

|  |
| --- |
| Business Template  **Subject areas** |
| **Logo / Image** |

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

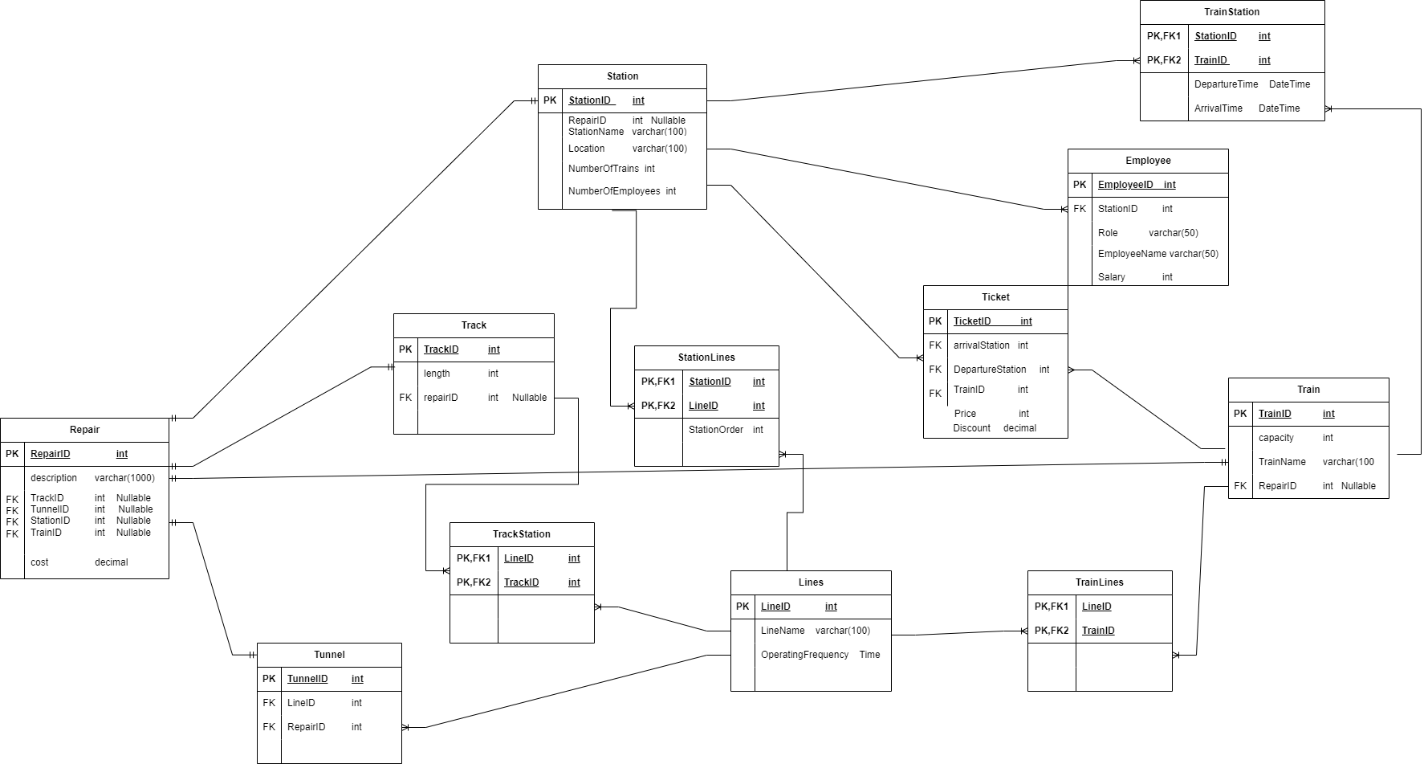
## Problems. Current Situation

## the Benefits of implementing a database. Project Vision

# Model description

## Definitions & Acronyms

## Logical Scheme



## Objects

Table Description

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| Station | StationID | PK | Int |
| RepairID | FK Nullable | Int |
| Location |  | Varchar(100) |
|  | NumberOfTrains |  | int |
|  | NumberOfEmployees |  | int |

Table station have One-to-Many relationship with table ticket because at one station there can be many tickets bought but you can buy exactly one ticket at one station. Because of that table Ticket needs StationID as FK and I added ArrivalStation and DepartureStation as foreign keys. Also it have One-to-Many relationship with Table employee because at one station there can be many employees but one employee works at one station, because of that Table Employee needs StationID as FK and I also added it in Employee table. Table station has Many-to-Many relationship with Table Train, because there can be many trains at one station and train can go at many station so because of that I created bridge table and Named it TrainStation and I will talk about it later. there is One-to-One relation between table Station and table Repair, I added RepairID as FK in Station table because we need information if this station is being repaired or not, and if not it should be Null and that’s why I added Nullable in the Table. And lastly there is again Many-to-Many relationship with table Line. Because one station can be at many lines and line can be at many stations. Because of that I created another bridge table and added StationID and LineID as composite primary key and also marked them as foreign keys.

Example with data

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| StationID | RepairID | Location | NumberOfTrains | NumberOfEmployees |
|  | Null | New-york |  |  |



|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| Ticket | TicketID | PK | Int |
| DepartureStation | FK | Int |
| ArrivalStation | FK | int |
| TrainID | FK | int |
|  | Price |  | int |
|  | Discount |  | decimal |

Ticket table has One-to-Many relationship with Station and I already talked about that. It also has One-to-Many relationship with table Train, because you can buy 1 ticket for 1 train and for one train there are multiple tickets to be bought. Because of that I added TrainID as FK. Ticket also has price and its type is integer and I also added Discount attribute, because in the text discount was mentioned. That’s it for Ticket Table.

Example with data

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| TicketID | DepartureStation | ArrivalStation | TrainID | Price | Discount |
| 1 | 23 | 11 | 123 | 30 | 12.65 |

Table Employee

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| Employee | EmployeeID | PK | Int |
| StationID | FK | Int |
| Role |  | Varchar(50) |
| EmployeeName |  | varchar |
|  | Salary |  | int |

Employee table has EmployeeID as PK so we can identify every employee uniquely. We have StationID as FK that’s because there is One-to-Many relationship between Employee and Station Tables, so if we want to know in which station does the employee work we need to add StationID at the table. Also employees have roles and that’s why I added it as a attribute, they also have name and Salary.

Example with data

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| EmployeeID | StationID | Role | EmployeeName | Salary |
| 1 | 23 | mechanic | daviti | 3000 |

Table Train

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| Train | TrainID | PK | Int |
| RepairID | FK Nullable | Int |
| TrainName |  | Varchar(50) |
| capacity |  | int |

In Train table we have TrainID as PK and it uniquely identifies every train, also we have RepairID as FK

So we have information if we need to repair train and if not we say that it will be null. Another attribute is TrainName, to be honest I don’t know if trains have names but I added it anyway and in my logic there can be trains with the same name so TrainName doesn’t uniquely identifies train that’s why it is not part of the PK. Train also has capacity and I added it as a attribute with data type int. as I have already mentioned we have Many-to-Many relationship between table Train and table Station. We have another Many-to-Many relationship between table Train and table Lines, because there can be many trains at one line and one train can be at many lines. We have One-to-Many relationship between table Train and table Ticket and because of that I added TrainID in Ticket table. and lastly we have One-to-One relationship between table Repair and table Train.

Example with data

|  |  |  |  |
| --- | --- | --- | --- |
| TrainID | RepairID | TrainName | capacity |
| 1 | 7 | Easter express | 200 |

Table Lines

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| Lines | LineID | PK | Int |
| LineName |  | Varchar(50) |
| OperatingFrequency |  | time |

In table Lines LineID is PK. I also added LineName as an attribute and operatingFrequency which means how often do trains come and go and its type is time. There is Many-to-Many relationship between table Lines and table Train and I have already talked about that. Also there is Many-to-Many relationship between table Lines and table Station again I already talked about in table Station. There is also another Many-to-Many relationship between Table Track and table lines. Because one line can pass through multiple Tracks and one track can be used by multiple lines. Also there is One-to-Many relationship between Table Lines and table Tunnel. One tunnel is associated with one line and one line can pass through multiple tunnels.

Example with data

|  |  |  |
| --- | --- | --- |
| LineID | LineName | OperatingFrequency |
| 1 | Saburtalo Line | 07:00 |

Table Track

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| Track | TrackID | PK | Int |
| RepairID | FK Nullable | Int |
| length |  | int |

In table Track TrackID is PK, repairID is FK from table Repair and it tells us if Track needs to repair and last attribute length which tells us track length. Table Track has One-to-One relationship with table Repair and has Many-to-Many relationship with table Lines.

Example with data

|  |  |  |
| --- | --- | --- |
| TrackID | RepairID | length |
| 1 | Null | 254 |

Table Tunnel

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| Tunnel | TrackID | PK | Int |
| LineID | FK | Int |
| RepairID | FK Nullable | Int |

Table tunnel has TrackID as a PK and LineID, RepairID as foreign keys. It has One-to-Many relationship with table Lines and that’s why we need LineID in this table and has One-to-One relationship with table Repair and we add repairID in this table to track if some tunnels are damaged.

Example with data

|  |  |  |
| --- | --- | --- |
| TunnelID | LineID | RepairID |
| 1 | 7 | Null |

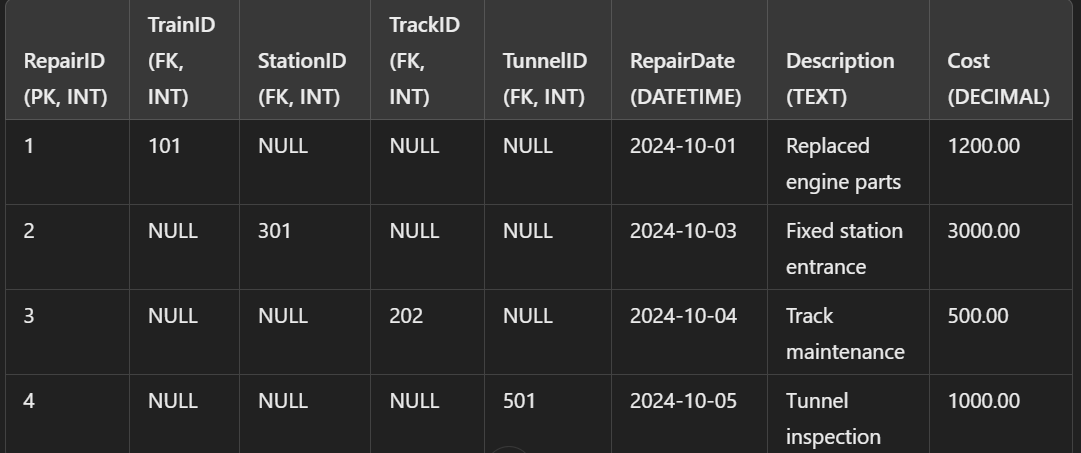
Table Repair

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| Repair | RepairID | PK | Int |
| description |  | Text |
| RepairDate |  | Date |
| cost |  | decimal |
|  | TrackID | FK Nullable | Int |
|  | TrainID | FK Nullable | int |
|  | StationID | FK Nullable | int |
|  | TunnelID | FK Nullable | int |

Table Repair has RepairID as PK, attribute description explains what was damaged and where, RepairDate says what is it for so does cost. This table has four One-to-One relationship with table Station,Track,Tunnel,Train.

I added RepairID in all four tables to keep track if something is damaged, I was thinking about not adding their PKs in RepairTable, but if we don’t add their PKs as FKs in this table how do we keep track if they are damaged. So I added all the keys in Table repair but they can also be Null of course.

Example with data(I tried to do this with this kind of table but I couldn’t fit in so I helped Chatgpt for example data)



Bridge Table-TrainLines

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| TrainLines | TrainID | PK,FK | Int |
| LineID | PK,FK | Int |

Table TrainLines is a bridge table and has TrainID and LineID as composite primary key and they are also foreign keys. I changed many-to-many relationship with two one-to-many relationship and that’s why I created this table.

Bridge Table-TrainStation

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| TrainStation | TrainID | PK,FK | Int |
| StationID | PK,FK | Int |

Table TrainStation is a bridge table and has TrainID and StationID as composite primary key and they are also foreign keys. I changed many-to-many relationship with two one-to-many relationship and that’s why I created this table.

Bridge Table-TrackStation

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| TrainStation | TrackID | PK,FK | Int |
| LineID | PK,FK | Int |

Table TrainStation is a bridge table and has TrackID and LineID as composite primary key and they are also foreign keys. I changed many-to-many relationship with two one-to-many relationship and that’s why I created this table.

Bridge Table-StationLines

|  |  |  |  |
| --- | --- | --- | --- |
| Table Name | Field name | Field Description | Data Type |
| StationLines | StationID | PK,FK | Int |
| LineID | PK,FK | Int |
|  | StationOrder |  | int |

Table TrainStation is a bridge table and has TrackID and LineID as composite primary key and they are also foreign keys. I changed many-to-many relationship with two one-to-many relationship and that’s why I created this table. And additionally added one attribute StationOrder to keep track how the stations are ordered in one line.

I think this is all about my logical scheme. In description it was written to use business template but I was confused and didn’t know how to use this template, I hope I used it correctly XD.