**Universidad ORT Uruguay**

**Facultad de Ingeniería**

**Escuela de Tecnología**

**OBLIGATORIO**

**BASE DE DATOS 1**

****

**Emiliano Mandacen – 331839**

****

**Matías Pietrafesa – 318802**

**M2E**

**Docente: Rafael Cohen**

**Analista en Tecnologías de la Información**

**21/11/2024**

Índice

[1. Introducción 3](#_Toc183105340)

[2. Modelo entidad relación (MER) 3](#_Toc183105341)

[2.1 Diagrama Entidad Relación 3](#_Toc183105342)

[2.2 Restricciones no estructurales 3](#_Toc183105343)

[3. Modelo relacional (MR) 4](#_Toc183105344)

[3.1. Restricciones 4](#_Toc183105348)

[3.2. Normalización 4](#_Toc183105349)

[4. SQL 5](#_Toc183105350)

[4.1 Data Definition Language (DDL) 5](#_Toc183105354)

[4.2 Data Manipulation Language (DML) 5](#_Toc183105355)

[4.3 Consultas SQL 6](#_Toc183105356)

[4.3.1. Consulta 1 6](#_Toc183105364)

[4.3.2. Consulta 2 6](#_Toc183105365)

[4.3.3. Consulta 3 6](#_Toc183105366)

[4.3.4. Consulta 4 7](#_Toc183105367)

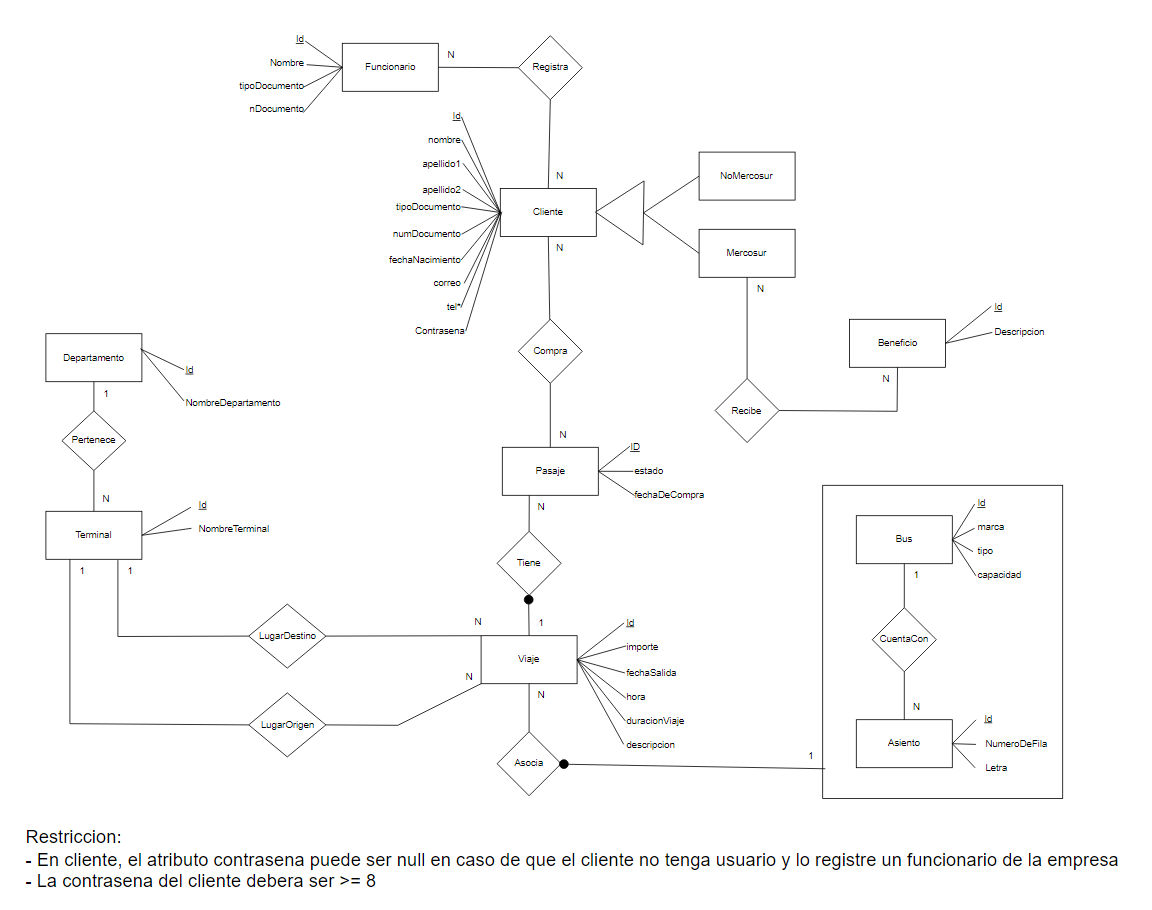
[4.3.5. Consulta 5 7](#_Toc183105368)

[4.4 Evidencia de ejecución de consultas 8](#_Toc183105369)

1. Introducción

A continuación, se verá toda la documentación relacionada al proyecto que llevamos a cabo. El mismo tiene como objetivo crear una base de datos para una empresa turística. Esta busca implementar un sistema que gestione la compra de pasajes para buses turísticos, simulando la realidad descrita por el cliente.

1. Modelo entidad relación (MER)
   1. Diagrama Entidad Relación



* 1. Restricciones no estructurales

En la entidad “Cliente”, el atributo contraseña puede ser nulo en caso de que el cliente no tenga usuario y lo registre un funcionario de la empresa.

La contraseña del cliente (en caso de tenerla) debe tener un largo de al menos ocho caracteres.

1. Modelo relacional (MR)

4. 1. Restricciones

En la entidad “Cliente”, el atributo contraseña puede ser nulo en caso de que el cliente no tenga usuario y lo registre un funcionario de la empresa.

La contraseña del cliente (en caso de tenerla) debe tener un largo de al menos ocho caracteres.

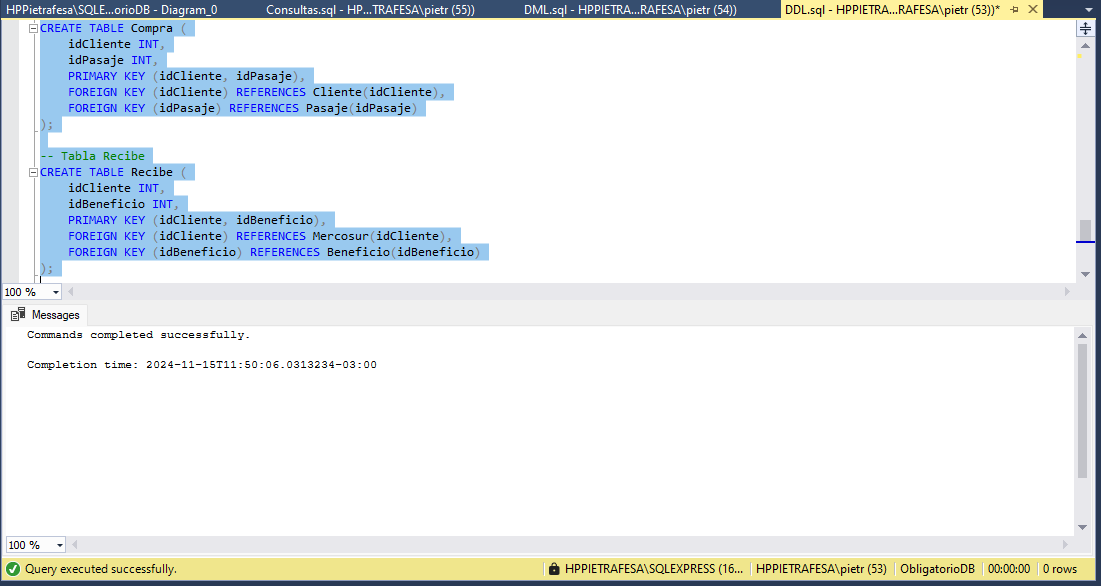
* 1. Normalización

Tablas en 3era FN

* Funcionario (idFuncionario, nombre, tipoDocumento, nDocumento)
* Cliente (idCliente, nombre, apellido1, apellido2, tipoDocumento, nDocumento, fechaNacimiento, correo, contraseña)
* ClienteTel (idCliente, telCliente)
* NoMercosur (idCliente)
* Mercosur (idCliente)
* Beneficio (idBeneficio, descripción)
* Pasaje (idPasaje, estado, fechaDeCompra, idViaje)
* Viaje (idViaje, importe, fechaSalida, hora, duracionViaje, descripción, idBus, idAsiento, idOrigen, idDestino)
* Bus (idBus, marca, tipo, capacidad)
* Asiento (idAsiento, numeroDeFila, letra, idBus)
* Departamento (idDepartamento, nombreDepartamento)
* Terminal (idTeminal, nombreTermina, idDepartamentol)
* Registra (idFuncionario, idCliente)
* Compra (idCliente, idPasaje)
* Recibe (idCliente, idBeneficio)

1. SQL

4. 1. Data Definition Language (DDL)



* 1. Data Manipulation Language (DML)

A screenshot of a computer

Description automatically generated

* 1. Consultas SQL

4. 3. 1. Consulta 1

Objetivo: Listar el o los nombres de los pasajeros con la mayor cantidad de pasajes comprados a su nombre.

El código desarrollado genera un “join” entre las tablas “compra” y “cliente” para mostrar el nombre y la cantidad de compras del cliente que más compras tiene registradas.

Código:  
select Cli.nombre, count(Co.idCliente) as 'Cantidad de compras'

from Compra Co

join Cliente Cli on Co.idCliente = Cli.idCliente

group by Cli.nombre

having COUNT(Co.idCliente) >= all (select count(co.idCliente) from Compra co group by co.idCliente);

* + 1. Consulta 2

Objetivo: Listar todos los datos de los buses con más de 35 asientos que no tengan asignado ningún destino que parta el día de mañana.

El código desarrollado genera un “join” entre “bus” y “viaje”, luego filtra los que tienen mas de 35 asientos y por ultimo se asegura de que los buses restantes no tengan ningún viaje programado para el día siguiente.

Código:  
select distinct b.Idbus,b.tipo,b.capacidad ,b.marca from bus b

join viaje v on b.idBus = v.idbus

where b.capacidad > 35

and b.idBus not in (select v.idbus from viaje v where v.fechaSalida = Dateadd(day,1,GETDATE()));

* + 1. Consulta 3

Objetivo: Listar todos los datos de los pasajeros para los cuales haya registrados en el sistema más de 5 pasajes comprados.

El código desarrollado muestra los datos de los clientes que tienen en el sistema más de 5 pasajes comprados haciendo un “join” entre “cliente” y “compra” y filtrando por la cantidad de pasajes.

Código:  
Select c.idCliente, c.nombre, c.apellido1, c.apellido2, c.tipoDocumento, c.nDocumento,

c.fechaNacimiento, c.correo, c.contrasena, COUNT(co.idPasaje) as 'Cantidad de viajes comprados' from Cliente c

join Compra co on co.idCliente = c.idCliente

GROUP BY c.idCliente, c.nombre, c.apellido1, c.apellido2, c.tipoDocumento, c.nDocumento,

c.fechaNacimiento, c.correo, c.contrasena

having count(co.idPasaje) >=5;

* + 1. Consulta 4

Objetivo: Listar idpasajero, nombre, apellidos y asiento (idasiento y fila) que correspondan a pasajes comprados para el destino cuyo idviaje es 255.

El código desarrollado hace múltiples join para mostrar los datos solicitados. Por último, filtra por el id del viaje.

Código:  
select distinct c.idCliente, cli.nombre, CONCAT(cli.apellido1, ' ', cli.apellido2) as 'Apellidos', a.letra , a.numeroDeFila

from Compra c

join Pasaje p on p.idPasaje = c.idPasaje

join Viaje v on v.idViaje = p.idViaje

join Cliente cli on cli.idCliente = c.idCliente

join Asiento a on a.Idbus = v.idBus

where v.idViaje = 255;

* + 1. Consulta 5

Objetivo: Listar todos los idviaje y cantidad de pasajes comprados durante el mes de Setiembre de este año para c/u de los destinos del pasajero cuyo correo es soyturista@gmail.com comprados en Setiembre del 2017. La lista debe estar ordenada por idviaje ascendente.

El código desarrollado lista el id del viaje y la cantidad de pasajes de los pasajes comprados en septiembre del 2017 por el pasajero con el mail “soyturista@gmail.com”.

Código:  
SELECT V.idViaje, COUNT(\*) AS cantidad\_pasajes

FROM Cliente C

JOIN Compra Co ON C.idCliente = Co.idCliente

JOIN Pasaje P ON Co.idPasaje = P.idPasaje

JOIN Viaje V ON P.idViaje = V.idViaje

WHERE C.correo = 'soyturista@gmail.com'

AND MONTH(P.fechaDeCompra) = 9

AND YEAR(P.fechaDeCompra) = 2017

GROUP BY V.idViaje

ORDER BY V.idViaje ASC;

* 1. Evidencia de ejecución de consultas

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated