

## **DOCUMENTACIÓN DEL PROCESO ETL**

**Equipo:** La computadora cuántica

Este documento describe el proceso de Extracción, Transformación y Carga (ETL) que alimenta el Data Warehouse (DW) representado con la base de datos HotelDW a partir de la base OLTP HotelOLTP ambas almacenadas en un contenedor de SQL Server en Docker. El flujo se implementa en Visual Studio 2022 mediante proyectos Integration Services (SSIS).

### **Arquitectura General**

La solución ETL sigue una arquitectura modular de paquetes por entidad. Cada paquete consta de fases de Extracción, Transformación y Carga hacia tablas destino en el DW con el esquema dbo.Entidad.

### **Entidades Incluidas**

- **Cliente:** Permite segmentar clientes, analizar nacionalidades y género para campañas de marketing y lealtad.
- **Habitacion:** Soporta dashboards de ocupación, tarifa promedio y análisis de capacidad.
- **Reserva:** Base para métricas de booking window, duración de estadías y forecasting de demanda.
- **Pago:** Fuente principal de ingresos habilita análisis de revenue por método y fecha.
- **Empleado:** Facilita KPIs de desempeño y planeación de recursos humanos.
- **Tarea:** Mide productividad y eficiencia operativa del personal.
- **Mantenimiento:** Permite controlar costos, tiempos fuera de servicio y calidad de instalaciones.
- **Producto:** Catálogo necesario para análisis de costos unitarios y categorías de artículos.
- **Inventario:** Alimenta reportes de rotación de stock y previsiones de abastecimiento.
- **Encuesta:** Mide satisfacción del cliente y detecta áreas de mejora.
- **Visita:** Ayuda a estimar tráfico, estancia promedio y promociones efectivas.

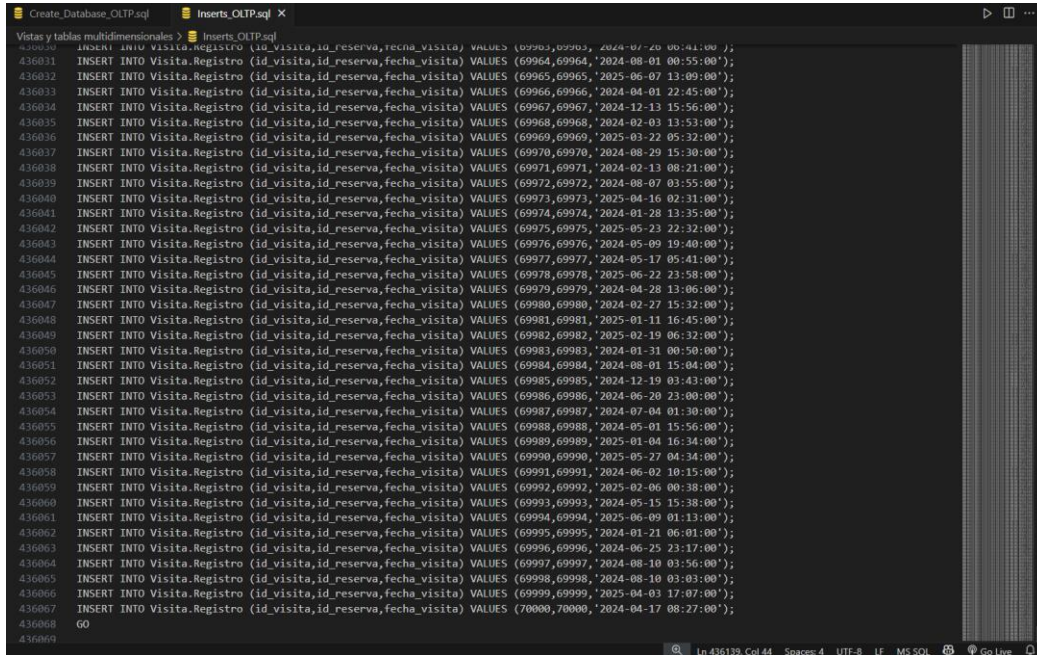
## Requisitos previos

- Visual Studio 2022 con extensión SQL Server Integration Services Projects.
- Docker con container de SQL Server 2022
- Azure Data Studio como IDE para SQL Server
- Usuario con permisos de lectura sobre HotelOLTP y escritura sobre HotelDW.
- Base de datos destino: HotelDW.
- Usuario con permisos de lectura en HotelOLTP y escritura en HotelDW.
- Base de datos origen: HotelOLTP (con la data cargada).

Para crear la base de datos ejecutamos el archivo Create\_Database\_OLTP.sql

```
Create_Database_OLTP.sql X
Vistas y tablas multidimensionales > Create_Database_OLTP.sql
1 CREATE DATABASE HotelOLTP;
2 USE HotelOLTP
3 GO
4
5 -- CREACION DE TODOS LOS SCHEMAS (INDISPENSABLE EJECUTAR UNO X UNO)
6 CREATE SCHEMA Cliente;
7 CREATE SCHEMA Habitacion;
8 CREATE SCHEMA Reserva;
9 CREATE SCHEMA Pago;
10 CREATE SCHEMA Empleado;
11 CREATE SCHEMA Tarea;
12 CREATE SCHEMA Mantenimiento;
13 CREATE SCHEMA Producto;
14 CREATE SCHEMA Inventario;
15 CREATE SCHEMA Encuesta;
16 CREATE SCHEMA Visita;
17
18 -- ESQUEMA: Cliente
19 CREATE TABLE Cliente.Nacionalidad (
20     id_nacionalidad INT PRIMARY KEY,
21     nombre NVARCHAR(50) NOT NULL UNIQUE
22 );
23
24 CREATE TABLE Cliente.Cliente (
25     id_cliente INT PRIMARY KEY,
26     nombre NVARCHAR(100) NOT NULL,
27     apellido NVARCHAR(100) NOT NULL,
28     email NVARCHAR(150) UNIQUE,
29     fecha_nacimiento DATE,
30     genero CHAR(1) CHECK (genero IN ('M', 'F', 'O')),
31     id_nacionalidad INT NOT NULL,
32     fecha_registro DATETIME DEFAULT GETDATE(),
33
34     CONSTRAINT fk_cliente_nacionalidad FOREIGN KEY (id_nacionalidad)
35     REFERENCES Cliente.Nacionalidad(id_nacionalidad)
36 );
37
```

Una vez creado ejecutamos el Script con las inserciones del dataset el cual es bastante grande y puede llegar a causar un poco de lag ya que tiene más de 400,000 inserts los cuales ejecutamos de un solo y ya tenemos la base de datos completa OLTP.



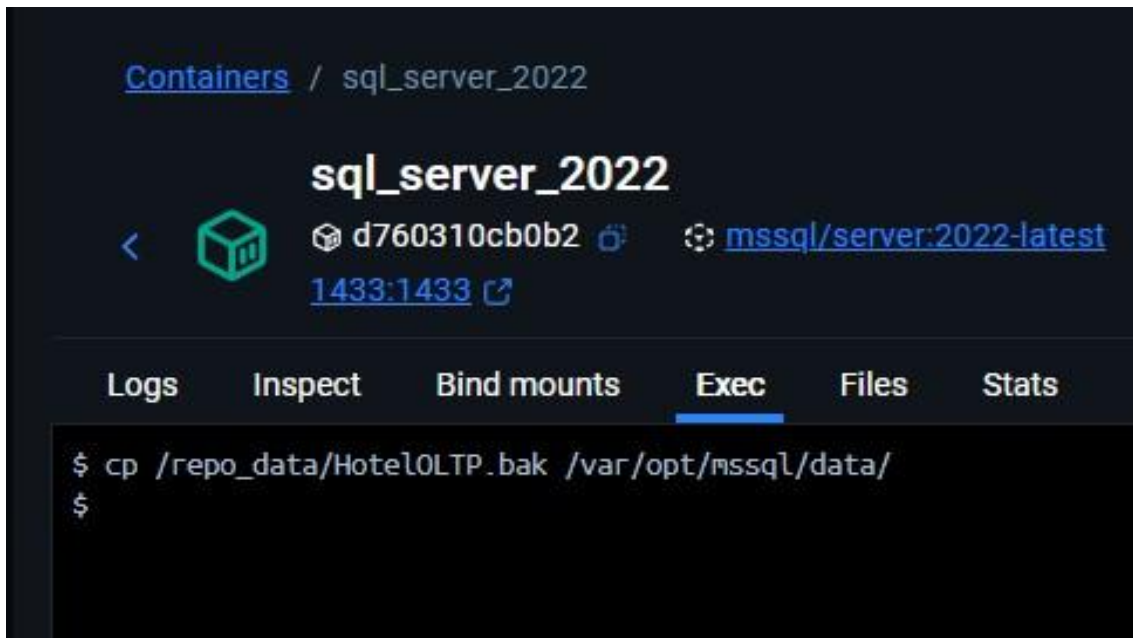
```
436029 INSERT INTO Visita.Registro (id_visita,id_reserva,fecha_visita) VALUES (69963,69963,'2024-07-20 00:41:00');
436030 INSERT INTO Visita.Registro (id_visita,id_reserva,fecha_visita) VALUES (69964,69964,'2024-08-01 00:55:00');
436031 INSERT INTO Visita.Registro (id_visita,id_reserva,fecha_visita) VALUES (69965,69965,'2025-06-07 13:09:00');
436032 INSERT INTO Visita.Registro (id_visita,id_reserva,fecha_visita) VALUES (69966,69966,'2024-04-01 22:45:00');
436033 INSERT INTO Visita.Registro (id_visita,id_reserva,fecha_visita) VALUES (69967,69967,'2024-12-13 15:56:00');
436034 INSERT INTO Visita.Registro (id_visita,id_reserva,fecha_visita) VALUES (69968,69968,'2024-02-03 13:53:00');
436035 INSERT INTO Visita.Registro (id_visita,id_reserva,fecha_visita) VALUES (69969,69969,'2025-03-22 05:32:00');
436036 INSERT INTO Visita.Registro (id_visita,id_reserva,fecha_visita) VALUES (69970,69970,'2024-08-29 15:30:00');
436037 INSERT INTO Visita.Registro (id_visita,id_reserva,fecha_visita) VALUES (69971,69971,'2024-02-13 08:21:00');
436038 INSERT INTO Visita.Registro (id_visita,id_reserva,fecha_visita) VALUES (69972,69972,'2024-08-07 03:55:00');
436039 INSERT INTO Visita.Registro (id_visita,id_reserva,fecha_visita) VALUES (69973,69973,'2025-04-16 02:31:00');
436040 INSERT INTO Visita.Registro (id_visita,id_reserva,fecha_visita) VALUES (69974,69974,'2024-01-28 13:35:00');
436041 INSERT INTO Visita.Registro (id_visita,id_reserva,fecha_visita) VALUES (69975,69975,'2025-05-23 22:32:00');
436042 INSERT INTO Visita.Registro (id_visita,id_reserva,fecha_visita) VALUES (69976,69976,'2024-05-09 19:40:00');
436043 INSERT INTO Visita.Registro (id_visita,id_reserva,fecha_visita) VALUES (69977,69977,'2024-05-17 05:41:00');
436044 INSERT INTO Visita.Registro (id_visita,id_reserva,fecha_visita) VALUES (69978,69978,'2025-06-22 23:58:00');
436045 INSERT INTO Visita.Registro (id_visita,id_reserva,fecha_visita) VALUES (69979,69979,'2024-04-28 13:06:00');
436046 INSERT INTO Visita.Registro (id_visita,id_reserva,fecha_visita) VALUES (69980,69980,'2024-02-27 15:32:00');
436047 INSERT INTO Visita.Registro (id_visita,id_reserva,fecha_visita) VALUES (69981,69981,'2025-01-11 16:45:00');
436048 INSERT INTO Visita.Registro (id_visita,id_reserva,fecha_visita) VALUES (69982,69982,'2025-02-19 06:32:00');
436049 INSERT INTO Visita.Registro (id_visita,id_reserva,fecha_visita) VALUES (69983,69983,'2024-01-31 00:50:00');
436050 INSERT INTO Visita.Registro (id_visita,id_reserva,fecha_visita) VALUES (69984,69984,'2024-08-01 15:04:00');
436051 INSERT INTO Visita.Registro (id_visita,id_reserva,fecha_visita) VALUES (69985,69985,'2024-12-19 03:43:00');
436052 INSERT INTO Visita.Registro (id_visita,id_reserva,fecha_visita) VALUES (69986,69986,'2024-06-20 23:00:00');
436053 INSERT INTO Visita.Registro (id_visita,id_reserva,fecha_visita) VALUES (69987,69987,'2024-07-04 01:30:00');
436054 INSERT INTO Visita.Registro (id_visita,id_reserva,fecha_visita) VALUES (69988,69988,'2024-05-01 15:56:00');
436055 INSERT INTO Visita.Registro (id_visita,id_reserva,fecha_visita) VALUES (69989,69989,'2025-01-04 16:34:00');
436056 INSERT INTO Visita.Registro (id_visita,id_reserva,fecha_visita) VALUES (69990,69990,'2025-05-27 04:34:00');
436057 INSERT INTO Visita.Registro (id_visita,id_reserva,fecha_visita) VALUES (69991,69991,'2024-06-02 10:15:00');
436058 INSERT INTO Visita.Registro (id_visita,id_reserva,fecha_visita) VALUES (69992,69992,'2025-02-06 00:38:00');
436059 INSERT INTO Visita.Registro (id_visita,id_reserva,fecha_visita) VALUES (69993,69993,'2024-05-15 15:38:00');
436060 INSERT INTO Visita.Registro (id_visita,id_reserva,fecha_visita) VALUES (69994,69994,'2025-06-09 01:13:00');
436061 INSERT INTO Visita.Registro (id_visita,id_reserva,fecha_visita) VALUES (69995,69995,'2024-01-21 06:01:00');
436062 INSERT INTO Visita.Registro (id_visita,id_reserva,fecha_visita) VALUES (69996,69996,'2024-06-25 23:17:00');
436063 INSERT INTO Visita.Registro (id_visita,id_reserva,fecha_visita) VALUES (69997,69997,'2024-08-10 03:56:00');
436064 INSERT INTO Visita.Registro (id_visita,id_reserva,fecha_visita) VALUES (69998,69998,'2024-08-10 03:03:00');
436065 INSERT INTO Visita.Registro (id_visita,id_reserva,fecha_visita) VALUES (69999,69999,'2025-04-03 17:07:00');
436066 INSERT INTO Visita.Registro (id_visita,id_reserva,fecha_visita) VALUES (70000,70000,'2024-04-17 08:27:00');
436067 GO
436068
436069
```

Y en consola iran apareciendo el numero de registros ingresado en cada una de las tablas puede tardar unos minutos por eso se creo el archivo HotelOLTP.bak para contar con el respaldo de esa base de datos y poder restaurarla de forma rápida y sin tantas complicaciones que puede causar ejecutar tantos registros.



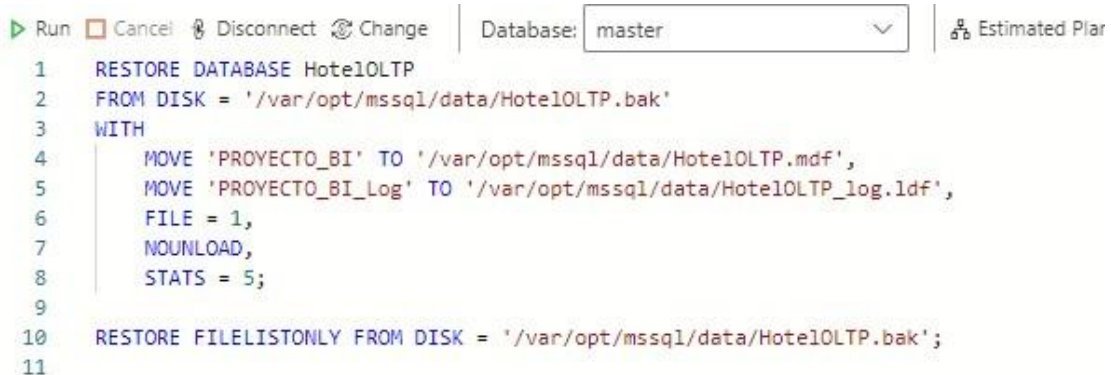
## Creación de las bases de datos

Para la creación de la base de datos HotelOLTP aplicamos el proceso utilizado para la importación de las bases de datos AdventureWorks importando el archivo de backup .bak dentro de la carpeta C:\sqlserver\_data posteriormente ejecutando el siguiente comando en la consola (Exec) de Docker cp /repo\_data/HotelOLTP.bak /var/opt/mssql/data/



The screenshot shows the Docker Desktop interface for a container named 'sql\_server\_2022'. The 'Exec' tab is selected, and the terminal shows the command 'cp /repo\_data/HotelOLTP.bak /var/opt/mssql/data/' being executed successfully.

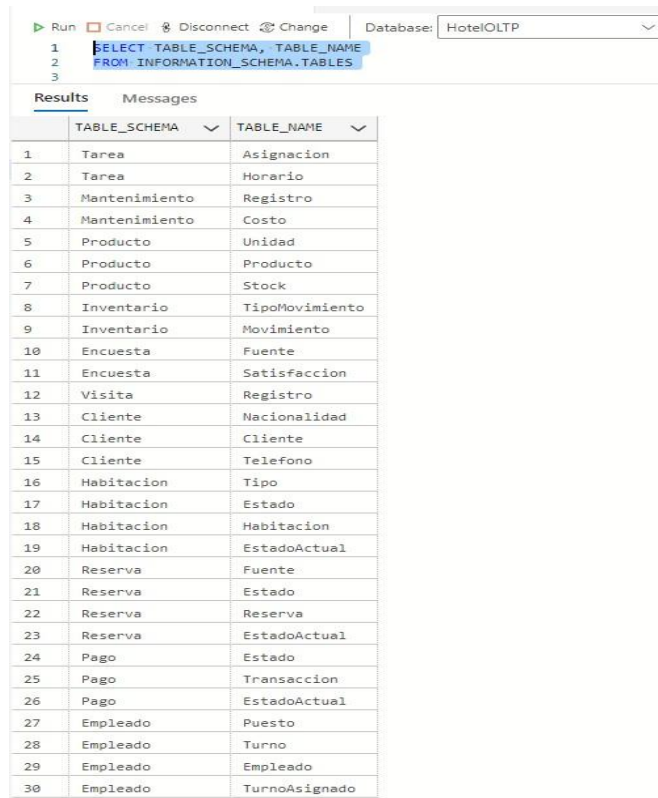
Posteriormente se restaura la base de datos desde Azure Data Studio con eso podemos restaurar la base de datos OLTP en una computadora diferente a donde se creo sin complicarnos tanto con el proceso.



The screenshot shows the Azure Data Studio interface with a SQL query entered in the editor. The query is a RESTORE DATABASE command for a database named 'HotelOLTP'. The 'Run' button is visible at the top left of the editor.

```
1 RESTORE DATABASE HotelOLTP
2 FROM DISK = '/var/opt/mssql/data/HotelOLTP.bak'
3 WITH
4     MOVE 'PROYECTO_BI' TO '/var/opt/mssql/data/HotelOLTP.mdf',
5     MOVE 'PROYECTO_BI_Log' TO '/var/opt/mssql/data/HotelOLTP_log.ldf',
6     FILE = 1,
7     NOUNLOAD,
8     STATS = 5;
9
10 RESTORE FILELISTONLY FROM DISK = '/var/opt/mssql/data/HotelOLTP.bak';
11
```

Podemos corroborar que todas las tablas fueron creadas seleccionando la base de datos y ejecutando el siguiente comando.



The screenshot shows the SQL Server Enterprise Manager interface. The 'Database' dropdown is set to 'HotelOLTP'. The query editor contains the following SQL command:

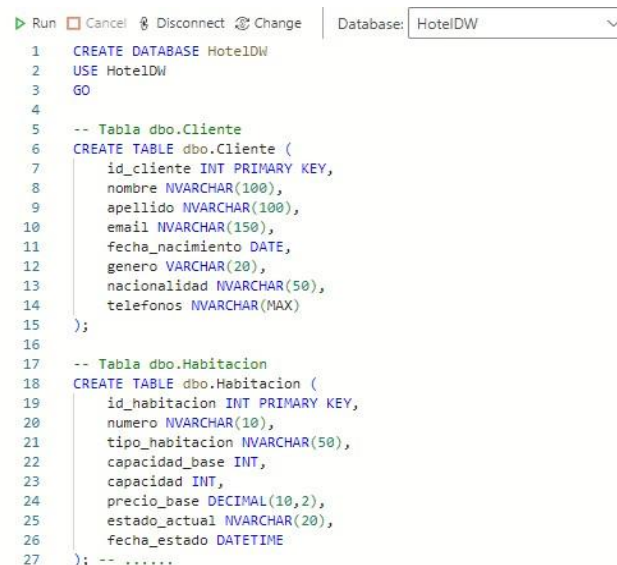
```
1 SELECT TABLE_SCHEMA, TABLE_NAME
2 FROM INFORMATION_SCHEMA.TABLES
3
```

The 'Results' tab is active, displaying a table with 30 rows and 2 columns: 'TABLE\_SCHEMA' and 'TABLE\_NAME'. The data is as follows:

	TABLE_SCHEMA	TABLE_NAME
1	Tarea	Asignacion
2	Tarea	Horario
3	Mantenimiento	Registro
4	Mantenimiento	Costo
5	Producto	Unidad
6	Producto	Producto
7	Producto	Stock
8	Inventario	TipoMovimiento
9	Inventario	Movimiento
10	Encuesta	Fuente
11	Encuesta	Satisfaccion
12	Visita	Registro
13	Cliente	Nacionalidad
14	Cliente	Cliente
15	Cliente	Telefono
16	Habitacion	Tipo
17	Habitacion	Estado
18	Habitacion	Habitacion
19	Habitacion	EstadoActual
20	Reserva	Fuente
21	Reserva	Estado
22	Reserva	Reserva
23	Reserva	EstadoActual
24	Pago	Estado
25	Pago	Transaccion
26	Pago	EstadoActual
27	Empleado	Puesto
28	Empleado	Turno
29	Empleado	Empleado
30	Empleado	TurnoAsignado

## Creación de la base de datos DW

Vamos a crear las tablas dbo para guardar la información procedente de la base de datos OLTP ejecutamos el siguiente script tras crear la base de datos



The screenshot shows the SQL Server Enterprise Manager interface. The 'Database' dropdown is set to 'HotelDW'. The query editor contains the following SQL script:

```
1 CREATE DATABASE HotelDW
2 USE HotelDW
3 GO
4
5 -- Tabla dbo.Cliente
6 CREATE TABLE dbo.Cliente (
7     id_cliente INT PRIMARY KEY,
8     nombre NVARCHAR(100),
9     apellido NVARCHAR(100),
10    email NVARCHAR(150),
11    fecha_nacimiento DATE,
12    genero VARCHAR(20),
13    nacionalidad NVARCHAR(50),
14    telefonos NVARCHAR(MAX)
15 );
16
17 -- Tabla dbo.Habitacion
18 CREATE TABLE dbo.Habitacion (
19     id_habitacion INT PRIMARY KEY,
20     numero NVARCHAR(10),
21     tipo_habitacion NVARCHAR(50),
22     capacidad_base INT,
23     capacidad INT,
24     precio_base DECIMAL(10,2),
25     estado_actual NVARCHAR(20),
26     fecha_estado DATETIME
27 ); -- .....
```

Ahora debemos crear el usuario que utilizaremos para realizar el ETL con permisos de lectura en la base de datos OLTP y con permisos de escritura para la base de datos Data Warehouse.

```
-- CREACION DE USUARIO PARA DASHBOARDS
CREATE LOGIN etl_user WITH PASSWORD = '*****';

USE HotelOLTP;
GO

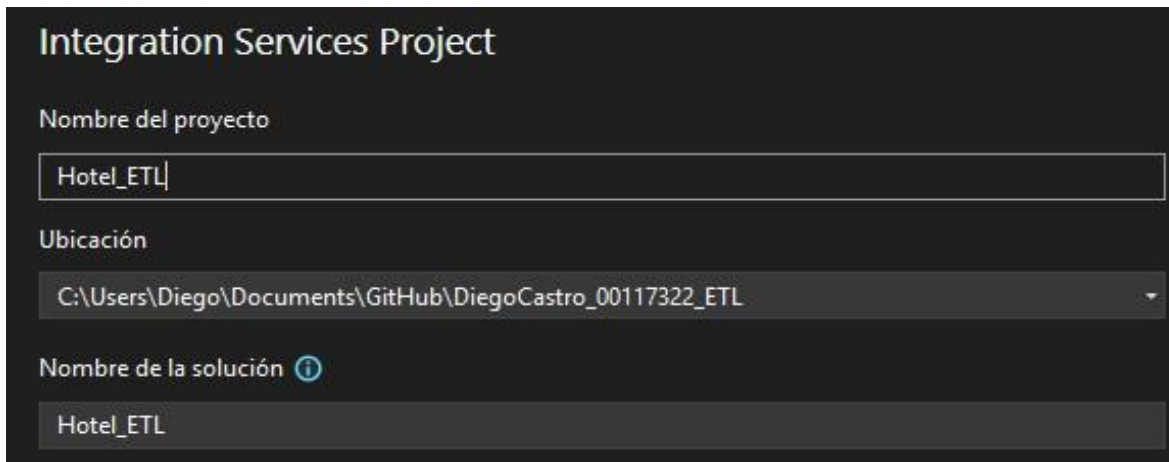
CREATE USER etl_user FOR LOGIN etl_user;
EXEC sp_addrolemember 'db_datareader', 'etl_user'; -- Permiso de lectura

USE HotelDW;
GO

CREATE USER etl_user FOR LOGIN etl_user;
EXEC sp_addrolemember 'db_datareader', 'etl_user'; -- Lectura
EXEC sp_addrolemember 'db_datawriter', 'etl_user'; -- Escritura
```

## Proceso de creación de la Solución SSIS

- 1- Crear el proyecto SSIS: •Abrir Visual Studio 2022 y crear un nuevo proyecto tipo Integration Services Project y nombrado como "Hotel\_ETL".



**Integration Services Project**

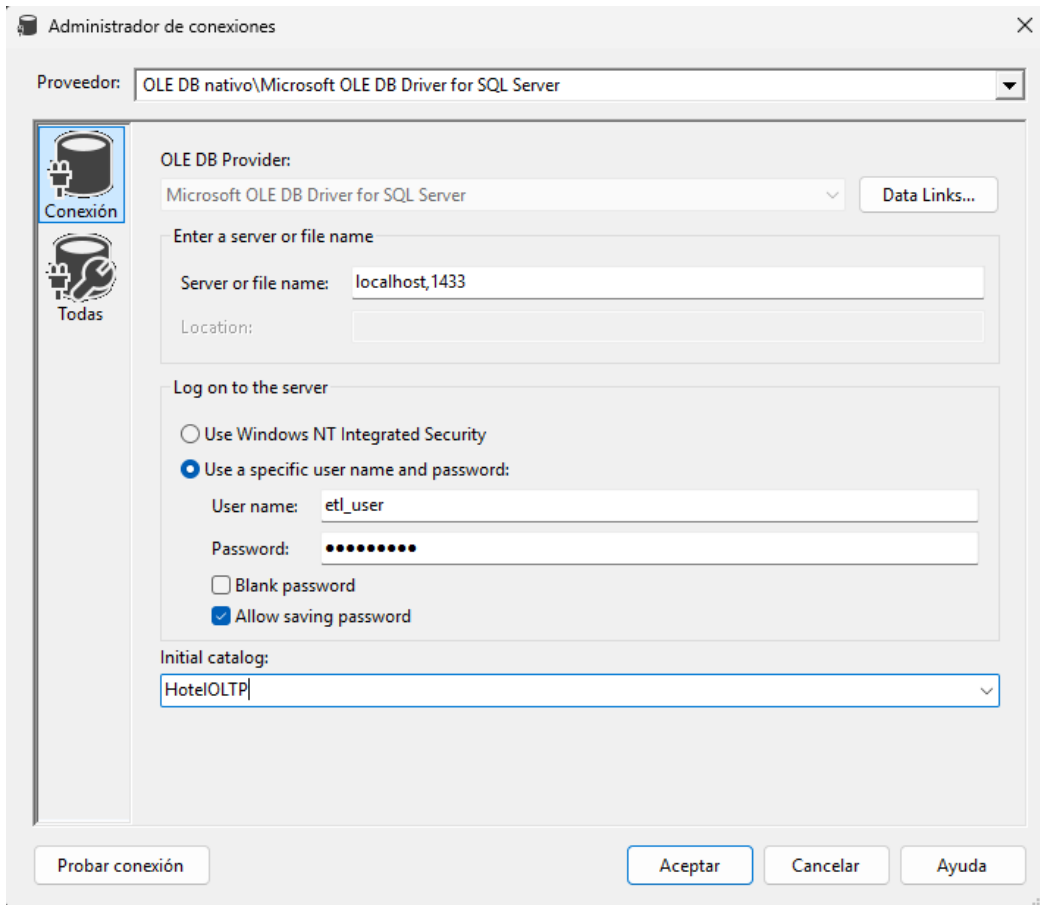
Nombre del proyecto  
Hotel\_ETL

Ubicación  
C:\Users\Diego\Documents\GitHub\DiegoCastro\_00117322\_ETL

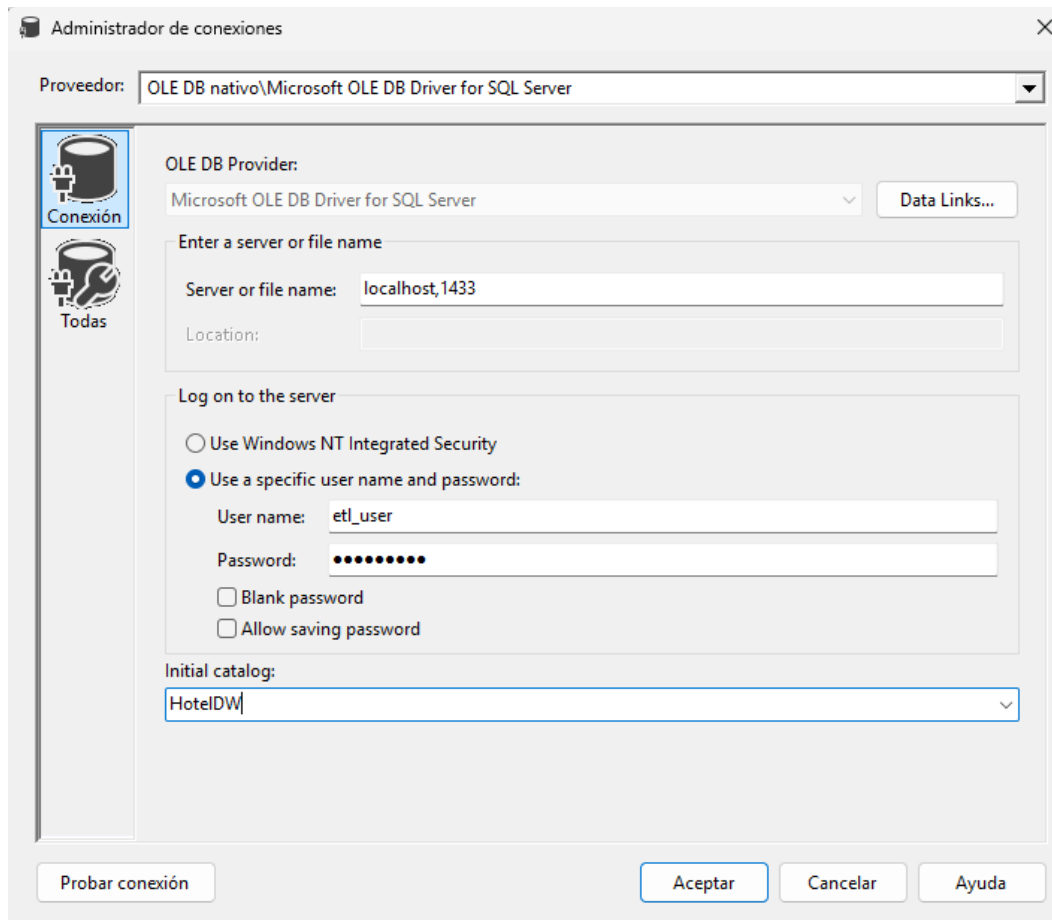
Nombre de la solución ⓘ  
Hotel\_ETL

2- Configurar conexiones: En la pestaña Connection Managers agregar dos conexiones tipo OLE DB.

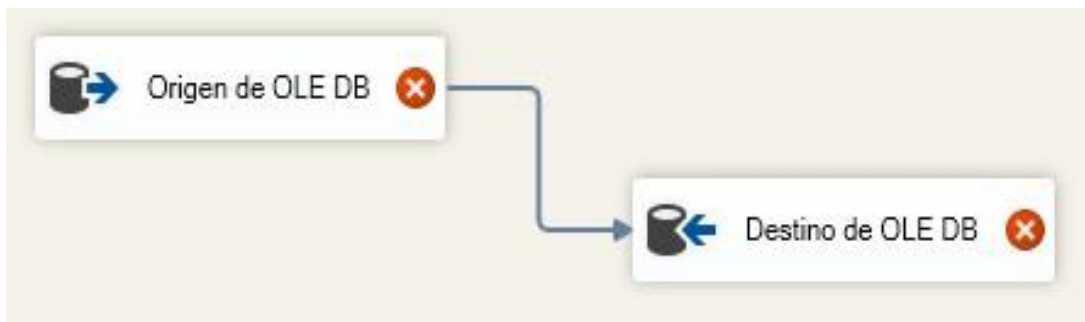
- Una conexión apuntando al origen de datos HotelOLTP.



- Una conexión apuntando al destino de datos HotelDW.



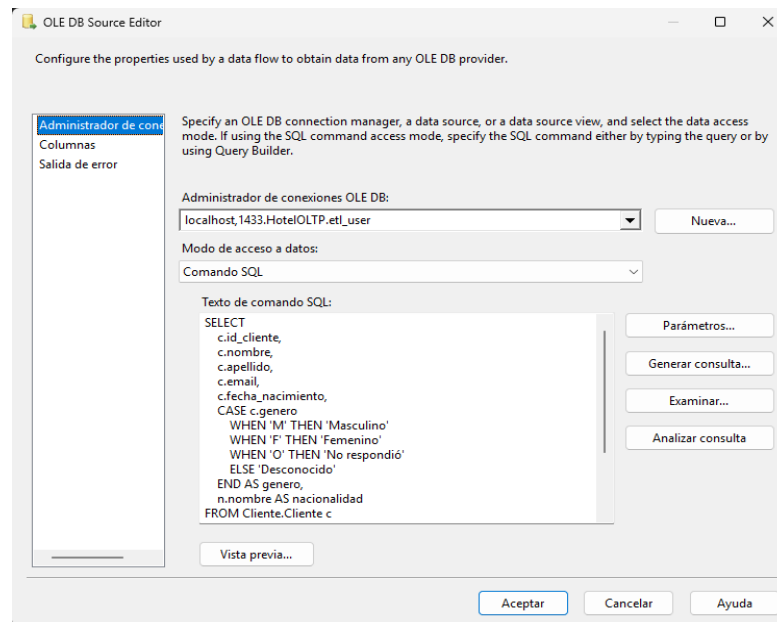
3. Conexión entre el origen y el destino: Debemos conectar el origen con el destino para que desaparezca el error.



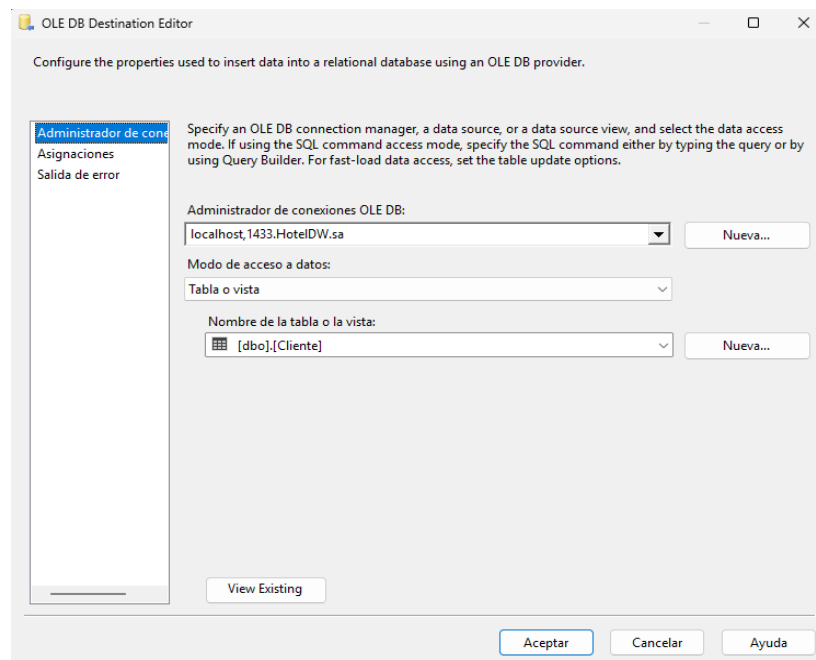
4. Configuración del ETL: Ahora tenemos que indicar que data es la que vamos a enviar desde el origen utilizando la consulta SQL.



- OLE DB Source: Ejecutar la consulta SQL para obtener datos desde HotelOLTP en este caso información de los clientes.

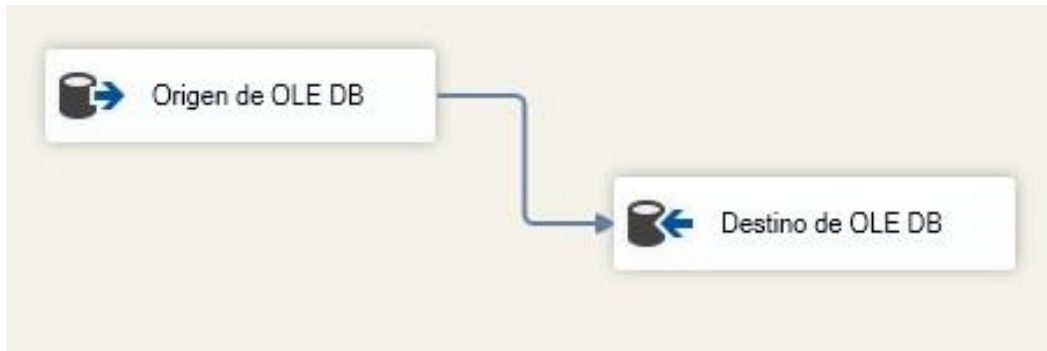


- OLE DB Destination: Establecer la conexión al DW y seleccionar la tabla destino dbo.Cliente donde se almacenara el resultado de la consulta.



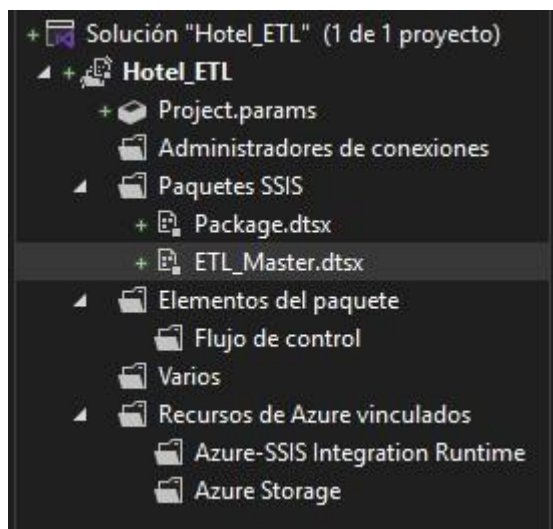
## 5. Validar los paquetes:

- Ahora ya podemos ejecutar el proyecto de SSIS para realizar el proceso del ETL y comprobar si es exitoso, pero como manejamos diferentes consultas para generar diferentes tablas realizaremos el siguiente paso para controlar varios subprocesos a la vez.

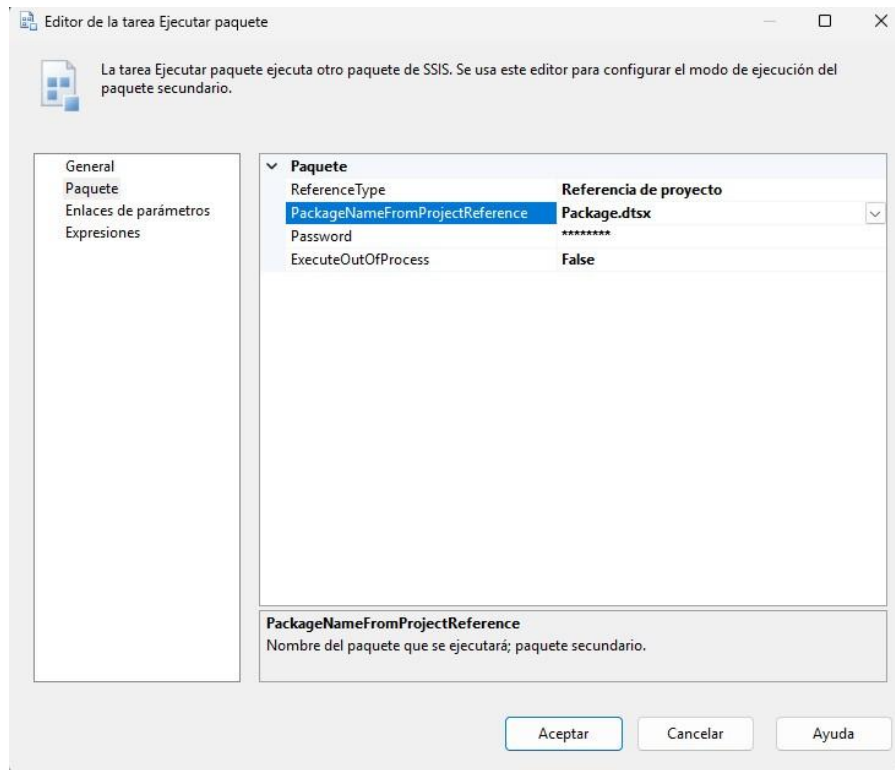


## 6. Crear Paquete Maestro:

- Agregar un paquete maestro (Master\_ETL.dtsx).

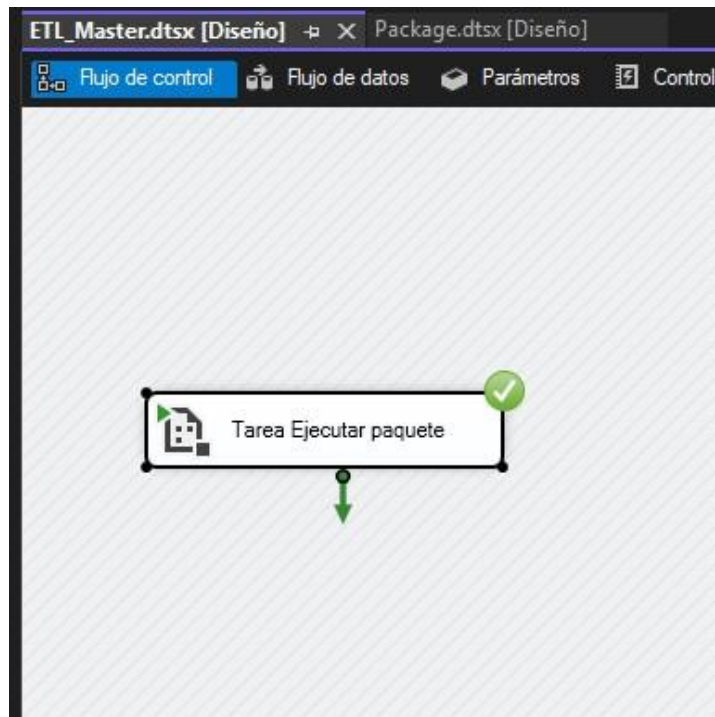


- Utilizar Execute Package Task para ejecutar secuencialmente cada paquete ETL creado anteriormente.

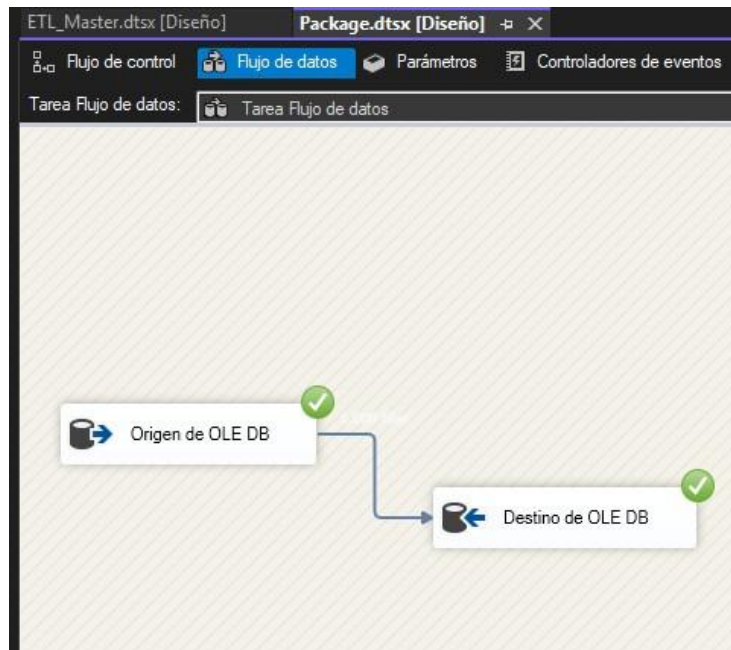


## 7. Ejecución Completa del proceso ETL:

- Ejecutar el paquete maestro desde Visual Studio.

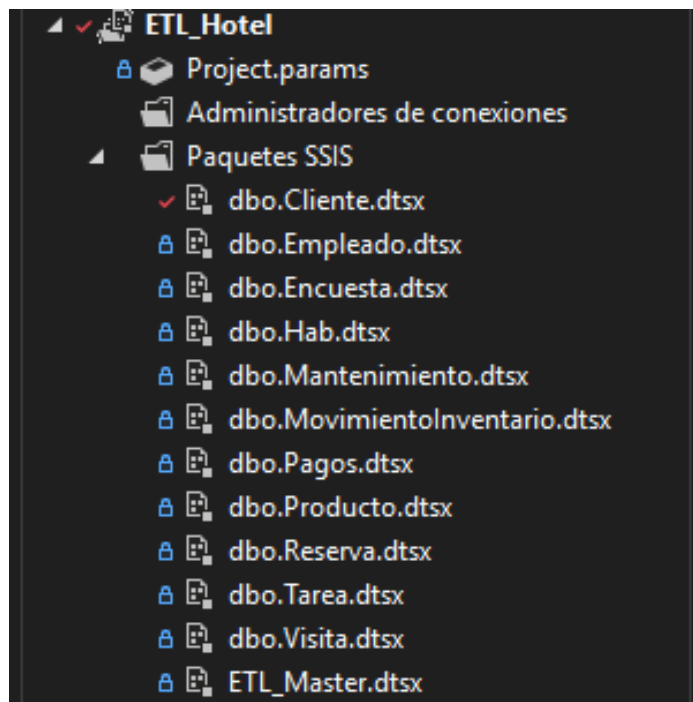


- Lo que ejecuta automáticamente el proceso de ETL creado anteriormente



#### 8. Creación de paquetes para todas las tablas

Ahora vamos a crear un nuevo paquete y vamos a renombrar con el estándar dbo.vista para cada una de las consultas del ETL.



Cada uno de los dbo deben contar con la misma estructura que ocupamos anteriormente en dbo.Cliente pero cambiando la consulta y la tabla donde se guardara la información

Por ejemplo, para dbo.hab tenemos la siguiente consulta en origen

OLE DB Source Editor

Configure the properties used by a data flow to obtain data from any OLE DB provider.

Administrador de conexiones OLE DB: localhost,1433.HotelOLTP.etl\_user Nueva...

Modo de acceso a datos: Comando SQL

Texto de comando SQL:

```
SELECT
  h.id_habitacion,
  h.numero,
  t.nombre AS tipo_habitacion,
  t.capacidad_base,
  h.capacidad,
  h.precio_base,
  e.nombre AS estado_actual,
  ea.fecha_actualizacion AS fecha_estado
FROM Habitacion.Habitacion h
INNER JOIN Habitacion.Tipo t ON h.id_tipo = t.id_tipo
LEFT JOIN Habitacion.EstadoActual ea ON h.id_habitacion =
  ea.id_habitacion
LEFT JOIN Habitacion.Estado e ON ea.id_estado = e.id_estado;
```

Parámetros...  
Generar consulta...  
Examinar...  
Analizar consulta

Vista previa...

Aceptar Cancelar Ayuda

Y en destino seleccionamos la tabla dbo.habitacion

OLE DB Destination Editor

Configure the properties used to insert data into a relational database using an OLE DB provider.

Administrador de conexiones OLE DB: localhost,1433.HotelDW.etl\_user Nueva...

Modo de acceso a datos: Carga rápida de tabla o vista

Nombre de la tabla o la vista: [dbo].[Habitacion] Nueva...

☐ Mantener valores de identidad ☒ Bloqueo de tabla  
☐ Mantener valores NULL ☒ Comprobar restricciones

Filas por lote: 1000

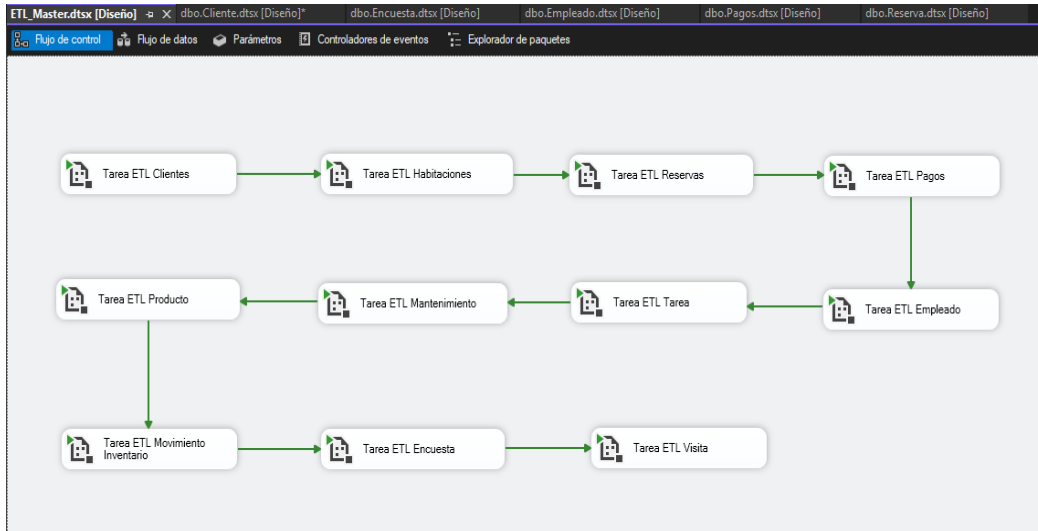
Tamaño máximo de confirmación de inserción: 2147483647

View Existing

Aceptar Cancelar Ayuda

Y hacemos el mismo proceso para el resto de vistas del ETL.

## 9. Unión de ejecuciones para cada paquete



Conectando cada una de las tareas de ejecución con su respectivo paquete.

Editor de la tarea Ejecutar paquete

La tarea Ejecutar paquete ejecuta otro paquete de SSIS. Se usa este editor para configurar el modo de ejecución del paquete secundario.

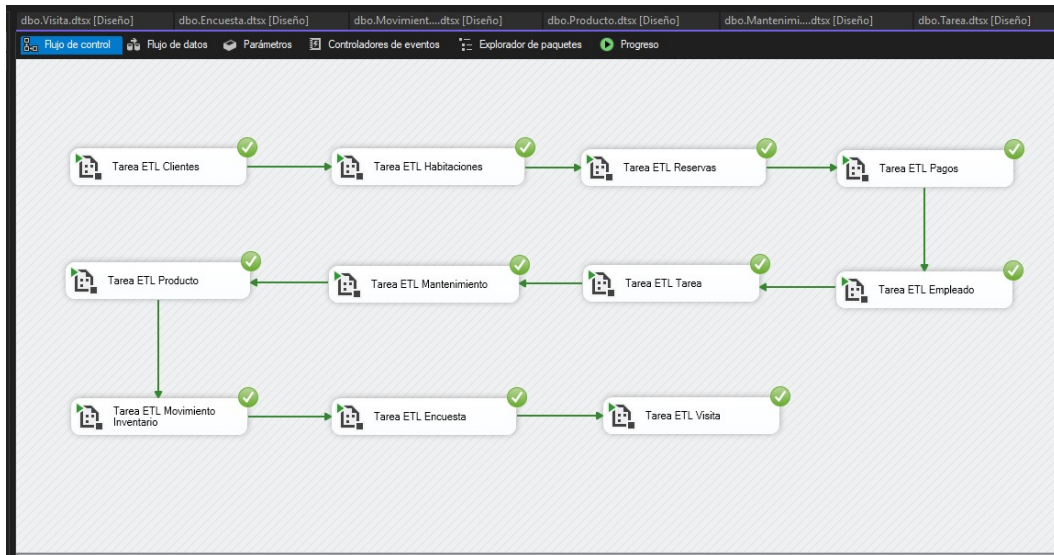
General  
**Paquete**  
Enlaces de parámetros  
Expresiones

Paquete	
ReferenceType	Referencia de proyecto
PackageNameFromProjectReference	dbo.Hab.dtsx
Password	*****
ExecuteOutOfProcess	False

**ReferenceType**  
Seleccione Referencia de proyecto para los paquetes secundarios de este proyecto. Seleccione Referencia externa para los paquetes secundarios ubicados fuera del proyecto (en el sistema ...)

Aceptar Cancelar Ayuda

Ahora al ejecutar el programa automáticamente se va a ir realizando el ETL para cada una de las tablas dbo una tras de otra hasta finalizar todo el proceso.



## 10. Comprobación de la transferencia de datos en Azure Data Studio:

- Realizar una consulta simple para comprobar la carga exitosa de los datos desde HotelDW por ejemplo para `select * from dbo.cliente` obtenemos la siguiente vista.

	id_cliente	nombre	apellido	email	fecha_nacimiento	genero	nacionalidad
1	1	Marisol	Ríos	marisol_rios@gmail.com	1960-09-02	Femenino	Brasil
2	2	Manuel	Núñez	manuel_nunez@gmail.com	1968-10-30	Masculino	Colombia
3	3	María	Martínez	maria_martinez@yahoo.com	2003-06-11	Femenino	Reino Unido
4	4	Carmen	Díaz	carmen_diaz@gmail.com	1961-04-24	Femenino	Argentina
5	5	Luis	Mejía	luis_mejia@hotmail.com	1966-02-24	Masculino	Honduras
6	6	Estela	Rodríguez	estela_rodriguez@yahoo.com	2009-12-02	Femenino	Canadá
7	7	Sebastián	Castro	sebastian_castro@outlook.com	1977-12-11	Masculino	Chile
8	8	Roberto	Pérez	roberto_perez@gmail.com	1989-12-02	Masculino	Argentina
9	9	Carlos	López	carlos_lopez@gmail.com	2012-09-16	Masculino	Panamá
10	10	Leticia	Morales	leticia_morales@outlook.com	1962-06-07	Femenino	Nicaragua
11	11	Manuel	Guerrero	manuel_guerrero@hotmail.com	1963-07-21	Masculino	Estados Unidos
12	12	Lucía	Castro	lucia_castro@outlook.com	1986-12-09	Femenino	Reino Unido
13	13	Francisco	Herrera	francisco_herrera@yahoo.com	2004-06-26	Masculino	Brasil
14	14	Enrique	Rodríguez	enrique_rodriguez@hotmail.com	2013-05-16	Masculino	Reino Unido
15	15	Hugo	Ortiz	hugo_ortiz@yahoo.com	1971-02-11	Masculino	Italia
16	16	Marisol	Torres	marisol_torres@gmail.com	2003-07-13	Femenino	Estados Unidos
17	17	Sebastián	Ramírez	sebastian_ramirez@outlook.com	2012-01-27	Masculino	Costa Rica
18	18	Hugo	Palacios	hugo_palacios@gmail.com	1967-01-16	Masculino	México
19	19	Ricardo	Rodríguez	ricardo_rodriguez@outlook.com	2005-06-17	Masculino	España
20	20	Claudia	Pérez	claudia_perez@yahoo.com	1966-07-20	Femenino	Italia
21	21	Pedro	Vega	pedro_vega@outlook.com	2001-03-25	Masculino	Colombia
22	22	Isabel	Cornejo	isabel_cornejo@hotmail.com	1975-09-03	Femenino	México
23	23	Ana	Silva	ana_silva@outlook.com	2004-10-26	Femenino	Argentina

Y podemos realizar los select count(\*) from dbo.tabla para corroborar que todos los datos fueron ingresados:

	Cantidad_Encuesta	▼
1	30000	

	Cantidad_Habitacion	▼
1	500	

	Cantidad_Pago	▼
1	70000	

	Cantidad_Reserva	▼
1	70000	

	Cantidad_Empleado	▼
1	1000	

	Cantidad_Tarea	▼
1	3000	

	Cantidad_Producto	▼
1	150	

## Conclusión

El proceso ETL descrito, implementado mediante Visual Studio y SSIS, permite una transferencia eficiente y confiable de los datos operativos desde la base transaccional HotelOLTP hacia el HotelDW alojado en SQL Server con Docker. La validación a través de SSIS en Visual Studio asegura la correcta carga y consistencia de los datos, estableciendo una base robusta y confiable para los análisis estratégicos, dashboards interactivos y la generación continua de reportes de inteligencia de negocios que apoyan decisiones informadas en la gestión del hotel.