

**Consigna**

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

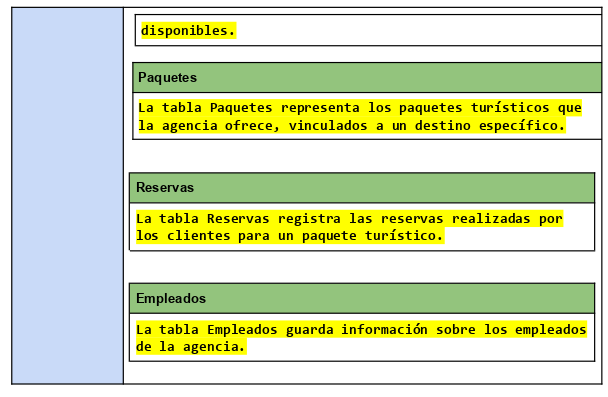
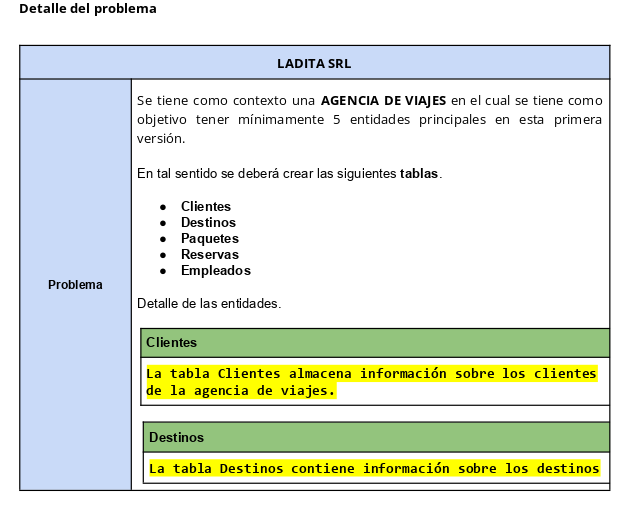
aplicados al siguiente escenario.

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.



**1. Diseño de base de datos (modelo lógico).**

**1.1. Dado el detalle explicado en la parte inicial de este documento debería generar una base de datos similar al siguiente:**



**Codigo de solucion del problema:**

create database Agencia\_de\_viaje\_Ladita\_SRL

Use Agencia\_de\_viaje\_Ladita\_SRL

Create table empleados(

nombreempleado varchar(50) not null,

apellidoempleado varchar(50) not null,

emailempleado varchar(50) not null,

cargo varchar(50) not null,

empleadoid int primary key identity(1,1)

);

create table destinos(

nombredestino varchar(100) not null,

descripcion text not null,

preciobase numeric(10,2) not null,

destinoid int primary key identity(1,1)

);

create table paquetes(

nombrepaquete varchar (100) not null,

descripcion text not null,

fechainicio date not null,

fechafin date not null,

preciototal numeric(10,2) not null,

paqueteid int primary key identity(1,1),

destinoid int,

foreign key (destinoid) references destinos(destinoid)

);

create table clientes (

nombre varchar(50) not null,

apellido varchar(50) not null,

email varchar(50) not null,

telefono varchar(20) not null,

clienteid int primary key identity(1,1)

);

create table reservas(

fechareserva date not null,

cantidadpersonas int not null,

preciototal numeric(10,2) not null,

estadoreserva varchar(20) not null,

reservaid int primary key identity (1,1),

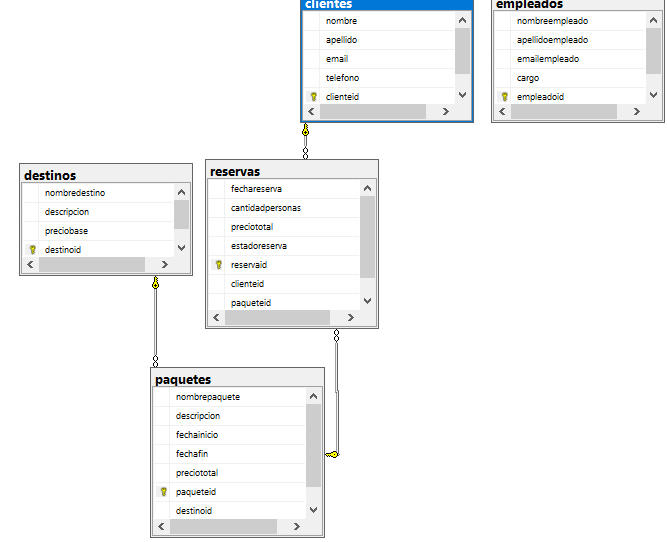
clienteid int,

paqueteid int,

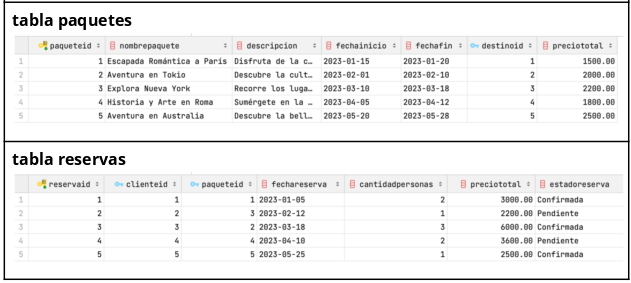
foreign key (clienteid) references clientes(clienteid),

foreign key (paqueteid) references paquetes(paqueteid)

);



**1.2. Los registros de cada tabla deberían quedar de la siguiente forma**



**Código de la inserción de registros:**

insert into clientes(nombre,apellido,email,telefono)

values ('John', 'Doe', 'john.doe@example.com', '555-1234'),

('Jane', 'Smith', 'jane.smith@example.com', '555-5678'),

('Bob', 'Johnson', 'bob.johnson@example.com', '555-9876'),

('Alice', 'Williams', 'alice.williams@example.com', '555-4321'),

('Charlie', 'Brown', 'charlie.brown@example.com', '555-8765')

insert into destinos(nombredestino,descripcion,preciobase)

values('Paris', 'Ciudad del amor y la luz', 1200.00),

('Tokyo', 'Metropolis moderna y vibrante', 1500.00),

('Nueva York', 'La ciudad que nunca duerme', 1800.00),

('Roma', 'Cuna de civilizaciones antigua', 1400.00),

('Sidney', 'Puerta de entrada a Australia', 1600.00)

insert into empleados(nombreempleado,apellidoempleado,emailempleado,cargo)

values ('Maria', 'Gomez', 'maria.gomez@example.com', 'Agente de Ventas'),

('Carlos', 'Perez', 'carlos.perez@example.com', 'Acesor de Viajes'),

('Laura', 'Rodriguez', 'laura.rodi¡riguez@example.com', 'Gerente de Sucursal'),

('Pedro', 'Martinez', 'pedro.martinez@example.com', 'Asistente Administrativo'),

('Ana', 'Lopez', 'ana.lopez@example.com', 'Especialistas en Destinos')

insert into paquetes(nombrepaquete,descripcion,fechainicio,fechafin,destinoid,preciototal)

values ('Escapada romantica a Paris', 'Disfruta de la c...', '2023-01-15', '2023-01-20', 1, 1500.00),

('Aventura en Tokyo', 'Descubre la cult...', '2023-02-01', '2023-02-10', 2, 2000.00),

('Explora Nueva York', 'Recorre los luga...', '2023-03-10', '2023-03-18', 3, 2200.00),

('Historia y Arte en Roma', 'Sumergete en la ...', '2023-04-05', '2023-04-12', 4, 1800.00),

('Aventura en Australia', 'Descubre la bell...', '2023-05-20', '2023-05-28', 5, 2500.00)

insert into reservas(clienteid, paqueteid, fechareserva, cantidadpersonas, preciototal, estadoreserva)

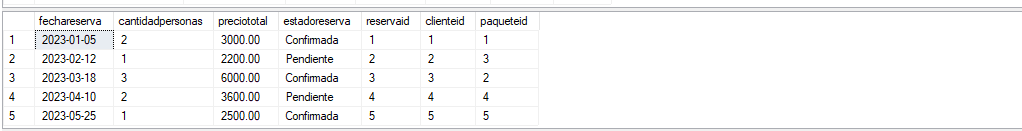
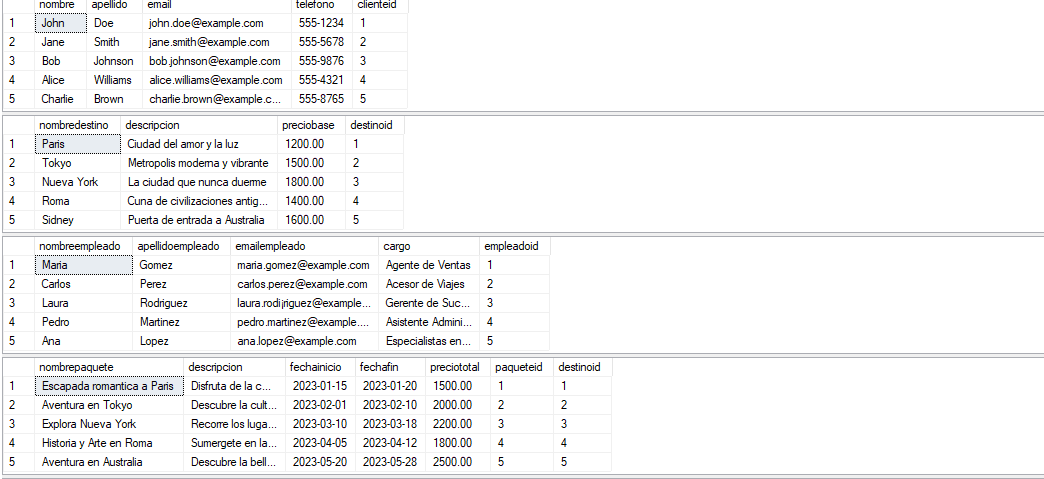
values (1, 1, '2023-01-05', 2, 3000.00, 'Confirmada'),

(2, 3, '2023-02-12', 1, 2200.00, 'Pendiente'),

(3, 2, '2023-03-18', 3, 6000.00, 'Confirmada'),

(4, 4, '2023-04-10', 2, 3600.00, 'Pendiente'),

(5, 5, '2023-05-25', 1, 2500.00, 'Confirmada')



**Posdata:** Las primary key de cada tabla fueron creadas con el identity(1,1).

**2. Manejo de conceptos**

**2.1. Ejemplo de DDL (Data Definition Language):**

DDL se utiliza para definir la estructura de la base de datos. Un ejemplo común es la creación de una tabla. Aquí tienes un ejemplo en SQL:

CREATE TABLE Empleados (

ID INT PRIMARY KEY,

Nombre VARCHAR(50),

Edad INT,

Departamento VARCHAR(30)

);

Este código crea una tabla llamada "Empleados" con columnas para ID, Nombre, Edad y Departamento.

**2.2. Ejemplo de DML (Data Manipulation Language):**

DML se utiliza para manipular los datos almacenados en la base de datos. Un ejemplo común es la inserción de datos en una tabla. Aquí tienes un ejemplo en SQL:

-- Actualizar el precio base de un destino

UPDATE destinos

SET preciobase = 1200.50

WHERE destinoid = 1;

Este código actualiza el precio base del destino con ID 1 en la tabla "destinos".

**2.3. Uso de INNER JOIN:**

INNER JOIN se utiliza para combinar registros de dos o más tablas basándose en una condición de coincidencia. Por ejemplo:

-- Obtener la información de un paquete y su destino correspondiente

SELECT paquetes.nombrepaquete, destinos.nombredestino

FROM paquetes

INNER JOIN destinos ON paquetes.destinoid = destinos.destinoid

WHERE paquetes.paqueteid = 1;

Este código utiliza INNER JOIN para obtener información de un paquete y el destino correspondiente..

**2.4. Definición de función de agregación:**

Una función de agregación realiza un cálculo en un conjunto de valores y devuelve un solo valor. Por ejemplo, calcular el promedio, la suma, el mínimo o el máximo de un conjunto de datos.

-- Calcular el total de reservas

SELECT COUNT(reservaid) AS TotalReservas

FROM reservas;

**2.5. Funciones de agregación comunes:**

- COUNT: Cuenta el número de filas en un conjunto de resultados.

- SUM: Calcula la suma de los valores en una columna.

- AVG: Calcula el promedio de los valores en una columna.

- MIN: Encuentra el valor mínimo en una columna.

- MAX: Encuentra el valor máximo en una columna.

**2.6. Uso de la función CONCAT en SQL Server:**

La función CONCAT se utiliza para concatenar (unir) dos o más cadenas de texto en SQL Server. Por ejemplo:

-- Crear una lista de clientes con su nombre completo

SELECT CONCAT(nombre, ' ', apellido) AS NombreCompleto

FROM clientes;

Este código concatena el nombre y el apellido de los clientes.

**2.7. Ejemplo de uso de COUNT:**

-- Contar la cantidad de reservas para un paquete específico

SELECT COUNT(reservaid) AS ReservasParaPaquete1

FROM reservas

Este código devuelve el número de reservas en la tabla "reservas".

**2.8. Ejemplo de uso de AVG:**

-- Calcular el promedio de la cantidad de personas en las reservas

SELECT AVG(reservaid)

FROM reservas;

Este código calcula el promedio de la cantidad de personas en la tabla "reservas".

**2.9. Ejemplo de uso de MIN-MAX:**

-- Encontrar la fecha de inicio más temprana y más tardía de los paquetes

SELECT MIN(fechainicio) AS FechaInicioMinima, MAX(fechafin) AS FechaFinMaxima

FROM paquetes;

**3. Manejo de consultas**

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

SELECT\*

FROM empleados

WHERE cargo = 'Agente de Ventas';

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

SELECT dest.nombredestino As Destino

FROM reservas as res

join paquetes as paq on res.paqueteid = paq.paqueteid

join destinos as dest on paq.destinoid = dest.destinoid

WHERE res.clienteid = 1;

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

SELECT paq.nombrepaquete as Paquete\_Adquirido

FROM reservas as res

JOIN paquetes as paq ON res.paqueteid = paq.paqueteid

JOIN clientes as cli ON res.clienteid = cli.clienteid

WHERE cli.nombre = 'Jane' AND cli.apellido = '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?**

La tabla empleados no esta relacionada con ninguna de las tablas que se pidió crear por lo cual no se puede llegar a saber que empleado atendió a los clientes.

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

SELECT SUM(cantidadpersonas) AS TotalPersonasReservadas

FROM reservas;

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

SELECT res.cantidadpersonas AS ReservasParaAventuraEnTokio

FROM paquetes as paq

JOIN reservas as res ON paq.paqueteid = res.paqueteid

WHERE paq.nombrepaquete = 'Aventura en Tokyo';