

“SQL Server Management Studio”

Tarea procesual hito 4
Estudiante: Kevin Mamani Mamani
Asignatura: BASE DE DATOS I
Carrera: Ingeniería de sistemas
Paralelo: (4)
Docente: Lic. William Barra Paredes

```
Python 3.4.1 Shell
File Edit Shell Debug Options Windows Help
Python 3.4.1 (v3.4.1:c0e311e010fc, May 18 2014, 10:38:22) [MSC v.1600 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> lista = []
>>> lista__errores = []
>>> def solopalabras(palabra):
>>>     if not isinstance(palabra, str):
>>>         global lista__errores
>>>         raise Exception("SOLO PUEDEN ESCRIBIRSE PALABRAS") and lista__errores.append(palabra)
>>>     global lista
>>>     lista.append(palabra)

>>> solopalabras("HOLA")
>>> lista
['HOLA']
>>> solopalabras("MUNDO")
>>> lista
['HOLA', 'MUNDO']
>>> solopalabras(88.99)
Traceback (most recent call last):
  File "<pyshell#8>", line 1, in <module>
    solopalabras(88.99)
  File "<pyshell#3>", line 4, in solopalabras
    raise Exception("SOLO PUEDEN ESCRIBIRSE PALABRAS") and lista__errores.append(palabra)
TypeError: exceptions must derive from BaseException
>>> # A PESAR DE LA EXCEPCIÓN, SE EJECUTA EL CÓDIGO AÑADIDO A raise:
>>> lista__errores
[88.99]
>>>
```

Fecha: 20/11/2023

The background is a blue gradient. In the corners, there are white line art illustrations of circuit boards or neural networks, with lines connecting to small circles.

**QUE ES DDL Y DML, ADICIONALMENTE MUESTRA
UN EJEMPLO EN LA BASE DE DATOS UNIFRANZITOS.**

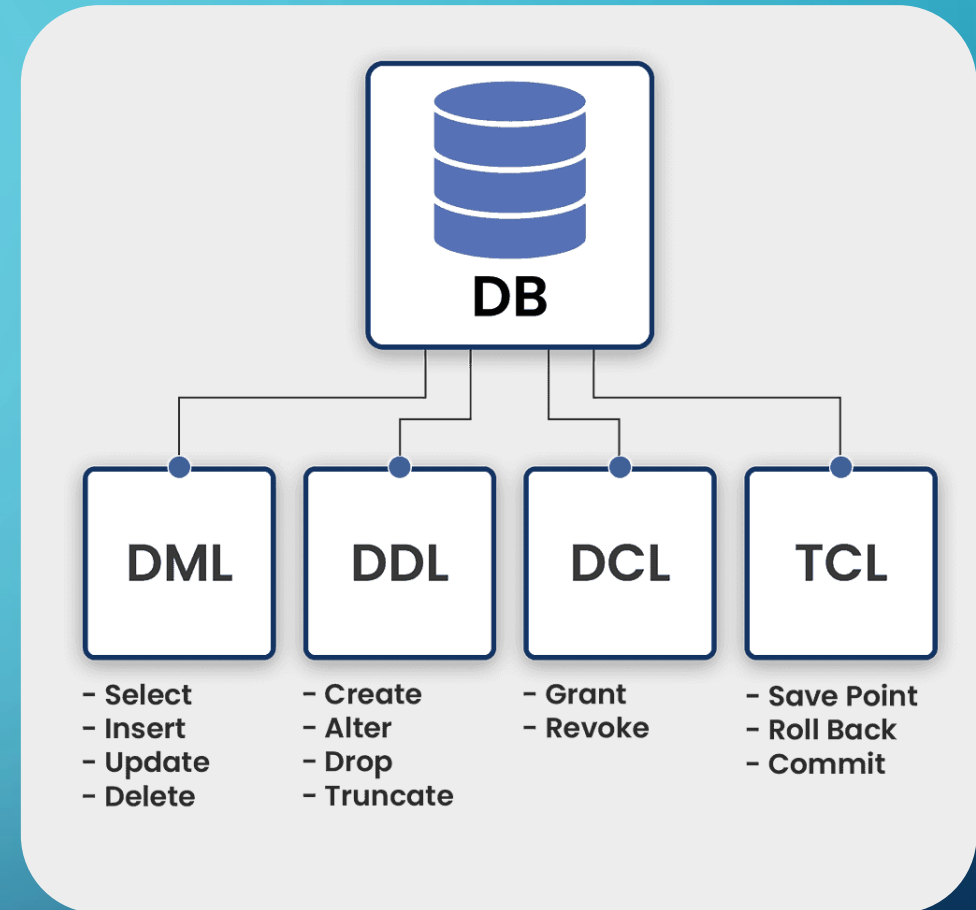
DDL Y DML SON DOS SUBCONJUNTOS DE SQL (STRUCTURED QUERY LANGUAGE) QUE SE UTILIZAN PARA REALIZAR OPERACIONES ESPECÍFICAS EN BASES DE DATOS.

1.DDL (DATA DEFINITION LANGUAGE - LENGUAJE DE DEFINICIÓN DE DATOS):

1. Se utiliza para definir y gestionar la estructura de la base de datos.
2. Incluye comandos como CREATE, ALTER, DROP, etc.
3. Los cambios realizados con DDL afectan la estructura de la base de datos.

2.DML (DATA MANIPULATION LANGUAGE - LENGUAJE DE MANIPULACIÓN DE DATOS):

1. Se utiliza para gestionar los datos almacenados en la base de datos.
2. Incluye comandos como SELECT, INSERT, UPDATE, DELETE, etc.
3. Los cambios realizados con DML afectan los datos almacenados en la base de datos.



Ejemplo DDL:

```
CREATE DATABASE UNIFRANZITOS;
```

```
USE UNIFRANZITOS;
```

```
CREATE TABLE campeonato
```

```
(  
    id_campeonato VARCHAR(12) PRIMARY KEY,  
    nombre_campeonato VARCHAR(30) NOT NULL,  
    sede VARCHAR(20) NOT NULL  
);
```

```
CREATE TABLE equipo
```

```
(  
    id_equipo VARCHAR(12) PRIMARY KEY,  
    nombre_equipo VARCHAR(30) NOT NULL,  
    categoria VARCHAR(8) NOT NULL ,  
    id_campeonato VARCHAR(12),  
    FOREIGN KEY (id_campeonato) REFERENCES campeonato (id_campeonato)  
);
```

Ejemplo DML:

Query1.sql - D:\OHGO\Desktop (59)*

```
c1 VARCHAR(50),  
edad INTEGER,  
id_equipo VARCHAR (100)  
FOREIGN KEY (id_equipo) REFERENCES equipo (id_equipo),  
);
```

```
INSERT INTO campeonato (id_campeonato, nombre_campeonato, sede)  
VALUES ('camp-111', 'Campeonato Unifranz', 'El Alto'),  
('camp-222', 'Campeonato Unifranz', 'Cochabamba');
```

```
DROP TABLE campeonato; --para borrar tabla
```

```
TRUNCATE TABLE campeonato; -- para borrar datos de la tabla
```

```
INSERT INTO equipo (id_equipo, nombre_equipo, categoria, id_campeonato)  
VALUES('equ-111', 'Google', 'VARONES', 'camp-111'),  
('equ-222', '404 Not found', 'VARONES', 'camp-111'),  
('equ-333', 'girls unifranz', 'MUJERES', 'camp-111');
```

The background is a blue gradient. In the corners, there are white line art designs resembling circuit boards or neural networks, with lines and small circles connecting them.

PARA QUE DRIVE INNER JOIN.

- La instrucción INNER JOIN en SQL se utiliza para combinar filas de dos o más tablas basándose en una condición de unión específica. Este tipo de unión retorna solo las filas que tienen coincidencias en ambas tablas en función de la condición establecida. En otras palabras, solo se incluyen en el resultado aquellas filas que cumplen con la condición de unión en ambas tablas.

```
SELECT columnas  
FROM tabla1  
INNER JOIN tabla2 ON tabla1.columna = tabla2.columna;
```

```
SELECT Empleados.Nombre, Empleados.Cargo, Departamentos.Nombre AS Departamento  
FROM Empleados  
INNER JOIN Departamentos ON Empleados.DepartamentoID = Departamentos.DepartamentoID;
```



APOYÁNDONOS EN EL CONCEPTO DE CONJUNTOS
MUESTRE LOS SIGUIENTE:

2.7.1. EJEMPLO DE INNER JOIN

2.7.2. ADJUNTAR UNA IMAGEN DE CONJUNTOS Y LA
CONSULTA SQL QUE REFLEJE EL INNER JOIN



EN ESTE EJEMPLO PODEMOS VER LA COMBINACIÓN QUE REALIZA INNER JOIN

```
Select cli.Nombre, cli.Apellido, dest.NombreDestino, paq.PaqueteID, emp.NombreEmpleado, emp.ApellidoEmpleado  
From Reservas as rev  
Inner join Clientes as cli on rev.ClienteID = cli.ClienteID  
Inner join Paquetes as paq on rev.PaqueteID = paq.PaqueteID  
Inner join Destinos as dest on paq.DestinoID = dest.DestinoID  
Inner join Empleados as emp on rev.EmpleadoID = emp.EmpleadoID
```

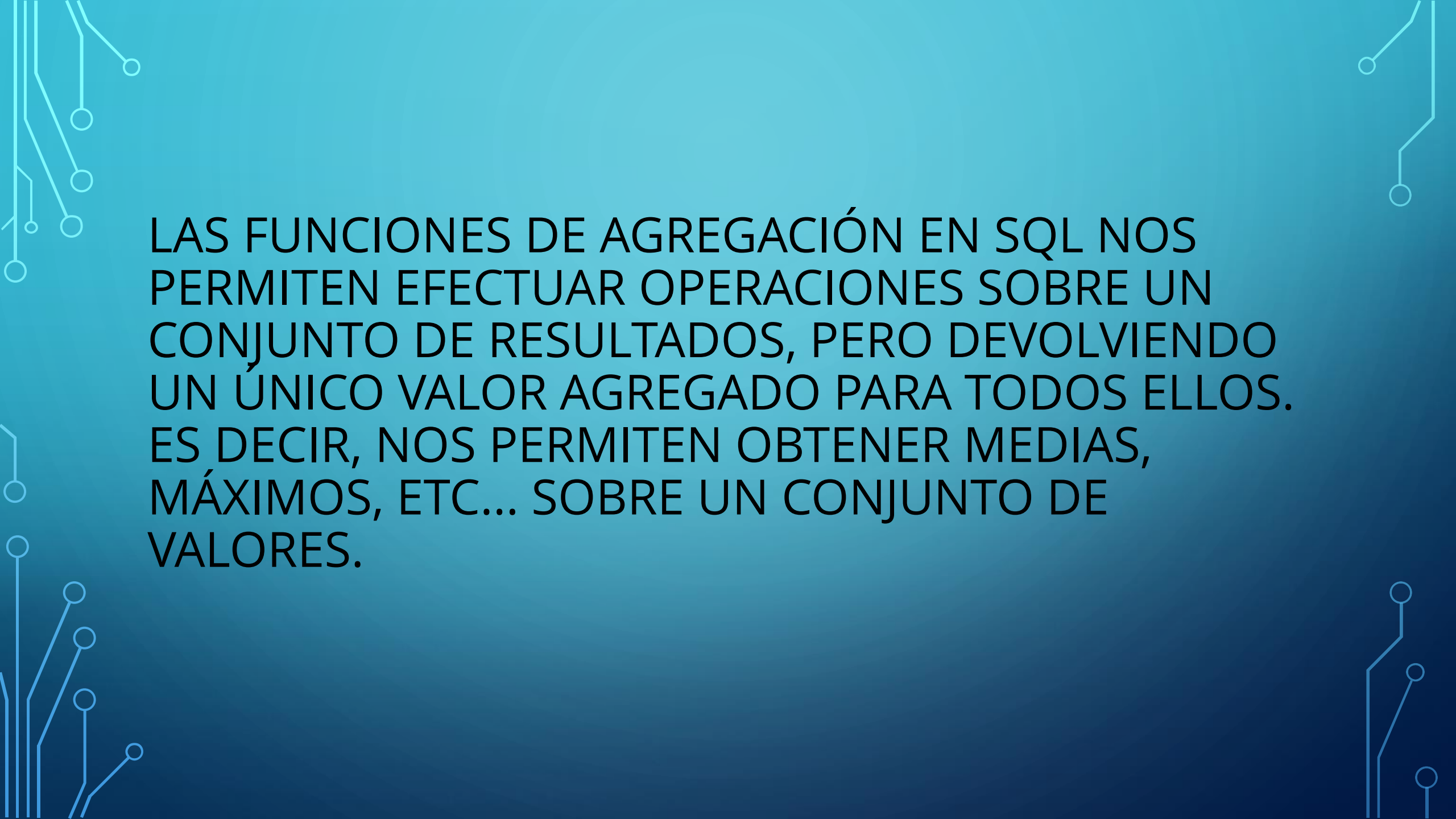
100 %

Resultados Mensajes

	Nombre	Apellido	NombreDestino	PaqueteID	NombreEmpleado	ApellidoEmpleado
1	John	Doe	Paris	1	Maria	Gomez
2	Jane	Smith	Tokio	2	Carlos	Perez
3	Bob	Jhonson	Nueva York	3	Laura	Rodriguez
4	Alice	Williams	Roma	4	Pedro	Martinez
5	Charlie	Brown	Sidney	5	Ana	Lopez

The background is a blue gradient. In the corners, there are white line art illustrations of circuit boards or neural network connections, consisting of lines and small circles.

DEFINA QUE ES UNA FUNCIÓN DE AGREGACIÓN.

The background is a blue gradient. In the corners, there are white line-art illustrations of circuit boards or neural networks, with lines connecting to small circles.

LAS FUNCIONES DE AGREGACIÓN EN SQL NOS PERMITEN EFECTUAR OPERACIONES SOBRE UN CONJUNTO DE RESULTADOS, PERO DEVOLVIENDO UN ÚNICO VALOR AGREGADO PARA TODOS ELLOS. ES DECIR, NOS PERMITEN OBTENER MEDIAS, MÁXIMOS, ETC... SOBRE UN CONJUNTO DE VALORES.

LISTE FUNCIONES DE AGREGACIÓN QUE CONOZCA

- ▶ **COUNT:** devuelve el número total de filas seleccionadas por la consulta.
- ▶ **MIN:** devuelve el valor mínimo del campo que especifiquemos.
- ▶ **MAX:** devuelve el valor máximo del campo que especifiquemos.
- ▶ **SUM:** suma los valores del campo que especifiquemos. Sólo se puede utilizar en columnas numéricas.
- ▶ **AVG:** devuelve el valor promedio del campo que especifiquemos. Sólo se puede utilizar en columnas numéricas.

PARA QUÉ SIRVE LA FUNCIÓN CONCAT EN SQL-SERVER

LA FUNCIÓN **SQL CONCAT** LE AYUDA A CONCATENAR EL CONTENIDO DE VARIAS COLUMNAS PARA OBTENER COMO RESULTADO UNA SOLA CADENA DE CARACTERES. ESTA FUNCIÓN SQL PUEDE SER ÚTIL PARA PONER LOS VALORES DE VARIAS COLUMNAS DE EXTREMO A EXTREMO PARA MOSTRAR SOLO UNA. POR LO TANTO, ES ÚTIL MOSTRAR UN RESULTADO FÁCIL DE LEER MIENTRAS SE MANTIENEN LOS DATOS EN VARIAS COLUMNAS PARA UN BUEN MANTENIMIENTO DE LOS DATOS.

EJEMPLO CONCAT:

The screenshot shows the Microsoft SQL Server Enterprise Manager interface. The 'Explorador de objetos' (Object Explorer) on the left shows the database structure. The 'SQLQuery2.sql' window in the center contains the following SQL query:

```
SELECT ID, CONCAT (Nombre, ' ', Apellido) , Correo electrónico,
AddressLine, Ciudad, Cargo
FROM Empleado
```

The 'Resultados' (Results) pane at the bottom displays the output of the query as a table with 4 rows. The first row is highlighted in yellow.

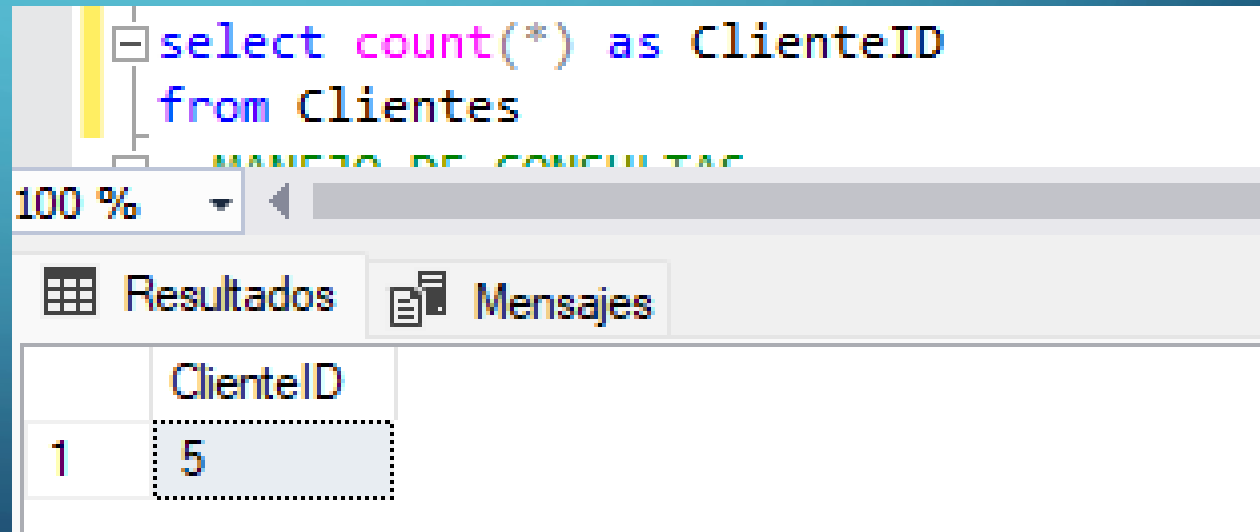
	ID	(Sin nombre de columna)	Correo electrónico	AddressLine	Ciudad	Cargo
1	1	Carlos Rodriguez	esaff@hotmail.com	edl	santiago	administrador
2	2	Jose Fernandez	pesa@hotmail.com	erf	mexico	Gerente de tienda
3	3	Pedro Acosta	pedr@gmail.com	ssss	Penique	Vendedor
4	4	David Rodriguez0	NULL	sess	mexico	vendedor

At the bottom of the window, a status bar indicates: 'Consulta ejecutada cor... | DESKTOP-UAAE5VQ\SQLEXPRESS ... | DESKTOP-UAAE5VQ\Admin ... | ejemplo | 00:00:00 | 4 filas'.

EJEMPLO COUNT:

```
create table Clientes
(
  ClienteID Integer Primary Key not null,
  Nombre    Varchar (50) not null,
  Apellido  Varchar (50) not null,
  Email     Varchar (50) not null,
  Telefono  Varchar (20)
);

--Damos los datos y valores a las relaciones de la tabla--
Insert Into Clientes (ClienteID, Nombre, Apellido, Email, Telefono)
Values (1, 'John', 'Doe', 'john.doe@example.com', '555-1234'),
      (2, 'Jane', 'Smith', 'jane.smith@example.com', '555-5678'),
      (3, 'Bob', 'Jhonson', 'bob.jhonson@example.com', '555-9876'),
      (4, 'Alice', 'Williams', 'alice.williams@example.com', '555-4321'),
      (5, 'Charlie', 'Brown', 'charlie.brown@example.com', '555-8765');
```



The screenshot shows a database query window with the following SQL query:

```
select count(*) as ClienteID
from Clientes
```

Below the query, there is a tab labeled "Resultados" (Results) which is active. The results are displayed in a table with one column, "ClienteID", and one row with the value "5".

	ClienteID
1	5

EJEMPLO AVG:

```
-- create table Destinos
(
  DestinoID      Integer Primary Key not null,
  NombreDestino  Varchar (100) not null,
  Descripcion    Text not null,
  PrecioBase     Numeric(10,2) not null
);

SELECT AVG(PrecioBase)
FROM Destinos

--Damos los datos y valores a las relaciones de la tabla--

Insert Into Destinos (DestinoID, NombreDestino, Descripcion, PrecioBase)
Values (1, 'París', 'Ciudad del amor y la luz', 1200.00),
      (2, 'Tokio', 'Metrópolis moderna y vibrante', 1500.00),
      (3, 'Nueva York', 'La ciudad que nunca duerme', 1800.00),
      (4, 'Roma', 'Cuna de la civilización antigua', 1400.00),
      (5, 'Sidney', 'Puerta de entrada a Australia', 1600.00);
```

00 %



Resultados



Mensajes

(Sin nombre de columna)

1

1500.000000

EJEMPLO MUESTRA UN EJEMPLO DEL USO DE MIN-MAX:

```
PrecioTotal Numeric(10,2) not null,  
Foreign Key (DestinoID) References Destinos (DestinoID)  
);
```

--Damos los datos y valores a las relaciones de la tabla--

```
Insert Into Paquetes (PaqueteID, NombrePaquete, Descripcion, FechaInicio, FechaFin, DestinoID, PrecioTotal)  
Values (1, 'Escapada Romantica a París', 'Disfruta de la comida', '2023-01-15', '2023-01-20', 1, 1500.00),  
(2, 'Aventura en Tokio', 'Descubre la cultura', '2023-02-01', '2023-02-10', 2, 2000.00),  
(3, 'Explora New York', 'Recorre los lugares mas increíbles', '2023-03-10', '2023-03-18', 3, 2200.00),  
(4, 'Historia y Arte en Roma', 'Sumergete en la aventura', '2023-04-05', '2023-04-12', 4, 1800.00),  
(5, 'Aventura en Australia', 'Descubre la belleza', '2023-05-20', '2023-05-28', 5, 2500.00);
```

```
SELECT MIN (PrecioTotal )  
FROM Paquetes  
SELECT MAX (PrecioTotal )  
FROM Paquetes
```

Resultados Mensajes

(Sin nombre de columna)

1500.00

(Sin nombre de columna)

2500.00