

Introduction To Database Systems - LAB

Project Phase 01

Submitted by

| Muhammad Sinnan Hamid | L1F21BSSE0036 |
| --- | --- |
| Hafiz Faizan Ahmed | L1F21BSSE0030 |

Submitted to

MR. Khizar Khiyat

Submission date:

2023-01-08

Tables Of Content:

1. [PART I](#_jknh1r70fiyl) 2
   1. [Description:](#_yy936lc3ig6f) 2
   2. [Assumptions:](#_j3zr5esqvlp3) 2
   3. [Procedure:](#_4u72ajnfn77h) 2
   4. [The Flow of System:](#_yzqkt65snulx)

3

1. [PART II](#_49h59ayfk3gs) 4
   1. [Entities:](#_sncers1cl130) 4
   2. [Attributes for Entities:](#_97qyno1pr5q9) 4
      1. [Stations:](#_pexeak4nuclr) 4
      2. [Lines:](#_494pom30yr2c) 4
      3. [Trains:](#_c80sb32zn2ma) 5
      4. [Schedules:](#_srntzcc88gds) 5
      5. [Fair:](#_9jh7p5wxwi9l) 5
      6. [Trips:](#_e984a8r08i3b) 5
      7. [Employees:](#_2bqtuk7b1x2e) 6
      8. [Customers:](#_3jpfg693tg74) 6
      9. [Tickets:](#_udfx2b4cimkq) 6
      10. [Maintenance:](#_jwdtl8jh67t9)

7

1. [PART III](#_r447n7sjxey9) 8
   1. [Relations Among Entities:](#_iw30bl321pks) 8
      1. [One-to-One relationships:](#_hpq6bovkgqdz) 8
      2. [One-to-Many relationships:](#_u3fxpyntu4dd) 8
      3. [Many-to-many relationships:](#_pj3hr9u85jfa) **8**

Metro Train Management System

# 

# PART I

## Description:

A database for a metro train system would be used to store and manage information about the stations, lines, trains, schedules, fares, and users of the system. The database would need to support queries for accessing and updating this information in an efficient and reliable way and would need to ensure the integrity and consistency of the data.

The database might include tables for storing information about the stations, lines, trains, fares, and tickets, as well as tables for storing information about the system users, the trips they take, maintenance, incidents, customer service inquiries, and fare payments.

## Assumptions:

These are some of the general assumptions we take while developing the database for this system:

1. The database will store information about the stations, lines, trains, and schedules of the metro system.
2. The database will store information about the fares charged by the metro system and the different types of tickets that are available.
3. The database will store information about the users of the metro system and the trips that they take.
4. The database will need to support queries for accessing and updating this information in an efficient and reliable way.
5. The database will need to ensure the integrity and consistency of the data, for example by using constraints and transactions to prevent inconsistencies or corruption.

## Procedure:

Here is a general process that you might follow when designing a database for a metro train system:

1. Identify the requirements and goals of the system: The first step in designing the database is to understand the specific needs and objectives of the metro train system. This might involve gathering requirements from stakeholders such as employees, management, or users of the system.
2. Identify the entities and their attributes: Once you have a clear understanding of the requirements and goals of the system, you can begin to identify the entities that will be needed to store and manage the relevant data. For each entity, you will also need to identify the attributes or fields that will be needed to store the relevant information.
3. Define the relationships between the entities: Next, you will need to define the relationships between the different entities in the database. This might involve identifying one-to-one, one-to-many, or many-to-many relationships between entities.
4. Implement the database: Once you have completed the design of the database, you can begin implementing it using a database management system such as MySQL. This might involve creating the tables and fields defined in the design, as well as any indexes, triggers, or stored procedures that are needed.
5. Test and optimize the database: After implementing the database, it is important to test and debug it to ensure that it is functioning correctly and efficiently. You may also want to optimize the database by indexing fields that are frequently searched or updated, or by partitioning large tables to improve query performance.

## The Flow of System:

Here is a general flow of how a metro train system might work, and how the database might be used to support it:

1. Users purchase tickets: Users of the metro train system can purchase tickets at stations, online, or through a mobile app. The *database* would be used to store information about the different types of tickets available (e.g. single ride, day pass, monthly pass), the fares for each ticket, and the users who have purchased tickets.
2. Trains operate according to schedules: The metro train system operates according to schedules, which specify the times that trains depart from each station and the intervals between trains. The *database* would be used to store the schedules for each line, as well as information about the trains that operate on each line.
3. Users board trains and tap their tickets: When users board a train, they use their ticket to access the system by tapping it on a fare gate or scanning it with a mobile device. The *database* would be used to verify the validity of the ticket and record the trip taken by the user.
4. Trains arrive at stations and users exit: When a train arrives at a station, users who are at their destination can exit the train. The *database* would be used to record the ending station for each trip taken by a user.
5. Maintenance and repairs are performed: The metro train system requires regular maintenance and repairs to ensure the safety and reliability of the system. The *database* would be used to store information about the maintenance and repair work that is performed, including the type of work, the cost, and any disruptions to service.

# PART II

## Entities:

This is a list of Entities that could be used to represent a Metro Train Management System in a MySql Database:

1. Stations
2. Lines
3. Trains
4. Schedules
5. Fair
6. Trips
7. Employees
8. Customers
9. Tickets
10. Maintenance

## Attributes for Entities:

Here we will identify attributes for each of the entities mentioned above:

### 

### **Stations:**

| ATTRIBUTES: |
| --- |
| Station\_ID (Primary Key) |
| Name |
| Location |
| No\_of\_Platforms |

### **Lines:**

| ATTRIBUTES: |
| --- |
| Line\_ID (Primary Key) |
| Line\_Name |
| List of Station Served by line |

### **Trains:**

| ATTRIBUTES: |
| --- |
| Train\_ID (Primary Key) |
| Line\_ID (Foreign Key) |
| Train\_Type |
| Capacity |

### 

### **Schedules:**

| ATTRIBUTES: |
| --- |
| Schedule\_ID (Primary Key) |
| Line\_ID (Foreign Key) |
| Station\_ID (Foreign Key) |
| Departure\_Time |

### 

### **Fair:**

| ATTRIBUTES: |
| --- |
| Fair\_ID (Primary Key) |
| Line\_ID (Foreign Key) |
| Type |
| Cost |

### 

### Trips**:**

| ATTRIBUTES: |
| --- |
| Trip\_ID (Primary Key) |
| Fair\_ID (Foreign Key) |
| User\_ID (Foreign Key) |
| Staring\_Station\_ID (Foreign Key) |
| Ending\_Station\_ID (Foreign Key) |
| Date\_and\_Time |
| Fare\_Status |

### 

### Employees**:**

| ATTRIBUTES: |
| --- |
| Employee\_ID (Primary Key) |
| Station\_ID (Foreign Key) |
| Name |
| Address |
| Email\_Address |
| Phone\_Number |

### 

### Customers**:**

| ATTRIBUTES: |
| --- |
| Customer\_ID (Primary Key) |
| Trip\_ID (Foreign Key) |
| Name |
| Email\_Address |
| Phone\_Number |

### 

### Tickets**:**

| ATTRIBUTES: |
| --- |
| Ticket\_ID (Primary Key) |
| Fair\_ID (Foreign Key) |
| User\_ID (Foreign Key) |
| Date\_Of\_Purchase |

### 

### Maintenance**:**

| ATTRIBUTES: |
| --- |
| Line\_ID(Foreign Key) |
| Departure\_Time |
| Arrival\_Time |
| Time\_Between\_Trains |

# PART III

## Relations Among Entities:

### One-to-One relationships:

* A train has one type (e.g. light rail, subway, monorail). A train type has one train.
* A user has one contact information (e.g. email, phone number). Contact information belongs to one user.

### One-to-Many relationships:

* A line has many stations. A station belongs to one line.
* A fare has many tickets. A ticket belongs to one fare.
* A Customer has many tickets. A ticket belongs to one Customer.
* A trip has one starting station and one ending station. A station can be the starting station or ending station for many trips.
* A Station has many Employees. An Employee has only one Station to work.

### Many-to-many relationships:

* A train can operate on many lines. A line can have many trains.
* A station can be served by many lines. A line can serve many stations.
* A trip can involve many fares. A fare can be used for many trips.