

Port Washington Line of LIRR Database Management System



Bearcat&Company

Group 12

Maria Bonde Sorensen - MARIA.SORENSEN@baruchmail.cuny.edu

Kejin Liu - KEJIN.LIU@baruchmail.cuny.edu

Tanesha Campbell - TANESHA.CAMPBELL@baruchmail.cuny.edu

Kenneth Annunziata - KENNETH.ANNUNZIATA@baruchmail.cuny.edu

Jeremy Perez - JEREMY.PEREZ1@baruchmail.cuny.edu

Table of Contents

<i>Executive summary.....</i>	<i>3</i>
<i>Business Scenario.....</i>	<i>3</i>
<i>ER Model using UML Notation.....</i>	<i>4</i>
Relationship sentences	4
<i>Conversion to Relational Model.....</i>	<i>5</i>
<i>Normalization.....</i>	<i>5</i>
<i>Creating the Database Schema with Structured Query Language</i>	<i>8</i>
Create Table.....	8
Relationship View	14
Adding Data to the Tables using SQL INSERT Statements	15
Queries - 5 Scenarios	17
<i>Database Application.....</i>	<i>20</i>
Navigation form	20
Reports	27
<i>Conclusion.....</i>	<i>30</i>

Executive summary

To give the MTA a better overview of their train line and the efficiency Bearcat&Company consulting company has been hired. This report will focus on one particular trainline – the Port Washington line. It will outline the work that has been done to create a database. This includes all stages from the drawing of the ER diagram, relationship sentences, the conversion to RDM, normalization process till creating the database in Access. The report will also show some navigation forms and in the end it will be concluded by reflecting on the work, and if database successfully solved the needs of the MTA.

Business Scenario

Bearcat&Company consulting company has been hired by the MTA to make the LIRR line schedule more reasonable during the pandemic, so the MTA can meet the needs of passengers and not waste too much money on excessive operations.

Following a track line from the LIRR, a database can be used to track the trains, schedules and routes. It will provide an overview on the availability of trains, their capacity and room for passengers, and see what trains are currently down. The database will also store passenger information and what type of ticket they have bought.

For this project we will start out following one train line (Port Washington line). We will need information about the time schedule. We also need to collect information about employees, how many, and their availability to make sure there are available train drivers. Additionally, we need information about the passengers, what they are booking and on which date.

Roles:

Maria Bonde Sorensen - Team leader

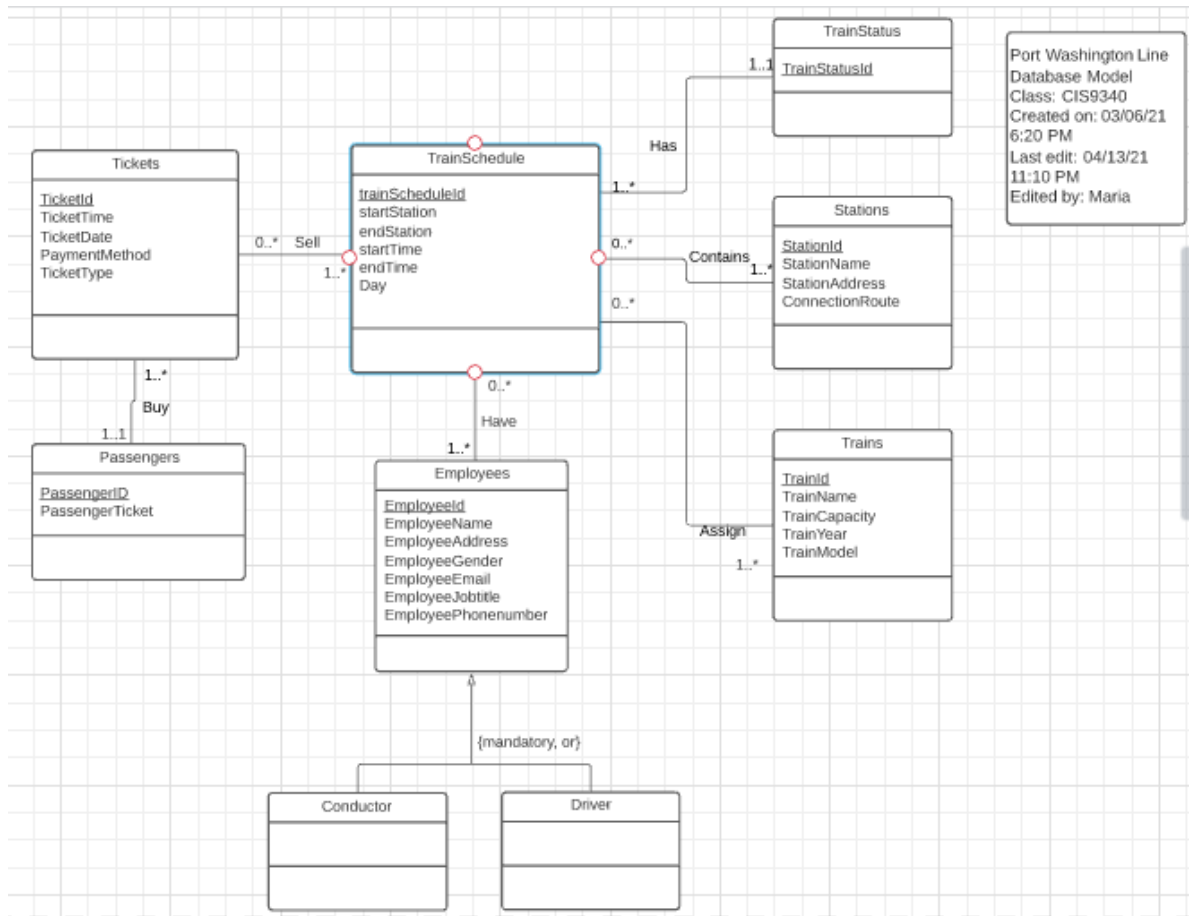
Tanesha Campbell - System analyst

Kenneth Annunziata - Database implementor

Kejin Liu - Application developer

Jeremy Perez - Document writer

ER Model using UML Notation



Relationship sentences

Passengers <-> Tickets = A passenger *must* buy one or more tickets; A ticket *must* be sold to one passenger.

Employees <-> TrainSchedule = An employee *may* have many trainschedules; A trainschedule *must* have one or many employees.

Tickets <-> TrainSchedule = A ticket *must* be sold for one or many transchedules; A trainschedule *may* sell many tickets.

TrainSchedule <-> Trains = A train schedule *must* be assigned to one or more trains; A train *may be* assigned zero or many train schedules.

TrainSchedule <-> Stations = A train schedule *must* contain one or more stations; A station *may* contain zero or more train schedules.

TrainSchedule <-> TrainStatus = A trainschedule *must* have a trainstatus; A trainStatus must have one or *many* TrainSchedules.

Conversion to Relational Model

Equivalent Relational Model based of ER-Model:

Tickets (TicketID, TicketTime, TicketDate, PaymentMethod, TicketType, PassengerID(fk))

Passengers (PassengerID, PassengerTicket)

TrainSchedule (trainScheduleID, startStation, endStation, startTime, endTime, Day, TrainStatusID(fk))

Employees (EmployeeID, EmployeeName, EmployeeAddress, EmployeeGender, EmployeeEmail, EmployeeJobTitle, EmployeePhonenumber)

Conductor (EmployeeID, EmployeeName, EmployeeAddress, EmployeeGender, EmployeeEmail, EmployeeJobTitle, EmployeePhonenumber)

Driver (EmployeeID, EmployeeName, EmployeeAddress, EmployeeGender, EmployeeEmail, EmployeeJobTitle, EmployeePhonenumber)

TrainStatus (TrainStatusID)

Stations(StationID, StationName, StationAddress, ConnectionRoute)

Trains(TrainID, TrainName, TrainCapacity, TrainYear, TrainModel)

Normalization

1st Normal Form: Single Valued Attributes, Attribute Domain not Changed, Unique name for Attribute/Columns

Tickets(TicketID,TicketTime,TicketDate,PaymentMethod,TicketType,TrainID,trainScheduleID, PassengerID (fk))

Passengers(PassengerID,PassengerName,Contact Number)

TrainSchedule(trainScheduleID,startStation,endStation,startTime,endTime,Day, TrainStatusID (fk))

Employees(EmployeeID,EmployeeName,EmployeeAddress,EmployeeGender,EmployeeJobTitle ,EmployeePhonenumber,EmployeeType)

EmployeeTrainSchedule(EmployeeID,trainScheduleID)

- Since an employee may have several train schedules, just adding the attribute trainscheduleID in the employee table makes it a multi-valued attribute. To remove this and to satisfy the prerequisite of the 1st Normal Form to have all the columns single valued, then a new table to record all the trainschedules of an employee needs to be created. Since an employee can either be a driver or a conductor, then an employeeType column is added instead of creating a duplicate table to record whether the employee is a conductor or a driver. Since they belong to a single company, then there is no need for a separate conductorID or DriverID.

Trains(TrainID,TrainName,TrainCapacity,TrainYear,TrainModel)

TrainScheduleTrain(TrainScheduleID,TrainID)

TrainScheduleTicket(TrainScheduleID,TicketID)

- Since tickets must be sold to one or more ticket schedules and train schedules must be assigned to one or more trains, there is a need to add additional tables to handle these, because if these are added to their mother table, then, there is a possibility of having multi-valued attributes.

Stations(StationID,StationName,StationAddress,ConnectionRout)

TrainScheduleStations(TrainScheduleID,StationID)

- ➔ A train schedule can have several stations, thus to solve this, a new table to record a specific train schedule, the station and the train assigned to that schedule and station needs to be created.

TrainStatus(TrainStatusID,TrainStatus)

2nd Normal Form: Be in first Normal Form and No Partial Dependency (A primary that is able to identify each row must be present)

Tickets (**TicketID**, TicketTime, TicketDate, PaymentMethod, TicketType, TrainID, trainScheduleID)

Passengers (**PassengerID**, PassengerName, Contact Number)

PassengerTicket (**PassengerID**, **TicketID**)

TrainSchedule (**trainScheduleID**, startStation, endStation, startTime, endTime, Day)

Employees (**EmployeeID**, EmployeeName, EmployeeAddress, EmployeeGender, EmployeeGender, EmployeeJobTitle, EmployeePhonenumber, EmployeeType)

EmployeeTrainSchedule (EmployeeID, **trainScheduleID**)

Trains(**TrainID**, TrainName, TrainCapacity, TrainYear,TrainModel)

TrainScheduleTrain (**TrainScheduleID**, **TrainID**)

Stations (**StationID**, StationName, StationAddress, ConnectionRoute)

TrainTrainScheduleStations (TrainScheduleID, **TrainID**, **StationID**)

TrainStatus (**TrainStatusID**, TrainStatus)

TrainScheduleTrainStatus (**TrainScheduleID**, **TrainStatusID**)

- ❖ All tables have identified primary keys. Some of the tables have composite primary keys identified. Composite primary keys are composed of two or more primary keys to identify the table.

3rd Normal Form: Must be in the Second Normal Form and does not have Transitive Dependency

Tickets(**TicketID**, TicketTime, TicketDate, PaymentMethod, TicketType, TrainID,trainScheduleID)

Passengers (**PassengerID**, PassengerName, ContactNumber)

PassengerTicket(**PassengerID**,**TicketID**)

TrainSchedule(**trainScheduleID**, startStation, endStation, startTime, endTime, Day)

Employees (**EmployeeID**, EmployeeName, EmployeeAddress, EmployeeGender,EmployeeEmail, EmployeeJobTitle, EmployeePhonenumber, EmployeeType)

EmployeeTrainSchedule(**EmployeeID**,trainScheduleID)

Trains(**TrainID**, TrainName, TrainCapacity, TrainYear, TrainModel)

TrainScheduleTrain (**TrainScheduleID**, **TrainID**)

Stations(**StationID**, StationName, StationAddress, ConnectionRoute)

TrainTrainScheduleStations (TrainScheduleID, **TrainID**,**StationID**)

TrainStatus(**TrainStatusID**, TrainStatus)

TrainScheduleTrainStatus (**TrainScheduleID**, **TrainStatusID**)

- ❖ Looking at the existing tables, there are no transitive dependencies. There are no non-prime attributes that are dependent on other non-prime attributes in each of the tables.
- ❖ Since all the conditions of the 3rd Normal Form have been met, the database is good to go. There is no need to move to the BCNF, 4th Normal Form and the 5th Normal Form.

Creating the Database Schema with Structured Query Language

Create Table

Employees:

```
CREATE TABLE Employees (  
EmployeeID NUMBER NOT NULL,  
EmployeeName VARCHAR (255) NOT NULL,  
EmployeeAddress VARCHAR (255),  
EmployeeGender VARCHAR (1),  
EmployeeEmail VARCHAR (255),  
EmployeeJobTitle VARCHAR (255),  
EmployeePhoneNumber VARCHAR (255)  
);
```

EmployeeTrainSchedule:

```
CREATE TABLE EmployeeTrainSchedule(  
EmployeeID NUMBER NOT NULL,  
TrainScheduleID NUMBER NOT NULL  
);
```

Passengers:

```
CREATE TABLE Passengers(  
PassengerID NUMBER NOT NULL,  
PassengerName VARCHAR (50),
```



```
ContactNumber VARCHAR (50)
```

```
);
```

PassengersTicket:

```
CREATE TABLE PassengerTicket(
```

```
PassengerID NUMBER NOT NULL,
```

```
TicketID NUMBER NOT NULL
```

```
);
```

Stations:

```
CREATE TABLE Stations (
```

```
StationID NUMBER NOT NULL,
```

```
StationName VARCHAR (255),
```

```
StationAddress VARCHAR (255),
```

```
ConnectionRoute VARCHAR (255)
```

```
);
```

Tickets:

```
CREATE TABLE Tickets(
```

```
TicketID NUMBER NOT NULL,
```

```
TicketTime TIME,
```

```
TicketDate DATE,
```

```
PaymentMethod VARCHAR (25),
```

```
TicketType VARCHAR (25),
```

```
TrainScheduleID NUMBER,
```

```
TrainID NUMBER
```

```
);
```

Trains:

```
CREATE TABLE Trains (
```

```
TrainID NUMBER NOT NULL,
```

```
TrainName VARCHAR (50),
```

```
TrainCapacity NUMBER,
```

```
TrainYear NUMBER,
```

```
TrainModel VARCHAR (50)
```

```
);
```

TrainSchedule:

```
CREATE TABLE TrainSchedule(
```

```
TrainScheduleID NUMBER NOT NULL,
```

```
startStation VARCHAR (255),
```

```
endStation VARCHAR (255),
```

```
startTime TIME,
```

```
endTime TIME,
```

```
Day VARCHAR (20)
```

```
);
```

TrainScheduleStations:

```
CREATE TABLE TrainScheduleStations (
```

```
TrainScheduleID NUMBER,
```

```
TrainID NUMBER NOT NULL,  
StationID NUMBER NOT NULL  
);
```

TrainScheduleTrain:

```
CREATE TABLE TrainScheduleTrain (  
TrainScheduleID NUMBER NOT NULL,  
TrainID NUMBER NOT NULL  
);
```

TrainScheduleTrainStatus:

```
CREATE TABLE TrainScheduleTrainStatus (  
TrainScheduleID NUMBER NOT NULL,  
TrainStatusID NUMBER NOT NULL  
);
```

TrainStatus:

```
CREATE TABLE TrainStatus (  
TrainStatusID NUMBER NOT NULL,  
TrainStatus VARCHAR (4)  
);
```

Add primary- and foreign keys

Employees

```
ALTER TABLE Employees  
ADD CONSTRAINT pk_EmployeeID PRIMARY KEY (EmployeeID);
```

EmployeeTrainSchedule

```
ALTER TABLE EmployeeTrainSchedule
ADD CONSTRAINT pk_etID PRIMARY KEY (EmployeeID,
TrainScheduleID);
```

```
ALTER TABLE EmployeeTrainSchedule
ADD CONSTRAINT fk_EmployeeID FOREIGN KEY (EmployeeID) REFERENCES
Employees (EmployeeID);
```

```
ALTER TABLE EmployeeTrainSchedule
ADD CONSTRAINT fk_TrainScheduleID3 FOREIGN KEY (TrainScheduleID)
REFERENCES TrainSchedule (TrainScheduleID);
```

Passengers

```
ALTER TABLE Passengers
ADD CONSTRAINT pk_PassengerID PRIMARY KEY (PassengerID);
```

PassengerTicket

```
ALTER TABLE PassengerTicket
ADD CONSTRAINT pk_PTID PRIMARY KEY (PassengerID, TicketID);
```

```
ALTER TABLE PassengerTicket
ADD CONSTRAINT fk_PassengerID FOREIGN KEY (PassengerID)
REFERENCES Passengers (PassengerID);
```

```
ALTER TABLE PassengerTicket
ADD CONSTRAINT fk_TicketID2 FOREIGN KEY (TicketID) REFERENCES
Tickets (TicketID);
```

Stations

```
ALTER TABLE Stations
ADD CONSTRAINT pk_StationID PRIMARY KEY (StationID);
```

Tickets

```
ALTER TABLE Tickets
ADD CONSTRAINT pk_TicketID PRIMARY KEY (TicketID);
```

```
ALTER TABLE Tickets
ADD CONSTRAINT fk_TrainScheduleID2 FOREIGN KEY (TrainScheduleID)
REFERENCES TrainSchedule (TrainScheduleID);
```

```
ALTER TABLE Tickets
ADD CONSTRAINT fk_TrainID2 FOREIGN KEY (TrainID) REFERENCES
Trains (TrainID);
```

Trains

```
ALTER TABLE Trains
ADD CONSTRAINT pl_TrainID PRIMARY KEY (TrainID);
```

TrainSchedule

```
ALTER TABLE TrainSchedule
ADD CONSTRAINT pk_TrainScheduleID PRIMARY KEY (TrainScheduleID);
```

TrainScheduleStations

```
ALTER TABLE TrainScheduleStations
ADD CONSTRAINT pk_ttsID PRIMARY KEY (StationID, TrainID);
```

```
ALTER TABLE TrainScheduleStations
ADD CONSTRAINT fk_TrainScheduleID4 FOREIGN KEY (TrainScheduleID)
REFERENCES TrainSchedule (TrainScheduleID);
```

```
ALTER TABLE TrainScheduleStations
ADD CONSTRAINT fk_TrainID3 FOREIGN KEY (TrainID) REFERENCES
Trains (TrainID);
```

```
ALTER TABLE TrainScheduleStations
ADD CONSTRAINT fk_StationID FOREIGN KEY (StationID) REFERENCES
Stations (StationID);
```

TrainScheduleTrain

```
ALTER TABLE TrainScheduleTrain
ADD CONSTRAINT pk_TrainIDTrainScheduleID PRIMARY KEY
(TrainScheduleID, TrainID);
```

```
ALTER TABLE TrainScheduleTrain
ADD CONSTRAINT fk_TrainID FOREIGN KEY (TrainID) REFERENCES
Trains (TrainID);
```

```
ALTER TABLE TrainScheduleTrain
ADD CONSTRAINT fk_TrainScheduleID FOREIGN KEY (TrainScheduleID)
REFERENCES TrainSchedule (TrainScheduleID);
```

TrainScheduleTrainStatus

```
ALTER TABLE TrainScheduleTrainStatus
ADD CONSTRAINT pk_ttID PRIMARY KEY (TrainScheduleID,
TrainStatusID);
```

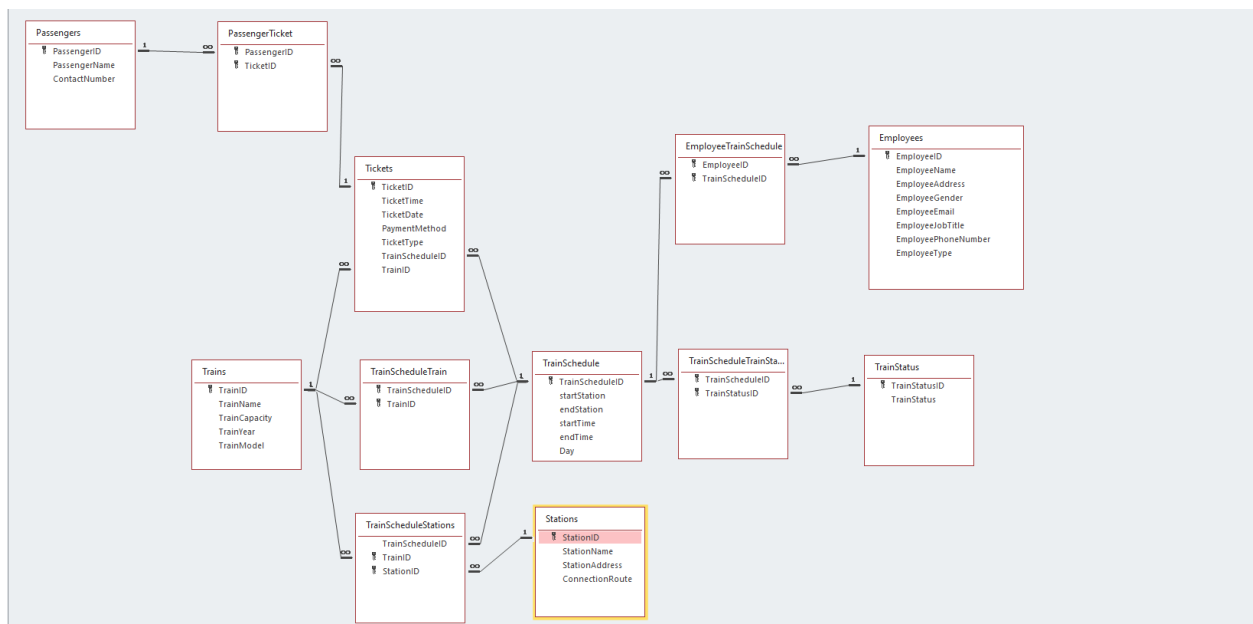
```
ALTER TABLE TrainScheduleTrainStatus
ADD CONSTRAINT fk_TrainScheduleID5 FOREIGN KEY (TrainScheduleID)
REFERENCES TrainSchedule (TrainScheduleID);
```

```
ALTER TABLE TrainScheduleTrainStatus
ADD CONSTRAINT fk_TrainScheduleID5 FOREIGN KEY (TrainScheduleID)
REFERENCES TrainSchedule (TrainScheduleID);
```

```
ALTER TABLE TrainScheduleTrainStatus
ADD CONSTRAINT fk_TrianStatusID FOREIGN KEY (TrainStatusID)
REFERENCES TrainStatus (TrainStatusID);
```

```
ALTER TABLE TrainStatus
ADD CONSTRAINT pk_TrainStatusID PRIMARY KEY (TrainStatusID);
```

Relationship View



Adding Data to the Tables using SQL INSERT Statements

Passengers:

```
INSERT INTO Passengers
VALUES (419, 'John Smith', '516-898-7486');
```

```
INSERT INTO Passengers
VALUES (878, 'Billy Rowe', '646-552-8763');
```

Tickets:

```
INSERT INTO Tickets
VALUES (17, #10:00:00 AM#, #4/21/2021#, "Cash", "E-Ticket", 1,
3705);
```

```
INSERT INTO Tickets
VALUES (8, #10:30 AM#, #4/20/2021#, "Credit", "Ticket", 6,
2351);
```

Trains:

```
INSERT INTO Trains
VALUES (4694, 'Kate', 1000, 1999, 'x10');
```

```
INSERT INTO Trains
VALUES (2351, 'Fredrick', 1000, 1997, 'x10');
```

TrainSchedule:

```
INSERT INTO TrainSchedule
VALUES (1, 'Port Washington', 'Plandome', #10:00 AM#, #10:15am#,
'Monday');
```

```
INSERT INTO TrainSchedule
VALUES (2, 'Port Washington', 'Manhasset', #8:30 AM#, #9:00am#,
'Monday');
```

Stations:

```
INSERT INTO Stations
```

```
VALUES (10, 'Port Washington', 'Haven Ave, Port Washington, NY
11050', 'Plandome');
```

```
INSERT INTO Stations
VALUES (20, 'Plandome', 'Plandome, NY 11030', 'Manhasset');
```

TrainStatus:

```
INSERT INTO TrainStatus
VALUES (0, 'Good Service');
```

```
INSERT INTO TrainStatus
VALUES (1, 'Down');
```

Employees:

```
INSERT INTO Employees
VALUES (4106, 'Melissa Turner', '6 Grand Lane Brooklyn, NY
11238', 'Female', 'm.turner@randatmail.com', 'Conductor', '762-
7448-72');
```

```
INSERT INTO Employees
VALUES (3729, 'Clark Armstrong', '488 Lincoln Drive Port
Washington, NY 11053', 'Male', 'c.armstrong@randatmail.com',
'Driver', '055-7034-48');
```

PassengerTicket:

```
INSERT INTO PassengerTicket
VALUES (993, 17);
```

```
INSERT INTO PassengerTicket
VALUES (878, 22);
```

TrainScheduleTrain:

```
INSERT INTO TrainScheduleTrain
VALUES (10, 4694);
```

```
INSERT INTO TrainScheduleTrain
VALUES (7, 3098);
```

TrainScheduleStations:

```
INSERT INTO TrainScheduleStations
```



```
VALUES (10, 4694, 40);
```

```
INSERT INTO TrainScheduleStations  
VALUES (7, 3098, 140);
```

TrainScheduleTrainStatus:

```
INSERT INTO TrainScheduleTrainStatus  
VALUES (1, 0);
```

```
INSERT INTO TrainScheduleTrainStatus  
VALUES (2, 0);
```

EmployeeTrainSchedule:

```
INSERT INTO EmployeeTrainSchedule  
VALUES (3955, 14);
```

```
INSERT INTO EmployeeTrainSchedule  
VALUES (3715, 6);
```

Queries - 5 Scenarios

1. This query shows which train schedules are currently down.

```
SELECT t.*, TrainStatus  
FROM (TrainSchedule t INNER JOIN TrainScheduleTrainStatus tts ON  
t.TrainScheduleID = tts.TrainScheduleID) INNER JOIN TrainStatus  
ts ON tts.TrainStatusID = ts.TrainStatusID  
WHERE tts.TrainStatusID = 1;
```

TrainScheduleID	startStation	endStation	startTime	endTime	Day	TrainStatus
6	Port Washington	Penn Station	10:30:00 AM	12:00:00 PM	Everyday	Down
11	Penn Station	Port Washington	1:00:00 PM	3:30:00 PM	Everyday	Down
16	Penn Station	Port Washington	6:00:00 PM	8:30:00 PM	Everyday	Down
*						

2. This query shows employees currently scheduled for a train time and the details of the train schedule they are scheduled to.

```

SELECT EmployeeName, ts.*
FROM (Employees e INNER JOIN EmployeeTrainSchedule ets ON
e.EmployeeID = ets.EmployeeID) INNER JOIN TrainSchedule ts ON
ets.TrainScheduleID = ts.TrainScheduleID

```

EmployeeName	TrainScheduleID	startStation	endStation	startTime	endTime	Day
Alexia Barrett	7	Port Washington	Great Neck	11:00:00 AM	11:45:00 AM	Tuesday
Chester Brooks	12	Penn Station	Port Washington	2:00:00 PM	4:30:00 PM	Everyday
Clark Armstrong	14	Penn Station	Port Washington	4:00:00 PM	6:30:00 PM	Everyday
David Chapman	5	Port Washington	Penn Station	9:30:00 AM	11:00:00 AM	Everyday
Eleanor Alexander	11	Penn Station	Port Washington	1:00:00 PM	3:30:00 PM	Everyday
Eleanor Alexander	10	Great Neck	Bayside	11:00:00 AM	11:45:00 AM	Friday
Evelyn Wright	16	Penn Station	Port Washington	6:00:00 PM	8:30:00 PM	Everyday
Evelyn Wright	2	Port Washington	Manhasset	8:30:00 AM	9:00:00 AM	Monday
Frederick Mitchell	16	Penn Station	Port Washington	6:00:00 PM	8:30:00 PM	Everyday
Frederick Mitchell	6	Port Washington	Penn Station	10:30:00 AM	12:00:00 PM	Everyday
John Cameron	8	Great Neck	Little Neck	10:00:00 AM	10:15:00 AM	Wednesday
Kelvin Dixon	14	Penn Station	Port Washington	4:00:00 PM	6:30:00 PM	Everyday
Kirsten Harris	13	Penn Station	Port Washington	3:00:00 PM	5:30:00 PM	Everyday
Lucas Nelson	9	Great Neck	Douglaston	8:30:00 AM	9:00:00 AM	Thursday
Lyndon Kelly	16	Penn Station	Port Washington	6:00:00 PM	8:30:00 PM	Everyday
Melissa Turner	2	Port Washington	Manhasset	8:30:00 AM	9:00:00 AM	Monday
Melissa Turner	6	Port Washington	Penn Station	10:30:00 AM	12:00:00 PM	Everyday
Ned Wilson	10	Great Neck	Bayside	11:00:00 AM	11:45:00 AM	Friday
Ned Wilson	14	Penn Station	Port Washington	4:00:00 PM	6:30:00 PM	Everyday
Ned Wilson	2	Port Washington	Manhasset	8:30:00 AM	9:00:00 AM	Monday
Robert Adams	5	Port Washington	Penn Station	9:30:00 AM	11:00:00 AM	Everyday
Robert Adams	16	Penn Station	Port Washington	6:00:00 PM	8:30:00 PM	Everyday
Ryan Cunningham	8	Great Neck	Little Neck	10:00:00 AM	10:15:00 AM	Wednesday
Ted Williams	7	Port Washington	Great Neck	11:00:00 AM	11:45:00 AM	Tuesday
Tyler Sullivan	3	Port Washington	Penn Station	7:30:00 AM	9:00:00 AM	Everyday
Victor Phillips	14	Penn Station	Port Washington	4:00:00 PM	6:30:00 PM	Everyday
Vivian Brooks	6	Port Washington	Penn Station	10:30:00 AM	12:00:00 PM	Everyday
Vivian Brooks	9	Great Neck	Douglaston	8:30:00 AM	9:00:00 AM	Thursday
*						

- This query shows all trains scheduled to operate everyday before 12pm and then orders the query by start time.

```

SELECT *
FROM TrainSchedule
WHERE Day = 'Everyday' AND startTime < #12:00 PM#
ORDER BY startTime;

```

TrainSchedu	startStation	endStation	startTime	endTime	Day
3	Port Washington	Penn Station	7:30:00 AM	9:00:00 AM	Everyday
4	Port Washington	Penn Station	8:30:00 AM	10:00:00 AM	Everyday
5	Port Washington	Penn Station	9:30:00 AM	11:00:00 AM	Everyday
6	Port Washington	Penn Station	10:30:00 AM	12:00:00 PM	Everyday
*					

4. This query shows all names for passengers who bought an e-ticket for a route that starts at Penn Station.

```
SELECT PassengerName, TicketType, startStation
FROM ((Passengers p INNER JOIN PassengerTicket pt ON
p.PassengerID = pt.PassengerID) INNER JOIN Tickets t ON
pt.TicketID = t.TicketID) INNER JOIN TrainSchedule ts ON
t.TrainScheduleID =ts.TrainScheduleID
WHERE t.TicketType = 'E-Ticket' AND ts.startStation = 'Penn
Station'
```

PassengerName ▾	TicketType ▾	startStation ▾
Rachel Bondy	E-Ticket	Penn Station
Steve Jobs	E-Ticket	Penn Station
Lillian Mcquire	E-Ticket	Penn Station
Steve Jobs	E-Ticket	Penn Station
Michael Bean	E-Ticket	Penn Station
*		

5. This query shows all trains that have a name starting with the letter 'A' or 'K' and also has a capacity of 2500 or more.

```
SELECT TrainID, TrainName, TrainCapacity
FROM Trains
WHERE (TrainName LIKE 'A*' OR TrainName LIKE 'K*') AND
TrainCapacity >= 2500
```

TrainID ▾	TrainName ▾	TrainCapacit ▾
3098	Kelvin	2500
1159	Aldus	2750
4583	Alford	5000
*		

Database Application

Navigation form

The screenshot displays the Microsoft Access interface. On the left is the 'Navigation pane' with a search bar and categories: Tables, Queries, and Forms. Under 'Forms', the 'Employees' form is selected and highlighted in red. The main window shows the 'Employees' form with a title bar and a search dropdown set to 'Amber Anderson'. The form contains several fields with their values:

Field Name	Value
EmployeeID	2958
EmployeeName	Amber Anderson
EmployeeAddress	8858 Swanson St. Rego Park, NY 11374
EmployeeGender	F
EmployeeEmail	a.anderson@randatmail.com
EmployeeJobTitle	Conductor
EmployeePhoneNumber	871-5188-67

This form allows you to find the ID, name, address, gender, email, job title, and phone number for a specific employee by entering their name.

All Access Objects

Search...

Tables

- Employees
- EmployeeTrainSchedule
- Passengers
- PassengerTicket
- Stations
- Tickets
- Trains
- TrainSchedule
- TrainScheduleStations
- TrainScheduleTrain
- TrainScheduleTrainStatus
- TrainStatus

Queries

Forms

- Employees
- EmployeeTrainSchedule
- Passengers
- PassengerTicket

EmployeeTrainSchedule

EmployeeID: 2644

TrainScheduleID: 5

This form allows you to see which train schedule a specific employee is assigned to by entering their employee ID.

All Access Objects

Search...

Tables

- Employees
- EmployeeTrainSchedule
- Passengers
- PassengerTicket
- Stations
- Tickets
- Trains
- TrainSchedule
- TrainScheduleStations
- TrainScheduleTrain
- TrainScheduleTrainStatus
- TrainStatus

Queries

Forms

- Employees
- EmployeeTrainSchedule
- Passengers
- PassengerTicket
- Stations
- Tickets

PassengerTicket

PassengerID: 254

TicketID: 17

This form displays a passengers ticket ID by entering their passenger ID.

The screenshot shows the Microsoft Access interface. On the left, the 'All Access Objects' task pane is open, displaying a list of tables and forms. The 'Passengers' form is selected under the 'Forms' section. The main window shows the 'Passengers' form with the following data:

Field	Value
PassengerID	254
PassengerName	Michael Bean
ContactNumber	646-321-6547

This form allows you to search for passengers name and contact info by using their passenger ID.

The screenshot shows the Microsoft Access interface. On the left, the 'All Access Objects' task pane is open, displaying a list of tables and forms. The 'Stations' form is selected under the 'Forms' section. The main window shows the 'Stations' form with the following data:

Field	Value
StationID	10
StationName	Port Washington
StationAddress	Haven Ave, Port Washington, NY 11050
ConnectionRoute	Plandome

This form gives you a station name, address, and connecting routes by entering a station ID.

All Access Objects << Employees EmployeeTrainSchedule Passengers PassengerTicket Stations Tickets

Search...

Tables

- Employees
- EmployeeTrainSchedule
- Passengers
- PassengerTicket
- Stations
- Tickets
- Trains
- TrainSchedule
- TrainScheduleStations
- TrainScheduleTrain
- TrainScheduleTrainStatus
- TrainStatus

Queries

Forms

- Employees
- EmployeeTrainSchedule
- Passengers
- PassengerTicket
- Stations
- Tickets**
- Trains
- TrainSchedule
- TrainScheduleStations

Tickets

Search for ticket

TicketID

TicketTime 7:30:00 AM

TicketDate 4/11/2021

PaymentMethod Cash

TicketType Ticket

TrainScheduleID 3

TrainID 3700

This form will return ticket time, ticket date, payment method, type of ticket, train schedule ID, and train ID given the ticket ID.

All Access Objects << EmployeeTrainSchedule Passengers PassengerTicket Stations

Search...

Tables

- Employees
- EmployeeTrainSchedule
- Passengers
- PassengerTicket
- Stations
- Tickets
- Trains
- TrainSchedule
- TrainScheduleStations
- TrainScheduleTrain
- TrainScheduleTrainStatus
- TrainStatus

Queries

Forms

- Employees
- EmployeeTrainSchedule
- Passengers
- PassengerTicket
- Stations
- Tickets
- Trains
- TrainSchedule
- TrainScheduleStations
- TrainScheduleTrain
- TrainScheduleTrainStatus**
- TrainStatus

TrainScheduleTrainStatus

TrainScheduleID

TrainStatusID 0

This form allows you to see a train's status by searching the train schedule ID.

All Access Objects

Search...

Tables

- Employees
- EmployeeTrainSchedule
- Passengers
- PassengerTicket
- Stations
- Tickets
- Trains
- TrainSchedule
- TrainScheduleStations
- TrainScheduleTrain
- TrainScheduleTrainStatus
- TrainStatus

Queries

Forms

- Employees
- EmployeeTrainSchedule
- Passengers
- PassengerTicket
- Stations
- Tickets
- Trains
- TrainSchedule

Trains

TrainID: 1352

TrainName: Camila

TrainCapacity: 1000

TrainYear: 1997

TrainModel: x10

This form will portray train name, capacity, year, and model when the train ID is entered.

All Access Objects

Search...

Tables

- Employees
- EmployeeTrainSchedule
- Passengers
- PassengerTicket
- Stations
- Tickets
- Trains
- TrainSchedule
- TrainScheduleStations
- TrainScheduleTrain
- TrainScheduleTrainStatus
- TrainStatus

Queries

Forms

- Employees
- EmployeeTrainSchedule
- Passengers
- PassengerTicket
- Stations
- Tickets
- Trains
- TrainSchedule
- TrainScheduleStations
- TrainScheduleTrain
- TrainScheduleTrainStatus
- TrainStatus

TrainScheduleTrain

TrainScheduleID: 1

TrainID: 3595

This form will show you a train ID when a train schedule ID is entered.

All Access Objects << >>

Search...

Tables

- Employees
- EmployeeTrainSchedule
- Passengers
- PassengerTicket
- Stations
- Tickets
- Trains
- TrainSchedule
- TrainScheduleStations
- TrainScheduleTrain
- TrainScheduleTrainStatus
- TrainStatus

Queries

Forms

- Employees
- EmployeeTrainSchedule
- Passengers
- PassengerTicket
- Stations
- Tickets
- Trains
- TrainSchedule
- TrainScheduleStations**
- TrainScheduleTrain

Employees EmployeeTrainSchedule Passengers PassengerTicket Stations

TrainScheduleStations

TrainScheduleID	<input type="text" value="15"/>
TrainID	<input type="text" value="3705"/>
StationID	<input type="text" value="10"/>

This form will allow you to input a train schedule ID to see the train ID and the station ID.

All Access Objects

Search...

Tables

- Employees
- EmployeeTrainSchedule
- Passengers
- PassengerTicket
- Stations
- Tickets
- Trains
- TrainSchedule
- TrainScheduleStations
- TrainScheduleTrain
- TrainScheduleTrainStatus
- TrainStatus

Queries

Forms

- Employees
- EmployeeTrainSchedule
- Passengers
- PassengerTicket
- Stations
- Tickets
- Trains
- TrainSchedule
- TrainScheduleStations
- TrainScheduleTrain
- TrainScheduleTrainStatus
- TrainStatus

Reports

Passengers PassengerTicket Stations Tickets Trains

TrainStatus

TrainStatusID	<input type="text" value="0"/>
TrainStatus	<input type="text" value="Good"/>

This form will show you what a specific train status ID signifies by entering the ID.

All Access Objects << >>

Search...

Tables

- Employees
- EmployeeTrainSchedule
- Passengers
- PassengerTicket
- Stations
- Tickets
- Trains
- TrainSchedule
- TrainScheduleStations
- TrainScheduleTrain
- TrainScheduleTrainStatus
- TrainStatus

Queries

Forms

- Employees
- EmployeeTrainSchedule
- Passengers
- PassengerTicket
- Stations
- Tickets
- Trains
- TrainSchedule**
- TrainScheduleStations
- TrainScheduleTrain
- TrainScheduleTrainStatus

TrainSchedule

TrainScheduleID

startStation

endStation

startTime

endTime

Day

This form shows you a train's start/end time, start/end station, and days the train operates by entering the train schedule ID.

Reports

Train schedule report

TrainSchedule - Status Report						
TrainScheduleID	startStation	endStation	startTime	endTime	Day	StatusID
6	Port Washington	Penn Station	10:30:00 AM	12:00:00 PM	Everyday	1
11	Penn Station	Port Washington	1:00:00 PM	3:30:00 PM	Everyday	1
16	Penn Station	Port Washington	6:00:00 PM	8:30:00 PM	Everyday	1
Saturday, May 15, 2021						Page 1 of 1

Employees on trainschedule report

Employees - TrainSchedule Report						
EmployeeName	TrainScheduleID	startStation	endStation	startTime	endTime	Day
Alexia Barrett	7	Port Washington	Great Neck	11:00:00 AM	11:45:00 AM	Tuesday
Chester Brooks	12	Penn Station	Port Washingto	2:00:00 PM	4:30:00 PM	Everyday
Clark Armstrong	14	Penn Station	Port Washingto	4:00:00 PM	6:30:00 PM	Everyday
David Chapman	5	Port Washington	Penn Station	9:30:00 AM	11:00:00 AM	Everyday
Eleanor Alexander	10	Great Neck	Bayside	11:00:00 AM	11:45:00 AM	Friday
Evelyn Wright	11	Penn Station	Port Washingto	1:00:00 PM	3:30:00 PM	Everyday
	16	Penn Station	Port Washingto	6:00:00 PM	8:30:00 PM	Everyday
Frederick Mitchell	2	Port Washington	Manhasset	8:30:00 AM	9:00:00 AM	Monday
	6	Port Washington	Penn Station	10:30:00 AM	12:00:00 PM	Everyday
John Cameron	16	Penn Station	Port Washingto	6:00:00 PM	8:30:00 PM	Everyday
	8	Great Neck	Little Neck	10:00:00 AM	10:15:00 AM	Wednesda
Kelvin Dixon	14	Penn Station	Port Washingto	4:00:00 PM	6:30:00 PM	Everyday
Kirsten Harris	13	Penn Station	Port Washingto	3:00:00 PM	5:30:00 PM	Everyday
Lucas Nelson	9	Great Neck	Douglaston	8:30:00 AM	9:00:00 AM	Thursday
Lyndon Kelly	16	Penn Station	Port Washingto	6:00:00 PM	8:30:00 PM	Everyday
Melissa Turner	2	Port Washington	Manhasset	8:30:00 AM	9:00:00 AM	Monday
Ned Wilson	6	Port Washington	Penn Station	10:30:00 AM	12:00:00 PM	Everyday
	14	Penn Station	Port Washingto	4:00:00 PM	6:30:00 PM	Everyday
Robert Adams	10	Great Neck	Bayside	11:00:00 AM	11:45:00 AM	Friday
	2	Port Washington	Manhasset	8:30:00 AM	9:00:00 AM	Monday
Ryan Cunningham	16	Penn Station	Port Washingto	6:00:00 PM	8:30:00 PM	Everyday
	5	Port Washington	Penn Station	9:30:00 AM	11:00:00 AM	Everyday
Ted Williams	8	Great Neck	Little Neck	10:00:00 AM	10:15:00 AM	Wednesda
Tyler Sullivan	7	Port Washington	Great Neck	11:00:00 AM	11:45:00 AM	Tuesday
Victor Phillips	3	Port Washington	Penn Station	7:30:00 AM	9:00:00 AM	Everyday
Vivian Brooks	14	Penn Station	Port Washingto	4:00:00 PM	6:30:00 PM	Everyday
	9	Great Neck	Douglaston	8:30:00 AM	9:00:00 AM	Thursday
	6	Port Washington	Penn Station	10:30:00 AM	12:00:00 PM	Everyday

Saturday, May 15, 2021

Page 1 of 1

Trainschedule before 12am

TrainSchedule - Before 12pm Report

TrainScheduleID	startStation	endStation	startTime	endTime	Day
3	Port Washington	Penn Station	7:30:00 AM	9:00:00 AM	Everyday
4	Port Washington	Penn Station	8:30:00 AM	10:00:00 AM	Everyday
5	Port Washington	Penn Station	9:30:00 AM	11:00:00 AM	Everyday
6	Port Washington	Penn Station	10:30:00 AM	12:00:00 PM	Everyday

Saturday, May 15, 2021

Page 1 of 1

Report on sold e-tickets

Passengers - E-Ticket from Penn Station Report

PassengerName	TicketType	startStation
Michael Bean	E-Ticket	Penn Station
Lillian Mcquire	E-Ticket	Penn Station
Steve Jobs	E-Ticket	Penn Station
	E-Ticket	Penn Station
Rachel Bondy	E-Ticket	Penn Station

Saturday, May 15, 2021

Page 1 of 1

Report on trains and their capacity

Trains Filter Report

TrainID	TrainName	TrainCapacity
3098	Kelvin	2500
1159	Aldus	2750
4583	Alford	5000

Saturday, May 15, 2021

Page 1 of 1

Conclusion

Overall, our group had a good experience completing this project. The project was a great test of our group's knowledge on the topics that were covered in this course, and it made it clear how this knowledge is useful for real practices. The most difficult thing for us was in the beginning when we had to figure out how to go about this project and make sure to keep our focus on what we were actually solving with this database. We quickly learned that the first phase of the project was essential as it will lay the basis of the rest of the project. Therefore we spent a lot of time with the first part where we built the ER diagram and then the RDM conversion and normalization. We tried as best as possible to keep ourselves on track with regular meetings to make sure we were all aligned and on the right path with our project. We were challenged at every stage, but we had a good team dynamic with collaborative efforts. If one member of the group members struggled with their specific part, other members stepped in to make sure the task was done correctly. This project helped us sharpen our skills in SQL for future use.

For the MTA we have built a database that we believe would be useful as it will give a broad overview of operations. We have successfully helped the MTA optimize the efficiency and reliability of the LIRR line.