Sri Lanka Institute of Information Technology

Web-based Train reservation system

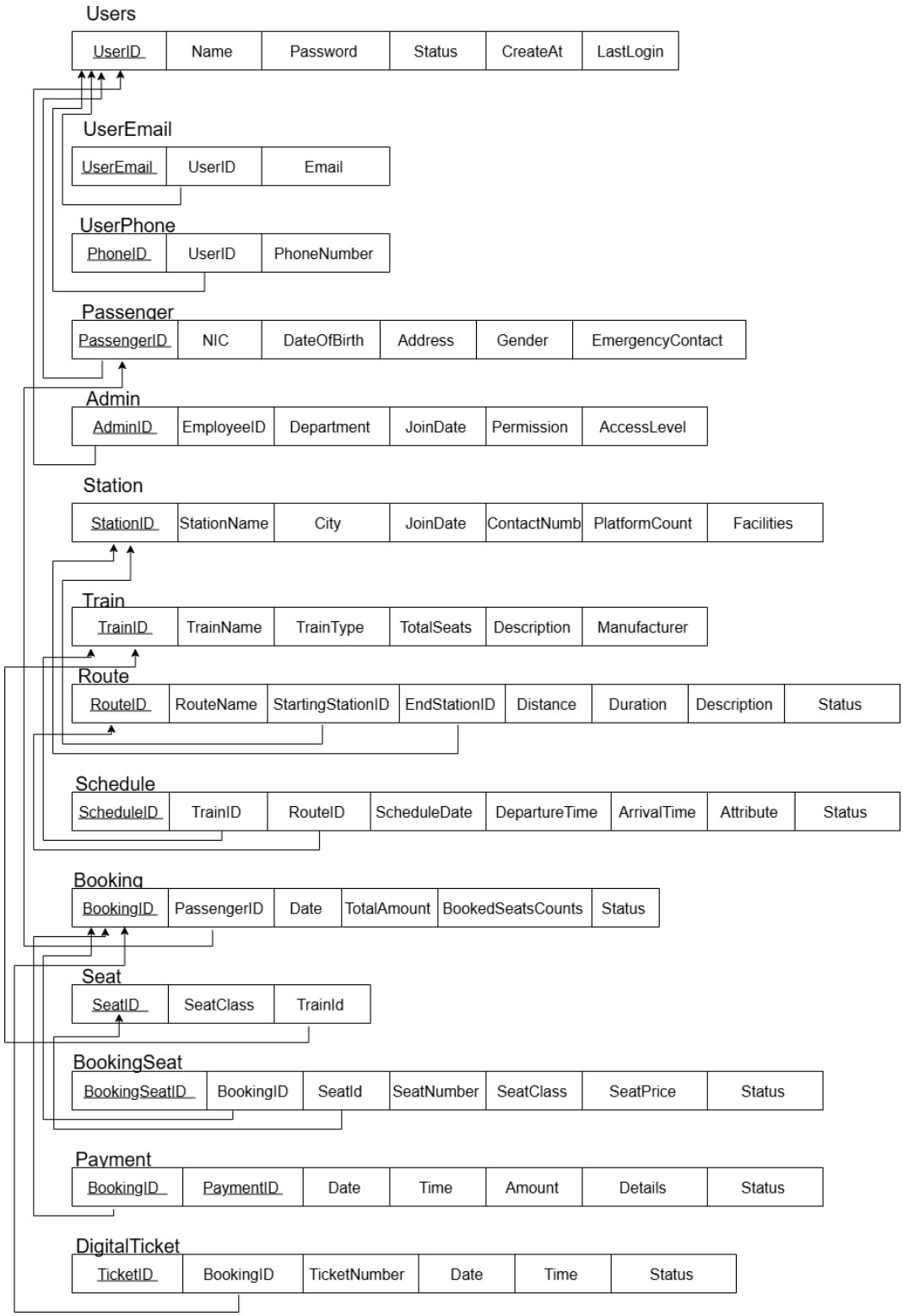
(Lanka Rail)

IT2140-Database Design and Development

Assignment – Part 02

Contents

Mapping EER to Relational Schema



Mapping choices for ISA

In this system, an ISA relationship exists between Users, Passenger, and Admin. The superclass Users contains all common attributes, while each subclass (Passenger, Admin) contains specific attributes. The subclass tables use the same primary key as the superclass, acting also as a foreign key, ensuring that every Passenger or Admin must first exist as a User. This approach maintains normalization, ensures data consistency, and clearly represents inheritance in the relational model.

SQL DDL Implementation

```
CREATE TABLE Users (
  UserID INT PRIMARY KEY,
  Name VARCHAR(100) NOT NULL,
  Password VARCHAR(100) NOT NULL,
  Status VARCHAR(20) DEFAULT 'Active',
  CreateAt DATE,
  LastLogin DATE
)

CREATE TABLE UserEmail (
  EmailID INT PRIMARY KEY,
  UserID INT NOT NULL,
  Email VARCHAR(100) NOT NULL,
  FOREIGN KEY (UserID) REFERENCES Users(UserID)
)

CREATE TABLE UserPhone (
  PhoneID INT PRIMARY KEY,
  UserID INT NOT NULL,
  PhoneNumber VARCHAR(15) NOT NULL,
  FOREIGN KEY (UserID) REFERENCES Users(UserID)
)

CREATE TABLE Passenger (
  PassengerID INT PRIMARY KEY,
  NIC VARCHAR(12) UNIQUE NOT NULL,
  DateOfBirth DATE,
  Address VARCHAR(255),
  Gender VARCHAR(10),
  EmergencyContact VARCHAR(15),
  FOREIGN KEY (PassengerID) REFERENCES Users(UserID)
)

CREATE TABLE Admin (
  AdminID INT PRIMARY KEY,
  EmployeeID VARCHAR(10) UNIQUE NOT NULL,
  Department VARCHAR(50),
  JoinDate DATE,
  Permission VARCHAR(50),
  AccessLevel VARCHAR(30),
  FOREIGN KEY (AdminID) REFERENCES Users(UserID)
)

CREATE TABLE Station (
  StationID INT PRIMARY KEY,
  StationName VARCHAR(100) NOT NULL,
  City VARCHAR(50),
```

```
  ContactNumber VARCHAR(15),
  PlatformCount INT,
  Facilities VARCHAR(255)
)

CREATE TABLE Train (
  TrainID INT PRIMARY KEY,
  TrainName VARCHAR(100) NOT NULL,
  TrainType VARCHAR(50),
  TotalSeats INT,
  Description VARCHAR(255),
  Manufacturer VARCHAR(100),
  Status VARCHAR(20) DEFAULT 'Active'
)

CREATE TABLE Route (
  RouteID INT PRIMARY KEY,
  RouteName VARCHAR(100) NOT NULL,
  StartingStationID INT NOT NULL,
  EndStationID INT NOT NULL,
  Distance DECIMAL(10,2),
  Duration VARCHAR(20),
  Description VARCHAR(255),
  Status VARCHAR(20),
  FOREIGN KEY (StartingStationID) REFERENCES Station(StationID),
  FOREIGN KEY (EndStationID) REFERENCES Station(StationID)
)

CREATE TABLE Schedule (
  ScheduleID INT PRIMARY KEY,
  TrainID INT NOT NULL,
  RouteID INT NOT NULL,
  ScheduleDate DATE,
  DepartureTime TIME,
  ArrivalTime TIME,
  Attribute VARCHAR(100),
  Status VARCHAR(20),
  FOREIGN KEY (TrainID) REFERENCES Train(TrainID),
  FOREIGN KEY (RouteID) REFERENCES Route(RouteID)
)

CREATE TABLE Booking (
  BookingID INT PRIMARY KEY,
  PassengerID INT NOT NULL,
  Date DATE,
  TotalAmount DECIMAL(10,2),
  BookedSeatsCounts INT,
  Status VARCHAR(20) DEFAULT 'Pending',
  FOREIGN KEY (PassengerID) REFERENCES Passenger(PassengerID)
)
```

```
CREATE TABLE Seat(
  SeatID INT PRIMARY KEY,
  SeatClass VARCHAR(20),
  trainid INT NOT NULL,
  FOREIGN KEY (trainid) REFERENCES Train(trainid)
)

CREATE TABLE BookingSeat (
  BookingSeatID INT PRIMARY KEY,
  BookingID INT NOT NULL,
  seatid INT NOT NULL,
  SeatNumber VARCHAR(10),
  SeatClass VARCHAR(20),
  SeatPrice DECIMAL(10,2),
  Status VARCHAR(20),
  FOREIGN KEY (BookingID) REFERENCES Booking(BookingID),
  FOREIGN KEY (seatid) REFERENCES Seat(seatid)
)

CREATE TABLE Payment (
  PaymentID INT NOT NULL,
  BookingID INT NOT NULL,
  Date DATE,
  Time TIME,
  Amount DECIMAL(10,2),
  Details VARCHAR(255),
  Status VARCHAR(20) DEFAULT 'Pending',
  PRIMARY KEY (BookingID, PaymentID),
  FOREIGN KEY (BookingID) REFERENCES Booking(BookingID)
)

CREATE TABLE DigitalTicket (
  TicketID INT PRIMARY KEY,
  BookingID INT NOT NULL,
  TicketNumber VARCHAR(20) UNIQUE NOT NULL,
  Date DATE,
  Time TIME,
  Status VARCHAR(20),
  FOREIGN KEY (BookingID) REFERENCES Booking(BookingID)
)
```

Insert Sample Data

Users Table

Results Messages

	UserID	Name	Password	Status	CreateAt	LastLogin
1	1	Alice Perera	pass123	Active	2025-01-01	2025-10-01
2	2	Bimal Silva	pass456	Active	2025-02-15	2025-10-05
3	3	Chathu Fernando	pass789	Active	2025-03-10	2025-10-07
4	4	Dilani Jayasinghe	pass321	Active	2025-04-12	2025-10-08
5	5	Eshan Wijesinghe	pass654	Active	2025-05-20	2025-10-09

UserEmail Table

Results Messages

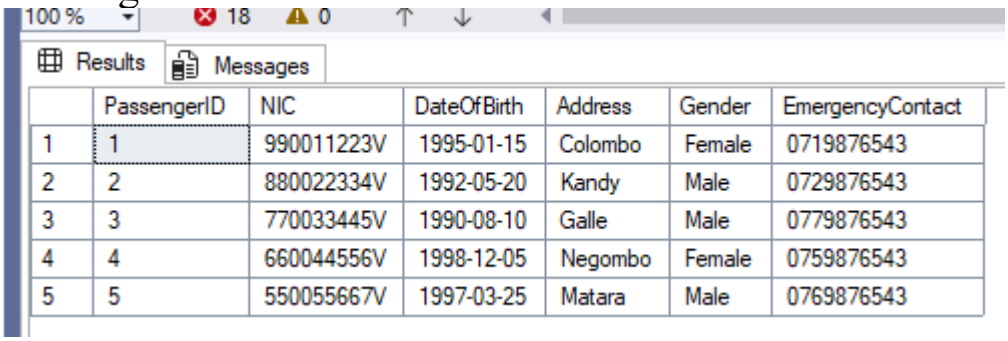
	EmailID	UserID	Email
1	1	1	alice@gmail.com
2	2	2	bimal@yahoo.com
3	3	3	chathu@hotmail.com
4	4	4	dilani@gmail.com
5	5	5	eshan@gmail.com

UserPhone Table

Results Messages

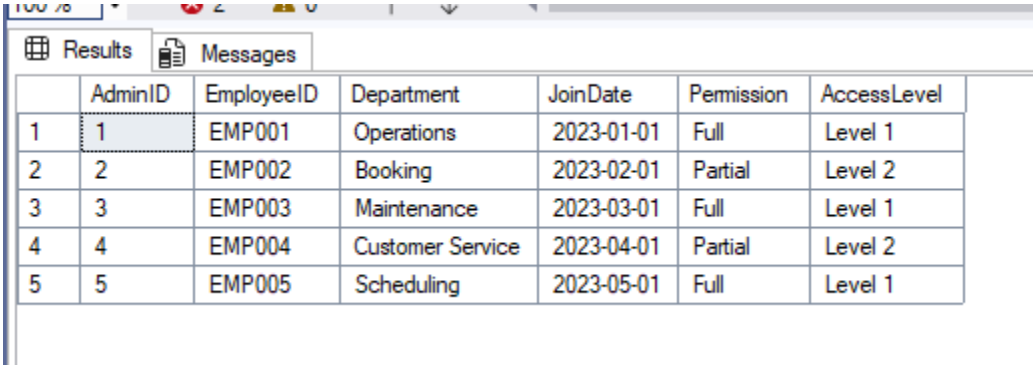
	PhoneID	UserID	PhoneNumber
1	1	1	0711234567
2	2	2	0722345678
3	3	3	0773456789
4	4	4	0754567890
5	5	5	0765678901

Passenger Table



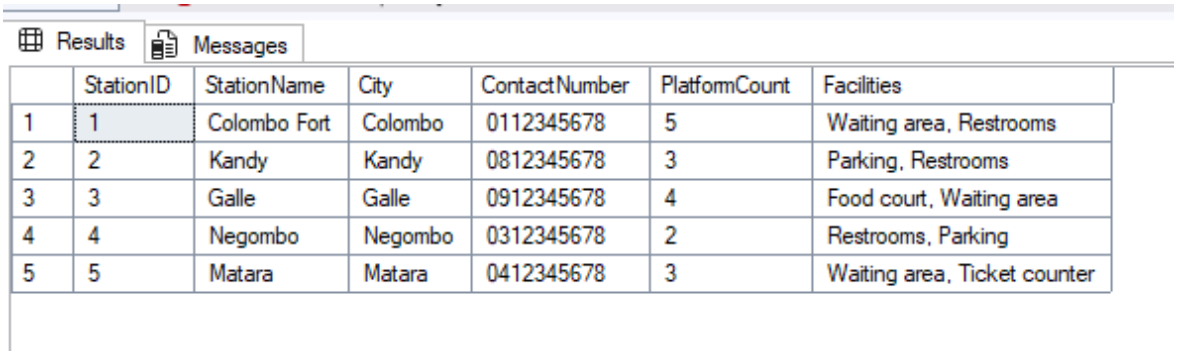
	PassengerID	NIC	DateOfBirth	Address	Gender	EmergencyContact
1	1	990011223V	1995-01-15	Colombo	Female	0719876543
2	2	880022334V	1992-05-20	Kandy	Male	0729876543
3	3	770033445V	1990-08-10	Galle	Male	0779876543
4	4	660044556V	1998-12-05	Negombo	Female	0759876543
5	5	550055667V	1997-03-25	Matara	Male	0769876543

Admin Table



	AdminID	EmployeeID	Department	JoinDate	Permission	AccessLevel
1	1	EMP001	Operations	2023-01-01	Full	Level 1
2	2	EMP002	Booking	2023-02-01	Partial	Level 2
3	3	EMP003	Maintenance	2023-03-01	Full	Level 1
4	4	EMP004	Customer Service	2023-04-01	Partial	Level 2
5	5	EMP005	Scheduling	2023-05-01	Full	Level 1

Station Table



	StationID	StationName	City	ContactNumber	PlatformCount	Facilities
1	1	Colombo Fort	Colombo	0112345678	5	Waiting area, Restrooms
2	2	Kandy	Kandy	0812345678	3	Parking, Restrooms
3	3	Galle	Galle	0912345678	4	Food court, Waiting area
4	4	Negombo	Negombo	0312345678	2	Restrooms, Parking
5	5	Matara	Matara	0412345678	3	Waiting area, Ticket counter

Train Table

Results Messages

	TrainID	TrainName	TrainType	TotalSeats	Description	Manufacturer	Status
1	1	Intercity Express	Express	200	Fast train Colombo-Kandy	Hitachi	Active
2	2	Coastal Line	Local	150	Scenic route Galle-Matara	Bombardier	Active
3	3	Night Rider	Night	100	Overnight service Colombo-Galle	Alstom	Active
4	4	City Connector	Express	180	Colombo to Negombo	Siemens	Active
5	5	Heritage Express	Tourist	120	Special sightseeing train	Hitachi	Active

Route Table

Results Messages

	RouteID	RouteName	StartingStationID	EndStationID	Distance	Duration	Description	Status
1	1	Colombo-Kandy Route	1	2	120.50	3h	Main intercity route	Active
2	2	Galle-Matara Route	3	5	65.00	1h30m	Scenic coastal route	Active
3	3	Colombo-Galle Route	1	3	120.00	2h30m	Express coastal train	Active
4	4	Colombo-Negombo Route	1	4	38.00	45m	Commuter route	Active
5	5	Heritage Tour	2	5	200.00	5h	Tourist scenic route	Active

Schedule Table

Results Messages

	ScheduleID	TrainID	RouteID	ScheduleDate	DepartureTime	ArrivalTime	Attribute	Status
1	1	1	1	2025-10-15	06:00:00.0000000	09:00:00.0000000	Express	Active
2	2	2	2	2025-10-16	08:30:00.0000000	10:00:00.0000000	Local	Active
3	3	3	3	2025-10-15	22:00:00.0000000	00:30:00.0000000	Night	Active
4	4	4	4	2025-10-17	07:00:00.0000000	07:45:00.0000000	Commuter	Active
5	5	5	5	2025-10-18	09:00:00.0000000	14:00:00.0000000	Tourist	Active

Seat Table

Results | Messages

	SeatID	SeatClass	trainid
1	1	First Class	1
2	2	Second Class	1
3	3	Third Class	2
4	4	First Class	3
5	5	Second Class	4
6	6	Third Class	5
7	7	First Class	5
8	8	Second Class	3
9	9	Third Class	2
10	10	Second Class	4

Booking Table

100 % | 19 | 0

Results | Messages

	BookingID	PassengerID	Date	TotalAmount	BookedSeatsCounts	Status
1	1	1	2025-10-05	5000.00	2	Confirmed
2	2	2	2025-10-06	3000.00	1	Pending
3	3	3	2025-10-07	4500.00	2	Confirmed
4	4	4	2025-10-08	2500.00	1	Pending
5	5	5	2025-10-09	6000.00	3	Confirmed

BookingSeat Table

Results | Messages

	BookingSeatID	BookingID	seatid	SeatNumber	SeatClass	SeatPrice	Status
1	1	1	1	A1	First Class	2500.00	Booked
2	2	1	2	B1	Second Class	2500.00	Booked
3	3	2	3	C1	Third Class	3000.00	Pending
4	4	3	4	A2	First Class	2250.00	Booked
5	5	3	8	B2	Second Class	2250.00	Booked

Payment Table

Results Messages

	PaymentID	BookingID	Date	Time	Amount	Details	Status
1	1	1	2025-10-05	10:00:00.0000000	5000.00	Credit Card	Paid
2	2	2	2025-10-06	11:30:00.0000000	3000.00	Cash	Pending
3	3	3	2025-10-07	09:45:00.0000000	4500.00	Bank Transfer	Paid
4	4	4	2025-10-08	14:00:00.0000000	2500.00	Cash	Pending
5	5	5	2025-10-09	15:30:00.0000000	6000.00	Credit Card	Paid

DigitalTicket Table

Results Messages

	TicketID	BookingID	TicketNumber	Date	Time	Status
1	1	1	TCK1001	2025-10-05	10:05:00.0000000	Active
2	2	2	TCK1002	2025-10-06	11:35:00.0000000	Pending
3	3	3	TCK1003	2025-10-07	09:50:00.0000000	Active
4	4	4	TCK1004	2025-10-08	14:05:00.0000000	Pending
5	5	5	TCK1005	2025-10-09	15:35:00.0000000	Active

SQL Queries s Outputs

1. Simple SELECT

```
256
257  SELECT TrainID, TrainName, TrainType, TotalSeats, Status
258  FROM Train
259  WHERE Status = 'Active'
260
```

100 % 18 0

Results Messages

	TrainID	TrainName	TrainType	TotalSeats	Status
1	1	Intercity Express	Express	200	Active
2	2	Coastal Line	Local	150	Active
3	3	Night Rider	Night	100	Active
4	4	City Connector	Express	180	Active
5	5	Heritage Express	Tourist	120	Active

This query displays all trains that are currently active in the system. It retrieves basic train details like ID, name, type, total seats, and status.

2. JOIN

```
260
261  SELECT B.BookingID,U.Name,B.TotalAmount,B.Status
262  FROM Booking B
263  JOIN Passenger P ON B.PassengerID = P.PassengerID
264  JOIN Users U ON P.PassengerID = U.UserID
265
```

100 % 18 0

Results Messages

	BookingID	Name	TotalAmount	Status
1	1	Alice Perera	5000.00	Confirmed
2	2	Bimal Silva	3000.00	Pending
3	3	Chathu Fernando	4500.00	Confirmed
4	4	Dilani Jayasinghe	2500.00	Pending
5	5	Eshan Wijesinghe	6000.00	Confirmed

This query joins Booking, Passenger, and Users tables to show booking details together with the passenger's name. It helps admin/staff quickly see who made each booking and its total cost

3. Aggregation

```
266  SELECT T.TrainName, SUM(B.BookedSeatsCounts) AS TotalSeatsBooked
267  FROM Booking B
268  JOIN BookingSeat BS ON B.BookingID = BS.BookingID
269  JOIN Seat S ON BS.SeatID = S.SeatID
270  JOIN Train T ON S.TrainID = T.TrainID
271  GROUP BY T.TrainName
272
```

100 % 18 0

Results Messages

	TrainName	TotalSeatsBooked
1	Coastal Line	1
2	Intercity Express	4
3	Night Rider	4

This query calculates the total number of seats booked per train. It uses the SUM() function to aggregate data from multiple tables.

4. GROUP BY / HAVING

```
273  SELECT T.TrainName, COUNT(B.BookingID) AS BookingCount
274  FROM Booking B
275  JOIN BookingSeat BS ON B.BookingID = BS.BookingID
276  JOIN Seat S ON BS.SeatID = S.SeatID
277  JOIN Train T ON S.TrainID = T.TrainID
278  GROUP BY T.TrainName
279  HAVING COUNT(B.BookingID) > 1
280
```

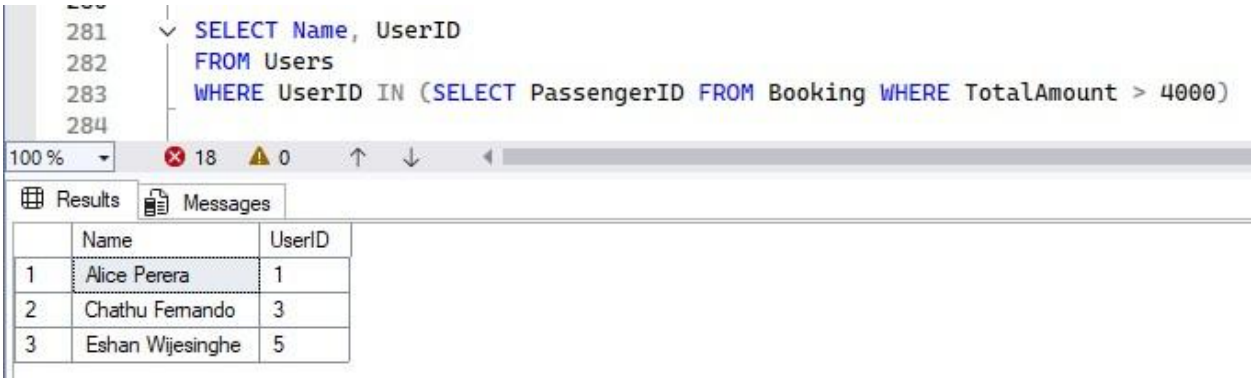
100 % 19 0

Results Messages

	TrainName	BookingCount
1	Intercity Express	2
2	Night Rider	2

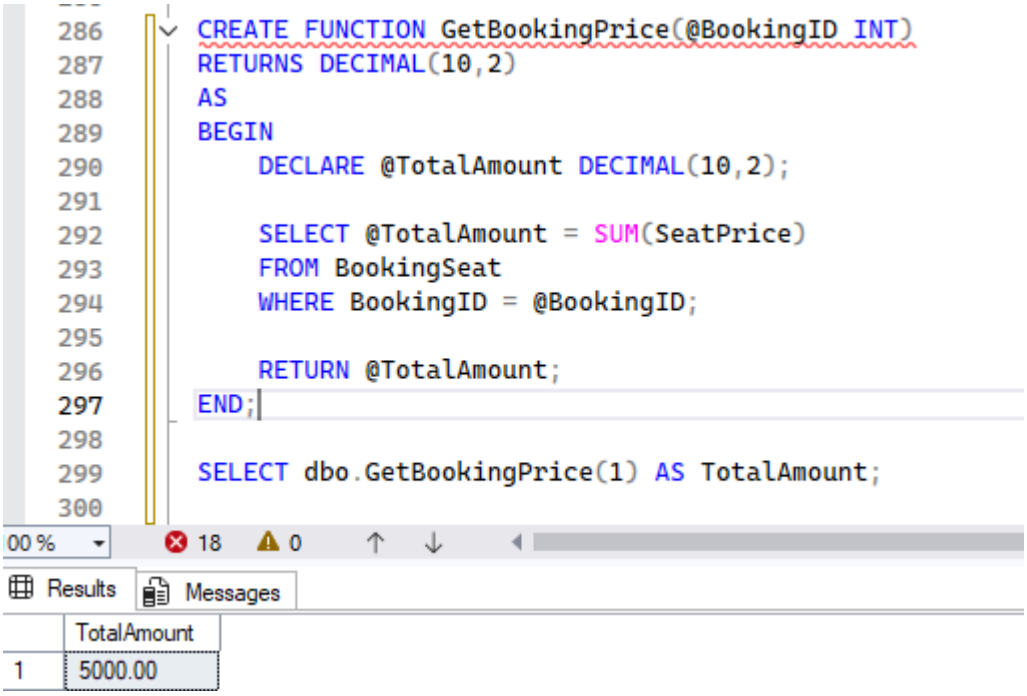
This query lists only those trains that have more than one booking. The HAVING clause filters grouped data after aggregation.

5. Subquery



This subquery finds all users who made bookings costing more than 4000. The inner query returns passenger IDs, which the outer query uses to fetch their names.

Stored Function/Procedure



```
286  CREATE FUNCTION GetBookingPrice(@BookingID INT)
287  RETURNS DECIMAL(10,2)
288  AS
289  BEGIN
290      DECLARE @TotalAmount DECIMAL(10,2);
291
292      SELECT @TotalAmount = SUM(SeatPrice)
293      FROM BookingSeat
294      WHERE BookingID = @BookingID;
295
296      RETURN @TotalAmount;
297  END;
298
299  SELECT dbo.GetBookingPrice(1) AS TotalAmount;
300
```

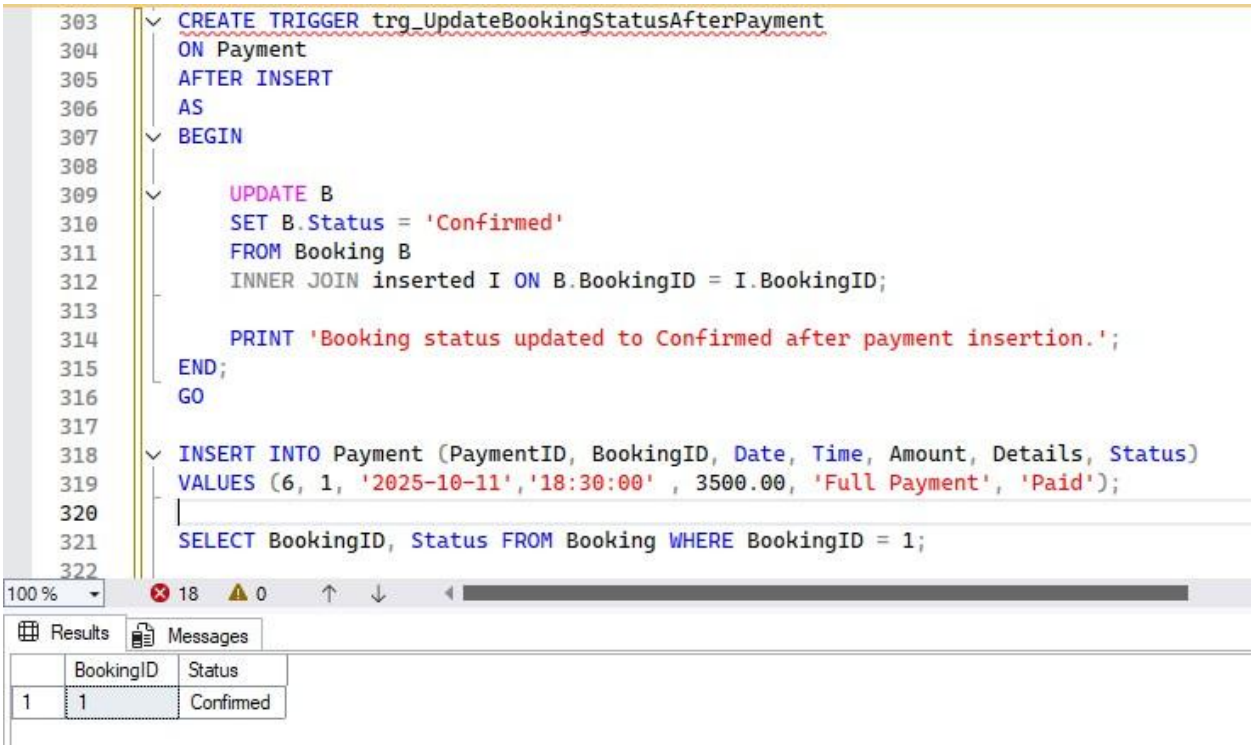
00 % 18 0

Results Messages

	TotalAmount
1	5000.00

This function calculates the total seat price for a specific booking by summing all seat prices in the BookingSeat table. @BookingID the ID of the booking to calculate total price for. IT Return The total amount as a decimal (10,2) value.

Trigger



```
303  CREATE TRIGGER trg_UpdateBookingStatusAfterPayment
304  ON Payment
305  AFTER INSERT
306  AS
307  BEGIN
308
309      UPDATE B
310      SET B.Status = 'Confirmed'
311      FROM Booking B
312      INNER JOIN inserted I ON B.BookingID = I.BookingID;
313
314      PRINT 'Booking status updated to Confirmed after payment insertion.';
315  END;
316  GO
317
318  INSERT INTO Payment (PaymentID, BookingID, Date, Time, Amount, Details, Status)
319  VALUES (6, 1, '2025-10-11','18:30:00' , 3500.00, 'Full Payment', 'Paid');
320
321  SELECT BookingID, Status FROM Booking WHERE BookingID = 1;
322
```

100 % 18 0

Results Messages

	BookingID	Status
1	1	Confirmed

The trigger runs after a new payment is added. It finds the related booking using BookingID. Updates that booking's status to 'Confirmed'.