**Final Project Report**

**[YOUR PROJECT TITLE]**

CSC[4710|6710] Database Systems

Spring 2021

Ruilin Wang

# Introduction

The interest of this DBMS is to benefit the public, as we seek to increase our knowledge and understanding of programming through this project. This is a project about the railway system, use New York and Atlanta railway station as the prototype, to introduce the railways, stations, ticket prices, different kind of passengers and etc. Although many people today rely on cars to do transportation, the majority of the population are still using public transportations such as bus, subway, and railway. As one of the riders at Atlanta stations, and with fit of the majority of our population, a DBMS for the railway tracking system is necessary.

# Requirements Analysis

In this DBMS, there are essential elements such as location of stations, passenger info(first name, middle name, last name, age, sex, email, phone number, identity(with or without pass, faculty or not), with discount or not), names of railway lines, number of compartments. Customer numbers are zero to unlimited, stations numbers are two to unlimited, railway line numbers are one to unlimited, gender should be male, female or prefer not to tell, age should be zero to 123(if there is a high world record please let me know), compartment numbers are two to 10, each name is one to 200 case.

## **Data Requirements**

* Size of varchar for names
* Size of INT and BIGINT for containing values
* Size of double for containing ticket prices

## **Functional Requirements**

* Data capacity due to number users in subway
* Data accuracy for the location and train
* Data update in time for current train location

# 

# Conceptual Design

Map your requirements and show your final ER design with min/max notation, key attributes and specialization/generalization/category relationships. You can add notes if you deviate from your data requirements.

Diagram

Description automatically generated

# Relational Design

Map your EER model to relational model. Make sure you follow the steps in the transformation algorithm. If you deviated from the EER model, you can add notes and your rationale.

Diagram

Description automatically generated

# Data Dictionary

Provide your data dictionary. Below you may find an example for employee and department we have seen in our lectures/discussions.

Table

Description automatically generated

# Implementation

Provide your source code for creating your tables one by one. Add them as code blocks. An example can be found below. Do not include insert statements. Include your queries and optionally your triggers and views. If you have implemented an application, briefly describe what your application can do and provide a simple use case/demo using screenshots.

|  |
| --- |
| # Create table statements for table “table\_name\_1”  create database railway;  use railway;  CREATE TABLE Passenger (  FName Varchar(80) not null,  MName varchar(80) not null,  LName varchar(80) not null,  Email varchar(80) null,  PNumber bigint not null,  Age int not null,  Gender varchar(30) not null,    primary key(PNumber)    );    CREATE TABLE Faculty (  FName varchar(80) not null,  MName varchar(80) not null,  LName varchar(80) not null,  Email varchar(80) null,  PNumber bigint not null,  Age int not null,  Gender varchar(30) not null,  SSN bigint not null,  Salary bigint not null,  primary key(SSN),  foreign key(PNumber) references Passenger(PNumber)  );    CREATE TABLE Customer (  FName varchar(80) not null,  MName varchar(80) not null,  LName varchar(80) not null,  Email varchar(80) not null,  PNumber bigint not null,  Age int not null,  Gender varchar(30) not null,  primary key(Email),  foreign key(PNumber) references Passenger(PNumber)  );    CREATE TABLE Station (  SName varchar(200) null,  Location varchar(200) null,  Population INT null,  LineName varchar(200) null  );    CREATE TABLE RailwayLine (  LineName varchar(200) null,  TName varchar(80) null,  SName varchar(200) not null,  Location varchar(200) not null,  UNumber BIGINT not null,  primary key(UNumber)  );    CREATE TABLE Train (  TName varchar(80) not null,  SName varchar(80) null,  LineName varchar(80) not null,  Speed int not null,  Compartment varchar(80) null,  Location varchar(80) null,  primary key(TName),  UNumber BIGINT not null,  foreign key(UNumber) references RailwayLine (UNumber)  );    CREATE TABLE Ticket (  OneWay double (20,2) null,  RoundT double (20,2) null,  3Hours double (20,2) null,  OneDay double (20,2) null,  Pass double (20,2) null,  Free double (1,0) null,  PNumber bigint not null,  foreign key(PNumber) references Passenger(PNumber)  ); |

# Conclusion and Summary

This project is a basic frame for a railway system database. I use the Atlanta and New York subway as sample to created this, it contains stations, railway lines, estimate population, location for the station and train. It allows administrators to check, update, modify all the information in this system, and display the requires. It also gives the option to store some frequent used views so administrators can look up whenever they need.