

MANEJO DE CONCEPTOS EN BASES DE DATOS



Nombre: Elio Vásquez
Gonzales

Muestra un ejemplo de DDL.

```
CREATE TABLE destinos (  
    destinoid INTEGER PRIMARY KEY,  
    nombredestino VARCHAR(100) NOT NULL,  
    descripcion TEXT,  
    preciobase NUMERIC(10,2) NOT NULL  
);
```

Muestra un ejemplo de DML

```
INSERT INTO destinos (destinoid, nombredestino, descripcion, preciobase)
VALUES
(1, 'Paris', '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);
```


Uso de inner join

INNER JOIN se utiliza en el contexto de bases de datos relacionales para combinar datos de dos o más tablas basándose en una condición de relación específica. La finalidad principal de INNER JOIN es obtener un conjunto de datos que combine registros de dos tablas según una condición común.

```
SELECT COUNT(*) AS total_reservas  
FROM reservas r  
INNER JOIN paquetes p ON r.paqueteid = p.paqueteid  
WHERE p.nombrepaquete = 'Aventura en Tokio';
```

DEFINA QUE ES UNA FUNCIÓN DE AGREGACIÓN.

Las funciones de agregación típicamente resumen la información de una columna o conjunto de columnas, proporcionando un resultado que representa alguna característica del conjunto de datos. Algunas de las funciones de agregación más comunes incluyen:

COUNT(): Calcula el número de filas en un conjunto de datos.

SUM(): Calcula la suma de los valores en una columna numérica.

AVG(): Calcula el promedio de los valores en una columna numérica.

MIN(): Encuentra el valor mínimo en una columna.

MAX(): Encuentra el valor máximo en una columna.

PARA QUÉ SIRVE LA FUNCIÓN CONCAT EN SQL-SERVER

La función CONCAT en SQL-Server se utiliza para concatenar o unir cadenas de texto. La concatenación es el proceso de unir dos o más cadenas para formar una cadena más larga. La función CONCAT toma dos o más expresiones de cadena como argumentos y las une en una única cadena. Es útil cuando necesitas combinar el contenido de varias columnas de texto en una sola columna o cuando deseas agregar texto adicional a una cadena existente.


Ejemplo:

```
SELECT CONCAT(Nombre, ' ', Apellido) AS NombreCompleto  
FROM Empleados;
```

Ejemplo de Count

Supongamos que tienes una tabla llamada "Pedidos" y quieres saber cuántos pedidos has recibido. Puedes utilizar `COUNT` de la siguiente manera:

sql

 Copy code


```
SELECT COUNT(*) AS TotalPedidos  
FROM Pedidos;
```

En este ejemplo, `COUNT(*)` cuenta el número total de filas en la tabla "Pedidos", y el resultado se muestra como "TotalPedidos".

Ejemplo de AVG

Imagina que tienes una tabla llamada "Ventas" con una columna "Monto" que representa el monto de ventas. Puedes usar `AVG` para calcular el monto promedio de las ventas:

sql

 Copy code


```
SELECT AVG(Monto) AS PromedioVentas  
FROM Ventas;
```

`AVG(Monto)` calcula el promedio de los montos de ventas en la tabla "Ventas", y el resultado se muestra como "PromedioVentas".

Ejemplo de: MIN-MAX

Considera una tabla llamada "Empleados" con una columna "Edad". Puedes utilizar `MIN` y `MAX` para encontrar la edad más joven y más vieja de los empleados:

sql

 Copy code


```
SELECT MIN(Edad) AS EdadMinima, MAX(Edad) AS EdadMaxima  
FROM Empleados;
```

En este ejemplo, `MIN(Edad)` devuelve la edad mínima y `MAX(Edad)` devuelve la edad máxima en la tabla "Empleados". Los resultados se muestran como "EdadMinima" y "EdadMaxima".

Ejemplo de Count

Supongamos que tienes una tabla llamada "Pedidos" y quieres saber cuántos pedidos has recibido. Puedes utilizar `COUNT` de la siguiente manera:

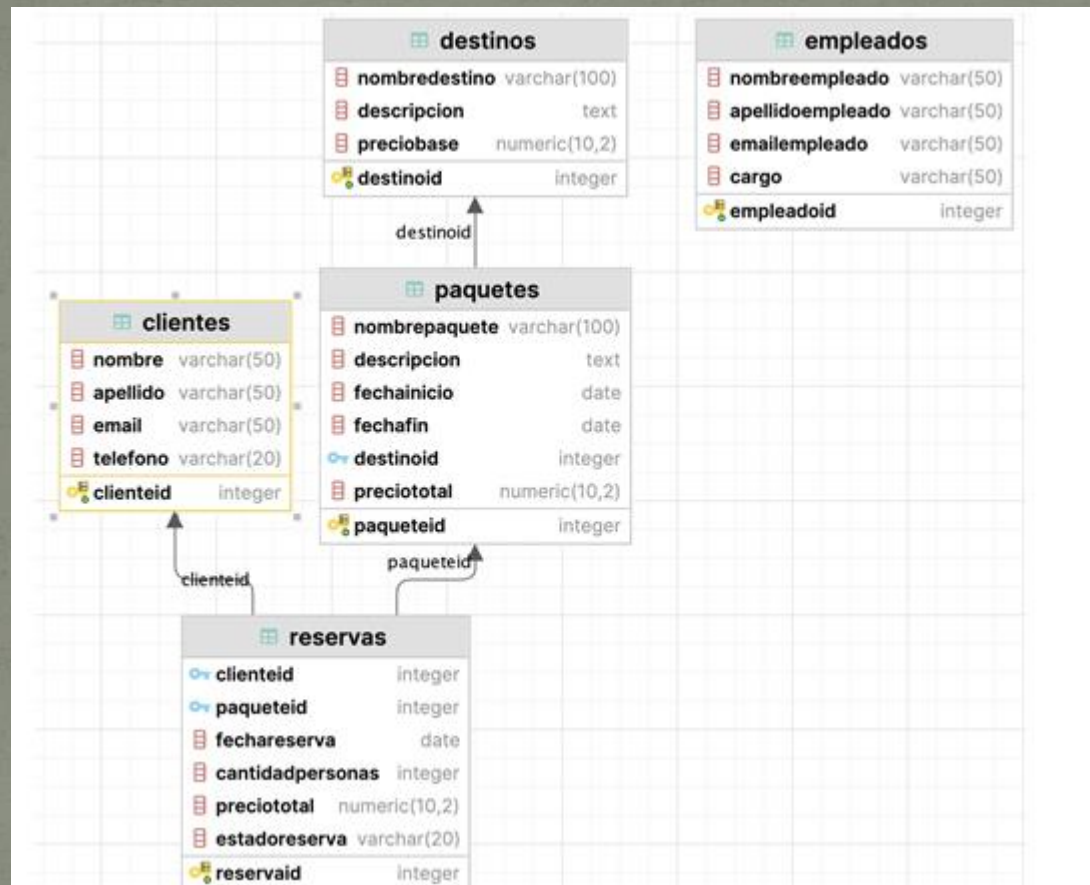
sql

 Copy code

```
SELECT COUNT(*) AS TotalPedidos  
FROM Pedidos;
```

En este ejemplo, `COUNT(*)` cuenta el número total de filas en la tabla "Pedidos", y el resultado se muestra como "TotalPedidos".

Ejercicio Base de Datos: Agencia de Viaje



```
create database agencia_viaje;
use agencia_viaje;

CREATE TABLE destinos (
    destinoid INTEGER PRIMARY KEY,
    nombredestino VARCHAR(100) NOT NULL,
    descripcion TEXT,
    preciobase NUMERIC(10,2) NOT NULL
);

INSERT INTO destinos (destinoid, nombredestino, descripcion, preciobase)
VALUES
(1, 'Paris', '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);

CREATE TABLE empleados (
    empleadoid INTEGER PRIMARY KEY,
    nombreempleado VARCHAR(50) NOT NULL,
    apellidoempleado VARCHAR(50) NOT NULL,
    email empleado VARCHAR(50) NOT NULL,
    cargo VARCHAR(50) NOT NULL
);

INSERT INTO empleados (empleadoid, nombreempleado, apellidoempleado, email empleado, cargo)
VALUES
(1, 'Maria', 'Gonez', 'maria.gonez@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', 'Especialista en Destinos');
```

```
CREATE TABLE clientes (  
    clienteid INTEGER PRIMARY KEY,  
    nombre VARCHAR(50) NOT NULL,  
    apellido VARCHAR(50) NOT NULL,  
    email VARCHAR(50) NOT NULL,  
    telefono VARCHAR(20) NOT NULL,  
);  
  
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', 'Johnson', 'bob.johnson@example.com', '555-9876'),  
(4, 'Alice', 'Williams', 'alice.williams@example.com', '555-4321'),  
(5, 'Charlie', 'Brown', 'charlie.brown@example.com', '555-8765');  
  
CREATE TABLE paquetes (  
    paqueteid INTEGER PRIMARY KEY,  
    nombrepaquete VARCHAR(100) NOT NULL,  
    descripcion TEXT,  
    fechainicio DATE,  
    fechafin DATE,  
    preciototal NUMERIC(10,2) NOT NULL,  
    destinoid INTEGER REFERENCES destinos(destinoid) NOT NULL  
);  
  
INSERT INTO paquetes (paqueteid, nombrepaquete, descripcion, fechainicio, fechafin, preciototal, destinoid)  
VALUES  
(1, 'Escapada Romántica a París', 'Disfruta de la Ciudad del amor y la luz', '2023-01-15', '2023-03-20', 1500.00, 1),  
(2, 'Aventura en Tokio', 'Descubre la Metrópolis moderna y vibrante', '2023-02-01', '2023-02-20', 2000.00, 2),  
(3, 'Explora Nueva York', 'Recorre la ciudad que nunca duerme', '2023-03-10', '2023-03-18', 2200.00, 3),  
(4, 'Historia y Arte en Roma', 'Sumérgete en la cuna de la civilización antigua', '2023-04-05', '2023-04-12', 1800.00, 4),  
(5, 'Aventura en Australia', 'Descubre la puerta de entrada a Australia', '2023-05-20', '2023-05-28', 2500.00, 5);
```



```
CREATE TABLE reservas (  
    reservaid INTEGER PRIMARY KEY,  
    clienteid INTEGER REFERENCES clientes(clienteid) NOT NULL,  
    paqueteid INTEGER REFERENCES paquetes(paqueteid) NOT NULL,  
    fechareserva DATE,  
    cantidadpersonas INTEGER,  
    preciototal NUMERIC(10,2) NOT NULL,  
    estadoreserva VARCHAR(20)  
);
```

```
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');
```

```

--¿Cuáles son los empleados que tienen el título "Agente de Ventas"?
SELECT * FROM empleados WHERE cargo = 'Agente de Ventas';

-- ¿Cuáles son los destinos de los paquetes turísticos reservados por el cliente con ID 1?
SELECT d.nombredestino
FROM destinos d
INNER JOIN paquetes p ON d.destinoid = p.destinoid
INNER JOIN reservas r ON p.paqueteid = r.paqueteid
WHERE r.clienteid = 1;

-- ¿Cuáles son los paquetes turísticos reservados por el cliente llamado "Jane Smith"?
SELECT p.nombrepaquete
FROM paquetes p
INNER JOIN reservas r ON p.paqueteid = r.paqueteid
INNER JOIN clientes c ON r.clienteid = c.clienteid
WHERE c.nombre = 'Jane' AND c.apellido = 'Smith';

-- ¿Cuáles son los nombres de los clientes, los destinos de sus paquetes turísticos y los nombres de los empleados que gestionaron esas reservas?
SELECT c.nombre, c.apellido, d.nombredestino, e.nombreempleado, e.apellidoempleado
FROM clientes c
INNER JOIN reservas r ON c.clienteid = r.clienteid
INNER JOIN paquetes p ON r.paqueteid = p.paqueteid
INNER JOIN destinos d ON p.destinoid = d.destinoid
INNER JOIN empleados e ON r.clienteid = e.empleadoid;

-- ¿Cuál es la cantidad total de personas que han reservado paquetes turísticos en la agencia?
SELECT SUM(cantidadpersonas) AS total_personas
FROM reservas;

-- ¿Cuántas reservas se han realizado para el paquete turístico llamado "Aventura en Tokio"?
SELECT COUNT(*) AS total_reservas
FROM reservas r
INNER JOIN paquetes p ON r.paqueteid = p.paqueteid
WHERE p.nombrepaquete = 'Aventura en Tokio';

```

MUCHAS GRACIAS.