Sprint 0

Project team 3 - Light Rail Management System(LRMS)

Team Members: Harshal Sharma, Shubhra Mishra, Darshak Mehta, Maryam Khabazi

PROJECT PROPOSAL

Content:

LRMS is a web based system that facilitates managing the functioning of the light rail transit. This system will aid in developing the whole functioning virtual.

Vision/Scope:

The goal is to create a light rail management system (LRMS) for a city to book tickets, check schedules, routes across the city, help track employee, inventory, and carriage details, etc.

Objective:

- Managing for light rail transit system.
- Booking ticket for light rail transit system.
- Checking the schedule and routes of light rail transit system.
- Tracking the cost of maintenance of Carriages, Stations, and Rails.

PROJECT ENVIRONMENT

Environment Setup:

1. Sublime → https://download.sublimetext.com/Sublime%20Text%20Build%203143%20x64%20Setup.exe
2. Xampp → https://www.apachefriends.org/xampp-files/7.1.9/xampp-win32-7.1.9-0-VC14-installer.exe
3. MySQL phpMyAdmin → http://www.phpmyadmin.net
4. MySQL Workbench → https://dev.mysql.com/get/Downloads/MySQLGUITools/mysql-workbench-community-6.3.9-winx64.msi
5. PHP Version 7.1.8 → http://php.net/docs.php

The code editor which is going to be used is Sublime Text version 3 for writing and editing the code in PHP. For MySQL Database Management System the project is going to use Xampp server version 3.2.2 and for the web interface the project is going to use phpMyAdmin for representing the database visually. We will be using MySQL Workbench in future, if phpmyadmin falls short of table row data or integrating environment. As of now, we do not require Workbench, and phpmyadmin will suffice.

HIGH LEVEL REQUIREMENTS

Initial user roles

User Role	Description
Passenger	Users who are making reservations on the LRMS. Passenger can buy ticket while selecting different kinds of tickets and can also view routes.
Admin	Administrative users for the system. Admins can view and update schedule, assign routes to the trains and assign drivers to the routes.
Employee	Users who will be the employees of the LRMS. Employee can view their payroll and can check their shifts.

Initial user story descriptions

Story ID	Story description
US1	As a Passenger I want to check the schedule of the trains so that I can get information about routes
US2	As a Passenger I want to book a ticket.

US3	As an Admin I want to update the schedule so that passengers can view the correct schedule.
US4	As an Admin I want to assign shift to employee.
US5	As an Employee I want to check my payroll.
US6	As an Employee I want to check my shift.

HIGH LEVEL CONCEPTUAL DESIGN

Entities:

Passenger

Admin

Employee

Schedule

** Route(this can be a potential entity for the future and can hold a separate identity apart from Schedule, but as of now it is analogous to Schedule)

Relationships:

Passenger checks **Schedule**

Admin manages **Schedule**

Admin assigns shifts to/manages **Employee**

Sprint 1

REQUIREMENTS

Story ID	Story description
US1	As an Admin I want to register into the system.
US2	As an Admin I want to login into the system.
US3	As an Admin I want to add station to the system.
US4	As an Admin I want to add schedule for a particular train.
US5	As an Admin I want to add routes for stations.
US6	As a Passenger I want to check the schedule of the trains so that I can get information about routes.
US7	As a Passenger I want to book a ticket.
US8	As an Admin I want to update the schedule so that passengers can view the correct schedule.
US9	As an Admin I want to assign shifts to employee.
US10	As an Employee I want to register to the system.
US11	As an Employee I want to login to the system.
US12	As an Employee I want to check my payroll.

US13	As an Employee I want to check my shift.
US14	As an Employee I want to update my information in the system, such as address, ...

CONCEPTUAL DESIGN

Entity: **Admin**
Attributes:

- <u>ssn</u>
- name (composite)
  - first_name
  - last_name
- email_id(unique)
- password

Entity: **Station**

Attributes:

- <u>id</u>
- name

Entity: **Route**

Attributes:

- <u>id</u>
- name
- start_station

end_station

Entity: **Schedule**

Attributes:

id

start_time

end_time

Justification: For now the start_time and end_time are considered single-valued, because we consider we have one train that runs in a same time all days of the week. However, in later sprints, we will change it to multi-valued by considering different time schedule for weekdays and weekend.

Relationship: **Admin** adds **Station**
Cardinality: One to Many
Participation:
Admin has total participation
Station has total participation
Justification: There will be only 1 admin for the system.

Relationship: **Admin** assigns **Schedule**
Cardinality: One to Many
Participation:
Admin has total participation
Schedule has total participation

Relationship: **Admin** adds **Route**
Cardinality: One to Many
Participation:
Admin has total participation
Route has total participation

Relationship: **Route** has **Station**
Cardinality: One to Many
Participation:
Route has total participation

Station has total participation

LOGICAL DESIGN

Table: Admin

Columns:

<u>ssn</u>

first_name

last_name

email_id(unique)

password

Table: Route

Columns:

<u>id</u>

name

start_station

end_station

Table: Station

Columns:

<u>id</u>

name

route_id [Foreign Key; references **id** of **Route**]

Table: Schedule

Columns:

id

start_time

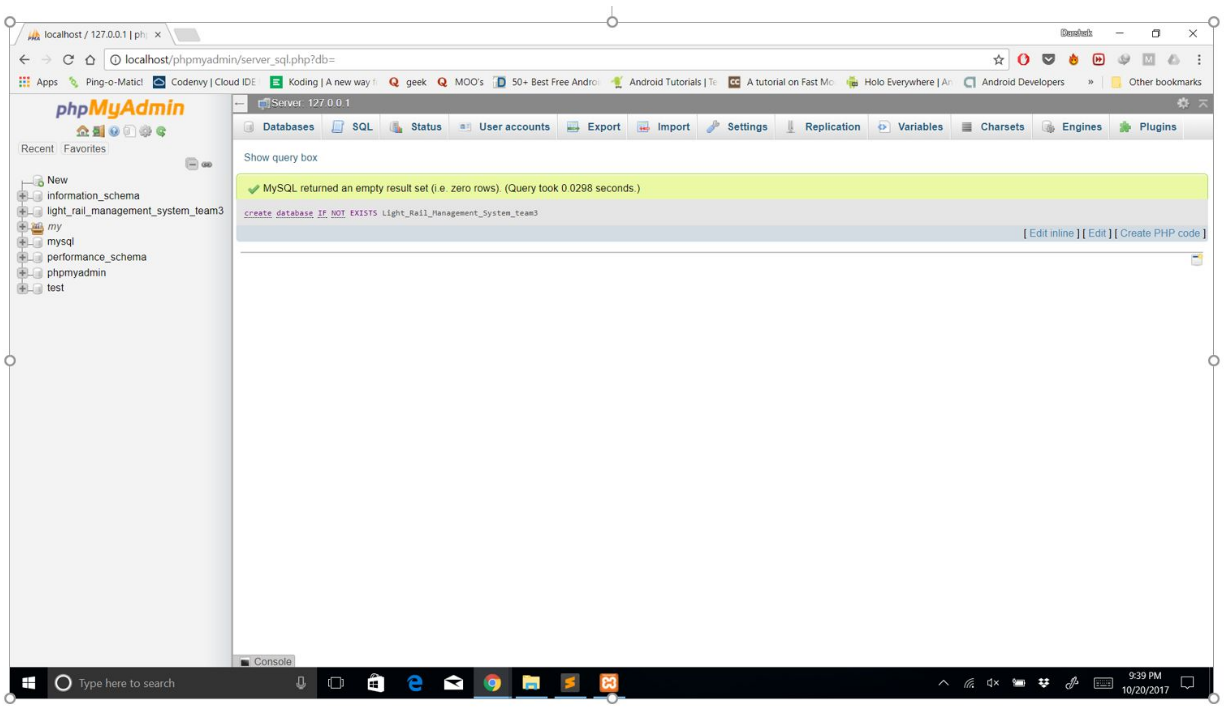
End_time

route_id [Foreign Key, references **id** of **Route**]

SQL QUERIES

1. CREATE DATABASE IF NOT EXISTS

Light_Rail_Management_System_Team3;


localhost / 127.0.0.1 | php
localhost/phpmyadmin/server_sql.php?db=
phpMyAdmin
Recent Favorites
New
information_schema
light_rail_management_system_team3
my
mysql
performance_schema
phpmyadmin
test
Server: 127.0.0.1
Databases SQL Status User accounts Export Import Settings Replication Variables Charsets Engines Plugins
Show query box
[Edit inline] [Edit] [Create PHP code]
Console
Type here to search
9:39 PM
10/20/2017

MySQL returned an empty result set (i.e. zero rows). (Query took 0.0298 seconds.)

```
create database IF NOT EXISTS Light_Rail_Management_System_team3
```

2. USE Light_Rail_Management_System_Team3;

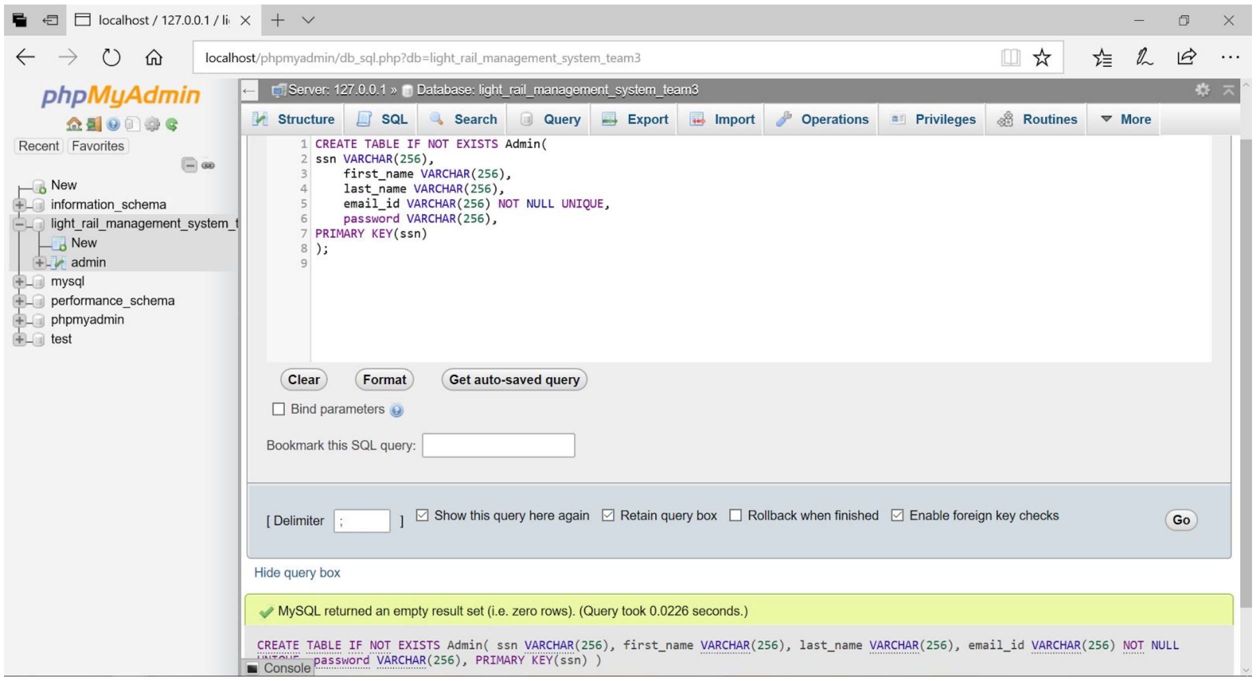
3. CREATE TABLE IF NOT EXISTS Admin(

ssn VARCHAR(256),

first_name VARCHAR(256),

last_name VARCHAR(256),

```
        email_id VARCHAR(256) NOT NULL UNIQUE,
        password VARCHAR(256),
PRIMARY KEY(ssn)
);
```



```
CREATE TABLE IF NOT EXISTS Admin(
ssn VARCHAR(256),
    first_name VARCHAR(256),
    last_name VARCHAR(256),
    email_id VARCHAR(256) NOT NULL UNIQUE,
    password VARCHAR(256),
PRIMARY KEY(ssn)
);
```

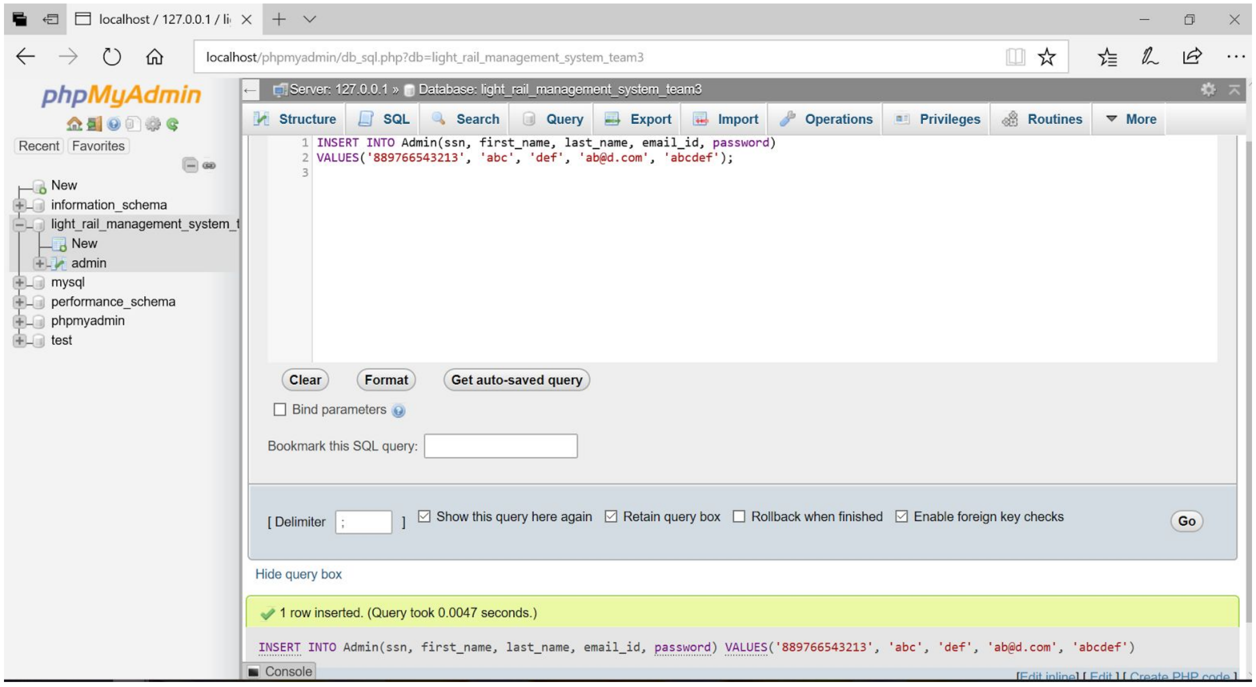
[Delimiter ;] ☑ Show this query here again ☑ Retain query box ☐ Rollback when finished ☑ Enable foreign key checks

MySQL returned an empty result set (i.e. zero rows). (Query took 0.0226 seconds.)

```
CREATE TABLE IF NOT EXISTS Admin( ssn VARCHAR(256), first_name VARCHAR(256), last_name VARCHAR(256), email_id VARCHAR(256) NOT NULL
UNIQUE, password VARCHAR(256), PRIMARY KEY(ssn) )
```

4. INSERT INTO Admin(ssn, first_name, last_name, email_id, password)

VALUES('889766543213', 'abc', 'def', 'ab@d.com', 'abcdef');


localhost / 127.0.0.1 / li
localhost/phpmyadmin/db_sql.php?db=light_rail_management_system_team3
phpMyAdmin
Recent
Favorites
New
information_schema
light_rail_management_system_t
New
admin
mysql
performance_schema
phpmyadmin
test
Server: 127.0.0.1 » Database: light_rail_management_system_team3
Structure
SQL
Search
Query
Export
Import
Operations
Privileges
Routines
More
Clear
Format
Get auto-saved query
Go
Hide query box
Console
[Edit inline] [Edit] [Create PHP code]

```
INSERT INTO Admin(ssn, first_name, last_name, email_id, password)
VALUES('889766543213', 'abc', 'def', 'ab@d.com', 'abcdef');
```

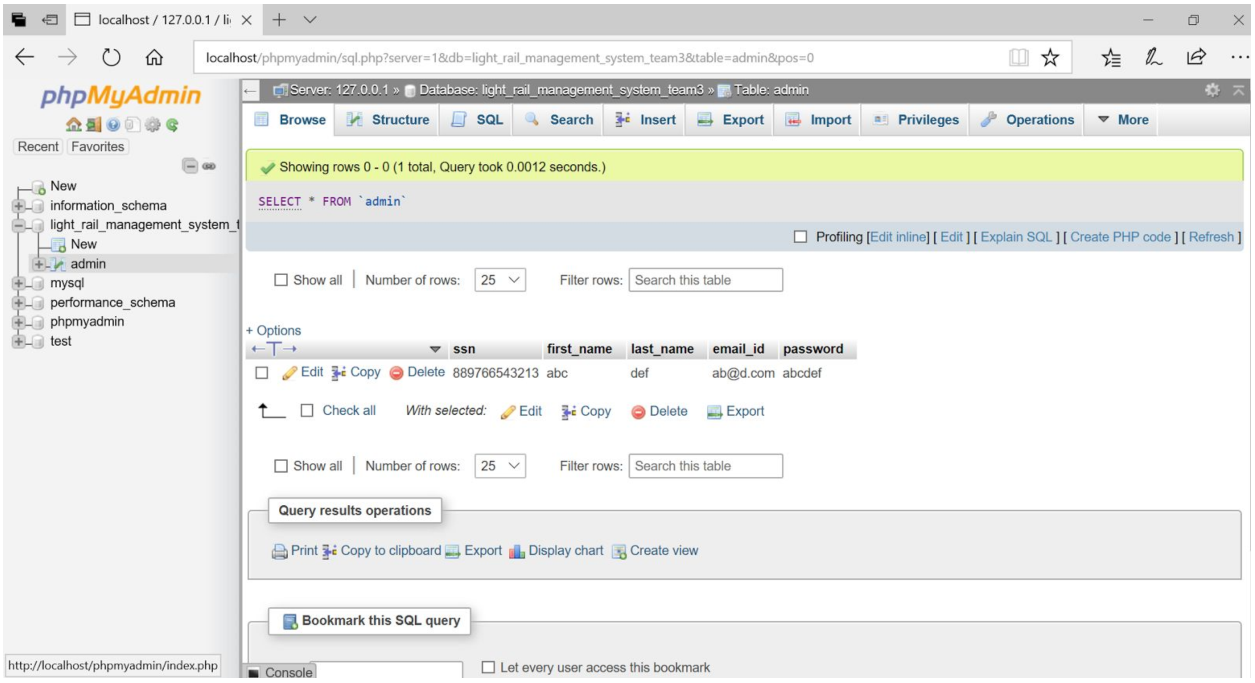
☐ Bind parameters

Bookmark this SQL query:

[Delimiter ;] ☑ Show this query here again ☑ Retain query box ☐ Rollback when finished ☑ Enable foreign key checks

✓ 1 row inserted. (Query took 0.0047 seconds.)

```
INSERT INTO Admin(ssn, first_name, last_name, email_id, password) VALUES('889766543213', 'abc', 'def', 'ab@d.com', 'abcdef')
```


localhost / 127.0.0.1 / li
localhost/phpmyadmin/sql.php?server=1&db=light_rail_management_system_team3&table=admin&pos=0
phpMyAdmin
Recent
Favorites
New
information_schema
light_rail_management_system_t
New
admin
mysql
performance_schema
phpmyadmin
test
Server: 127.0.0.1 » Database: light_rail_management_system_team3 » Table: admin
Browse
Structure
SQL
Search
Insert
Export
Import
Privileges
Operations
More
Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
+ Options
Check all
With selected:
Edit
Copy
Delete
Export
Query results operations
Print
Copy to clipboard
Export
Display chart
Create view
Bookmark this SQL query
Let every user access this bookmark
http://localhost/phpmyadmin/index.php
Console

✓ Showing rows 0 - 0 (1 total, Query took 0.0012 seconds.)

```
SELECT * FROM `admin`
```

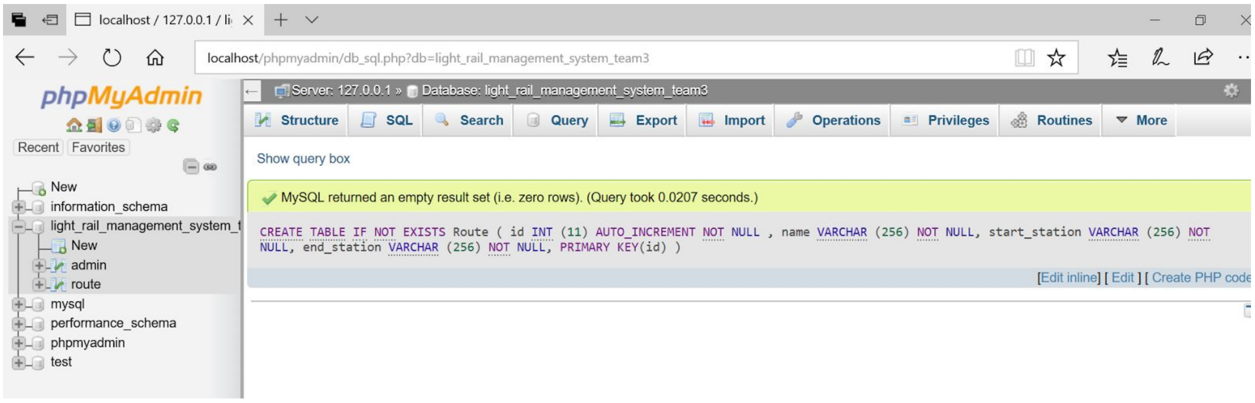
☐ Show all | Number of rows: 25 Filter rows: Search this table

	ssn	first_name	last_name	email_id	password
Edit Copy Delete	889766543213	abc	def	ab@d.com	abcdef

5. CREATE TABLE IF NOT EXISTS Route (

id INT (11) AUTO_INCREMENT NOT NULL ,

```
        name VARCHAR (256) NOT NULL,
        start_station VARCHAR (256) NOT NULL,
        end_station VARCHAR (256) NOT NULL,
        PRIMARY KEY (id)
);
```


localhost / 127.0.0.1 / li
localhost/phpmyadmin/db_sql.php?db=light_rail_management_system_team3
phpMyAdmin
Recent
Favorites
New
information_schema
light_rail_management_system_t
New
admin
route
mysql
performance_schema
phpmyadmin
test
Server: 127.0.0.1 » Database: light_rail_management_system_team3
Structure
SQL
Search
Query
Export
Import
Operations
Privileges
Routines
More
Show query box
[Edit inline] [Edit] [Create PHP code

MySQL returned an empty result set (i.e. zero rows). (Query took 0.0207 seconds.)

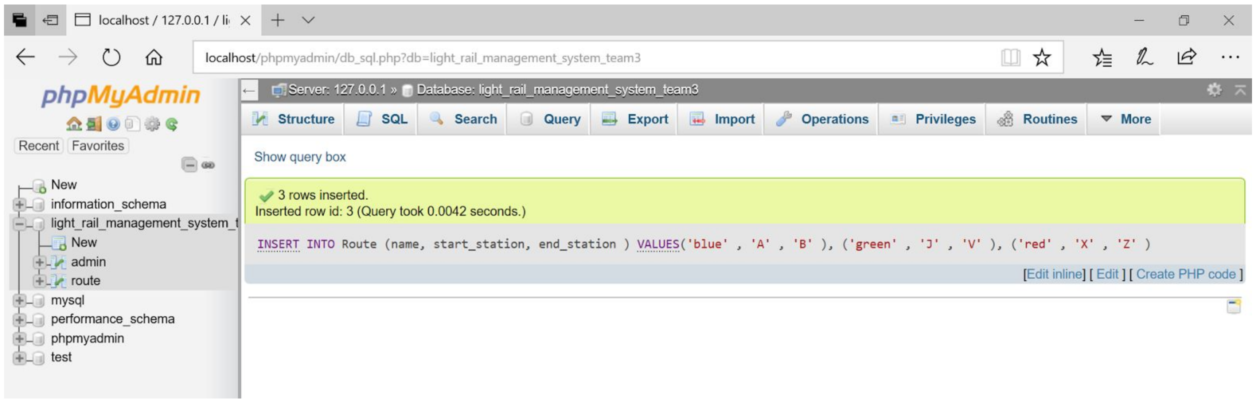
```
CREATE TABLE IF NOT EXISTS Route ( id INT (11) AUTO_INCREMENT NOT NULL , name VARCHAR (256) NOT NULL, start_station VARCHAR (256) NOT
NULL, end_station VARCHAR (256) NOT NULL, PRIMARY KEY(id) )
```

6. INSERT INTO Route (name, start_station, end_station)

VALUES('blue' , 'A' , 'B'),

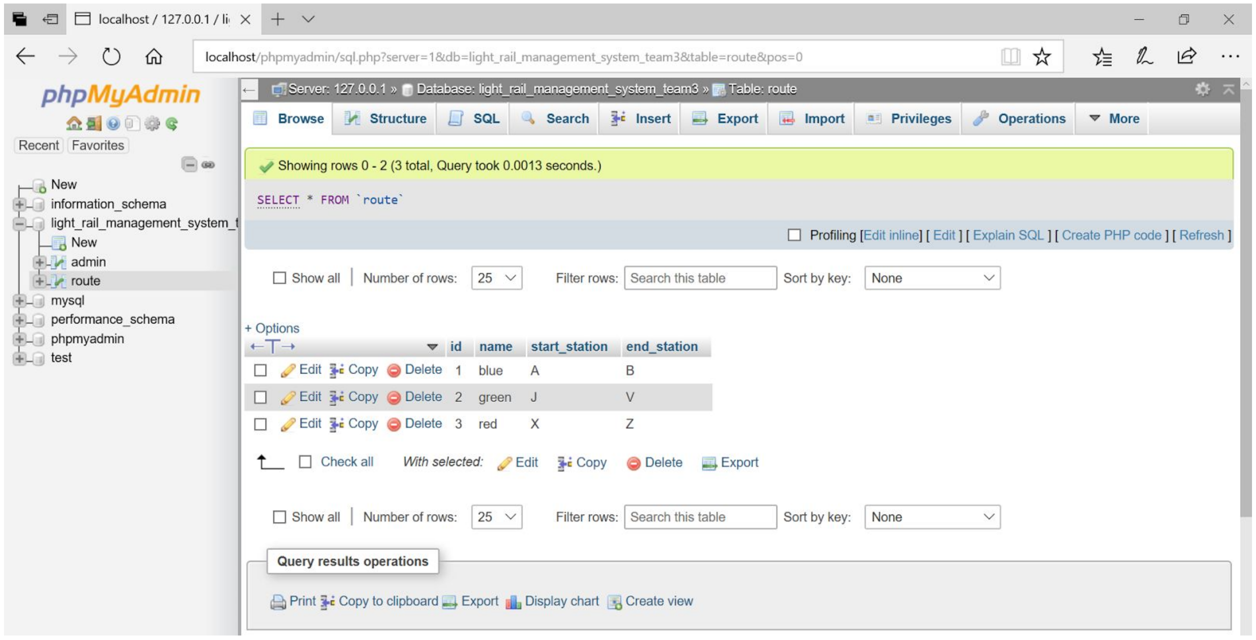
('green' , 'J' , 'V'),

('red' , 'X' , 'Z');


localhost / 127.0.0.1 / li
localhost/phpmyadmin/db_sql.php?db=light_rail_management_system_team3
phpMyAdmin
Recent
Favorites
New
information_schema
light_rail_management_system_t
New
admin
route
mysql
performance_schema
phpmyadmin
test
Server: 127.0.0.1 » Database: light_rail_management_system_team3
Structure
SQL
Search
Query
Export
Import
Operations
Privileges
Routines
More
Show query box
[Edit inline] [Edit] [Create PHP code]

3 rows inserted.
Inserted row id: 3 (Query took 0.0042 seconds.)

```
INSERT INTO Route (name, start_station, end_station ) VALUES('blue' , 'A' , 'B' ), ('green' , 'J' , 'V' ), ('red' , 'X' , 'Z' )
```


localhost/phpmyadmin/sql.php?server=1&db=light_rail_management_system_team3&table=route&pos=0

Server: 127.0.0.1 » Database: light_rail_management_system_team3 » Table: route

Showing rows 0 - 2 (3 total, Query took 0.0013 seconds.)

```
SELECT * FROM `route`
```

id	name	start_station	end_station
1	blue	A	B
2	green	J	V
3	red	X	Z

7. CREATE TABLE IF NOT EXISTS Station(

id INT(11) AUTO_INCREMENT NOT NULL,

name VARCHAR(256) NOT NULL,

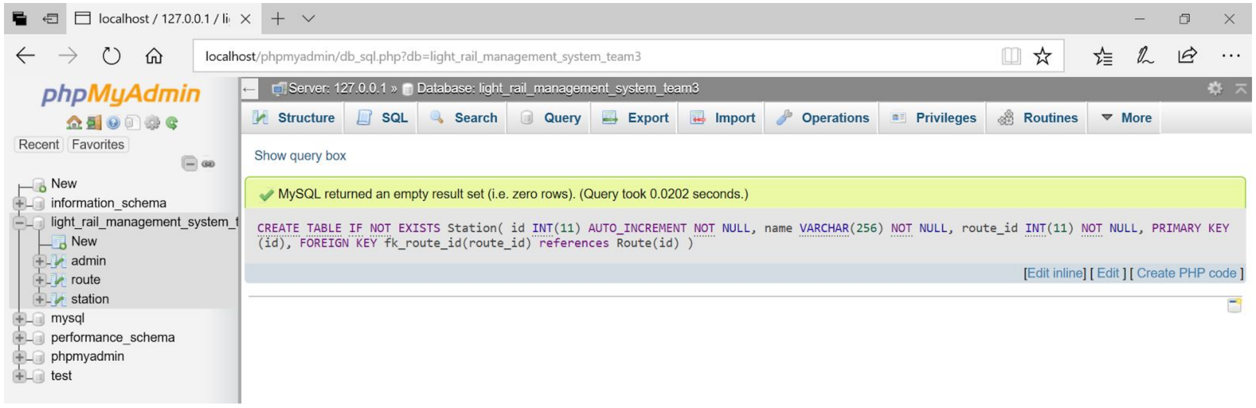
route_id INT(11) NOT NULL,

PRIMARY KEY (id),

FOREIGN KEY fk_route_id(route_id)

references Route(id)

);


localhost/phpmyadmin/db_sql.php?db=light_rail_management_system_team3

Server: 127.0.0.1 » Database: light_rail_management_system_team3

MySQL returned an empty result set (i.e. zero rows). (Query took 0.0202 seconds.)

```
CREATE TABLE IF NOT EXISTS Station( id INT(11) AUTO_INCREMENT NOT NULL, name VARCHAR(256) NOT NULL, route_id INT(11) NOT NULL, PRIMARY KEY
(id), FOREIGN KEY fk_route_id(route_id) references Route(id) )
```

8. INSERT INTO Station(name, route_id)

VALUES('A', 1),

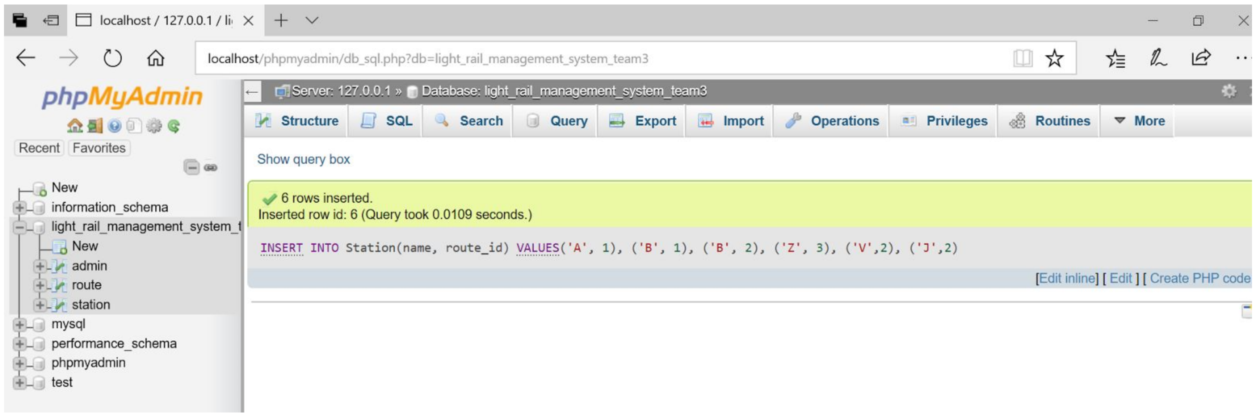
('B', 1),

('B', 2),

('Z', 3),

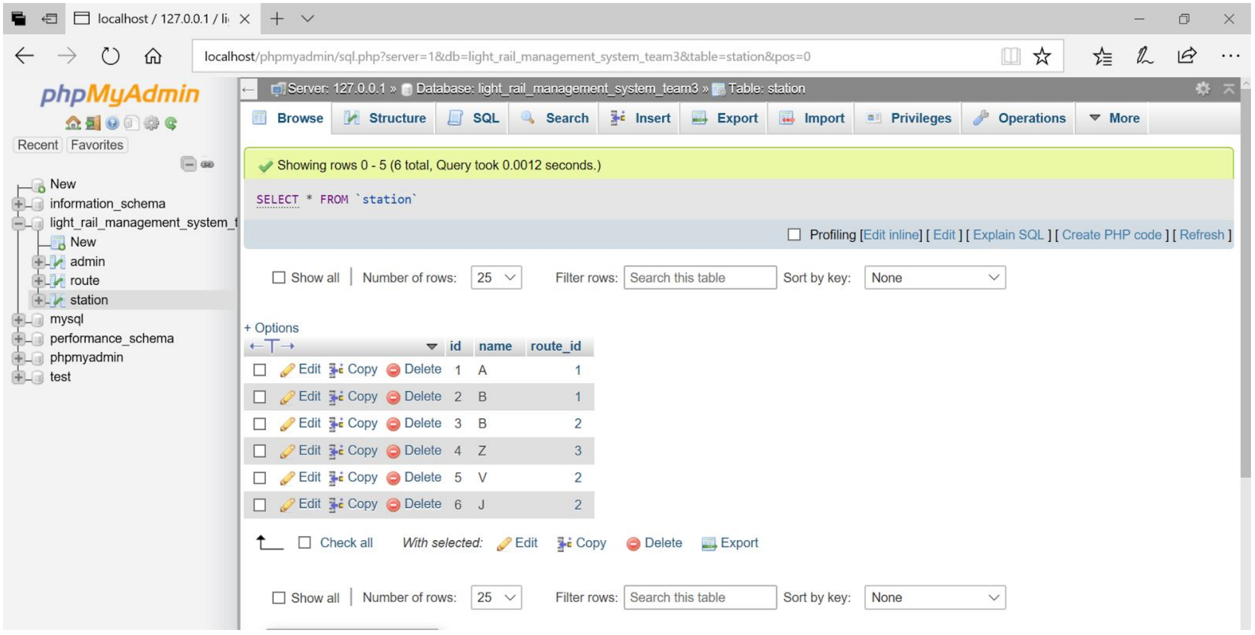
('V',2),

('J',2);


localhost / 127.0.0.1 / li
localhost/phpmyadmin/db_sql.php?db=light_rail_management_system_team3
phpMyAdmin
Recent
Favorites
New
information_schema
light_rail_management_system_t
New
admin
route
station
mysql
performance_schema
phpmyadmin
test
Server: 127.0.0.1 » Database: light_rail_management_system_team3
Structure
SQL
Search
Query
Export
Import
Operations
Privileges
Routines
More
Show query box
[Edit inline] [Edit] [Create PHP code

6 rows inserted.
Inserted row id: 6 (Query took 0.0109 seconds.)

```
INSERT INTO Station(name, route_id) VALUES('A', 1), ('B', 1), ('B', 2), ('Z', 3), ('V',2), ('J',2)
```


localhost / 127.0.0.1 / li
localhost/phpmyadmin/sql.php?server=1&db=light_rail_management_system_team3&table=station&pos=0
phpMyAdmin
Recent
Favorites
New
information_schema
light_rail_management_system_t
New
admin
route
station
mysql
performance_schema
phpmyadmin
test
Server: 127.0.0.1 » Database: light_rail_management_system_team3 » Table: station
Browse
Structure
SQL
Search
Insert
Export
Import
Privileges
Operations
More
Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
Show all
Number of rows: 25
Filter rows: Search this table
Sort by key: None
+ Options
Check all
With selected: Edit Copy Delete Export

Showing rows 0 - 5 (6 total, Query took 0.0012 seconds.)

```
SELECT * FROM `station`
```

id	name	route_id
1	A	1
2	B	1
3	B	2
4	Z	3
5	V	2
6	J	2

9. CREATE TABLE IF NOT EXISTS Schedule(

id INT(11) AUTO_INCREMENT NOT NULL,

name VARCHAR(256) NOT NULL,

start_time TIME NOT NULL,

end_time TIME NOT NULL,

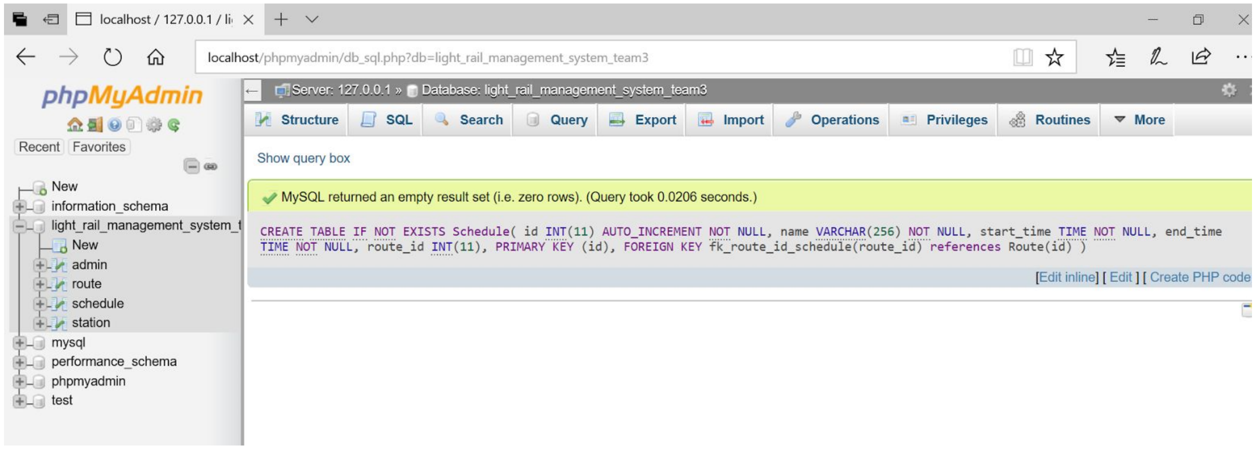
route_id INT(11),

PRIMARY KEY (id),

FOREIGN KEY fk_route_id_schedule(route_id)

references Route(id)

);



MySQL returned an empty result set (i.e. zero rows). (Query took 0.0206 seconds.)

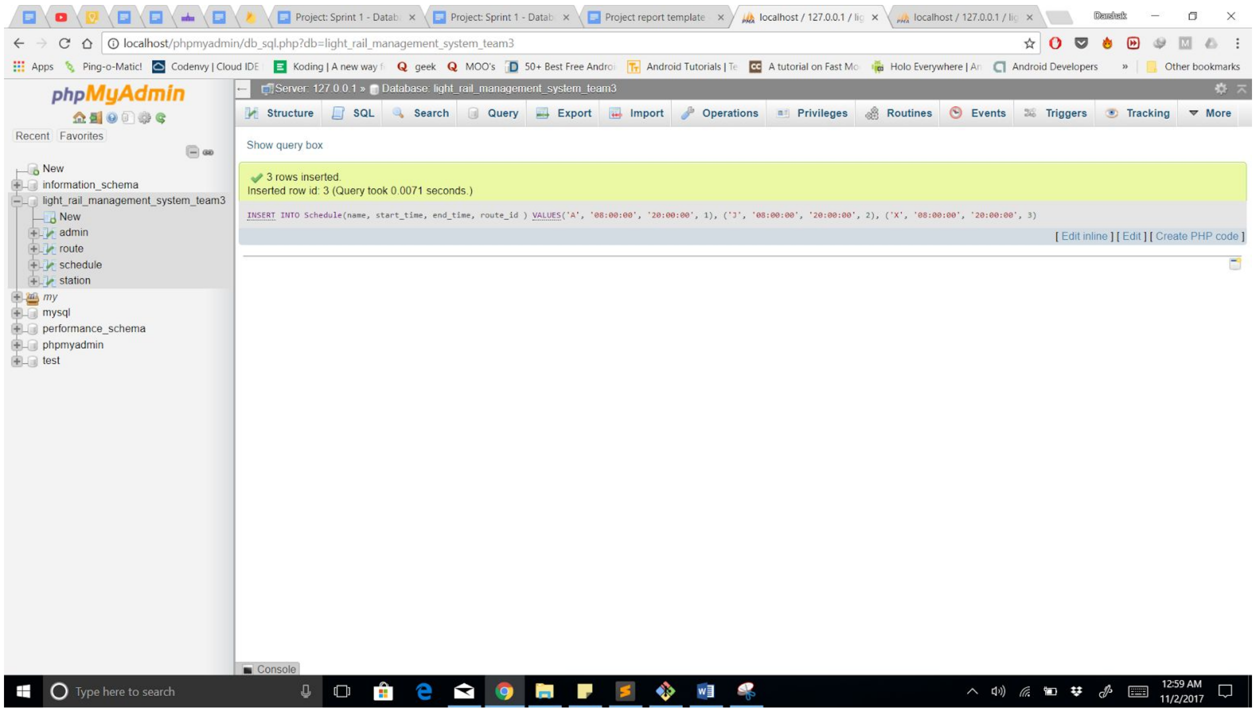
```
CREATE TABLE IF NOT EXISTS Schedule( id INT(11) AUTO_INCREMENT NOT NULL, name VARCHAR(256) NOT NULL, start_time TIME NOT NULL, end_time TIME NOT NULL, route_id INT(11), PRIMARY KEY (id), FOREIGN KEY fk_route_id_schedule(route_id) references Route(id) )
```

10. INSERT INTO Schedule(name, start_time, end_time, route_id)

VALUES('A', '08:00:00', '20:00:00', 1),

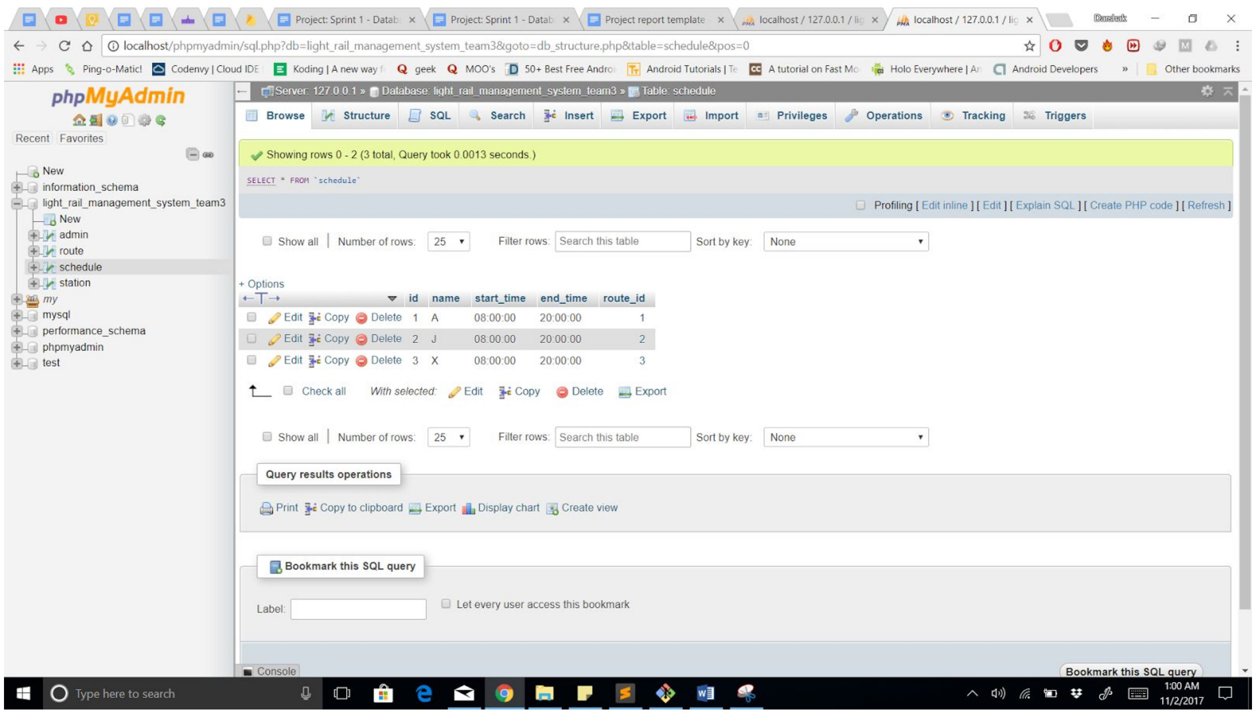
('J', '08:00:00', '20:00:00', 2),

('X', '08:00:00', '20:00:00', 3);



3 rows inserted.
Inserted row id: 3 (Query took 0.0071 seconds.)

```
INSERT INTO Schedule(name, start_time, end_time, route_id ) VALUES('A', '08:00:00', '20:00:00', 1), ('J', '08:00:00', '20:00:00', 2), ('X', '08:00:00', '20:00:00', 3)
```


Project: Sprint 1 - Datab
Project: Sprint 1 - Datab
Project report template
localhost / 127.0.0.1 / lig
localhost / 127.0.0.1 / lig
phpMyAdmin
Recent Favorites
New
information_schema
light_rail_management_system_team3
New
admin
route
schedule
station
my
mysql
performance_schema
phpmyadmin
test
Browse
Structure
SQL
Search
Insert
Export
Import
Privileges
Operations
Tracking
Triggers
Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
+ Options
Check all
With selected: Edit Copy Delete Export
Query results operations
Print Copy to clipboard Export Display chart Create view
Bookmark this SQL query
Let every user access this bookmark
Console
Type here to search
1:00 AM
11/2/2017

localhost/phpmyadmin/sql.php?db=light_rail_management_system_team3&goto=db_structure.php&table=schedule&pos=0

Server: 127.0.0.1 » Database: light_rail_management_system_team3 » Table: schedule

Showing rows 0 - 2 (3 total, Query took 0.0013 seconds.)

```
SELECT * FROM `schedule`
```

Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

	id	name	start_time	end_time	route_id
Edit Copy Delete	1	A	08:00:00	20:00:00	1
Edit Copy Delete	2	J	08:00:00	20:00:00	2
Edit Copy Delete	3	X	08:00:00	20:00:00	3

Label:

Sprint 2

REQUIREMENTS

Story ID	**Story description**
US1	As an Admin I want to register into the system.
US2	As an Admin I want to login into the system.
US3	As an Admin I want to add station to the system.
US4	As an Admin I want to add trains for the LRMS.
US5	As an Admin I want to add a list of stations for routes.
US6	As an Admin I want to add trains for routes .
US7	As an Admin I want to add schedule (where, schedule includes information about route, stations, trains, and the timings) for a train belonging to a route.
US8	As a Passenger I want to know all of the routes in the LR system and all the stations that a particular route contains.
US9	As a Passenger I want to (check the schedule) timing of different trains for any particular station.
US10	As a Passenger I want to check the schedule of a train so that I can get information about arrival time of the train across different stations on a particular route.
US11	As a Passenger I want to book a ticket.

US12	As an Admin I want to update the schedule so that passengers can view the correct schedule.
US13	As an Admin I want to assign shifts to employee.
US14	As an Employee I want to register to the system.
US15	As an Employee I want to login to the system.
US16	As an Employee I want to check my payroll.
US17	As an Employee I want to check my shift.
US18	As an Employee I want to update my information in the system, such as address, ...
US19	As an admin I want to add an alert about changes in schedule and routes, or if there is any accident (event) (detours, delays, cancellation)

CONCEPTUAL DESIGN

Entity: **Admin**

Attributes:

- ssn
- name (composite)
    - first_name
    - last_name
- email_id(unique)
- password

Entity: **Station**

Attributes:

- id
- name

Entity: **Route**

Attributes:

- id
- name
- duration

Note: Duration is in Minutes for a Route

Entity: **Schedule**

Attributes:

- id
- estimated_time
- actual_time
- delay
- train_status
- station_status

Entity : Train

id

Entity : Passenger

ticket_id

ticket_type

booking_time

ticket_status

Relationship: **Admin** adds **Station**
Cardinality: One to Many
Participation:

- Admin has total participation
- Station has total participation

Assumptions: There will be only 1 admin for the system.

Relationship: **Admin** assigns **Schedule**
Cardinality: One to Many
Participation:

- Admin has total participation

Schedule has total participation

Relationship: **Admin** adds **Route**
Cardinality: One to Many
Participation:
Admin has total participation
Route has total participation

Relationship: **Admin** adds **Train**
Cardinality: One to Many
Participation:
Admin has total participation
Train has total participation

Assumption: There will be only 1 admin for the system.

Relationship: **Route** has **Station**
Cardinality: One to Many
Participation:
Route has total participation
Station has total participation
Assumptions: Here, we consider that the stations which are not functional yet do not show up on our system, so we don't have any station which does not belongs to any route. Therefore, we have a total participation for Station.

Relationship: **Route** has **Train**
Cardinality: One to Many
Participation:
Route has total participation
Train has total participation

Relationship: **Train** has **Schedule**
Cardinality: One to One
Participation:

Train has total participation
Schedule has total participation

Assumption: Right now we are assuming that every train in our system is up and functional, so every train must have a schedule. Therefore, we have a total participation for train.

Relationship: **Passenger** checks **Route**
Cardinality: Many to Many
Participation:

Passenger has partial participation
Route has partial participation

Relationship: **Passenger** checks **Schedule**
Cardinality: Many to Many
Participation:

Passenger has partial participation
Schedule has partial participation

Relationship: **Passenger** books **Ticket**
Cardinality: One to One
Participation:

Passenger has total participation
Ticket has total participation

LOGICAL DESIGN

Table: Admin
Columns:

- ssn
- first_name
- last_name
- email_id(unique)
- password

Table: Route
Columns:

- name
- duration

Note: duration is the time in minutes for a particular route.

Table: Train
Columns:

- id
- route_name [Foreign Key; references name of Route]

Justification: Since the relationship is one to many between Train and Route hence we have included the primary key of Route table into Train table. We did not use cross-reference approach because it is not as efficient in this case.

Table: Station
Columns:

- id
- name

Table: Schedule
Columns:

- id
- train_status
- station_status
- estimated_time
- actual_time
- delay

station_id [Foreign Key, references id of Station]
train_id [Foreign Key, references id of Train]
route_name [Foreign Key, references name of Route]

Justification: In Schedule table, we have shown Many-to-Many relationship between Train, Station and Route. Therefore we have used the cross reference approach to relate the primary keys of corresponding tables in the Schedule table.

Table: Passenger

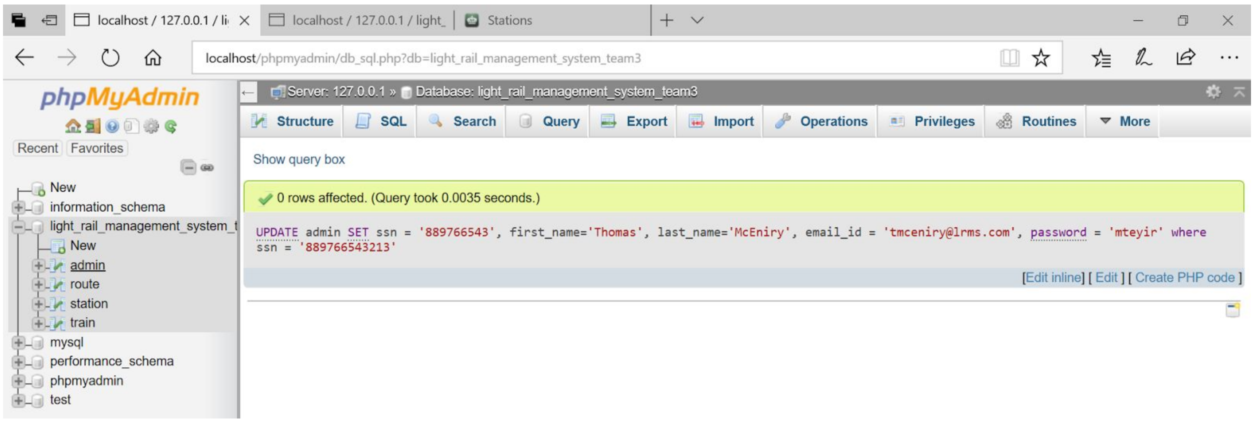
Columns:

<u>ticket_id</u>
ticket_type
booking_time
ticket_status
schedule_id [Foreign Key, references id of Schedule]

Justification: In passenger table, each passenger selects one train from the provided schedule, therefore we used schedule_id as the foreign key to depict the relationship between these entities.

SQL Queries

1. UPDATE admin SET ssn = '889766543', first_name='Thomas', last_name='McEniry', email_id = 'tmceniry@lrms.com', password = 'mteyir' where ssn = '889766543213'


localhost/phpmyadmin/db_sql.php?db=light_rail_management_system_team3
Server: 127.0.0.1 » Database: light_rail_management_system_team3
Structure
SQL
Search
Query
Export
Import
Operations
Privileges
Routines
More
Show query box
New
information_schema
light_rail_management_system_t
admin
route
station
train
mysql
performance_schema
phpmyadmin
test
[Edit inline] [Edit] [Create PHP code]

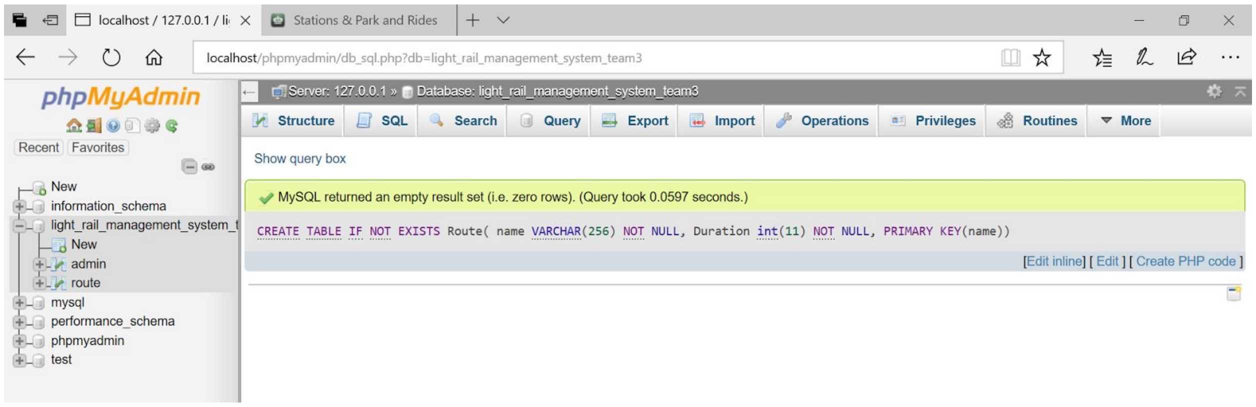
```
0 rows affected. (Query took 0.0035 seconds.)

UPDATE admin SET ssn = '889766543', first_name='Thomas', last_name='McEniry', email_id = 'tmceniry@lrms.com', password = 'mteyir' where
ssn = '889766543213'
```

2. CREATE TABLE IF NOT EXISTS Route(

name VARCHAR(256) NOT NULL,

Duration int(11) NOT NULL,

PRIMARY KEY(name));


localhost / 127.0.0.1 / li
Stations & Park and Rides
localhost/phpmyadmin/db_sql.php?db=light_rail_management_system_team3
phpMyAdmin
Recent
Favorites
New
information_schema
light_rail_management_system_t
New
admin
route
mysql
performance_schema
phpmyadmin
test
Server: 127.0.0.1 » Database: light_rail_management_system_team3
Structure
SQL
Search
Query
Export
Import
Operations
Privileges
Routines
More
Show query box
[Edit inline] [Edit] [Create PHP code]

MySQL returned an empty result set (i.e. zero rows). (Query took 0.0597 seconds.)

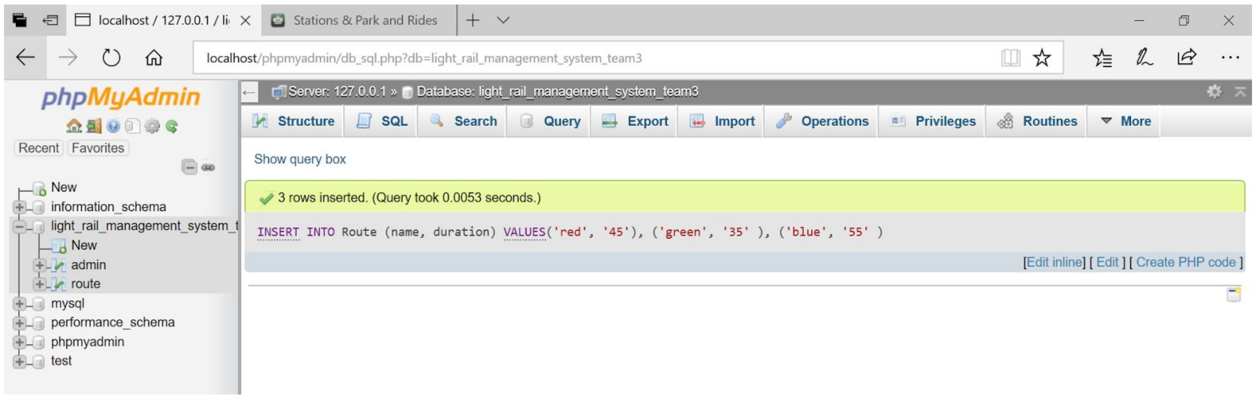
```
CREATE TABLE IF NOT EXISTS Route( name VARCHAR(256) NOT NULL, Duration int(11) NOT NULL, PRIMARY KEY(name))
```

3. INSERT INTO Route (name, duration)

VALUES('red', '45'),

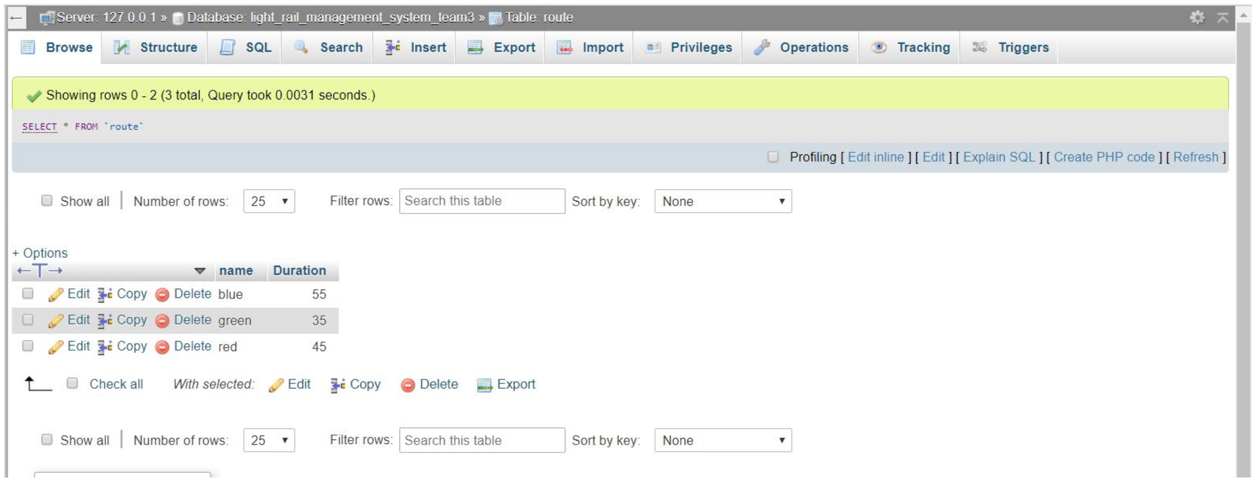
('green', '35'),

('blue', '55');


localhost / 127.0.0.1 / li
Stations & Park and Rides
localhost/phpmyadmin/db_sql.php?db=light_rail_management_system_team3
phpMyAdmin
Recent
Favorites
New
information_schema
light_rail_management_system_t
New
admin
route
mysql
performance_schema
phpmyadmin
test
Server: 127.0.0.1 » Database: light_rail_management_system_team3
Structure
SQL
Search
Query
Export
Import
Operations
Privileges
Routines
More
Show query box
[Edit inline] [Edit] [Create PHP code]

3 rows inserted. (Query took 0.0053 seconds.)

```
INSERT INTO Route (name, duration) VALUES('red', '45'), ('green', '35' ), ('blue', '55' )
```


Server: 127.0.0.1 » Database: light_rail_management_system_team3 » Table: route
Browse
Structure
SQL
Search
Insert
Export
Import
Privileges
Operations
Tracking
Triggers

✓ Showing rows 0 - 2 (3 total, Query took 0.0031 seconds.)

```
SELECT * FROM `route`
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None
+ Options

	name	Duration
Edit Copy Delete	blue	55
Edit Copy Delete	green	35
Edit Copy Delete	red	45

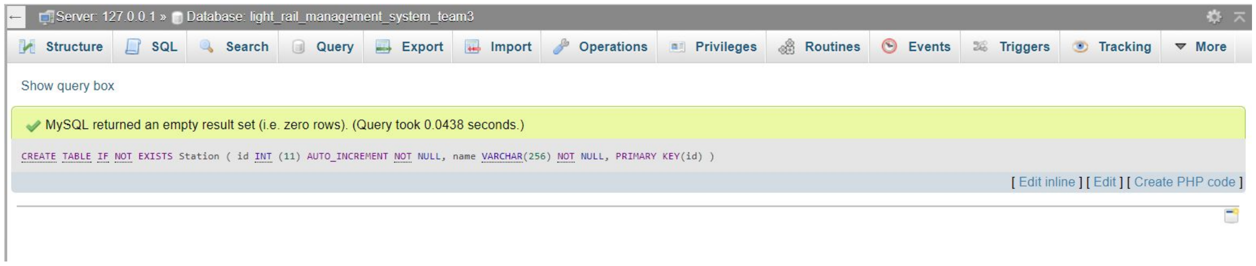
Check all With selected: Edit Copy Delete Export
Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

6. CREATE TABLE IF NOT EXISTS Station (

id INT (11) AUTO_INCREMENT NOT NULL,

name VARCHAR(256) NOT NULL,

PRIMARY KEY(id));


Server: 127.0.0.1 » Database: light_rail_management_system_team3
Structure
SQL
Search
Query
Export
Import
Operations
Privileges
Routines
Events
Triggers
Tracking
More
Show query box

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0438 seconds.)

```
CREATE TABLE IF NOT EXISTS Station ( id INT (11) AUTO_INCREMENT NOT NULL, name VARCHAR(256) NOT NULL, PRIMARY KEY(id) )
```

[Edit inline] [Edit] [Create PHP code]

7. INSERT INTO station(name)

VALUES ('South Blvd'),

('Sharon Rd West'),

('Arrowood'),

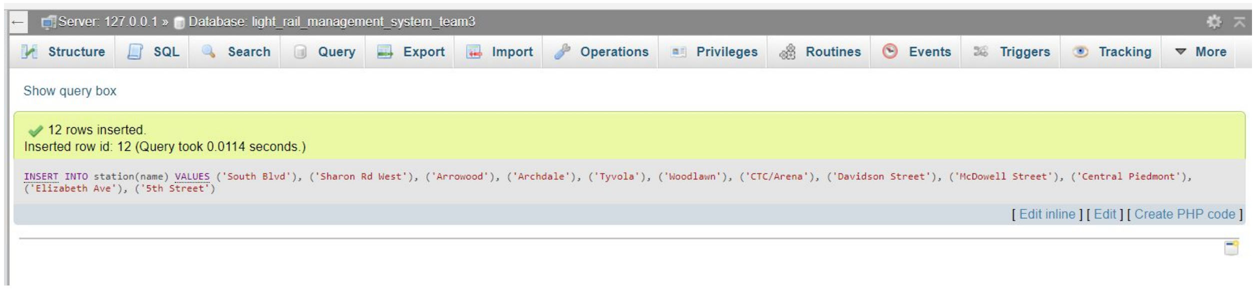
('Archdale'),

('Tyvola'),

('Woodlawn'),

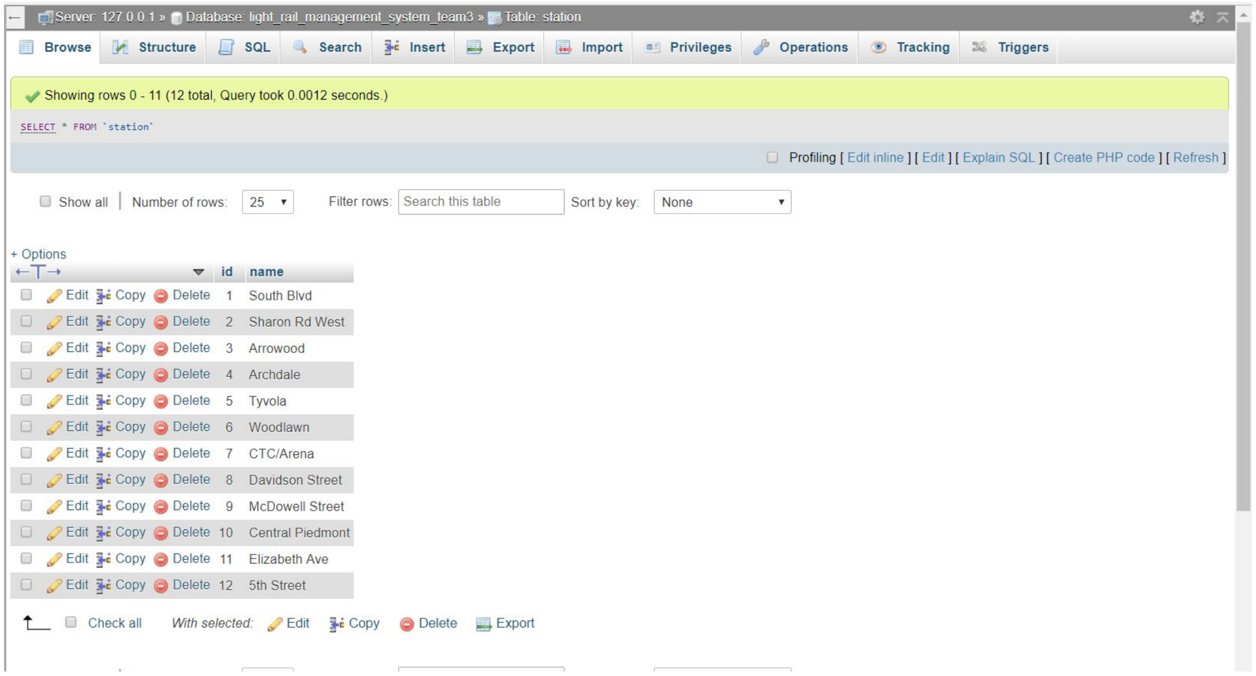
('CTC/Arena'),

```
('Davidson Street'),
('McDowell Street'),
('Central Piedmont'),
('Elizabeth Ave'),
('5th Street');
```



12 rows inserted.
Inserted row id: 12 (Query took 0.0114 seconds.)

```
INSERT INTO station(name) VALUES ('South Blvd'), ('Sharon Rd West'), ('Arrowood'), ('Archdale'), ('Tyvola'), ('Woodlawn'), ('CTC/Arena'), ('Davidson Street'), ('McDowell Street'), ('Central Piedmont'),
('Elizabeth Ave'), ('5th Street')
```



Showing rows 0 - 11 (12 total, Query took 0.0012 seconds.)

```
SELECT * FROM `station`
```

Number of rows: 25 · Sort by key: None

id	name
1	South Blvd
2	Sharon Rd West
3	Arrowood
4	Archdale
5	Tyvola
6	Woodlawn
7	CTC/Arena
8	Davidson Street
9	McDowell Street
10	Central Piedmont
11	Elizabeth Ave
12	5th Street

```
8. CREATE TABLE IF NOT EXISTS Train(
id INT(11) NOT NULL,
```

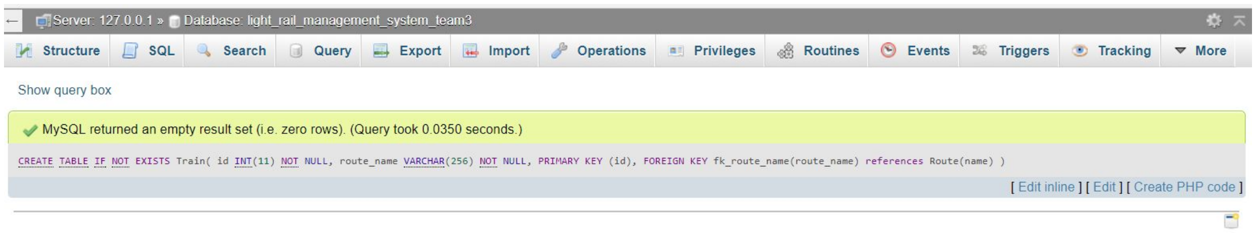
route_name VARCHAR(256) NOT NULL,

PRIMARY KEY (id),

FOREIGN KEY fk_route_name(route_name)

references Route(name)

);


Server: 127.0.0.1 » Database: light_rail_management_system_team3
Structure
SQL
Search
Query
Export
Import
Operations
Privileges
Routines
Events
Triggers
Tracking
More
Show query box
[Edit inline] [Edit] [Create PHP code]

MySQL returned an empty result set (i.e. zero rows). (Query took 0.0350 seconds.)

```
CREATE TABLE IF NOT EXISTS Train( id INT(11) NOT NULL, route_name VARCHAR(256) NOT NULL, PRIMARY KEY (id), FOREIGN KEY fk_route_name(route_name) references Route(name) )
```

9. INSERT into Train (id, route_name)

VALUES ('111', 'red'),

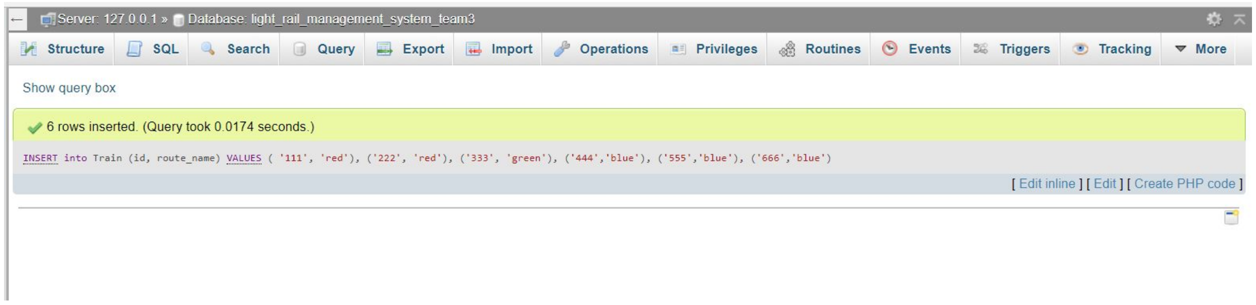
('222', 'red'),

('333', 'green'),

('444','blue'),

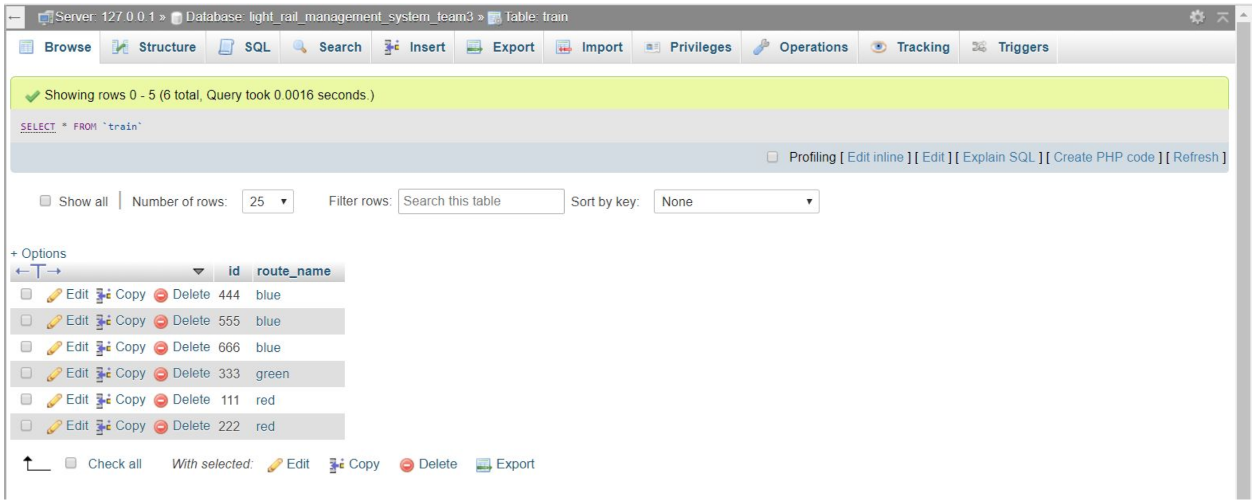
('555','blue'),

('666','blue');


Server: 127.0.0.1 » Database: light_rail_management_system_team3
Structure
SQL
Search
Query
Export
Import
Operations
Privileges
Routines
Events
Triggers
Tracking
More
Show query box
[Edit inline] [Edit] [Create PHP code]

6 rows inserted. (Query took 0.0174 seconds.)

```
INSERT into Train (id, route_name) VALUES ( '111', 'red'), ('222', 'red'), ('333', 'green'), ('444','blue'), ('555','blue'), ('666','blue')
```


Server: 127.0.0.1 » Database: light_rail_management_system_team3 » Table: train
Browse
Structure
SQL
Search
Insert
Export
Import
Privileges
Operations
Tracking
Triggers

Showing rows 0 - 5 (6 total, Query took 0.0016 seconds.)

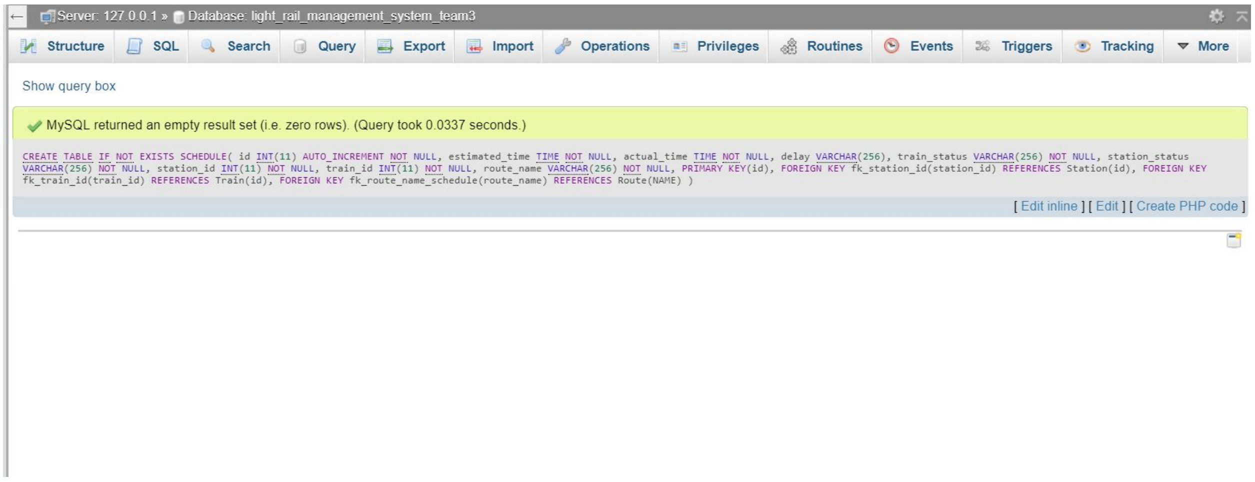
```
SELECT * FROM `train`
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None
+ Options

id	route_name
444	blue
555	blue
666	blue
333	green
111	red
222	red

Check all With selected: Edit Copy Delete Export

```
10. CREATE TABLE IF NOT EXISTS SCHEDULE(
        id INT(11) AUTO_INCREMENT NOT NULL,
        estimated_time TIME NOT NULL,
        actual_time TIME NOT NULL,
        delay VARCHAR(256),
        train_status VARCHAR(256) NOT NULL,
        station_status VARCHAR(256) NOT NULL,
        station_id INT(11) NOT NULL,
        train_id INT(11) NOT NULL,
        route_name VARCHAR(256) NOT NULL,
        PRIMARY KEY(id), FOREIGN KEY fk_station_id(station_id)
        REFERENCES Station(id),
        FOREIGN KEY fk_train_id(train_id) REFERENCES Train(id),
        FOREIGN KEY fk_route_name_schedule(route_name) REFERENCES
Route(NAME)
);
```

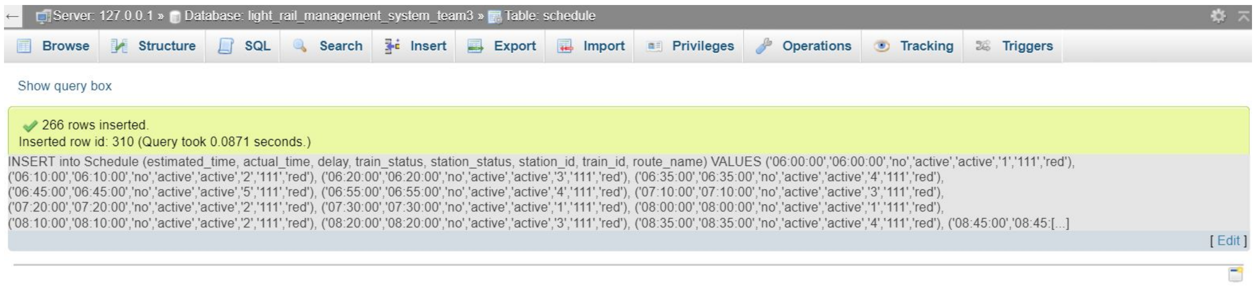


```
MySQL returned an empty result set (i.e. zero rows). (Query took 0.0337 seconds.)

CREATE TABLE IF NOT EXISTS SCHEDULE( id INT(11) AUTO_INCREMENT NOT NULL, estimated_time TIME NOT NULL, actual_time TIME NOT NULL, delay VARCHAR(256), train_status VARCHAR(256) NOT NULL, station_status
VARCHAR(256) NOT NULL, station_id INT(11) NOT NULL, train_id INT(11) NOT NULL, route_name VARCHAR(256) NOT NULL, PRIMARY KEY(id), FOREIGN KEY fk_station_id(station_id) REFERENCES Station(id), FOREIGN KEY
fk_train_id(train_id) REFERENCES Train(id), FOREIGN KEY fk_route_name_schedule(route_name) REFERENCES Route(NAME) )
```

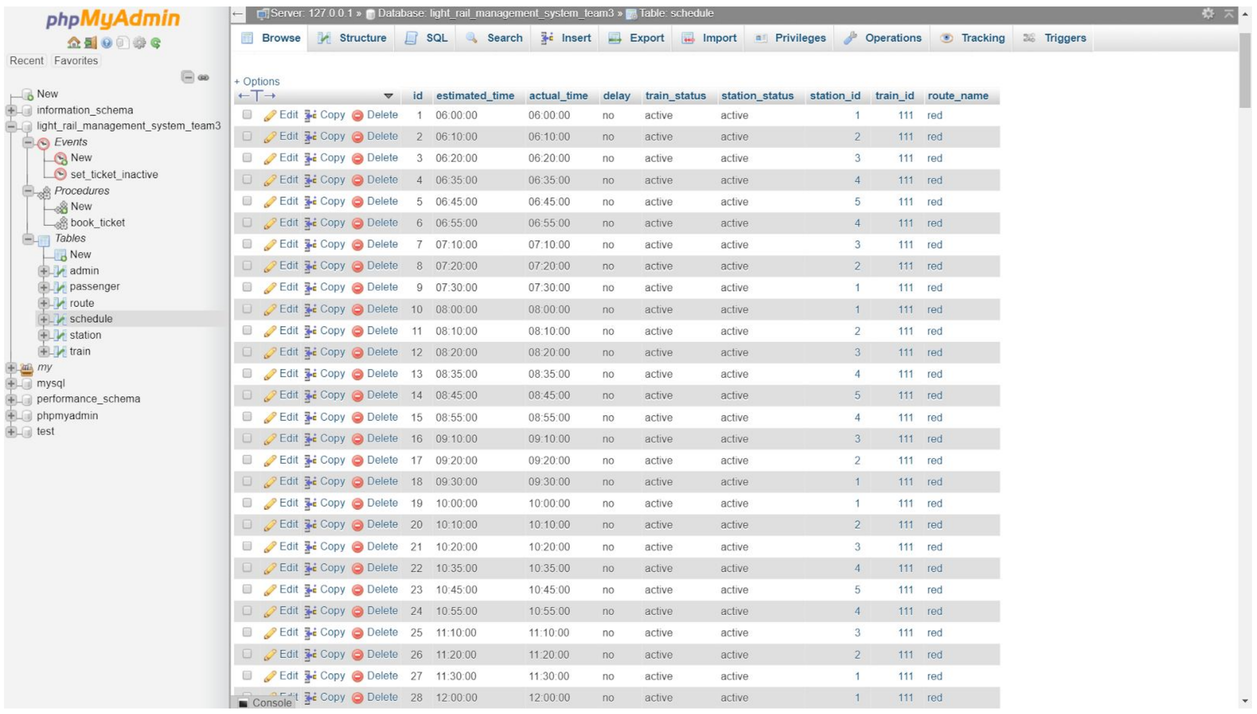
11. INSERT into Schedule (estimated_time, actual_time, delay, train_status, station_status, station_id, train_id, route_name)

VALUES

('06:00:00','06:00:00','no','active','active','1','111','red');

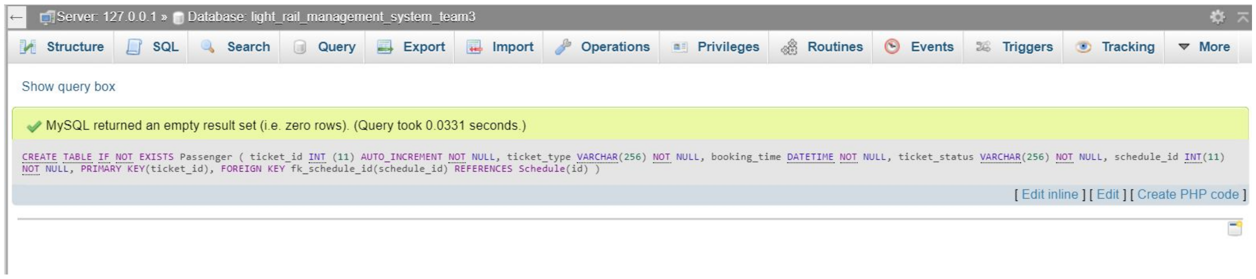


```
266 rows inserted.
Inserted row id: 310 (Query took 0.0871 seconds.)
INSERT into Schedule (estimated_time, actual_time, delay, train_status, station_status, station_id, train_id, route_name) VALUES ('06:00:00','06:00:00','no','active','active','1','111','red'),
('06:10:00','06:10:00','no','active','active','2','111','red'), ('06:20:00','06:20:00','no','active','active','3','111','red'), ('06:35:00','06:35:00','no','active','active','4','111','red'),
('06:45:00','06:45:00','no','active','active','5','111','red'), ('06:55:00','06:55:00','no','active','active','4','111','red'), ('07:10:00','07:10:00','no','active','active','3','111','red'),
('07:20:00','07:20:00','no','active','active','2','111','red'), ('07:30:00','07:30:00','no','active','active','1','111','red'), ('08:00:00','08:00:00','no','active','active','1','111','red'),
('08:10:00','08:10:00','no','active','active','2','111','red'), ('08:20:00','08:20:00','no','active','active','3','111','red'), ('08:35:00','08:35:00','no','active','active','4','111','red'), ('08:45:00','08:45:[...]
```



id	estimated_time	actual_time	delay	train_status	station_status	station_id	train_id	route_name
1	06:00:00	06:00:00	no	active	active	1	111	red
2	06:10:00	06:10:00	no	active	active	2	111	red
3	06:20:00	06:20:00	no	active	active	3	111	red
4	06:35:00	06:35:00	no	active	active	4	111	red
5	06:45:00	06:45:00	no	active	active	5	111	red
6	06:55:00	06:55:00	no	active	active	4	111	red
7	07:10:00	07:10:00	no	active	active	3	111	red
8	07:20:00	07:20:00	no	active	active	2	111	red
9	07:30:00	07:30:00	no	active	active	1	111	red
10	08:00:00	08:00:00	no	active	active	1	111	red
11	08:10:00	08:10:00	no	active	active	2	111	red
12	08:20:00	08:20:00	no	active	active	3	111	red
13	08:35:00	08:35:00	no	active	active	4	111	red
14	08:45:00	08:45:00	no	active	active	5	111	red
15	08:55:00	08:55:00	no	active	active	4	111	red
16	09:10:00	09:10:00	no	active	active	3	111	red
17	09:20:00	09:20:00	no	active	active	2	111	red
18	09:30:00	09:30:00	no	active	active	1	111	red
19	10:00:00	10:00:00	no	active	active	1	111	red
20	10:10:00	10:10:00	no	active	active	2	111	red
21	10:20:00	10:20:00	no	active	active	3	111	red
22	10:35:00	10:35:00	no	active	active	4	111	red
23	10:45:00	10:45:00	no	active	active	5	111	red
24	10:55:00	10:55:00	no	active	active	4	111	red
25	11:10:00	11:10:00	no	active	active	3	111	red
26	11:20:00	11:20:00	no	active	active	2	111	red
27	11:30:00	11:30:00	no	active	active	1	111	red
28	12:00:00	12:00:00	no	active	active	1	111	red

```
12. CREATE TABLE IF NOT EXISTS Passenger (
        ticket_id INT (11) AUTO_INCREMENT NOT NULL,
        ticket_type VARCHAR(256) NOT NULL,
    booking_time DATETIME NOT NULL,
        ticket_status VARCHAR(256) NOT NULL,
    schedule_id INT(11) NOT NULL,
        PRIMARY KEY(ticket_id),
    FOREIGN KEY fk_schedule_id(schedule_id)
    REFERENCES Schedule(id)
);
```



Server: 127.0.0.1 » Database: light_rail_management_system_team3
Structure SQL Search Query Export Import Operations Privileges Routines Events Triggers Tracking More
Show query box

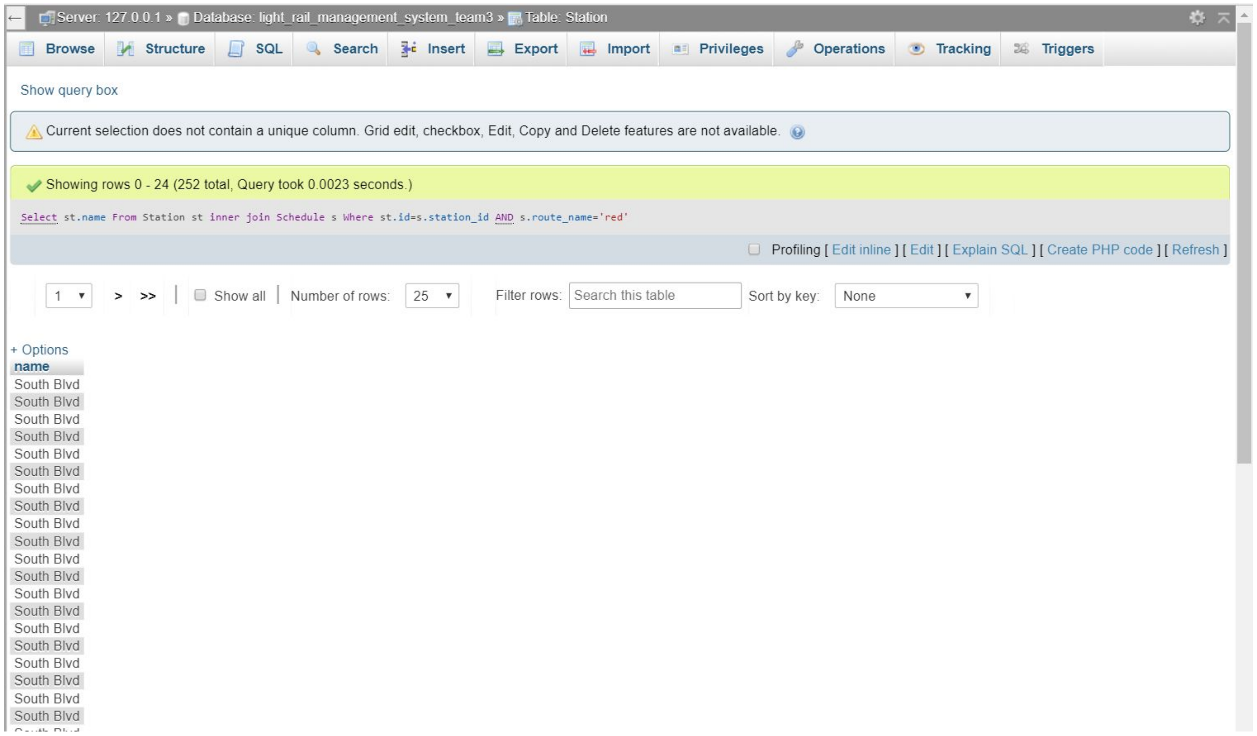
MySQL returned an empty result set (i.e. zero rows). (Query took 0.0331 seconds.)

```
CREATE TABLE IF NOT EXISTS Passenger ( ticket_id INT (11) AUTO_INCREMENT NOT NULL, ticket_type VARCHAR(256) NOT NULL, booking_time DATETIME NOT NULL, ticket_status VARCHAR(256) NOT NULL, schedule_id INT(11)
NOT NULL, PRIMARY KEY(ticket_id), FOREIGN KEY fk_schedule_id(schedule_id) REFERENCES Schedule(id) )
```

[Edit inline] [Edit] [Create PHP code]

SQL QUERIES

- Select sation_name
  From Station st inner join Schedule s
  Where st.id=s.station_id AND s.route_name='red'



Server: 127.0.0.1 » Database: light_rail_management_system_team3 » Table: Station
Browse Structure SQL Search Insert Export Import Privileges Operations Tracking Triggers
Show query box

Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available.

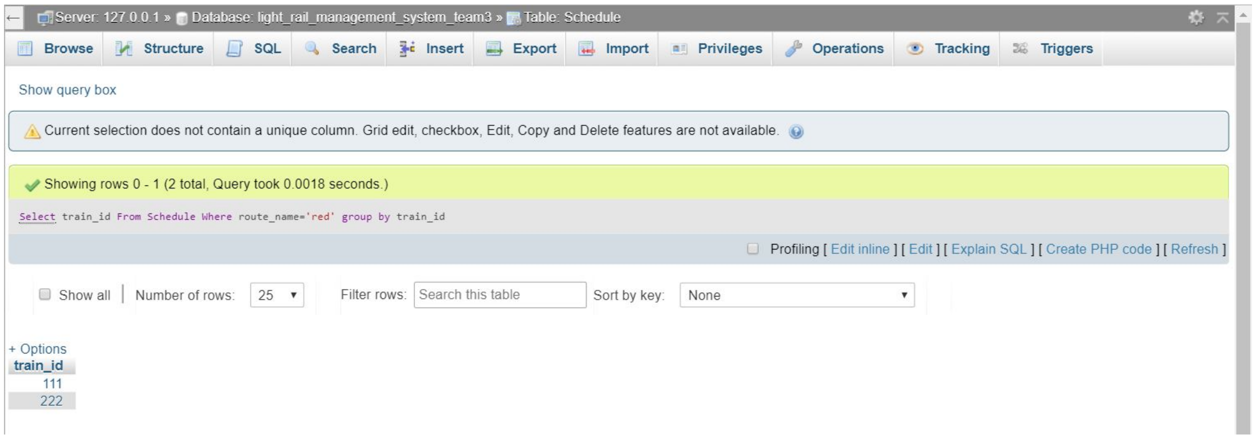
Showing rows 0 - 24 (252 total, Query took 0.0023 seconds.)

```
Select st.name From Station st inner join Schedule s Where st.id=s.station_id AND s.route_name='red'
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
1 > >> Show all Number of rows: 25 Filter rows: Search this table Sort by key: None
+ Options

```
name
South Blvd
South Blvd
South Blvd
South Blvd
South Blvd
South Blvd
South Blvd
South Blvd
South Blvd
South Blvd
South Blvd
South Blvd
South Blvd
South Blvd
South Blvd
South Blvd
South Blvd
South Blvd
South Blvd
South Blvd
```

- Select train_id
  From Schedule
  Where route_name='red' group by train_id;



Server: 127.0.0.1 » Database: light_rail_management_system_team3 » Table: Schedule
Browse
Structure
SQL
Search
Insert
Export
Import
Privileges
Operations
Tracking
Triggers
Show query box

Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available.

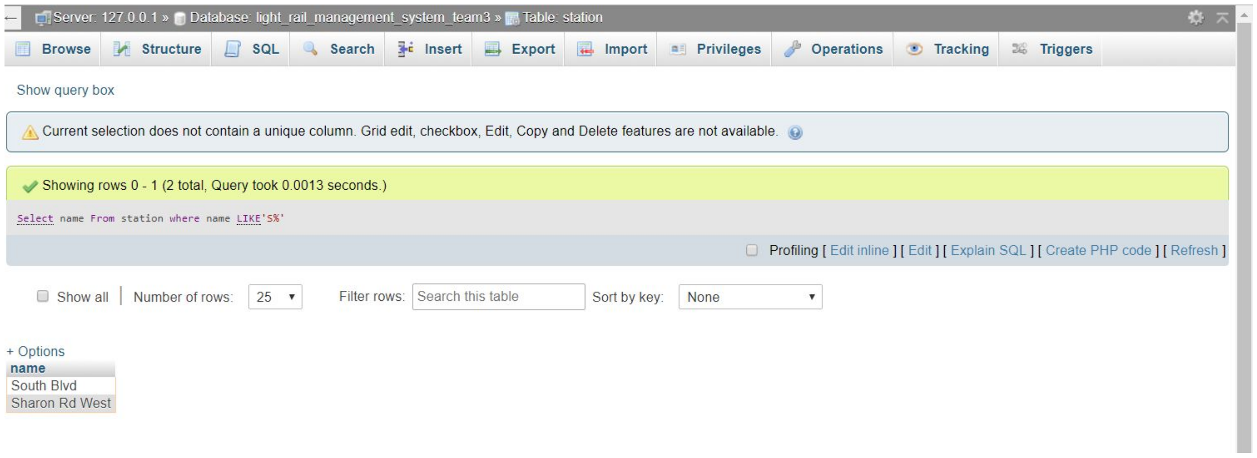
Showing rows 0 - 1 (2 total, Query took 0.0018 seconds.)

```
Select train_id From Schedule Where route_name='red' group by train_id
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
Show all
Number of rows: 25
Filter rows: Search this table
Sort by key: None
+ Options

train_id
111
222

- Select name
  From station
  where name LIKE 'S%';



Server: 127.0.0.1 » Database: light_rail_management_system_team3 » Table: station
Browse
Structure
SQL
Search
Insert
Export
Import
Privileges
Operations
Tracking
Triggers
Show query box

Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available.

Showing rows 0 - 1 (2 total, Query took 0.0013 seconds.)

```
Select name From station where name LIKE'S%'
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
Show all
Number of rows: 25
Filter rows: Search this table
Sort by key: None
+ Options

name
South Blvd
Sharon Rd West

VIEWS AND STORED PROCEDURES

Views

1. **View: Passenger_route_station**
   Goal: This view will display for the passengers, the name of the route and all the stations it goes to, along with the schedule for all the trains on that route. Passenger can get information of timing of trains running on a route.

2. **Admin_train_delay**
   Goal: The goal of this view is to display for admin the trains which have got delayed and their current time so that he can make changes to the actual time of arrival depending upon the delays.

Stored Procedure

1. **Book_ticket [ticket_Status]**
   Parameters: schedule_id (IN)
   Goal: The goal of this stored procedure is to book a ticket for a passenger.

2. **Admin_Delay**

   Parameters: delay time (IN) , train_id (IN)
   Goal: The goal of this stored procedure is to enable admin to add delay time for a particular train depending upon the train id.

3. **Call_station**
   Parameters: station_name (IN)
   Goal: The goal of this stored procedure is to display for a particular station its route name, all the upcoming trains(their train_ids) and the upcoming stations and its respective schedule for a particular station selected by the passenger.

4. **Call_route**
   Parameters: route_name (IN)
   Goal: The goal of this stored procedure is to display route name, train id and station name with its respective schedule for a particular route selected by the passenger.

Events

1. **Event Name: Set Train Inactive**
   Event-type: Recurring event
   Goal: After the specified time at night(as specified in Schedule), some trains will be out of service, that is, their status will become inactive

2. **Event name: Set Ticket Inactive**
   Event-type: Recurring event
   Goal: Setting all the active ticket-status(since currently we are considering only day-tickets) to inactive at 12.00 AM.

3. **Event name: Set Train Status** [Update the schedule table for all the delayed trains to run back on time on next day]
   Event-type: Recurring event
   Goal: Setting all the delayed trains back to normal on the next day and updating the status of all the trains on scheduled time.

Sprint 3

REQUIREMENTS

Story ID	**Story description**
US1	As an Admin I want to register into the system.
US2	As an Admin I want to login into the system.
US3	As an Admin I want to add station to the system.
US4	As an Admin I want to add trains for the LRMS.
US5	As an Admin I want to add a list of stations for routes.
US6	As an Admin I want to add trains for routes.
US7	As an Admin I want to add schedule (where schedule includes information about route, stations, trains, and the timings) for a train belonging to a route.
US8	As a Passenger I want to know all of the routes in the LR system and all the stations that a particular route contains.
US9	As a Passenger I want to check(the schedule) timing of different trains for any particular station.
US10	As a Passenger I want to check the schedule of a particular train so that I can get information about arrival time of the train across different stations on a particular route.
US11	As a passenger, I want to view ticket-types, which includes two-hour ticket, one-day ticket, weekly ticket, and monthly ticket along with their price.
US12	As a passenger, I want to check all incoming and outgoing trains from a particular station.

	instead of incoming and outgoing since these words state the purpose of this User Story better.
US13	As a Passenger I want to book a ticket.
US14	As an Admin, I want to update the delay for active trains as well as for particular station.
US15	As a passenger, I want to know remaining time(in minutes/hours) of a train, incoming on a particular route.
US16	As an admin I want to add an alert about changes in schedules, if there is any event (accident, detours, delays, cancellation).

CONCEPTUAL DESIGN

Entity: **Admin**
Attributes:

email_id

name (composite)

first_name

last_name

password

Entity: **Station**

Attributes:

id

name

Entity: **Route**

Attributes:

name

duration

Note: Duration is in Minutes for a Route

Entity: **Schedule**

Attributes:

id

estimated_time

actual_time

delay (derived)

train_status

station_status

Alert Message

Entity : Train

id

Entity : Passenger

ticket_id

ticket_type

booking_time

ending_time (derived)

ticket_status

Entity : Ticket

ticket_type

price

Relationship: **Admin** adds **Station**
Cardinality: One to Many
Participation:
Admin has total participation
Station has total participation
Assumptions: There will be only 1 admin for the system.

Relationship: **Admin** assigns **Schedule**
Cardinality: One to Many
Participation:
Admin has total participation
Schedule has total participation

Assumption: There will be only 1 admin for the system.

Relationship: **Admin** adds **Route**
Cardinality: One to Many
Participation:
Admin has total participation
Route has total participation

Assumption: There will be only 1 admin for the system.

Relationship: **Admin** adds **Train**
Cardinality: One to Many
Participation:
Admin has total participation
Train has total participation

Assumption: There will be only 1 admin for the system.

Relationship: **Route** has **Station**
Cardinality: One to Many
Participation:

Route has total participation
Station has total participation

Assumptions: Here, we consider that the stations which are not functional yet do not show up on our system, so we don't have any station which does not belongs to any route. Therefore, we have a total participation for Station.

Relationship: **Route** has **Train**
Cardinality: One to Many
Participation:

Route has total participation
Train has total participation

Relationship: **Train** has **Schedule**
Cardinality: One to One
Participation:

Train has total participation
Schedule has total participation

Assumption: Right now we are assuming that every train in our system is up and functional, so every train must have a schedule. Therefore, we have a total participation for train.

Relationship: **Passenger** checks **Route**
Cardinality: Many to Many
Participation:

Passenger has partial participation
Route has partial participation

Relationship: **Passenger** checks **Schedule**
Cardinality: Many to Many
Participation:

Passenger has partial participation
Schedule has partial participation

Relationship: **Passenger** books **Ticket**
Cardinality: One to One
Participation:
Passenger has total participation
Ticket has total participation

LOGICAL DESIGN WITH HIGHEST NORMAL FORMS AND INDEXES

Table: Admin
Columns:
email_id
first_name
last_name
password

Justification of primary key: Earlier in our admin table, we had two candidate keys i.e. ssn and email_id, which can independently determine non-prime attributes. So, to normalize the table to second normal form, we removed one of the unique keys which is the ssn. So, now primary key of the Admin table is email_id.

Highest normalization level: 4NF

Indexes: admin_index
Index #1:Type: Clustered
Columns: <email_id>
Justification: It is very likely that a user might search for admin using the email_id in order to contact. It is clustered, because the row is stored on disk in the same order as index.

```
CREATE INDEX admin_index
On Admin(email_id);
```

Table: Route
Columns:
name
duration

Note: duration is in minutes for a particular route.

Highest normalization level: 4NF

Indexes: route_index
Index #1: type: clustered
Columns: <name>
Justification: We are retaining 'name' as the index since, the highest probability is that a user will look into the Route table by searching the route-name. It is clustered, since the rows are stored on disk in the same order as index.

CREATE INDEX route_index
On Route(name);

Table: Train
Columns:
id
route_name [Foreign Key; references name of Route]
Justification: Since the relationship is one to many between Train and Route hence we have included the primary key of Route table into Train table.

Highest normalization level: 4NF

Indexes: train_index
Index #1: type (non-clustered)
Columns: <route_name>
Justification: Chances are that the user will want to look what trains are there for a specific route. Also, it is non-clustered since there are more than one train for a particular route so, rows are not stored on disk in the same order as index.

CREATE INDEX train_index
On Train(route_name);

Table: Station
Columns:
id

name

Highest normalization level: 4NF

Indexes: station_index

Index #1: type: clustered

Columns: <name>

Justification: Chances are that the user will want to look what stations are there in the system by searching through the station name. Also, it is clustered since the rows are stored on disk in the same order as index.

```
CREATE INDEX station_index
On Station(name);
```

Table: Schedule

Columns:

- id
- train_status
- station_status
- estimated_time
- actual_time
- station_id [Foreign Key, references id of Station]
- train_id [Foreign Key, references id of Train]
- alert_message

Justification: In Schedule table, we have shown Many-to-Many relationship between Train and Station. Therefore, we have used the cross reference approach to relate the primary keys of corresponding tables in the Schedule table.

Highest normalization level: 4NF

Indexes: schedule_index

Index #1: type: non-clustered

Columns: <actual_time, station_id>

Justification: There is a high chance that the user will want to search that what trains are coming to a particular station at a particular time. Also, it is non-clustered since the rows are not stored on disk in the same order as index.

CREATE INDEX schedule_index
On Schedule(actual_time, station_id);

Table: Ticket

Columns:

ticket_type
price

Highest normalization level: 4NF

Indexes: ticket_index

Index #1: type: clustered

Columns: <ticket_type>

Justification: There is a high chance that the user will want to search that what ticket types are present for him/her as an option to travel in the system. Also, it is clustered since the rows are stored on disk in the same order as index.

CREATE INDEX ticket_index
On Ticket(ticket_type);

Table: Passenger

Columns:

ticket_id
booking_time
ending_time
ticket_status
schedule_id [Foreign Key, references id of Schedule]
ticket_type[Foreign Key, references ticket_type of Ticket]

Justification: In passenger table, each passenger selects one train from the provided schedule, therefore we used schedule_id as the foreign key to depict the relationship between these entities.

Highest normalization level: 2NF
Justification (if below 4NF): We did not normalize further because, it makes sense to consider the ending_time of the ticket and store it, which will not change ever.

Indexes: passenger_index

Index #1: type: non-clustered

Columns: <ending_time, ticket_type>

Justification: There is a high chance that the user will want to search that what ticket type is booked and until what time it is valid. Also, it is non-clustered since the rows are not stored on disk in the same order as index.

```
CREATE INDEX passenger_index
On Passenger(ending_time, ticket_type);
```