Business Description

Best Subway Company

Date: 2023-03-02

Prepared by:

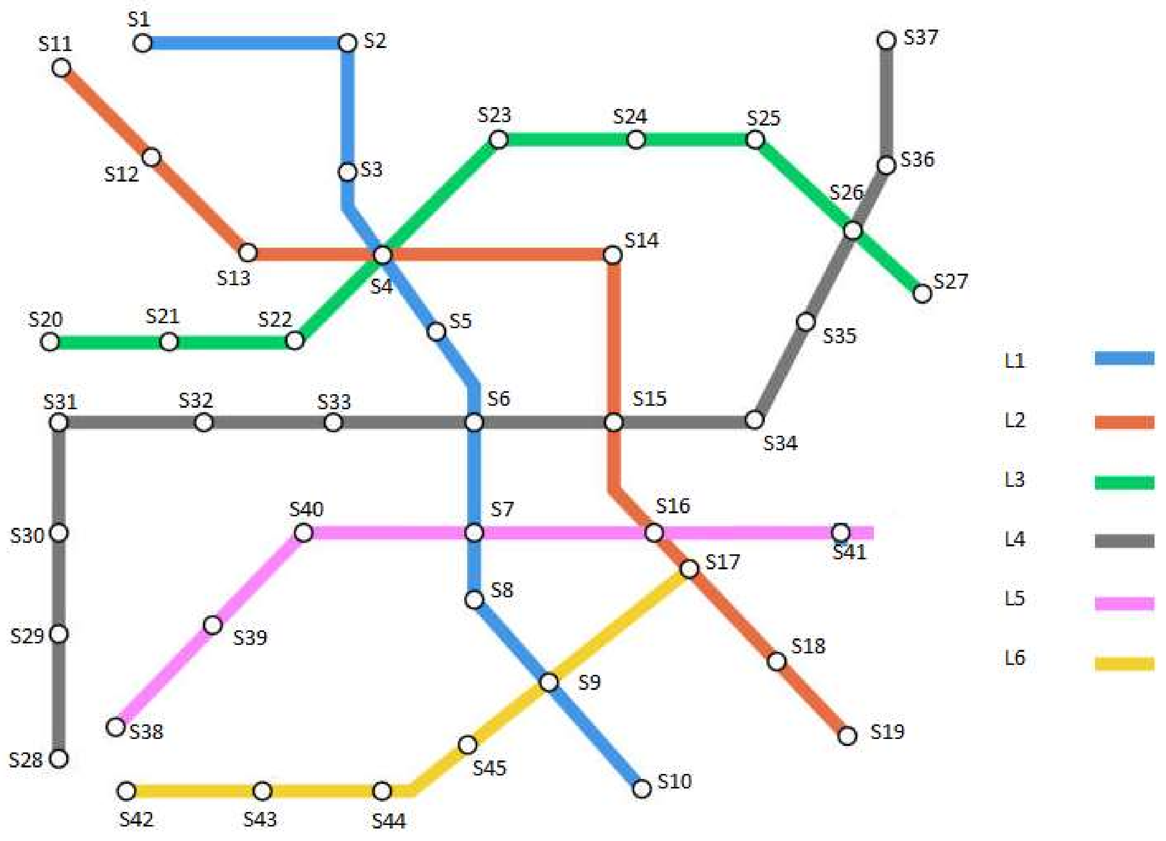
Barbara Kowalczyk

1. BUSINESS DESCRIPTION

1.1. Business background

Best Subway Company is a part of the City Transportation Authority in Warsaw. The company provides the biggest transportation network in the city, serving a population of over 200 thousand people across a vast area. The Best Subway Company comprises a fleet of 86 subway rail cars, 45 underground stations, and 200 km of rails.

The company has distinguished 6 different lines (from L1 to L6), which cross each other. Each line consists of several unique sections, meaning every section serves only one subway line (lines do not coincide).



The company hires a group of 37 ticket inspectors to check whether passengers in the vehicle have valid tickets and/or valid documents entitling them to a free or reduced fare. The tickets are checked around the clock, including nights, on all days of the week. Inspectors work always in a team meaning each control is conducted by a group of 2 to 4 inspectors. The composition of the team is not fixed and each week a new work schedule is created for every inspector according to their needs and current human resources.

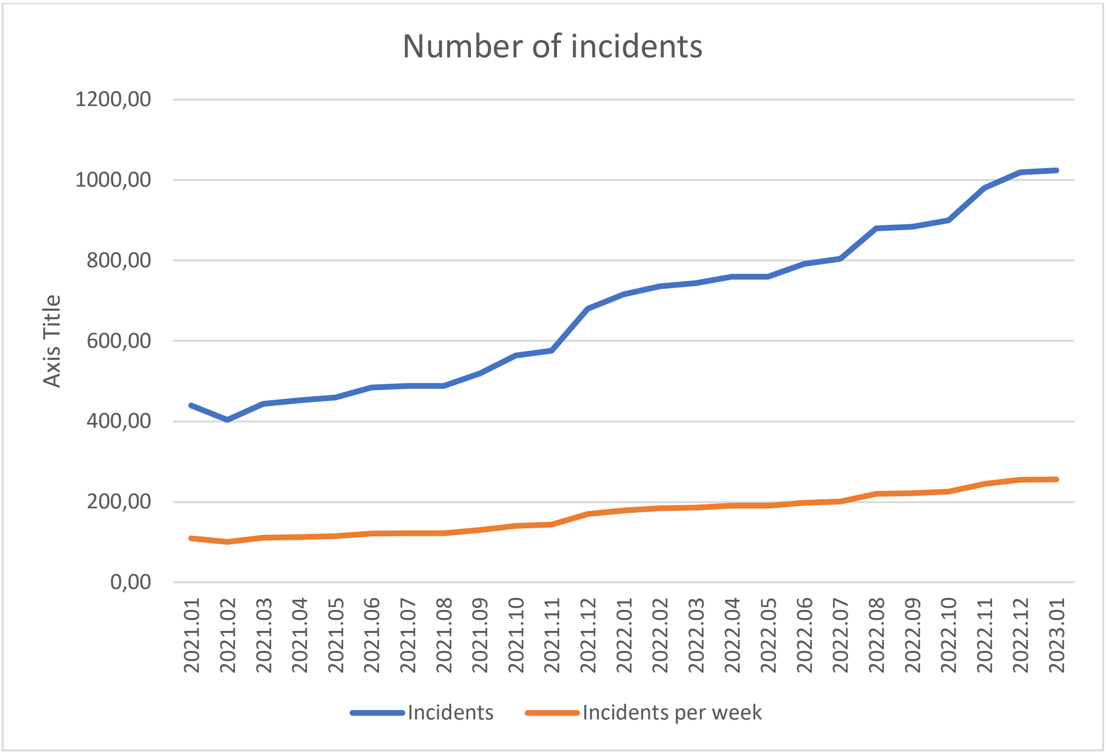
Every inspector is equipped with an official ticket inspector's ID card.

A passenger who does not have a valid ticket or a document entitling them to free travel is obliged to leave the rail car at the next station.

Passengers can buy the tickets in the ticket machines installed at the selected railway stations or out of the stations at the company's partner's newsstands.

1.2. Problems. Current Situation

Over the past 24 months, the number of incidents has significantly increased, when by *an incident* we understand *every separate case of a passenger not having a valid ticket*. Statistically, in January 2021 there were 100 incidents per week, whereas in January 2023 inspectors were issuing over 250 fines a week.



According to the estimation provided by the Company's management, passengers traveling without tickets lead to a loss of over 250.000 PLN net annually. At the same time, the possible income from the fines which should have been issued but were not (as the number of working inspectors is not sufficient) is estimated for over 500.000 PLN annually.

Hiring new employees will not solve the problem since every FTE is a cost of at least 80.000 PLN net annually.

The best solution is to re-think the way the Company and inspectors work and adjust it to reality based on the reliable data model.

1.3. Benefits of implementing a database. Project Vision

Assumptions

The Project scope assumes the creation of a dedicated database to allow monitoring of the number of incidents.

The data powering the database will come from the inspectors' activity. Each team of inspectors will be equipped with a tablet with dedicated software - ***Incident Base App***. This app will be used by all teams simultaneously in order to track every incident which takes place.

When an inspector notices a passenger without a ticket, populate dedicated fields in the app and **register the incident**. In addition, inspectors will gather data about the passenger and create an ID account for them. The account can be used by the inspector during any further incident caused by the particular passenger.

Moreover, the fine amount will depend on the number of incidents caused by the passenger. During the incident registration, an inspector checks the passenger ID in the *Incident Base App* and checks the history of the passenger. Based on that, they issue a fine for the amount depending on the number of former incidents committed by the passenger.

Each incident is entered into the *Incident Base App* and collected in the database.

Database description

The main table in the database will be an *Incident* table, which describes every case that occurred. It will contain all the details provided by the inspectors during an incident registration.

The *Incident* table will be linked to *the Passenger* table in order to relate the incident with a concrete person and easily find the recidivists.

The passenger's data and the *Unit* table will be added during registration meaning the database will not keep all the possible units (streets, cities, regions, countries) by default.

The database will be supplemented with additional entities in order to find patterns in passengers' behaviors. The new entities will cover aspects such as lines and sections, stations, and locations. These data will be added by the DB administrator and cannot be changed to any extent by the inspectors during the incident registration process.

Also, information about the inspectors will be contained to enable quick checks of their activity and performance.

Benefits

Aspects that can be investigated with the database include but are not limited to:

- The sections and lines with the highest number of incidents – locations (section/line/district) inspectors should focus on the most.

- The most common time when the incidents take place.
- Correlation between the incidents and the ticket machines' availability (where new ticket machines should be located in the first place)
- Group the most likely to avoid buying tickets.

Answering those questions would allow mitigating the losses by changing the inspectors' shifts and covering the riskiest lines in the first place. Moreover, it let to examine locations and chose the vital one for new ticket machine installation. Monitoring the passenger's data will be highly useful for marketing teams and help to prepare the campaigns addressed to a certain group of passengers.

In conclusion, collected data will help to improve the infrastructure and business way of working in order to provide more convenience for users, but also economically efficient services.

2. MODEL DESCRIPTION

2.1. Definitions & Acronyms

Business Definitions

Section	–	a unique part of the subway's railway system between two stations. Each section has a starting station and an ending station assigned. Currently, there are 48 sections (2 per each pair of stations). Each section is assigned one line.
Line	-	a unique group of sections connected by stations. Each line has also its starting and end stations.
Incident	-	every separate case of a passenger not having a valid ticket.
Incident base app	-	an app used by ticket inspectors to register incidents and create passenger accounts.
Area	-	a unique part of the city of Warsaw described by a district name and a street name.
Unit	-	a more general type of area used to describe the place of living of passengers. The unit can be a country, region, ,city and street within the city. Since the passengers can come from different locations (not only Warsaw city), the idea is to assign them to a street in a specific city (without describing the specific neighborhood). The Unit in combination with the house number and the rest of the passenger's data (name, date of birth) would let to identify the person.
DMF system	-	a system of describing the location of a point in the geographical net. The longitude and latitude are described by the number of degrees.

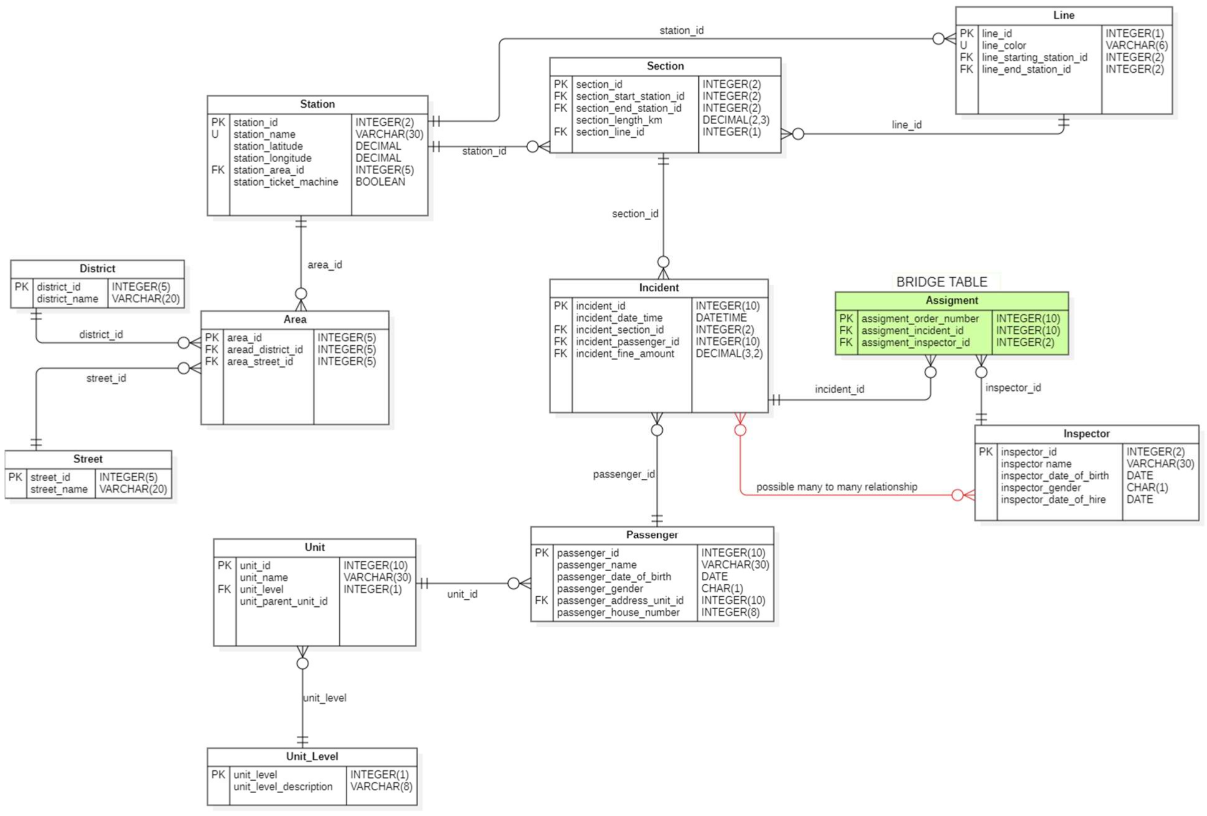
Database Definitions

Entity	-	the table in a database, which contains data referring to a specific object or concept, e.g., the entity Passengers contains data describing every single passenger.
Record	-	a row in a table.
Attribute	-	an attribute is a characteristic or a property of an entity that is being represented by the table. It is a column or a field that describes some aspect of the data that is being stored in the table, e.g., in a table that represents a passenger, attributes include name, address, and phone number. Attributes are stored in the columns.
Relation	-	(also relationship or link) is a connection or an association between the data in one table and the data in another table, usually based on a common field (called PK and FK) in the two tables. The relationship allows data to be retrieved or manipulated across the tables using queries or other database operations.
One-to-one	-	a type of relationship between two tables in which each record in one table is associated with at most one record in the other table.
One-to-many	-	a type of relationship between two tables in which each record in one table can be associated with one or many records in the other table, but each record in the second table can only be associated with one record in the first table.
Many-to-Many	-	a type of relationship between two tables in which each record in one table can be associated with one or many records in the other table, and vice versa. This means that for each record in the first table, there may be one or more corresponding records in the second table, and for each record in the second table, there may be one or more corresponding records in the first table.
Data Type	-	a classification of data that specifies the type of values that can be stored in a particular column of a table. The data type determines how the data is stored, processed, and retrieved, and it defines the set of operations that can be performed on the data.

Acronyms

PK	-	Primary Key – a unique identifier that is used to identify a record or a row in a database table.
FK	-	Foreign Key - a field in a database table that references the primary key of another table, establishing a link between the two tables.
AK	-	Alternate Key – a column with unique values, which could act as a Primary Key.
N	-	Nullable – column which may keep the null values (e.g., when a passenger does not agree for sharing his personal data, an inspector leaves the relevant field in the Incident base App empty – that will cause the null value occurs in the relevant record in the database).

2.2. Logical Scheme


station_id
Line
PK line_id INTEGER(1)
U line_color VARCHAR(6)
FK line_starting_station_id INTEGER(2)
FK line_end_station_id INTEGER(2)
Section
PK section_id INTEGER(2)
FK section_start_station_id INTEGER(2)
FK section_end_station_id INTEGER(2)
section_length_km DECIMAL(2,3)
FK section_line_id INTEGER(1)
line_id
Station
PK station_id INTEGER(2)
U station_name VARCHAR(30)
station_latitude DECIMAL
station_longitude DECIMAL
FK station_area_id INTEGER(5)
station_ticket_machine BOOLEAN
station_id
section_id
area_id
District
PK district_id INTEGER(5)
district_name VARCHAR(20)
district_id
street_id
Area
PK area_id INTEGER(5)
FK aread_district_id INTEGER(5)
FK area_street_id INTEGER(5)
Street
PK street_id INTEGER(5)
street_name VARCHAR(20)
Incident
PK incident_id INTEGER(10)
incident_date_time DATETIME
FK incident_section_id INTEGER(2)
FK incident_passenger_id INTEGER(10)
FK incident_fine_amount DECIMAL(3,2)
BRIDGE TABLE
Assigment
PK assigment_order_number INTEGER(10)
FK assigment_incident_id INTEGER(10)
FK assigment_inspector_id INTEGER(2)
incident_id
inspector_id
Inspector
PK inspector_id INTEGER(2)
inspector name VARCHAR(30)
inspector_date_of_birth DATE
inspector_gender CHAR(1)
inspector_date_of_hire DATE
passenger_id
possible many to many relationship
Unit
PK unit_id INTEGER(10)
unit_name VARCHAR(30)
FK unit_level INTEGER(1)
unit_parent_unit_id
unit_id
Passenger
PK passenger_id INTEGER(10)
passenger_name VARCHAR(30)
passenger_date_of_birth DATE
passenger_gender CHAR(1)
FK passenger_address_unit_id INTEGER(10)
passenger_house_number INTEGER(8)
unit_level
Unit_Level
PK unit_level INTEGER(1)
unit_level_description VARCHAR(8)

2.3. Objects

Table Description

The below table contains the description of all the entities (tables) designed in the database.

Each table consists of several columns (attributes).

Each attribute has a specific data type declared to ensure the data's consistency, integrity, and accuracy in the database.

No	Entity Name	Name of Attribute	Attribute Description	Data Type
1	Section	section_id	The unique name of each section in the metro system e.g. 1,2,3 PK	int(2)
		section_start_station_id	The ID number of the assigned starting station, e.g. 1,2,3,…,45 FK	int(2)
		section_end_station_id	The ID number of the assigned ending station, e.g. 1,2,3,…,45 FK	int(2)
		section_length_km	Length of the railway track between start station and end station in kilometers; rounded to 3 decimal places, e.g. 25,124	decimal(2,3)
		section_line_id	The ID of the Line, the section is assigned to. FK	int(1)
2	Line	line_id	The unique ID of the existing line, e.g. 1,2,3. PK	int(1)
		line_colour	The assigned color of the line, e.g. blue, red. U, AK	varchar(6)
		line_starting_station_id	The ID number of the assigned starting station, e.g. 1,2,3,…,45 FK, U	int(2)
		line_end_station_id	The ID number of the assigned ending station, e.g. 1,2,3,…,45 FK, U	int(2)
3	Station	station_id	The unique ID of the railway station, e.g. 1,2,3,…,45 PK	int(2)
		station_name	The unique name of the railway station e.g. "Warsaw Center". U, AK	varchar(30)
		station_latitude	Stations latitude written in the DMF system, e.g. 50.35078940622989	decimal(20)
		station_longitude	Stations longitude written in the DMF system, e.g. 18.90965635687011	decimal(20)
		station_area_id	The ID of the city area where the station is located, eg. 1,2,3…999 FK	int(5)

		station_ticket_machine	Defines whether the station has a ticket machine installed. It can take only 2 values: - TRUE when a station has a ticket machine - FALSE when a station does not have a ticket machine.	boolean
4	Area	area_id	The unique ID of every area distinguished within the city, e.g. 1,2,3…999 PK	int(5)
		area_district_id	The district ID number assigned to the area, e.g. 1,2,3…999 FK	int(5)
		area_street_id	The street ID number that is assigned to the area. e.g. 1,2,3..999 FK	int(5)
5	District	district_id	Unique district ID e.g. 1,2,3…999 PK	int(5)
		district_name	The name of the district with the given ID.	varchar(20)
6	Street	street_id	Unique street ID e.g. 1,2,3,…,999 PK	int(5)
		street_name	The name of the street with the given ID.	varchar(20)
7	Incident	incident_id	The unique ID number of every incident that occurred, including historic data, e.g. 1,2,3,…99999999 PK	int(10)
		incident_date_time	The time and date when the incident took place e.g. 5/15/2023 8:30:52 AM	datetime
		incident_section_id	The section ID in which the incident occurred e.g/ 1,2,3 FK	int(2)
		incident_passenger_id	The passenger ID number e.g. 123456 FK	int(10)
		incident_fine_amount	The amount of the fine issued in PLN e.g. 100.00, 250.00.	decimal(3,2)
8	Passenger	passenger_id	The unique number given to a passenger without a ticket, e.g. 123456 PK	int(10)
		passenger_name	First and Last Name of the passenger e.g. Jan Kowalski	varchar(30)

		passenger_date_of_birth	Passenger's date of birth, e.g. 1990-12-15	date
		passenger_gender	Passenger's gender - female/male/diverse, e.g. F, M, D	char(1)
		passenger_address_unit_id	The ID of the lowest level unit (street) the passenger lives in - or next to in case the passenger's city of living is not listed, e.g. PL123, DE1006 FK	int(10)
		passenger_house_number	Passenger house number e.g. 1, 15, 1234	int(8)
9	Unit	unit_id	The unique number given to each independent unit, e.g. PL123, DE1006. PK	int(10)
		unit_name	The name of the unit, e.g. *Poland, Silesia, Katowice, Mariacka*	varchar(30)
		unit_level	The level at which a given unit is placed in the administrative structure. There are 4 levels considered: - country-level 1 - region-level 2 - city-level 3, - street-level 4 e.g. 1,2,3,4 FK	int(1)
		unit_parent_unit_id	The unit_id of the unit that is placed above the considered unit in the administrative hierarchy, for instance for London (city) the parent unit is Greater London (region), e.g. PL123, DE1006.	varchar(10)
10	Unit_Level	unit_level	The unique level corresponding to the administrative hierarchy - one of four: - country-level 1 - region-level 2 - city-level 3, - street-level 4 e.g. 1,2,3. PK	int(1)
		unit_level_description	Textual description of each level, e.g. City, Region, Country.	varchar(8)

11	Assignment (link table)	assigment_order_number	The order number assigned to every tuple - serves as a PK for an Assignment table, e.g. 1,2,3 PK	int(10)
		assigment_incident_id	The ID number of the incident that occurred, e.g. INC1, INC2….INC12345 FK	int (10)
		assigment_inspector_id	The ID number of the inspector involved in the considered incident, e.g. INS_01, INS_30 FK	int(2)
12	Inspector	inspector_id	The unique inspector's ID number, e.g. INS_01, INS_30 PK	int(2)
		inspector name	First and Last Name of the inspector e.g. Anna Nowak	varchar(30)
		inspector_date_of_birth	Inspector's date of birth, e.g. 1990-12-15	date
		inspector_gender	Inspector's gender - female/male/diverse, e.g. F, M, D	char(1)
		inspector_date_of_hire	Inspector's date of hire, e.g. 1990-12-15	date

Tables Relations Comments

Entities are linked to each other in a logical way. There are different types of relationships, such as one-to-one, one-to-many, and many-to-many, which determine how the data is related and how it can be accessed.

In the schema, the many-to-many relationship occurs between the *Incident* table and *Inspector* table since each incident could be assigned to many inspectors and each inspector can be linked to many incidents. Therefore, a link table was designed – *Assignment. The Assignment* table includes every possible pair of inspectors and the incidents, they attend.

This and every other relation appearing in the database are described in the table below. The order number of the relations is also added to the ER diagram:

No	Type of relationship	Entity	Attribute	Description
1	one-to-many	Incident	incident_section_id	Every incident takes place in one section. One section is a place where many incidents can occur.
		Section	section_id	
2	one-to-many	Section	section_line_id	Every line consists of many sections, but every section is a part of only one line.
		Line	line_id	
3	one-to-one	Line	line_starting_station_id	Every line has only one starting station, and every station is a starting point for specifically one line.
		Station	station_id	
4	one-to-one	Line	line_end_station_id	Every line has only one end station, and every station is an endpoint for specifically one line.
		Station	station_id	
5	one-to-many	Station	station_area_id	One station can have only one area assigned, but one area can be vast enough to have more stations on its surface.
		Area	area_id	
6	one-to-many	Area	area_district_id	Every area can have only one district name assigned, while the district can cover several areas.
		District	district_id	
7	one-to-many	Area	area_street_id	Every area can have only one street name assigned, while the street can be long enough to run through several areas.
		Street	street_id	
8	one-to-many	Incident	incident_passenger_id	Every incident was committed by one single passenger, but every passenger can be responsible for many incidents.
		Passenger	passenger_id	
9	one-to-many	Passenger	passenger_unit_id	Every passenger lives in one unit (city/street), but one unit can be a place of living for many passengers (few people can live on one street od the city).
		Unit	unit_id	
10	one-to-many	Unit	unit_level	Each unit has one level (e.g. Poland - level 4, Warsaw - level -2), but each level is assigned to many units (e.g. level 4 stands for countries, meaning Poland, Belarus, Germany and others have the same level).
		Unit_level	unit_level	
11	one-to-many	Incident	incident_id	A supporting connection. Every incident has its own combination of inspectors, who took place in it.
		Assignment	assignment_incident_id	
12	one-to-many	Assignment	assignment_inspector_id	A supporting connection. Every inspector participated in several incidents.
		Inspector	inspector_id	
13	many-to-many (resolved)	Incident	incident_inspector_id	Every incident is conducted by a group of inspectors, and every inspector was involved in several incidents. This relation has been split into 2 one-to-many relations (11 and 12).
		Inspector	inspector_id	
14	one_to_many	Unit	unit_id	The specific relationship between two tuples (rows) in one table. In the table, all kinds of units are listed (country, region, city, street). To every unit, a parent unit is assigned e.g. to region is a parent unit for the city.
			unit_parent_unit_id	

My approach to modeling step-by-step

I based on the E-R technique and followed the steps described in the learning materials.

I could distinguish several steps taken during the modeling process:

1. Background and problem design
   The creation of a comprehensive business idea was like a pre-work exercise. I read a lot about the chosen business specifics (subway) and try to find a precise problem that can apply to it.
   After that, I thought about what could be the current state – what my imagined organization has and what is outside its scope. What data could the subway management board members already have? Do they keep it somewhere? If yes, where? In an old version of the database or maybe in a binder? What is the source of the data the company has? Who (or what) produced it?
   To summarize, I consider this part as a thought element, since everything I form at the beginning determines the proposed solution.

2. Proposed the solution
   Knowing the background, I could propose a solution. The first step here was to define what kind of data to gather. What data could be useful for solving the established problem? Not everything is important, so I needed to set a border somewhere. That was quite difficult since at the beginning I tended to collect every piece of information After listing down everything which came to my mind, I had to rethink the whole set of data and remove all redundant entities.
   When I had the data needed to solve the issue, I thought who will collect that data and how? Who will upload them into the database? I know what information I need, but how to get it?
   So, I create a solution (The Incident Base App) and define how it will work.

3. Creation of the database
   This process was also split into smaller steps.
   a. Creating the entities and their attributes – I had already known what information I want to collect so the only activity here was to create logical entities (separate entities for every object/concept)
   b. Marking the primary and foreign keys
   c. Set the relationships between tables – I noticed one many-to-many relation, hence the additional step here was to create a link table.
   d. Declaration of the data type – since the data type depends on the actual signs we keep in a cell/column, I also proposed concrete ways how the data will look (for instance the metro lines id would be described by letter "L" and numbers from 1 to 6).