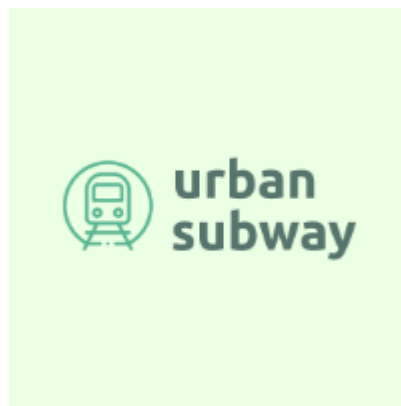




URBAN SUBWAY



Legal Notice:

This document contains privileged and/or confidential information and may not be disclosed, distributed or reproduced without the prior written permission of EPAM®.

Confidential

CONTENTS

1	BUSINESS DESCRIPTION	3
1.1	Business background	3
1.2	Problems. Current Situation	3
1.3	the Benefits of implementing a database. Project Vision	3
2	MODEL DESCRIPTION	4
2.1	Definitions & Acronyms	4
2.2	Logical Scheme	4
2.3	Objects	5

1 BUSINESS DESCRIPTION

1.1 BUSINESS BACKGROUND

Urban Subway is a metropolitan public transportation company responsible for managing an extensive network of underground train lines connecting key districts of the city. Every day, thousands of passengers rely on its services for commuting, and the system must coordinate complex interactions between stations, trains, schedules, and ticketing systems.

1.2 PROBLEMS. CURRENT SITUATION

Historically, operational data—such as train schedules, passenger ticketing, and station management—has been managed through separate subsystems or partially manual methods. This fragmentation has made it difficult to synchronize real-time information across departments. As the network expands and passenger volumes increase, inefficiencies have emerged in tracking train operations, validating tickets, and ensuring accurate scheduling.

To sustain future growth and improve service reliability, Urban Subway aims to centralize all operational data within an integrated relational database. This system will serve as a unified foundation for digital transformation, supporting advanced analytics, passenger management, and real-time monitoring.

Currently, data about passengers, trains, and tickets is stored across multiple independent files or applications. This causes:

- **Inconsistent information** between ticketing, scheduling data.
- **Difficulties in monitoring** train movements and performance metrics.
- **Delayed reporting** due to manual data consolidation.
- **Limited scalability** as operations and passenger volumes grow.

These inefficiencies lead to service delays, operational errors, and poor decision-making based on outdated or incomplete information.

1.3 THE BENEFITS OF IMPLEMENTING A DATABASE. PROJECT VISION

Implementing the **Urban Subway Database** provides a unified and efficient system for managing all operational, passenger, and scheduling data. The key benefits include:

- **Centralized Data Management** – All information about passengers, trains, stations, and tickets is stored in one consistent system, reducing duplication and data inconsistencies.
- **Enhanced Decision-Making** – Reliable, real-time data supports performance monitoring, demand forecasting, and informed management decisions.
- **Data Integrity and Accuracy** – Enforced relationships and constraints ensure data consistency across departments, improving reliability.

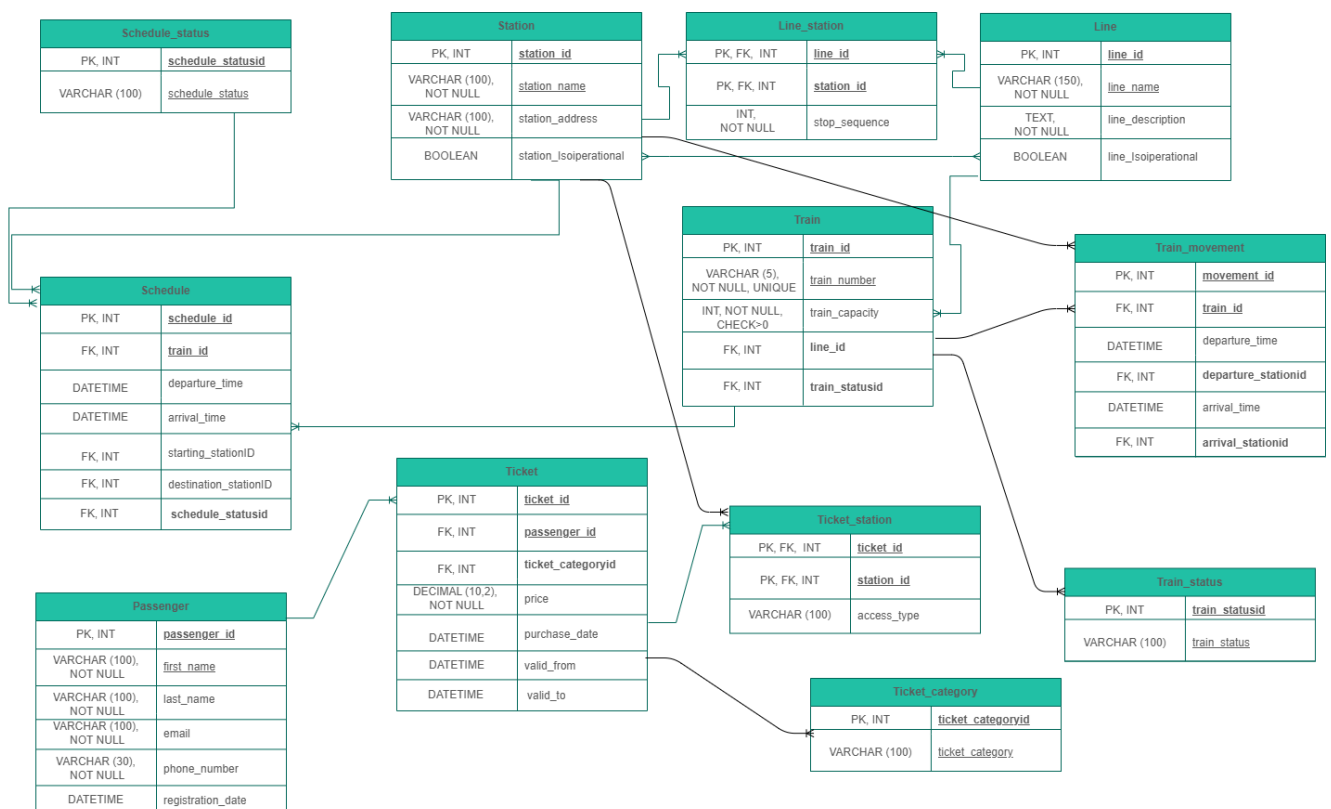
- **Foundation for Future Automation** – The structured database will support integration with digital tools such as mobile ticketing, dashboards, and predictive analytics.

2 MODEL DESCRIPTION

2.1 DEFINITIONS & ACRONYMS

- **PK** – Primary Key
- **FK** – Foreign Key

2.2 LOGICAL SCHEME



2.3 OBJECTS

1. STATION

Table Name	Field name	Field Description	Data Type
Station	station_id	Unique station identifier, PK	INT
	station_name	Name of the station	VARCHAR (100), NOT NULL
	station_address	Address of the Station	VARCHAR (100), NOT NULL
	station_Isoperational	Whether the station is active or not	BOOLEAN

Relationship: Station (M) - Line (M), bridge table Line_station

Station (1) - Ticket (M)

Station (1) - Schedule (M)

Station (1) - Train_movement (M)

Example with data

station_id	station_name	station_address	station_Isoperational
1	Central Square	1 Republic street	Active

2. LINE

Table Name	Field name	Field Description	Data Type
Line	line_id	Unique subway line identifier, PK	INT
	line_name	Name of the line	VARCHAR (150), NOT NULL
	line_description	Description of the Line	Text, NOT NULL
	line_Isoperational	Whether the line is active or not	BOOLEAN

Relationship: Station (M) - Line (M), bridge table Line_station

Line (1) - Train (M)

Example with data

line_id	line_name	line_description	line_Isoperational
1	Red line	New extension connecting nothern districts	Active

3. LINE_STATION

Table Name	Field name	Field Description	Data Type
Line_station	line_id	Connected line, PK, FK	INT
	station_id	Connected station, PK, FK	INT
	stop_sequence	Order of the station in the line	INT, NOT NULL

Relationship: Station (1)- (M) Line_station (M) - Line (1)

Example with data

line_id	station_id	stop_sequence
1	1	7

4. TRAIN

Table Name	Field name	Field Description	Data Type
Train	train_id	Unique train identifier, PK	INT
	train_number	Train number/code	VARCHAR (5), NOT NULL, UNIQUE
	train_capacity	Passenger capacity	INT, NOT NULL, CHECK >0
	line_id	Line the train serves, FK	INT
	train_statusid	Train current status, FK	INT

Relationship: Train (M) - Line (1)

Train (1) - Train_status (M)

Train (1) - Train_movement (M)

Train (1) - Schedule (M)

Example with data

train_id	train_number	train_capacity	line_ID	train_statusid
1	M056	250	5	1

5. TRAIN_STATUS

Table Name	Field name	Field Description	Data Type
Train_status	train_statusid	Unique train status identifier, PK	INT
	train_status	Train current status (In Transit, Idle, Maintenance)	VARCHAR (100)

Relationship: Train (1) - Train_status (M)

Example with data

train_statusid	train_status
1	In Transit

6. SCHEDULE

Table Name	Field name	Field Description	Data Type
Schedule	schedule_id	Unique schedule identifier, PK	INT
	train_id	Assigned train, FK	INT
	departure_time	Scheduled start time	DATETIME
	arrival_time	Scheduled arrival time	DATETIME
	starting_stationid	Starting station, FK	INT
	destination_stationid	Final station, FK	INT
	schedule_statusid	shchedule status, FK	INT

Relationship: Train (1) - Schedule (M)

Station (1) - Shchedule (M)

Shchedule_status (1) - Schedule (M)

Example with data

schedule_id	train_id	departure_time	arrival_time	starting_stationid	destination_stationid	schedule-statusid
205	1001	2025-01-02 09:15:00	2025-01-02 09:45:00	1	3	1

7. SCHEDULE_STATUS

Table Name	Field name	Field Description	Data Type
Schedule_status	schedule_statusid	Unique schedule status identifier, PK	INT
	schedule_status	Schedule status (On time, delayed, completed, canceled)	VARCHAR (100)

Relationship: Schedule_status (M) - Schedule (1)

Example with data

schedule_statusid	schedule_status
1	On Time

8. PASSENGER

Table Name	Field name	Field Description	Data Type
Passenger	passenger_id	Unique passenger identifier, PK	INT
	first_name	Passenger first name	VARCHAR (100), NOT NULL
	last_name	Passenger last name	VARCHAR (100), NOT NULL
	email	Passenger email	VARCHAR (100), NOT NULL
	Phone_number	Passenger phone number	VARCHAR (30), NOT NULL
	registration_date	Passenger registration date	DATETIME

Relationship: Passenger (1) - Ticket (M)

Example with data

passenger_id	First_name	last_name	email	phone_number	registration_date
10258	Liana	Sahakyan	lianasahakyan@gmail.com	+37477622183	2025-03-18 08:17:32

9. TICKET

Table Name	Field name	Field Description	Data Type
Ticket	ticket_id	Unique ticket identifier, PK	INT
	passenger_id	Unique passenger identifier, FK	INT
	ticket_categoryid	Unique ticket type, FK	INT
	price	Ticket price,	DECIMAL (10,2), NOT NULL
	purchase_date	Ticket purchase date	DATETIME
	valid_from	Ticket valid first date	DATETIME
	valid_to	Ticket valid last date	DATETIME

Relationship: Ticket (M) - Passenger (1)

Ticket (M) - Station (M), bridge table Ticket_station

Ticket (1) - Ticket_category (M)

Example with data

ticket_id	passenger_id	ticket_categoryid	price	purchase_sate	valid_from	valid-to
56894	1001	1	3,55	2025-01-02 09:45:00	2025-01-02 09:45:00	2025-02-02 09:44:59

10. TICKET_CATEGORY

Table Name	Field name	Field Description	Data Type
Ticket_category	ticket_categoryid	Unique ticket identifier, PK	INT
	ticket_category	Ticket type (Single, Monthly, Multi, Yearly)	VARCHAR (100)

Relationship: Ticket_category (M) - Ticket (1)

Example with data

ticket_categoryid	ticket_category
1	Single

11. TICKET_STATION

Table Name	Field name	Field Description	Data Type
Ticket_station	ticket_id	Related ticket, PK, FK	INT
	station_id	Station allowed, PK, FK	INT
	access_type	Valid direction (Entry, Exit, Both)	VARCHAR (100)

Relationship: Ticket (1)- (M) Ticket_station (M) - Station (1)

Example with data

ticket_id	station_id	access_type
10528	3	Both

12. TRAIN_MOVEMENT

Table Name	Field name	Field Description	Data Type
Train_Movement	movement_id	Unique movement identifier, PK	INT
	train_id	Train moving, FK	INT
	departure_time	When train departed	DATETIME
	Departure_stationid	Starting station , FK	INT
	arrival_time	When train arrived	DATETIME
	arrival_stationid	Station reached , FK	INT

Relationship: Train_Movement (M) - Station (1)

Train_Movement (M) - Train (1)

Example with data

movement_id	train_id	Departure_time	Departure_stationid	arrival_time	arrival_stationid
5680	260	2025-17-04 09:45:00	8	2025-17-04 09:55:00	9