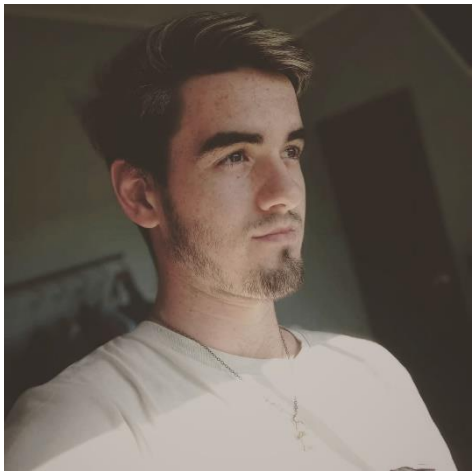
Universidad ORT Uruguay
Facultad de Ingeniería
Escuela de Tecnología

Obligatorio Base De Datos

Grupo M2A

Agustín Butrico – 339579



Valentín Latorre – 339103

Profesor Rafael Alejandro Cohen Abut.

Índice

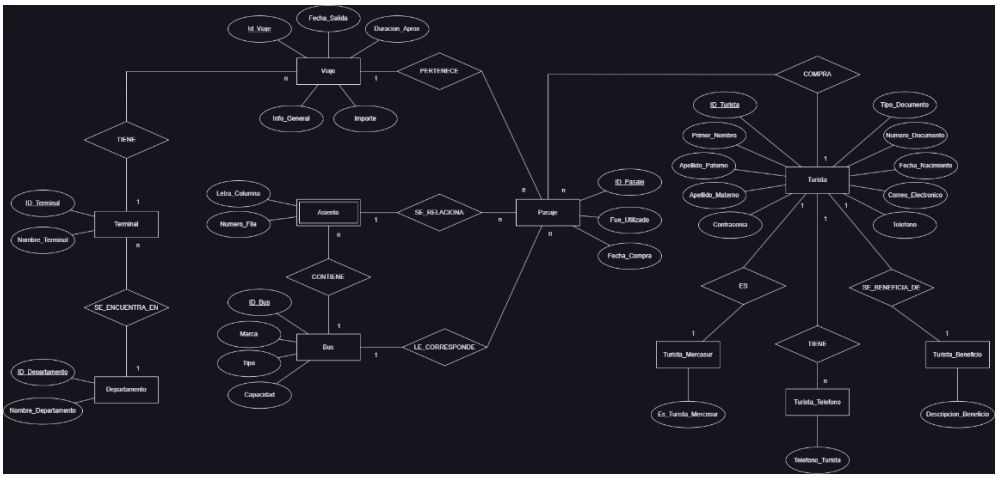
1. Introducción 3
2. Modelo entidad relación (MER) 3
2.1 Diagrama Entidad Relación 3
2.2 Restricciones no estructurales 3
2.3 Justificaciones y otras consideraciones 3
3. Modelo relacional (MR) 3
3.1. Restricciones 4
3.2. Normalización 4
4. SQL 4
4.1 Data Definition Language (DDL) 4
4.2 Data Manipulation Language (DML) 7
4.3 Consultas SQL 11
4.3.1. Consulta 1 11
4.3.2. Consulta 2 11
4.3.3. Consulta 3 12
4.3.4. Consulta 4 12
4.3.5. Consulta 5 13
4.4 Evidencia de ejecución de consultas 13

1. Introducción

El problema consiste en desarrollar un sistema para gestionar la compra y administración de pasajes para tours en buses turísticos.

2. Modelo entidad relación (MER)

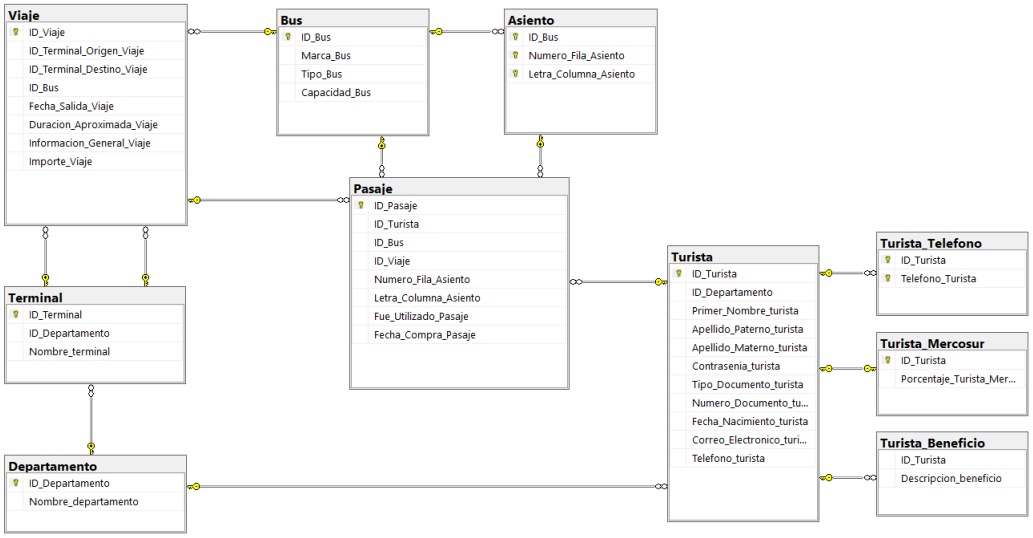
2.1 Diagrama Entidad Relación



2.2 Restricciones no estructurales

2.3 Justificaciones y otras consideraciones

3. Modelo relacional (MR)



3.1.Restricciones

3.2.Normalización

Se realizo la normalización de la tabla Telefono

<<Incluir conversión si corresponde y su justificación>>

4. SQL

4.1 Data Definition Language (DDL)

```
CREATE DATABASE Obligatorio_BD;
USE Obligatorio_BD;

CREATE TABLE Bus (
    ID_Bus INT NOT NULL,
    Marca_Bus VARCHAR(30) NOT NULL,
    Tipo_Bus VARCHAR(30) NOT NULL,
    Capacidad_Bus INT NOT NULL,
    primary key (ID_Bus)
);

CREATE TABLE Asiento (
    ID_Bus INT NOT NULL,
    Numero_Fila_Asiento INT NOT NULL,
    Letra_Columna_Asiento CHAR(2),
    primary key (ID_Bus, Numero_Fila_Asiento,
Letra_Columna_Asiento),
    foreign key (ID_Bus) references Bus(ID_Bus)
);

CREATE TABLE Departamento (
    ID_Departamento INT NOT NULL,
    Nombre_departamento VARCHAR(30),
    primary key (ID_Departamento),
);
```

```sql
CREATE TABLE Turista (
    ID_Turista INT NOT NULL,
    ID_Departamento INT NOT NULL,
    Primer_Nombre_turista VARCHAR(20) NOT NULL,
    Apellido_Paterno_turista VARCHAR(20) NOT NULL,
    Apellido_Materno_turista VARCHAR(20) NOT NULL,
    Contrasenia_turista VARCHAR(20) NOT NULL,
    Tipo_Documento_turista VARCHAR(20) NOT NULL,
    Numero_Documento_turista VARCHAR(20) NOT NULL,
    Fecha_Nacimiento_turista Date NOT NULL,
    Correo_Electronico_turista VARCHAR(30) NOT
NULL,
    Telefono_turista VARCHAR(15),
    primary key (ID_Turista),
    foreign key (ID_Departamento) references
Departamento(ID_Departamento)
);

CREATE TABLE Turista_Beneficio (
    ID_Turista INT NOT NULL,
    Descripcion_beneficio VARCHAR(200) NOT NULL,
    foreign key (ID_Turista) references
Turista(ID_Turista)
);

CREATE TABLE Turista_Mercosur (
    ID_Turista INT NOT NULL,
    Porcentaje_Turista_Mercosur Varchar(20),
    primary key (ID_Turista),
    foreign key (ID_Turista) references
Turista(ID_Turista)
);

CREATE TABLE Turista_Telefono (
    ID_Turista INT NOT NULL,
    Telefono_Turista varchar(20),
    primary key (ID_Turista, Telefono_Turista),
    foreign key (ID_Turista) references
Turista(ID_Turista)
)

CREATE TABLE Terminal (
```

```
    ID_Terminal INT NOT NULL,
    ID_Departamento INT NOT NULL,
    Nombre_terminal VARCHAR(30) NOT NULL,
    primary key (ID_Terminal),
    foreign key (ID_Departamento) references
Departamento(ID_Departamento)
);

CREATE TABLE Viaje (
    ID_Viaje INT NOT NULL,
    ID_Terminal_Origen_Viaje INT NOT NULL,
    ID_Terminal_Destino_Viaje INT NOT NULL,
    ID_Bus INT NOT NULL,
    Fecha_Salida_Viaje DATETIME NOT NULL,
    Duracion_Aproximada_Viaje TIME NOT NULL,
    Informacion_General_Viaje TEXT NOT NULL,
    Importe_Viaje INT NOT NULL,
    primary key (ID_Viaje),
    foreign key (ID_Terminal_Origen_Viaje)
references Terminal(ID_Terminal),
    foreign key (ID_Terminal_Destino_Viaje)
references Terminal(ID_Terminal),
    foreign key (ID_Bus) references Bus(ID_Bus),
    CONSTRAINT chk_origen_destino CHECK
(ID_Terminal_Origen_Viaje <>
ID_Terminal_Destino_Viaje) -- Restricci►n que
chequea que el origen y destino sean distintos
);

CREATE TABLE Pasaje (
    ID_Pasaje INT NOT NULL,
    ID_Turista INT NOT NULL,
    ID_Bus INT NOT NULL,
    ID_Viaje INT NOT NULL,
    Numero_Fila_Asiento INT NOT NULL,
    Letra_Columna_Asiento CHAR(2),
    Fue_Utilizado_Pasaje BIT NOT NULL, -- 0: no
utilizado, 1: utilizado
    Fecha_Compra_Pasaje DATETIME NOT NULL,
    primary key (ID_Pasaje),
    foreign key (ID_Turista) references
Turista(ID_Turista),
```

```
    foreign key (ID_Bus) references Bus(ID_Bus),
    foreign key (ID_Viaje) references
Viaje(ID_Viaje),
    foreign key (ID_Bus, Numero_Fila_Asiento,
Letra_Columna_Asiento) references Asiento(ID_Bus,
Numero_Fila_Asiento, Letra_Columna_Asiento)
);
```

4.2 Data Manipulation Language (DML)

```
USE Obligatorio_BD;
SET DATEFORMAT YMD;

-- Insertar datos en la tabla Turista_Beneficio
INSERT INTO Turista_Beneficio (ID_Turista,
Descripcion_beneficio) VALUES
(1, 'Descuento en hotel'),
(2, 'Acceso a tour guiado'),
(3, 'Seguro de viaje');

-- Insertar datos en la tabla Turista_Mercosur
INSERT INTO Turista_Mercosur (ID_Turista,
Porcentaje_Turista_Mercosur) VALUES
(1, 'Estándar'),
(2, 'VIP'),
(3, 'Estudiante');

-- Insertar datos en la tabla Turista_Telefono
INSERT INTO Turista_Telefono (ID_Turista,
Telefono_Turista) VALUES
(1, '09488999019'),
(2, '096876432'),
(3, '349589685'),
(4, '123455555');

-- Insertar datos en la tabla Turista
INSERT INTO Turista (ID_Turista, ID_Departamento,
Primer_Nombre_turista, Apellido_Paterno_turista,
Apellido_Materno_turista, Contrasenia_turista,
Tipo_Documento_turista, Numero_Documento_turista,
Fecha_Nacimiento_turista,
```

```sql
Correo_Electronico_turista, Telefono_turista)
VALUES
(1, 1, 'Juan', 'Pérez', 'García', 'contraseña123',
'DNI', '12345678', '1990-05-15',
'juan.perez@example.com', '987654321'),
(2, 2,  'Ana', 'Lopez', 'Martínez',
'contraseña456', 'Pasaporte', 'A98765432', '1985-
02-20', 'ana.lopez@example.com', '123456789'),
(3, 3, 'Carlos', 'Mendoza', 'Fernández',
'contraseña789', 'Cédula', '98765432', '1995-08-
30', 'carlos.mendoza@example.com', '555123456'),
(4, 1, 'Esteban', 'Carlos', 'Jose', 'pass1234',
'DNI', '33444556', '1990-05-20',
'soyturista@gmail.com', '123455555');
-- Insertar datos en la tabla Departamento
INSERT INTO Departamento (ID_Departamento,
Nombre_departamento) VALUES
(1, 'Turismo'),
(2, 'Logística'),
(3, 'Finanzas');

-- Insertar datos en la tabla Terminal
INSERT INTO Terminal (ID_Terminal,
ID_Departamento, Nombre_terminal) VALUES
(1, 1, 'Terminal A'),
(2, 1, 'Terminal B'),
(3, 2, 'Terminal C');

-- Insertar datos en la tabla Viaje
INSERT INTO Viaje (ID_Viaje,
ID_Terminal_Origen_Viaje,
ID_Terminal_Destino_Viaje, ID_Bus,
Fecha_Salida_Viaje, Duracion_Aproximada_Viaje,
Informacion_General_Viaje, Importe_Viaje) VALUES
(1, 1, 2, 1, '2024-11-20 08:00:00', '02:30:00',
'Tour desde Terminal A a Terminal B', 100),
(2, 1, 3, 2, '2024-11-21 15:30:00', '03:00:00',
'Tour desde Terminal A a Terminal C', 150),
(3, 2, 3, 3, '2024-11-19 09:00:00', '01:45:00',
'Tour desde Terminal B a Terminal C', 120),
(4, 1, 3, 1, '2024-11-22 07:30:00', '02:15:00',
'Tour desde Terminal A a Terminal C', 120),
```

```
(5, 3, 1, 2, '2024-11-23 11:00:00', '03:00:00',
'Tour desde Terminal C a Terminal A', 130),
(6, 2, 1, 3, '2024-11-24 14:00:00', '01:50:00',
'Tour desde Terminal B a Terminal A', 110),
(7, 1, 2, 1, '2024-11-25 10:30:00', '02:45:00',
'Tour desde Terminal A a Terminal B', 100),
(8, 3, 2, 2, '2024-11-26 16:00:00', '03:15:00',
'Tour desde Terminal C a Terminal B', 140),
(9, 2, 3, 3, '2024-11-27 08:45:00', '01:40:00',
'Tour desde Terminal B a Terminal C', 120),
(10, 1, 3, 1, '2024-11-28 09:30:00', '02:00:00',
'Tour desde Terminal A a Terminal C', 125),
(11, 1, 2, 1, '2017-09-10 08:00:00', '02:30:00',
'Viaje de A a B', 100),
(12, 1, 3, 2, '2017-09-15 15:30:00', '03:00:00',
'Viaje de A a C', 150),
(13, 2, 3, 3, '2017-09-20 09:00:00', '01:45:00',
'Viaje de B a C', 120),
(14, 2, 3, 3, '2017-09-14 05:00:00', '07:45:00',
'Viaje de B a C', 620); --


-- Insertar datos en la tabla Asiento
INSERT INTO Asiento(ID_Bus, Numero_Fila_Asiento,
Letra_Columna_Asiento) VALUES
(1, 24, 'AA'),
(2, 12, 'AB'),
(3, 28, 'AC'),
(2, 5, 'A'),
(2, 12, 'B'),
(3, 28, 'A'),
(3, 15, 'C'),
(1, 18, 'D'),
(2, 8, 'E'),
(3, 30, 'F'),
(1, 37, 'AN'),
(1, 47, 'GH'),
(3, 7, 'U'),
(2, 21,'QE'),
(3, 32,'LP'),
(3, 11, 'AF'); --
```

```sql
-- Insertar datos en la tabla Bus
INSERT INTO Bus (ID_Bus, Marca_Bus, Tipo_Bus,
Capacidad_Bus) VALUES
(1, 'Mercedes', 'Autobús turístico', 40),
(2, 'Volvo', 'Autobús de lujo', 50),
(3, 'Scania', 'Autobús económico', 35);


-- Insertar datos en la tabla Pasaje
INSERT INTO Pasaje (ID_Pasaje, ID_Turista, ID_Bus,
ID_Viaje, Numero_Fila_Asiento,
Letra_Columna_Asiento, Fue_Utilizado_Pasaje,
Fecha_Compra_Pasaje) VALUES
(1, 1, 1, 1, 24, 'AA', 0, '2024-11-15'),
(2, 2, 2, 2, 12, 'AB', 1, '2024-11-16'),
(3, 3, 3, 3, 28, 'AC', 0, '2024-11-17'),
(4, 1, 3, 4, 28, 'A', 0, '2024-11-20'),
(5, 1, 2, 5, 8, 'E', 1, '2024-11-21'),
(6, 1, 1, 6, 18, 'D', 0, '2024-11-22'),
(7, 1, 2, 7, 8, 'E', 0, '2024-11-23'),
(8, 1, 3, 8, 30, 'F', 1, '2024-11-24'),
(9, 1, 1, 10, 37, 'AN', 0, '2017-09-01'),
(10, 1, 1, 11, 47, 'GH', 1, '2017-09-02'),
(11, 1, 2, 12, 7, 'U', 0, '2017-09-05'),
(12, 1, 2, 12, 21, 'QE', 1, '2017-09-15'),
(13, 1, 3, 13, 32, 'LP', 0, '2017-09-25'),
(14, 4, 3, 14, 11, 'AF', 0, '2017-09-13');
```

4.3 Consultas SQL

4.3.1. Consulta 1

```sql
/*
1. Listar el o los nombres de los pasajeros con la mayor cantidad de pasajes comprados a
su nombre.

/Explicacion de la Consulta:/

T. Turista: Tabla principal que contiene los datos de los Turistas

HAVING COUNT(*) combinado con un filtro que selecciona

SELECT TOP: muestra el número máximo de pasajes comprados usando SELECT TOP 1 ... ORDER BY COUNT(*) DESC.

*/

SELECT
    T.ID_Turista,
    T.Primer_Nombre_Turista,
    T.Apellido_Paterno_Turista,
    T.Apellido_Materno_Turista,
    P.Cantidad_Pasajes

FROM Turista T

JOIN
    ( SELECT ID_Turista, COUNT(*) AS Cantidad_Pasajes

      FROM Pasaje GROUP BY ID_Turista HAVING COUNT(*) = (SELECT TOP 1 COUNT(*)

      FROM Pasaje GROUP BY ID_Turista ORDER BY COUNT(*) DESC
    )
) P ON T.ID_Turista = P.ID_Turista;

/*
```

4.3.2. Consulta 2

```sql
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.

/Explicación de la Consulta:/

B. Bus: Tabla principal que contiene los datos de los buses.

LEFT JOIN:  Vincula los buses con los viajes asignados.

B.Capacidad_Bus > 35: Filtra los buses con más de 35 asientos.

CAST(V.Fecha_Salida_Viaje AS DATE) <> CAST(DATEADD(DAY, 1, GETDATE()) AS DATE): Excluye buses que tengan un viaje programado
para el día de mañana.

V.ID_Viaje IS NULL: Incluye buses sin viajes asignados.

GETDATE(): Devuelve la fecha y hora.

*/
SELECT B.* FROM Bus B LEFT JOIN Viaje V ON B.ID_Bus = V.ID_Bus

AND CAST(V.Fecha_Salida_Viaje AS DATE) = CAST(DATEADD(DAY, 1, GETDATE()) AS DATE)

WHERE B.Capacidad_Bus > 35 AND V.ID_Viaje IS NULL;
```

4.3.3. Consulta 3

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

SELECT: Tabla principal que contiene los datos de los Turistas.

JOIN PASAJE P: Vincula a los turistas con los pasajes que han comprado cada uno, basándose en la columna común ID_Turista.

GROUP BY: Agrupa los resultados por las columnas seleccionadas.

HAVING COUNT: Cuenta cuántos pasajes están asociados con cada Turista.
*/
SELECT T.ID_Turista,
    T.Primer_Nombre_turista,
    T.Apellido_Paterno_turista,
    T.Apellido_Materno_turista,
    T.Numero_Documento_turista,
    T.Contrasenia_turista,
    T.Correo_Electronico_turista,
    T.Telefono_turista,
    T.Tipo_Documento_turista

FROM Turista T

JOIN Pasaje P on P.ID_Turista = T.ID_Turista

GROUP BY T.ID_Turista,
    T.Primer_Nombre_turista,
    T.Apellido_Paterno_turista,
    T.Apellido_Materno_turista,
    T.Numero_Documento_turista,
    T.Contrasenia_turista,
    T.Correo_Electronico_turista,
    T.Telefono_turista,
    T.Tipo_Documento_turista

HAVING COUNT(P.ID_Pasaje)>5
```

4.3.4. Consulta 4

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

/Explicación de la consulta:/
SELECT: Establece los campos a mostrar, el ID del turista, su nombre, apellidos y los detalles del asiento.

FROM Pasaje p: Especifica la tabla Pasaje para la busqueda.

JOIN Turista t: Se une con la tabla Turista usando la clave foránea ID_Turista para obtener la información del turista.

JOIN Asiento a: Se une con la tabla Asiento usando ID_Bus, Numero_Fila_Asiento y Letra_Columna_Asiento
 para obtener la información del asiento correspondiente.

JOIN Viaje v: Se une con la tabla Viaje usando ID_Viaje para filtrar los registros del viaje cuyo ID_Viaje es 255.

WHERE v.ID_Viaje = X: Filtra los resultados para solo mostrar los pasajes del viaje con ID X.

*/
SELECT  t.ID_Turista,
        t.Primer_Nombre_turista,
        t.Apellido_Paterno_turista,
        t.Apellido_Materno_turista,
        a.ID_Bus,
        a.Numero_Fila_Asiento,
        a.Letra_Columna_Asiento

FROM Pasaje p JOIN Turista t ON p.ID_Turista = t.ID_Turista

    JOIN Asiento a ON p.ID_Bus = a.ID_Bus AND p.Numero_Fila_Asiento = a.Numero_Fila_Asiento

    AND p.Letra_Columna_Asiento = a.Letra_Columna_Asiento

    JOIN Viaje v ON p.ID_Viaje = v.ID_Viaje

WHERE
    v.ID_Viaje = 5; -- Puede ser cualquier valor que se tenga que buscar
```

4.3.5. Consulta 5

```
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.

/Explicación de la consulta:/

SELECT: p.ID_Viaje: Muestra el ID del viaje.

COUNT(p.ID_Pasaje): Cuenta la cantidad de pasajes comprados para cada viaje.

FROM Pasaje p: Especifica la tabla Pasaje como la base de la consulta.

JOIN Turista t: Se une con la tabla Turista usando la clave foránea ID_Turista para vincular los pasajes con los datos del turista.

WHERE: t.Correo_Electronico_turista = 'soyturista@gmail.com': Filtra los resultados para el turista con el correo especificado.

FORMAT(p.Fecha_Compra_Pasaje, 'yyyy-MM') = '2017-09': Asegura que solo se consideren las compras realizadas
durante septiembre de 2017.

GROUP BY p.ID_Viaje: Agrupa los resultados por ID_Viaje para calcular la cantidad de pasajes por viaje.

ORDER BY p.ID_Viaje ASC: Ordena la lista por ID_Viaje en orden ascendente.

*/

SELECT p.ID_Viaje, COUNT(p.ID_Pasaje) AS Cantidad_Pasajes

FROM Pasaje p JOIN Turista t ON p.ID_Turista = t.ID_Turista

WHERE t.Correo_Electronico_turista = 'soyturista@gmail.com'
      AND DATEPART(YEAR, p.Fecha_Compra_Pasaje) = 2017
      AND DATEPART(MONTH, p.Fecha_Compra_Pasaje) = 9

GROUP BY p.ID_Viaje ORDER BY p.ID_Viaje ASC;
```

4.4 Evidencia de ejecución de consultas

Consulta 1:

```
SELECT
    T.ID_Turista,
    T.Primer_Nombre_Turista,
    T.Apellido_Paterno_Turista,
    T.Apellido_Materno_Turista,
    P.Cantidad_Pasajes

FROM Turista T

JOIN
    ( SELECT ID_Turista, COUNT(*) AS Cantidad_Pasajes

      FROM Pasaje GROUP BY ID_Turista HAVING COUNT(*) = (SELECT TOP 1 COUNT(*)

      FROM Pasaje GROUP BY ID_Turista ORDER BY COUNT(*) DESC
    )
) P ON T.ID_Turista = P.ID_Turista;

/*
```

Resultados | Mensajes

	ID_Turista	Primer_Nombre_Turista	Apellido_Paterno_Turista	Apellido_Materno_Turista	Cantidad_Pasajes
1	1	Juan	Pérez	García	11

Consulta 2:

```
GETDATE(): Devuelve la fecha y hora.

*/
SELECT B.* FROM Bus B LEFT JOIN Viaje V ON B.ID_Bus = V.ID_Bus

AND CAST(V.Fecha_Salida_Viaje AS DATE) = CAST(DATEADD(DAY, 1, GETDATE()) AS DATE)

WHERE B.Capacidad_Bus > 35 AND V.ID_Viaje IS NULL;

/*
3. Listar todos los datos de los pasajeros para los cuales haya registrados en el sistema
```

Resultados | Mensajes

	ID_Bus	Marca_Bus	Tipo_Bus	Capacidad_Bus
1	2	Volvo	Autobús de lujo	50

Consulta 3:

```
SELECT T.ID_Turista,
    T.Primer_Nombre_turista,
    T.Apellido_Paterno_turista,
    T.Apellido_Materno_turista,
    T.Numero_Documento_turista,
    T.Contrasenia_turista,
    T.Correo_Electronico_turista,
    T.Telefono_turista,
    T.Tipo_Documento_turista

FROM Turista T

JOIN Pasaje P on P.ID_Turista = T.ID_Turista

GROUP BY T.ID_Turista,
    T.Primer_Nombre_turista,
    T.Apellido_Paterno_turista,
    T.Apellido_Materno_turista,
    T.Numero_Documento_turista,
    T.Contrasenia_turista,
    T.Correo_Electronico_turista,
    T.Telefono_turista,
    T.Tipo_Documento_turista

HAVING COUNT(P.ID_Pasaje)>5
```

Resultados | Mensajes

	ID_Turista	Primer_Nombre_turista	Apellido_Paterno_turista	Apellido_Materno_turista	Numero_Documento_turista	Contrasenia_turista	Correo_Electronico_turista	Telefono_turista	Tipo_Documento_turista
1	1	Juan	Pérez	García	12345678	contraseña123	juan.perez@example.com	987654321	DNI

Consulta 4:

```sql
*/
SELECT  t.ID_Turista,
        t.Primer_Nombre_turista,
        t.Apellido_Paterno_turista,
        t.Apellido_Materno_turista,
        a.ID_Bus,
        a.Numero_Fila_Asiento,
        a.Letra_Columna_Asiento

FROM Pasaje p JOIN Turista t ON p.ID_Turista = t.ID_Turista

    JOIN Asiento a ON p.ID_Bus = a.ID_Bus AND p.Numero_Fila_Asiento = a.Numero_Fila_Asiento

    AND p.Letra_Columna_Asiento = a.Letra_Columna_Asiento

    JOIN Viaje v ON p.ID_Viaje = v.ID_Viaje

WHERE
    v.ID_Viaje = 5; -- Puede ser cualquier valor que se tenga que buscar

/*
```

Resultados | Mensajes

	ID_Turista	Primer_Nombre_turista	Apellido_Paterno_turista	Apellido_Materno_turista	ID_Bus	Numero_Fila_Asiento	Letra_Columna_Asiento
1	1	Juan	Pérez	García	2	8	E

Consulta 5:

```sql
SELECT p.ID_Viaje, COUNT(p.ID_Pasaje) AS Cantidad_Pasajes

FROM Pasaje p JOIN Turista t ON p.ID_Turista = t.ID_Turista

WHERE t.Correo_Electronico_turista = 'soyturista@gmail.com'
      AND DATEPART(YEAR, p.Fecha_Compra_Pasaje) = 2017
      AND DATEPART(MONTH, p.Fecha_Compra_Pasaje) = 9

GROUP BY p.ID_Viaje ORDER BY p.ID_Viaje ASC;
```

Resultados | Mensajes

	ID_Viaje	Cantidad_Pasajes
1	14	1