

|  |
| --- |
| 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

Hard to track stations, lines, trains and schedules. Maintenance and infrastructure history are easy to lose. Creating reports and analyzing travel and sales is extremely time-consuming.

## the Benefits of implementing a database. Project Vision

It will reduce errors, save time, and improve operational efficiency. The system will enable accurate reporting and data-driven decisions. Overall, it will ensure smoother and more organized subway management.

# Model description

## Definitions & Acronyms

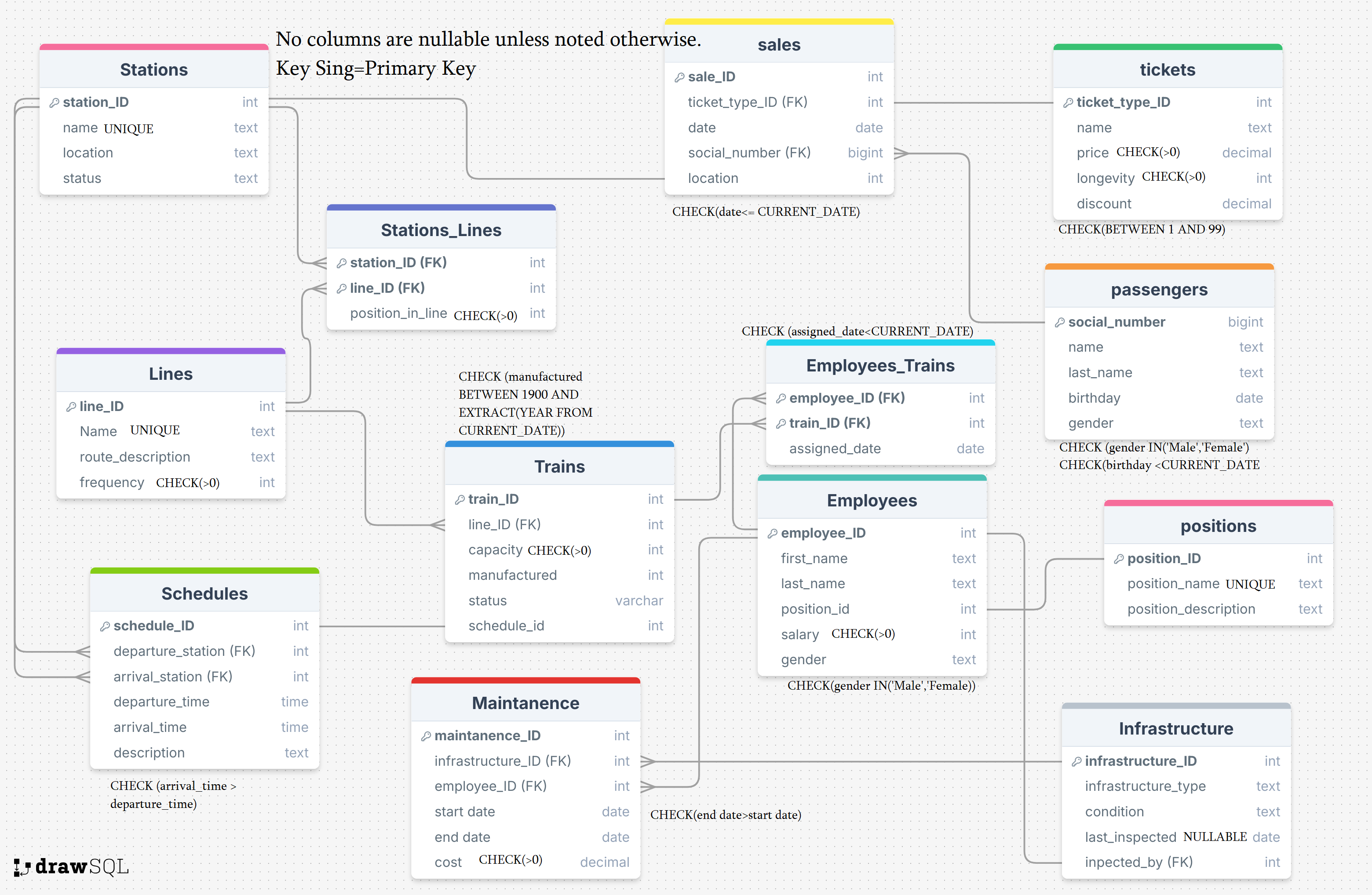
PK (Primary Key): A unique identifier for each record in a table. Denoted as the key symbol in this case.

FK (Foreign Key): A field that links one table to another.

“INT” data type – integer

“Text”, “decimal”, “date” and “time” are also data types.

## Logical Scheme



## Objects

|  |  |  |  |
| --- | --- | --- | --- |
| Stations | Field name | Field Description | Data Type |
| Table 1 | Station\_ID | PK | Int |
| name |  | Text |
|  | location |  | Text |
|  | status | Active/Closed/Under repair etc. | Text |

Keeps info about every station in the system. One station- one ID- one row.

Example with data:

|  |  |  |  |
| --- | --- | --- | --- |
| Station ID | name | location | status |
| 5 | State University | Vaja-Pshavela Ave, 99 | Active |

|  |  |  |  |
| --- | --- | --- | --- |
| Lines | Field name | Field Description | Data Type |
| Table 2 | Line ID | PK | Int |
| Name |  | Text |
|  | Route description |  | Text |
|  | frequency | Trains per hour | Int |

Keeps info about every line in the system. One line – One row.

Example with data

|  |  |  |  |
| --- | --- | --- | --- |
| Line ID | name | Route Description | Frequency |
| 3 | The Smooth Line | Connects the east side and north town | 12 |

|  |  |  |  |
| --- | --- | --- | --- |
| Stations\_Lines | Field name | Field Description | Data Type |
| Table 3 | Station\_ID | FK from table 1 - Stations | Int |
| Line\_ID | FK from table 2 -Lines | Int |
|  | Position in line | Tells us which position the given station holds in the given line. | Int |

Keeps info about which station is which line. This is a **many-to-many relationship** because every line has multiple stations and some stations belong to more than one line. This table has two foreign keys (line ID and station ID) which together make a composite key.

Example with data:

|  |  |  |
| --- | --- | --- |
| Station ID | Line ID | Position in line |
| 3 | 2 | 5 |

|  |  |  |  |
| --- | --- | --- | --- |
| Trains | Field name | Field Description | Data Type |
| Table 4 | Train\_ID | PK | Int |
| Line\_ID | Says on which line the train travels. FK from table 2 - Lines | Int |
|  | Capacity | Says how many passengers the train carries. | Int |
|  | Manufactured | Manufacture date | int |
|  | status | Active/Closed/Under repair etc. | text |
|  | Shedule\_ID | Which schedule the train follows. FK from table 5- schedules. | int |

Keeps info about every train in the system.

Example with data:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Train\_ID | Line ID | Capacity | Manufactured | status | Schedule\_id |
| 13 | 1 | 50 | 2009 | Under repair | 5 |

|  |  |  |  |
| --- | --- | --- | --- |
| Schedules | Field name | Field Description | Data Type |
| Table 5 | Schedule\_ID | PK | Int |
| Line\_ID | Says on which line the trains of this schedule travel. FK from table 2 - Lines | Int |
|  | Departure station | Says where the schedule starts. FK from table 1 –stations. | int |
|  | Arrival station | Says where the schedule ends. FK from table 1 –stations. | int |
|  | Departure time | Says when the schedule starts. | time |
|  | Arrival time | Says when the schedule ends. | time |
|  | Description |  | text |

Keeps info about every possible schedule each train might have.

Example with data:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Schedule\_ID | Line ID | Departure Station | Arrival station | Departure time | Arrival time | Description |
| 1 | 3 | 4 | 14 | 6:30 | 7:15 | Earliest departure of the day to take passengers to the first out of town train which leaves at 7:30 near station 14. |

|  |  |  |  |
| --- | --- | --- | --- |
| Employees | Field name | Field Description | Data Type |
| Table 6 | Employee ID | PK | Int |
| First name |  | text |
|  | Last name |  | text |
|  | position | FK from table 13 | text |
|  | salary |  | decimal |
|  | gender |  | text |

Keeps track of every subway employee.

Example with data:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Employee ID | Name | Last name | Position | Salary | Gender |
| 1245 | Giorgi | Giorgadze | Driver | 3,000 | Male |

|  |  |  |  |
| --- | --- | --- | --- |
| Employees trains | Field name | Field Description | Data Type |
| Table 7 | Employee ID | FK from table 6 - Employees | int |
| Train\_ID | Fk from table 4 -tains | int |
|  | Assigned date | When were they assigned to that train | date |

Tells us which employee is assigned to which train. This is another **many-to-many relationship** as multiple workers can work on one train and one worker can work on many trains.

Example with data:

|  |  |  |
| --- | --- | --- |
| Employee ID | Train Id | Assigned Date |
| 1245 | 17 | 20.12.2021 |

|  |  |  |  |
| --- | --- | --- | --- |
| Infrastructure | Field name | Field Description | Data Type |
| Table 8 | Infrastructure ID | PK | int |
| Infrastructure type | Describes the infrastructue. (Train, track, tunnel etc.) | text |
|  | condition | Optimal/Needs Repairs/Inactive | text |
|  | Last inspected | When was the condition last checked | date |
|  | Inspected by | Which employee carried out the inspection. FK from table 6 – employees. | int |

Keeps track of every subway infrastructure.

Example with data:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Infrastructure ID | Infrastructure type | condition | Last inspected | Inspected by |
| 220 | Tunnel | optimal | 11.07.2025 | 2108 |

|  |  |  |  |
| --- | --- | --- | --- |
| Maintenance | Field name | Field Description | Data Type |
| Table 9 | Maintenance ID | PK | int |
| Infrastructure ID | Which object was maintained. FK from table 8 - Infrastructure | int |
|  | Employee ID | Which employees carried out the maintenance. FK from table 6- employees | int |
|  | Start date | When did the maintenance start. | date |
|  | End date | When it ended | date |
|  | cost | How much it cost | decimal |

Keeps track of every maintenance in the subway system.

Example with data:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Maintenance ID | Infrastructure ID | Employee ID | Start date | End date | cost |
| 35 | 99 | 146 | 01.09.2020 | 07.07.2021 | 77,000 |

|  |  |  |  |
| --- | --- | --- | --- |
| Passengers | Field name | Field Description | Data Type |
| Table 10 | Social number | PK | bigint |
| name |  | text |
|  | Last name |  | text |
|  | birthday |  | date |
|  | gender |  | text |

Keeps track of every passenger that has ever bought a discount ticket of our subway system.

Example with data:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Social Number | name | Last name | birthday | gender |
| 35001023369 | Mariam | Lomidze | 05.05.1999 | Female |

|  |  |  |  |
| --- | --- | --- | --- |
| Tickets | Field name | Field Description | Data Type |
| Table 11 | Ticket type id | PK | int |
| name |  | text |
|  | price |  | decimal |
|  | longevity | How many months from purchase is the ticket active | int |
|  | discount | How much of the full price is discounted | decimal |

Lists every type of discount ticket that we sell.

Example with data:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Ticket type id | name | price | longevity | discount |
| 2 | The medium ticket | 60 | 12 | 0.5 |

|  |  |  |  |
| --- | --- | --- | --- |
| Sales | Field name | Field Description | Data Type |
| Table 12 | Sale ID | PK | int |
| Ticket type id | Which ticket was sold. FK from table 11-tickets | int |
|  | date | When was the ticket sold | date |
|  | Social number | Social number of buyers. FK from table 10 - passengers | bigint |
|  | location | On which station was the sale made. Fk from table one - Stations | int |

Keeps track of every sale of discount tickets.

Example with data:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sale ID | Ticket type id | date | Social number | location |
| 25589 | 1 | 26.10.2025 | 445213975521 | 5 |

|  |  |  |  |
| --- | --- | --- | --- |
| Positions | Field name | Field Description | Data Type |
| Table 13 | Position ID | PK | int |
| Position name |  | text |
|  | Postion description |  | text |

Stores info about every position a subway worker could have.

Example with data:

|  |  |  |
| --- | --- | --- |
| Postion ID | Position name | Positoin description |
| 7 | Driver | Drives trains |