

UNIVERSIDAD NACIONAL DE INGENIERÍA
CENTRO DE PROYECCIÓN Y RESPONSABILIDAD SOCIAL
UNIVERSITARIA

CEPS - UNI



MICROSOFT SQL SERVER IV – INTELIGENCIA DE NEGOCIOS
INFORME N°2

INTEGRANTES:

- **BARZOLA BENITO, KEF MAEKEL**
- **QUISPE MESTANZA, KEVIN FABIAN**
- **FALCON ASCONA, ALEX MAXIMILIANO**

DOCENTE:

- **ERIC GUSTAVO CORONEL CASTILLO**

LIMA - PERÚ

2021

CONTENIDO

1.	CREACION DE LA TABLA DIMENSIONAL	4
2.	CARGA DE DATOS.....	5
3.	VERIFICACIÓN.....	16
4.	ENLACE:	17

JARDINEROS UNIDOS SAC.



A través del presente informe daremos a conocer los procedimientos previos para cargar los datos en nuestro Modelo Dimensional de la base de datos Jardinería, el cual ha sido detallado en el informe previo, desde el programa Visual Studio 2019.

DimEmpleado	Persona el cual ofrecen nuestro producto, encargado de realizar las transacciones.
DimCliente	Persona o entidad el cual adquiere nuestro producto.
DimProducto	Objeto o material que se ofrece y adquiere los clientes.
DimTiempo	Periodos de tiempo que definen el flujo de las transacciones.
DimHVenta	Proceso el cual los clientes obtienen nuestros productos.

1. CREACION DE LA TABLA DIMENSIONAL

Se procede a crear la tabla dimensional “**Dimensiones**” en el SQL Server con los atributos que fueron indicados en el informe anterior,

```
create table dimempleado
(
  idempleado      varchar(25) not null      primary key,
  nombreemple     varchar(25) not null,
  apellidoemple   varchar(25) not null,
  localidademple  varchar(25) not null,
  puesto          varchar(25) not null,
  telefonticparticular varchar(25) not null)

```

Figura 1. Tabla DimEmpleado.

```
create table dimcliente
(
  idcliente      varchar(25) not null      primary key,
  nombrecliente  varchar(25) not null,
  apellidocliente varchar(25) not null,
  direccioncliente varchar(25) not null,
  distritocliente varchar(25) not null,
  paiscliente    varchar(25) not null)

```

Figura 2. Tabla DimCliente.

```
create table dimproducto
(
  idproducto      varchar(25) not null      primary key,
  nombreproducto  varchar(25) not null,
  precioproducto  money      not null,
  stock           int        not null,
  fechacompra     date       not null)

```

Figura 3. Tabla DimProducto.

```
create table dimtiempo
(
  idtiempo      varchar(25) not null      primary key,
  fecha         datetime    not null,
  ndia          tinyint     not null,
  nmes          varchar(25) not null,
  año           smallint    not null)

```

Figura 4. Tabla DimTiempo.

```
create table hventas
(
    idventas varchar(25) not null primary key,
    idcliente varchar(25) not null,
    idproducto varchar(25) not null,
    idempleado varchar(25) not null,
    idtiempo varchar(25) not null,
    cantidad int not null,
    total int not null
)
```

Figura 5. Tabla DimHVenta.

2. CARGA DE DATOS

Después de crear la base de datos destino, “**Dimensiones**” (Modelo Dimensional), se procede a realizar la conexión con la base de datos fuente, Jardinería (Modelo Transaccional), a través del Visual Studio 2019.

a) Dimensión Cliente:

1. En el codificador del flujo de control agregamos una Tarea Ejecutar SQL, para limpiar la tabla y que no se repitan los datos, y una Tarea flujo de datos (**Cliente**) donde podremos cargar los datos.

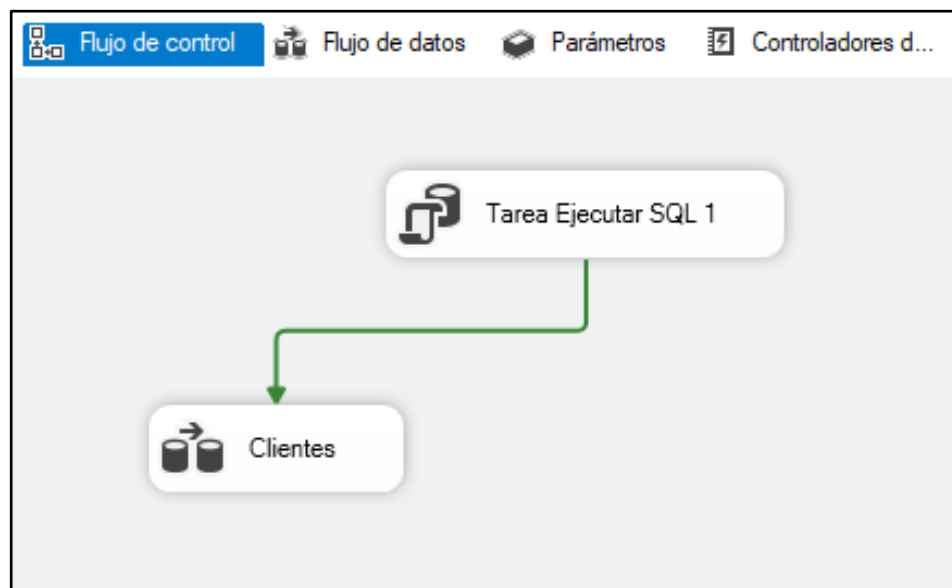


Figura 6. Flujo de control Cliente.

2. En el flujo de datos de Datos (**Cliente**) agregaremos un origen OLE DB y un destino OLE DB.

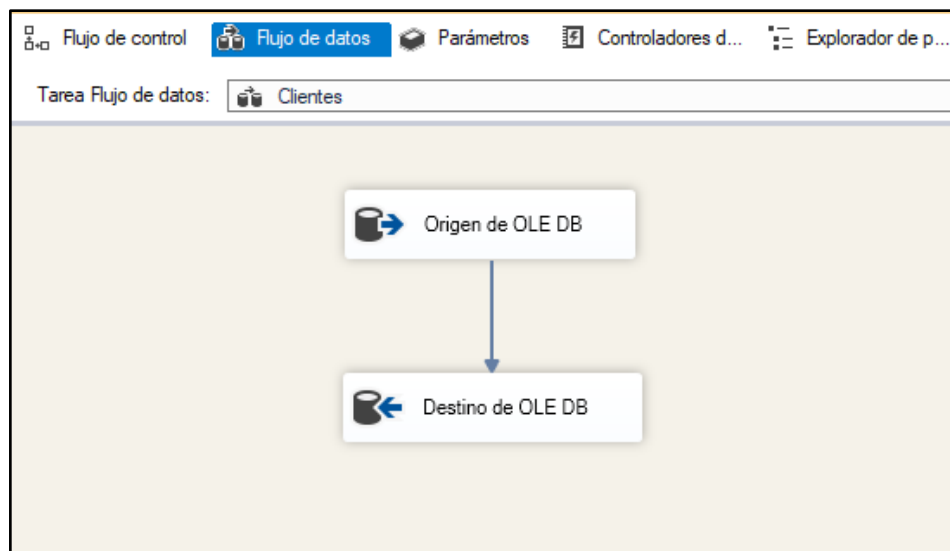


Figura 7. Flujo de datos Cliente.

3. En el origen OLE DB usamos la conexión creada y usaremos el modo de acceso de comando SQL como origen.

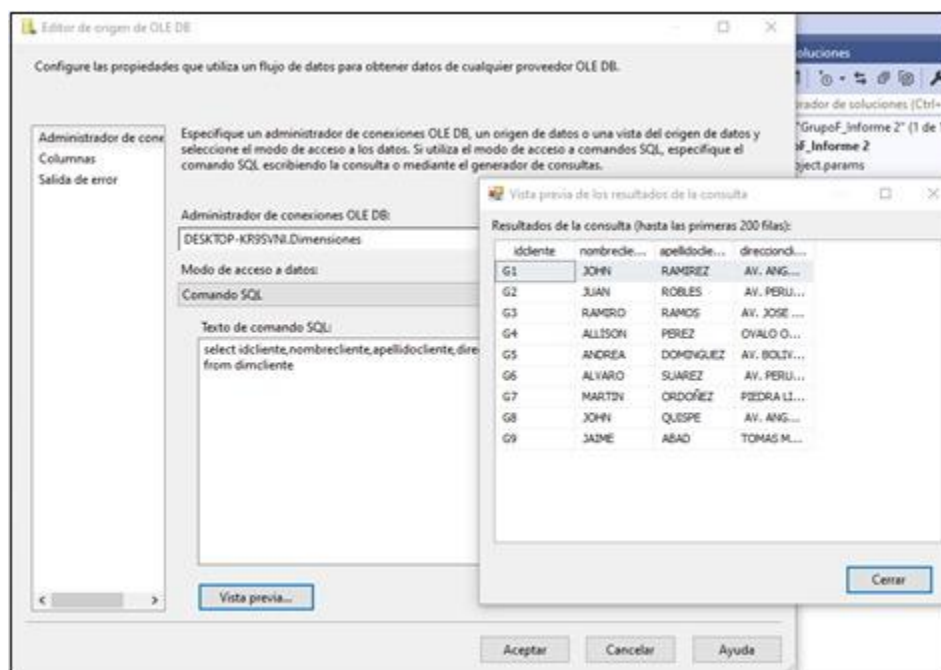


Figura 8. Conexión origen OLE DB Cliente.

4. En el destino OLE DB igualmente usamos la conexión creada en SQL – **DimCliente** y la vamos a generar la relación directa.

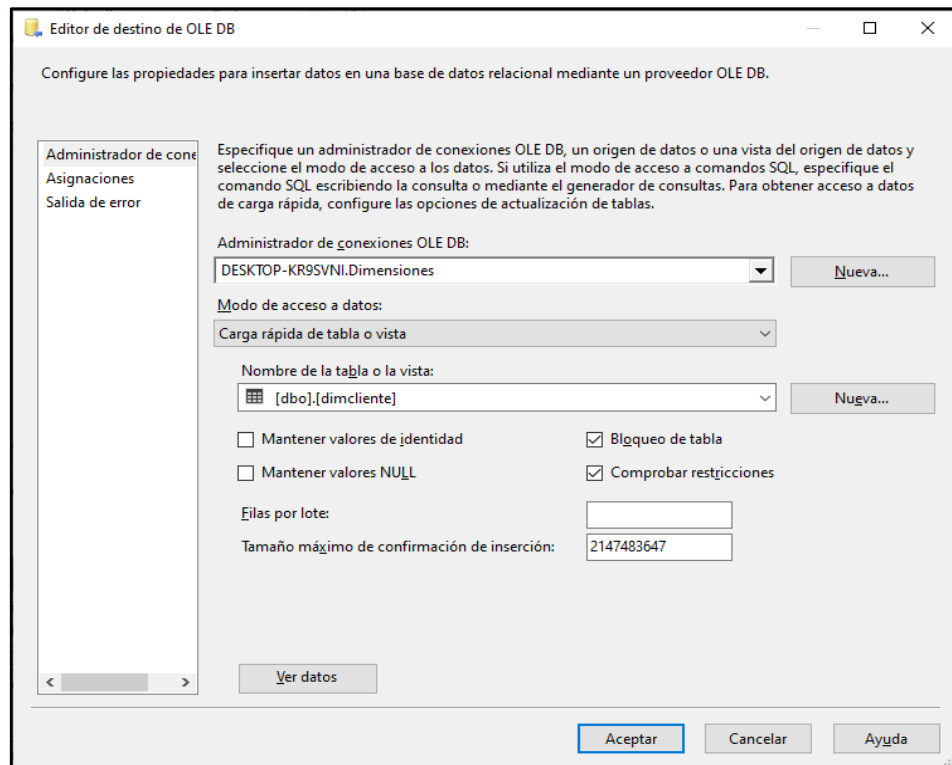


Figura 9. Conexión destino OLE DB Cliente.

b) Dimensión Empleado:

1. En el codificador del flujo de control agregamos una Tarea Ejecutar SQL, para limpiar la tabla y que no se repitan los datos, y una Tarea flujo de datos (**Empleado**) donde podremos cargar los datos.

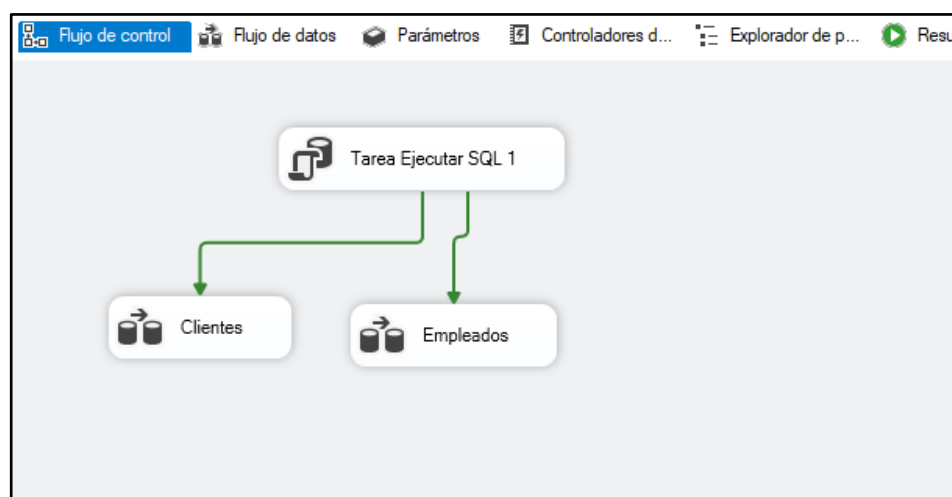


Figura 10. Flujo de control Empleado

2. En el flujo de datos de Datos (**Empleado**) agregaremos un origen OLE DB

y un destino OLE DB.

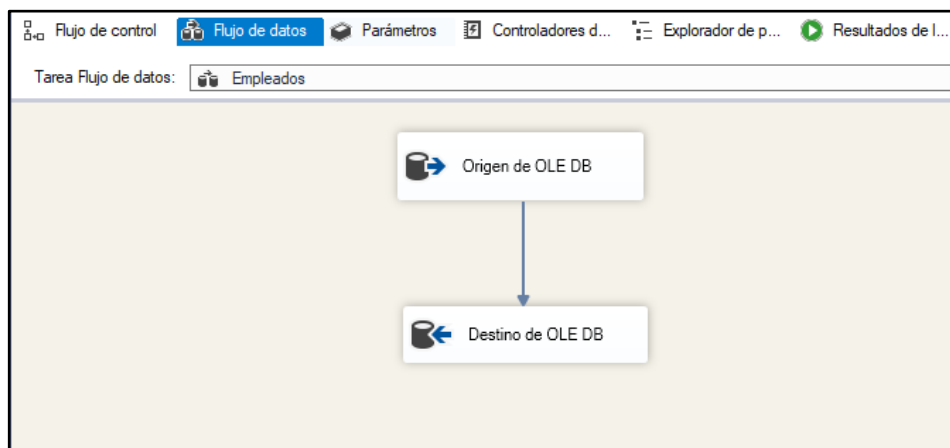


Figura 11. Flujo de datos Empleado.

3. En el origen OLE DB usamos la conexión creada y usaremos el modo de acceso de comando SQL como origen.

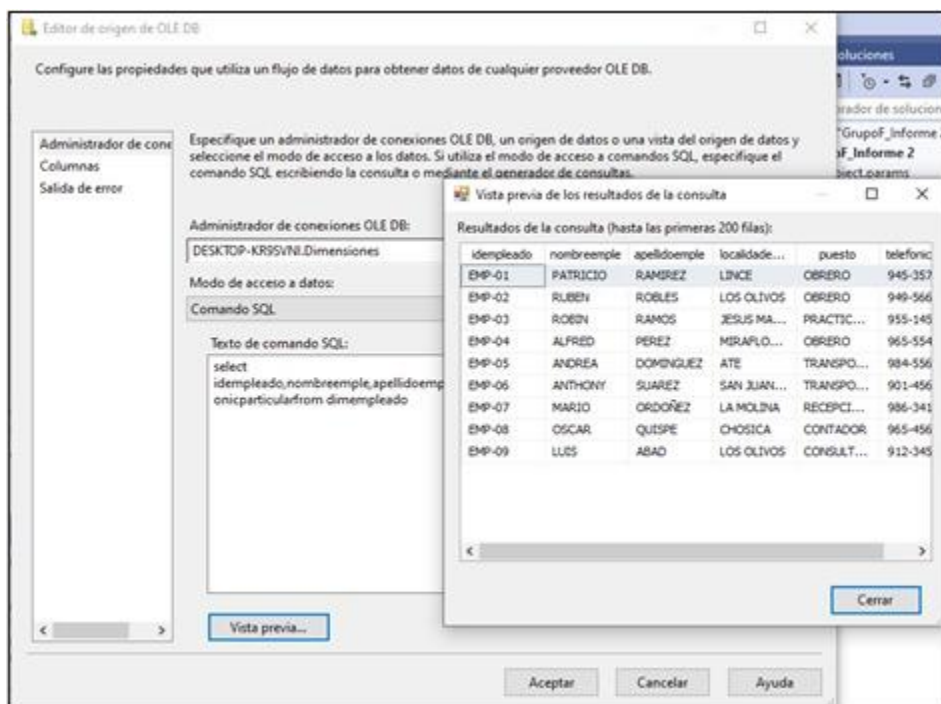


Figura 12. Conexión origen OLE DB Empleado.

4. En el destino OLE DB igualmente usamos la conexión creada en SQL – **DimEmpleado** y la vamos a generar la relación directa.

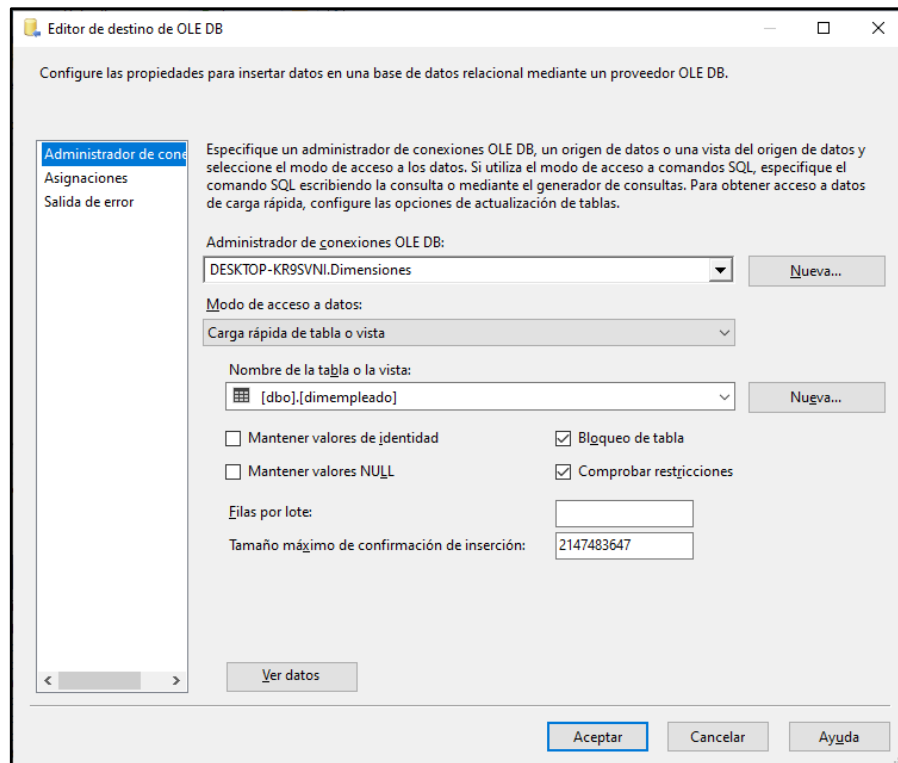


Figura 13. Conexión destino OLE DB Empleado.

c) Dimensión Producto:

1. En el codificador del flujo de control agregamos una Tarea Ejecutar SQL, para limpiar la tabla y que no se repitan los datos, y una Tarea flujo de datos (**Producto**) donde podremos cargar los datos.

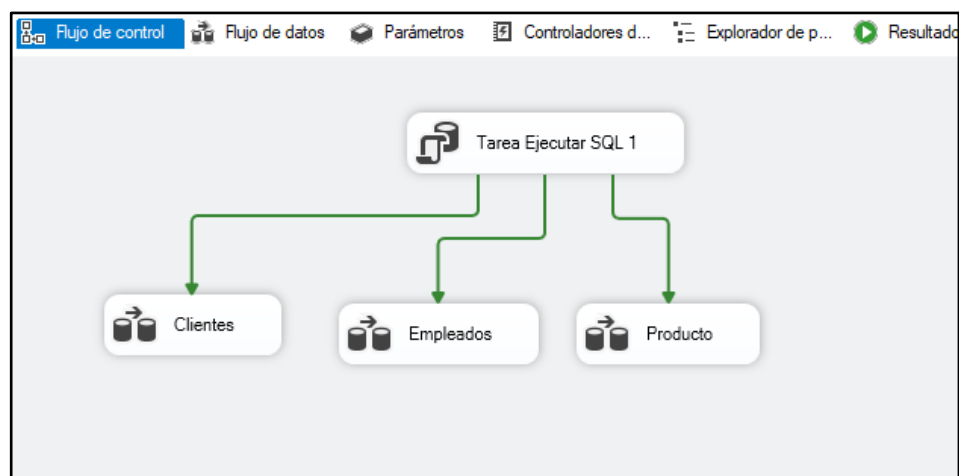


Figura 14. Flujo de control Producto.

2. En el flujo de datos de Datos (**Producto**) agregaremos un origen OLE DB

y un destino OLE DB.

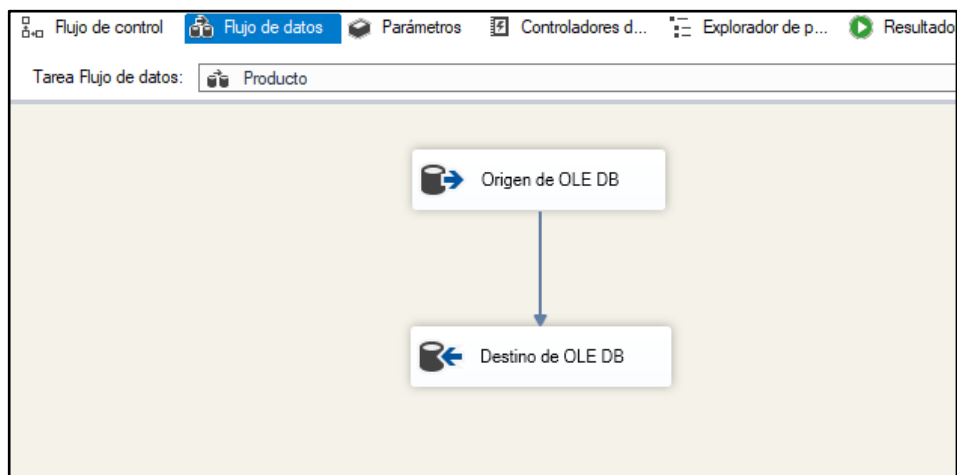


Figura 15. Flujo de datos Producto.

3. En el origen OLE DB usamos la conexión creada y usaremos el modo de acceso de comando SQL como origen.

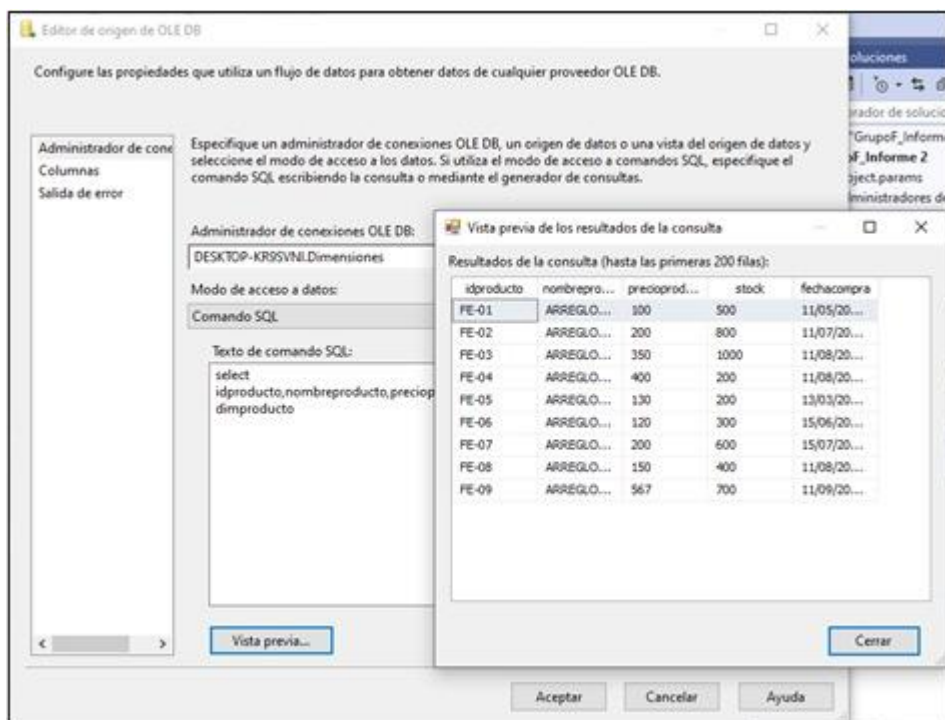


Figura 16. Conexión origen OLE DB Producto.

4. En el destino OLE DB igualmente usamos la conexión creada en SQL – **DimProducto** y la vamos a generar la relación directa.

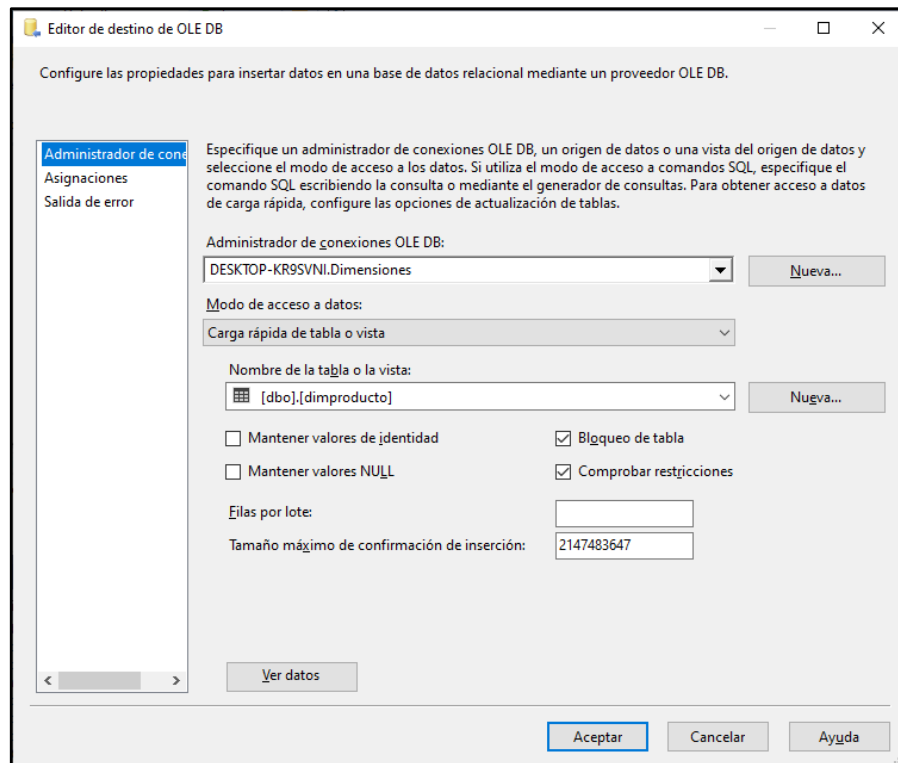


Figura 17. Conexión destino OLE DB Producto.

d) Dimensión Tiempo:

1. En el codificador del flujo de control agregamos una Tarea Ejecutar SQL, para limpiar la tabla y que no se repitan los datos, y una Tarea flujo de datos (**Tiempo**) donde podremos cargar los datos.

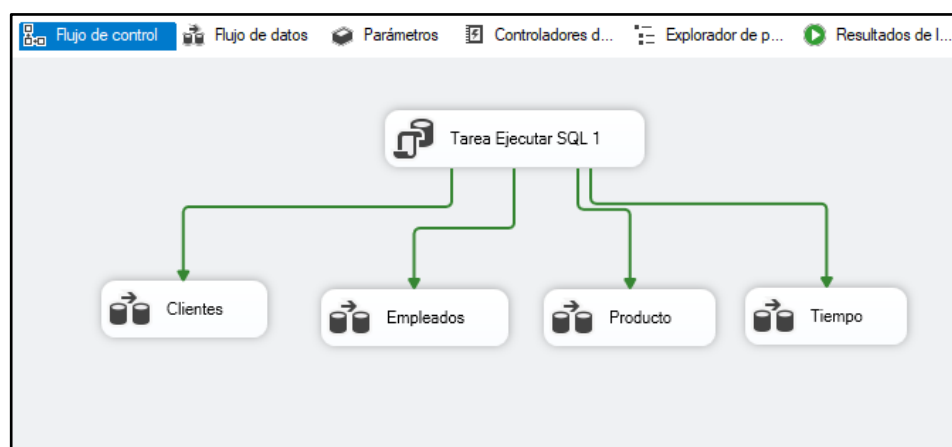


Figura 18. Flujo de control Tiempo.

2. En el flujo de datos de Datos (**Tiempo**) agregaremos un origen OLE DB, una columna derivada para los datos tiempo y un destino OLE DB.



Figura 19. Flujo de datos Tiempo.

3. En el origen OLE DB usamos la conexión creada y usaremos el modo de acceso de comando SQL como origen.

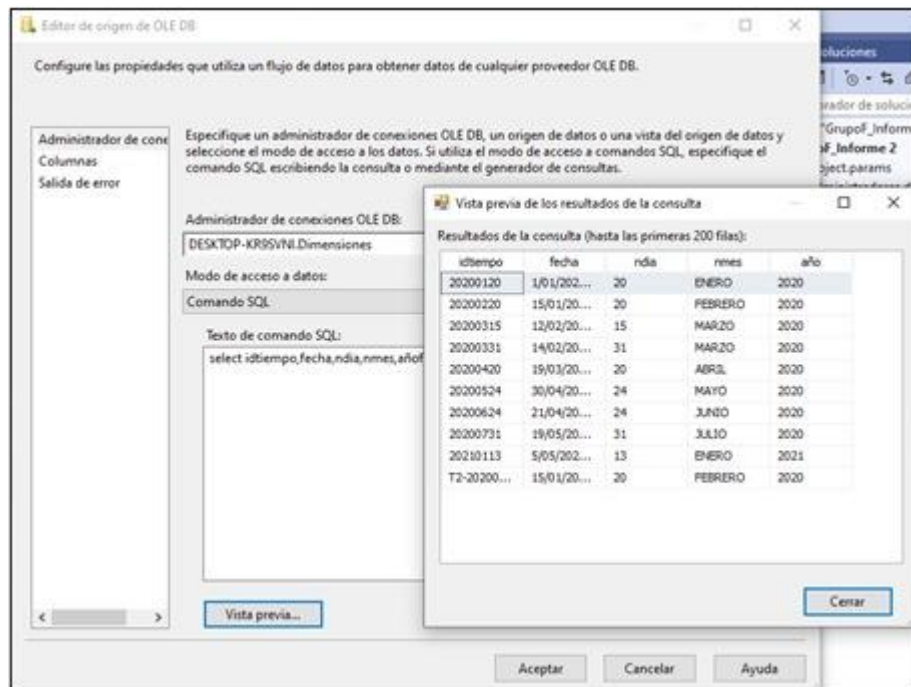


Figura 20. Conexión origen OLE DB Tiempo.

4. En la columna derivada que se agregó, se procede a separar la fecha en día, mes y año.

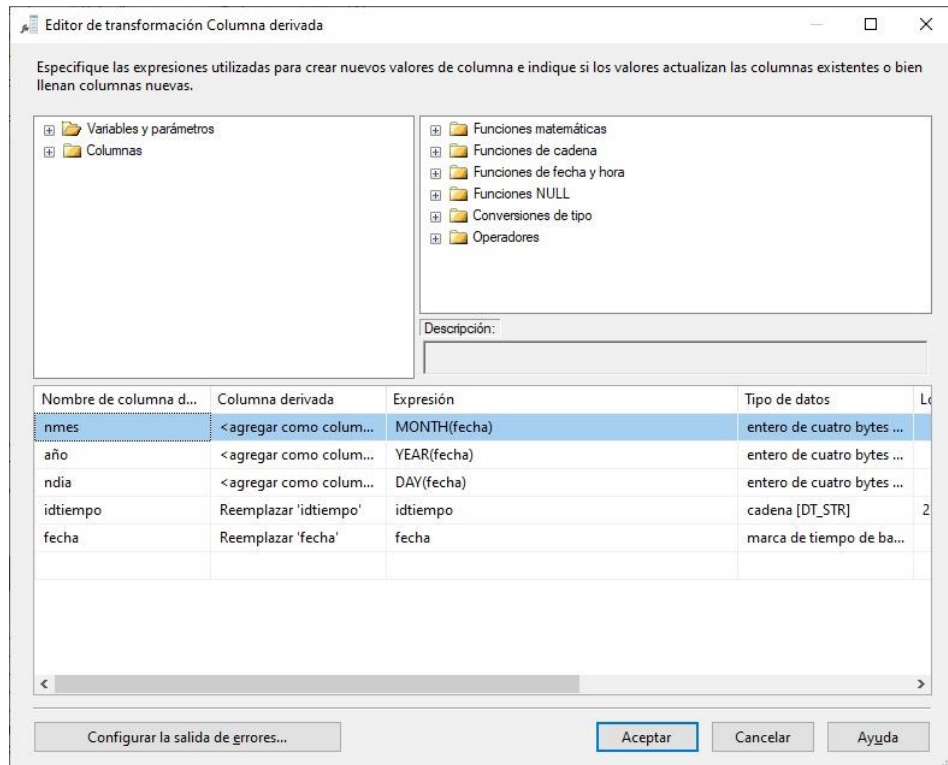


Figura 21. Columna derivada editada Tiempo.

- En el destino OLE DB igualmente usamos la conexión creada en SQL – DimTiempo y la vamos a generar la relación directa.

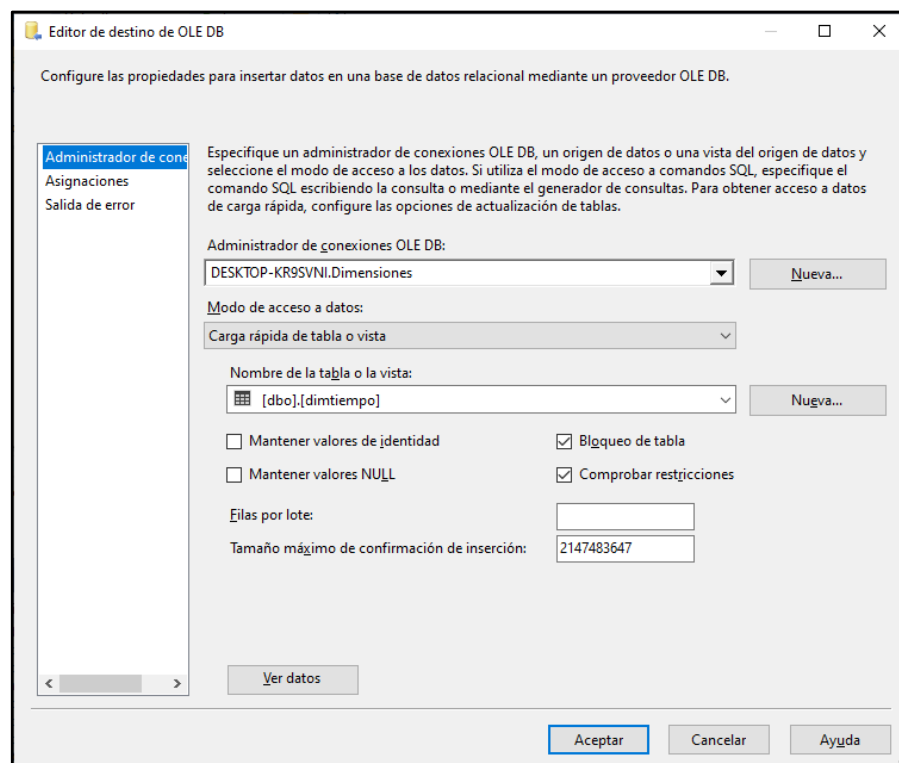


Figura 22. Conexión destino OLE DB Tiempo.

e) Dimensión Ventas:

1. En el codificador del flujo de control agregamos una Tarea Ejecutar SQL, para limpiar la tabla y que no se repitan los datos, y una Tarea flujo de datos (**Ventas**) donde podremos cargar los datos.

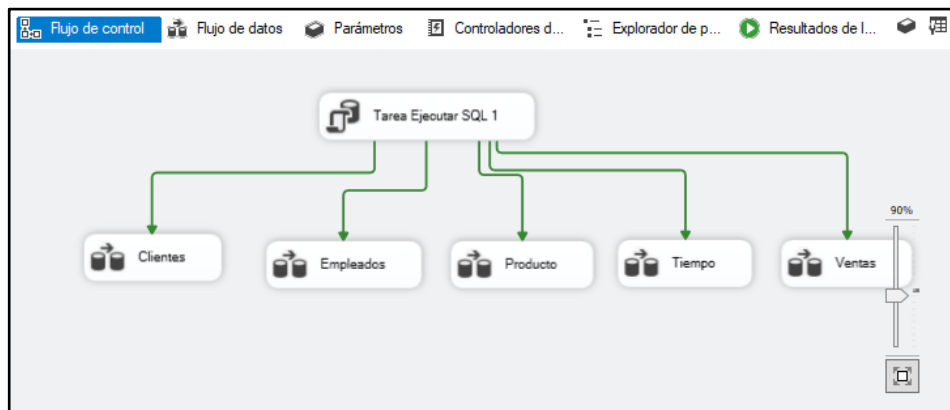


Figura 23. Flujo de control de Ventas

2. En el flujo de datos de Datos (**Ventas**) agregaremos un origen OLE DB y un destino OLE DB.

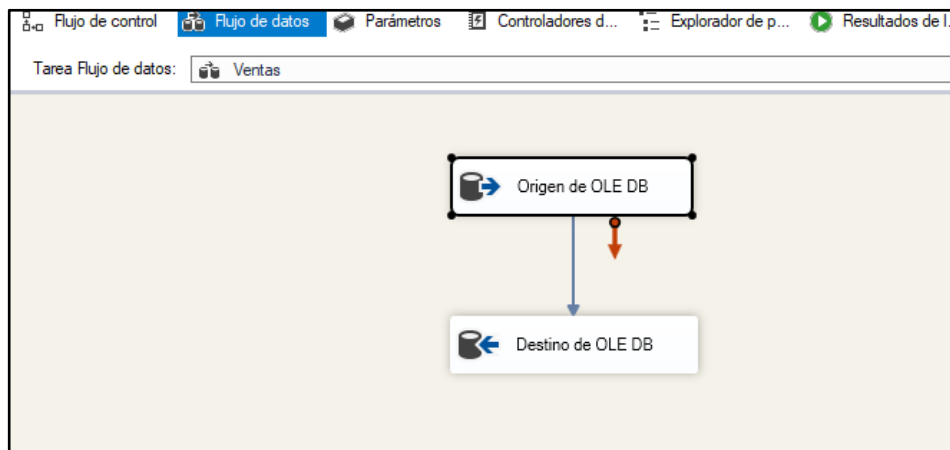


Figura 24. Flujo de datos Ventas

3. En el origen OLE DB usamos la conexión creada y usaremos el modo de acceso de comando SQL como origen.

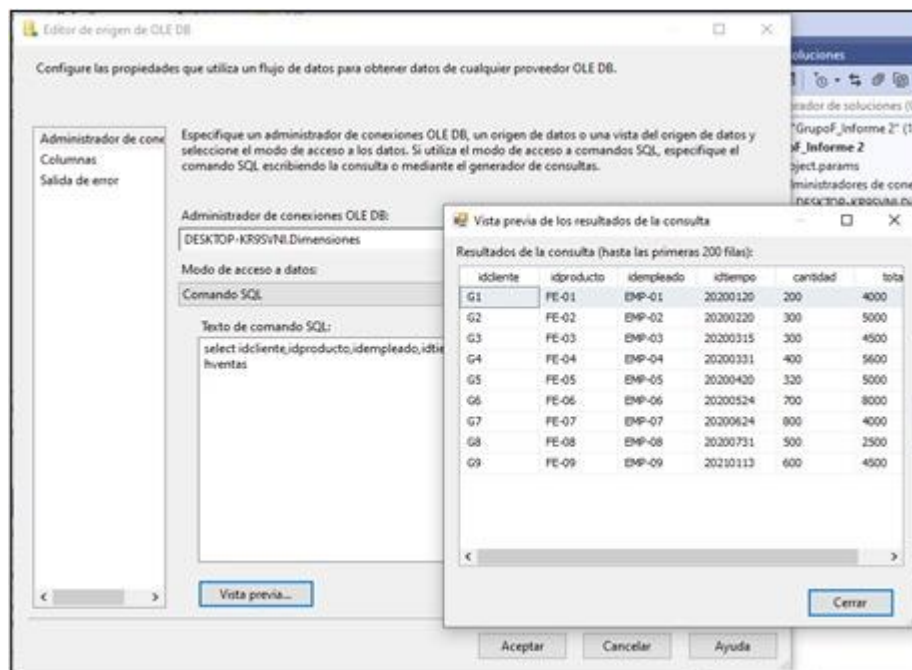


Figura 25. Conexión origen OLE DB Ventas.

- En el destino OLE DB igualmente usamos la conexión creada en SQL – **DimVentas** y la vamos a generar la relación directa.

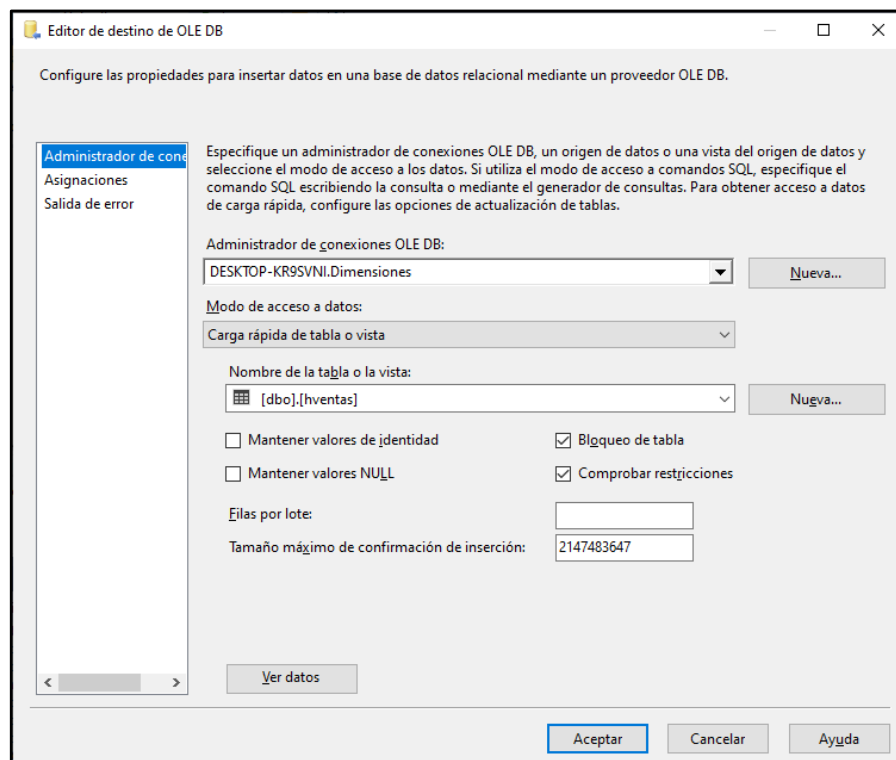


Figura 26. Conexión destino OLE DB Ventas.

5. Luego se procede a añadir un comando para evitar crear duplicación de data al ejecutar el flujo de control más de una vez.

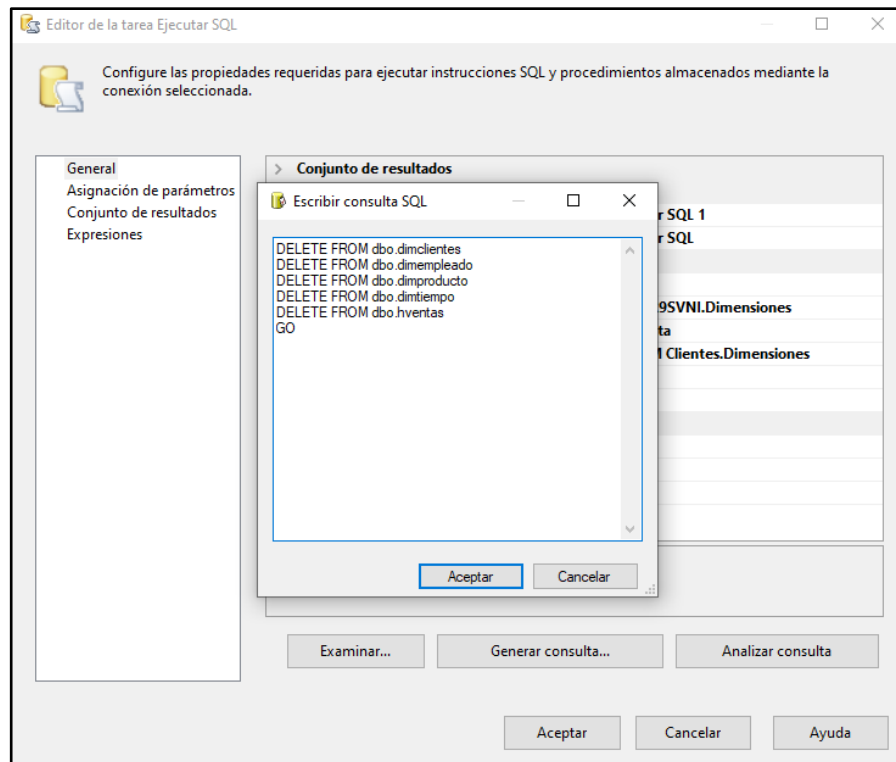


Figura 27. Flujo de control modificado.

3. VERIFICACIÓN

Luego se procede a ejecutar el flujo de control obteniendo la siguiente figura.



Figura 28. Verificación de las conexiones del proyecto.

Se procede a verificar la carga de datos en el SQL Server de la base de datos dimensional.


```

SELECT * FROM dbo.dimcliente
SELECT * FROM dbo.dimempleado
SELECT * FROM dbo.dimproducto
SELECT * FROM dbo.dimtiempo
SELECT * FROM dbo.hventas

```

Query executed successfully. DESKTOP-KR9SVNI (15.0 RTM) | DESKTOP-KR9SVNI\PC (67) | master | 00:00:00 | 54 rows

idcliente	nombrecliente	apellidoscliente	direccioncliente	districtocliente	paiscliente
G1	JOHN	RAMIREZ	AV. ANGAMOS 1786	SAN BORJA	PERU
G2	JUAN	ROBLES	AV. PERU 345	LINCE	PERU
G3	RAMIRO	RAMOS	AV. JOSE PARDO 140	MIRAFLORES	PERU
G4	ALLISON	PEREZ	OVALO OLIVAR 445	MIRAFLORES	PERU
G5	ANDREA	DOMINGUEZ	AV. BOLIVAR 1300	PUEBLO LIBRE	PERU
G6	ALVARO	SUAREZ	AV. PERU 120	SAN MARTIN	PERU
G7	MARTIN	ORDOÑEZ	PIEDRA LIZA 200	RIMAC	PERU
G8	JOHN	QUISPE	AV. ANGAMOS EST...	SAN BORJA	PERU

idempleado	nombreemple	apellidosemple	localidademple	puesto	telefonoparticular
EMP-01	PATRICIO	RAMIREZ	LINCE	OBTERO	945-357-679
EMP-02	RUBEN	ROBLES	LOS OLIVOS	OBTERO	949-566-787
EMP-03	ROBIN	RAMOS	JESUS MARIA	PRACT...	955-145-667
EMP-04	ALFRED	PEREZ	MIRAFLORES	OBTERO	965-554-778
EMP-05	ANDREA	DOMINGU...	ATE	TRANS...	984-556-333
EMP-06	ANTHONY	SUAREZ	SAN JUAN	TRANS...	901-456-225

idproducto	nombreproducto	precioproducto	stock	fechacompra
FE-01	ARREGLO SIMPLE	100.00	500	2020-05-11
FE-02	ARREGLO MATRIMONIAL	200.00	800	2020-07-11
FE-03	ARREGLO CEREMONIA	350.00	1000	2020-08-11
FE-04	ARREGLO FIEST	400.00	200	2020-08-11
FE-05	ARREGLO SORPRESA	130.00	200	2020-03-13
FE-06	ARREGLO EMPRESARIAL	120.00	300	2020-06-15
FE-07	ARREGLO DIA DE LA M...	200.00	600	2020-07-15
FE-08	ARREGLO DE COLORES	150.00	400	2020-08-11

idtiempo	fecha	ndia	nmes	año
20200120	2021-01-01 23:48:25.730	20	ENERO	2020
20200220	2021-01-15 23:48:25.730	20	FEBRERO	2020
20200315	2021-02-12 23:48:25.730	15	MARZO	2020
20200331	2021-02-14 23:48:25.730	31	MARZO	2020
20200420	2021-03-19 23:48:25.730	20	ABRIL	2020
20200524	2021-04-30 23:48:25.730	24	MAYO	2020
20200624	2021-04-21 23:48:25.730	24	JUNIO	2020
20200731	2021-05-19 23:48:25.730	31	JULIO	2020

idcliente	idproducto	idempleado	idtiempo	cantidad	total
G1	FE-01	EMP-01	20200120	200	4000
G2	FE-02	EMP-02	20200220	300	5000
G3	FE-03	EMP-03	20200315	300	4500
G4	FE-04	EMP-04	20200331	400	5600
G5	FE-05	EMP-05	20200420	320	5000
G6	FE-06	EMP-06	20200524	700	8000

Query executed successfully. DESKTOP-KR9SVNI (15.0 RTM) | DESKTOP-KR9SVNI\PC (67) | master | 00:00:00 | 54 rows

Figura 29. Verificación de la conexión en el SQL Server.

4. ENLACE:

<https://www.youtube.com/watch?v=O1uVXDyl0m0>