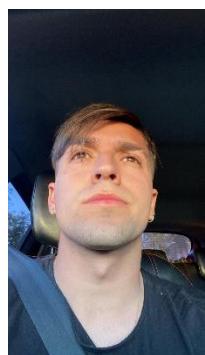


**Universidad ORT Uruguay
Facultad de Ingeniería
Escuela de Tecnología**

OBLIGATORIO

BASE DE DATOS 1



Sebastián Hohl - 327007



Matías Oreiro - 239479

M2E

Docente: Rafael Cohen

Analista en tecnologías de la información / Analista Programador

Fecha de entrega (20/11/2024)

Índice

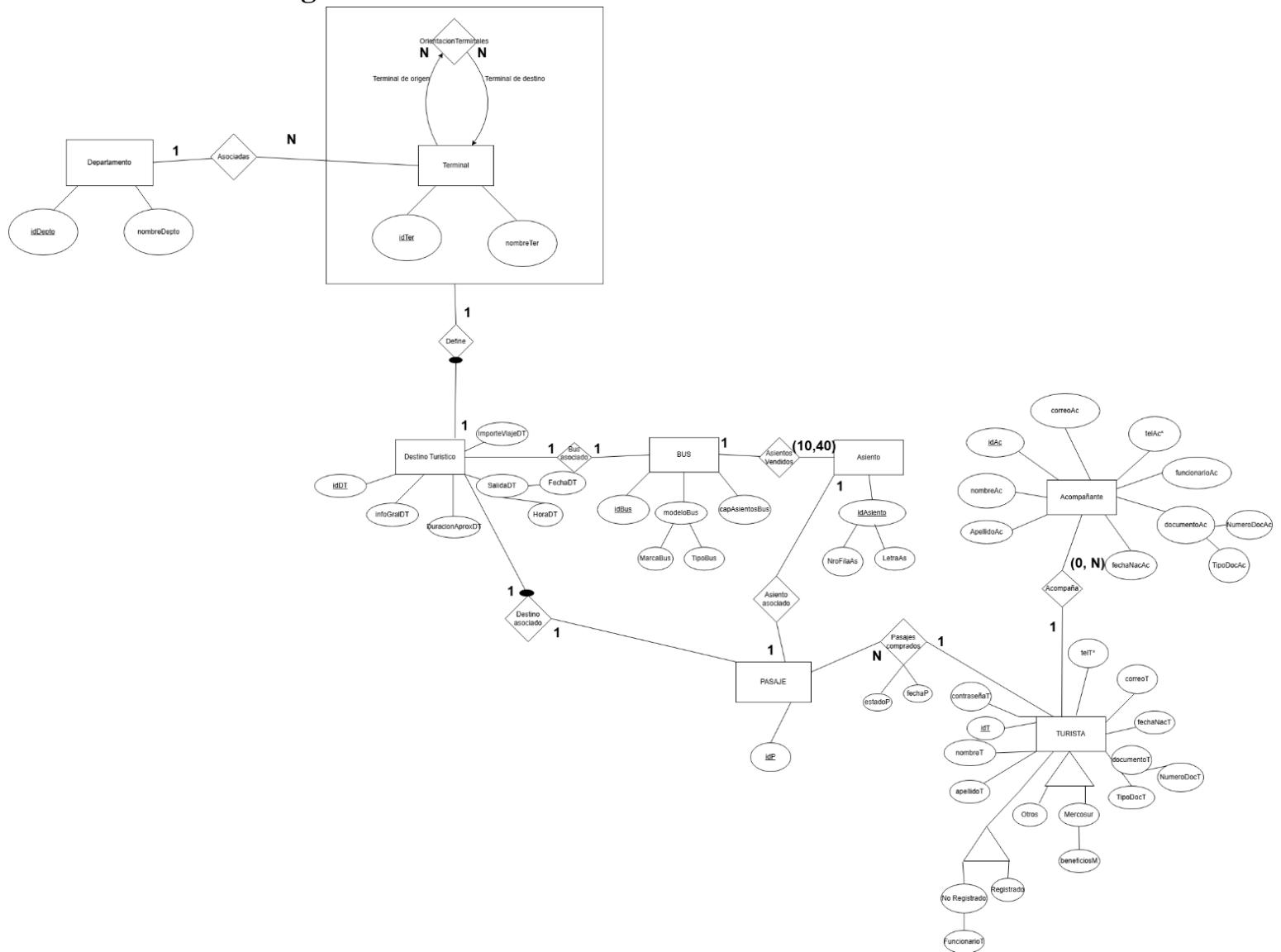
1.	Introducción.....	4
2.	Modelo entidad relación (MER).....	4
2.1	Diagrama Entidad Relación.....	4
2.1	Restricciones no estructurales	5
2.2	Justificaciones y otras consideraciones	5
3.	Modelo relacional (MR)	5
3.1.	Restricciones.....	6
3.2.	Normalización	6
4.	SQL.....	7
4.1	Data Definition Language (DDL)	7
4.2	Data Manipulation Language (DML).....	16
4.3	Consultas SQL.....	23
5.3.1.	Consulta 1	23
5.3.2.	Consulta 2.....	24
5.3.3.	Consulta 3	25
5.3.4.	Consulta 4.....	26
5.3.5.	Consulta 5	27
4.4	Evidencia de ejecución de consultas	27

1. Introducción

[Aquí se espera que se presente una descripción general del problema a resolver]

2. Modelo entidad relación (MER)

2.1 Diagrama Entidad Relación



2.2 Restricciones no estructurales

- 1- En Turista, la contraseña debe tener al menos 8 caracteres.
- 2- En Turista, el idT es un número autoincremental.
- 3- En Turista, el correo debe ser único en la base de datos.
- 4- En la autorelación de Terminal, la terminal de origen y la terminal de destino deben ser diferentes.
- 5- En Pasaje, el idP es un número autoincremental.
- 6- En Pasaje, el estado sólo puede contener los valores “Usado” o “No usado”.
- 7- En Pasaje, la fechaP no puede ser menor a la fecha actual ya que no se pueden comprar ni cancelar pasajes para destinos ya realizados.
- 8- Si el pasaje es comprado por un usuario no registrado o un acompañante que no tenga usuario un funcionario de la compañía registrará los datos del pasajero.

2.2 Justificaciones y otras consideraciones

- 1- Todos los datos ingresados son los que se esperan.
- 2- En el correo se espera un @ y un mail correcto.

3. Modelo relacional (MR)

Turista(idT, idAc, nombreT, apellidoT, documentoT, fechaNacT, correoT, contraseñaT, telT*, registro, mercosur)

Acompañante(idAc, nombreAc, apellidoAc, fechaNacAc, documentoAc, funcionarioAc, telAc*, correoAc)

Pasaje(idP, estadoP, fechaP, idT)

Asiento(idAsiento, idP, idBus)

Bus(idBus, modeloBus, capAsientosBus)

DestinoTuristico(idDT, infoGralDt, duracionAproxDT, salidaDT, importeViajeDT, idBus)

Terminal(idTer, nombreTer, idDepto)

OrientacionTerminales(terminalDeOrigen, terminalDeDestino, idTer)

Departamento(idDepto, nombreDepto)

3.1. Restricciones

- 1- En todas las entidades, los id son únicos.

3.2. Normalización

<<Incluir conversión si corresponde y su justificación>>

Turista (idT(PK), nombreT, apellidoT, correoT, contraseñaT, tipoDocT, numeroDocT, fechaNacT)

Telefono (idT (FK), telT (PK))

NoRegistrado (idT(FK), funcionarioT)

Mercosur (idT(FK), beneficiosM)

Acompañante (idAc (PK), idT (FK), nombreAc, apellidoAc, correoAc, tipoDocAc, numeroDocAc, fechaNacAc, funcionarioAc)

TelefonoAc (idAc (FK), telAc (PK))

Pasaje (idP(PK), idT(FK), NroFilaAs(FK), LetraAs(FK), idDT(FK), estadoP, fechaP)

Asiento (NroFilaAs(PK), LetraAs(PK), idBus(FK))

Bus (idBus(PK), idDT(FK), MarcaBus, TipoBus, capAsientosBus)

DestinoTuristico (idDT(PK), idTerOrigen(FK), idTerDestino(FK), infoGralDT, DuracionAproxDT, FechaDT, HoraDT, ImporteViajeDT)

Terminal (idTer(PK), idDept(PK), nombreTer)

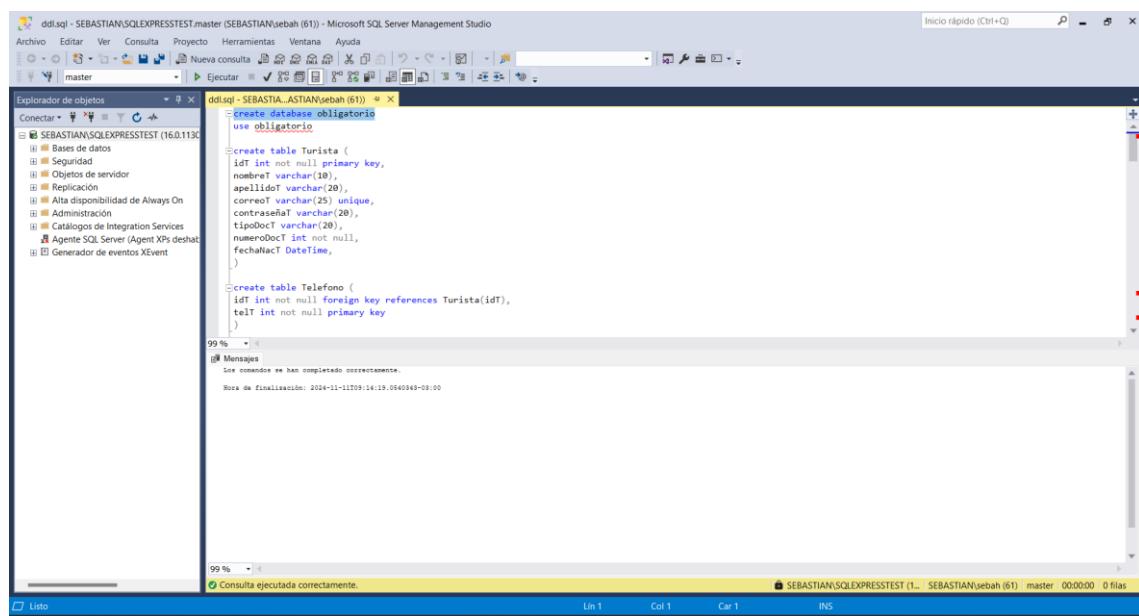
OrientacionTerminales (idTerOrigen(FK), idTerDestino(FK))

Departamento (idDept(PK), nombreDept)

4. SQL

4.1 Data Definition Language (DDL)

<<Incluir evidencia de ejecución de script entregado (ddl.sql) que genera la estructura de la base de datos>>



The screenshot shows the Microsoft SQL Server Management Studio interface. The title bar reads "ddl.sql - SEBASTIAN\SQLEXPRESSTEST.master (SEBASTIAN\sebah (61)) - Microsoft SQL Server Management Studio". The main window displays the following DDL script:

```
create database obligatorio
use obligatorio

create table Turista (
    idT int primary key,
    nombreT varchar(10),
    apellidoT varchar(20),
    correoT varchar(25) unique,
    contraseñaT varchar(20),
    tipoDocT varchar(20),
    numeroDocT int not null,
    fechaNacT DateTime
)

create table Telefono (
    idT int not null foreign key references Turista(idT),
    telfT int not null primary key
)
```

Below the script, the status bar indicates "Consulta ejecutada correctamente." and "0 filas". The bottom right corner shows the page number "7".

The screenshot shows the Microsoft SQL Server Management Studio (SSMS) interface. The title bar reads "ddl.sql - SEBASTIAN\SQLEXPRESSTEST obligatorio (SEBASTIAN\uebah (61)) - Microsoft SQL Server Management Studio". The menu bar includes Archivo, Editar, Ver, Consulta, Proyecto, Herramientas, Ventana, Ayuda. The toolbar has icons for New Query, Run, Stop, Refresh, and Save. The left sidebar is titled "Explorador de objetos" and shows the database structure: "SEBASTIAN\SQLEXPRESSTEST (16.0.1130)" with its objects: Bases de datos, Replicación, Seguridad, Replicación, Alta disponibilidad de Always On, Administración, Catalogos de Integration Services, Agente SQL Server (Agent XPs deshabilitado), and Generador de eventos XEvent. The main pane displays the following T-SQL code:

```
CREATE DATABASE obligatorio
USE obligatorio

CREATE TABLE Turista (
    idT int not null primary key,
    nombreT nvarchar(20),
    apellidoT varchar(20),
    correoT varchar(25) unique,
    contraseñaT varchar(20),
    tipoDocT varchar(20),
    numeroDocT int not null,
    fechaNacT DateTime,
    )

CREATE TABLE Telefono (
    idT int not null foreign key references Turista(idT),
    telT int not null primary key
)
```

The status bar at the bottom indicates "Consulta ejecutada correctamente." and "SEBASTIAN\SQLEXPRESSTEST (1... SEBASTIAN\uebah (61)) obligatorio 000000 0 filas".

File Edit View Query Project Tools Window Help

Quick Launch (Ctrl+Q)

File New Query

Execute

Object Explorer

Connect

MATIAS\SQLPRESTEST (SQL Server)

obligatorio

dd1.sql - MATIAS\JMAATA\matia (S1) - Microsoft SQL Server Management Studio

```
CREATE DATABASE obligatorio
USE obligatorio

CREATE TABLE Turista (
    idT INT NOT NULL PRIMARY KEY,
    nombre VARCHAR(20),
    apellido1 VARCHAR(20),
    apellido2 VARCHAR(20),
    correct VARCHAR(25) UNIQUE,
    contrasena VARCHAR(20),
    tipoDoc VARCHAR(20),
    numeroDoc INT NOT NULL,
    fechaNac DATE
)

CREATE TABLE Telefono (
    idT INT NOT NULL FOREIGN KEY REFERENCES Turista(idT),
    tel1 INT NOT NULL PRIMARY KEY
)

CREATE TABLE Nacimiento (
    idT INT NOT NULL FOREIGN KEY REFERENCES Turista(idT),
    Nacionalidad VARCHAR(10)
)

CREATE TABLE Mercador (
    idT INT NOT NULL FOREIGN KEY REFERENCES Turista(idT),
     100 %
)
```

Messages

Command completed successfully.

Completion time: 2024-11-11T09:49:57.9027268-08:00

100 % 

Query executed successfully.

MATIAS\SQLPRESTEST (16...) | MATIAS\matia (S1) | obligatorio | 000000 | 0 rows

The screenshot shows the Microsoft SQL Server Management Studio interface. The title bar reads "ddi.sql - SEBASTIAN\SOLEXPRESSTEST obligatorio (SEBASTIAN\sebah (61)) - Microsoft SQL Server Management Studio". The left pane is the "Explorador de objetos" (Object Explorer) showing the database structure of "SEBASTIAN\SOLEXPRESSTEST (16.0.1130)". The right pane contains the SQL script "ddi.sql" with the following content:

```
apellidoT varchar(20),
correoElectronicoT(25) unique,
contraseñaT varchar(20),
tipoDocT varchar(20),
numeroDocT int not null,
fechaNacT DateTime,
)

create table Telefono (
idT int not null foreign key references Turista(idT),
telT int not null primary key
)

create table NoRegistrado (
idT int not null foreign key references Turista(idT),
funcionarioT varchar(10)
)

99 % ▶ 4
0 Mensajes
Los comandos se han completado correctamente.
Hora de finalización: 2024-11-11T09:16:32.9276172-03:00
```

The status bar at the bottom indicates "Consulta ejecutada correctamente." and "0 filas".

The screenshot shows the Microsoft SQL Server Management Studio interface. The title bar reads "ddi.sql - SEBASTIAN\SOLEXPRESSTEST obligatorio (SEBASTIAN\sebah (61)) - Microsoft SQL Server Management Studio". The left pane is the "Explorador de objetos" (Object Explorer) showing the database structure of "SEBASTIAN\SOLEXPRESSTEST (16.0.1130)". The right pane contains the SQL script "ddi.sql" with the following content:

```
tipodocT varchar(20),
numeroDocT int not null,
fechaNacT DateTime,
)

create table Telefono (
idT int not null foreign key references Turista(idT),
telT int not null primary key
)

create table NoRegistrado (
idT int not null foreign key references Turista(idT),
funcionarioT varchar(10)
)

create table Mercosur (
idT int not null foreign key references Turista(idT),
beneficiosM varchar(100)
)

99 % ▶ 4
0 Mensajes
Los comandos se han completado correctamente.
Hora de finalización: 2024-11-11T09:16:51.0022888-03:00
```

The status bar at the bottom indicates "Consulta ejecutada correctamente." and "0 filas".

File Edit View Query Project Tools Window Help

Quick Launch (Ctrl+Q)

Object Explorer

Connections: MÁTIAS\SQLEXPRESSTEST (MÁTIAS\maita [1]) - Microsoft SQL Server Management Studio

New Query New Results New Script New Object Explorer New Task List New File New Database New Table New View New StoredProcedure New Function New Index New Constraint New Rule New Rule Set New Trigger New Rule Set

Execute

obligatorio

obsp - MÁTIAS\SQLEXPRESSTEST (MÁTIAS\maita [1]) - >

```
funcionarioID int primary key,
funcionarioNombre varchar(10),
funcionarioApellido varchar(10),
funcionarioDNI int unique,
funcionarioEmail varchar(50),
funcionarioTeléfono varchar(15),
funcionarioContraseña varchar(25),
funcionarioEstado bit,
funcionarioFechaAlta date,
funcionarioUltimaModificacion date,
```

create table Funcionario (
 funcionarioID int primary key,
 funcionarioNombre varchar(10),
 funcionarioApellido varchar(10),
 funcionarioDNI int unique,
 funcionarioEmail varchar(50),
 funcionarioTeléfono varchar(15),
 funcionarioContraseña varchar(25),
 funcionarioEstado bit,
 funcionarioFechaAlta date,
 funcionarioUltimaModificacion date,

```
funcionarioID int primary key,
funcionarioNombre varchar(10),
funcionarioApellido varchar(10),
funcionarioDNI int unique,
funcionarioEmail varchar(50),
funcionarioTeléfono varchar(15),
funcionarioContraseña varchar(25),
funcionarioEstado bit,
funcionarioFechaAlta date,
funcionarioUltimaModificacion date,
```

create table Mercosur (
 idf int not null foreign key references Turista(idT),
 beneficiarioID int not null,
 beneficiarioNombre varchar(100),
 beneficiarioApellido varchar(100),
 beneficiarioDNI int unique,
 beneficiarioEmail varchar(50),
 beneficiarioTeléfono varchar(15),
 beneficiarioContraseña varchar(25),
 beneficiarioEstado bit,
 beneficiarioFechaAlta date,
 beneficiarioUltimaModificacion date,

create table Acompañante (
 idAc int not null primary key,
 idf int not null foreign key references Turista(idT),
 nombreAc varchar(10),
 apellidoAc varchar(20),
 correoAc varchar(25) unique,
 telefonoAc varchar(15),
 numeroDocAc int not null,
 fechaNacAc Date,
 funcionarioAcompañanteID int not null,
 funcionarioAcompañanteNombre varchar(10),
 funcionarioAcompañanteApellido varchar(10),
 funcionarioAcompañanteDNI int unique,
 funcionarioAcompañanteEmail varchar(50),
 funcionarioAcompañanteTeléfono varchar(15),
 funcionarioAcompañanteContraseña varchar(25),
 funcionarioAcompañanteEstado bit,
 funcionarioAcompañanteFechaAlta date,
 funcionarioAcompañanteUltimaModificacion date,

create table TelefonoAc (
 idTeleAc int not null foreign key references Acompañante(idAc),
 telAc int not null primary key
)

create table Departamento (
 idDep int primary key,
 nombreDep varchar(50) unique,
 descripcionDep varchar(250),
 ubicacionDep varchar(100),
 telefonoDep varchar(15),
 emailDep varchar(50),
 webDep varchar(100),
 contactoDep varchar(100),
 fechaNacDep Date,
 funcionarioDepID int not null,
 funcionarioDepNombre varchar(10),
 funcionarioDepApellido varchar(10),
 funcionarioDepDNI int unique,
 funcionarioDepEmail varchar(50),
 funcionarioDepTeléfono varchar(15),
 funcionarioDepContraseña varchar(25),
 funcionarioDepEstado bit,
 funcionarioDepFechaAlta date,
 funcionarioDepUltimaModificacion date
)

Messages

Command completed successfully.

Completion time: 2024-11-11T09:44:48.499941-08:00

100 %

Query executed successfully.

ddl.sql - SEBASTIAN\SELEXPRESSTEST obligatorio (SEBASTIAN\sebah (61)) - Microsoft SQL Server Management Studio

Explorador de objetos

ddl.sql - SEBASTIAN\SELEXPRESSTEST (16.0.1130)

- Bases de datos
- Seguridad
- Objetos de servidor
- Replicación
- Alta disponibilidad de Always On
- Administración
- Catálogos de Integration Services
- Agente SQL Server (Agent XPs deshabilitado)
- Generador de eventos XEvent

```

CREATE TABLE Acompanante (
    idAcom int not null primary key,
    apellidoAcom varchar(20),
    nombreAcom varchar(25) unique,
    tipoDocAcom int not null,
    numeroDocAcom int not null,
    fechaNacAcom DateTime,
    funcionarioAcom varchar(10)
)

CREATE TABLE TelefonoAcom (
    idAcom int not null foreign key references Acompanante(idAcom),
    telAcom int not null primary key
)

CREATE TABLE Departamento (
    idDepto int not null primary key,
    nombreDepto varchar(20)
)

```

99 %

Mensajes

Los comandos se han completado correctamente.

Hora de finalización: 2024-11-11T09:16:39.0215883-03:00

Consulta ejecutada correctamente.

Lin 42 Col 1 Car 1 INS

ddl.sql - SEBASTIAN\SELEXPRESSTEST obligatorio (SEBASTIAN\sebah (61)) - Microsoft SQL Server Management Studio

Explorador de objetos

ddl.sql - SEBASTIAN\SELEXPRESSTEST (16.0.1130)

- Bases de datos
- Seguridad
- Objetos de servidor
- Replicación
- Alta disponibilidad de Always On
- Administración
- Catálogos de Integration Services
- Agente SQL Server (Agent XPs deshabilitado)
- Generador de eventos XEvent

```

CREATE TABLE TelefonoAcom (
    idAcom int not null foreign key references Acompanante(idAcom),
    telAcom int not null primary key
)

CREATE TABLE Departamento (
    idDepto int not null primary key,
    nombreDepto varchar(20)
)

CREATE TABLE Terminal (
    idTer int not null primary key,
    idDepto int foreign key references Departamento(idDepto),
    nombreTer varchar(20)
)

```

99 %

Mensajes

Los comandos se han completado correctamente.

Hora de finalización: 2024-11-11T09:16:51.0061812-03:00

Consulta ejecutada correctamente.

Lin 47 Col 1 Car 1 INS

```
ddl.sql - SEBASTIAN\SQLEXPRESSTEST obligatorio (SEBASTIAN\sebah (61)) - Microsoft SQL Server Management Studio

Explorador de objetos
Conectar
Nueva consulta Ejecutar
Explorador de objetos
ddl.sql - SEBASTIAN\ASTIAN(sebah (61)) - x

[+] SEBASTIAN\SQLEXPRESSTEST (16.0.113C)
  [+] Bases de datos
  [+] Seguridad
  [+] Objetos del servidor
  [+] Replicación
  [+] Alta disponibilidad de Always On
  [+] Administración
  [+] Catalogos de Integration Services
  [+] Agente SQL Server (Agent XPs deshabilitado)
  [+] Generador de eventos XEvent

CREATE TABLE Departamento (
    idDepto int NOT NULL PRIMARY KEY,
    nombreDepto VARCHAR(20)
)

CREATE TABLE Terminal (
    idTer int NOT NULL PRIMARY KEY,
    idDepto int FOREIGN KEY REFERENCES Departamento(idDepto),
    nombreTer VARCHAR(20)
)

CREATE TABLE OrientacionTerminales (
    idTerOrigen int FOREIGN KEY REFERENCES Terminal(idTer),
    idTerDestino int FOREIGN KEY REFERENCES Terminal(idTer),
    primary key(idTerOrigen, idTerDestino)
)

Mensajes
Las comandos se han completado correctamente.

Hora de finalización: 2024-11-11T09:17:04.9495660-08:00
```

The screenshot shows the Microsoft SQL Server Management Studio interface. The title bar reads "ddl.sql - SEBASTIAN\SQLEXPRESSTEST. obligatorio (SEBASTIAN\sebah (61)) - Microsoft SQL Server Management Studio". The left pane is the "Explorador de objetos" (Object Explorer) showing the database structure. The right pane is the "Query Editor" with the following SQL script:

```
create table Departamento (
    idDepto int not null primary key,
    nombreDepto varchar(20)
)

create table Terminal (
    idTer int not null primary key,
    idDepto int foreign key references Departamento(idDepto),
    nombreTer varchar(20)
)

create table OrientacionTerminales (
    idTerOrigen int foreign key references Terminal(idTer),
    idTerDestino int foreign key references Terminal(idTer),
    primary key(idTerOrigen, idTerDestino)
)
```

The status bar at the bottom indicates "Consulta ejecutada correctamente." (Query executed successfully.)

ddl.sql - MATIAS\SQLEXPRESSTEST\obligatorio (MATIAS\matia (5)) - Microsoft SQL Server Management Studio

```

CREATE TABLE Terminal (
    idTer INT NOT NULL PRIMARY KEY,
    idDept INT FOREIGN KEY REFERENCES Departamento(idDept),
    nombreTer VARCHAR(20)
)

CREATE TABLE OrientacionTerminales (
    idTerOrigen INT FOREIGN KEY REFERENCES Terminal(idTer),
    idTerDestino INT FOREIGN KEY REFERENCES Terminal(idTer),
    primary key(idTerOrigen, idTerDestino)
)

CREATE TABLE DestinoTuristico (
    idOT INT NOT NULL PRIMARY KEY,
    idTerOrigen INT NOT NULL,
    idTerDestino INT NOT NULL,
    infoturistica VARCHAR(100),
    infoturistaID INT NOT NULL,
    DuracionPromedioOT INT NOT NULL,
    PrecioPromedioOT INT NOT NULL,
    HorarioOT TIME,
    ImportanciaOT INT NOT NULL,
    FOREIGN KEY (idTerOrigen, idTerDestino) REFERENCES OrientacionTerminales(idTerOrigen, idTerDestino)
)

CREATE TABLE Bus (
    idBus INT NOT NULL PRIMARY KEY,
    ...
)

```

100 %

Messages

Command completed successfully.

Completion time: 2024-11-11T09:46:20.1468294-03:00

Ready

Query executed successfully.

Ln 54 Col 1 Ch 1 INS

MATIAS\SQLEXPRESSTEST (16.0...) MATIAS\matia (5) obligatorio 00:00:00 0 rows

ddl.sql - SEBASTIAN\SQLEXPRESSTEST\obligatorio (SEBASTIAN\sebah (61)) - Microsoft SQL Server Management Studio

```

CREATE TABLE DestinoTuristico (
    idOT INT NOT NULL PRIMARY KEY,
    HoraDT VARCHAR(10),
    ImportanteOT INT NOT NULL,
    FOREIGN KEY (idTerOrigen, idTerDestino) REFERENCES OrientacionTerminales(idTerOrigen, idTerDestino)
)

CREATE TABLE Bus (
    idBus INT NOT NULL PRIMARY KEY,
    idOT INT NOT NULL FOREIGN KEY REFERENCES DestinoTuristico(idOT),
    MarcaBus VARCHAR(20),
    TipoBus VARCHAR(20),
    capacidadSilla INT NOT NULL
)

CREATE TABLE Asiento (
    NroAsiento INT NOT NULL,
    LetraAsiento CHAR(1),
    idBus INT NOT NULL FOREIGN KEY REFERENCES Bus(idBus),
    ...
)

```

99 %

Mensajes

Los comandos se han completado correctamente.

Hora de finalización: 2024-11-11T09:17:50.4464149-08:00

Consulta ejecutada correctamente.

Ln 76 Col 1 Ch 1 INS

SEBASTIAN\SQLEXPRESSTEST (1...) SEBASTIAN\sebah (61) obligatorio 00:00:00 0 filas

File Edit View Query Project Tools Window Help

New Query

Execute

Object Explorer

Connect to Server

MATIAS\SQLPRESTEST (SQL Server)

MATIAS\matia (S1) ->

```
CREATE TABLE MarcaBus
(
    MarcaBus VARCHAR(20),
    TiposBus VARCHAR(20),
    CapacidadSilla INT NOT NULL
)

CREATE TABLE Asiento
(
    NroFilas INT NOT NULL,
    LetraL INT NOT NULL,
    IdAsiento INT NOT NULL,
    FOREIGN KEY REFERENCES Bus(IdBus),
    PRIMARY KEY(NroFilas, LetraL)
)

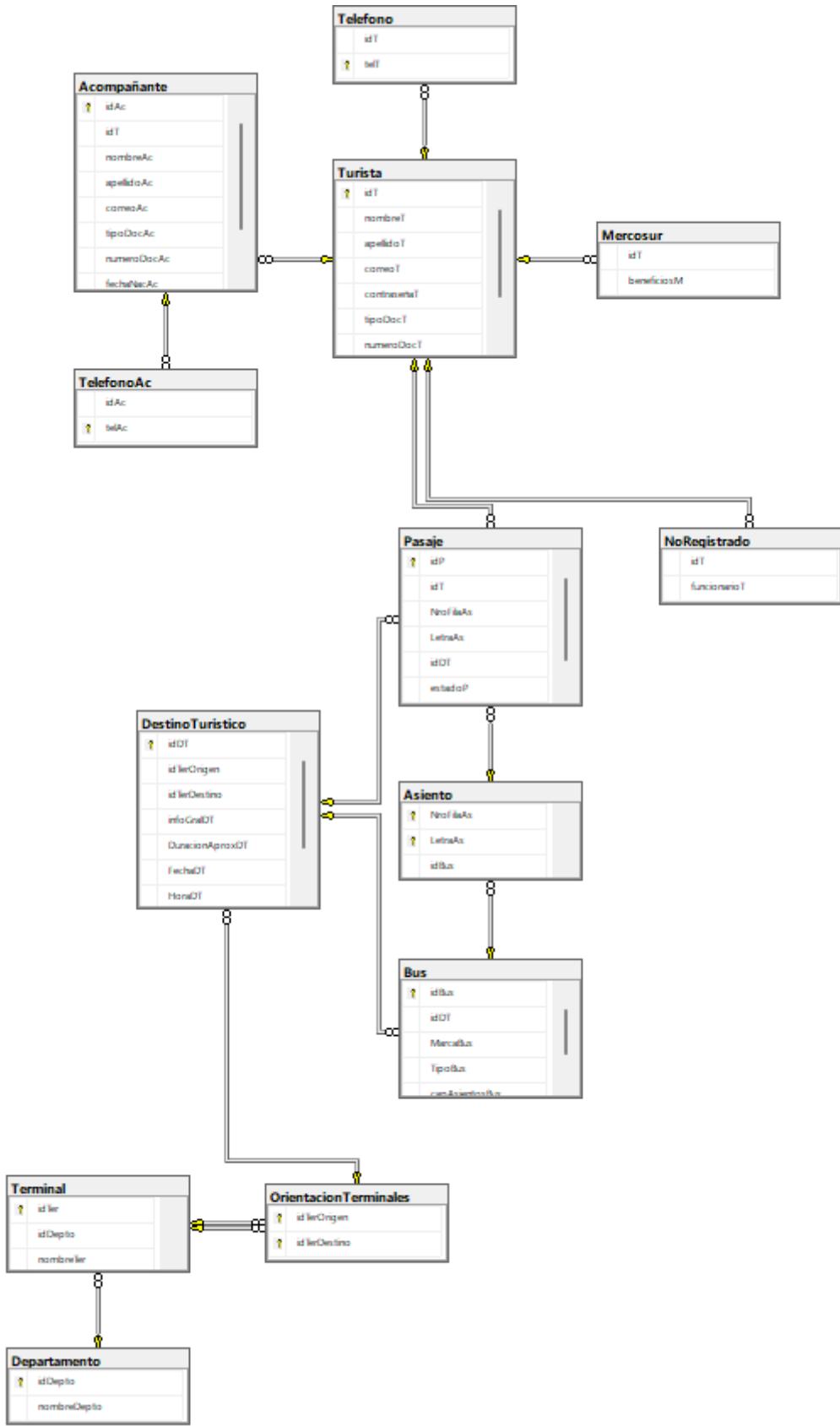
CREATE TABLE Pasaje
(
    IdP INT NOT NULL PRIMARY KEY,
    IdB INT NOT NULL,
    NroFilas INT NOT NULL,
    LetraL VARCHAR(1),
    Asiento INT NOT NULL,
    Estado VARCHAR(10),
    FechaH DATE,
    Turista INT,
    FOREIGN KEY (IdB) REFERENCES Turista(IdT),
    FOREIGN KEY (IdT) REFERENCES DestinoTuristico(IdD),
    FOREIGN KEY (NroFilas, LetraL) REFERENCES Asiento(NroFilas, LetraL)
)
```

Messages

Command completed successfully.

Completion time: 2024-11-11T09:45:52.4888999-08:00

Query executed successfully.



4.2 Data Manipulation Language (DML)

<< Incluir evidencia de ejecución del script entregado (dml.sql) que genera los datos de prueba en la base de datos>>

The screenshot shows a ChatGPT interface with a sidebar containing various topics like ChatGPT, Product Owner, Explorar GPT, Hoy, etc. The main area displays 8 examples of SQL INSERT statements for a 'Turista' table. Below the examples is an explanation of the fields:

Ejemplos de inserciones en SQL Server

```
insert into Turista (idT, nombreT, apellidoT, correoT, contraseñaT, tipoDocT, numeroDocT, fechaNacT)
VALUES
(1, 'Carlos', 'Pérez', 'carlos.perez@example.com', 'Password123', 'DNI', '12345678', '1999-01-01'),
(2, 'María', 'Gómez', 'maria.gomez@example.com', 'SecurePass123', 'PAS', 'M12345678', '1985-05-20'),
(3, 'Javier', 'Rodríguez', 'javier.rodriguez@example.com', 'JavaPass789', 'DNI', '07894321', '1995-03-20'),
(4, 'Ana', 'López', 'ana.lopez@example.com', 'LopezPass001', 'NIE', 'Y12345678', '1995-03-20'),
(5, 'Lucía', 'Martínez', 'lucia.martinez@example.com', 'LucyPass123', 'DNI', '23456789', '1995-03-20'),
(6, 'Pedro', 'Hernández', 'pedro.hernandez@example.com', 'PedroPass123', 'PAS', '56789432', '1985-07-15'),
(7, 'Sofía', 'Torres', 'sofia.torres@example.com', 'TorresPass2004', 'DNI', '34567890', '2004-06-15'),
(8, 'Miguel', 'Ramírez', 'miguel.ramirez@example.com', 'Miguel999', 'NIE', 'X2345678A', '1999-01-01');
```

Explicación de los campos:

- **idT**: ID del turista (entero).
- **nombreT**: Nombre del turista.
- **apellidoT**: Apellido del turista.
- **correoT**: Correo electrónico (debe ser único si tienes esa restricción).
- **contraseñaT**: Contraseña del turista.

The screenshot shows the SSMS interface with a query window titled 'SQLQueryLog - MATIAS\SQLEXPRESSTEST\obligatorio (MATIAS)\matia (70)'.

The script 'dml.sql' contains two parts:

```
-- Carga de datos de Turistas
use obligatorio
go
-- Ejemplo de uso de insert
insert into Turista (idT, nombreT, apellidoT, correct, contraseñaT, tipoDocT, numeroDocT, fechaNacT) values (1, 'Pedro', 'Hernández', 1, 'Pedro', 'Hernández', 'DNI', '12345678', '1999-01-01')
insert into Turista (idT, nombreT, apellidoT, correct, contraseñaT, tipoDocT, numeroDocT, fechaNacT) values (2, 'Sofía', 'Torres', 1, 'Sofía', 'Torres', 'DNI', '23456789', '2004-06-15')
insert into Turista (idT, nombreT, apellidoT, correct, contraseñaT, tipoDocT, numeroDocT, fechaNacT) values (3, 'Juan', 'Martínez', 1, 'Juan', 'Martínez', 'DNI', '34567890', '1995-03-20')
insert into Turista (idT, nombreT, apellidoT, correct, contraseñaT, tipoDocT, numeroDocT, fechaNacT) values (4, 'Ana', 'López', 1, 'Ana', 'López', 'DNI', '45678901', '1995-03-20')
insert into Turista (idT, nombreT, apellidoT, correct, contraseñaT, tipoDocT, numeroDocT, fechaNacT) values (5, 'Lucía', 'Martínez', 1, 'Lucía', 'Martínez', 'DNI', '56789432', '1995-03-20')
insert into Turista (idT, nombreT, apellidoT, correct, contraseñaT, tipoDocT, numeroDocT, fechaNacT) values (6, 'Pedro', 'Hernández', 1, 'Pedro', 'Hernández', 'DNI', '56789432', '1985-07-15')
insert into Turista (idT, nombreT, apellidoT, correct, contraseñaT, tipoDocT, numeroDocT, fechaNacT) values (7, 'Miguel', 'Ramírez', 1, 'Miguel', 'Ramírez', 'DNI', '56789432', '1999-01-01')
insert into Turista (idT, nombreT, apellidoT, correct, contraseñaT, tipoDocT, numeroDocT, fechaNacT) values (8, 'Andrés', 'Gómez', 1, 'Andrés', 'Gómez', 'DNI', '67890123', '1997-09-27')

-- Carga de datos de Teléfonos
use obligatorio
go
insert into Telefono (idT, telT) values (1, 987654321)
insert into Telefono (idT, telT) values (2, 989435712)
insert into Telefono (idT, telT) values (3, 989559723)
insert into Telefono (idT, telT) values (4, 989585719)
insert into Telefono (idT, telT) values (5, 989595713)
insert into Telefono (idT, telT) values (6, 989565724)
insert into Telefono (idT, telT) values (7, 989565716)
insert into Telefono (idT, telT) values (8, 989525716)
insert into Telefono (idT, telT) values (9, 989515711)
insert into Telefono (idT, telT) values (10, 9893458812)
```

The execution results show 10 rows affected, indicating successful insertion of 10 rows of data into the 'Turista' and 'Telefono' tables.

SQL Server Object Explorer

File Edit View Query Project Tools Window Help

Quick Launch (Ctrl+Q)

Object Explorer

Comment - M-MATIAS\matia (70) = < >

MATIAS\SQLPRESTEST (SQL Server)

Databases System Databases Database Snapshots Logins ALGEBRA_SQL EjClase obligatorio Ejercicio Security Server Objects Applications Always On High Availability Management Integration Services Catalogs SQL Server Agent (Agent XPS disabled) XEvent Profiler

SQLQueryLog - M-MATIAS\matia (70) = < >

```
insert into Telefono (IdT, telT) values (3, 996375713)
insert into Telefono (IdT, telT) values (4, 897387719)
insert into Telefono (IdT, telT) values (5, 988359239)
insert into Telefono (IdT, telT) values (5, 988599713)
insert into Telefono (IdT, telT) values (6, 893575714)
insert into Telefono (IdT, telT) values (7, 988359250)
insert into Telefono (IdT, telT) values (8, 897387719)
insert into Telefono (IdT, telT) values (8, 894325716)
insert into Telefono (IdT, telT) values (9, 992337711)
insert into Telefono (IdT, telT) values (10, 893349812)

-- Carga de datos de No registrado
insert into NoRegistrado (IdT, funcionarioID) values (7, 'Ana')
insert into NoRegistrado (IdT, funcionarioID) values (8, 'Ramon')
insert into NoRegistrado (IdT, funcionarioID) values (10, 'Jose')

-- Carga de datos de Mercosur
insert into Mercosur (IdT, beneficiosM) values (1, '10% de descuento en restaurantes')
insert into Mercosur (IdT, beneficiosM) values (5, '20% de descuento en excursiones')
insert into Mercosur (IdT, beneficiosM) values (7, '15% de descuento en tienda de ropa')
insert into Mercosur (IdT, beneficiosM) values (9, 'Seguro de viaje gratuito')

Completion time: 2024-11-17T23:27:36.8158872-09:00
```

100 % - < >

Messages

(1 row affected)

(1 row affected)

(1 row affected)

Query executed successfully.

SQLQueryLog - MATIAS\SQLPRESTEST\obligatorio (MATIAS\matia (70)) - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

New Query Object Explorer Task List Object Explorer Properties

... Execute

SQLQueryLog - M..MATIAS\matia (70) = X

```
Insert into Telefono (idT, telf) values (3, 896375213)
Insert into Telefono (idT, telf) values (4, 897385739)
Insert into Telefono (idT, telf) values (4, 894669329)
Insert into Telefono (idT, telf) values (5, 898937513)
Insert into Telefono (idT, telf) values (6, 895375184)
Insert into Telefono (idT, telf) values (7, 895365716)
Insert into Telefono (idT, telf) values (8, 897887719)
Insert into Telefono (idT, telf) values (9, 892315715)
Insert into Telefono (idT, telf) values (9, 892315711)
Insert into Telefono (idT, telf) values (10, 893345812)

-- Carga de datos de NoRegistrado
Insert into NoRegistrado (idT, funcionarioID) values (7, 'Juan')
Insert into NoRegistrado (idT, funcionarioID) values (8, 'Ramon')
Insert into NoRegistrado (idT, funcionarioID) values (10, 'Jose')

-- Carga de datos de Mercosur
Insert into Mercosur (idT, beneficiosM) values (1, '10% de descuento en restaurantes')
Insert into Mercosur (idT, beneficiosM) values (5, '20% de descuento en excursiones')
Insert into Mercosur (idT, beneficiosM) values (7, '15% de descuento en tienda de ropa')
Insert into Mercosur (idT, beneficiosM) values (9, 'Seguro de viaje gratuito')

100% + 4
Messages

(1 row affected)
(1 row affected)
(1 row affected)
(1 row affected)

Completion time: 2024-11-17T23:27:48.5441111-08:00
```

Ready

Query executed successfully.

SQLQuery1 - MATIAS\SQLEXPRESSTEST\obligatorio (MATIAS\matia70) - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

Quick Launch (Ctrl+Q)

Object Explorer

Connections > MATIAS\SQLEXPRESSTEST (SQL Server) > Databases > master > obligatorio

SQLQuery1 [M..MATIAS\matia70] = <execute>

```
Insert into Mercador (idT, beneficioM) values (5, '20% de descuento en excursiones')
Insert into Mercador (idT, beneficioM) values (7, '10% de descuento en tienda de ropa')
Insert into Mercador (idT, beneficioM) values (9, 'Seguro de viaje gratuito')

-- Carga de datos de Acoplamientos
Insert into Acoplamiento (idBc, idC, nombreAc, apellidoAc, correosAc, tipoDocAc, numeroDocAc, fechaNacAc, funcionarioAc) values 1, 2, 'Carlos', 'Sanchez', 'Csanchez@gmail.com', 'CI', 34089843, '10-05-1990', 'Alberto'
Insert into Acoplamiento (idBc, idC, nombreAc, apellidoAc, correosAc, tipoDocAc, numeroDocAc, fechaNacAc, funcionarioAc) values 2, 2, 'Ana', 'Sanchez', 'Apears@gmail.com', 'CI', 44039441, '17-08-1993', 'Alberto'
Insert into Acoplamiento (idBc, idC, nombreAc, apellidoAc, correosAc, tipoDocAc, numeroDocAc, fechaNacAc, funcionarioAc) values 3, 7, 'Jose', 'Diez', 'Jdiez@gmail.com', 'CI', 23039444, '11-02-1981', 'retira'
Insert into Acoplamiento (idBc, idC, nombreAc, apellidoAc, correosAc, tipoDocAc, numeroDocAc, fechaNacAc, funcionarioAc) values 4, 9, 'Quito', 'Rodríguez', 'Qrodriguez@gmail.com', 'CI', 15534567, '12-08-1985', 'retira'
Insert into Acoplamiento (idBc, idC, nombreAc, apellidoAc, correosAc, tipoDocAc, numeroDocAc, fechaNacAc, funcionarioAc) values 5, 10, 'Hector', 'Vigilozas', 'Hvigelozas@gmail.com', 'CI', 57585549, '25-08-2007', 'Sebastián'

-- Carga de datos de Telefono Acoplamiento
Insert into TelefonoAc (idBc, telAc) values (1, 0930399863)
Insert into TelefonoAc (idBc, telAc) values (2, 094728791)
Insert into TelefonoAc (idBc, telAc) values (3, 096516569)
Insert into TelefonoAc (idBc, telAc) values (4, 097154596)
Insert into TelefonoAc (idBc, telAc) values (5, 099372467)

-- Carga de datos de Departamento
Insert into Departamento (idBpto, nombreDept) values (1, 'Salta')

Completion time: 2024-11-17T23:19:20.8800000-03:00

100 % - 4 rows
Messages
```

1 row affected

Query executed successfully.

SQLQuery1.sql - MATIAS\SQLPRESTEST\obligatorio (MATIAS\matia70) - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

Quick Launch (Ctrl+Q)

Object Explorer

Comment - V obligatorio

MATIAS\SQLPRESTEST (SQL Server)

- Databases
- System Databases
- File and Filegroups
- Algebria
- ALGEBRIA_SQL
- Ej1Clase
- obligatorio
- Security
- Server Objects
- Always On Availability
- Management
- Integration Services Catalogs
- SQL Server Agent (Agent XPs disabled)
- XE Event Profiler

SQLQuery1.sql - M..MATIAS\matia70 [1] - >

```
insert into TelefonoAc (idAc, telAc) values (2, 094728791)
insert into TelefonoAc (idAc, telAc) values (3, 09511569)
insert into TelefonoAc (idAc, telAc) values (4, 09511556)
insert into TelefonoAc (idAc, telAc) values (5, 099372467)
insert into TelefonoAc (idAc, telAc) values (5, 094688767)
insert into TelefonoAc (idAc, telAc) values (5, 098245364)

-- Carga de datos de Departamento
insert into Departamento (idDept, nombreDept) values (1, 'Salvo')
insert into Departamento (idDept, nombreDept) values (2, 'Montealegre')
insert into Departamento (idDept, nombreDept) values (3, 'Payandé')
insert into Departamento (idDept, nombreDept) values (4, 'Artigas')
insert into Departamento (idDept, nombreDept) values (5, 'Canelones')
insert into Departamento (idDept, nombreDept) values (6, 'Carrasco')

-- Carga de datos de Terminales
insert into Terminal (idTer, idDept) values (1, 2)
insert into Terminal (idTer, idDept) values (2, 3)
insert into Terminal (idTer, idDept) values (3, 4)
insert into Terminal (idTer, idDept) values (4, 5)
insert into Terminal (idTer, idDept) values (5, 5)
insert into Terminal (idTer, idDept) values (6, 1)

Completion time: 2024-11-17T23:29:17.1744465-08:00
```

1 row(s) affected

Query executed successfully.

Ln 67 Col 1 Ch 1 ING

MATIAS\SQLPRESTEST (16...) MATIAS\matia70 (1) obligatorio 00:00:00 0 rows

SQLQueryLog - MATIAS\SQLEXPRESSTEST (Mati... - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

New Query Object Explorer Task List Object Explorer

Quick Launch (Ctrl+Q)

obligatorio

SQLQueryLog - M... - MATIAS\matia (70) = < X

```
Insert into Departamento (IdDepto, nombreDepto) values (1, 'Salta')
Insert into Departamento (IdDepto, nombreDepto) values (2, 'Montevideo')
Insert into Departamento (IdDepto, nombreDepto) values (3, 'Bogota')
Insert into Departamento (IdDepto, nombreDepto) values (4, 'Lima')
Insert into Departamento (IdDepto, nombreDepto) values (5, 'Fray Bentos')
Insert into Departamento (IdDepto, nombreDepto) values (6, 'Canelones')

-- Carga de datos de Terminales
Insert into Terminal (IdTer, IdDepto) values (1, 2)
Insert into Terminal (IdTer, IdDepto) values (2, 3)
Insert into Terminal (IdTer, IdDepto) values (3, 4)
Insert into Terminal (IdTer, IdDepto) values (4, 6)
Insert into Terminal (IdTer, IdDepto) values (5, 5)
Insert into Terminal (IdTer, IdDepto) values (6, 1)

-- Carga de datos de Orientacion Terminales
Insert into OrientacionTerminales (IdTerOrigen, IdTerDestino) values (1, 6)
Insert into OrientacionTerminales (IdTerOrigen, IdTerDestino) values (2, 5)
Insert into OrientacionTerminales (IdTerOrigen, IdTerDestino) values (3, 4)
Insert into OrientacionTerminales (IdTerOrigen, IdTerDestino) values (4, 3)
Insert into OrientacionTerminales (IdTerOrigen, IdTerDestino) values (5, 2)

(5 row(s) affected)
Messages

(1 row affected)

Completion time: 2024-11-17T23:29:35.2213146-03:00
```

88 %

Query executed successfully.

Ready

Le78 Col1 Ch1 ING

MATIAS\SQLEXPRESSTEST (Mati... - MATIAS\matia (70) obligatorio 00:00:00 0 rows

SQLQuery1.MATIAS\SQLPRESTEST.Obligatorios (MATIAS\matia (70)) - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Window Help

Quick Launch (Ctrl+Q) ↗

Object Explorer

Connections

MATIAS\SQLPRESTEST (SQL Server)

- Databases
 - System Databases
 - Database Snapshots
 - Replication
- ALGEBRA_SQL
- EjClase
- obligatorio
- Database Diagrams
- Tables
 - System Tables
 - VTables
 - Common Tables
 - Graph Tables
 - Acomaplante
 - Asiento
 - Cuentas
 - Departamento
 - DestinoTuristico
 - Destino
 - Destinatario
 - OrientacionTerminales
 - Pasaje
 - Telefono
 - Terminal
 - TerminalAC
 - Columns
 - Keys
 - Check Constraints
 - Triggers
 - Indexes
 - Statistics
 - Jobs
 - Replicated
- Views
- External Resources
- Synonyms
- Programmability
- Query Store
- Service Broker
- Storage
- Security
- practica
- Security
- Server Objects
- Replication

SQLQuery1.MATIAS\matia (70) - < x

```
insert into Departamento (iddepto, nombreDept) values (1, 'Salta')
insert into Departamento (iddepto, nombreDept) values (2, 'Montevideo')
insert into Departamento (iddepto, nombreDept) values (3, 'Paysandú')
insert into Departamento (iddepto, nombreDept) values (4, 'Artigas')
insert into Departamento (iddepto, nombreDept) values (5, 'Fray Bentos')
insert into Departamento (iddepto, nombreDept) values (6, 'Canelones')

-- Carga de datos de Terminales
insert into Terminal (idTer, idDept, nombreTer) values (1, 2, 'Free Cruces')
insert into Terminal (idTer, idDept, nombreTer) values (2, 3, 'Terminal Paysandú')
insert into Terminal (idTer, idDept, nombreTer) values (3, 4, 'Terminal Artigas')
insert into Terminal (idTer, idDept, nombreTer) values (4, 5, 'Terminal Fray Bentos')
insert into Terminal (idTer, idDept, nombreTer) values (5, 5, 'Terminal Fray Bentos')
insert into Terminal (idTer, idDept, nombreTer) values (6, 1, 'Terminal Salto')

-- Carga de datos de Orientación Terminales
insert into OrientacionTerminales (idTerOrigen, idTerDestino) values (1, 6)
insert into OrientacionTerminales (idTerOrigen, idTerDestino) values (2, 5)
insert into OrientacionTerminales (idTerOrigen, idTerDestino) values (3, 4)
insert into OrientacionTerminales (idTerOrigen, idTerDestino) values (4, 3)
insert into OrientacionTerminales (idTerOrigen, idTerDestino) values (5, 2)
```

10 % ▶ 4

Messages

```
(1 row affected)
```

Completion time: 2024-11-17T23:36:31.1429988-03:00

100 % ▶ 4

Query executed successfully.

SQLQuery1.sql - MATIAS\SQLEXPRESSTEST\obligatorio (MATIAS\matia (70)) - Microsoft SQL Server Management Studio

```

-- Carga de datos de Terminales
Insert into Terminal (idTer, idDept, nombreTer) values (1, 2, 'Tres Cruces')
Insert into Terminal (idTer, idDept, nombreTer) values (2, 3, 'Terminal Paysandú')
Insert into Terminal (idTer, idDept, nombreTer) values (3, 4, 'Terminal Artigas')
Insert into Terminal (idTer, idDept, nombreTer) values (4, 6, 'Terminal Canciones')
Insert into Terminal (idTer, idDept, nombreTer) values (5, 5, 'Terminal Fray Bentos')
Insert into Terminal (idTer, idDept, nombreTer) values (6, 1, 'Terminal Salto')

-- Carga de datos de OrientacionTerminales
Insert into OrientacionTerminales (idTerOrigen, idTerDestino) values (1, 4)
Insert into OrientacionTerminales (idTerOrigen, idTerDestino) values (2, 5)
Insert into OrientacionTerminales (idTerOrigen, idTerDestino) values (3, 4)
Insert into OrientacionTerminales (idTerOrigen, idTerDestino) values (4, 3)
Insert into OrientacionTerminales (idTerOrigen, idTerDestino) values (5, 2)
Insert into OrientacionTerminales (idTerOrigen, idTerDestino) values (6, 1)

-- Carga de datos de Destino Turistico
Insert into DestinoTuristico (idDT, idTerDestino, infodr1DT, DuracionAproxDT, FechaDT, HoraDT, ImporteViajeDT) values (1,1,'Maximo_Confort', 2:30,'20-06-2024', 19:30, 1300 )

```

100 % - 4 rows

Messages

```

(1 row affected)
Completion time: 2024-11-17T23:36:49.6649298-03:00

```

Query executed successfully.

SQLQuery1.sql - MATIAS\SQLEXPRESSTEST\obligatorio (MATIAS\matia (70)) - Microsoft SQL Server Management Studio

```

-- Carga de datos de Terminales
Insert into Terminal (idTer, idDept, nombreTer) values (1, 2, 'Tres Cruces')
Insert into Terminal (idTer, idDept, nombreTer) values (2, 3, 'Terminal Paysandú')
Insert into Terminal (idTer, idDept, nombreTer) values (3, 4, 'Terminal Artigas')
Insert into Terminal (idTer, idDept, nombreTer) values (4, 6, 'Terminal Canciones')
Insert into Terminal (idTer, idDept, nombreTer) values (5, 5, 'Terminal Fray Bentos')
Insert into Terminal (idTer, idDept, nombreTer) values (6, 1, 'Terminal Salto')

-- Carga de datos de OrientacionTerminales
Insert into OrientacionTerminales (idTerOrigen, idTerDestino) values (1, 4)
Insert into OrientacionTerminales (idTerOrigen, idTerDestino) values (2, 5)
Insert into OrientacionTerminales (idTerOrigen, idTerDestino) values (3, 4)
Insert into OrientacionTerminales (idTerOrigen, idTerDestino) values (4, 3)
Insert into OrientacionTerminales (idTerOrigen, idTerDestino) values (5, 2)
Insert into OrientacionTerminales (idTerOrigen, idTerDestino) values (6, 1)

-- Carga de datos de Destino Turistico
Insert into DestinoTuristico (idDT, idTerDestino, infodr1DT, DuracionAproxDT, FechaDT, HoraDT, ImporteViajeDT) values (1,1,'Montevideo - Salto', 220,'2025-12-10', '00:30:00', 1200 )
Insert into DestinoTuristico (idDT, idTerDestino, infodr1DT, DuracionAproxDT, FechaDT, HoraDT, ImporteViajeDT) values (2,6,'Salto - Montevideo', 240,'2024-09-22', '11:30:00', 1100 )
Insert into DestinoTuristico (idDT, idTerDestino, infodr1DT, DuracionAproxDT, FechaDT, HoraDT, ImporteViajeDT) values (3,3,'Artigas - Canciones', 360,'2025-01-01', '18:50:00', 1050 )
Insert into DestinoTuristico (idDT, idTerDestino, infodr1DT, DuracionAproxDT, FechaDT, HoraDT, ImporteViajeDT) values (4,5,'Fray Bentos - Paysandú', 115,'2024-11-14', '22:30:00', 610 )

```

100 % - 4 rows

Messages

```

(1 row affected)
Completion time: 2024-11-17T23:49:07.4055809-03:00

```

Query executed successfully.

SQLQueryLog - MATIAS\SQLEXPRESSTEST\obligatorio (MATIAS\matia (70)) - Microsoft SQL Server Management Studio

```

-- Carga de datos de OrientacionTerminales
Insert into DestinoTuristico (IdOT, IdTerDestino, IdTerOrigen, DuracionaproxDT, FechaDT, HorasDT, ImporteVlajeDT) values (1,6, 'Montevideo - Salto', 220 , '2025-10-10', '00:30:00', 1200 )
Insert into DestinoTuristico (IdOT, IdTerDestino, IdTerOrigen, DuracionaproxDT, FechaDT, HorasDT, ImporteVlajeDT) values (2,5, 'Salto - Montevideo', 240 , '2024-05-22', '11:30:00', 1100 )
Insert into DestinoTuristico (IdOT, IdTerDestino, IdTerOrigen, DuracionaproxDT, FechaDT, HorasDT, ImporteVlajeDT) values (3,3, 'Artigas - Canales', 360 , '2025-01-01', '10:50:00', 1000 )
Insert into DestinoTuristico (IdOT, IdTerDestino, IdTerOrigen, DuracionaproxDT, FechaDT, HorasDT, ImporteVlajeDT) values (4,5, 'Playa Bentos - Paysandú', 115 , '2024-11-12', '21:30:00', 610 )

-- Carga de datos de Bus
Insert into Bus (IdBus, IdOT, MarcaBus, TiposBus, capalientosBus) values (1,1,'Mercedes', 'Doble piso', 40)
Insert into Bus (IdBus, IdOT, MarcaBus, TiposBus, capalientosBus) values (2,2,'Volvo', 'Microbus', 20)
Insert into Bus (IdBus, IdOT, MarcaBus, TiposBus, capalientosBus) values (3,3,'Mercedes', 'Microbus', 15)
Insert into Bus (IdBus, IdOT, MarcaBus, TiposBus, capalientosBus) values (4,4,'Meracpolin', 'Comfort', 30)

--Carga de datos de Asiento
Insert into Asiento (NroFilas, LetrasA, IdBus) values ('5', 'A', 1)
Insert into Asiento (NroFilas, LetrasA, IdBus) values ('2', 'A', 1)
Insert into Asiento (NroFilas, LetrasA, IdBus) values ('3', 'B', 1)
Insert into Asiento (NroFilas, LetrasA, IdBus) values ('4', 'C', 1)

(1 row affected)
(1 row affected)
(1 row affected)
(1 row affected)

Completion time: 2024-11-17T23:51:12.5191842-03:00

```

100 %

Query executed successfully.

SQLQueryLog - M..MATIAS\matia (70) - Microsoft SQL Server Management Studio

```

-- Carga de datos de Bus
Insert into Bus (IdBus, IdOT, MarcaBus, TiposBus, capalientosBus) values (1,1,'Mercedes', 'Doble piso', 40)
Insert into Bus (IdBus, IdOT, MarcaBus, TiposBus, capalientosBus) values (2,2,'Volvo', 'Microbus', 20)
Insert into Bus (IdBus, IdOT, MarcaBus, TiposBus, capalientosBus) values (3,3,'Mercedes', 'Microbus', 15)
Insert into Bus (IdBus, IdOT, MarcaBus, TiposBus, capalientosBus) values (4,4,'Meracpolin', 'Comfort', 30)

--Carga de datos de Asiento
Insert into Asiento (NroFilas, LetrasA, IdBus) values ('5', 'A', 1)
Insert into Asiento (NroFilas, LetrasA, IdBus) values ('2', 'A', 1)
Insert into Asiento (NroFilas, LetrasA, IdBus) values ('3', 'B', 1)
Insert into Asiento (NroFilas, LetrasA, IdBus) values ('5', 'C', 2)
Insert into Asiento (NroFilas, LetrasA, IdBus) values ('4', 'C', 2)
Insert into Asiento (NroFilas, LetrasA, IdBus) values ('4', 'D', 2)
Insert into Asiento (NroFilas, LetrasA, IdBus) values ('5', 'D', 1)
Insert into Asiento (NroFilas, LetrasA, IdBus) values ('5', 'E', 3)
Insert into Asiento (NroFilas, LetrasA, IdBus) values ('6', 'E', 3)
Insert into Asiento (NroFilas, LetrasA, IdBus) values ('6', 'F', 4)

--Carga de datos de Paisaje
Insert into Paisaje (IdP, IdT, NroFilasA,LetrasA, IdOT, estadoP, fechaP) values (1,1, '2', 'A', 1, 'Activo', '20-06-2024')

(1 row affected)

Completion time: 2024-11-17T23:52:13.0459477-03:00

```

100 %

Query executed successfully.

4.3 Consultas SQL

5.3.1. Consulta 1

1. Contamos cuántos pasajes ha comprado cada turista usando JOIN y COUNT().
2. Agrupamos por turista (GROUP BY) para obtener el total de pasajes por cada uno.
3. Buscamos el máximo número de pasajes comprados usando una subconsulta.
4. Filtramos los resultados con HAVING para mostrar solo aquellos turistas que tienen la cantidad máxima de pasajes.

5.3.2. Consulta 2

The screenshot shows the Microsoft SQL Server Management Studio interface. On the left, the Object Explorer tree shows the database structure for 'MATIAS\SQLEXPRESSTEST' (SQL Server 16.0.1). In the center, a query window displays the following T-SQL code:

```

-- Consulta 2
-- Filtro Bus #
-- Consulta 1
SELECT MAX(cantidad_pasajes)
FROM (
    SELECT COUNT(*) AS cantidad_pasajes
    FROM Pasaje
    GROUP BY Idt
) AS subconsulta
;

-- Consulta 2
SELECT Bus #
FROM Bus
WHERE
    B.capAsientosBus > 35
    AND B.IdT IN (
        SELECT D.IdT
        FROM DestinoTuristico D
        WHERE D.FechaI = '2023-01-01', D.FechaF = '2023-01-31' AS DATE
    )
;

```

The results grid below the code shows one row of data:

idBus	IdT	MarcasBus	TipoBus	capAsientosBus
1	1	Mercedes	Doble piso	40

At the bottom of the screen, a status bar indicates: 'Query executed successfully.' and 'MATIAS\SQLEXPRESSTEST (16.0... MATIAS\matia (53) obligatorio 00:00:00 1 rows'.

1. Filtro por cantidad de asientos:

- B.capAsientosBus > 35: Selecciona solo los buses que tienen más de 35 asientos.
2. Excluir buses con destino para mañana:

- **B.idDT NOT IN (...):** Excluye aquellos buses que tienen un destino turístico programado para el día de mañana.
- La subconsulta busca todos los destinos (DestinoTuristico) cuya fecha (FechaDT) sea igual a la fecha de mañana. Para obtener la fecha de mañana:
 - En MySQL: CURDATE() + INTERVAL 1 DAY.
 - En SQL Server: GETDATE().

5.3.3. Consulta 3

The screenshot shows the Microsoft SQL Server Management Studio interface. The title bar reads "consultas.sql - MATIAS\SQLEXPRESSTEST\obligatorio (MATIAS\matia (54)) - Microsoft SQL Server Management Studio". The main window displays a T-SQL script with several numbered comments (1 through 5) explaining specific parts of the query. The script uses joins, group by, and having clauses to filter results based on travel dates and passenger counts. The results pane at the bottom shows one row of data from the query execution.

```

-- Consulta 3
SELECT *
FROM DestinoTuristico D
WHERE D.FechaDT = DATEADD(DAY, 1, CAST(GETDATE() AS DATE))
;

-- Consulta 4
SELECT distinct
    P.idP,
    P.nombreP,
    P.apellidoP,
    T.nombreT,
    T.apellidoT,
    T.apellido1T,
    T.apellido2T,
    T.nroFilials,
    A.nroFilials,
    A.Letraiks
FROM
    Pasaje P
JOIN
    Turista T ON P.idT = T.idT
JOIN
    Asiento A ON P.NroFilials = A.NroFilials AND P.Letraiks = A.Letraiks
WHERE
    P.idT = 255;
;

-- HAVING COUNT(P.idP) > 5;
;

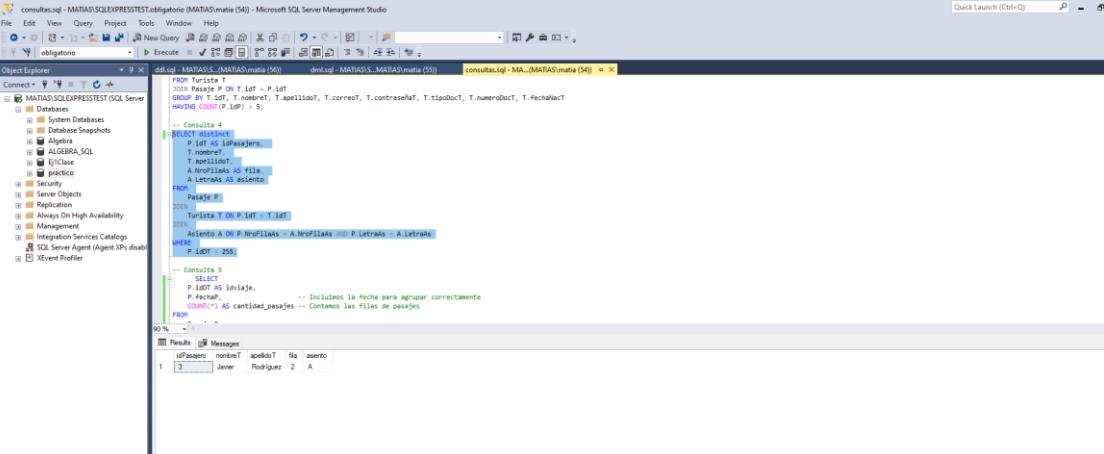
-- Consulta 5
SELECT *
FROM DestinoTuristico D
WHERE D.FechaDT = DATEADD(DAY, 1, CAST(GETDATE() AS DATE))
GROUP BY D.idDT, D.destinoT
HAVING COUNT(P.idP) > 5;
;

```

ID	nombreT	apellidoT	correoT	contraseñaT	tipoDocT	númeroDocT	fechaNacT
1	Matias	Otero	matias@gmail.com	12345678	CI	12345678	2000-12-11

1. **JOIN entre Turista y Pasaje:** Unimos las tablas para relacionar cada pasajero con sus pasajes.
2. **GROUP BY T.idT, T.nombreT, T.apellidoT, ...:** Incluimos todas las columnas de Turista en el GROUP BY que aparecen en la lista de selección (SELECT). Esto es necesario en SQL Server.
3. **HAVING COUNT(P.idP) > 5:** Filtramos los resultados para mostrar solo aquellos pasajeros que tienen más de 5 pasajes registrados.

5.3.4. Consulta 4



consultas.sql - MATIAS\SQLPRESTEST\matias (MATIAS\matia) - Microsoft SQL Server Management Studio

File Edit View Query Project Tools Windows Help

Execute

Object Explorer

Connect +

MATIAS\SQLPRESTEST (SQL Server)

- Databases
- System Databases
- Database Snapshots
- Algebra
- Adventure_
- Adventure_SQL
- ejercitacion
- practico
- Security
- Server Objects
- Application
- Always On High Availability
- Management
- Integration Services Catalogs
- SQL Server Agent (Agent XPs disabled)
- Event Profiler

File Explorer

Toolbox

Task List

Properties

File

Help

Quick Launch (Ctrl+Q)

consultas.sql - MATIAS\matia (54) ->

```
GO
-- Consulta 4
SELECT DISTINCT
    P.idT AS IdPasajero,
    P.nombreT,
    P.apellidoT,
    T.numerodoct,
    T.contrasenaT,
    T.tipodocT,
    T.numerodocT,
    T.fechaNac
FROM
    Pasaje P
    INNER JOIN Turista T ON P.idT = T.idT
    JOIN Alumno A ON P.NroPlacaA = A.NroPlacaA
    WHERE
        P.idT = 255
-- Consulta 5
SELECT
    D.alumnoT AS IdAlumno,
    D.fechaNac AS FechaNacimiento,
    D.fechaNac AS Fecha,
    D.fechaNac AS FechaNac
    -- Incluimos la fecha para agrupar correctamente
    COUNT(*) AS cantidad_pasajes -- Contamos las filas de pasajes
FROM
    Pasaje D
    GROUP BY
        D.alumnoT, D.fechaNac, D.fechaNac, D.fechaNac
    HAVING
        D.idT = 255
    ORDER BY
        D.fechaNac DESC
```

Results

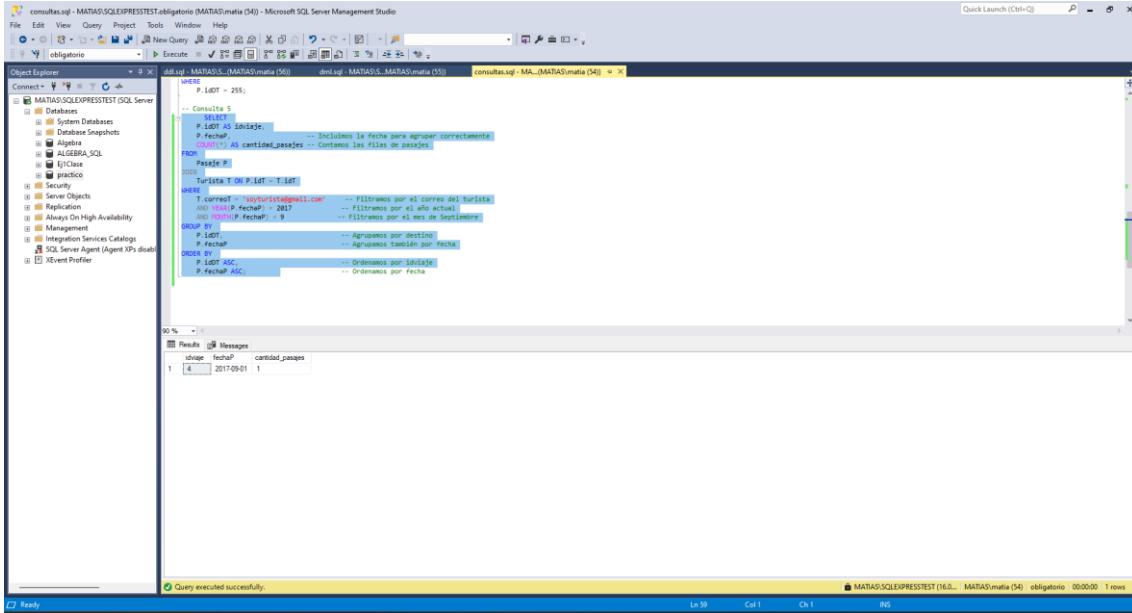
idPasajero	nombreT	apellidoT	nro asiento
3	Javier	Rodríguez	2

Messages

Query executed successfully.

1. **JOIN Pasaje P con Turista T:** Relacionamos los pasajes con los turistas que los compraron ($P.idT = T.idT$).
 2. **A.NroFilaAs AS fila, A.LetraAs AS asiento:** Utilizamos las columnas NroFilaAs y LetraAs para representar el asiento. Estos dos campos juntos identifican de manera única un asiento en el bus.
 3. **JOIN Pasaje P con asiento:** Relacionamos los pasajes con los asientos utilizando NroFilaAs y LetraAs, que juntos identifican un asiento único.
 4. **WHERE P.idDT = 255:** Filtramos para mostrar solo los pasajes comprados para el destino cuyo idDT es 255.

5.3.5. Consulta 5



```
consultas.sql - MATIAS\SQLEXPRESSTEST\obligatorio (MATIAS\matia (54)) - Microsoft SQL Server Management Studio
File Edit View Query Project Tools Window Help
Connect to... New Query DML Log - consultas.sql - MATIAS\matia (54) - x
[...]
Object Explorer
MATIAS\SQLEXPRESSTEST (SQL Server)
Databases
System Databases
Database Snapshots
ALGEBRA
ALGEBRA_SQL
EjClase
practico
Script
Server Objects
Replication
Always On High Availability
Integration Services Catalogs
SQL Server Agent (Agent XPs disabled)
XEvent Profiler
dml.log - MATIAS\matia (54) - x
consultas.sql - MATIAS\matia (54) - x
WHERE
P.IdOT = 215;
-- Consulta 5
SELECT
P.IdOT AS idViaje,
P.FechaP AS fechaPasaje -- Incluimos la fecha para agrupar correctamente
FROM [dbo].[pasaje] AS cantidad_pasajes -- Contenemos las filas de pasajes
JOIN [dbo].[Turista] T ON P.IdT = T.IdT
WHERE
T.Correo = 'soyturista@gmail.com' -- FILTRAMOS por el correo del turista
AND YEAR(P.fechaP) = 2017 -- FILTRAMOS por el año actual
AND MONTH(P.fechaP) = 9 -- FILTRAMOS por el mes de Septiembre
GROUP BY
P.IdOT,
P.FechaP -- Agrupamos por destino
-- Agrupamos también por fecha
ORDER BY
P.IdOT ASC,
P.FechaP ASC; -- Ordenamos por idViaje
-- Ordenamos por fecha
Results Messages
idViaje fechaPasaje cantidad_pasajes
1 4 2017-09-01 1
Query executed successfully.
Ln 59 Col 1 Ch 1 Info
Ready
```

- JOIN:** Unimos las tablas Pasaje y Turista usando el idT para encontrar los pasajes del turista con el correo **soyturista@gmail.com**.
- WHERE:** Filtramos los pasajes comprados en septiembre de 2017 usando las funciones **YEAR()** y **MONTH()** en la columna **fechaP**.
- COUNT():** Contamos la cantidad de pasajes por destino (**idDT**).
- GROUP BY:** Agrupamos por el destino para contar los pasajes por cada uno.
- ORDER BY:** Ordenamos los resultados por **idDT** (id del viaje) de forma ascendente.

4.4 Evidencia de ejecución de consultas

<<Por cada consulta, incluir evidencia de ejecución (capturas de pantalla) de las consultas entregadas (consultas.sql) sobre la base de datos generada con los datos de prueba cargados desde el archivo entregado (dml.sql)>>

