

Universidad ORT Uruguay
Facultad de Ingeniería
Escuela de Tecnología

OBLIGATORIO

BASE DE DATOS 1



[Lautaro Zapata – 294574]



[Juan Scocozza– 341106]

Grupo: M2F

Docente: Nicolás Blanco

Analista en Tecnologías de la Información.

Fecha de entrega 20-11-2024

Índice

1.	Introducción.....	3
2.	Modelo entidad relación (MER).....	3
2.1	Restricciones no estructurales	3
3.	Modelo relacional (MR)	4
3.1.	Restricciones.....	7
3.2.	Normalización	7
4.	SQL.....	8
4.1	Data Manipulation Language (DML).....	8
4.2	Consultas SQL.....	10
4.3.1.	Consulta 1	10
4.3.2.	Consulta 2	10
4.3.3.	Consulta 3	11
4.3.4.	Consulta 4.....	11
4.3.5.	Consulta 5	11
4.3	Evidencia de ejecución de consultas	12

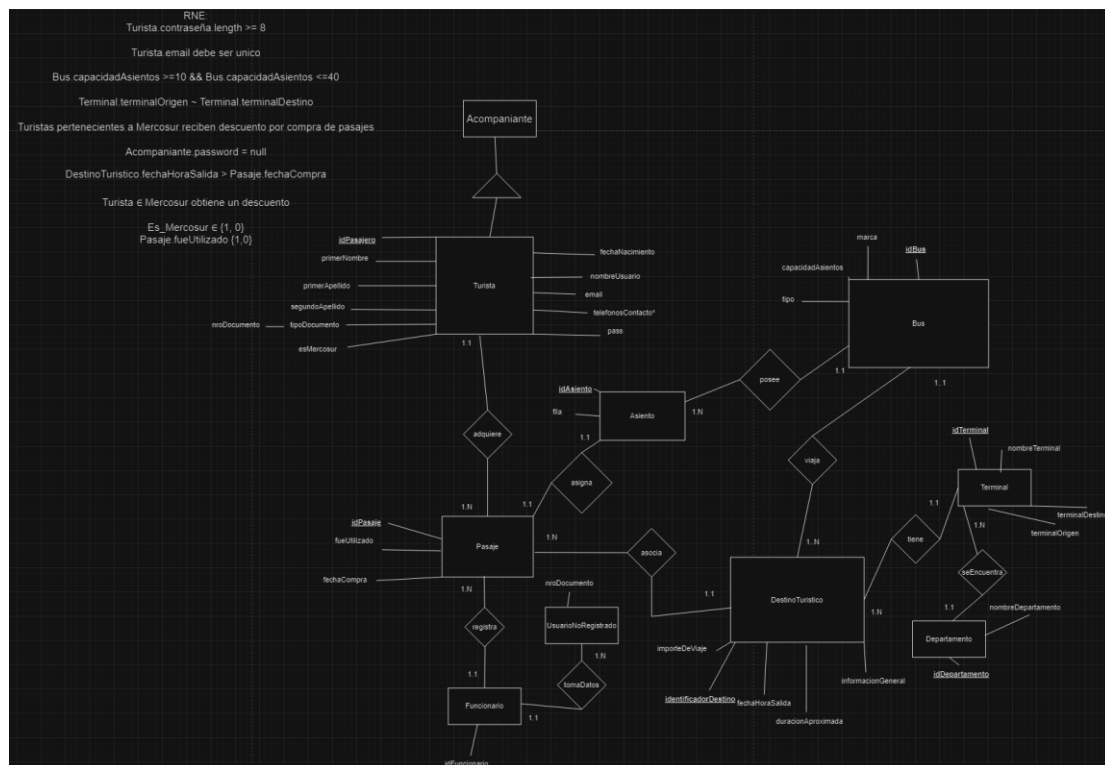
1. Introducción

Nos enfrentamos al problema de desarrollar una base de datos para una empresa de gestión turística donde se gestione la compra de pasajes para tours en buses turísticos, operando en una modalidad express.

2. Modelo entidad relación (MER)

LINK DEL MER:

<https://drive.google.com/file/d/1PeRJZXRNf-9fXx7BBWwvI8mShK0BePQX/view?usp=sharing>



2.1 Restricciones no estructurales

RNE:

Turista.contraseña.length >= 8

Turista.email debe ser unico

Bus.capacidadAsientos >=10 && Bus.capacidadAsientos <=40

Terminal.terminalOrigen ~ Terminal.terminalDestino

Turistas pertenecientes a Mercosur reciben descuento por compra de pasajes

Acompañante.password = null

DestinoTuristico.fechaHoraSalida > Pasaje.fechaCompra

Turista ∈ Mercosur obtiene un descuento

Es_Mercosur ∈ {1, 0}

Pasaje.fueUtilizado {1,0}

3. Modelo relacional (MR)

Bus (idBus, marca, capacidadAsientos, tipo)

PK: idBus

RNE: Bus.capacidadAsientos >=10 && Bus.capacidadAsientos <=40

Asiento (idAsiento, fila, idBus)

PK : idAsiento

FK: BUS.idBus

DestinoTuristico

(identificadorDestino, fechaHoraSalida, duracionAproximada, informacionGeneral, importeDeViaje, idBus, idTerminal)

PK: identificadorDestino

FK: BUS.idBus

FK: Terminal.idTerminal

Terminal (idTerminal, nombreTerminal, terminalOrigen, terminalDestino, idDepartamento)

PK: idTerminal

FK: Departamento.idDepartamento

RNE: Terminal.terminalOrigen ~ Terminal.terminalDestino

Departamento (idDepartamento, nombreDepartamento)

PK: idDepartamento

Pasaje (idPasaje, fueUtilizado, fechaCompra, identificadorDestino, idFuncionario, idPasajero, idAsiento)

PK : idPasaje

FK: Funcionario.idFuncionario

FK: DestinoTuristico.identificadorDestino

FK: Asiento.idAsiento

FK: Turista.idPasajero

RNE: DestinoTuristico.fechaHoraSalida > Pasaje.fechaCompra

RNE: Pasaje.fueUtilizado { 1,0 }

Turista (idPasajero, primerNombre, primerApellido, segundoApellido, nroDocumento, fechaNacimiento, nombreUsuario, email, pass, esMercosur)

PK : idPasajero

RNE: Turistas pertenecientes a Mercosur reciben descuento por compra de pasajes

RNE: Turista \in Mercosur obtiene un descuento

RNE: Es_Mercosur \in {1, 0}

RNE: Turista.contraseña.length \geq 8

RNE: Turista.email debe ser único

Turista_TelefonosContacto (idPasajero, telefonosContacto)

PK : idPasajero, telefonosContacto

FK: Turista.idPasajero

Acompañante(idPasajero, idFuncionario)

PK: idPasajero

FK: Turista.idPasajero

FK: Funcionario, idFuncionario

RNE: Acompañante.password = null

Funcionario(idFuncionario)

PK: idFuncionario

UsuarioNoRegistrado(nroDocumento, idFuncionario)

PK: nroDocumento

FK: Funcionario.idFuncionario

3.1. Restricciones

1. Turista.contraseña.length >= 8
2. Turista.email debe ser único
3. Turistas pertenecientes a Mercosur reciben descuento por compra de pasajes
4. Turista ∈ Mercosur obtiene un descuento
5. Es_Mercosur ∈ {true, false}
6. Bus.capacidadAsientos >=10 && Bus.capacidadAsientos <=40
7. Terminal.terminalOrigen ~ Terminal.terminalDestino
8. Acompañante.password = null
10. DestinoTuristico.fechaHoraSalida > Pasaje.fechaCompra

3.2. Normalización

El Modelo Relacional se encuentra ya realizado en una 3FN. En la 1FN suprimimos los atributos multivaluados. Está también en 2FN ya que cumple la 1FN y cada atributo no clave de la relación es dependencia funcional completa de la PK. Por último, pasamos a 3FN ya que no debe existir ningún atributo que dependa funcionalmente de atributos que no son clave.

4. SQL

4.1 Data Manipulation Language (DML)

Capturas con las inserciones de datos mediante CHAT GPT. Algunas respuestas fueron erróneas o contenían errores, por este motivo manipulamos los datos manualmente a partir de los datos generados por la IA.



Creame 15 pasajes y que se repitan entre los idPasajeros



Aquí tienes un ejemplo de cómo insertar 15 pasajes para diferentes `idPasajero`, repitiendo los pasajeros entre los pasajes, utilizando los valores de `idAsiento` de la tabla `Asiento` para asignar los asientos correctamente.

Inserción de 15 pasajes

sql

Copiar código

```
INSERT INTO Pasaje (fechaUtilizado, fechaCompra, identificadorDestino, idFuncionario, idPasajero)
VALUES
(1, '2024-10-12', 1, 1, 1), -- Pasaje 1
(0, '2024-10-12', 1, 1, 2, 5), -- Pasaje 2
(1, '2024-10-12', 2, 2, 1, 10), -- Pasaje 3
(1, '2024-10-13', 3, 3, 3, 15), -- Pasaje 4
(0, '2024-10-13', 4, 4, 4, 20), -- Pasaje 5
(1, '2024-10-13', 5, 5, 2, 1), -- Pasaje 6
(0, '2024-10-14', 1, 1, 3, 5), -- Pasaje 7
(1, '2024-10-14', 2, 2, 1, 10), -- Pasaje 8
(0, '2024-10-14', 3, 3, 4, 15), -- Pasaje 9
(1, '2024-10-14', 4, 4, 2, 20), -- Pasaje 10
(0, '2024-10-15', 5, 5, 3, 1), -- Pasaje 11
(1, '2024-10-15', 1, 1, 4, 5), -- Pasaje 12
(0, '2024-10-15', 2, 2, 1, 10), -- Pasaje 13
(1, '2024-10-15', 3, 3, 2, 15), -- Pasaje 14
(0, '2024-10-15', 4, 4, 3, 20); -- Pasaje 15
```



Necesito hacer la inserción de datos para esta tabla

```
CREATE TABLE Asiento(idBus INTEGER,
idAsiento INTEGER PRIMARY KEY IDENTITY,
fila VARCHAR(2)
CONSTRAINT FK_idBus FOREIGN KEY (idBus) REFERENCES
Bus(idBus)
)
```

Necesito que la fila sea una letra.

TEN EN CUENTA QUE ESTOS SON MIS BUSES

```
INSERT INTO Bus (marca, capacidadAsientos, tipo) VALUES
('Mercedes', '40', 'Turismo'),
('Volvo', '36', 'Turismo'),
('Hyundai', '20', 'Mini');
```

```
INSERT INTO Bus (marca, capacidadAsientos, tipo) VALUES
('Scania', '30', 'Turismo'),
('MAN', '25', 'Mini'),
('Iveco', '38', 'Turismo'),
('Mercedes', '12', 'Mini'),
('Volvo', '15', 'Mini'),
('Scania', '35', 'Turismo'),
('MAN', '28', 'Mini');
```

```
sql Copiar código

-- BUS VOLVO 2
INSERT INTO Asiento (idBus, fila) VALUES
(2, 'A'), (2, 'A'), (2, 'A'), (2, 'A'), (2, 'A'), (2, 'A'), (2, 'A'), (2, 'A'), (2, 'A'), (2, 'A'),
(2, 'B'), (2, 'B'), (2, 'B'), (2, 'B'), (2, 'B'), (2, 'B'), (2, 'B'), (2, 'B'), (2, 'B'), (2, 'B'),
(2, 'C'), (2, 'C'), (2, 'C'), (2, 'C'), (2, 'C'), (2, 'C'), (2, 'C'), (2, 'C'), (2, 'C'), (2, 'C'),
(2, 'D'), (2, 'D'), (2, 'D'), (2, 'D'), (2, 'D'), (2, 'D'), (2, 'D'), (2, 'D');

-- BUS HYUNDAI
INSERT INTO Asiento (idBus, fila) VALUES
(3, 'A'), (3, 'A'), (3, 'A'), (3, 'A'), (3, 'A'), (3, 'A'), (3, 'A'), (3, 'A'), (3, 'A'), (3, 'A'),
(3, 'B'), (3, 'B'), (3, 'B'), (3, 'B'), (3, 'B'), (3, 'B'), (3, 'B'), (3, 'B'), (3, 'B'), (3, 'B'),
(3, 'C'), (3, 'C'), (3, 'C'), (3, 'C'), (3, 'C'), (3, 'C'), (3, 'C'), (3, 'C'), (3, 'C'), (3, 'C'),
(3, 'D'), (3, 'D'), (3, 'D'), (3, 'D'), (3, 'D'), (3, 'D'), (3, 'D'), (3, 'D');

-- BUS SCANIA
INSERT INTO Asiento (idBus, fila) VALUES
(4, 'A'), (4, 'A'), (4, 'A'), (4, 'A'), (4, 'A'), (4, 'A'), (4, 'A'), (4, 'A'), (4, 'A'), (4, 'A'),
(4, 'B'), (4, 'B'), (4, 'B'), (4, 'B'), (4, 'B'), (4, 'B'), (4, 'B'), (4, 'B'), (4, 'B'), (4, 'B'),
(4, 'C'), (4, 'C'), (4, 'C'), (4, 'C'), (4, 'C'), (4, 'C'), (4, 'C'), (4, 'C'), (4, 'C'), (4, 'C'),
(4, 'D'), (4, 'D'), (4, 'D'), (4, 'D'), (4, 'D'), (4, 'D'), (4, 'D'), (4, 'D');

-- BUS MAN
INSERT INTO Asiento (idBus, fila) VALUES
(5, 'A'), (5, 'A'), (5, 'A'), (5, 'A'), (5, 'A'), (5, 'A'), (5, 'A'), (5, 'A'), (5, 'A'), (5, 'A'),
(5, 'B'), (5, 'B'), (5, 'B'), (5, 'B'), (5, 'B'), (5, 'B'), (5, 'B'), (5, 'B'), (5, 'B'), (5, 'B'),
(5, 'C'), (5, 'C'), (5, 'C'), (5, 'C'), (5, 'C'), (5, 'C'), (5, 'C'), (5, 'C'), (5, 'C'), (5, 'C'),
(5, 'D'), (5, 'D'), (5, 'D'), (5, 'D'), (5, 'D'), (5, 'D'), (5, 'D'), (5, 'D');

-- BUS IVECO
INSERT INTO Asiento (idBus, fila) VALUES
(6, 'A'), (6, 'A'), (6, 'A'), (6, 'A'), (6, 'A'), (6, 'A'), (6, 'A'), (6, 'A'), (6, 'A'), (6, 'A'),
(6, 'B'), (6, 'B'), (6, 'B'), (6, 'B'), (6, 'B'), (6, 'B'), (6, 'B'), (6, 'B'), (6, 'B'), (6, 'B'),
(6, 'C'), (6, 'C'), (6, 'C'), (6, 'C'), (6, 'C'), (6, 'C'), (6, 'C'), (6, 'C'), (6, 'C'), (6, 'C'),
(6, 'D'), (6, 'D'), (6, 'D'), (6, 'D'), (6, 'D'), (6, 'D'), (6, 'D'), (6, 'D');
```

4.2 Consultas SQL

4.3.1. Consulta 1

```
SELECT t.primerNombre, COUNT(p.idPasaje) AS cantidadPasajes
FROM Turista t
JOIN Pasaje p ON t.idPasajero = p.idPasajero
GROUP BY t.idPasajero, t.primerNombre
```

4.3.2. Consulta 2

```
SELECT *
FROM Bus b
JOIN DestinoTuristico d ON b.idBus = d.idBus
```

WHERE b.capacidadAsientos > 35 AND d.fechaHoraSalida > DATEADD(DAY, 1, GETDATE()) --Se le agrega un DIA a la FECHA ACTUAL

4.3.3. Consulta 3

SELECT t.idPasajero, t.email, t.esMercosur,t.fechaNacimiento, t.nroDocumento, t.nombreUsuario, t.pass, t.primerApellido, t.segundoApellido, t.primerNombre, tc.telefonosContacto , COUNT(p.idPasaje) AS cantidadPasajes

FROM Turista t

JOIN Pasaje p ON t.idPasajero = p.idPasajero

JOIN Turista_TelefonosContacto tc ON t.idPasajero = tc.idPasajero

GROUP BY t.idPasajero, t.email, t.esMercosur,t.fechaNacimiento, t.nroDocumento, t.nombreUsuario, t.pass, t.primerApellido, t.segundoApellido, t.primerNombre, tc.telefonosContacto

HAVING COUNT(p.idPasaje) >= 5

4.3.4. Consulta 4

SELECT t.idPasajero,t.primerApellido, t.segundoApellido, t.primerNombre, a.idAsiento, a.fila

FROM Pasaje p

JOIN Turista t ON p.idPasajero = t.idPasajero

JOIN Asiento a ON p.idAsiento = a.idAsiento

WHERE p.idPasaje = 255

4.3.5. Consulta 5

SELECT p.idPasaje, COUNT(p.idPasaje) AS cantidadPasajes, t.email, p.fechaCompra

FROM Pasaje p

JOIN Turista t ON p.idPasajero = t.idPasajero

GROUP BY p.idPasaje, t.email, p.fechaCompra

HAVING p.fechaCompra > '08/31/2017' AND p.fechaCompra <= '09/30/2017' AND
t.email = 'soyturista@gmail.com'

4.3 Evidencia de ejecución de consultas

-- CONSULTA 1 : Listar el o los nombres de los pasajeros con la mayor cantidad de pasajes comprados a su nombre.

	primerNombre	cantidadPasajes
1	Juan	5
2	María	5
3	Carlos	2
4	Ana	2
5	Luis	2

-- CONSULTA 2: Listar todos los datos de los buses con más de 35 asientos que no tengan asignado ningún destino que parta el día de mañana

	idBus	marca	capacidadAsientos	tipo	identificadorDestino	fechaHoraSalida	duracionAproximada	informacionGeneral	importeDeViaje	idBus	idTerminal
1	1	Mercedes	40	Turismo	1	2024-12-15 08:00:00.000	3	Viaje a Canelones	100	1	1
2	2	Volvo	36	Turismo	2	2024-12-16 09:00:00.000	2	Viaje a Colonia	120	2	3

-- CONSULTA 3: Listar todos los datos de los pasajeros para los cuales haya registrados en el sistema más de 5 pasajes comprados.

	idPasajero	email	esMercosur	fechaNacimiento	nroDocumento	nombreUsuario	pass	primerApellido	segundoApellido	primerNombre	telefonosContacto	cantidadPasajes
1	1	soyturista@gmail.com	1	1990-01-01	123456789	juanperez	password123	Pérez	Gómez	Juan	1234567890	6
2	2	maria@example.com	0	1985-05-23	987654321	marialopez	password456	López	González	María	2345678901	6

-- CONSULTA 4: Listar idpasajero, nombre, apellidos y asiento (idasiento y fila) que correspondan a pasajes comprados para el destino cuyo idviaje es 255.

	idPasajero	primerApellido	segundoApellido	primerNombre	idAsiento	fila
1	2	López	González	María	5	A

--CONSULTA 5: 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

	idPasaje	cantidadPasajes	email	fechaCompra
1	11	1	soyturista@gmail.com	2017-09-15
2	14	1	soyturista@gmail.com	2017-09-15