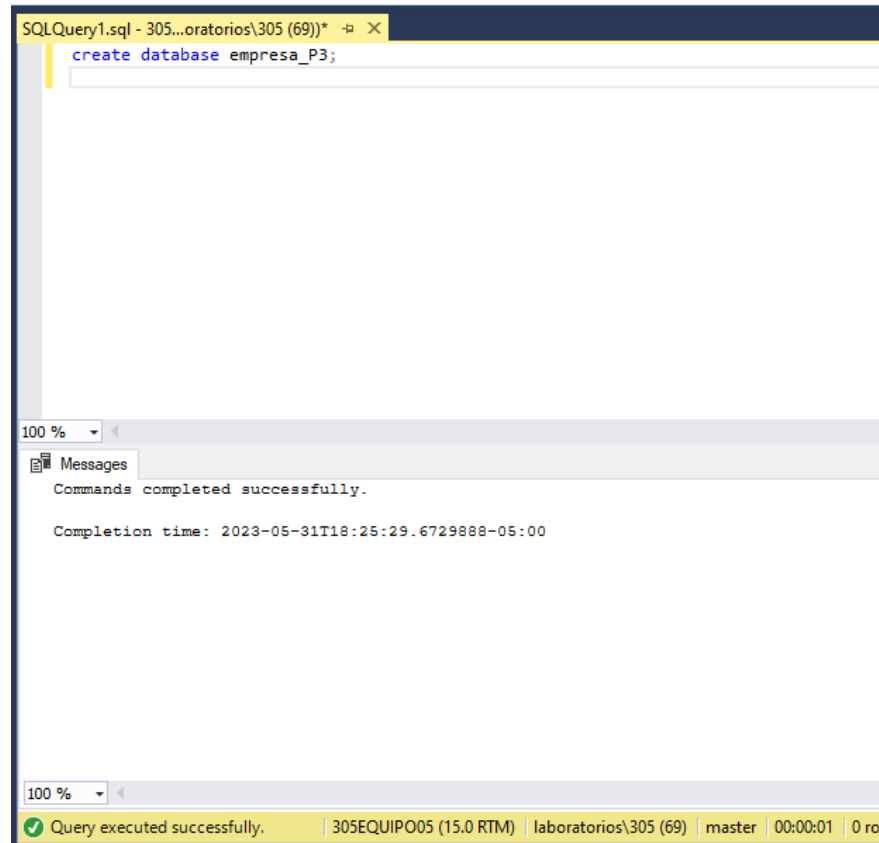

--PARCIAL DE TERCER CORTE DE LENGUAJE ESTRUCTURADO SQL
--MOISES ZABALETA CRUZ
--1033810224
--Email: MZABALETAC@ACADEMIA.USBBOG.EDU.CO

-1- Diseñe la siguiente estructura de tablas, haga las correcciones necesarias para su modelamiento y verifique si hace falta crear más tablas.

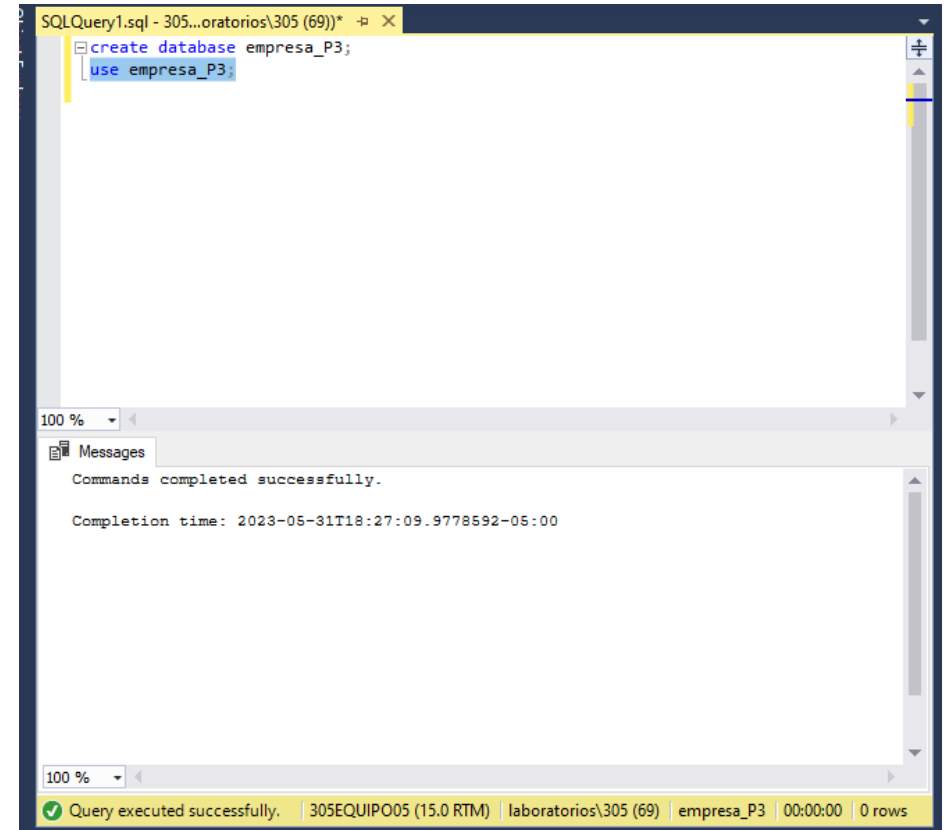
-1.1- creación de base de datos

`create database empresa_P3;`



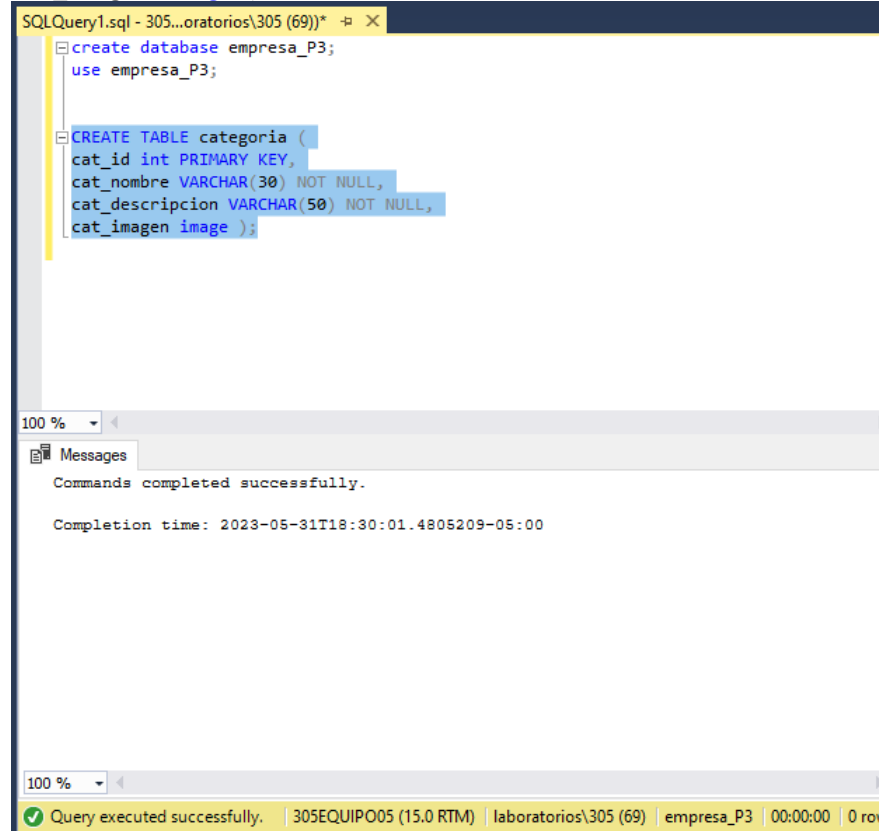
-1.2-uso de la base de datos

`use empresa_P3;`



-1.3-creacion de tabla categoria

```
CREATE TABLE categoria (  
cat_id int PRIMARY KEY,  
cat_nombre VARCHAR(30) NOT NULL,  
cat_descripcion VARCHAR(50) NOT NULL,  
cat_imagen image );
```



The screenshot shows the SQL Server Enterprise Manager interface. The top pane displays the SQL query for creating the 'categoria' table. The bottom pane shows the 'Messages' tab with the confirmation 'Commands completed successfully.' and the completion time '2023-05-31T18:30:01.4805209-05:00'. The status bar at the bottom indicates 'Query executed successfully.' and provides details about the query execution, including the server name '305EQUIPO05 (15.0 RTM)', the database 'laboratorios\305 (69)', the schema 'empresa_P3', the execution time '00:00:00', and the rows affected '0 rows'.

```
SQLQuery1.sql - 305...oratorios\305 (69))* - X  
create database empresa_P3;  
use empresa_P3;  
  
CREATE TABLE categoria (  
cat_id int PRIMARY KEY,  
cat_nombre VARCHAR(30) NOT NULL,  
cat_descripcion VARCHAR(50) NOT NULL,  
cat_imagen image );  
  
100 %  
Messages  
Commands completed successfully.  
  
Completion time: 2023-05-31T18:30:01.4805209-05:00  
  
100 %  
Query executed successfully. | 305EQUIPO05 (15.0 RTM) | laboratorios\305 (69) | empresa_P3 | 00:00:00 | 0 rows affected
```

-1.4-creacion de tabla proveedor

```
CREATE TABLE proveedor (  
pro_id int PRIMARY KEY,  
pro_cia VARCHAR(30) NOT NULL,  
pro_titulo VARCHAR(50),  
pro_dir VARCHAR(50),  
pro_ciudad VARCHAR(40),  
pro_region VARCHAR(30),  
pro_codigoPostal VARCHAR(10),  
pro_pais VARCHAR(30),  
pro_tlf VARCHAR(60),  
pro_fax VARCHAR(40),  
pro_pagina VARCHAR(100) );
```

```
SQLQuery1.sql - 305...oratorios\305 (69))* -+ X
CREATE TABLE proveedor (
  pro_id int PRIMARY KEY,
  pro_cia VARCHAR(30) NOT NULL,
  pro_titulo VARCHAR(50),
  pro_dir VARCHAR(50),
  pro_ciudad VARCHAR(40),
  pro_region VARCHAR(30),
  pro_codigoPostal VARCHAR(10),
  pro_pais VARCHAR(30),
  pro_tlf VARCHAR(60),
  pro_fax VARCHAR(40),
  pro_pagina VARCHAR(100) );

100 %
Messages
Commands completed successfully.

Completion time: 2023-05-31T18:32:17.2155695-05:00

100 %
Query executed successfully. | 305EQUIPO05 (15.0 RTM) | laboratorios\305 (69) | empresa_P3 | 00:00:00 | 0 rows
```

```
CREATE TABLE articulo (
  art_id int PRIMARY KEY,
  art_nombre VARCHAR(30) NOT NULL,
  pro_id int NOT NULL,
  cat_id int NOT NULL,
  art_cant_por_und int,
  art_pu money NOT NULL,
  art_stock int,
  art_und_en_ordenes int,
  art_nivel_para_ordenar int,
  art_descontinuado bit,
  FOREIGN KEY (pro_id) REFERENCES proveedor (pro_id),
  FOREIGN KEY (cat_id) REFERENCES categoria (cat_id));
```

-1.4-creacion de tabla articulo con foráneas de las otras tablas

```
SQLQuery1.sql - 305...ratorios\305 (69))* -> X
CREATE TABLE articulo (
  art_id int PRIMARY KEY,
  art_nombre VARCHAR(30) NOT NULL,
  pro_id int NOT NULL,
  cat_id int NOT NULL,
  art_cant_por_und int,
  art_pu money NOT NULL,
  art_stock int,
  art_und_en_ordenes int,
  art_nivel_para_ordenar int,
  art_descontinuado bit,
  FOREIGN KEY (pro_id) REFERENCES proveedor (pro_id),
  FOREIGN KEY (cat_id) REFERENCES categoria (cat_id));

100 %
Messages
Commands completed successfully.

Completion time: 2023-05-31T18:35:24.1588552-05:00

100 %
Query executed successfully. | 305EQUIPO05 (15.0 RTM) | laboratorios\305 (69) | empresa_P3 | 00:00:00 | 0 ro
```

```
SQLQuery1.sql - 305...ratorios\305 (69))* -> X
FOREIGN KEY (cat_id) REFERENCES categoria (cat_id));

CREATE TABLE cliente (
  cli_id int PRIMARY KEY,
  cli_cc VARCHAR(30) NOT NULL,
  cli_nom VARCHAR(50),
  cli_dir VARCHAR(50),
  cli_ciudad VARCHAR(40),
  cli_region VARCHAR(30),
  cli_codigoPostal VARCHAR(10),
  cli_pais VARCHAR(30),
  cli_tlf VARCHAR(60));

--1. Realizar un procedimiento almacenado que devuelva los clientes
--(Customers) según el país (Country).

100 %
Messages
Commands completed successfully.

Completion time: 2023-05-31T18:48:59.9427439-05:00

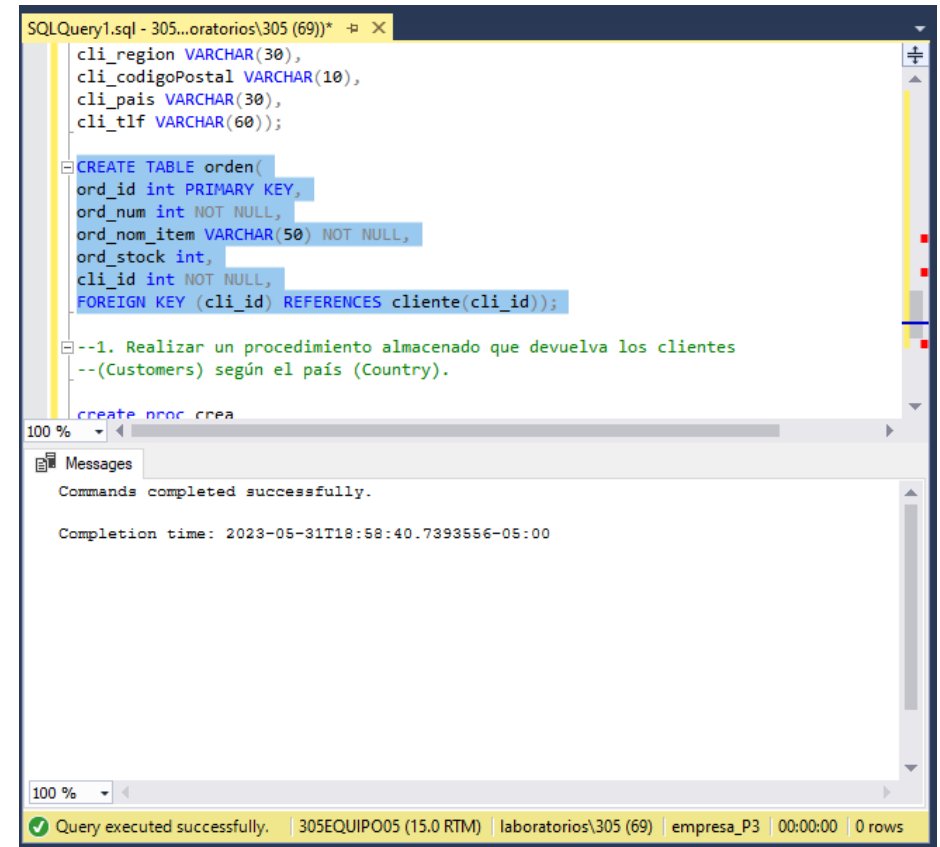
100 %
Query executed successfully. | 305EQUIPO05 (15.0 RTM) | laboratorios\305 (69) | empresa_P3 | 00:00:00 | 0 rows
```

- 1.4-creacion de tabla articulo
- 2. Realice las Siguietes consultas.

1.5 creación de tabla cliente con foráneas de las otras tablas

1.6 creacion de tabla orden

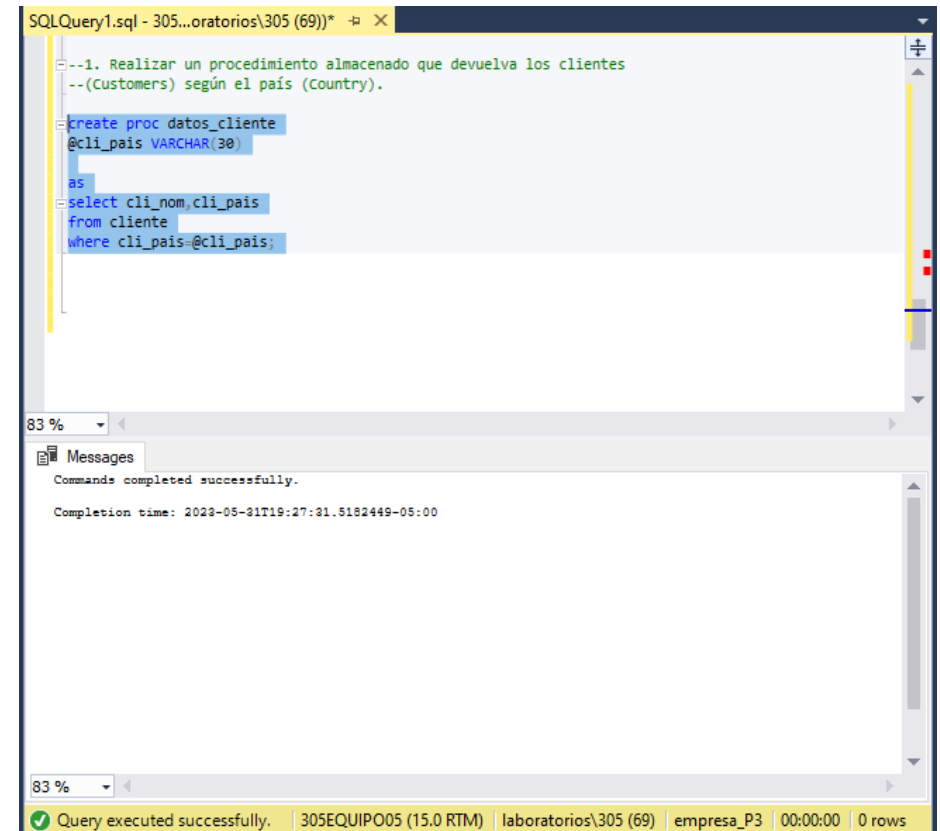
```
CREATE TABLE orden(  
ord_id int PRIMARY KEY,  
ord_num int NOT NULL,  
ord_nom_item VARCHAR(50) NOT NULL,  
ord_stock int,  
cli_id int NOT NULL,  
FOREIGN KEY (cli_id) REFERENCES cliente(cli_id));
```



```
--1. Realizar un procedimiento almacenado que devuelva los  
clientes  
--(Customers) según el país (Country).
```

```
create proc datos_cliente  
@cli_pais VARCHAR(30)
```

```
as  
select cli_nom,cli_pais  
from cliente  
where cli_pais=@cli_pais;
```



The screenshot shows the SQL Server Enterprise Manager interface. The top pane displays a SQL query window titled 'SQLQuery1.sql - 305...oratorios\305 (69))' with the following code:

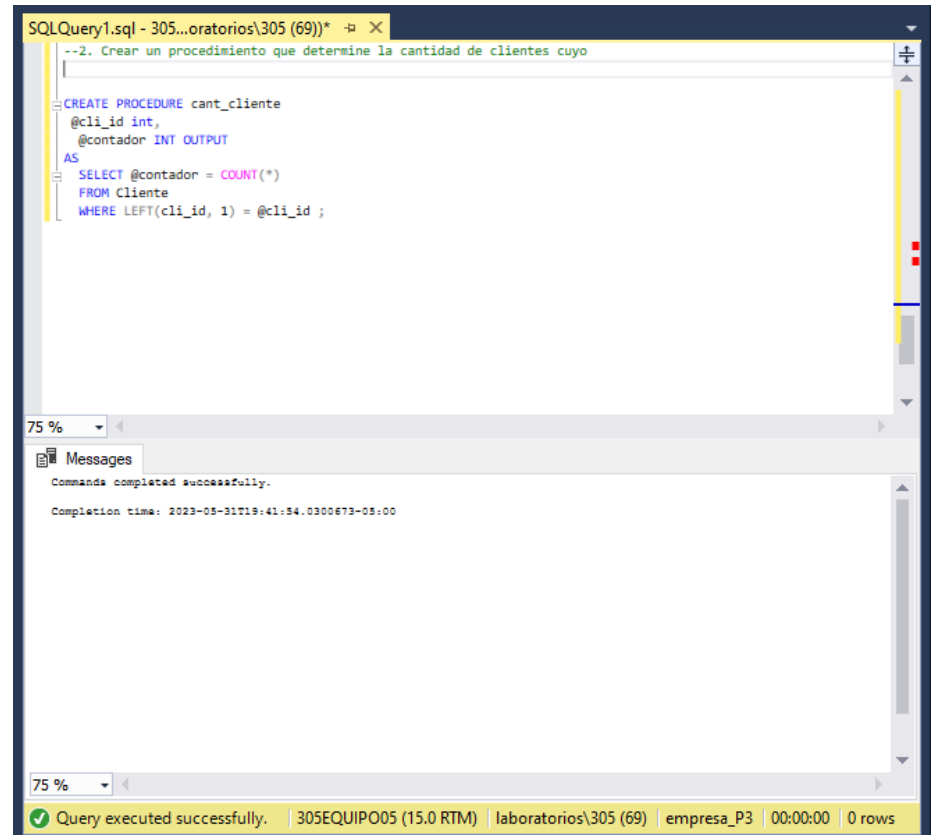
```
--1. Realizar un procedimiento almacenado que devuelva los clientes  
--(Customers) según el país (Country).  
  
create proc datos_cliente  
@cli_pais VARCHAR(30)  
as  
select cli_nom,cli_pais  
from cliente  
where cli_pais=@cli_pais;
```

The bottom pane shows the 'Messages' tab with the following output:

```
Commands completed successfully.  
  
Completion time: 2023-05-31T19:27:31.5182449-05:00
```

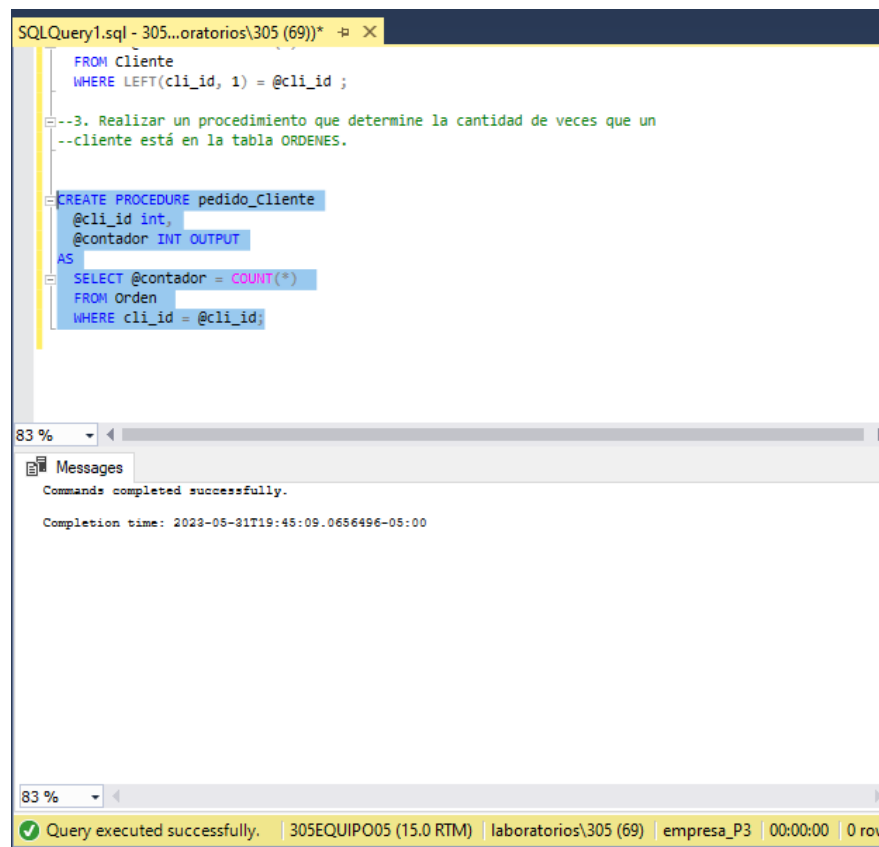
The status bar at the bottom indicates: 'Query executed successfully. | 305EQUIPO05 (15.0 RTM) | laboratorios\305 (69) | empresa_P3 | 00:00:00 | 0 rows'.

```
CREATE PROCEDURE cant_cliente
@cli_id int,
@contador INT OUTPUT
AS
    SELECT @contador = COUNT(*)
    FROM Cliente
    WHERE LEFT(cli_id, 1) = @cli_id ;
```



3. Realizar un procedimiento que determine la cantidad de veces que un cliente está en la tabla ORDENES.

```
CREATE PROCEDURE pedido_Cliente
    @cli_id int,
    @contador INT OUTPUT
AS
    SELECT @contador = COUNT(*)
    FROM Orden
    WHERE cli_id = @cli_id;
```



The screenshot shows the SQL Server Enterprise Manager interface. The top pane displays a SQL query window titled 'SQLQuery1.sql - 305...oratorios\305 (69))' containing the following code:

```
FROM Cliente
WHERE LEFT(cli_id, 1) = @cli_id ;

--3. Realizar un procedimiento que determine la cantidad de veces que un
--cliente está en la tabla ORDENES.

CREATE PROCEDURE pedido_Cliente
    @cli_id int,
    @contador INT OUTPUT
AS
    SELECT @contador = COUNT(*)
    FROM Orden
    WHERE cli_id = @cli_id;
```

The bottom pane shows the 'Messages' tab with the following text:

```
Commands completed successfully.
Completion time: 2023-05-31T19:45:09.0656496-05:00
```

The status bar at the bottom indicates: 'Query executed successfully. | 305EQUIPO05 (15.0 RTM) | laboratorios\305 (69) | empresa_P3 | 00:00:00 | 0 rows affected'.

4. Realizar un procedimiento que seleccione todos los registros de tabla Productos y Categorías a la vez.

```
CREATE PROCEDURE ProductosCategorias
AS
    SELECT * FROM Artículo
    FULL JOIN Categoria ON Artículo.cat_id = Categoria.cat_id
```



```
SQLQuery1.sql - 305...ratorios\305 (69))* -> X
CREATE PROCEDURE pedido_cliente
    @cli_id int,
    @contador INT OUTPUT
AS
SELECT @contador = COUNT(*)
FROM Orden
WHERE cli_id = @cli_id;

--4. Realizar un procedimiento que seleccione todos los registros de tabla
--Productos y Categorías a la vez.
CREATE PROCEDURE ProductosCategorias
AS
SELECT * FROM Articulo
FULL JOIN Categoria ON Articulo.cat_id = Categoria.cat_id;
EXEC ProductosCategorias

83 %
Messages
Commands completed successfully.
Completion time: 2023-05-31T19:50:27.3242778-05:00

83 %
Query executed successfully. | 305EQUIPO05 (15.0 RTM) | laboratorios\305 (69) | empresa_P3 | 00:00:00 | 0 ro
```

5. Realizar un procedimiento que seleccione todos los productos que no corresponde a la categoría bebidas (1)

```
CREATE PROCEDURE NoBebidas
AS
SELECT * FROM Articulo WHERE cat_id != 1;

EXEC NoBebidas
```

```
SQLQuery1.sql - 305...ratorios\305 (69))* - X
--SELECT * FROM Artículo
--FULL JOIN Categoria ON Artículo.cat_id = Categoria.cat_id;

EXEC ProductosCategorias

--5. Realizar un procedimiento que seleccione todos los productos que no
--corresponde a la categoría bebidas (1)
CREATE PROCEDURE NoBebidas
AS
    SELECT * FROM Artículo WHERE cat_id != 1;

EXEC NoBebidas

83 %
Messages
Commands completed successfully.
Completion time: 2023-05-21T19:54:38.2291618-05:00

83 %
Query executed successfully. | 305EQUIPO05 (15.0 RTM) | laboratorios\305 (69) | empresa_P3 | 00:00:00 | 0 rows
```

6. Realizar un procedimiento que obtenga la cantidad de registros que no corresponde a condimentos

```
CREATE PROCEDURE Prod_No_Condimentos
    @cantidad INT OUTPUT
AS

    SELECT @cantidad = COUNT(*) FROM Artículo WHERE cat_id
<> 2;
```

```
SQLQuery1.sql - 305...ratorios\305 (69))* - X

--5. Realizar un procedimiento que seleccione todos los productos que no
--corresponde a la categoría bebidas (1)
CREATE PROCEDURE NoBebidas
AS
    SELECT * FROM Artículo WHERE cat_id != 1;

EXEC NoBebidas

--6. Realizar un procedimiento que obtenga la cantidad de registros que no
--corresponde a condimentos
CREATE PROCEDURE Prod_No_Condimentos
    @cantidad INT OUTPUT
AS
    SELECT @cantidad = COUNT(*) FROM Artículo WHERE cat_id <> 2;

83 %
Messages
Commands completed successfully.
Completion time: 2023-05-21T19:57:41.8927944-05:00

83 %
Query executed successfully. | 305EQUIPO05 (15.0 RTM) | laboratorios\305 (69) | empresa_P3 | 00:00:00 | 0 rows
```

7. Realizar un procedimiento que seleccionar todos los campos de los registros que no corresponden a categoría mariscos de la tabla productos.

```
CREATE PROCEDURE Prod_No_Mar
AS
```

```
SELECT * FROM Artículo WHERE cat_id <> 3;
```

```
SQLQuery1.sql - 305...oratorios\305 (69)) * -> X
--corresponde a condimentos
CREATE PROCEDURE Prod_No_Condimentos
@cantidad INT OUTPUT
AS
SELECT @cantidad = COUNT(*) FROM Artículo WHERE cat_id <> 2;

--7. Realizar un procedimiento que seleccionar todos los campos de los
--registros que no corresponden a categoría mariscos de la tabla
--productos.
CREATE PROCEDURE Prod_No_Mar
AS
SELECT * FROM Artículo WHERE cat_id <> 3;
```

83 %

Messages

Commands completed successfully.

Completion time: 2023-05-31T19:59:07.1072276-05:00

83 %

Query executed successfully. | 305EQUIPO05 (15.0 RTM) | laboratorios\305 (69) | empresa_P3 | 00:00:00 | 0 rows

8. Realizar un procedimiento que seleccionar los campos nombre del producto y precio (únicamente) de los productos diferentes a carnes.

```
CREATE PROCEDURE Prod_No_Carne
AS
```

```
SELECT art_nombre, art_pu FROM Artículo WHERE cat_id <>
```

```
SQLQuery1.sql - 305...oratorios\305 (69)) * -> X
--7. Realizar un procedimiento que seleccionar todos los campos de los
--registros que no corresponden a categoría mariscos de la tabla
--productos.
CREATE PROCEDURE Prod_No_Mar
AS
SELECT * FROM Artículo WHERE cat_id <> 3;

--8. Realizar un procedimiento que seleccionar los campos nombre del
--producto y precio (únicamente) de los productos diferentes a carnes.
CREATE PROCEDURE Prod_No_Carne
AS
SELECT art_nombre, art_pu FROM Artículo WHERE cat_id <> 4;
```

83 %

Messages

Commands completed successfully.

Completion time: 2023-05-31T20:00:23.3390552-05:00

83 %

Query executed successfully. | 305EQUIPO05 (15.0 RTM) | laboratorios\305 (69) | empresa_P3 | 00:00:00 | 0 rows

9. Realizar un procedimiento que obtenga la cantidad de productos granos y cereales.
10. Realizar un procedimiento que seleccione los campos nombre del producto y precio (únicamente) de los quesos y carnes.
11. Realizar un procedimiento que seleccionar los campos nombre del producto, precio y stock (únicamente) de las frutas secas y mariscos.
12. Realizar un procedimiento que seleccionar el promedio de los precios de los confites.