a) Project Outline and Database Outline - Updated Version:

AnyTrak sells $100,000 worth of train tickets every year. Instead of paper tracking, they will use a database driven website in order to track train routes and stations. They are having trouble keeping track of which Routes a certain Trains is on and what Stations certain Routes travel through. With the use of a database, they will be able to more efficiently look up and update Routes. Each Trains will also have one and only one main Stations that they reside at. Trains must have one main Stations while every station doesn't necessarily need a main Train. Each Routes only has one Trains running it, but a Trains can run one or multiple Routes (1:M). Routes can possibly have no Trains running it and Trains can possibly run no Routes. Routes will have multiple Stations that they go through and Stations can have multiple Routes that go through it (M:M).

Entities:

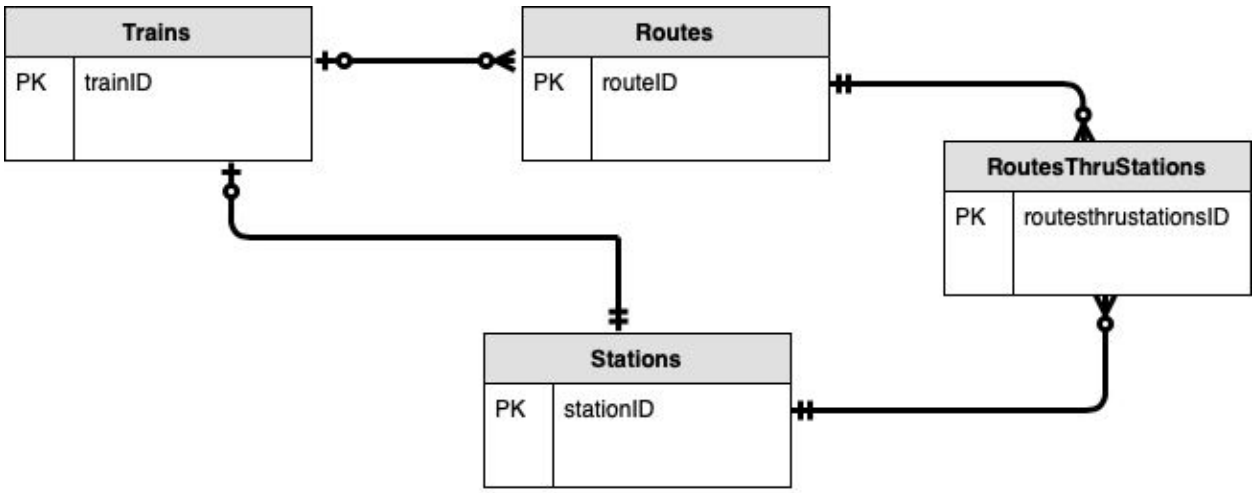
- **Trains: The physical train itself. Will be assigned route(s) to travel on.**
  - trainID: int, auto_increment, unique, not NULL, PK
  - stationID: FK
  - conductorfirstname: varchar
  - conductorlastname: varchar
  - model: varchar, not NULL
  - cost: int, not NULL
  - capacity: int, not NULL
  - Relationship: 1: M relationship between Trains and Routes is implemented with trainID as a FK inside of Routes
  - Relationship: 1:1 relationship between Trains and Stations with stationID as FK in Trains

- **Stations: House trains and are the start and ending point of every train's journey.**
  - stationID: int, auto_increment, unique, not NULL, PK
  - stationname: varchar, not NULL, unique
  - address: varchar, not NULL
  - state: varchar, not NULL
  - city: varchar, not NULL
  - zipcode: int, not NULL
  - Relationship: 1:M relationship between Stations and RoutesThruStations with StationID as FK in RoutesThruStations (Part of M:M between Routes and Stations)
  - Relationship: 1:1 relationship between Trains and Stations with StationID as FK in Trains

- **Routes: The system by which trains navigate through to arrive at specific stations.**
    - routeID: int, auto_increment, unique, not NULL, PK
    - trainID: FK
    - routename: varchar, not NULL, unique
    - ticketprice: int
    - Relationship: 1: M relationship between Trains and Routes is implemented with trainID as a FK inside of Routes
    - Relationship: 1:M relationship between Routes and RoutesThruStations with routeID as FK in RoutesThruStations (Part of M:M between Routes and Stations)
- **RoutesThruStations: What route leads to what station.**
    - routesthrustationsID: PK, auto_increment, not NULL, unique
    - routeID: FK
    - stationID: FK
    - travelduration: int
    - milestraveled: int
    - Relationship: 1:M relationship between Routes and RoutesThruStations with routeID as FK in RoutesThruStations (Part of M:M between Routes and Stations)
    - Relationship: 1:M relationship between Stations and RoutesThruStations with stationID as FK in RoutesThruStations (Part of M:M between Routes and Stations)

---

b) Entity-Relationship Diagram



---

c) Schema