

|  |
| --- |
| Business Template  **Subway (metro)** |
| [IMG_256](https://www.google.com/url?sa=i&url=https://en.wikipedia.org/wiki/File:Baltimore_Metro_Subway_train_car_logo.svg&psig=AOvVaw3DniBWg-Fe_HeyrF-NeSb2&ust=1711856616654000&source=images&cd=vfe&opi=89978449&ved=0CBIQjRxqFwoTCND1nP6Im4UDFQAAAAAdAAAAABAR" \t "/Users/abror2142/Documents/x/_blank) |

Contents

[1 Business Description 3](#_Toc62212630)

[1.1 Business background 3](#_Toc62212631)

[1.2 Problems. Current Situation 3](#_Toc62212632)

[1.3 The benefits of implementing a database. Project Vision 3](#_Toc62212633)

[2 Model description 3](#_Toc62212634)

[2.1 Definitions & Acronyms 3](#_Toc62212635)

[2.2 Logical Scheme 3](#_Toc62212636)

[2.3 Objects 3](#_Toc62212637)

# 

# Business Description

## Business background

The subway system, Metro M (name of the business), is an underground railway system used to transport large numbers of passengers within urban and suburban areas. Metro M was built in late 2000, and since then served millions of customers with its outstanding services. Now, it attracted many investors to moderinize the service in line with the current advances in high-tech world.

## Problems. Current Situation

Now Metro M, has 5 lines and more than enough trains to operate on time. Here is the full map of the Metro M (imitating Tashkent Metro Map):



Currently, there is an outdated system, which do not track everything digitally, and have no software to store all information about the system. Our main task is to develop such a system to store data about:

* subway stations
* lines with routes, schedules, operating frequencies, required trains and employees
* train arrival and departure (schedule of station)
* tickets including ticket types, prices, discounts, promotions
* maintanance of the subway infrastructure, tunnels tracks, and stations.

By talking with a bussiness staff we got following information about the system:

1. Line is the metro service provided for certain area, and colors are used to differentiate them.  
    2) The route can be one line or the sum of different lines with intersection  
    3) The train will belong to one route not one line  
    4) One station is the part of both a line and route, and relationship is that the station may serve for many routes and many lines as well.  
    5) Trains will start moving reverse after the last station in the route  
    6) Every train has specific subway lines.  
    7) Trains will need x minutes to perform geting people in and get people off the people and open and close the doors.

## the Benefits of implementing a database. Project Vision

By implementing this project to the current subway system, We can improve its service quality by identifying these things:

* The data on the stations will give the people information about the station, its lines and platforms which makes the finding correct train easier.
* The system will have detailed and accurate information about the lines and schedules. This is crucial for providing high-quality service to the passangers.
* The system will have accurate information about the daily volume of requests for this service. With this information, it can predict the peak hours, and know its service capacity.
* The overall sales of ticket and the full amount of money it generates daily, quarterly or yearly.
* The detailed information about the passengers who used the metro service which can be used for marketing purposes and/or for promotion purposes by offering right amount of discount to the passengers who are loyal to the metro.
* Besides, the new system offers the information about the maintenance work done in the whole subway system, their date, cost, and descriptions.

## Definitions & Acronyms

Station is the point where trains stop. One station can have many lines, one station can have many platforms.

Route = one line or many lines. e.g. Sergeli line + Chilonzor line + Yunusobod line = One Route, but the name of the route does not have any meaning to customer. So giving a name to the route is optional. Besides, the train goes on the route, changing lines on the way.

Line is a combination of stations and the path to get there.

Schedules give information about the timeline will be placed at the station. One station has one schedule only. I includes data about on which platform and when the train will come and go.

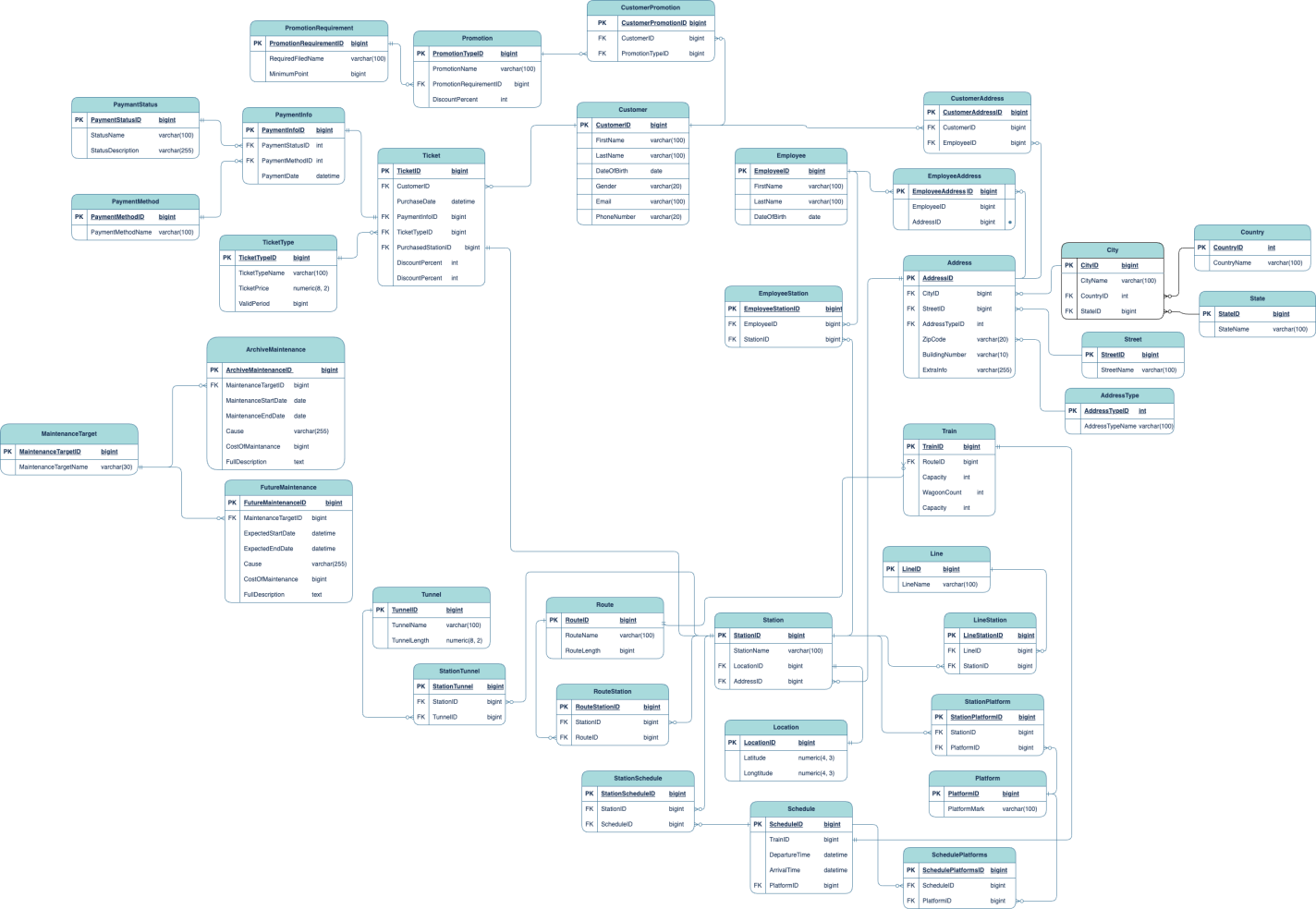
Tickets are have a specific customer, and customer may have many tickets. Ticket is simple. Once it is purchased the customer can use it to enter the station and no need to buy a ew ticket when changing lines.

Ticket types can be one time, for one day, for one week, for one month or may be a yearly subscription.

Prices are calculated after applying discount to the real price of the ticket. Real price is determined by its type.

Discounts are calculated according to the points the customer have. First we consider customers points then we identify its promotion level and discount percent by joining it with promotion, then we store this info to the ticket detail.

## Logical Scheme



## Objects

**Address**

Table Description

This table is used to provide addresses for all kind of other tables. For example ,it is used for customer, employee, station address.

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| Address | AddressID | PK | bigint |
| CityID | FK | bigint |
|  | StreetID | FK | bigint |
|  | ZipCode |  | varchar(20) |
|  | BuildingNumber |  | varchar(20) |
|  | ExtraInfo |  | varchar(255) |

Comments on table relationships

City and Street is created seperately. And ZipCode can may include letters, so varchar is used.

Example with data

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field Name 1 | Field name 2 | Field name 3 | Field name 4 | Field name 5 | Field Name 6 |
| 1 | 1 | 1 | 150100 | 12/5 | - |

**City**

Table Description

City table describes not only city name but also the country and state the city situated.

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| City | CityID | PK | bigint |
| CityName |  | varchar(100) |
|  | CountryID | FK | int |
|  | StateID | FK | bigint |

Comments on table relationships

City table has 2 FKs and only one independent column which is CityName

Example with data

|  |  |  |  |
| --- | --- | --- | --- |
| CityID | CityName | CountryID | StateID |
| 1 | Tashkent | 3 | 2 |

**Country**

Table Description

Country table includes possible countries for the city table.

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| Country | CountryID | PK | int |
| CountryName |  | varchar(100) |

Comments on table relationships

Example with data

|  |  |
| --- | --- |
| CountryID | CountryName |
| 3 | Uzbekistan |

**State**

Table Description

State is used for creating values for city table.

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| State | StateID | PK | bigint |
| StateName |  | varchar(100) |

Comments on table relationships

State table specifies only statename for now but could inlcude more information in the real world.

Example with data

|  |  |
| --- | --- |
| StateID | StateName |
| 2 | Tashkent |

**Street**

Table Description

Street table is used to provide information about the street in address table.

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| Street | StreetID | PK | bigint |
| StreetName |  | varchar(100) |

Comments on table relationships

This table is in relationship with address table only.

Example with data

|  |  |
| --- | --- |
| StreetID | StreetName |
| 1 | Wall Street |

**AddressType**

Table Description

This table is used to specify different kinds of address types. These types may include work, home .. or other types. With these e.g. a customer can have both work and home address.

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| AddressType | AddressTypeID | PK | Int |
| AddressTypeName |  | varchar(100) |

Comments on table relationships

This table is in relationship with the address table.

Example with data

|  |  |
| --- | --- |
| AddressTypeID | AddressTypeID |
| 1 | home |
| 2 | work |

**Employee**

Table Description

These table includes information about the employees in the Metro M

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| Employee | EmployeeID | PK | bigint |
| FirstName |  | varchar(100) |
|  | LastName |  | varchar(100) |
|  | DateOfBirth |  | date |

Comments on table relationships

An employee does have an address. But it has included in the seperate table

Example with data

|  |  |  |  |
| --- | --- | --- | --- |
| EmployeeID | FirstName | LastName | DateOfBirth |
| 1 | John | Doe | 12.09.1994 |

**EmployeeAddress**

Table Description

EmployeeAddress Table includes information about address of the employees. It is used to provide many-to-many relationship as one employee can have home and work address.

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| EmployeeAddress | EmployeeAddressID | PK | bigint |
| EmployeeID | FK | bigint |
|  | AddressID | FK | bigint |

Comments on table relationships

This is used to provide many-to-many relationship.

Example with data

|  |  |  |
| --- | --- | --- |
| EmployeeAddressID | EmployeeID | AddressID |
| 1 | 2 | 1 |

**EmployeeStation**

Table Description

This table is used to provide information about the work place(station) of an employee.

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| EmployeeStation | EmployeeStationID | PK | bigint |
| EmployeeID | FK | bigint |
|  | StationID | FK | bigint |

Comments on table relationships

The table provides many-to-many relationship between employees and stations. Because one employee can work in more than one station.

Example with data

|  |  |  |
| --- | --- | --- |
| EmployeeStationID | EmployeeID | StationID |
| 1 | 2 | 1 |

**Customer**

Table Description

Customer table contains information about the customers of the Metro M.

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| Customer | CustomerID | PK | bigint |
| FistName |  | varchar(100) |
|  | LastName |  | varchar(100) |
|  | DateOfBirth |  | date |
|  | Gender | can be: male, female | varchar(10) |
|  | Email |  | varchar(100) |
|  | PhoneNumber |  | varchar(20) |
|  | Point |  | bigint |

Comments on table relationships

Customers can buy many tickets and can have many types of promotions at a time(like age over 60 and loyaty points more than 1000). They collect points as they use the subway and they can get different types of benefits using them. It also may indicate their loyalty.

Example with data

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| CustomerID | FirstName | LastName | DateOfBirth | Gender | Email | PhoneNumber | Point |
| 1 | Jane | Doe | 12.12.1982 | female | jane@mail.com | 0123-45-6-78 | 100 |

**CustomerAddress**

Table Description

CustomerAddress describes many to many relationship between customers and addresses.

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| CustomerAddress | CustomerAddressID | PK | bigint |
| CustomerID | FK | bigint |
|  | AddressID | FK | bigint |

Comments on table relationships

This is many-to-many relationship, as one customer may have different addresses at the same time, like the employees.

Example with data

|  |  |  |
| --- | --- | --- |
| CustomerAddressID | CustomerID | AddressID |
| 1 | 1 | 1 |

**CustomerPromotion**

Table Description

This table is used to have many to many relationship between Customer and Promotion. One Customer may have many promotions, and one promotion can be applied to many customers as well.

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| CustomerPromotion | CustomerPromotionID | PK | bigint |
| CustomerID | FK | bigint |
|  | PromotionID | FK | bigint |

Comments on table relationships

This is many-to-many relationship between customers and promotions they have.

Example with data

|  |  |  |
| --- | --- | --- |
| CustomerPromotionID | CustomerID | PromotionID |
| 1 | 1 | 2 |
| 2 | 1 | 1 |

**Promotion**

Table Description

Promotions table is used to determine promotion infomrtaion like discount percent, and requiremnet to be eligible to this promotion.

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| Promotion | PromotionID | PK | bigint |
| PromotionName |  | varchar(100 |
|  | PromotionRequirementID | FK | bigint |
|  | DiscoutPercent |  | int |

Comments on table relationships

Promotions table one specific requirement to select a group of people for promoting.

This requirements are specified in other table.

Example with data

|  |  |  |  |
| --- | --- | --- | --- |
| PromotionID | PromotionName | PromotionRequirementID | DiscountPercent |
| 1 | Discount for Elderly | 1 | 30 |

**PromotionRequirement**

Table Description

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| PromotionRequirement | PromotionRequirementID | PK | bignt |
| RequiredFieldName | e.g. age | varchar(100) |
|  | MinimumPoints | e.g. 60 | bigint |

Comments on table relationships

This table includes RequiredFieldname and a certain constraint to apply. for example age > 60

Example with data

|  |  |  |
| --- | --- | --- |
| PromotionRequirementID | RequiredFieldName | MinimumPoints |
| 1 | age | 60 |

**Ticket**

Table Description

Ticket table shows the information about the ticket that has been bought by a customer.

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| Ticket | TicketID | PK | bigint |
| CustomerID | FK | bigint |
|  | PurchaseDate |  | date |
|  | PaymentInfo | FK | bigint |
|  | TicketTypeID | FK | int |
|  | PurchasedStationID | FK | bigint |
|  | DiscountPercent |  | int |

Comments on table relationships

Ticket table has many relations with other tables including paymentinfo and the station which it has ben sold. ticket type. In this case ticket type can include the real price of the ticket, and discount percent is calculated according to the customers promotion levels.

Example with data

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| TicketID | CustomerID | PurchaseDate | PaymentInfo | TicketTypeID | | PurchasedStationID | DiscountPercent |
| 1 | 1 | 12.02.20224 | 1 | 1 | 1 | | 30 |

**PaymentInfo**

Table Description

PaymentInfo table contains informetion about the specific ticket.

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| PaymentInfo | PaymentInfoID | PK | bigint |
| PaymentStatusID | FK | int |
|  | PaymentMethodID | FK | int |
|  | PaymentDate |  | datetime |

Comments on table relationships

This table is in relation with PaymentStatus and PaymentMethod tables. They indicate how the payment has been done.

Example with data

|  |  |  |  |
| --- | --- | --- | --- |
| PaymentInfoID | PaymentStatusID | PaymentMethodID | PaymentDate |
| 1 | 1 | 1 | 11.02.2024 |

**PaymentMethod**

Table Description

PaymentMethod table is used to show in which way the payment has been paid.

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| PaymentMethod | PaymentMethodID | PK | int |
| PaymentMethodName | e.g. cash, visa | varchar(100) |

Comments on table relationships

This table is in relation with PaymentInfo to support it

Example with data

|  |  |
| --- | --- |
| PaymentMethodID | PaymentMethodName |
| 1 | Cash |
| 2 | CreditCard |
| 3 | DebitCard |

**PaymentStatus**

Table Description

<PaymentStatus gives information about the status of the payement mentioned in paymentInfo table

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| PaymentStatus | PaymentStatusId | PK | int |
| StatusName | e.g. pending | varchar(100) |
|  | StatusDescription |  | varchar(255) |

Comments on table relationships

This table supports the Payment Info table field, and makes many-to-one relationship with it.

Example with data

|  |  |  |
| --- | --- | --- |
| PaymentStatusID | PaymentStatusName | StatusDescription |
| 1 | pending | You need to complete payment process. |
| 2 | paid | You completed your payment. |
| 3 | cancelled | Your payment has been cancelled. |

**TicketType**

Table Description

TicketType table is used to separate the different categories of the ticket types.

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| TicketType | TicketTypeID | PK | bignt |
| TicketTypeName |  | varchar(100) |
|  | TicketPrice | e.g. $5.50 | numeric(8, 2) |
|  | ValidPeriod | 86400 | bigint |

Comments on table relationships

This table is in reltaion with ticket table. ValidPeriod is measured in seconds.

Example with data

|  |  |  |  |
| --- | --- | --- | --- |
| TicketTypeID | TicketTypeName | TicketPrice | ValidPeriod |
| 1 | One-Day | 20.90 | 86400 |

**Train**

Table Description

Train table describes information about one train, including its routes.

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| Train | TrainID | PK | bigint |
| RouteID | FK | bigint |
|  | Capacity |  | int |
|  | WagoonCount |  | int |

Comments on table relationships

The train is in relation with routes.

Example with data

|  |  |  |  |
| --- | --- | --- | --- |
| TrianID | RouteID | Capacity | WagoonCount |
| 1 | 2 | 100 | 5 |

**Line**

Table Description

Lines can be one part of the route, and they include distence between several stations.

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| Station | LineID | PK | bigint |
| LineName |  | varchar(100) |

Comments on table relationships

This line table is in relation with stations.

Example with data

|  |  |
| --- | --- |
| LineID | LineName |
| 1 | Uzbekistan Line |

**LineStation**

Table Description

The LineStations table is used to show the many-to-many relationship between the line and station.

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| LineStation | LineStation | PK | bigint |
| LineID | FK | bigint |
|  | StationID | FK | bigint |

Comments on table relationships

This make the relationship between the two entities.

Example with data

|  |  |  |
| --- | --- | --- |
| LineStationID | LineID | StationID |
| 1 | 1 | 2 |
| 2 | 1 | 1 |

**Station**

Table Description

Station Table is one of the core tables in the project. As it has many dependencies, like schedule, employee, line, route, platform and more.

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| Station | StationID | PK | bigint |
| StationName |  | varchar(100) |
|  | LoacationID | FK | bigint |
|  | AddresssID | FK | bigint |

Comments on table relationships

It has two one-to-many relations with other tables, Location and Address

Example with data

|  |  |  |  |
| --- | --- | --- | --- |
| StationID | StationName | LocationID | AddresssId |
| 1 | Aybek | 2 | 3 |

**StationLocation**

Table Description

The StationLocation table is used to show the information about the location of the station.

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| StationLocation | StationLocationID | PK | bigint |
| StationID | FK | bigint |
|  | LocationID | FK | bigint |

Comments on table relationships

The relationship describes many-to-many relation.

Example with data

|  |  |  |
| --- | --- | --- |
| StationLocationID | StationID | LocationID |
| 1 | 2 | 12 |

**Location**

Table Description

The Location table shows the information about the latitude and longtitude of the place.

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| Location | LocationID | PK | bigint |
| Latitude |  | numeric(4, 3) |
|  | Longtitude |  | numeric(4, 3) |

Comments on table relationships

The

Example with data

|  |  |  |
| --- | --- | --- |
| LocationId | Latitude | Longtitude |
| 1 | 432.47 | 12.409 |

**Platform**

Table Description

Platform is used to show the different platforms of the station and their name(mark).

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| Platform | PlatformID | PK | bigint |
| PlatformMark |  | varchar(100) |

Comments on table relationships

This table will have many dependent tables, like schedule.

Example with data

|  |  |
| --- | --- |
| PlatformID | PlatformMark |
| 1 | 6-a |

**StationPlatform**

Table Description

The StationPlatform table describes how many platforms have one station.

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| StationPlatform | StationPlatformID | PK | bigint |
| StationID | FK | bigint |
|  | PlatformID | FK | bigint |

Comments on table relationships

This tabe is used to describe many-to-many relationship betweeen the stations and platforms.

Example with data

|  |  |  |
| --- | --- | --- |
| StationPlatformID | StationID | PlatformID |
| 1 | 2 | 11 |

**Schedule**

Table Description

The Schedule Table gives information about the train departure and arrival in stations. This will include the info about the platform too.

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| Schedule | ScheduleID | PK | bigint |
| TrainID | FK | bigint |
|  | PlatformID | FK | bigint |
|  | DepartureTime |  | datetime |
|  | ArrivalTime |  | datetime |

Comments on table relationships

This table includes relationships with Train, and Platforms

Example with data

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ScheduleID | TrainID | PlatformID | DepartureTime | ArrivalTime |
| 1 | 1 | 11 | 1.02.2024 13:00 | 1.02.2024 12:30 |

**SchedulePlatform**

Table Description

The SchedulePlatform table is here to show that one train can leave in many platform like 1-a to 1-d.

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| SchedulePlatform | SchedulePlatformID | PK | bigint |
| ScheduleID | FK | bigint |
|  | PlatformID | FK | bigint |

Comments on table relationships

This tabe is used to describe many-to-many relationship betweeen the Schedule and platforms.

Example with data

|  |  |  |
| --- | --- | --- |
| StationPlatformID | ScheduleID | PlatformID |
| 1 | 2 | 1 |

**StationSchedule**

Table Description

The StationSchedule is used to show that this schedule is for one station.

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| StationPlatform | ScheduleStationID | PK | bigint |
| ScheduleID | FK | bigint |
|  | StationID | FK | bigint |

Comments on table relationships

This table describes many-to-many relationship between the schedule and the station.

Example with data

|  |  |  |
| --- | --- | --- |
| ScheduleStationID | ScheduleID | StationID |
| 1 | 2 | 2 |

**Route**

Table Description

The route table describes information about the route, which can be described as the sum of many lines.

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| Route | RouteID | PK | bigint |
| RouteName |  | varchar(100) |
|  | RouteLength |  | bigint |

Comments on table relationships

The Route has the trains so one route can have many trains and one train can have only one route.

Example with data

|  |  |  |
| --- | --- | --- |
| RouteID | RouteName | RouteLength |
| 1 | Navoiy Route | 80 |

**RouteStation**

Table Description

The RouteStation table is used to show the fact that one Route can have many station on the other hand one station can be part of many routes.

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| RotuteStation | RouteStationID | PK | bigint |
| RouteID | FK | bigint |
|  | StationID | FK | bigint |

Comments on table relationships

This table show the many-to-many relationship between the routes and stations.

Example with data

|  |  |  |
| --- | --- | --- |
| RouteStationID | RouteID | StationID |
| 1 | 2 | 1 |

**Tunnel**

Table Description

The Tunnel Table describes data on the existing tunnels between stations.

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| Tunnel | TunnelID | PK | bigint |
| TunnelName |  | varchar(100) |
|  | TunnelLength |  | numeric(8, 2) |

Comments on table relationships

This table can have many dependencies, like tunnels between stations

Example with data

|  |  |  |
| --- | --- | --- |
| TunnelID | TunnelName | TunnelLength |
| 1 | 2 | 1 |

**TunnelStation**

Table Description

The TunnelStation table describes the information about the tunnels between stations.

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| TunnelStation | TunnelStationID | PK | bigint |
| TunnelID | FK | bigint |
|  | StationID | FK | bigint |

Comments on table relationships

This table is used to describe many-to-many relationship between tunnels and station.

Example with data

|  |  |  |
| --- | --- | --- |
| TunnelStationID | TunnelID | StationID |
| 1 | 2 | 3 |

**ArchieveMaintenance**

Table Description

This table describes information about the Maintenance work done in the past.

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| ArchiveMaintenance | ArchiveMaintenanceID | PK | bigint |
| MaintenanceTargetID | FK | bigint |
|  | MaintenanceStartDate |  | date |
|  | MaintenanceEndDate |  | date |
|  | Cause |  | varchar(255) |
|  | CostOfMaintenance |  | bigint |
|  | FullDescription |  | text |

Comments on table relationships

This table has one relationship with MaintenanceTarget table.

Example with data

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ArchiveMaintenanceID | MaintenanceTargetID | MaintenanceStartDate | MaintenanceEndDate | Cause | CostOfMaintenance | FullDescription |
| 1 | 2 | 12.02.2022 | 15.02.2022 | Broken ceiling | 50,000 | \*New ceiling has been installed |

**FutureMaintenance**

Table Description

This table describes information about the planned Maintenance work which will be done in the future.

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| ArchiveMaintenance | ArchiveMaintenanceID | PK | bigint |
| MaintenanceTargetID | FK | bigint |
|  | ExpectedStartDate |  | date |
|  | ExpectedEndDate |  | date |
|  | Cause |  | varchar(255) |
|  | CostOfMaintenance |  | bigint |
|  | FullDescription |  | text |

Comments on table relationships

This table also, has one relationship with MaintenanceTarget table.

Example with data

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ArchiveMaintenanceID | MaintenanceTargetID | ExpectedStartDate | ExpectedEndDate | Cause | CostOfMaintenance | FullDescription |
| 1 | 2 | 02.05.2024 | 18.05.2024 | \*renovation | 250,000 | \*New floor will be installed |

**MaintenanceTarget**

Table Description

The MaintenanceTarget table is used to show the Target of the maintenance. it can be a station, train, route, tunnel anything.

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| MaintenanceTarget | MaintenanceTargetID | PK | bigint |
| MaintenanceTargetName |  | varchar(30) |

Comments on table relationships

This table includes the table name as the target. By doing so we can have any table maintenance info in three tables.

Example with data

|  |  |
| --- | --- |
| MaintenanceTargetID | MaintenanceTargetName |
| 1 | Station |