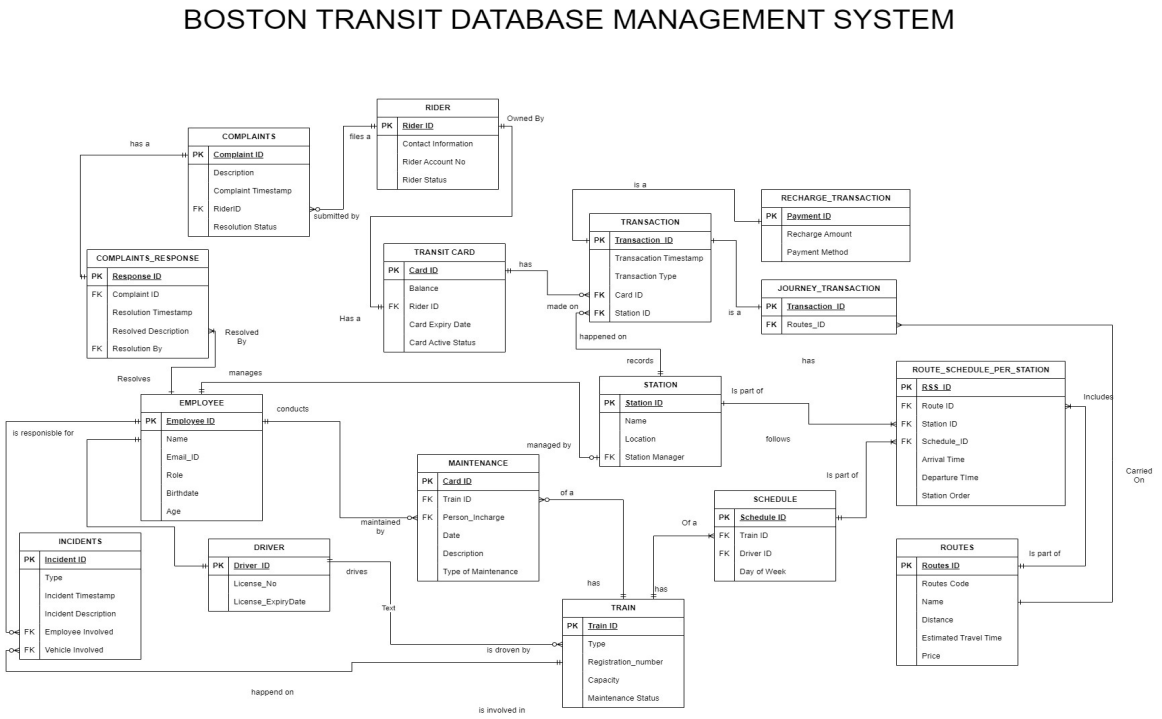
BOSTON TRANSIT DATABASE MANAGEMENT SYSTEM

This project aims to design and implement a comprehensive database system to manage the operations of the Massachusetts Bay Transportation Authority (MBTA) system in Boston. The scope of the project encompasses various aspects of public transportation, including personnel management, route planning, rider information, financial transactions, and incident tracking.

ENTITY RELATIONSHIP DIAGRAM:



ENTITIES:

EMPLOYEE:
Manages information about MBTA employees, including roles, contact details, and birthdates.

ROUTES:
Organizes transportation routes with details such as route codes, distances, estimated travel times, and prices.

RIDER:
Tracks individual riders using the system, assigns unique rider account numbers, and manages contact information and status.

TRANSIT_CARD:
Maintains details about transit cards, their balances, and transactions, including recharges and journey-related activities.

DRIVER:
Handles information related to drivers, including license details and expiration dates.

STATION:
Organizes data about various stations, including names, locations, and station managers.

TRAIN:
Manages details about different types of trains, including their types, registration numbers, capacities, and maintenance statuses.

SCHEDULE:
Specifies schedules for trains and drivers, indicating the day of the week for each schedule.

ROUTE_SCHEDULE_PER_STATION:
Connects routes, stations, and schedules, detailing arrival and departure times and station order.

COMPLAINTS:
Captures and manages rider complaints, tracking their resolution status.

COMPLAINTS_RESPONSE:
Records responses to rider complaints, triggering status updates upon resolution.

TRANSACTION:
Manages financial transactions, including card recharges and journey-related activities.

JOURNEY_TRANSACTION:
Connects journey-related transactions to specific routes.

RECHARGE_TRANSACTION:
Handles financial transactions related to card recharges.

INCIDENTS:
Records and tracks incidents such as technical issues, on-site or on-road problems, and security concerns.

Relationships between entities:

Employee - Train:
An employee can be assigned to operate a specific Train as a driver. Employees are responsible for the maintenance and inspection of Trains.

Employee - Stations:
Employees may work at transit stations, serving as station attendants or security personnel. Station employees ensure the safety and functionality of station facilities.

Incidents - Employee:
Employees may be involved in incidents, such as accidents or safety violations. Employees are responsible for reporting incidents and following incident response procedures.

Incidents - Train:
Incidents can involve Trains, such as accidents or breakdowns.
Trains may be temporarily taken out of service due to incidents.

Rider - Transit Card:
Each rider has one transit card, and each transit card is associated with one rider.

Payments - Transit Card:
Each transit card can have multiple payment transactions recorded on it.

Stations - Payments:
Each station can have multiple payment transactions recorded, indicating payments made at the station.

Stations - Routes:
Stations are typically located along specific transit routes.

Trains - Routes:
Each train is associated with a specific route. This relationship ensures that a train follows a particular path defined by its assigned route.

Routes - Schedule:
Routes have associated schedules that dictate when Trains operate on them. Schedule adherence is crucial for maintaining route efficiency.

Schedule - Trains:
Each schedule is linked to a specific train, indicating which train operates according to that Schedule.

Schedule - Stations:

The "Departure Station" and "Arrival Station" attributes in the "Schedule" entity are foreign keys that link to the "Stations" entity. This establishes the relationship between a schedule and the stations involved in a particular journey.

Complaints - Rider:
Riders may file complaints related to their experiences while using transit services. Complaints may include issues with employee behavior, Train conditions, or payment disputes.

Maintenance - Train:
Trains require regular maintenance and inspections to ensure their proper operation. Maintenance schedules are based on Train usage and mileage.

FEATURES

1. STORED PROCEDURES

STR_PROC_GET_COMPLAINT_STATUS

- **What it does:** Retrieves the current status of a specific complaint.
- **Parameters:** @RiderID, @ComplaintID, @ComplaintDescription
- **Output:** A string message in @OutputMessage that describes the status of the complaint.
- **Use:** Allows customer service to quickly inform riders about the progress or resolution of their complaints.

STR_PROC_ADD_COMPLAINT_RESPONSE

- **What it does:** Records a response to a specific complaint.
- **Parameters:** @ComplaintID, @ResponseTimestamp, @ResponseDescription, @ResolvedBy
- **Output:** No explicit output, but it updates the complaint record with a response.
- **Use:** Used by complaint handlers to log their actions on a complaint, which helps in maintaining accountability and service quality.

STR_PROC_GET_SCHEDULE_BY_STATION_ID

- **What it does**: Fetches the train schedule for a given station.
- **Parameters**: @StationID
- **Output**: A result set that includes train names, scheduled departure and arrival times, and days of operation.
- **Use**: Essential for both staff and riders to understand train timings and plan accordingly.

STR_PROC_GET_SCHEDULE_BETWEEN_TWO_STATIONS

- **What it does:** Retrieves the train schedule between two specified stations.
- **Parameters:** @StartStationID, @EndStationID
- **Output:** A result set with train names, start and end stations, and scheduled times.
- **Use:** Provides scheduling information for passengers planning a trip between two specific locations.

2. USER-DEFINED FUNCTIONS

dbo.CalculateAge

- **What it does:** Calculates the age of an individual based on their birthdate.
- **Parameters:** @Birthdate (DATE) - The birthdate of the individual.
- **Output:** Integer representing the calculated age.
- **Use:** Useful for age-based validations and displaying age without storing it in the database, ensuring that the information is always up to date.

dbo.GenerateRiderAccountNo

- **What it does**: Generates a unique rider account number by adding a prefix to an auto-incremented RIDER_ID.
- **Parameters**: @RiderID (BIGINT) - The unique identifier for the rider.
- **Output**: A string that is the new rider account number.
- **Use**: Ensures that each rider has a unique account number, which is essential for account management and identification in the system.

3. TRIGGERS

TGR_VALIDATE_RIDER_BEFORE_ADDING_COMPLAINT

- **What it does**: Ensures that a complaint can only be added if the rider exists in the database.
- **Use**: Prevents the insertion of invalid complaints, maintaining data integrity.

TGR_CHECK_RESOLVER_ROLE

- **What it does:** Validates that only employees with the 'Complaint Resolver' role can resolve complaints.
- **Use:** Enforces role-based permissions, ensuring that only authorized personnel can act on complaints.

TGR_INSERT_RIDER_ACCOUNT_NO

- **What it does:** Automatically generates and assigns a unique rider account number after a rider record is created.
- **Use:** Simplifies the process of account creation and ensures that each rider has a unique identifier.

TGR_UPDATE_COMPLAINT_STATUS

- **What it does:** Updates the status of a complaint to 'Resolved' after a response is added.
- **Use:** Automates the process of complaint resolution tracking, reducing manual updates.

4. VIEWS

VIEW_DETAILED_RIDER_INFO

- **What it does:** Consolidates comprehensive information about riders.
- **Fields involved:** Rider details, transit card information, transaction history.
- **Output:** A unified view of rider accounts and their transaction activities.
- **Use:** Offers a quick reference for customer service to assist riders and manage accounts.

VIEW_INCIDENTS_BY_DATE

- **What it does:** Lists all incidents that occurred on a particular date.
- **Fields involved:** Incident details, involved employee, and the vehicle.
- **Output:** A list of incidents filtered by date.
- **Use:** Useful for operational reviews, safety audits, and incident tracking.

VIEW_ENHANCED_STATION_SCHEDULE_TRAFFIC

- **What it does:** Provides an overview of station activities, including train schedules and route details.
- **Fields involved:** Station details, route information, train schedules, and maintenance status.
- **Output**: Detailed station and route data.
- **Use:** Helps in managing station operations and preparing for peak traffic times.

5. TABLE - LEVEL CHECK CONSTRAINTS

CK_EMPLOYEE_ROLE CHECK:
Ensures that the role of an employee is one of the predefined set of valid roles.

CK_PAYMENT_METHOD CHECK:
Validates that the payment method is within the allowed methods.

CK_INCIDENT_TYPE CHECK:
Confirms that the incident type is within the specified categories.

CK_TRANSACTION_TYPE CHECK:
Checks that the transaction type falls into one of the allowed types.

CK_EMPLOYEE_AGE CHECK:
Verifies that the employee's age is within the required limits.

CK_CONTACT_INFO CHECK:
Ensures that the contact information meets the defined format or length criteria.

6. NON-CLUSTERED INDEXES

IDX_INCIDENT_TIMESTAMP

- **What it does**: Optimizes search queries on the INCIDENTS table by indexing the INCIDENT_TIMESTAMP column.
- **Use**: Improves performance for queries that retrieve incidents based on the timestamp, such as finding all incidents that occurred on a specific date.