

Generación de script SQL mediante Forward Engineer de MySQL Workbench

Sentencias SQL para la creación de la BD:

```
CREATE TABLE cliente (  
  cliente_id INT NOT NULL IDENTITY,  
  first_name VARCHAR(45) NOT NULL,  
  last_name VARCHAR(45) NOT NULL,  
  email VARCHAR(50) NULL DEFAULT NULL,  
  PRIMARY KEY (cliente_id),  
  INDEX idx_last_name (last_name ASC));
```

```
CREATE TABLE linea_producto (  
  lineaproducto_id INT NOT NULL IDENTITY,  
  title VARCHAR(255) NOT NULL,  
  description TEXT NULL,  
  last_update TIMESTAMP NULL,  
  INDEX idx_title (title ASC),  
  PRIMARY KEY (lineaproducto_id));
```

```
CREATE TABLE producto (  
  producto_id INT NOT NULL IDENTITY,  
  lineaproducto_id INT NOT NULL,  
  name VARCHAR(45) NOT NULL,  
  PRIMARY KEY (producto_id),  
  INDEX idx_fk_film_id (lineaproducto_id ASC),  
  INDEX idx_store_id_film_id (lineaproducto_id ASC),  
  CONSTRAINT fk_inventory_film
```

```
FOREIGN KEY (lineaproducto_id)
REFERENCES linea_producto (lineaproducto_id));
```

```
CREATE TABLE empleado (
    empleado_id INT NOT NULL IDENTITY,
    first_name VARCHAR(45) NOT NULL,
    last_name VARCHAR(45) NOT NULL,
    email VARCHAR(50) NULL DEFAULT NULL,
    username VARCHAR(16) NOT NULL,
    password VARCHAR(40) NULL DEFAULT NULL,
    last_update TIMESTAMP NULL,
    PRIMARY KEY (empleado_id));
```

```
CREATE TABLE orden_venta (
    ordenventa_id INT NOT NULL IDENTITY,
    producto_id INT NOT NULL,
    cliente_id INT NOT NULL,
    return_date DATETIME NULL,
    empleado_id INT NOT NULL,
    last_update TIMESTAMP NULL,
    PRIMARY KEY (ordenventa_id),
    INDEX idx_fk_inventory_id (producto_id ASC),
    INDEX idx_fk_customer_id (cliente_id ASC),
    INDEX idx_fk_staff_id (empleado_id ASC),
    CONSTRAINT fk_rental_staff
        FOREIGN KEY (empleado_id)
        REFERENCES empleado (empleado_id),
    CONSTRAINT fk_rental_inventory
        FOREIGN KEY (producto_id)
```

```
REFERENCES producto (producto_id),  
CONSTRAINT fk_rental_customer  
FOREIGN KEY (cliente_id)  
REFERENCES cliente (cliente_id));
```

1.3. Consulta de pedidos y clientes.

Azure:

Sentencia DDL para Crear:

```
CREATE TABLE ejercicio_13 (  
    id int IDENTITY,  
    pedido varchar(5) NOT NULL,  
    cliente varchar(20) NOT NULL,  
    pais varchar(20) NOT NULL,  
    precio INT NOT NULL,  
    PRIMARY KEY (id)  
);
```

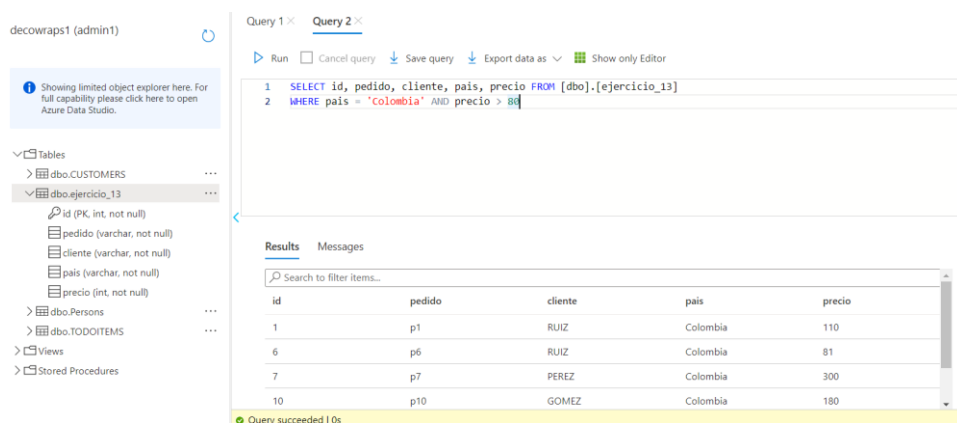
Sentencia DML para Insertar:

```
INSERT INTO [dbo].[ejercicio_13]  
(pedido, cliente, pais, precio)  
VALUES  
( 'p1', 'RUIZ', 'Colombia', 110),  
( 'p2', 'PEREZ', 'Argentina', 300),  
( 'p3', 'GOMEZ', 'Brasil', 250),
```

```
( 'p4', 'RUIZ', 'Colombia', 37),
( 'p5', 'GOMEZ', 'Colombia', 60),
( 'p6', 'RUIZ', 'Colombia', 81),
( 'p7', 'PEREZ', 'Colombia', 300),
( 'p8', 'GOMEZ', 'Peru', 250),
( 'p9', 'RUIZ', 'Bolivia', 490),
( 'p10', 'GOMEZ', 'Colombia', 180),
( 'p11', 'RUIZ', 'Colombia', 490),
( 'p12', 'GOMEZ', 'Colombia', 220);
```

Parte de la consulta que obtiene la lista por país Colombia y precio mayor a 80:

```
SELECT id, pedido, cliente, pais, precio FROM [dbo].[ejercicio_13]
WHERE pais = 'Colombia' AND precio > 80
```



The screenshot shows the SQL Server Enterprise Manager interface. On the left, the 'Object Explorer' pane displays the database structure for 'decowrapst (admin1)', including tables like 'dbo.CUSTOMERS', 'dbo.ejercicio_13', 'dbo.Persons', and 'dbo.TODOITEMS'. The 'dbo.ejercicio_13' table is selected, showing its columns: 'id' (PK, int, not null), 'pedido' (varchar, not null), 'cliente' (varchar, not null), 'pais' (varchar, not null), and 'precio' (int, not null). The main window displays 'Query 2' with the following SQL code:

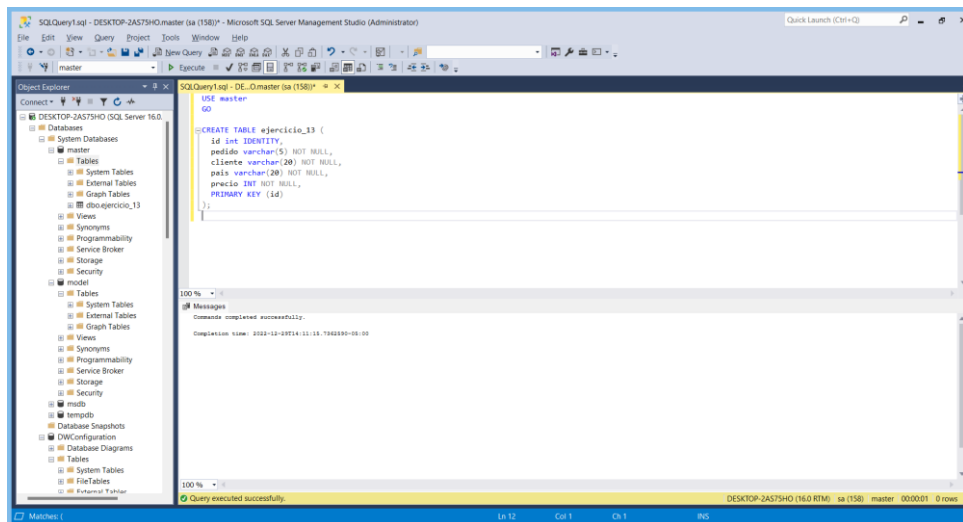
```
1 SELECT id, pedido, cliente, pais, precio FROM [dbo].[ejercicio_13]
2 WHERE pais = 'Colombia' AND precio > 80
```

Below the query editor, the 'Results' pane shows the output of the query as a table with 5 columns: 'id', 'pedido', 'cliente', 'pais', and 'precio'. The results are as follows:

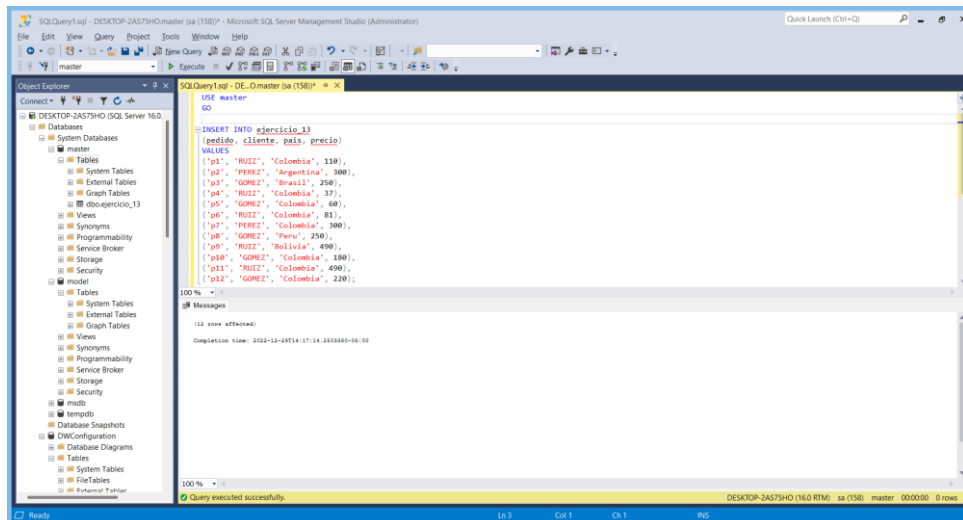
id	pedido	cliente	pais	precio
1	p1	RUIZ	Colombia	110
6	p6	RUIZ	Colombia	81
7	p7	PEREZ	Colombia	300
10	p10	GOMEZ	Colombia	180

A status bar at the bottom indicates 'Query succeeded | 0s'.

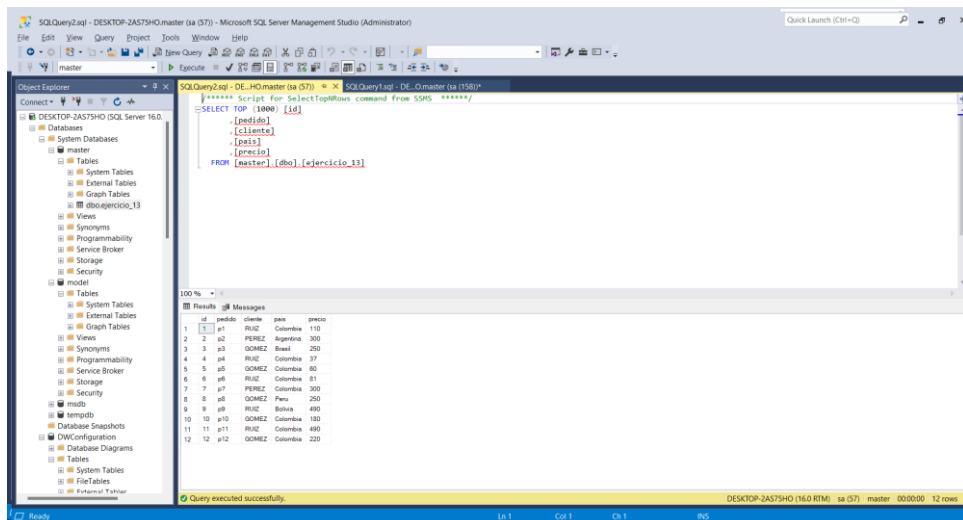
Local:



Sentencia DDL para Crear Local

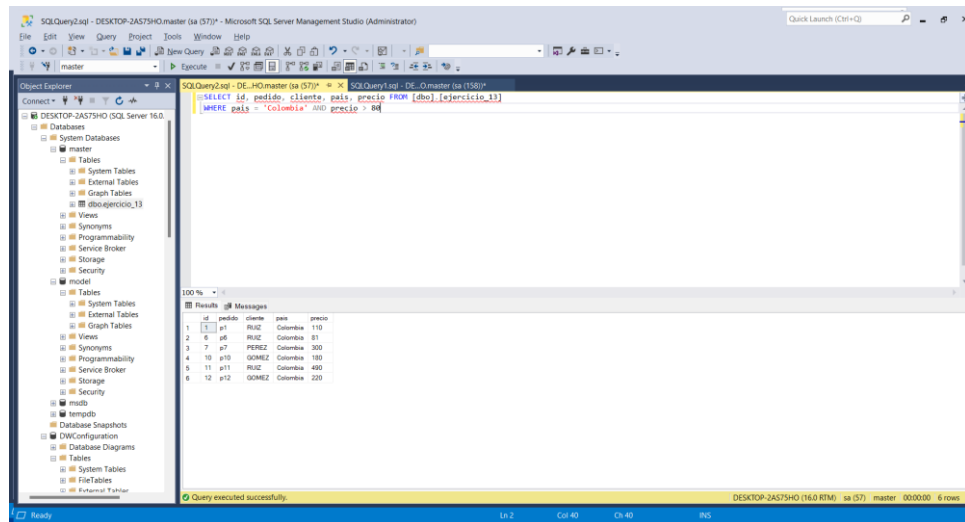


Sentencia DML para Insertar Local



Consulta que muestra registros insertados

Parte de la consulta que obtiene la lista por país Colombia y precio mayor a 80:



The screenshot shows the Microsoft SQL Server Enterprise Manager interface. The left pane displays the 'Object Explorer' with the 'DESKTOP-JA57SHO' server selected. The 'Query' tab is active, showing a SQL query: `SELECT id, pedido, cliente, pais, precio FROM [dbo].[ejercicio_13] WHERE pais = 'Colombia' AND precio > 80`. The 'Results' pane at the bottom displays the query output as a table with 6 rows and 5 columns: id, pedido, cliente, pais, and precio. The status bar at the bottom indicates 'Query executed successfully' and '6 rows'.

	id	pedido	cliente	pais	precio
1	1	p1	RUZ	Colombia	110
2	6	p6	RUZ	Colombia	81
3	7	p7	PEREZ	Colombia	300
4	10	p10	DOMEZ	Colombia	180
5	11	p11	RUZ	Colombia	490
6	12	p12	DOMEZ	Colombia	220

Consulta ejercicio 1.3. parte 1

2. Desarrollo

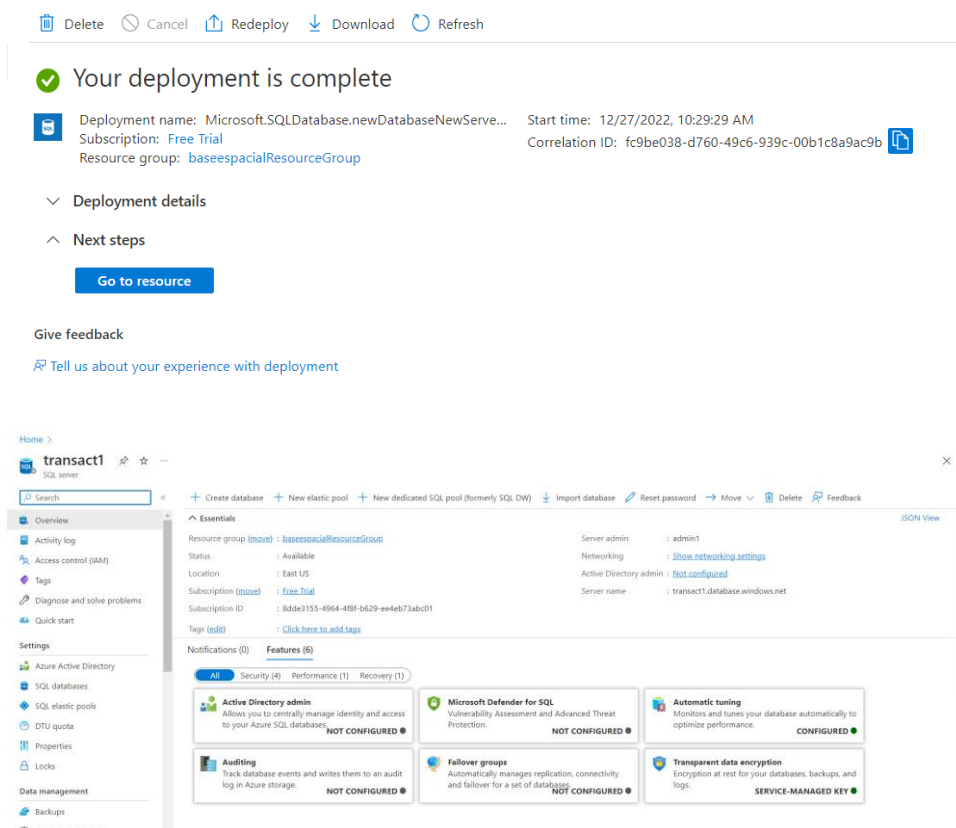
Para la implementación de la aplicación backend, se siguió el siguiente tutorial:

<https://learn.microsoft.com/en-us/aspnet/core/tutorials/first-web-api?view=aspnetcore-6.0&tabs=visual-studio>

* Repositorio aplicación backend: <https://github.com/da8y01/DWback1>

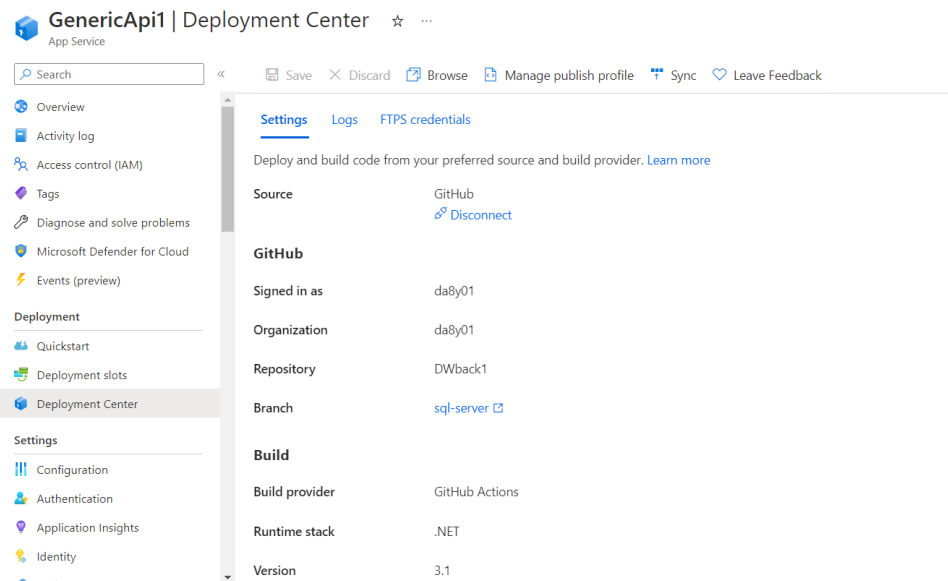
* Repositorio aplicación frontend: <https://github.com/da8y01/DWfront1>

* Configurar base de datos SQL Server en Azure: transact1.database.windows.net

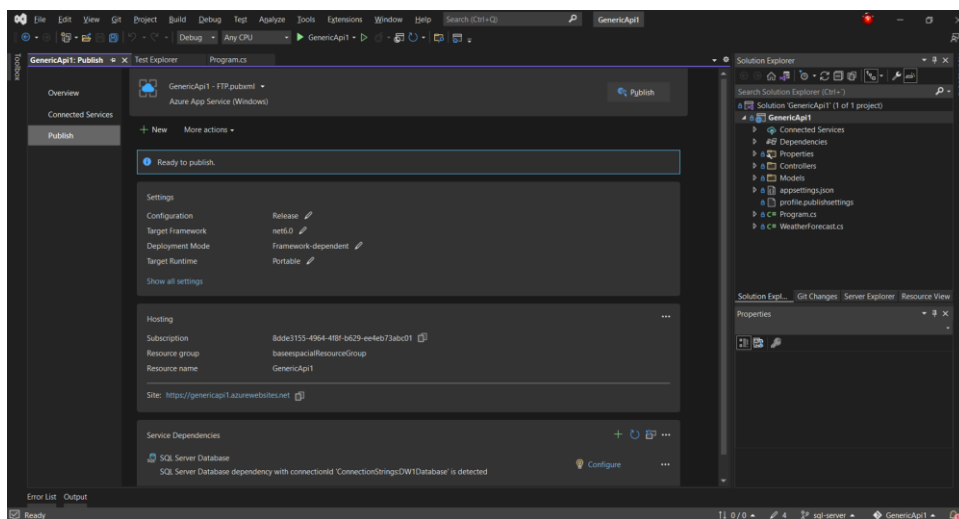


Para el despliegue, debido a un error presentado en la instalación de VS2022 Community Edition usada, se consultó diversa documentación; el método usado fue el siguiente:

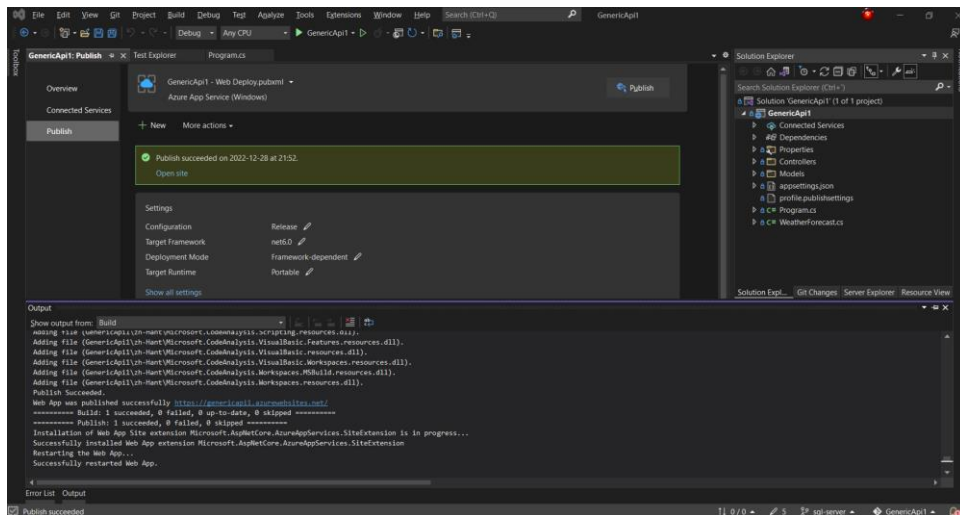
- En Azure conectar el repositorio de GitHub de la aplicación backend al AppService.



- Descargar el “Publish profile” de Azure.
- Importar el publish profile en VS2022.



- Correr el despliegue desde ahí (VS2022).



Bibliografía y links

<https://learn.microsoft.com/en-us/aspnet/core/tutorials/first-web-api?view=aspnetcore-6.0&tabs=visual-studio>

<https://learn.microsoft.com/en-us/azure/app-service/tutorial-dotnetcore-sqldb-app>

<https://learn.microsoft.com/en-us/ef/core/dbcontext-configuration/>