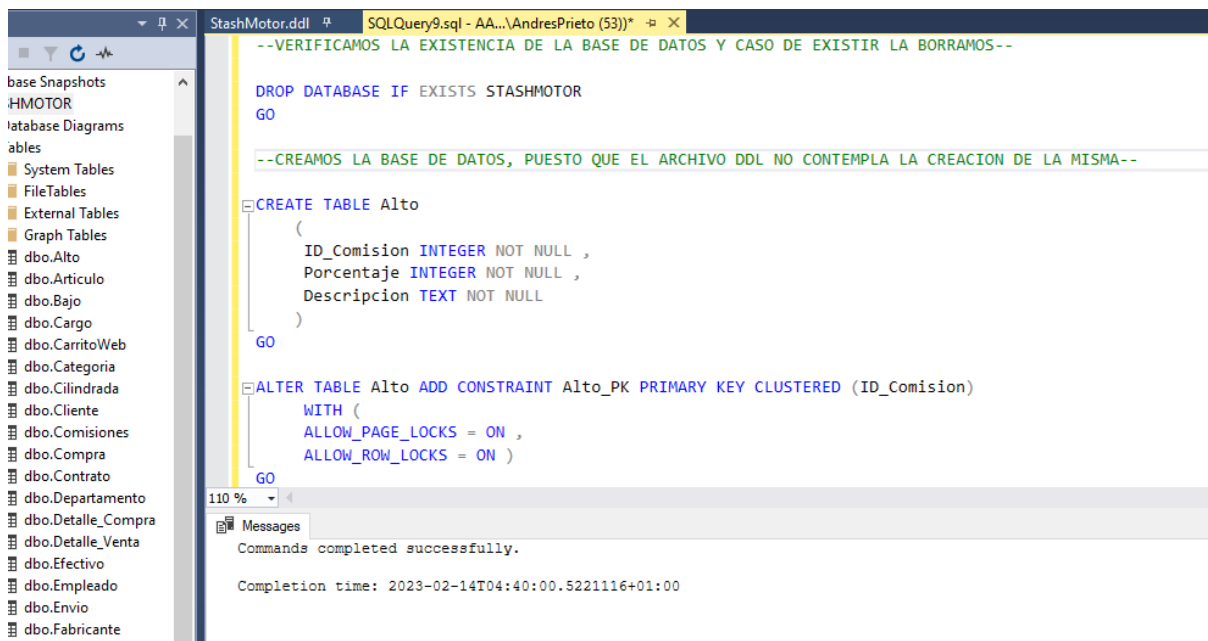


## 2.2.3 SQL SERVER

[Volver al Índice →](#)

Una vez verificada la existencia de la base de datos y la creación de la misma (debemos hacerlo manualmente ya que el fichero que se generó anteriormente no contempla la creación de la misma.)

A continuación copiamos el script generado y verificamos la correcta ejecución de nuestra base de datos.



```
--VERIFICAMOS LA EXISTENCIA DE LA BASE DE DATOS Y CASO DE EXISTIR LA BORRAMOS--  
  
DROP DATABASE IF EXISTS STASHMOTOR  
GO  
  
--CREAMOS LA BASE DE DATOS, PUESTO QUE EL ARCHIVO DDL NO CONTEMPLA LA CREACION DE LA MISMA--  
  
CREATE TABLE Alto  
(  
    ID_Comision INTEGER NOT NULL ,  
    Porcentaje INTEGER NOT NULL ,  
    Descripcion TEXT NOT NULL  
)  
GO  
  
ALTER TABLE Alto ADD CONSTRAINT Alto_PK PRIMARY KEY CLUSTERED (ID_Comision)  
WITH (  
    ALLOW_PAGE_LOCKS = ON ,  
    ALLOW_ROW_LOCKS = ON )  
GO
```

110 %

Messages

Commands completed successfully.

Completion time: 2023-02-14T04:40:00.5221116+01:00

## 2.3 DE SQL DATA MODELER A SQL

[Volver al Índice →](#)

Para ello debemos inicialmente asegurarnos de contar con lo siguiente:

1. Credenciales de Oracle para conectar con SQL Server
2. Correcto funcionamiento de los servicios de SQL Server
3. Activación y correcta configuración del usuario SA
4. Configuración del Firewall (puerto 1433)

Una vez nos aseguramos de contar con todo lo anterior procedemos a la importación de la base de datos.

de Importación de Diccionario de Datos

The screenshot shows a dialog box titled "Nueva / Seleccionar Conexión a Base de Datos". It has a tabbed interface with "Nombre de Cone..." and "Detalles de Cone...". The "Detalles de Cone..." tab is active, showing the following fields and options:

- Name:** A text input field.
- Tipo de Base de Datos:** A dropdown menu set to "SQLServer".
- Usuario:** A text input field containing "sa".
- Contraseña:** A text input field with masked characters (dots).
- ☐ **Guardar Contraseña**
- ☐ **Usar Contraseña por Defecto**
- ☐ **Usar Autenticación de Windows**
- Nombre del Host:** A text input field containing "localhost".
- Puerto:** A text input field containing "1433".

At the bottom, there is an "Estado:" label and a row of buttons: "Ayuda", "Guardar", "Borrar", "Probar", "Conectar", and "Cancelar".

Una vez se establezca la conexión seleccionamos la base de datos que deseamos importar así como los objetos de la misma.

Asistente de Importación de Diccionario de Datos

Seleccione el esquema/base de datos que desea importar.

Selección	Nombre	Tipo
<input type="checkbox"/>	<b>msdb</b>	DATABASE
<input type="checkbox"/>	DatabaseMailUserRole	Schema
<input type="checkbox"/>	db_ssisadmin	Schema
<input type="checkbox"/>	db_ssisdtuser	Schema
<input type="checkbox"/>	db_ssisoperator	Schema
<input type="checkbox"/>	dbo	Schema
<input type="checkbox"/>	guest	Schema
<input type="checkbox"/>	managed_backup	Schema
<input type="checkbox"/>	smart_admin	Schema
<input type="checkbox"/>	SQLAgentOperatorRole	Schema
<input type="checkbox"/>	SQLAgentReaderRole	Schema
<input type="checkbox"/>	SQLAgentUserRole	Schema
<input type="checkbox"/>	sys	Schema
<input type="checkbox"/>	TargetServersRole	Schema
<input checked="" type="checkbox"/>	<b>STASHMOTOR</b>	DATABASE
<input checked="" type="checkbox"/>	dbo	Schema
<input checked="" type="checkbox"/>	guest	Schema
<input checked="" type="checkbox"/>	sys	Schema
<input type="checkbox"/>	<b>tempdb</b>	DATABASE
<input type="checkbox"/>	dbo	Schema
<input type="checkbox"/>	guest	Schema
<input type="checkbox"/>	sys	Schema

Filtro:  ☐ Todos Seleccionados

Importar a: **Relational\_1** ☐ Intercambiar Modelo de Destino **SQL Server 2008**

Asistente de Importación de Diccionario de Datos

Seleccione los objetos que desea importar.

Selección	Base de Datos	Usuario	Nombre de Objeto
<input checked="" type="checkbox"/>	STASHMOTOR	dbo	Alto
<input checked="" type="checkbox"/>	STASHMOTOR	dbo	Articulo
<input checked="" type="checkbox"/>	STASHMOTOR	dbo	Bajo
<input checked="" type="checkbox"/>	STASHMOTOR	dbo	Cargo
<input checked="" type="checkbox"/>	STASHMOTOR	dbo	CarritoWeb
<input checked="" type="checkbox"/>	STASHMOTOR	dbo	Categoria
<input checked="" type="checkbox"/>	STASHMOTOR	dbo	Cilindrada
<input checked="" type="checkbox"/>	STASHMOTOR	dbo	Cliente
<input checked="" type="checkbox"/>	STASHMOTOR	dbo	Comisiones
<input checked="" type="checkbox"/>	STASHMOTOR	dbo	Compra
<input checked="" type="checkbox"/>	STASHMOTOR	dbo	Contrato
<input checked="" type="checkbox"/>	STASHMOTOR	dbo	Departamento
<input checked="" type="checkbox"/>	STASHMOTOR	dbo	Detalle_Compra
<input checked="" type="checkbox"/>	STASHMOTOR	dbo	Detalle_Venta
<input checked="" type="checkbox"/>	STASHMOTOR	dbo	Efectivo
<input checked="" type="checkbox"/>	STASHMOTOR	dbo	Empleado
<input checked="" type="checkbox"/>	STASHMOTOR	dbo	Envio
<input checked="" type="checkbox"/>	STASHMOTOR	dbo	Fabricante
<input checked="" type="checkbox"/>	STASHMOTOR	dbo	Inventario
<input checked="" type="checkbox"/>	STASHMOTOR	dbo	Localidad
<input checked="" type="checkbox"/>	STASHMOTOR	dbo	Medio
<input checked="" type="checkbox"/>	STASHMOTOR	dbo	Mod_Pago
<input checked="" type="checkbox"/>	STASHMOTOR	dbo	Modelo
<input checked="" type="checkbox"/>	STASHMOTOR	dbo	Municipio

Tablas Vistas Conexiones Usuarios Roles Reglas Valores por Defecto Procedimientos Secuencias Tipos de Dato

Filtro:

