DATABASE MANAGEMENT SYSTEMS

(UCS310)

2CO4

Transport Enquiry System

Submitted to: Mr. Anil Vashisht

Submitted by:

Bhumika Singh (102103114)

Serra Verma (102103731)

Jiya (102283024)

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

THAPAR INSTITUTE OF ENGINEERING AND TECHNOLOGY, (A DEEMED TO BE UNIVERSITY), PATIALA, PUNJAB

INDIA

Jan-May 2023

TABLE OF CONTENT

Introduction

The Bus Transport Enquiry System is a database management system designed to manage various aspects of a bus transport company's operations.

The Bus Transport Enquiry System is a software-based solution designed to assist commuters in accessing information related to bus schedules, routes, fares, and other important details. The system serves as a one-stop-shop for individuals who rely on public transportation, providing them with a user-friendly system that they can use to plan their daily commute and ensure timely arrival at their destination.

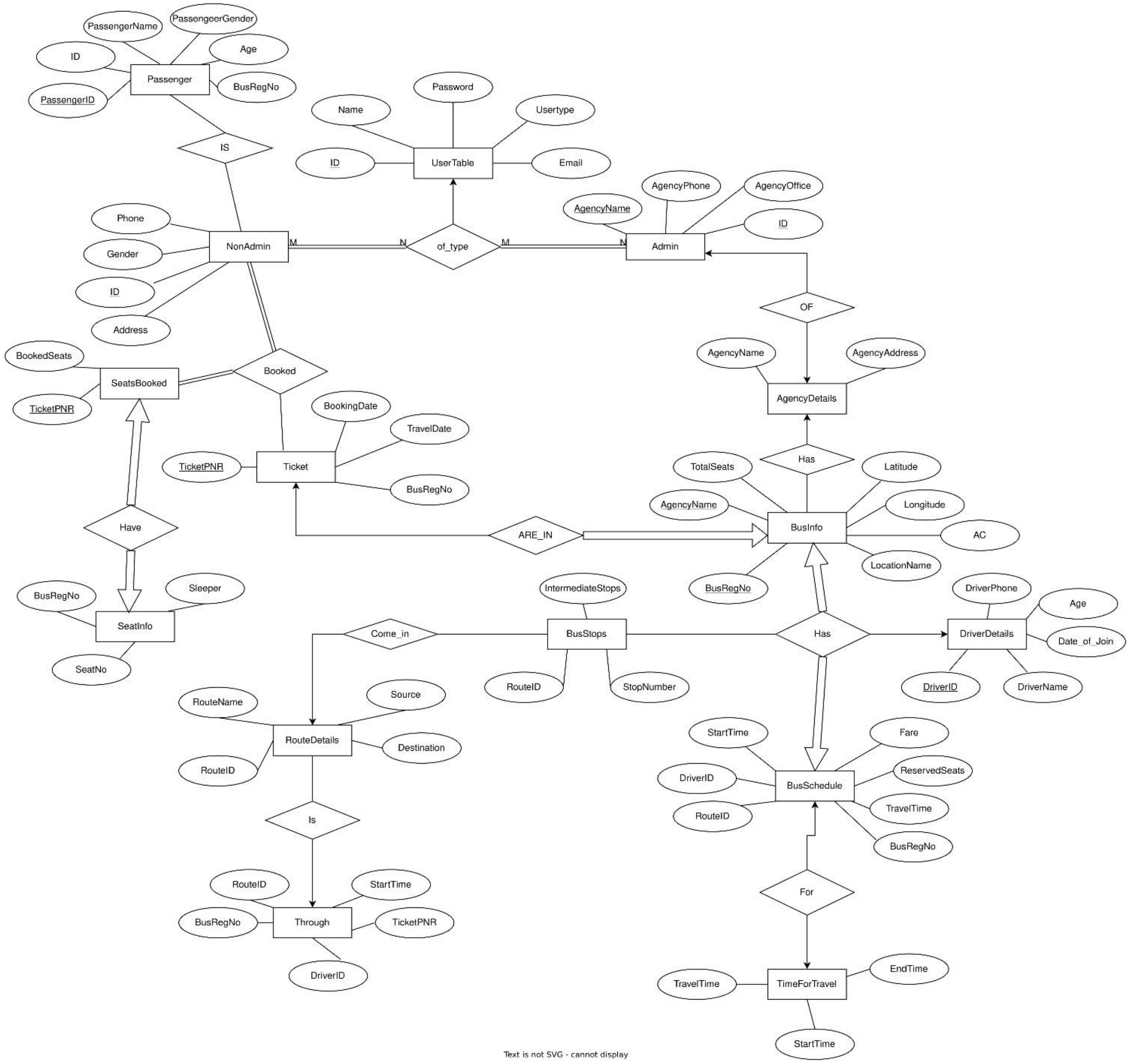
With the Bus Transport Enquiry System, users can access information at any time on the location of their bus and any potential delays or changes to the schedule. The system is designed to be highly reliable and accurate, using SQL and data analytics to provide the most up-to-date information possible. The system provides users with information about fares and ticket prices, enabling them to make informed decisions about their travel expenses. The system enables bus operators to improve their service delivery by providing users with accurate and up-to-date information about bus schedules, routes, and fares.

The Bus Transport Enquiry System is a valuable tool for both passengers and bus operators. The system provides users with real-time information about bus schedules, routes, fares, and other related information, enabling them to plan their journeys more efficiently. The system also enables bus operators to improve their service delivery by providing users with accurate and up-to-date information about bus transport services. Overall, the Bus Transport Enquiry System is a great addition to the transportation industry, providing users with a convenient and efficient way to access information about bus transport services.

ER Diagram

The above ER Diagram is not normalized.

It is first normalized and then the diagram shown is the direct conversion to Tables.



Normalization

Third Normal Form (3NF): Ensure that each non-primary key attribute depends only on the primary key and not on any other non-primary key attribute.

Table: UserTable

Columns:

- ID (PK)
- Email (UNIQUE, NOT NULL)
- Name
- Password (NOT NULL)
- Usertype (NOT NULL)

Table: NonAdmin

Columns:

- ID (PK, FK to UserTable.ID)
- Gender
- Phone (UNIQUE, NOT NULL)
- Address

Table: Admin

Columns:

- ID (PK, FK to UserTable.ID)
- AgencyName (NOT NULL)
- AgencyPhone (UNIQUE, NOT NULL)
- AgencyOffice

Table: AgencyDetails

Columns:

- AgencyName (PK)
- AgencyAddress

Table: BusInfo

Columns:

- BusRegnNo (PK)
- AgencyName (FK to Admin.AgencyName)
- TotalSeats (DEFAULT 40)
- AC (DEFAULT 0)
- LocationName
- Latitude
- Longitude

Table: BusSchedule

Columns:

- BusRegnNo (FK to BusInfo.BusRegnNo)
- RouteID (CHK > 0)
- StartTime (CHK >= 0 AND < 2400)
- Fare (CHK > 0)
- ReservedSeats (DEFAULT 0)
- TravelTime (CHK > 0)

PK: RouteID, StartTime

Table: TimeForTravel

Columns:

- TravelTime (PK, CHK > 0)
- StartTime (PK, CHK >= 0 AND < 24)
- EndTime (CHK >= 0 AND < 24)

Table: RouteDetails

Columns:

- RouteID (PK, CHK > 0)
- RouteName (NOT NULL)
- Source (NOT NULL)
- Destination (NOT NULL)
- AproxDistance

Table: BusStops

Columns:

- RouteID (FK to RouteDetails.RouteID, CHK > 0)
- IntermediateStops (NOT NULL)
- StopNumber (CHK > 0)

PK: RouteID, StopNumber

Table: DriverDetails

Columns:

- DriverID (PK)
- DriverName (NOT NULL)
- DriverPhone
- Age
- Date_Of_Join

Table: Ticket

Columns:

- TicketPNR (PK)
- BusRegnNo (FK to BusInfo.BusRegnNo)
- BookingDate
- TravelDate

CHK: TravelDate > BookingDate + 2

Table: SeatsBooked

Columns:

- TicketPNR (PK, FK to Ticket.TicketPNR)
- BookedSeats (PK)

Table: SeatInfo

Columns:

- BusRegnNo (PK, FK to BusInfo.BusRegnNo)
- SeatNo (CHK > 40)
- Sleeper (DEFAULT 0)

Table: Passenger

Columns:

- ID (FK to NonAdmin.ID)
- BusRegnNo (FK to BusInfo.BusRegnNo)
- PassengerID (PK, CHK > 0)
- PassengerName
- PassengeGender
- Age (CHK > 5)

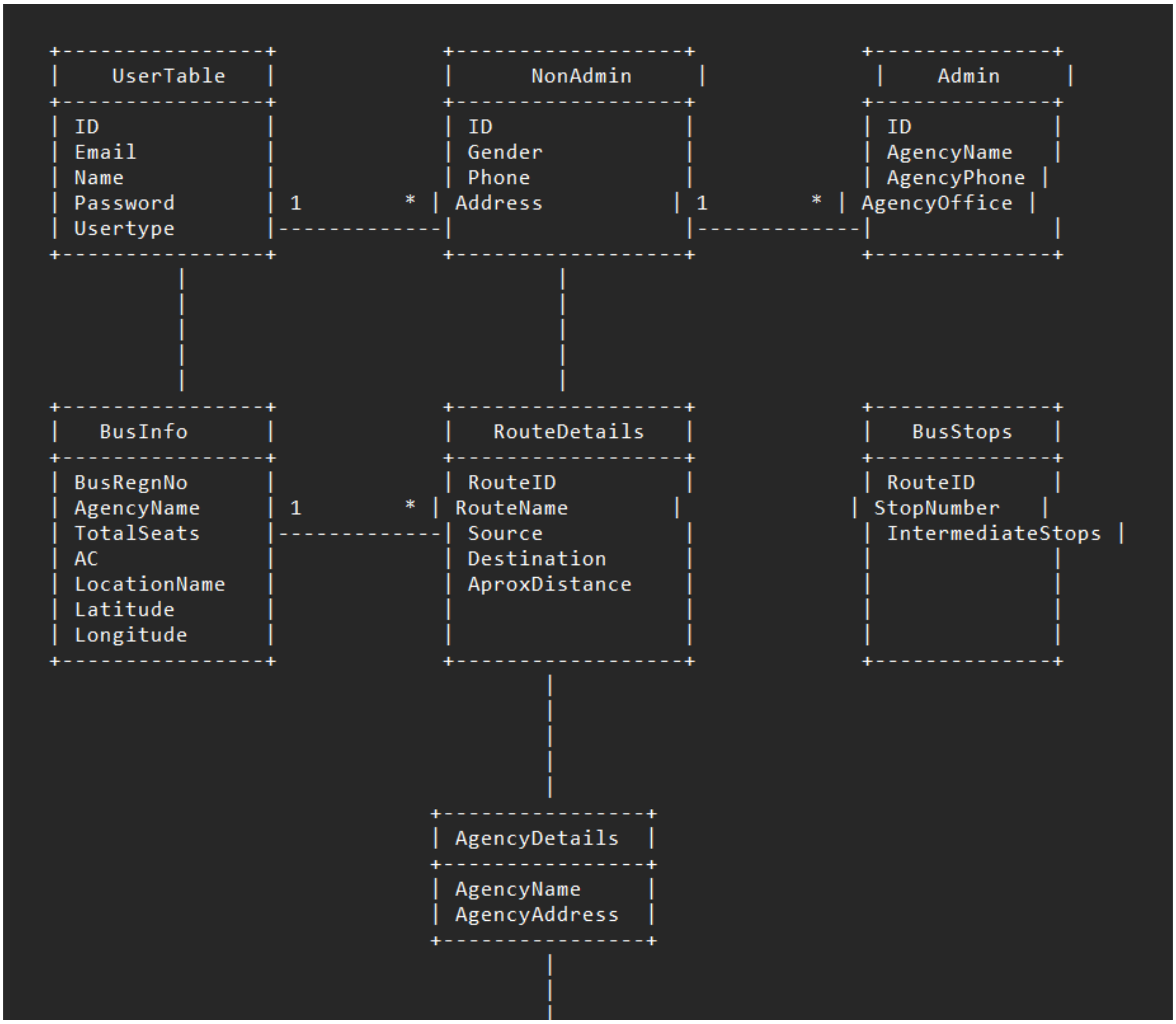
Table: Through

Columns:

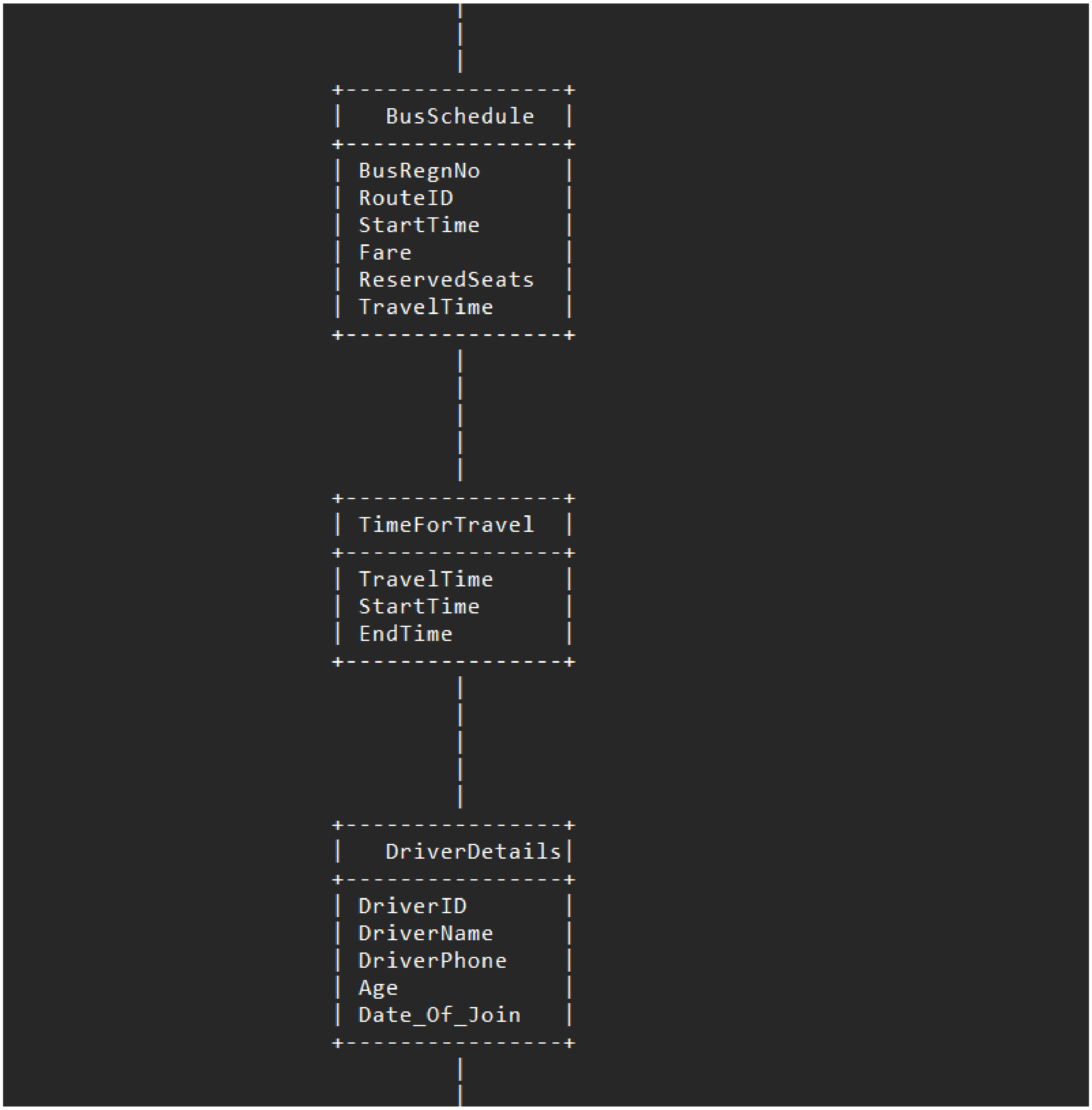
- RouteID (FK to RouteDetails.RouteID)
- DriverID (FK to DriverDetails.DriverID)
- BusRegnNo (FK to BusInfo.BusRegnNo)
- TicketPNR (FK to Ticket.TicketPNR, CHK > 0)

ER Diagram to Tables

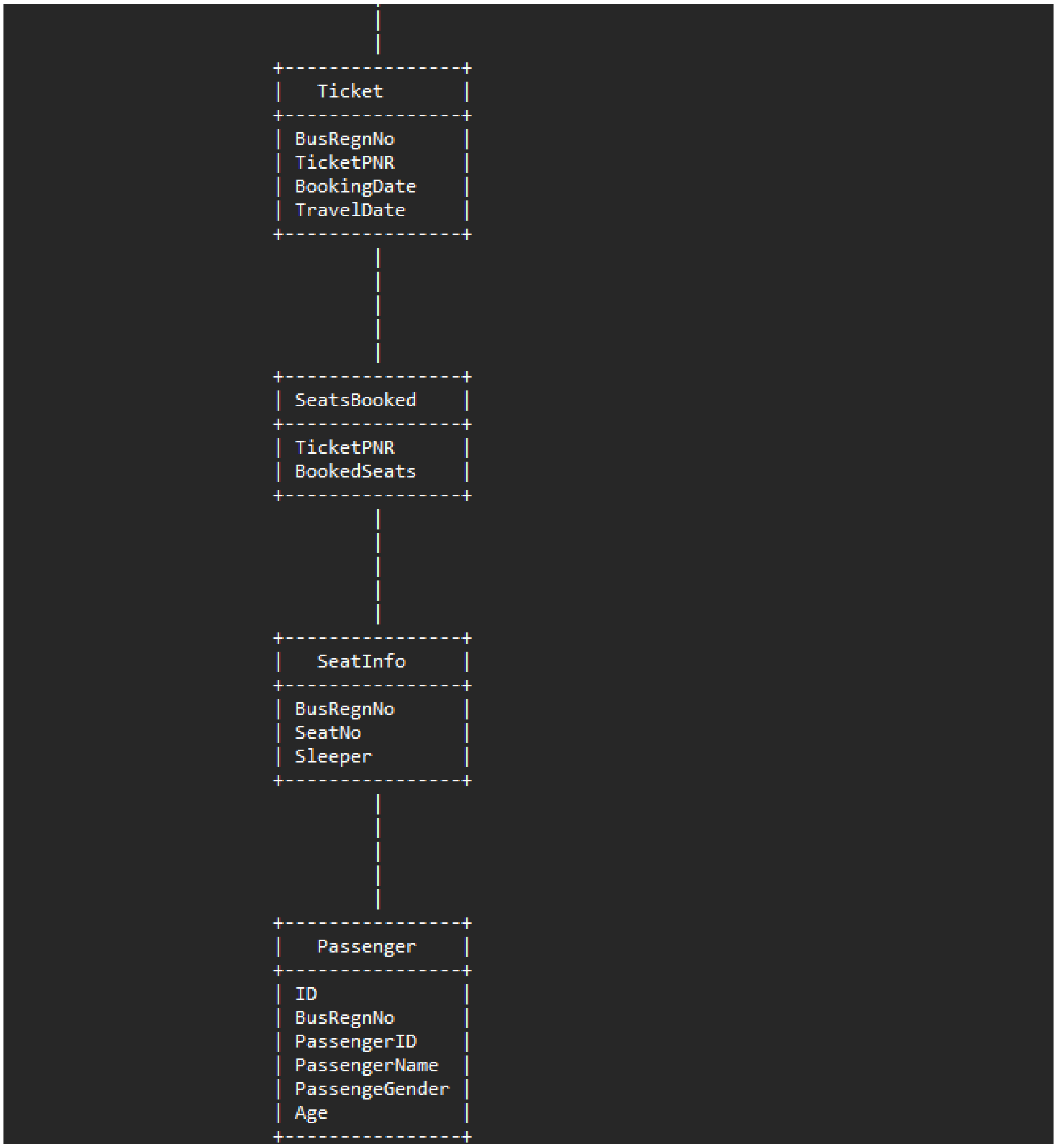
The tables are now in third normal form (3NF), and all data is properly organized to ensure data integrity and minimize data redundancy.



```
+----------------+             +-------------------+             +--------------+
|    UserTable   |             |      NonAdmin      |             |    Admin     |
+----------------+             +-------------------+             +--------------+
| ID             |             | ID                |             | ID           |
| Email          |             | Gender            |             | AgencyName   |
| Name           |             | Phone             |             | AgencyPhone |
| Password       | 1        * | Address           | 1        * | AgencyOffice |
| Usertype       |-------------|                   |-------------|              |
+----------------+             +-------------------+             +--------------+
          |                              |
          |                              |
          |                              |
          |                              |
          |                              |
+----------------+             +-------------------+             +--------------+
|   BusInfo      |             |   RouteDetails    |             |   BusStops   |
+----------------+             +-------------------+             +--------------+
| BusRegnNo      |             | RouteID           |             | RouteID      |
| AgencyName     | 1        * | RouteName         |             | StopNumber   |
| TotalSeats     |-------------| Source            |             | IntermediateStops |
| AC             |             | Destination       |             |              |
| LocationName   |             | AproxDistance     |             |              |
| Latitude       |             |                   |             |              |
| Longitude      |             |                   |             |              |
+----------------+             +-------------------+             +--------------+
                                        |
                                        |
                                        |
                                        |
                                        |
                              +-----------------+
                              | AgencyDetails  |
                              +-----------------+
                              | AgencyName     |
                              | AgencyAddress  |
                              +-----------------+
                                        |
                                        |
                                        |
```



```
         |
         |
         |
+----------------+
|   BusSchedule  |
+----------------+
| BusRegnNo      |
| RouteID        |
| StartTime      |
| Fare           |
| ReservedSeats  |
| TravelTime     |
+----------------+
         |
         |
         |
         |
         |
+----------------+
| TimeForTravel  |
+----------------+
| TravelTime     |
| StartTime      |
| EndTime        |
+----------------+
         |
         |
         |
         |
         |
+----------------+
|   DriverDetails|
+----------------+
| DriverID       |
| DriverName     |
| DriverPhone    |
| Age            |
| Date_Of_Join   |
+----------------+
         |
         |
```



```
                                 |
                                 |
                        +----------------+
                        |   Ticket       |
                        +----------------+
                        | BusRegnNo      |
                        | TicketPNR      |
                        | BookingDate    |
                        | TravelDate     |
                        +----------------+
                                 |
                                 |
                                 |
                                 |
                                 |
                        +----------------+
                        | SeatsBooked    |
                        +----------------+
                        | TicketPNR      |
                        | BookedSeats    |
                        +----------------+
                                 |
                                 |
                                 |
                                 |
                                 |
                        +----------------+
                        |   SeatInfo     |
                        +----------------+
                        | BusRegnNo      |
                        | SeatNo         |
                        | Sleeper        |
                        +----------------+
                                 |
                                 |
                                 |
                                 |
                                 |
                        +----------------+
                        |   Passenger    |
                        +----------------+
                        | ID             |
                        | BusRegnNo      |
                        | PassengerID    |
                        | PassengerName  |
                        | PassengeGender |
                        | Age            |
                        +----------------+
```

Here is the relational database schema based on the ER diagram:

```
UserTable (
  ID INT PRIMARY KEY,
  Email VARCHAR(255) UNIQUE NOT NULL,
  Name VARCHAR(255) NOT NULL,
  Password VARCHAR(255) NOT NULL,
  Usertype VARCHAR(255) NOT NULL)
```

```
NonAdmin (
  ID INT PRIMARY KEY,
  Gender VARCHAR(255) NOT NULL,
  Phone VARCHAR(255) NOT NULL,
  Address VARCHAR(255),
  FOREIGN KEY (ID) REFERENCES UserTable(ID) ON DELETE CASCADE
)

Admin (
  ID INT PRIMARY KEY,
  AgencyName VARCHAR(255) NOT NULL,
  AgencyPhone VARCHAR(255) NOT NULL,
  AgencyOffice VARCHAR(255) NOT NULL,
  FOREIGN KEY (ID) REFERENCES UserTable(ID) ON DELETE CASCADE
)

BusInfo (
  BusRegnNo VARCHAR(255) PRIMARY KEY,
  AgencyName VARCHAR(255) NOT NULL,
  TotalSeats INT NOT NULL,
  AC BOOLEAN NOT NULL,
  LocationName VARCHAR(255),
  Latitude DECIMAL(10,8),
  Longitude DECIMAL(11,8),
  FOREIGN KEY (AgencyName) REFERENCES Admin(AgencyName) ON DELETE CASCADE
)
```

```
RouteDetails (
  RouteID INT PRIMARY KEY,
  RouteName VARCHAR(255) NOT NULL,
  Source VARCHAR(255) NOT NULL,
  Destination VARCHAR(255) NOT NULL,
  AproxDistance DECIMAL(10,2) NOT NULL
)

BusStops (
  RouteID INT NOT NULL,
  StopNumber INT NOT NULL,
  IntermediateStops VARCHAR(255),
  PRIMARY KEY (RouteID, StopNumber),
  FOREIGN KEY (RouteID) REFERENCES RouteDetails(RouteID) ON DELETE CASCADE
)

AgencyDetails (
  AgencyName VARCHAR(255) PRIMARY KEY,
  AgencyAddress VARCHAR(255) NOT NULL
)

BusSchedule (
  BusRegnNo VARCHAR(255) NOT NULL,
  RouteID INT NOT NULL,
  StartTime TIME NOT NULL,
  Fare DECIMAL(10,2) NOT NULL,
  ReservedSeats INT NOT NULL,
  TravelTime TIME NOT NULL,
  PRIMARY KEY (BusRegnNo, RouteID, StartTime),
  FOREIGN KEY (BusRegnNo) REFERENCES BusInfo(BusRegnNo) ON DELETE CASCADE,
  FOREIGN KEY (RouteID) REFERENCES RouteDetails(RouteID) ON DELETE CASCADE
)

TimeForTravel (
  TravelTime TIME PRIMARY KEY,
  StartTime TIME NOT NULL,
  EndTime TIME NOT NULL
)

DriverDetails (
  DriverID INT PRIMARY KEY,
  DriverName VARCHAR(255) NOT NULL,
  DriverPhone VARCHAR(255) NOT NULL,
  Age INT NOT NULL,
  Date_Of_Join DATE NOT NULL
)

Ticket (
  BusRegnNo VARCHAR(255) NOT NULL,
  TicketPNR VARCHAR(255) PRIMARY KEY,
  BookingDate DATE NOT NULL,
  TravelDate DATE NOT NULL,
  FOREIGN KEY (BusRegnNo) REFERENCES BusInfo(BusRegnNo) ON DELETE CASCADE
)

SeatsBooked (
  TicketPNR VARCHAR(255) NOT NULL,
  BookedSeats VARCHAR(255) NOT NULL,
  PRIMARY KEY (TicketPNR),
  FOREIGN KEY (TicketPNR) REFERENCES Ticket(TicketPNR) ON DELETE CASCADE)
```

```
SeatInfo (
  BusRegnNo VARCHAR(255) NOT NULL,
  SeatNo INT NOT NULL,
  Sleeper BOOLEAN NOT NULL,
  PRIMARY KEY (BusRegnNo, SeatNo),
  FOREIGN KEY (BusRegnNo) REFERENCES BusInfo(BusRegnNo) ON DELETE CASCADE
)
```

SQL (Code & Outputs)

```
CREATE TABLE UserTable (
ID NUMBER,
Email VARCHAR(40) NOT NULL UNIQUE,
Name VARCHAR(30),
Password VARCHAR(15) NOT NULL,
Usertype VARCHAR(10) NOT NULL,
PRIMARY KEY (ID)
);

CREATE TABLE NonAdmin (
ID NUMBER,
Gender VARCHAR(7),
Phone NUMERIC(10,0) NOT NULL UNIQUE CONSTRAINT Phone1_chk CHECK (Phone > 999999999),
Address VARCHAR(100),
PRIMARY KEY(ID),
FOREIGN KEY(ID) REFERENCES UserTable(ID)
);

CREATE TABLE Admin (
ID NUMBER,
AgencyName VARCHAR(35) NOT NULL,
AgencyPhone VARCHAR(10) NOT NULL UNIQUE CONSTRAINT Phone2_chk CHECK (AgencyPhone > 999999999),
AgencyOffice VARCHAR(50),
PRIMARY KEY(AgencyName),
FOREIGN KEY(ID) REFERENCES UserTable(ID)
);
```

```
CREATE TABLE BusInfo (
BusRegnNo VARCHAR(15) NOT NULL,
AgencyName VARCHAR(35) NOT NULL,
TotalSeats INTEGER DEFAULT 40,
AC NUMERIC(1) DEFAULT 0,
LocationName VARCHAR(20),
Latitude NUMERIC(17,10),
Longitude NUMERIC(17,10),
PRIMARY KEY(BusRegnNo),
FOREIGN KEY(AgencyName) REFERENCES Admin(AgencyName)
);

CREATE TABLE AgencyDetails (
AgencyName VARCHAR(35) NOT NULL,
AgencyAddress VARCHAR(50) NOT NULL,
PRIMARY KEY (AgencyName)
);

CREATE TABLE BusSchedule (
BusRegnNo VARCHAR(15) NOT NULL,
RouteID INTEGER NOT NULL CONSTRAINT Routeid1_chk CHECK (RouteID > 0),
StartTime NUMERIC(4,2) CONSTRAINT Start_chk1 CHECK (StartTime >= 0 AND StartTime < 2400),
Fare INTEGER CONSTRAINT Fair_chk CHECK (Fare > 0),
ReservedSeats INTEGER DEFAULT 0,
TravelTime NUMERIC(10) CONSTRAINT EstTim1_chk CHECK (TravelTime > 0),
PRIMARY KEY(RouteID, StartTime),
FOREIGN KEY(BusRegnNo) REFERENCES BusInfo(BusRegnNo)
);
```

```
CREATE TABLE TimeForTravel (
TravelTime NUMERIC(10) CONSTRAINT EstTim2_chk CHECK (TravelTime > 0),
StartTime NUMERIC(4,2) CONSTRAINT Start1_chk CHECK (StartTime >= 0 AND StartTime < 24),
EndTime NUMERIC(4,2) CONSTRAINT End1_chk CHECK (EndTime >= 0 AND EndTime < 24),
PRIMARY KEY(TravelTime, StartTime)
);

CREATE TABLE RouteDetails (
RouteID INTEGER CONSTRAINT Routeid4_chk CHECK (RouteID > 0),
RouteName VARCHAR(30) NOT NULL,
Source VARCHAR(30) NOT NULL,
Destination VARCHAR(30) NOT NULL,
AproxDistance NUMERIC(6,2),
PRIMARY KEY (RouteID)
);

CREATE TABLE BusStops (
RouteID INTEGER NOT NULL CONSTRAINT Routeid_chk CHECK (RouteID > 0),
IntermediateStops VARCHAR(20) NOT NULL,
StopNumber INTEGER NOT NULL CONSTRAINT Stop_num CHECK (StopNumber > 0),
PRIMARY KEY(RouteID, StopNumber)
);
CREATE TABLE DriverDetails (
DriverID integer,
DriverName VARCHAR(20) NOT NULL,
DriverPhone NUMERIC(10),
Age NUMERIC(3),
Date_Of_Join DATE,
PRIMARY KEY (DriverID)
);

CREATE TABLE Ticket (
BusRegnNo VARCHAR(15) NOT NULL,
TicketPNR NUMBER,
BookingDate DATE,
TravelDate DATE,
primary key(TicketPNR),
constraint dat_chk check(TravelDate>BookingDate+2),
foreign key(BusRegnNo) references BusInfo(BusRegnNo)
);

create table SeatsBooked(
TicketPNR integer,
BookedSeats integer,
primary key(TicketPNR,BookedSeats)
);
--drop table SeatsBooked;

create table SeatInfo(
BusRegnNo varchar(15) not null unique,
SeatNo integer constraint SeatNo_chk check(SeatNo>40),
Sleeper numeric(1) default 0,
foreign key(BusRegnNo) references BusInfo(BusRegnNo) on delete cascade
);

create table Passenger(
ID number,
BusRegnNo varchar(15) not null,
PassengerID integer constraint passid_chk check(PassengerID>0),
PassengerName varchar(20),
PassengeGender varchar(7),
Age integer constraint age2_chk check(age>5),
primary key(PassengerID),
foreign key(ID) references NonAdmin(ID) on delete cascade,
foreign key(BusRegnNo) references BusInfo(BusRegnNo)
);

create table Through(
RouteID integer,
DriverID integer,
BusRegnNo varchar(50) not null,
TicketPNR integer not null constraint pnr4_chk check(TicketPNR>0),
foreign key (RouteID) references RouteDetails(RouteID),
foreign key (DriverID) references DriverDetails(DriverID),
foreign key (TicketPNR) references Ticket(TicketPNR)
);


--drop table Through;
CREATE OR REPLACE TRIGGER at_least_one_trips
BEFORE  UPDATE ON through
FOR EACH ROW
DECLARE
trip_count NUMBER;
BEGIN
SELECT COUNT(*) INTO trip_count
FROM through
```

```
WHERE BusRegnNo = :NEW.BusRegnNo;
IF trip_count = 0 THEN
RAISE_APPLICATION_ERROR(-20001, 'At least one trip must exist for the given bus registration number.');
END IF;
END;

INSERT INTO UserTable (ID, Email, Name, Password, Usertype)
VALUES (1, 'johndoe@example.com', 'John Doe', 'mypassword', 'Admin');
INSERT INTO UserTable (ID, Email, Name, Password, Usertype)
VALUES (2, 'janesmith@example.com', 'Jane Smith', 'myotherpassword', 'Non-Admin');
INSERT INTO UserTable (ID, Email, Name, Password, Usertype)
VALUES (3, 'bobross@example.com', 'Bob Ross', 'happylittletree', 'Non-Admin');
INSERT INTO UserTable (ID, Email, Name, Password, Usertype)
VALUES (4, 'foo@example.com', 'Foo Bar', 'foobar', 'Admin');

INSERT INTO NonAdmin (ID, Gender, Phone, Address)
VALUES (2, 'Female', 1234567890, '123 Main St');
INSERT INTO NonAdmin (ID, Gender, Phone, Address)
VALUES (3, 'Male', 9876543210, '456 Elm St');
INSERT INTO NonAdmin (ID, Gender, Phone, Address)
VALUES (4, 'Male', 5555555555, '789 Oak St');
INSERT INTO NonAdmin (ID, Gender, Phone, Address)
VALUES (1, 'Female', 1111111111, '999 Maple St');

INSERT INTO AgencyDetails (AgencyName, AgencyAddress)
VALUES ('ABC Travels', '123 Main Street');
INSERT INTO AgencyDetails (AgencyName, AgencyAddress)
VALUES('XYZ Bus Lines', '456 Elm Street');
INSERT INTO AgencyDetails (AgencyName, AgencyAddress)
```

```
VALUES ('MNO Tours', '789 Oak Street');
INSERT INTO AgencyDetails (AgencyName, AgencyAddress)
VALUES ('PQR Transport', '1011 Maple Street');



INSERT INTO Admin (ID, AgencyName, AgencyPhone, AgencyOffice)
VALUES (1, 'ABC Bus Company', '5555555555', '123 Main St');
INSERT INTO Admin (ID, AgencyName, AgencyPhone, AgencyOffice)
VALUES (2, 'XYZ Bus Company', '5555555565', '789 Elm St');


INSERT INTO BusInfo (BusRegnNo, AgencyName, TotalSeats, AC, LocationName, Latitude, Longitude)
VALUES ('BUS1234', 'ABC Bus Company', 50, 1, 'New York City', 40.7128, -74.0060);
INSERT INTO BusInfo (BusRegnNo, AgencyName, TotalSeats, AC, LocationName, Latitude, Longitude)
VALUES ('BUS5678', 'ABC Bus Company', 40, 0, 'Boston', 42.3601, -71.0589);
INSERT INTO BusInfo (BusRegnNo, AgencyName, TotalSeats, AC, LocationName, Latitude, Longitude)
VALUES ('BUS91011', 'XYZ Bus Company', 30, 1, 'Chicago', 41.8781, -87.6298);
INSERT INTO BusInfo (BusRegnNo, AgencyName, TotalSeats, AC, LocationName, Latitude, Longitude)
VALUES ('BUS121314', 'XYZ Bus Company', 60, 0, 'Los Angeles', 34.0522, -118.2437);


INSERT INTO BusSchedule (BusRegnNo, RouteID, StartTime, Fare, ReservedSeats, TravelTime)
VALUES ('BUS5678', 2, 18.00, 700, 15, 7);
INSERT INTO BusSchedule (BusRegnNo, RouteID, StartTime, Fare, ReservedSeats, TravelTime)
VALUES ('BUS91011', 3, 10.30, 900, 10, 9);
INSERT INTO BusSchedule (BusRegnNo, RouteID, StartTime, Fare, ReservedSeats, TravelTime)
VALUES ('BUS121314', 4, 23.15, 1100, 20, 12);
```

```
INSERT INTO TimeForTravel (TravelTime, StartTime, EndTime)
VALUES (8.5, 6.0, 14.5);
INSERT INTO TimeForTravel (TravelTime, StartTime, EndTime)
VALUES (7.0, 8.0, 15.0);
INSERT INTO TimeForTravel (TravelTime, StartTime, EndTime)
VALUES (9.5, 10.0, 19.5);
INSERT INTO TimeForTravel (TravelTime, StartTime, EndTime)
VALUES (12.0, 12.0, 23.0);

INSERT INTO BusStops VALUES (1, 'Stop1', 1);
INSERT INTO BusStops VALUES (1, 'Stop2', 2);
INSERT INTO BusStops VALUES (2, 'Stop3', 1);
INSERT INTO BusStops VALUES (2, 'Stop4', 2);

INSERT INTO DriverDetails(DriverID,DriverName,DriverPhone,Age) VALUES (1, 'John Doe', 1234567890, 35);
INSERT INTO DriverDetails (DriverID,DriverName,DriverPhone,Age) VALUES (2, 'Jane Smith', 0987654321, 28);
INSERT INTO DriverDetails (DriverID,DriverName,DriverPhone,Age) VALUES (3, 'Robert Lee', 5551234567, 42);
INSERT INTO DriverDetails (DriverID,DriverName,DriverPhone,Age) VALUES (4, 'Susan Brown', 1112223333, 23);



insert into DriverDetails values('126', 'Prithvi', '9834534565','29','17-MAY-17');

insert into BusStops values('4','Chennai','7');

--delete from TimeForTravel;
```

```
insert into RouteDetails values(4,'BGL-CNI','Bangalore','Chennai',500);
insert into RouteDetails values(3,'CNI-BGL','Chennai','Bangalore',500);
insert into RouteDetails values(2,'BGL-TRR','Bangalore','Tirupur',700);
insert into RouteDetails values(1,'TRR-BGL','Tirupur','Bangalore',700);

insert into TimeForTravel values('12','17.30','05.30');
--alter table Through modify column BusRegnNo varchar(50);



insert into TimeForTravel values('15','19.00','10.00');
insert into TimeForTravel values('14','19.45','9.45');
insert into TimeForTravel values('15','19.15','10.15');
insert into TimeForTravel values('17','19.00','12.00');
insert into TimeForTravel values('15','20.00','11.00');

INSERT INTO Ticket (BusRegnNo, TicketPNR, BookingDate, TravelDate)
VALUES ('BUS121314', '1006', '25-APR-2023', '30-APR-2023');

INSERT INTO Ticket (BusRegnNo, TicketPNR, BookingDate, TravelDate)
VALUES ('BUS5678', '1007', '01-MAY-2023', '04-MAY-2023');

INSERT INTO Ticket (BusRegnNo, TicketPNR, BookingDate, TravelDate)
VALUES ('BUS91011', '1009', '25-APR-2023', '30-APR-2023');

INSERT INTO Ticket (BusRegnNo, TicketPNR, BookingDate, TravelDate)
VALUES ('BUS121314', '1008', '01-MAY-2023', '04-MAY-2023');



--drop table Ticket;
```

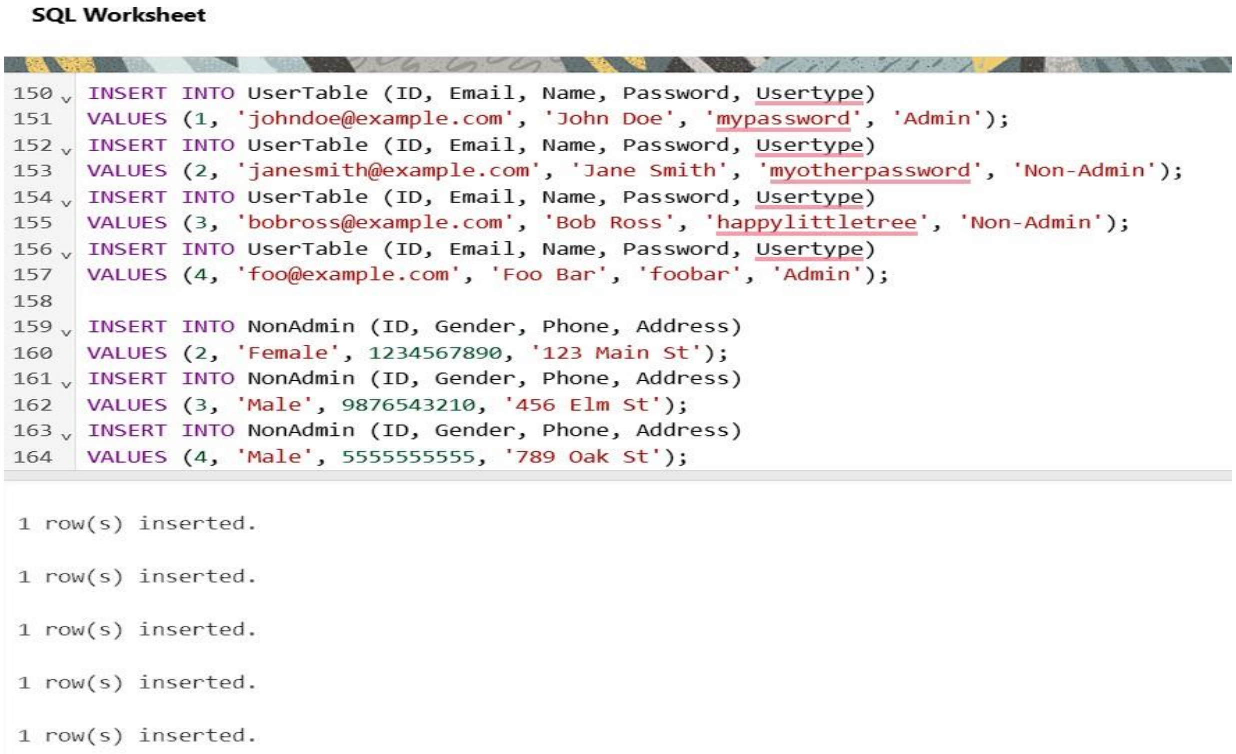
```
insert into SeatsBooked values('1006','25');
insert into SeatsBooked values('1006','26');
insert into SeatsBooked values('1006','27');
insert into SeatsBooked values('1007','20');
insert into SeatsBooked values('1007','21');
insert into SeatsBooked values('1008','25');
insert into SeatsBooked values('1009','14');
insert into SeatsBooked values('1009','15');
insert into SeatsBooked values('1009','16');
insert into SeatsBooked values('1009','17');
insert into SeatsBooked values('1010','1');
insert into SeatsBooked values('1011','10');
insert into SeatsBooked values('1011','12');
insert into SeatsBooked values('1011','19');


--alter table TimeForTravel modify column StartTime numeric(4,2);



INSERT INTO Through (RouteID, DriverID, BusRegnNo, TicketPNR)
VALUES (1, 1, 'BUS121314', '1006');
INSERT INTO Through (RouteID, DriverID,  BusRegnNo, TicketPNR)
VALUES (2, 2, 'BUS5678', '1007');
INSERT INTO Through (RouteID, DriverID, BusRegnNo, TicketPNR)
VALUES (3, 3,  'BUS91011', '1009');
INSERT INTO Through (RouteID, DriverID, BusRegnNo, TicketPNR)
VALUES (4, 4, 'BUS121314', '1008');
```

```sql
CREATE OR REPLACE TRIGGER TimeTravel
after insert on BusSchedule
for each row
begin
if(:new.TravelTime + :new.StartTime > 24.00) then
insert into TimeForTravel(TravelTime, StartTime, EndTime)
values(:new.TravelTime, :new.StartTime, :new.TravelTime + :new.StartTime - 24);
else
insert into TimeForTravel(TravelTime, StartTime, EndTime)
values(:new.TravelTime, :new.StartTime, :new.TravelTime + :new.StartTime);
end if;
end;
```

```sql
CREATE OR REPLACE procedure totalrevenue
AS
cursor cur is
select AgencyName, sum(fare) as AGENCYNAMESUM
from BusInfo
natural join BusSchedule
group by AgencyName;
v_agencyname BusInfo.AgencyName%TYPE;
v_agencysum BusSchedule.Fare%TYPE;
begin
open cur;
loop
fetch cur into v_agencyname, v_agencysum;
exit when cur%NOTFOUND;
dbms_output.put_line(v_agencyname || ': ' || v_agencysum);
end loop;
close cur; end;
```

```
-- Call the procedure
call totalrevenue;

select AgencyName,sum(fare) from BusInfo natural join BusSchedule group by AgencyName;

SELECT *
FROM UserTable
LEFT JOIN NonAdmin ON UserTable.ID = NonAdmin.ID
LEFT JOIN Admin ON UserTable.ID = Admin.ID
LEFT JOIN BusInfo ON Admin.AgencyName = BusInfo.AgencyName
LEFT JOIN AgencyDetails ON Admin.AgencyName = AgencyDetails.AgencyName
LEFT JOIN BusSchedule ON BusInfo.BusRegnNo = BusSchedule.BusRegnNo
LEFT JOIN TimeForTravel ON BusSchedule.TravelTime = TimeForTravel.TravelTime AND
BusSchedule.StartTime = TimeForTravel.StartTime
LEFT JOIN RouteDetails ON BusSchedule.RouteID = RouteDetails.RouteID
LEFT JOIN BusStops ON BusSchedule.RouteID = BusStops.RouteID
LEFT JOIN Ticket ON BusSchedule.BusRegnNo= Ticket.BusRegnNo
LEFT JOIN SeatsBooked ON Ticket.TicketPNR = SeatsBooked.TicketPNR
LEFT JOIN SeatInfo ON BusSchedule.BusRegnNo = SeatInfo.BusRegnNo
LEFT JOIN Passenger ON Passenger.BusRegnNo = BusSchedule.BusRegnNo
LEFT JOIN Through ON BusSchedule.RouteID = Through.RouteID  AND
BusSchedule.BusRegnNo = Through.BusRegnNo
```

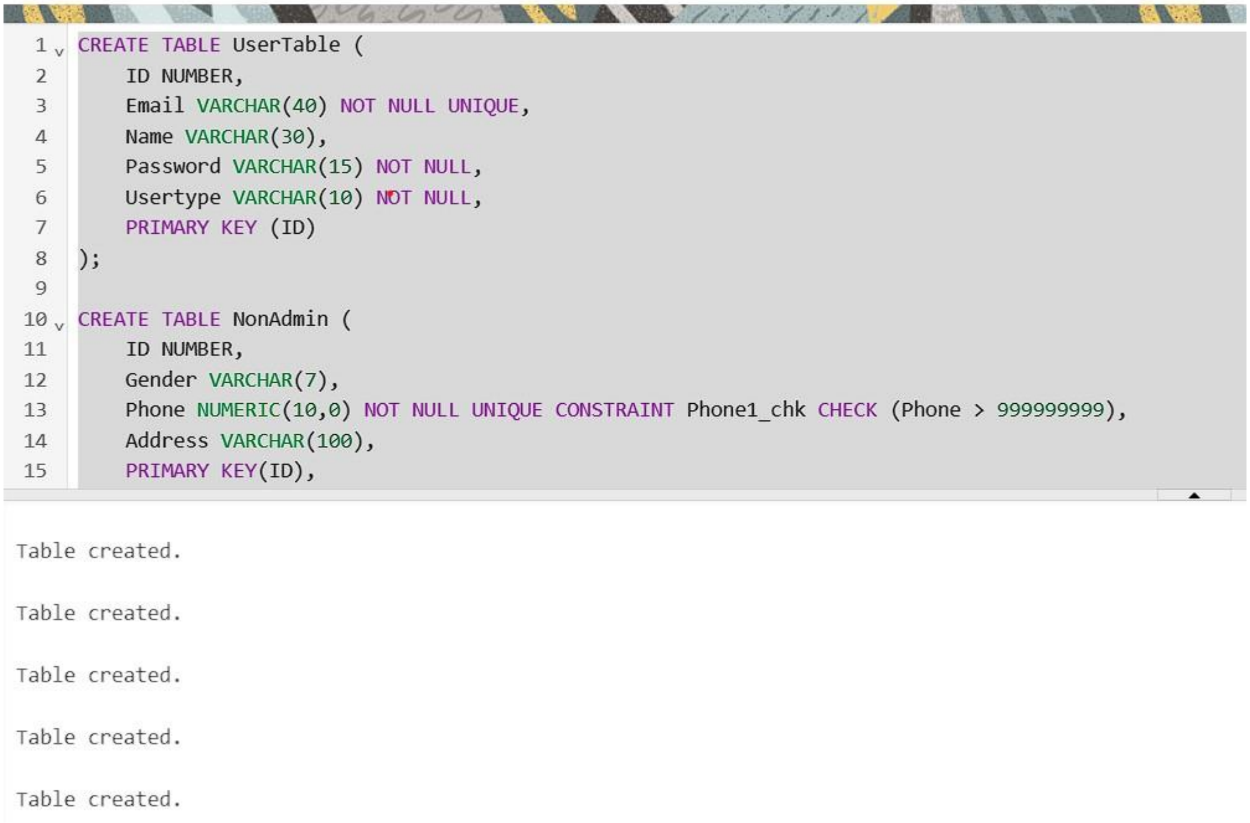
OUTPUT



```
INSERT INTO UserTable (ID, Email, Name, Password, Usertype)
VALUES (1, 'johndoe@example.com', 'John Doe', 'mypassword', 'Admin');
INSERT INTO UserTable (ID, Email, Name, Password, Usertype)
VALUES (2, 'janesmith@example.com', 'Jane Smith', 'myotherpassword', 'Non-Admin');
INSERT INTO UserTable (ID, Email, Name, Password, Usertype)
VALUES (3, 'bobross@example.com', 'Bob Ross', 'happylittletree', 'Non-Admin');
INSERT INTO UserTable (ID, Email, Name, Password, Usertype)
VALUES (4, 'foo@example.com', 'Foo Bar', 'foobar', 'Admin');

INSERT INTO NonAdmin (ID, Gender, Phone, Address)
VALUES (2, 'Female', 1234567890, '123 Main St');
INSERT INTO NonAdmin (ID, Gender, Phone, Address)
VALUES (3, 'Male', 9876543210, '456 Elm St');
INSERT INTO NonAdmin (ID, Gender, Phone, Address)
VALUES (4, 'Male', 5555555555, '789 Oak St');
```

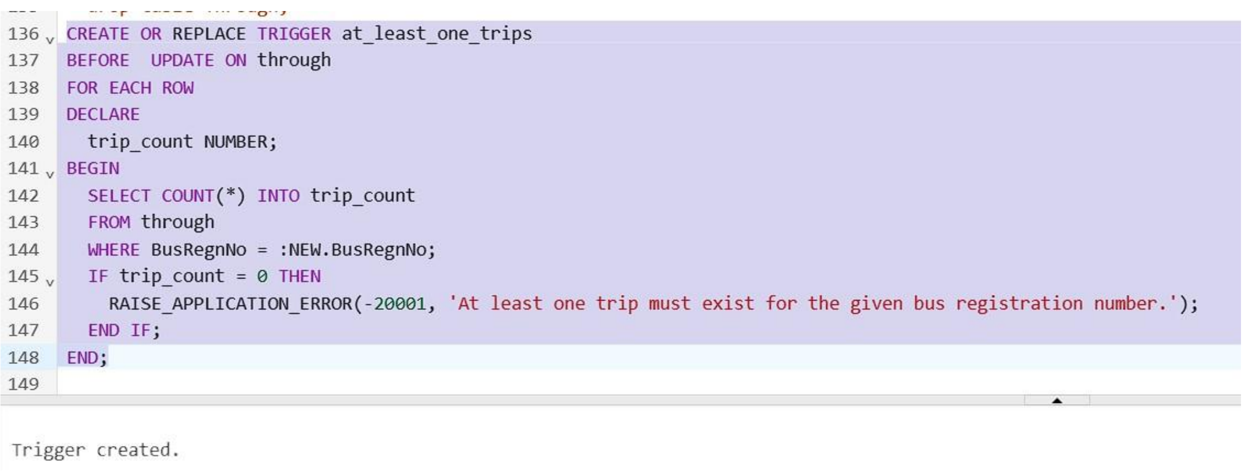
```
1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.

1 row(s) inserted.
```

SQL Worksheet



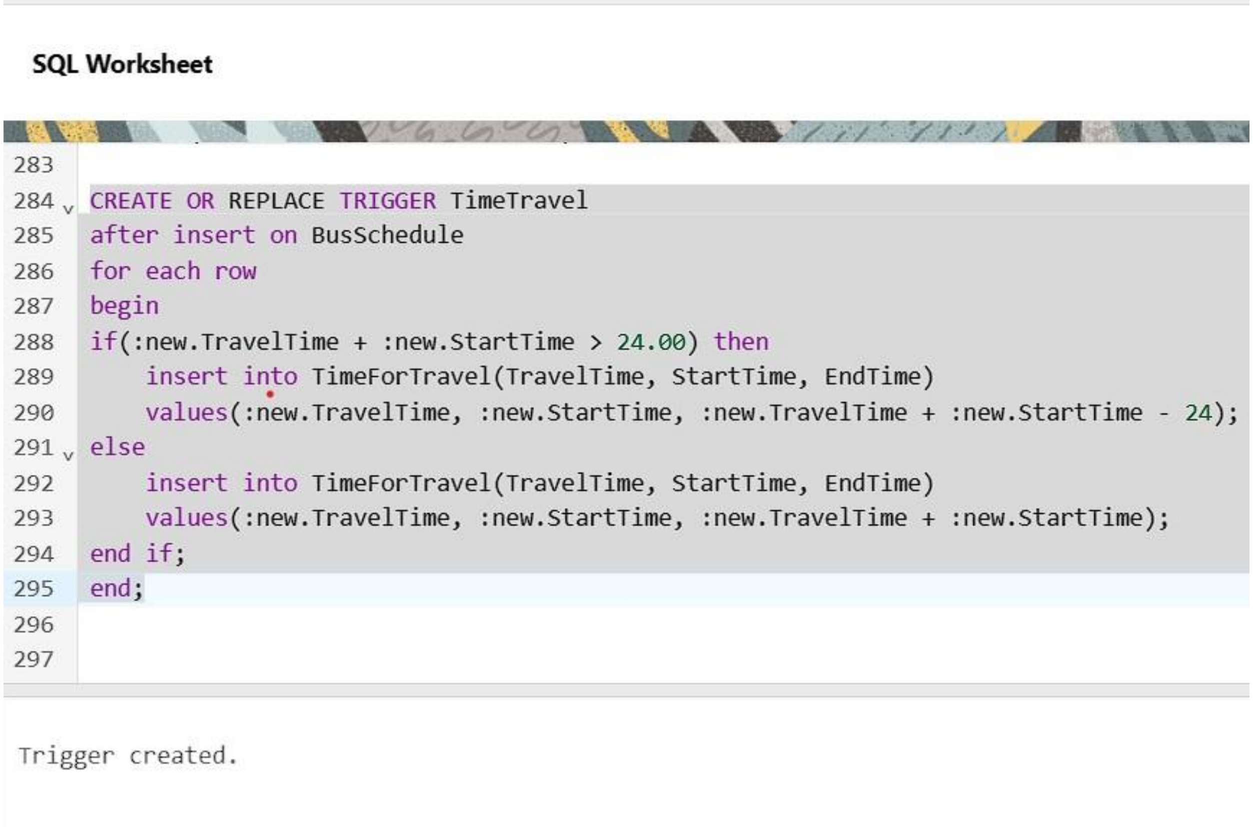
```
CREATE TABLE UserTable (
    ID NUMBER,
    Email VARCHAR(40) NOT NULL UNIQUE,
    Name VARCHAR(30),
    Password VARCHAR(15) NOT NULL,
    Usertype VARCHAR(10) NOT NULL,
    PRIMARY KEY (ID)
);

CREATE TABLE NonAdmin (
    ID NUMBER,
    Gender VARCHAR(7),
    Phone NUMERIC(10,0) NOT NULL UNIQUE CONSTRAINT Phone1_chk CHECK (Phone > 999999999),
    Address VARCHAR(100),
    PRIMARY KEY(ID),
```

```
Table created.

Table created.

Table created.

Table created.

Table created.
```



```
  drop table through;
CREATE OR REPLACE TRIGGER at_least_one_trips
BEFORE  UPDATE ON through
FOR EACH ROW
DECLARE
  trip_count NUMBER;
BEGIN
  SELECT COUNT(*) INTO trip_count
  FROM through
  WHERE BusRegnNo = :NEW.BusRegnNo;
  IF trip_count = 0 THEN
    RAISE_APPLICATION_ERROR(-20001, 'At least one trip must exist for the given bus registration number.');
  END IF;
END;
```

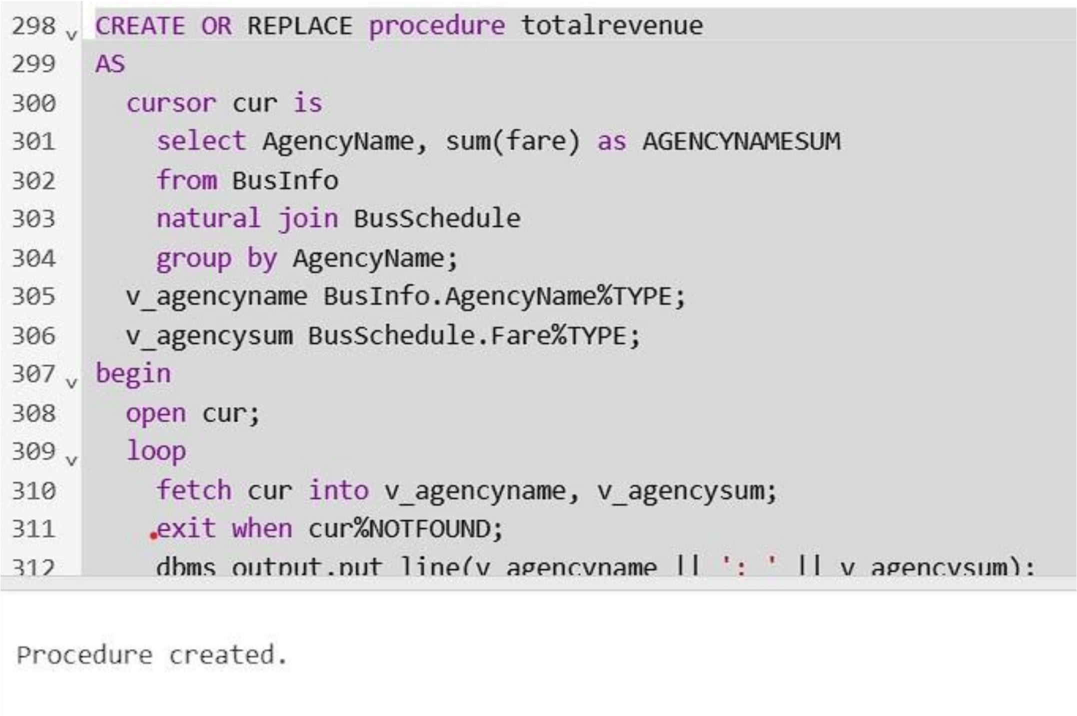
```
Trigger created.
```



SQL Worksheet

```
CREATE OR REPLACE TRIGGER TimeTravel
after insert on BusSchedule
for each row
begin
if(:new.TravelTime + :new.StartTime > 24.00) then
    insert into TimeForTravel(TravelTime, StartTime, EndTime)
    values(:new.TravelTime, :new.StartTime, :new.TravelTime + :new.StartTime - 24);
else
    insert into TimeForTravel(TravelTime, StartTime, EndTime)
    values(:new.TravelTime, :new.StartTime, :new.TravelTime + :new.StartTime);
end if;
end;
```

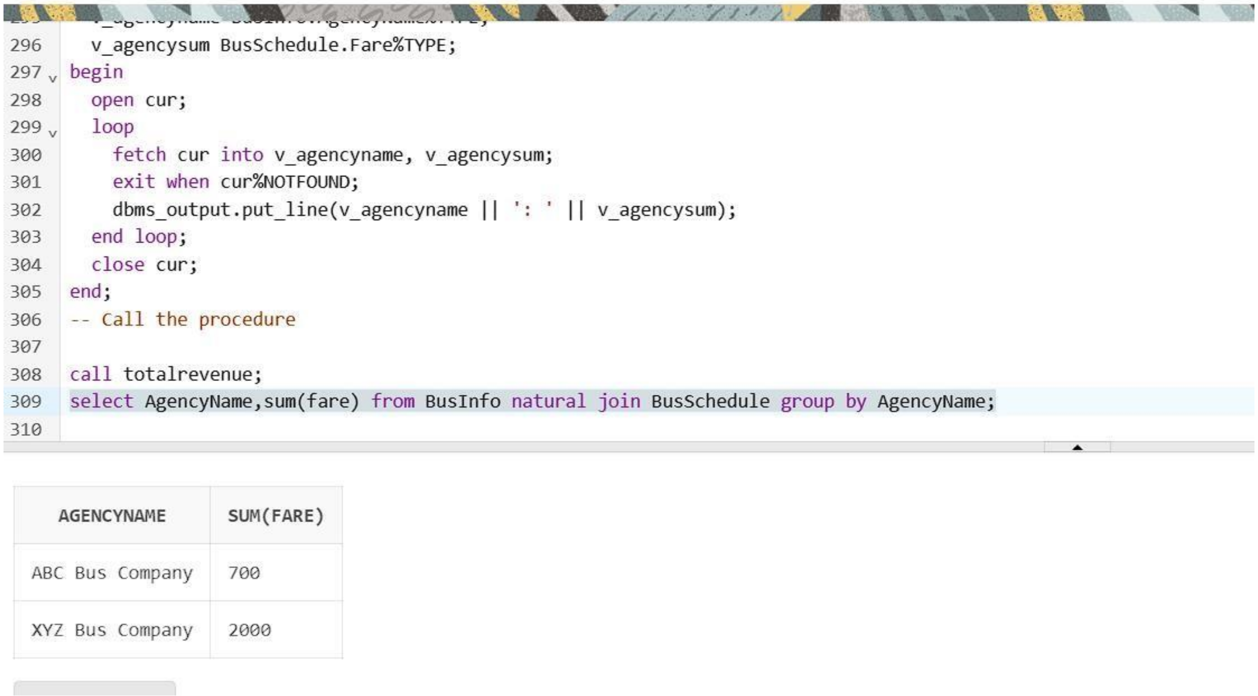
```
Trigger created.
```



```
CREATE OR REPLACE procedure totalrevenue
AS
  cursor cur is
    select AgencyName, sum(fare) as AGENCYNAMESUM
    from BusInfo
    natural join BusSchedule
    group by AgencyName;
  v_agencyname BusInfo.AgencyName%TYPE;
  v_agencysum BusSchedule.Fare%TYPE;
begin
  open cur;
  loop
    fetch cur into v_agencyname, v_agencysum;
    exit when cur%NOTFOUND;
    dbms_output.put_line(v_agencyname || ': ' || v_agencysum);
```

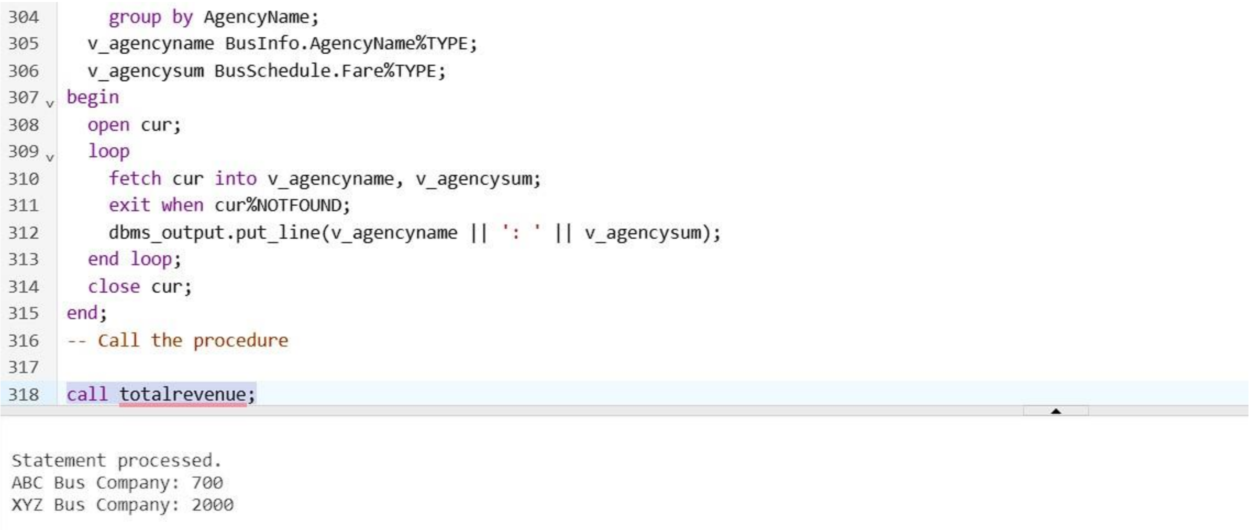
```
Procedure created.
```

SQL Worksheet



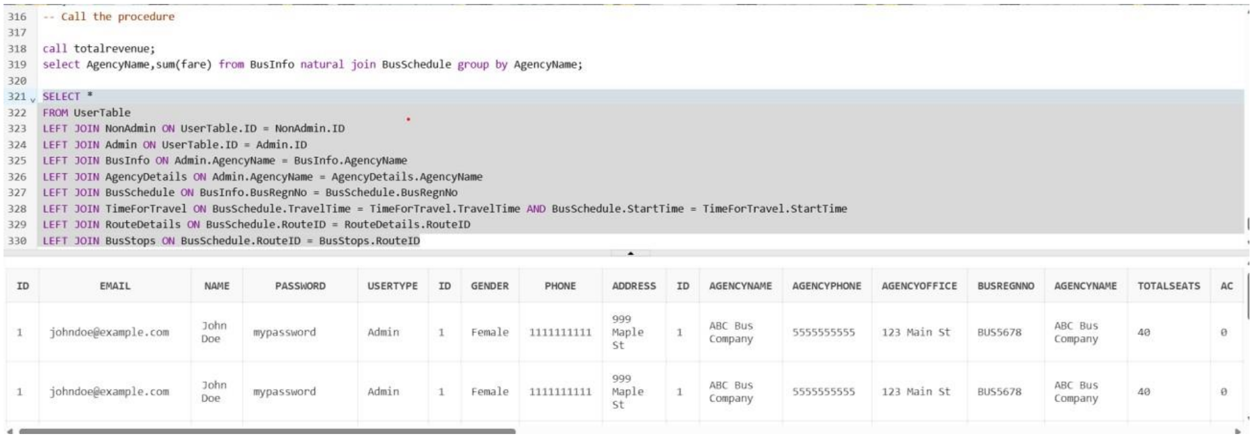
```
  v_agencyname BusInfo.AgencyName%TYPE;
  v_agencysum BusSchedule.Fare%TYPE;
begin
  open cur;
  loop
    fetch cur into v_agencyname, v_agencysum;
    exit when cur%NOTFOUND;
    dbms_output.put_line(v_agencyname || ': ' || v_agencysum);
  end loop;
  close cur;
end;
-- Call the procedure

call totalrevenue;
select AgencyName,sum(fare) from BusInfo natural join BusSchedule group by AgencyName;

```

AGENCYNAME	SUM(FARE)
ABC Bus Company	700
XYZ Bus Company	2000



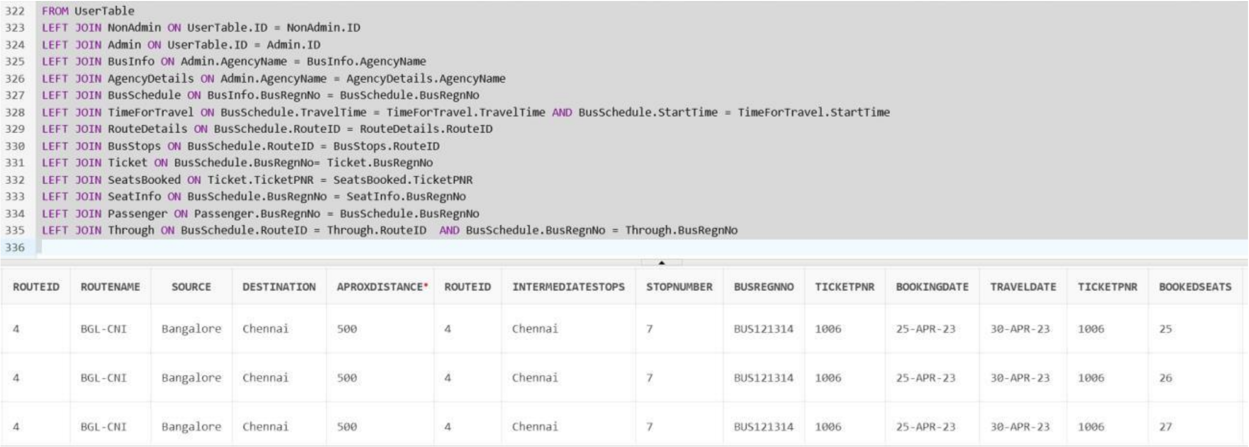
```
    group by AgencyName;
  v_agencyname BusInfo.AgencyName%TYPE;
  v_agencysum BusSchedule.Fare%TYPE;
begin
  open cur;
  loop
    fetch cur into v_agencyname, v_agencysum;
    exit when cur%NOTFOUND;
    dbms_output.put_line(v_agencyname || ': ' || v_agencysum);
  end loop;
  close cur;
end;
-- Call the procedure

call totalrevenue;
```

```
Statement processed.
ABC Bus Company: 700
XYZ Bus Company: 2000
```



```sql
-- Call the procedure

call totalrevenue;
select AgencyName,sum(fare) from BusInfo natural join BusSchedule group by AgencyName;

SELECT *
FROM UserTable
LEFT JOIN NonAdmin ON UserTable.ID = NonAdmin.ID
LEFT JOIN Admin ON UserTable.ID = Admin.ID
LEFT JOIN BusInfo ON Admin.AgencyName = BusInfo.AgencyName
LEFT JOIN AgencyDetails ON Admin.AgencyName = AgencyDetails.AgencyName
LEFT JOIN BusSchedule ON BusInfo.BusRegnNo = BusSchedule.BusRegnNo
LEFT JOIN TimeForTravel ON BusSchedule.TravelTime = TimeForTravel.TravelTime AND BusSchedule.StartTime = TimeForTravel.StartTime
LEFT JOIN RouteDetails ON BusSchedule.RouteID = RouteDetails.RouteID
LEFT JOIN BusStops ON BusSchedule.RouteID = BusStops.RouteID
```

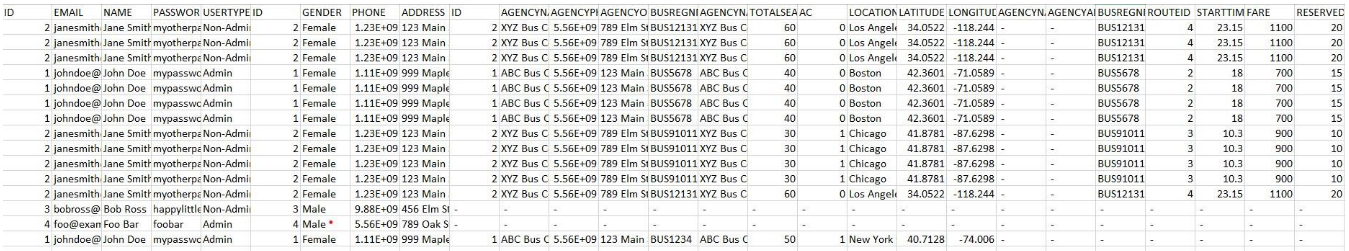
ID	EMAIL	NAME	PASSWORD	USERTYPE	ID	GENDER	PHONE	ADDRESS	ID	AGENCYNAME	AGENCYPHONE	AGENCYOFFICE	BUSREGNNO	AGENCYNAME	TOTALSEATS	AC
1	johndoe@example.com	John Doe	mypassword	Admin	1	Female	1111111111	999 Maple St	1	ABC Bus Company	5555555555	123 Main St	BUS5678	ABC Bus Company	40	0
1	johndoe@example.com	John Doe	mypassword	Admin	1	Female	1111111111	999 Maple St	1	ABC Bus Company	5555555555	123 Main St	BUS5678	ABC Bus Company	40	0



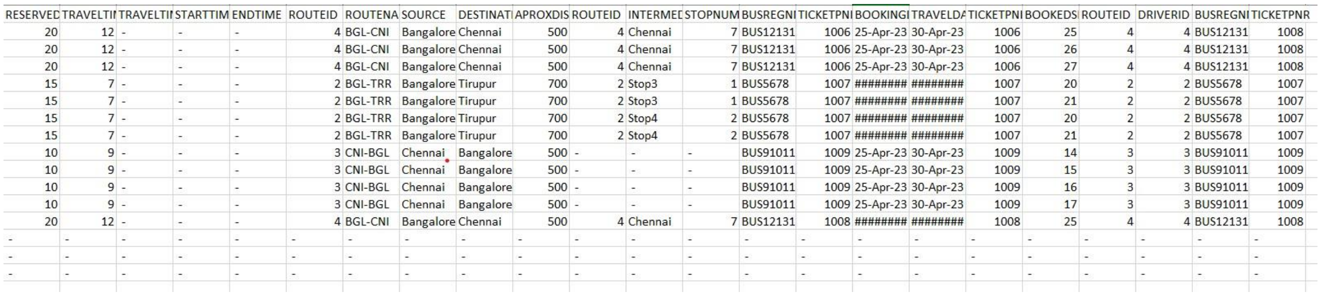
```sql
FROM UserTable
LEFT JOIN NonAdmin ON UserTable.ID = NonAdmin.ID
LEFT JOIN Admin ON UserTable.ID = Admin.ID
LEFT JOIN BusInfo ON Admin.AgencyName = BusInfo.AgencyName
LEFT JOIN AgencyDetails ON Admin.AgencyName = AgencyDetails.AgencyName
LEFT JOIN BusSchedule ON BusInfo.BusRegnNo = BusSchedule.BusRegnNo
LEFT JOIN TimeForTravel ON BusSchedule.TravelTime = TimeForTravel.TravelTime AND BusSchedule.StartTime = TimeForTravel.StartTime
LEFT JOIN RouteDetails ON BusSchedule.RouteID = RouteDetails.RouteID
LEFT JOIN BusStops ON BusSchedule.RouteID = BusStops.RouteID
LEFT JOIN Ticket ON BusSchedule.BusRegnNo= Ticket.BusRegnNo
LEFT JOIN SeatsBooked ON Ticket.TicketPNR = SeatsBooked.TicketPNR
LEFT JOIN SeatInfo ON BusSchedule.BusRegnNo = SeatInfo.BusRegnNo
LEFT JOIN Passenger ON Passenger.BusRegnNo = BusSchedule.BusRegnNo
LEFT JOIN Through ON BusSchedule.RouteID = Through.RouteID  AND BusSchedule.BusRegnNo = Through.BusRegnNo
```

ROUTEID	ROUTENAME	SOURCE	DESTINATION	APROXDISTANCE*	ROUTEID	INTERMEDIATESTOPS	STOPNUMBER	BUSREGNNO	TICKETPNR	BOOKINGDATE	TRAVELDATE	TICKETPNR	BOOKEDSEATS
4	BGL-CNI	Bangalore	Chennai	500	4	Chennai	7	BUS121314	1006	25-APR-23	30-APR-23	1006	25
4	BGL-CNI	Bangalore	Chennai	500	4	Chennai	7	BUS121314	1006	25-APR-23	30-APR-23	1006	26
4	BGL-CNI	Bangalore	Chennai	500	4	Chennai	7	BUS121314	1006	25-APR-23	30-APR-23	1006	27

Output in Excel .csv File



ID	EMAIL	NAME	PASSWOR	USERTYPE	ID	GENDER	PHONE	ADDRESS	ID	AGENCYN	AGENCYPI	AGENCYO	BUSREGNI	AGENCYN	TOTALSEA	AC	LOCATION	LATITUDE	LONGITUD	AGENCYN	AGENCYAI	BUSREGNI	ROUTEID	STARTTIM	FARE	RESERVED
2	janesmith	Jane Smith	myotherpa	Non-Admi	2	Female	1.23E+09	123 Main	2	XYZ Bus C	5.56E+09	789 Elm St	BUS12131	XYZ Bus C	60	0	Los Angele	34.0522	-118.244	-	-	BUS12131	4	23.15	1100	20
2	janesmith	Jane Smith	myotherpa	Non-Admi	2	Female	1.23E+09	123 Main	2	XYZ Bus C	5.56E+09	789 Elm St	BUS12131	XYZ Bus C	60	0	Los Angele	34.0522	-118.244	-	-	BUS12131	4	23.15	1100	20
2	janesmith	Jane Smith	myotherpa	Non-Admi	2	Female	1.23E+09	123 Main	2	XYZ Bus C	5.56E+09	789 Elm St	BUS12131	XYZ Bus C	60	0	Los Angele	34.0522	-118.244	-	-	BUS12131	4	23.15	1100	20
1	johndoe@	John Doe	mypasswc	Admin	1	Female	1.11E+09	999 Maple	1	ABC Bus C	5.56E+09	123 Main	BUS5678	ABC Bus C	40	0	Boston	42.3601	-71.0589	-	-	BUS5678	2	18	700	15
1	johndoe@	John Doe	mypasswc	Admin	1	Female	1.11E+09	999 Maple	1	ABC Bus C	5.56E+09	123 Main	BUS5678	ABC Bus C	40	0	Boston	42.3601	-71.0589	-	-	BUS5678	2	18	700	15
1	johndoe@	John Doe	mypasswc	Admin	1	Female	1.11E+09	999 Maple	1	ABC Bus C	5.56E+09	123 Main	BUS5678	ABC Bus C	40	0	Boston	42.3601	-71.0589	-	-	BUS5678	2	18	700	15
1	johndoe@	John Doe	mypasswc	Admin	1	Female	1.11E+09	999 Maple	1	ABC Bus C	5.56E+09	123 Main	BUS5678	ABC Bus C	40	0	Boston	42.3601	-71.0589	-	-	BUS5678	2	18	700	15
2	janesmith	Jane Smith	myotherpa	Non-Admi	2	Female	1.23E+09	123 Main	2	XYZ Bus C	5.56E+09	789 Elm St	BUS91011	XYZ Bus C	30	1	Chicago	41.8781	-87.6298	-	-	BUS91011	3	10.3	900	10
2	janesmith	Jane Smith	myotherpa	Non-Admi	2	Female	1.23E+09	123 Main	2	XYZ Bus C	5.56E+09	789 Elm St	BUS91011	XYZ Bus C	30	1	Chicago	41.8781	-87.6298	-	-	BUS91011	3	10.3	900	10
2	janesmith	Jane Smith	myotherpa	Non-Admi	2	Female	1.23E+09	123 Main	2	XYZ Bus C	5.56E+09	789 Elm St	BUS91011	XYZ Bus C	30	1	Chicago	41.8781	-87.6298	-	-	BUS91011	3	10.3	900	10
2	janesmith	Jane Smith	myotherpa	Non-Admi	2	Female	1.23E+09	123 Main	2	XYZ Bus C	5.56E+09	789 Elm St	BUS91011	XYZ Bus C	30	1	Chicago	41.8781	-87.6298	-	-	BUS91011	3	10.3	900	10
2	janesmith	Jane Smith	myotherpa	Non-Admi	2	Female	1.23E+09	123 Main	2	XYZ Bus C	5.56E+09	789 Elm St	BUS12131	XYZ Bus C	60	0	Los Angele	34.0522	-118.244	-	-	BUS12131	4	23.15	1100	20
3	bobross@	Bob Ross	happylittle	Non-Admi	3	Male	9.88E+09	456 Elm St	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
4	foo@exam	Foo Bar	foobar	Admin	4	Male	5.56E+09	789 Oak S	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
1	johndoe@	John Doe	mypasswc	Admin	1	Female	1.11E+09	999 Maple	1	ABC Bus C	5.56E+09	123 Main	BUS1234	ABC Bus C	50	1	New York	40.7128	-74.006	-	-	-	-	-	-	-



RESERVED	TRAVELTI	TRAVELTI	STARTTIM	ENDTIME	ROUTEID	ROUTENA	SOURCE	DESTINATI	APROXDIS	ROUTEID	INTERMED	STOPNUM	BUSREGNI	TICKETPNI	BOOKINGI	TRAVELDA	TICKETPNI	BOOKEDS	ROUTEID	DRIVERID	BUSREGNI	TICKETPNR
20	12	-	-	-	4	BGL-CNI	Bangalore	Chennai	500	4	Chennai	7	BUS12131	1006	25-Apr-23	30-Apr-23	1006	25	4	4	BUS12131	1008
20	12	-	-	-	4	BGL-CNI	Bangalore	Chennai	500	4	Chennai	7	BUS12131	1006	25-Apr-23	30-Apr-23	1006	26	4	4	BUS12131	1008
20	12	-	-	-	4	BGL-CNI	Bangalore	Chennai	500	4	Chennai	7	BUS12131	1006	25-Apr-23	30-Apr-23	1006	27	4	4	BUS12131	1008
15	7	-	-	-	2	BGL-TRR	Bangalore	Tirupur	700	2	Stop3	1	BUS5678	1007	########	########	1007	20	2	2	BUS5678	1007
15	7	-	-	-	2	BGL-TRR	Bangalore	Tirupur	700	2	Stop3	1	BUS5678	1007	########	########	1007	21	2	2	BUS5678	1007
15	7	-	-	-	2	BGL-TRR	Bangalore	Tirupur	700	2	Stop4	2	BUS5678	1007	########	########	1007	20	2	2	BUS5678	1007
15	7	-	-	-	2	BGL-TRR	Bangalore	Tirupur	700	2	Stop4	2	BUS5678	1007	########	########	1007	21	2	2	BUS5678	1007
10	9	-	-	-	3	CNI-BGL	Chennai	Bangalore	500	-	-	-	BUS91011	1009	25-Apr-23	30-Apr-23	1009	14	3	3	BUS91011	1009
10	9	-	-	-	3	CNI-BGL	Chennai	Bangalore	500	-	-	-	BUS91011	1009	25-Apr-23	30-Apr-23	1009	15	3	3	BUS91011	1009
10	9	-	-	-	3	CNI-BGL	Chennai	Bangalore	500	-	-	-	BUS91011	1009	25-Apr-23	30-Apr-23	1009	16	3	3	BUS91011	1009
10	9	-	-	-	3	CNI-BGL	Chennai	Bangalore	500	-	-	-	BUS91011	1009	25-Apr-23	30-Apr-23	1009	17	3	3	BUS91011	1009
20	12	-	-	-	4	BGL-CNI	Bangalore	Chennai	500	4	Chennai	7	BUS12131	1008	########	########	1008	25	4	4	BUS12131	1008
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Conclusion

In conclusion, the Bus Transport Enquiry System is an efficient and effective way to manage and monitor the bus transportation system. It provides information to the passengers about the arrival and departure of buses, the estimated time of arrival at their destination, and other relevant information. It also allows bus companies to manage their fleet efficiently, keep track of their drivers' performance, and optimize their routes to reduce travel time and fuel consumption. The system also facilitates communication between passengers, drivers, and bus companies, leading to better customer satisfaction.

The Bus Transport Enquiry System is a modern solution to the challenges facing the bus transportation industry and its implementation can revolutionize the way people travel andimprove the quality of life for millions of passengers.

References

[1] https://livesql.oracle.com/

[2] https://www.oracletutorial.com/oracle-basics/

[3] https://www.tutorialspoint.com/plsql/plsql_triggers.htm

[4] https://www.tutorialspoint.com/plsql/plsql_procedures.htm

[5] https://www.tutorialspoint.com/plsql/plsql_cursors.htm