
Procesual Hito 4 Base de Datos I - 2023
Bases de Datos - SQL Server
by Jamil Alex Quispe Valencia
1 0 1 1 0 1 1 0 1 1 0 0 1 1 0 1 1 0 1 1 0 1 1 1 0 1 1 0 1 1 1 0 1

</ Consigna Y Detalle del problema

- Diseñe un sistema de Base de Datos Relacional utilizando el gestor de Base de Datos SQL Server teniendo como premisa el uso de buenas prácticas en diseño de la base de datos aplicados al siguiente escenario. Una comunidad de estudiantes de la nación UNIFRANZ están desarrollando un sistema para una agencia de viaje de nombre: LADITA SRL. En una de las fases de desarrollo se estableció como objetivo la definición de la base de datos relacional. Por consiguiente se deberá plantear el diseño de la base de datos para la agencia de viajes.

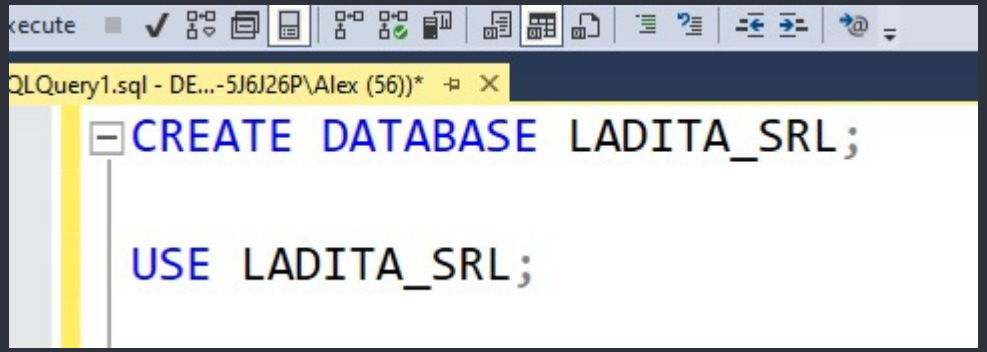
  Se tiene como contexto una AGENCIA DE VIAJES en el cual se tiene como objetivo tener mínimamente 5 entidades principales en esta primera versión. En tal sentido se deberá crear las siguientes tablas.

  - Clientes
  - Destinos
  - Paquetes
  - Reservas
  - Empleados

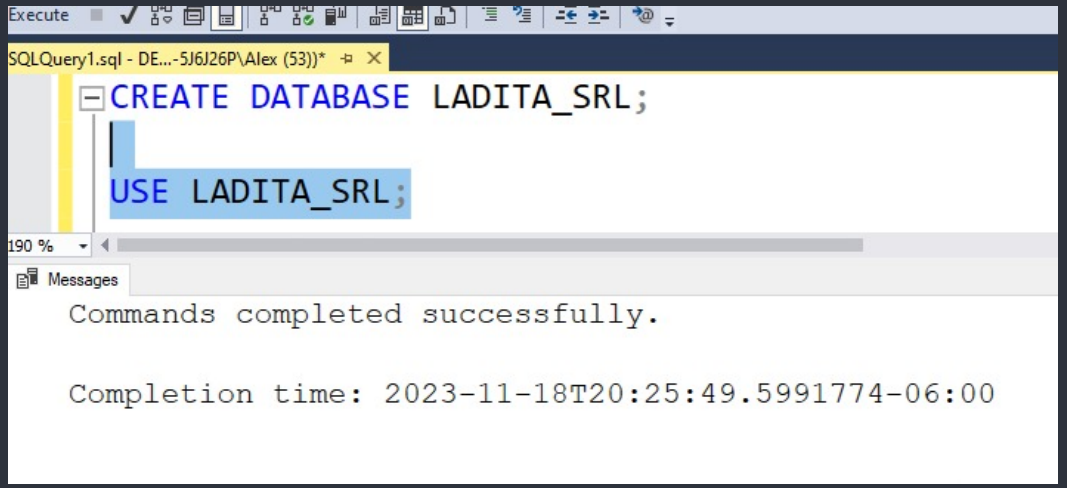
1 0 1 1 0 1 1 0 1 1 0 1 1 0 0 1 1 0 1 1 0 1 1 0 1 1 0 1 1 1 0 1 1 0 1 1 0 1 1 1 1 1 0 1

</ I. - Diseñamos nuestra propia base de datos con los Detalles y consignas asignadas:

Creamos una base de datos con el nombre “LADITA_SRL”



Selección de la base de datos recién creada: la base de datos “LADITA_SRL”



Creamos la tabla Clientes .

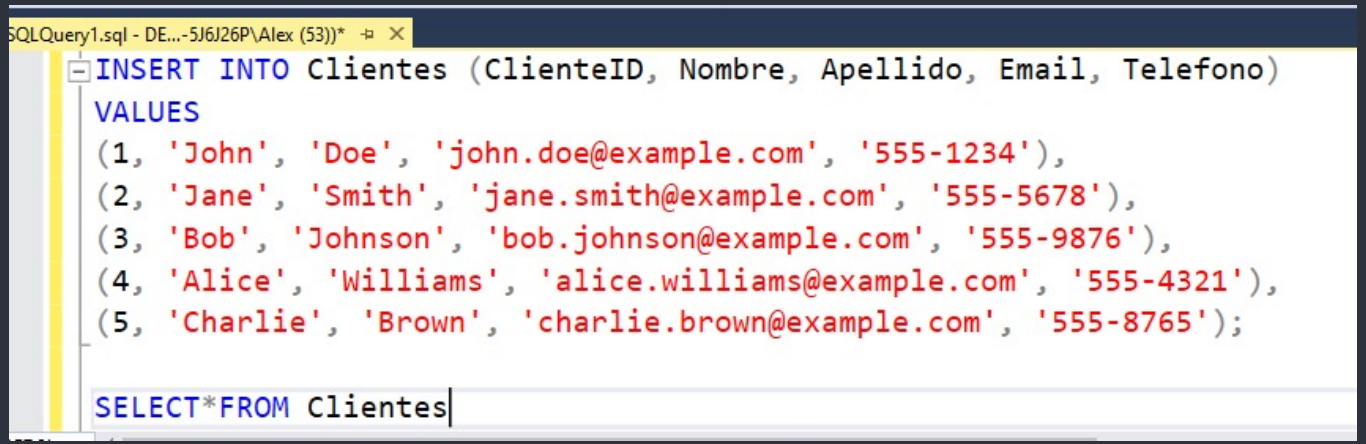
SQLQuery1.sql - DE...-5J6J26P\Alex (53))*

```
CREATE TABLE Clientes (
    ClienteID INT PRIMARY KEY,
    Nombre VARCHAR(50),
    Apellido VARCHAR(50),
    Email VARCHAR(50),
    Telefono VARCHAR(20)
);
```

157 %

Diseño a replicar de la tabla

Insertamos datos en la tabla Clientes :

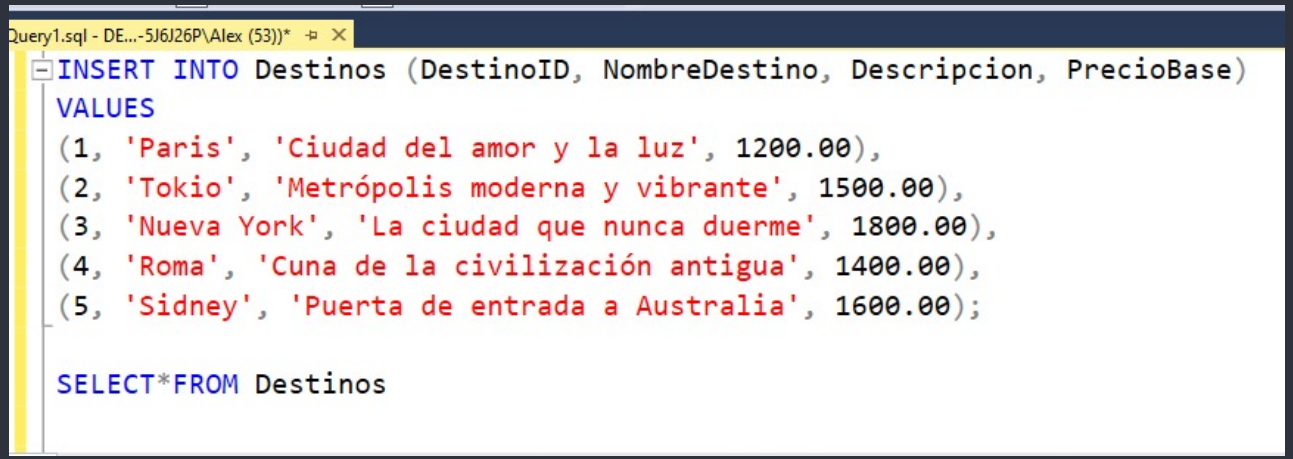


Creamos la tabla Destinos:

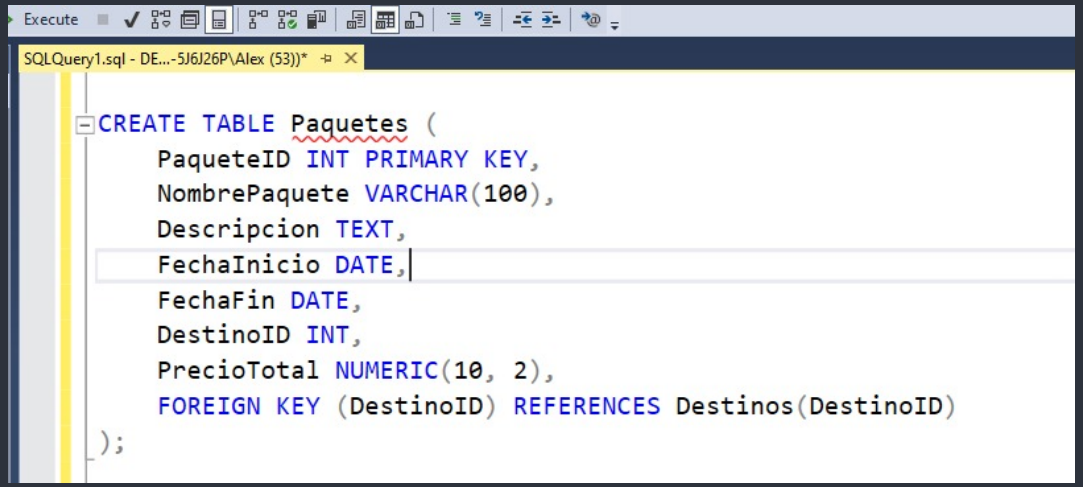
```sql
CREATE TABLE Destinos (
    DestinoID INT PRIMARY KEY,
    NombreDestino VARCHAR(100),
    Descripcion TEXT,
    PrecioBase  NUMERIC(10, 2)
);
```

Diseño a replicar de la tabla

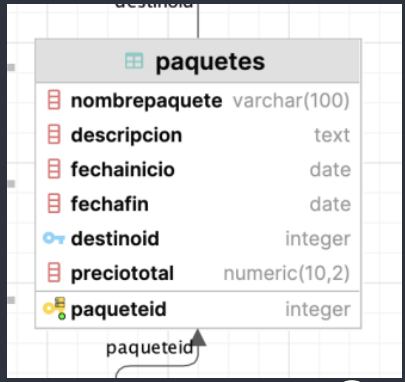
Insertamos datos en la tabla Destinos:



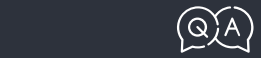
Creamos la tabla Paquetes:



Diseño a replicar de la tabla



Insertamos en la tabla Paquetes:



```
INSERT INTO Paquetes (PaqueteID, NombrePaquete, Descripcion, FechaInicio, FechaFin, DestinoID, PrecioTotal)
VALUES
(1, 'Escapada Romantica a Paris', 'Disfruta de la c_', '2023-01-15', '2023-01-20', 1, 1500.00),
(2, 'Aventura en Tokio', 'Descubre la cult_', '2023-02-01', '2023-03-10', 2, 2000.00),
(3, 'Explora Nueva York', 'Recorre los Luga_', '2023-03-10', '2023-03-18', 3, 2200.00),
(4, 'Historia y Arte en Roma', 'Sumérgete en la _', '2023-04-05', '2023-04-12', 4, 1800.00),
(5, 'Aventura en Australia', 'Descubre la bell_', '2023-05-20', '2023-05-28', 5, 2500.00);

SELECT*FROM Paquetes
```

Creamos la tabla Reservas:

```sql
CREATE TABLE Reservas (
    ReservaID INT PRIMARY KEY,
    ClienteID INT,
    PaqueteID INT,
    FechaReserva DATE,
    CantidadPersonas INT,
    PrecioTotal NUMERIC(10, 2),
    EstadoReserva VARCHAR(20),
    FOREIGN KEY (ClienteID) REFERENCES Clientes(ClienteID),
    FOREIGN KEY (PaqueteID) REFERENCES Paquetes(PaqueteID)
);
```

Diseño a replicar de la tabla

Insertamos datos en la tabla Reservas:

```
);

INSERT INTO Reservas (ReservaID, ClienteID, PaqueteID, FechaReserva, CantidadPersonas, PrecioTotal, EstadoReserva)
VALUES
(1, 1, 1, '2023-01-05', 2, 3000.00, 'Confirmada'),
(2, 2, 3, '2023-02-12', 1, 2200.00, 'Pendiente'),
(3, 3, 2, '2023-03-18', 3, 6000.00, 'Confirmada'),
(4, 4, 4, '2023-04-10', 2, 3600.00, 'Pendiente'),
(5, 5, 5, '2023-05-25', 1, 2500.00, 'Confirmada');

SELECT*FROM Reservas
```

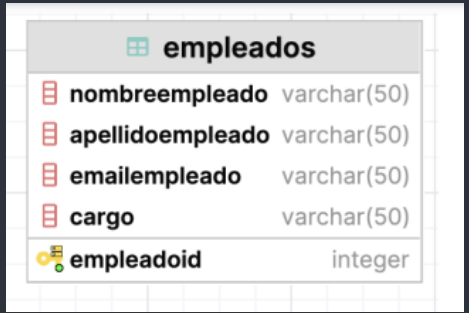
Creamos la tabla Empleados:

```
CREATE TABLE Empleados (
    EmpleadoID INT PRIMARY KEY,
    NombreEmpleado VARCHAR(50),
    ApellidoEmpleado VARCHAR(50),
    EmailEmpleado VARCHAR(50),
    Cargo VARCHAR(50)
);
```

```
Messages
Commands completed successfully.

Completion time: 2023-11-18T21:59:58.2758319-06:00
```

Diseño a replicar de la tabla



Insertamos datos en la tabla Empleados:

```
INSERT INTO Empleados (EmpleadoID, NombreEmpleado, ApellidoEmpleado, EmailEmpleado, Cargo)
VALUES
(1, 'Maria', 'Gomez', 'maria.gomez@example.com', 'Agente de Ventas'),
(2, 'Carlos', 'Perez', 'carlos.perez@example.com', 'Asesor de Viajes'),
(3, 'Laura', 'Rodriguez', 'laura.rodriguez@example.com', 'Gerente de Sucursal'),
(4, 'Pedro', 'Martinez', 'pedro.martinez@example.com', 'Asistente Administrativo'),
(5, 'Ana', 'Lopez', 'ana.Lopez@example.com', 'Asistente Administrativo Especialista en Destinos');
SELECT*FROM Empleados
```

</ Con el comando SELECT*FROM mostramos todas las tablas creadas

```
SELECT*FROM Clientes;
SELECT*FROM Destinos;
SELECT*FROM Paquetes;
SELECT*FROM Reservas;
SELECT*FROM Empleados;
```

</ Comparamos si los datos ingresados son los mismos que deben estar ingresados. TABLA Clientes

143 %

Results | Messages

	ClienteID	Nombre	Apellido	Email	Telefono
1	1	John	Doe	john.doe@example.com	555-1234
2	2	Jane	Smith	jane.smith@example.com	555-5678
3	3	Bob	Johnson	bob.johnson@example.com	555-9876
4	4	Alice	Williams	alice.williams@example.com	555-4321
5	5	Charlie	Brown	charlie.brown@example.com	555-8765

tabla clientes

WHERE | ORDER BY

	clienteid	nombre	apellido	email	telefono
1	1	John	Doe	john.doe@example.com	555-1234
2	2	Jane	Smith	jane.smith@example.com	555-5678
3	3	Bob	Johnson	bob.johnson@example.com	555-9876
4	4	Alice	Williams	alice.williams@example.com	555-4321
5	5	Charlie	Brown	charlie.brown@example.com	555-8765

</ Comparamos si los datos ingresados son los mismos que deben estar ingresados. TABLA Destinos

Results | Messages

	DestinoID	NombreDestino	Descripcion	PrecioBase
1	1	Paris	Ciudad del amor y la luz	1200.00
2	2	Tokio	Metrópolis moderna y vibrante	1500.00
3	3	Nueva York	La ciudad que nunca duerme	1800.00
4	4	Roma	Cuna de la civilización antig...	1400.00
5	5	Sidney	Puerta de entrada a Australia	1600.00

tabla destinos

	destinoid	nombredestino	descripcion	preciobase
1	1	París	Ciudad del amor y la luz	1200.00
2	2	Tokio	Metrópolis moderna y vibrante	1500.00
3	3	Nueva York	La ciudad que nunca duerme	1800.00
4	4	Roma	Cuna de la civilización antigua	1400.00
5	5	Sidney	Puerta de entrada a Australia	1600.00

</ Comparamos si los datos ingresados son los mismos que deben estar ingresados. TABLA Paquetes

	PaqueteID	NombrePaquete	Descripcion	FechaInicio	FechaFin	DestinoID	PrecioTotal
1	1	Escapada Romantica a Paris	Disfruta de la c_	2023-01-15	2023-01-20	1	1500.00
2	2	Aventura en Tokio	Descubre la cult_	2023-02-01	2023-03-10	2	2000.00
3	3	Explora Nueva York	Recorre los Lug...	2023-03-10	2023-03-18	3	2200.00
4	4	Historia y Arte en Roma	Sumérgete en l...	2023-04-05	2023-04-12	4	1800.00
5	5	Aventura en Australia	Descubre la bell_	2023-05-20	2023-05-28	5	2500.00

tabla paquetes

	paqueteid	nombrepaquete	descripcion	fechainicio	fechafin	destinoid	preciototal
1	1	Escapada Romántica a París	Disfruta de la c…	2023-01-15	2023-01-20	1	1500.00
2	2	Aventura en Tokio	Descubre la cult…	2023-02-01	2023-02-10	2	2000.00
3	3	Explora Nueva York	Recorre los luga…	2023-03-10	2023-03-18	3	2200.00
4	4	Historia y Arte en Roma	Sumérgete en la …	2023-04-05	2023-04-12	4	1800.00
5	5	Aventura en Australia	Descubre la bell…	2023-05-20	2023-05-28	5	2500.00

</ Comparamos si los datos ingresados son los mismos que deben estar ingresados. TABLA Reservas

	ReservaID	ClienteID	PaqueteID	FechaReserva	CantidadPersonas	PrecioTotal	EstadoReserva
1	1	1	1	2023-01-05	2	3000.00	Confirmada
2	2	2	3	2023-02-12	1	2200.00	Pendiente
3	3	3	2	2023-03-18	3	6000.00	Confirmada
4	4	4	4	2023-04-10	2	3600.00	Pendiente
5	5	5	5	2023-05-25	1	2500.00	Confirmada

tabla reservas

	reservaid	clienteid	paqueteid	fechareserva	cantidadpersonas	preciototal	estadoreserva
1	1	1	1	2023-01-05	2	3000.00	Confirmada
2	2	2	3	2023-02-12	1	2200.00	Pendiente
3	3	3	2	2023-03-18	3	6000.00	Confirmada
4	4	4	4	2023-04-10	2	3600.00	Pendiente
5	5	5	5	2023-05-25	1	2500.00	Confirmada

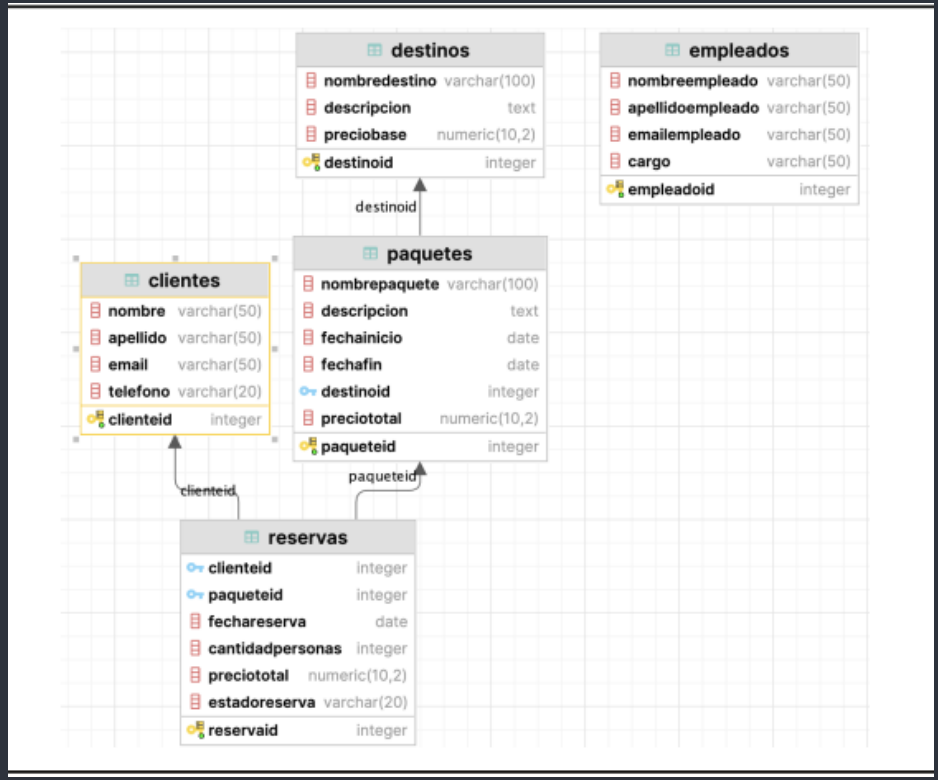
</ Comparamos si los datos ingresados son los mismos que deben estar ingresados. TABLA Empleados

	EmpleadoID	NombreEmpleado	ApellidoEmpleado	EmailEmpleado	Cargo
1	1	Maria	Gomez	maria.gomez@example.com	Agente de Ventas
2	2	Carlos	Perez	carlos.perez@example.com	Asesor de Viajes
3	3	Laura	Rodriguez	laura.rodriguez@example....	Gerente de Suc...
4	4	Pedro	Martinez	pedro.martinez@example....	Asistente Admini...
5	5	Ana	Lopez	ana.Lopez@example.com	Asistente Admini...

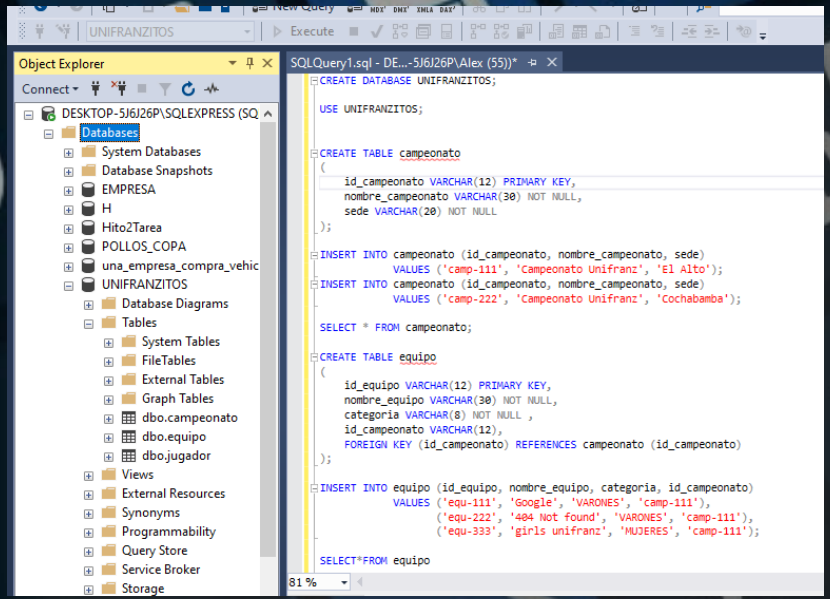
tabla empleados

	empleadoid	nombreempleado	apellidoempleado	emailempleado	cargo
1	1	Maria	Gomez	maria.gomez@example.com	Agente de Ventas
2	2	Carlos	Perez	carlos.perez@example.com	Asesor de Viajes
3	3	Laura	Rodriguez	laura.rodriguez@example.com	Gerente de Sucursal
4	4	Pedro	Martinez	pedro.martinez@example.com	Asistente Administrativo
5	5	Ana	Lopez	ana.lopez@example.com	Especialista en Destinos

</ Hacemos una vista generala las tablas



</ II. - Manejo de conceptos



</ 2.1. Muestra un ejemplo de DDL.

¿Que es DDL ? Los comandos DDL (Data Definition Language o Lenguaje de Definición de Datos) en SQL se utilizan para definir, crear, modificar y eliminar la estructura de la base de datos.

```
CREATE TABLE Destinos (
    DestinoID INT PRIMARY KEY,
    NombreDestino VARCHAR(100),
    Descripcion TEXT,
    PrecioBase  NUMERIC(10, 2)
);
```

</ 2.2. Muestra un ejemplo de DML.

```sql
INSERT INTO Clientes (ClienteID, Nombre, Apellido, Email, Telefono)
VALUES (6, 'Elena', 'Gonzalez', 'elena.gonzalez@example.com', '555-2468');
```

</ 2.3. Para que sirve INNER JOIN.

El INER JOIN es un tipo de operación de combinación en SQL que combina registros de dos tablas basados en una condición especificada en la cláusula ON.

En el contexto de bases de datos relacionales, el INNER JOIN une filas de dos tablas en función de una condición de igualdad entre columnas de ambas tablas. Solo devuelve filas que tienen valores coincidentes en ambas tablas, es decir, aquellas filas que cumplen la condición de unión especificada.

Por ejemplo, si tienes dos tablas: Clientes y Reservas con columnas comunes como ClienteID, puedes utilizar INNER JOIN para combinar la información de estas tablas basándote en el ClienteID

</ 2.4. Defina que es una función de agregación.

Una función de agregación en SQL es una función que opera en un conjunto de valores para devolver un solo valor que resuma los datos de ese conjunto. Estas funciones realizan cálculos en múltiples filas de una tabla y devuelven un único resultado, lo que las hace muy útiles para resumir, analizar o presentar información en consultas SQL.

</ 2.5. Liste funciones de agregación que conozca.

Algunas de las funciones de agregación más comunes incluyen:

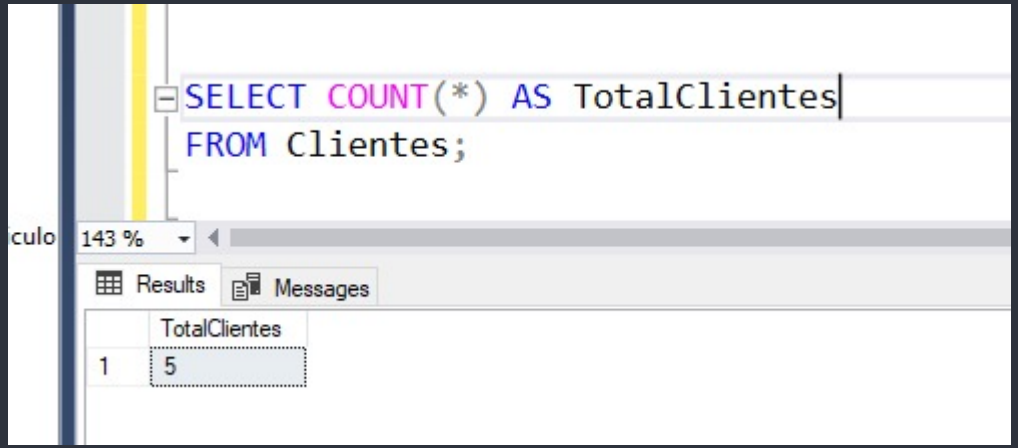
SUM: Calcula la suma de los valores en una columna.
AVG: Calcula el promedio de los valores en una columna numérica.
COUNT: Cuenta el número de filas o valores distintos en una columna.
MAX: Devuelve el valor máximo de una columna.
MIN: Devuelve el valor mínimo de una columna.
Estas funciones se utilizan con la cláusula SELECT para realizar cálculos en los datos y resumir la información según sea necesario. Por ejemplo, podrías usar SUM para encontrar la suma total de los ingresos de una empresa, o COUNT para contar el número de clientes en una base de datos

</ 2.6. Para qué sirve la función CONCAT en SQL-Server

La función CONCAT en SQL Server se utiliza para concatenar o unir cadenas de texto provenientes de una o más columnas o valores. Básicamente, concatena dos o más cadenas para formar una sola cadena más larga.

</ 2.7.Muestra un ejemplo del uso de COUNT

Supongamos que queremos contar cuántos clientes hay en nuestra tabla Clientes en la base de datos LADITA_DB. Podemos usar la función COUNT para realizar esta operación:



</2.8. Muestra un ejemplo del usos de AVG

Si deseamos calcular el promedio de la edad de los clientes en la tabla Clientes de la base de datos LADITA_DB, podríamos utilizar la función AVG. Supongamos que la tabla Clientes tiene una columna llamada Edad. Aquí está cómo podríamos calcular ese promedio:

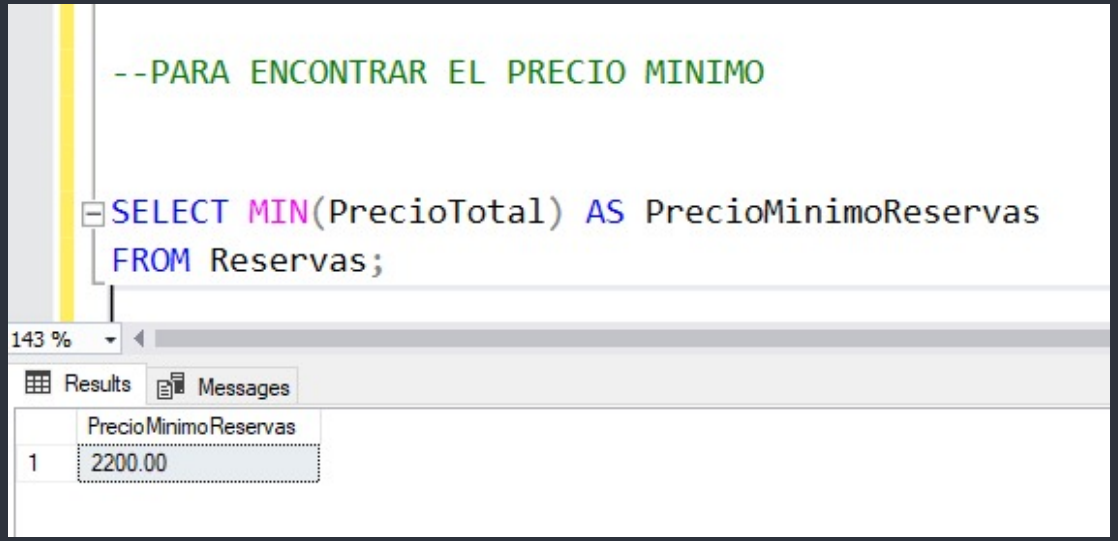
```
--EJEMPLO DE AVG


SELECT AVG(Edad) AS PromedioEdad
FROM Clientes;
```

</ 2.9. Muestra un ejemplo del uso de MIN

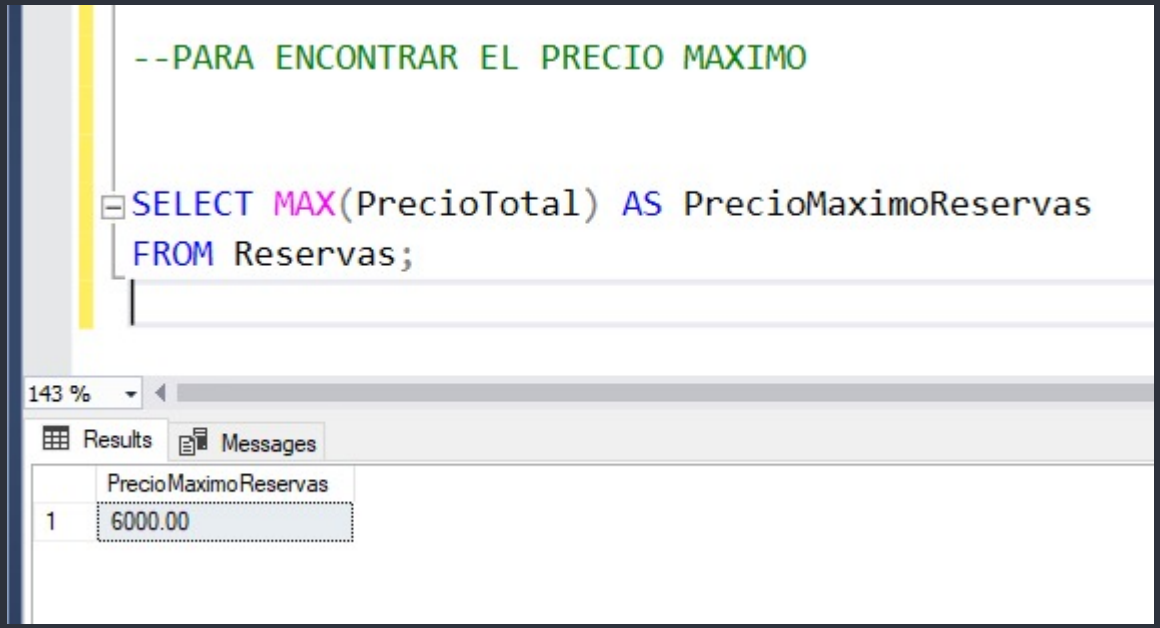
Veamos cómo encontrar el precio mínimo y máximo de estas reservas:

Para encontrar el precio mínimo:



</ 2.9. Muestra un ejemplo del uso de MAX

Y para encontrar el precio máximo:



3. Manejo de consultas

3.1. ¿Cuáles son los empleados que tienen el título "Agente de Ventas"?

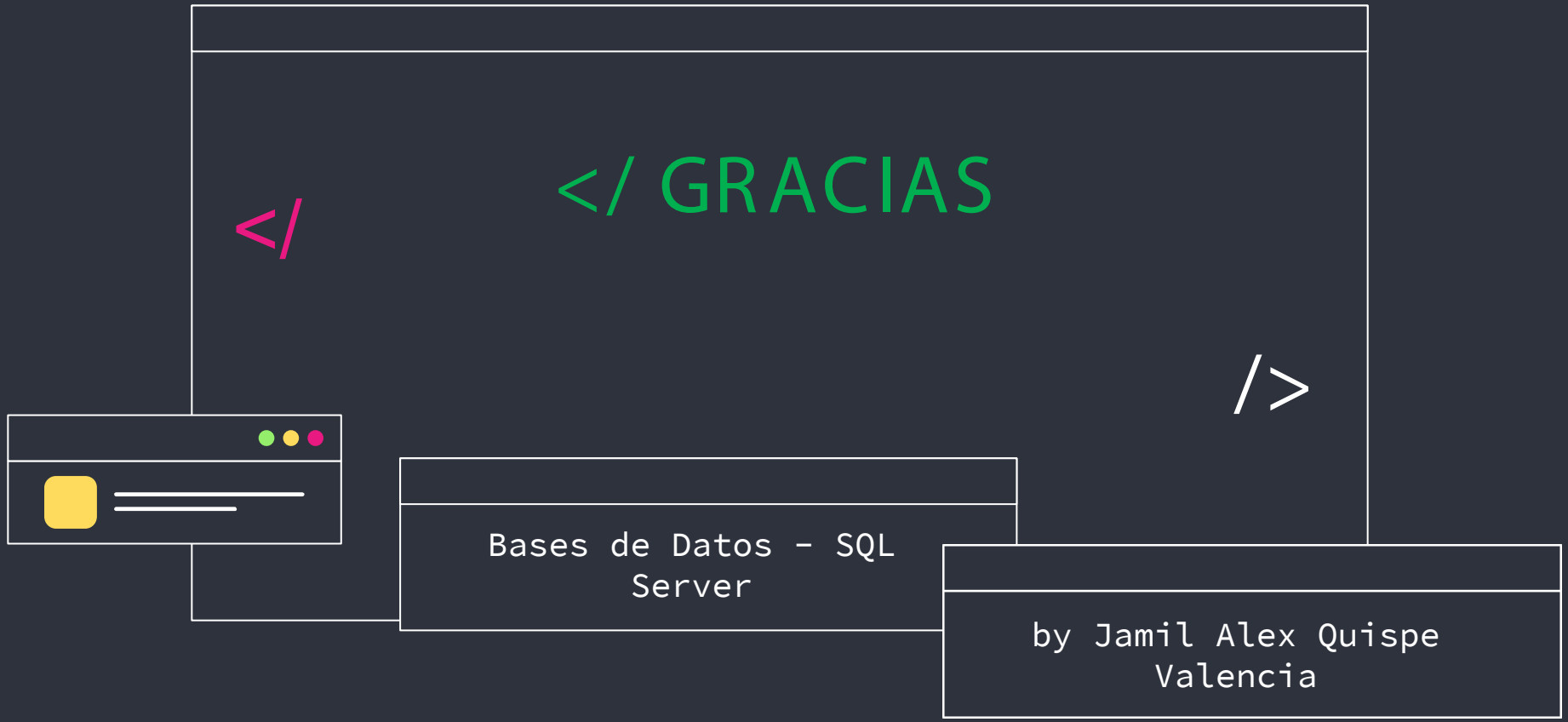
3.2. ¿Cuáles son los destinos de los paquetes turísticos reservados por el cliente con ID 1?

3.3. ¿Cuáles son los paquetes turísticos reservados por el cliente llamado "Jane Smith"?

3.4. ¿Cuáles son los nombres de los clientes, los destinos de sus paquetes
turísticos y los nombres de los empleados que gestionaron esas reservas?

3.5. ¿Cuál es la cantidad total de personas que han reservado paquetes turísticos en la agencia?

3.6. ¿Cuántas reservas se han realizado para el paquete turístico llamado "Aventura en Tokio"?


</ GRACIAS
</
/>
Bases de Datos - SQL Server
by Jamil Alex Quispe Valencia
1 0 1 1 0 1 1 0 1 1 0 0 1 1 0 1 1 0 1 1 0 1 1 0 1 1 0 1 1 0 1 1 1 0 1