



Project Report | Documentation

MINI PROJECT Metro System Database Project

NITIN KUMAR
119CS0031

KANISHK KUNAL
119CS0015

This page is intentionally left blank

Indian Institute of Information Technology Design and Manufacturing, Kurnool.

(An institute of National importance)
Kurnool – 518007 (India)



Department of Computer Science & Engineering

MINI PROJECT Metro System Database Project

Submitted by : Kanishk Kunal(Roll no. 119CS0015)
Nitin Kumar(Roll no. 119CS0031)

Submission Date : June 5, 2021

Supervisor : Dr. R. PraneethaSree
Department of Computer Science & Engg.
IIITDM, Kurnool (Andhra Pradesh, India)

Subject : Database Management System Laboratory
CS205P, IV Semester (B.Tech. CSE)

Session : Even Semester 2021

This page is intentionally left blank

PREFACE

This project title **Metro System Database Management Project** is one of the academic projects we (as a team of 2 members) take in Database Management System Laboratory course (4th Semester) of Bachelor in Technology (Computer Science & Engineering), here at IIIT, Kurnool. This project was started in May 2021 and merely ends at the submission date by May 29. (Work in off days).

The complete report is organized into small chapters, sections, subsections, Appendices, etc. Snapshots and screenshots are taken for clear idea of what's going. We've tried to cover all information regarding project to be assemble in report. Further they can be merged into report in future as per requirements. The foundations of project started in January, when new session starts and topic of project decided. Some points such that project management, scalability, risk analysis, version control, aims, etc. were discusses primarily. The major aims set at beginning of this project was -

- ✓ To gain the technical knowledge and experience and practical hands Behind database, their management systems, and other technologies.
- ✓ A clear idea of working title incorporating traditional accent but with some modern software engineering principles, practices, and standards.
- ✓ Improving team skills under fully supervision (Project management Including risk analysis and Soft-skills).
- ✓ To present a final result implementation, its analysis, and evaluation with proper documentation/project report.

This document is neither a fully formal documentation nor a research based report, but it inherits many attributes of both in it. Here are some highlights of the project -

- ❖ A single document having combination of documentation, project report, reference manual, tutorial, user manual for the output.
- ❖ Practical implementation of Database System use for Metro system
- ❖ Parallel implementation of Database in many Databases platforms, here we Uses Oracle database workbench and WPS Office/VSC Base as major two platform using SQL as major query language and writing documentation.
- ❖ Incorporation of Software Engineering elements such as principles, Practices, life-cycle, etc.

Acknowledgement

With due respect, we express our deep sense of gratitude to our respected and guide, coordinator, supervisor, instructor **Dr. R PraneethaSree (PhD)**, for her valuable help and guidance. We are thankful for the encouragement and motivation that she has given us in completing this project successfully. Her rigorous evaluation and constructive criticism was of great assistance. It is imperative for us to mention the fact that this project could not have been accomplished without the periodic suggestion and advice of our supervisor.

We are also grateful to our respected director **Dr. DVNL Somayajulu (PhD)** for permitting us to utilize all the necessary facilities of the college.

We are thankful to all the other faculties, Professors, Associate Professors, Assistant Professors, staff-members, teaching assistants, laboratory attendants, seniors, and our fellow branch mates of our work culture for their kind co-operation, periodic evaluation, help, and support. We thank all others who directly/indirectly involved in this project and make it a successful. We also thank all other teams for maintaining an active communication aside of healthy competition and sportsman spirit.

We would also like to express our deep appreciation towards our family members for making us such kind to do ‘more’ by providing the much needed support and encouragement from all aspects of life. At last, we recall our gratitude to the one eternal almighty: The God.

Abstract

A Metro System is one which helps us to **maintain an organizational data** **Regarding a metro system**. Since it is very difficult to maintain a record of data items and this information manually or in file system, we are **creating a database for the complete management of resources and tasks** of metro system.

A Metro System database is one which is used for operational ease in place of traditional file system or Excel worksheets for information storage and retrieval with customized user queries implemented in SQL and is a fully functionally required software prototype with proper user interface and some random software engineering standards.

Organizing database for data collection and retrieval helps us to maintain data easily. We design it with the help of [Entity-Relationship diagram](#), UML diagrams, data flow diagrams, and using tables & proper User interfaces. It consists of several steps to be followed and many web pages are to be maintained simultaneously. It is a working prototype of a database schema which can further be enhanced.

For main functionality, we have to maintain many records such as **metro numbers, metro route, departure time, arrival time, metro type, fare, managements, department, customers**, etc. When we want to travel we reserve seat (Additional feature for Metro System but it is not compulsory for all passengers to reserve seats for them, they can still check in at last moment in case of availability in first in first serve order), for this we have to fill the reservation form which includes entries such as metro number, metro name, departure time, arrival time, route, metro type. So, we should gather all the information regarding all these.

Keywords: Metro System, Railway, Database, Prototype, Software Engineering principles

DATABASE ENVIRONMENT SYSTEM

A database management system (DBMS) is a collection of programs that **enables users to create and maintain a database**. The DBMS is a general-purpose software system that facilities the processes of defining, constructing, manipulating and sharing databases among various users and applications.

Defining a database involves specifying the data types, structures and constraints of the data to be stored in the database.

The database definition or description information is also stored by the DBMS in the form of a database catalog or dictionary, it is called **Metadata**. Constructing the database is the process of storing the data on some storage medium that is controlled by the DBMS. Manipulating a database includes functions such as querying the database to retrieve specific data, updating the database to reflect changes in the mini world and generating reports from the data. Sharing a database allows multiple users and programs to access the database simultaneously.

An **application program accesses the database by sending queries or requests for data to DBMS**. A query typically causes some data to be retrieved; a transaction may cause some data to be read and some data to be written into the database.

Other important functions provided by DBMS include **protecting the database and maintaining it over a long period of time**, protection includes system protection against hardware or software malfunction and security protection against unauthorized or malicious access.

A typical large database may have a life cycle of many years, so the DBMS must be able to maintain the database system by allowing the system to evolve as requirements change over time.

It is not absolutely necessary to use **general-purpose DBMS** software to implement a computerized database. We could write our own set of programs to create and maintain the database, in effect creating our own special purpose DBMS software. In either case, whether we use a general-purpose DBMS or not we usually have deployed a considerable amount of complex software.

Advantages of using DBMS approach

Using a DBMS to manage data has many advantages:

- **Data independence:** application program should not, ideally, be expected to details of data representation and storage, the DBMS provides an abstract view of the data that hides such details.
- **Efficient Data Access:** A DBMS utilizes a variety of sophisticated techniques to store and retrieve data efficiently. This feature is especially important if the data is to be stored on an external device.
- **Data integrity and security:** if data is always accessed through DBMS, the DBMS can enforce integrity constraints. For example, before inserting salary information for an employee, the DBMS can check that the department budget is not exceeded. Also, it can enforce access controls that govern what data is visible to different classes of users.
- **Data Administration:** when several users share data, centralizing the administration of data can offer significant improvements. Experienced professionals who understand the nature of the data being managed, and how different groups of users use it, it can be responsible for organizing the data representation to minimize redundancy and for fine-tuning the storage of the data to make retrieval efficient.
- **Concurrent Access and Crash Recovery:** A DBMS schedules concurrent accesses to the data in such a manner that users can think of the data as being accessed by only one user at a time. Further, the DBMS protects users from the effects of system failures.
- **Reduced Application Development Time:** clearly, the DBMS supports important functions that are common to many applications accessing data in the DBMS. This, in conjunction with the high-level interface to data, facilities quick application development. DBMS applications are also likely to be more robust than a similar stand-alone application because many important tasks are handled by the DBMS.

Problem Statement

The major task is to make a functional Database system for a Metro Management for a network using oracle database workbench for writing SQL queries and codes. WPS editor is used for documentation and making ER diagrams. User Interfaces are built for proper handling of user queries and data entrance. The project is prepared with the help of standard software engineering techniques, software project management guidelines, proper documentation & report preparation, etc.

Tasks :

- Planning and Analysis of commercially available software on metro system
- Database implementations
- Standard testing of all implementations
- Performance analysis of all implementations (to evaluate all implementations)
- Providing a user interface for end & front users, administrators, etc.
- Make a fully functional prototype.

SOFTWARE AND HARDWARE REQUIREMENTS

Minimum System/Software/Hardware requirements :

- ❖ Windows 7/Linux 3/Mac OS/Chromium OS
- ❖ Memory requirements: 64 MB (RAM).
- ❖ Recommend 128 MB.
- ❖ Secondary memory requirements (Hare disk): 10 GB (ROM).
- ❖ Recommend 256/512 GB
- ❖ Oracle MySQL Workbench/LibreOffice Base
- ❖ Recommend MySQL Workbench/Command Line Client
- ❖ Writer and diagram designing software such as MS Vision or LibreOffice Draw.
- ❖ Clock Speed: 866 MHz
- ❖ Virtual Memory: 32 bits
- ❖ Cache Memory: 512KB, etc

Methodology

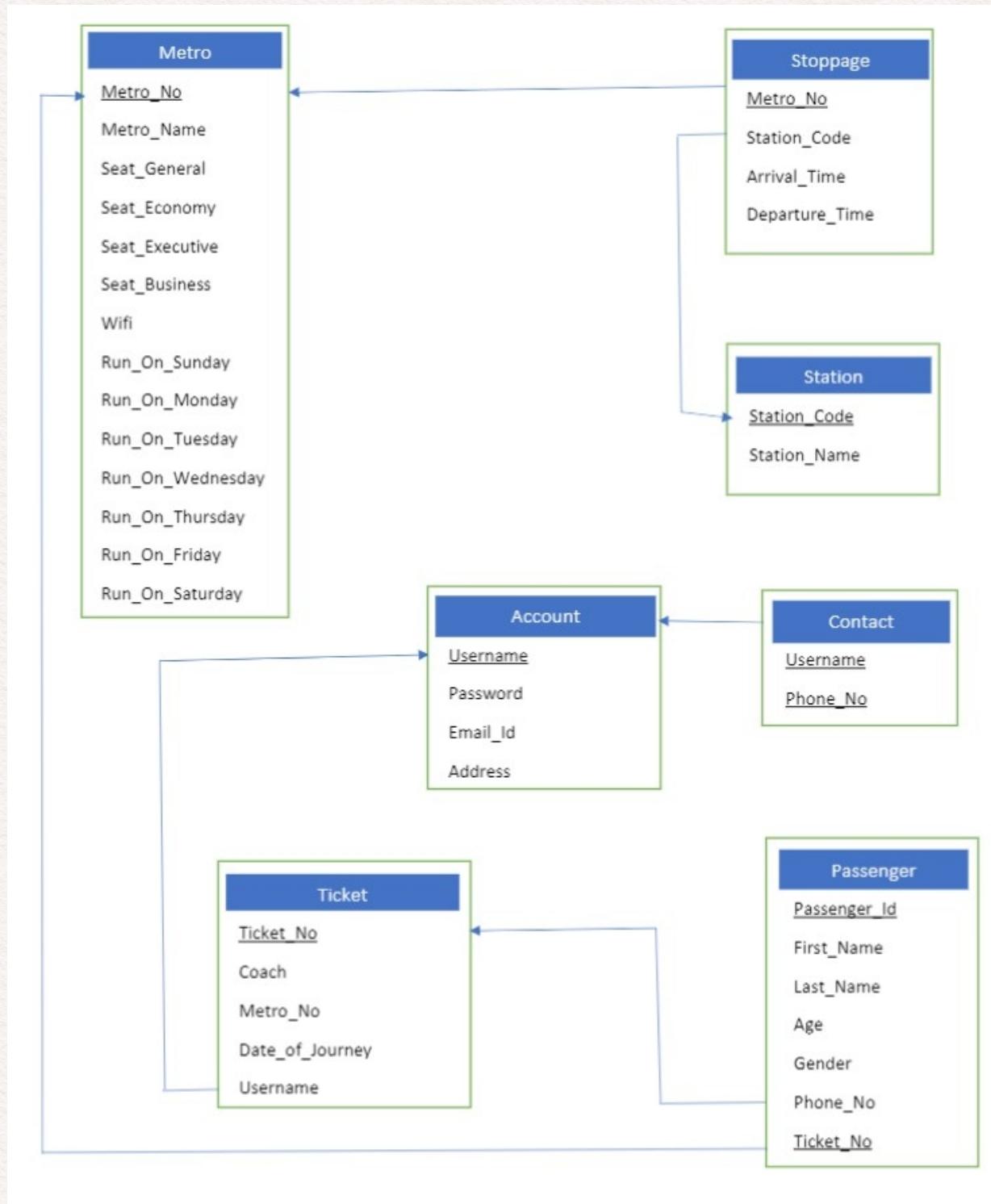
The methodology adopted in the project is simple and is easy to implement. **Functions and Object oriented methodology** constitute a major portion of methodology. Project is divided into individual tasks. Tasks are then taken as initial problem and solved individually and have to integrate them at last. Objectives are set first and then we tried to fulfill all objectives asserted.

We've first implemented preliminary Entity-Relationship diagram in 2 platforms namely- MySQL Workbench (Windows 10) and WPS editor. We then intuitively analyzed which platform is more suitable for this particular schema and required functionality. After that we enhance that schema to further more relations, entities, relationships, forms, queries, etc.

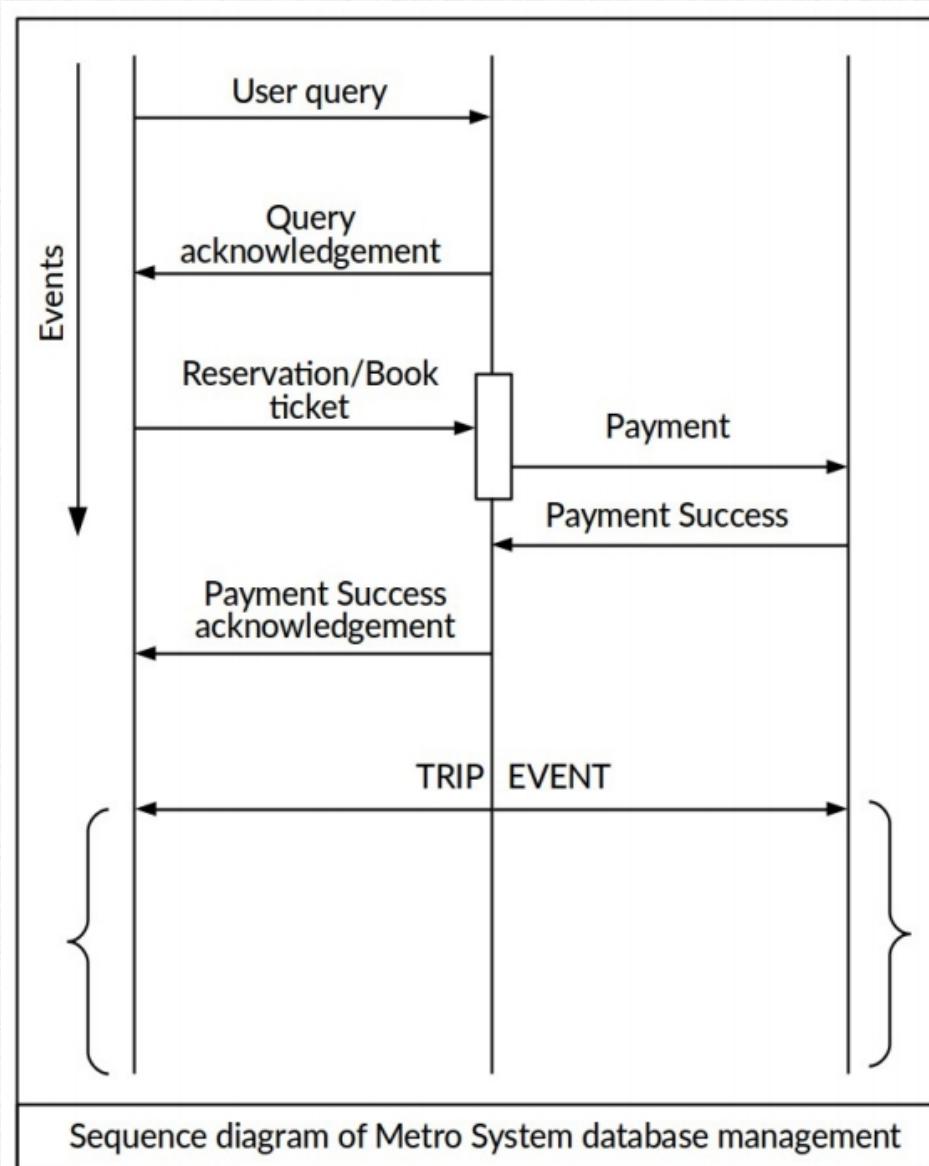
We've often used many Software engineering principles in the project to make it well organized, formal, and more productive. Use of Open-Source tools helps us to contribute Free software foundation and Open-source community. We've modelled the project structure into various diagrams such as Entity-Relationship diagrams, Use-Case diagrams, Sequence diagrams, Data flow diagrams.

-----CHAPTER 1 END

ER-Diagram



Sequence-Diagram



Tables

	USERNAME	PASSWORD	EMAIL_ID	ADDRESS
1	mark	plier123	markplier@gmail.com	Jahangirpuri, India
2	david	salmon123	david@gmail.com	Adarsh Nagar, India
3	amy	santiago123	amy@gmai.l.com	Azad Pur, India
4	terry	crews1234	terry@gmail.com	Model Town, India
5	raymond	holt1234	raymond@gmail.com	GTB Nagar, India
6	jack	peralta123	jack@gmail.com	Civil Lines, India
7	srikant	familyman123	srikant@gmail.com	Delhi Main, India

	PASSENGER_ID	FIRST_NAME	LAST_NAME	GENDER	PHONE_NO	TICKET_NO	AGE	CLASS
1	101	Harry	Howard	M	9898989310	10001	25	economy
2	102	Louis	Torres	M	7898989310	10002	34	executive
3	103	Peter	Thompson	M	3398989310	10003	42	general
4	104	Sean	Ramirez	F	4898989310	10004	26	executive
5	105	Ernest	Lee	M	5898989310	10005	18	economy
6	106	Todd	Jones	M	7898989510	10006	46	business
7	107	Harold	Bryant	M	5898989310	10007	34	economy
8	108	Fred	Riveria	M	2898989310	10008	19	economy
9	109	Kathy	Edwards	F	6898989310	10009	20	executive
10	110	Allen	Walker	M	7898989310	10010	29	general
11	111	Ashley	Lee	F	1898939313	10011	54	business

	USERNAME	PHONE_NO
1	amy	7071475390
2	david	9876543210
3	jack	9650367698
4	mark	8899887766
5	raymond	7411452250
6	srikant	9968254144
7	terry	8009224040

	METRO_NO	METRO_NAME	SEAT_GENERAL	SEAT_ECONOMY	SEAT_EXECUTIVE	SEAT_BUSINESS	WIFI	RUN_ON_SUNDAY
1	RI01D	Red Line	479	47	96	192	N	Y
2	YO02N	Yellow Line	480	43	96	192	N	Y
3	BI03D	Blue Line	432	48	80	144	N	N
4	GO04D	Green Line	432	48	80	144	N	N

	TICKET_NO	COACH	METRO_NO	DATE_OF_JOURNEY	USERNAME
1	10001	EC-1	RI01D	05-01-21	mark
2	10002	EX-1	RI01D	05-01-21	amy
3	10003	GE-1	G004D	05-01-21	david
4	10004	EX-1	RI01D	05-01-21	srikant
5	10005	EC-1	RI01D	05-01-21	jack
6	10006	B-1	Y002N	06-01-21	raymond
7	10007	EC-1	Y002N	06-01-21	terry
8	10008	EC-1	G004D	06-01-21	david
9	10009	EX-1	G004D	07-01-21	mark
10	10010	G-1	BI03D	07-01-21	mark
11	10011	B-1	BI03D	07-01-21	mark

	STATION_CODE	STATION_NAME
1	JAH	Jahangirpuri
2	ADN	Adarsh Nagar
3	AZP	Azad Pur
4	MOT	Model Town
5	GTB	GTB Nagar
6	CIL	Civil Lines
7	DEM	Delhi Main

Entity and Attributes

File Edit Selection View Go Run Terminal Help project.sql - Visual Studio Code

```
C:\> Users > NITIN > Downloads > project.sql
41 Gender char(1) NOT NULL,
42 Phone_No char(10) DEFAULT NULL,
43 Ticket_No int NOT NULL,
44 Age int NOT NULL,
45 /* This helps us to maintain an organizational data
46 Class varchar(20) NOT NULL, structure so it's difficult to maintain a record of data
47 PRIMARY KEY (Passenger_Id,Ticket_No), we are creating a
48 ); database for the passengers and their tasks and of course
49 
```

Defining a database involves specifying the data types, structures and constraints of the data to be stored in the database.

```
50 INSERT ALL system or Excel worksheets for information storage
51 INTO Passenger VALUES('101', 'Harry', 'Howard', 'M', '9898989310', '10001,25', 'economy')
52 INTO Passenger VALUES('102', 'Louis', 'Torres', 'M', '7898989310', '10002,34', 'executive') information is also stored by the DBMS in
53 INTO Passenger VALUES('103', 'Peter', 'Thompson', 'M', '3198989310', '10003,42', 'general') memory, it is called Metadata. Constructing
54 INTO Passenger VALUES('104', 'Sean', 'Ramirez', 'F', '4898989310', '10004,26', 'executive') a database includes functions such as
55 INTO Passenger VALUES('105', 'Ernest', 'Lee', 'M', '5898989310', '10005,18', 'economy') data, update and data to reflect
56 INTO Passenger VALUES('106', 'Todd', 'Jones', 'M', '7898989510', '10006,46', 'business') changes in the real world. Hence a
57 INTO Passenger VALUES('107', 'Harold', 'Bryant', 'M', '5898989310', '10007,34', 'economy') DBMS provides tools and programs to access the database
58 INTO Passenger VALUES('108', 'Fred', 'Riveria', 'M', '2898989310', '10008,19', 'economy') and retrieve data through queries or requests for
59 INTO Passenger VALUES('109', 'Kathy', 'Edwards', 'F', '6898989310', '10009,20', 'executive') information. In addition, a transaction
60 INTO Passenger VALUES('110', 'Allen', 'Walker', 'M', '7898989310', '10010,29', 'general') DBMS include protecting the database
61 INTO Passenger VALUES('111', 'Ashley', 'Lee', 'F', '1898939313', '10011,54', 'business') against unauthorized or malicious access.
62 
```

A typical large database may have a life cycle of many years, so the DBMS must be able to evolve the database system by allowing the system to evolve as requirements change over time.

```
63 CREATE TABLE Station (
64     Station_Code char(5) NOT NULL, Prototype, Software Engineering
65     Station_Name varchar(25) NOT NULL,
66     PRIMARY KEY (Station_Code)
67 );
68 
```

It is not absolutely necessary to use general-purpose DBMS software to manage a database. Instead, we could write our own set of programs to create and maintain the database, in effect creating our own purpose DBMS software. In either case, whether we use a general-purpose DBMS or not we usually have deployed a considerable amount of complex software.

```
69 
```

70

```
71 INSERT ALL
72     INTO Station VALUES('JAH', 'Jahangirpuri')
73     INTO Station VALUES('ADM', 'Adarsh Nagar')
74     INTO Station VALUES('AZP', 'Azad Pur')
75     INTO Station VALUES('MOT', 'Model Town')
76     INTO Station VALUES('GTB', 'GTB Nagar')
77     INTO Station VALUES('CIL', 'Civil Lines')
78     INTO Station VALUES('DEM', 'Delhi Main')
79 
```

80

```
81 SELECT * FROM dual; Documentation and making ER diagrams. User
82 functions and object oriented methodology constitute a major part of
83 the methodology. Project is divided into individual tasks. Tasks are then taken as
84 several days work to gather handling of user queries and data entries.
85 CREATe TABLE Metro (
86     Metro_No varchar(20) NOT NULL,
87 
```

The methodology adopted in the project is simple and is easy to implement. Functions and Object oriented methodology constitute a major part of methodology. Project is divided into individual tasks. Tasks are then taken as several days work to gather handling of user queries and data entries. Objectives are set first and then we need to fulfill all objectives mentioned.

Using a DBMS to manage data has many advantages:

- Data Independence:** application program should not, ideally, be expected to details of data representation and storage, the DBMS provides an abstract view of the data that hides such details.
- Efficient Data Access:** A DBMS utilizes a variety of sophisticated techniques to store and retrieve data efficiently. This feature is especially important if the data is to be stored on an external device.
- Data Integrity and security:** if data is always accessed through DBMS, the DBMS can enforce integrity constraints. For example, before inserting salary information into the database, the DBMS can check that the departmental budget is not exceeded. Also, it can enforce access controls that govern what data is visible to different classes of users.
- Data Administration:** when several users share data, centralizing the administration of data can offer significant improvements. Experienced DBAs can administer the data more effectively, and different users from different groups of users can use it. It can be responsible for organizing the data representation to minimize redundancy and for fine-tuning the storage of the data to make retrieval efficient.
- Concurrent Access and Cache Recovery:** A DBMS schedules concurrent access to the data in such a manner that users can access the data as though they were the only ones using it at a time. Further, the DBMS protects users from the effects of system failures.
- Reduced Application Development Time:** clearly, the DBMS supports important functions that are common to many applications accessing data in the DBMS. Thus, in comparison with the high-level interface to data, facilities such as concurrent development, DBMS applications are much likely to be faster than a similar stand-alone application because many important tasks are handled by the DBMS.

CHAPTER 2

ER-Diagram

The screenshot shows the Microsoft Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Title Bar:** project.sql - Visual Studio Code
- Code Editor:** The main editor window displays the `project.sql` file, which contains SQL code for creating tables and inserting data into the Metro System database.
- ER-Diagram Tab:** A tab labeled "ER-Diagram" is visible on the right side of the interface.
- Status Bar:** Shows the current file path as C:\Users\NITIN\Downloads\project.sql, the line number (1), and the status bar message "CodeTogether 8.0.1".

File Edit Selection View Go Run Terminal Help

project.sql - Visual Studio Code

C:\Users\Nitin\Downloads> project.sql

```
79  SELECT * FROM dual;
```

```
80
```

```
81 CREATE TABLE Metro (
```

```
82     Metro_No varchar(6) NOT NULL,
```

```
83     Metro_Name varchar(25) NOT NULL,
```

```
84     Seat_General int NOT NULL,
```

```
85     Seat_Economy int NOT NULL,
```

```
86     Seat_Executive int NOT NULL,
```

```
87     Seat_Business int NOT NULL,
```

```
88     Wifi char(1) NOT NULL,
```

```
89     Run_On_Sunday char(1) NOT NULL,
```

```
90     Run_On_Monday char(1) NOT NULL,
```

```
91     Run_On_Tuesday char(1) NOT NULL,
```

```
92     Run_On_Wednesday char(1) NOT NULL,
```

```
93     Run_On_Thursday char(1) NOT NULL,
```

```
94     Run_On_Friday char(1) NOT NULL,
```

```
95     Run_On_Saturday char(1) NOT NULL,
```

```
96     PRIMARY KEY (Metro_No)
```

```
);
```

```
97     -- An application program can access the database by sending queries or requests for data to DBMS. A query typically contains some data to be retrieved, a transaction may contain statements to be read and another statement to be written into the database.
```

```
98     -- Other important functions provided by DBMS are as follows:
```

```
99     -- 1. Data protection: In order to ensure the integrity of the data, protection includes system protection against hardware or software malfunction and security protection against unauthorized or malicious access.
```

```
100    -- INTO Metro VALUES ('R101D','Red Line',479,49,96,192,'N','Y','Y','Y','Y','Y','Y','Y');
```

```
101    -- of many years, so the DBMS must
```

```
102    -- INTO Metro VALUES ('Y002N','Yellow Line',480,43,96,192,'N','Y','Y','Y','Y','Y','Y','Y');
```

```
103    -- INTO Metro VALUES ('B103D','Blue Line',432,48,80,144,'N','N','Y','Y','Y','Y','Y','Y');
```

```
104    -- INTO Metro VALUES ('G004D','Green Line',432,48,80,144,'N','N','Y','Y','Y','Y','Y','Y');
```

```
105    -- SELECT * FROM dual;
```

```
106
```

```
CREATE TABLE Ticket (
```

```
107     Ticket_No INT NOT NULL,
```

```
108     Coach varchar(10) NOT NULL,
```

```
109     Metro_No VARCHAR(6) NOT NULL,
```

```
110     Date_of_Journey date NOT NULL,
```

```
111     Username varchar(15) NOT NULL,
```

```
112     PRIMARY KEY (Ticket_No),
```

```
113     FOREIGN KEY (Username) REFERENCES Account (Username),
```

```
114     FOREIGN KEY (Metro_No) REFERENCES Metro (Metro_No)
```

```
);
```

```
115
```

```
116     -- Methodology adopted in this project is simple and is easy to implement.
```

```
117     -- Entity and Object oriented methodology constitute a major portion of
```

```
118     -- Project. Project is divided into individual tasks. Tasks are then taken as
```

```
119     -- sub-tasks and solved individually and have to integrate them at last.
```

```
117  INSERT ALL
```

```
118      INTO Ticket VALUES('10001','EC-1','R101D','5-JAN-2021','mark')
```

```
119      INTO Ticket VALUES('10002','EX-1','R101D','5-JAN-2021','amy')
```

A database management system (DBMS) is a collection of programs that enables users to create and maintain a database. The DBMS is a general-purpose software system that facilitates the processes of defining, constructing, maintaining and sharing databases among various users and applications.

Defining a database involves specifying the data types, structures and constraints of the data to be stored in the database.

The database definition or description information is also stored by the DBMS in the form of a database catalog or dictionary, it is called Metadatas. Constructing the database is the process of storing the data on some storage medium that is controlled by the DBMS. Manipulating a database includes functions such as querying the database to retrieve specific data, updating the database to reflect changes in the real world and generating reports from the data. Sharing a database allows multiple users and programs to access the database simultaneously.

An application program accesses the database by sending queries or requests for data to DBMS. A query typically contains some data to be retrieved, a transaction may contain statements to be read and another statement to be written into the database.

Other important functions provided by DBMS are as follows:

- Data independence:** application program should not, ideally, be expected to details of data representation and storage, the DBMS provides an abstract view of the data that hides such details.
- Efficient Data Access:** A DBMS utilizes a variety of sophisticated techniques to store, and retrieve data efficiently. This feature is especially important if the data is to be stored on an external device.
- Data Integrity and security:** if data is always accessed through DBMS, the DBMS can enforce integrity constraints. For example, before inserting salary information for an employee, the DBMS can check that the department budget is not exceeded. Also, it can enforce access controls that govern what data is visible to different classes of users.
- Data Administration:** when several users share data, centralizing the administration of data can offer significant improvements. Experienced professionals who understand the nature of the data being managed, and how different groups of users use it, can be responsible for organizing the data and defining its logical boundaries and for fine-tuning the storage of the data to make retrieval efficient.
- Concurrent Access and Crash Recovery:** A DBMS schedules concurrent access to the data in such a manner that users can think of the data as being accessed by only one user at a time. Further, the DBMS protects users from the effects of system failures.
- Reduced Application Development Time:** clearly, the DBMS supports important functions that are common to many applications accessing data in the DBMS. This, in conjunction with the high-level interface to data, facilitates quick application development. DBMS applications are also likely to be more robust than a series of stand-alone applications because many important tasks are handled by the DBMS.

Methodology

The methodology adopted in the project is simple and is easy to implement. Entity and Object oriented methodology constitute a major portion of the project. Project is divided into individual tasks. Tasks are then taken as sub-tasks and solved individually and have to integrate them at last. They are not first and then we need to fulfill all objectives associated.

CHAPTER 2

ER-Diagram

File Edit Selection View Go Run Terminal Help project.sql - Visual Studio Code

C:\> Users > NITIN > Downloads > project.sql

```
122     INTO Ticket VALUES('10005', 'EC-1', 'RIO1D', '5-JAN-2021', 'jack')
123     INTO Ticket VALUES('10006', 'B-1', 'Y002N', '6-JAN-2021', 'raymond')
124     INTO Ticket VALUES('10007', 'EC-1', 'Y002N', '6-JAN-2021', 'terry')
125     INTO Ticket VALUES('10008', 'EC-1', 'GO04D', '6-JAN-2021', 'david') management system (DBMS) is a collection of programs that enables
126     INTO Ticket VALUES('10009', 'EX-1', 'GO04D', '7-JAN-2021', 'mark') creation and maintenance of a database. The DBMS is a general-purpose
127     INTO Ticket VALUES('10010', 'G-1', 'BI03D', '7-JAN-2021', 'mark') system that facilitates the processes of defining, constructing,
128     INTO Ticket VALUES('10011', 'B-1', 'BI03D', '7-JAN-2021', 'mark') maintaining and sharing databases among various users and applications
129 SELECT * FROM dual;
```

A database involves specifying the data types, structures and constraints of the data to be stored in the database.

The database definition or description information is also stored by the DBMS in the form of a database catalog or dictionary, it is called Metadatas. Constructing the database requires the DBMS to interact with an external vendor who is connected to the DBMS. Managing a database involves functions such as querying the database to retrieve specific data, updating the database to reflect changes in the mini world and generating reports from the data. Sharing a database allows multiple users and programs to access the database simultaneously.

An application program accesses the database by issuing queries or requests for data to the DBMS. A query typically causes some data to be retrieved, a transaction may cause some data to be read and some data to be written into the database.

Other important functions provided by DBMS include protecting the database and maintaining it over a long period of time, protection includes system protection against hardware or software malfunction and security protection against unauthorized access.

Data Administration: when several users share data, maintaining the data integrity becomes a major concern. Every user must be aware of the professionals who understand the nature of the data being managed, and how different groups of users use it, it can be responsible for organizing the data representation to minimize redundancy and for fine-tuning the storage of the data to make retrieval efficient.

Concurrent Access and Cache Recovery: A DBMS schedules concurrent accesses to the data in such a manner that users can think of the data as being accessed by only one user at a time. Further, the DBMS protects users from the effects of system failures.

Reduced Application Development Time: clearly, the DBMS supports relational framework that are common to many applications accessing data in DBMS. In this conjunction with the high-level interface to data, facilitates quick application development. DBMS applications are also likely to be more robust than a similar stand-alone application because many important tasks are handled by the DBMS.

ER-Diagram

CHAPTER 2

FD and Normalization

METRO

metro_no ->
 (metro_name,seat_general,seat,economy,seat_executive,seat_business,wifi,run_on_sunday,run_on_monday,
 run_on_tuesday run_on_wednesday,run_on_thursday,run_on_friday, run_on_saturday)

STOPPAGE

(`Metro_No` , `Station_Code`) -> (`Arrival_Time` , `Departure_Time`)

TICKET

'Ticket_No' -> ('Metro_No` `Date_of_Journey` `Username`)

STATION

'Station_Code' -> 'Station_Name'

PASSENGER

'Passenger_Id' -> ('First_Name` `Last_Name` `Gender` `Phone_No` `Ticket_No` `Age` `Class`)

ACCOUNT

'Username' ->(`Password` `Email_Id` `Address`)

FIRST NORMAL FORM:

As per the rule of first normal form, an attribute (column) of a table cannot hold multiple values. It should hold only atomic values. The above schema is in 1NF since all the attributes are atomic and not multi valued. Since a passenger could have multiple phone numbers, it would violate the 1NF rules. Hence we have created a separate table called contact to handle this.

SECOND NORMAL FORM:

A table is said to be in 2NF if both the following conditions hold:

-Table is in 1NF (First normal form) -No non-prime attribute is dependent on the proper subset of any candidate key of table. If in Passenger table we consider ticket_no and first_name as the candidate key, then date_of_birth would depend only on the name and it would violate the 2NF.

THIRD NORMAL FORM:

A table design is said to be in 3NF if both the following conditions hold:-Table must be in 2NF -Transitive functional dependency of non-prime attribute on any super key should be removed.

Our schema follows the above rules and hence is in 3NF.

Trigger

A trigger has been created which is invoked each time a ticket is canceled. The trigger helps in increasing the number of seats in a coach after cancellation.

```

trigger.sql x
C: > Users > NITIN > Downloads > trigger.sql
1  create or replace trigger cancellation
2    before delete on Ticket
3    for each row
4    declare
5      metrono metro.metro_no%type;
6      ticketno passenger.ticket_no%type;
7      p_class passenger.class%type;
8
9      BEGIN
10        metrono := :old.metro_no;
11        ticketno := :old.ticket_no;
12        select class into p_class
13        from passenger
14        where ticket_no = ticketno;
15        if p_class='general' then
16          UPDATE Metro
17          set Seat_General = Seat_General + 1
18          WHERE Metro_No = metrono;
19        elsif p_class='economy' then
20          UPDATE Metro
21          set Seat_economy = Seat_economy + 1
22          WHERE Metro_No = metrono;
23        elsif p_class='executive' then
24          UPDATE Metro
25          set Seat_Executive = Seat_Executive + 1
26          WHERE Metro_No = metrono;
27        elsif p_class='business' then
28          UPDATE Metro
29          set Seat_Business = Seat_Business + 1
30          WHERE Metro_No = metrono;
31      end if;
32    /
33
34  delete from ticket where ticket_no = 10001;

```

Problem Statement

The major task is to make a functional Database system for a Metro Management

Methodology

Using a DBMS to manage data has many advantages:

- **Data independence:** application program should not, ideally, be aware of details of data representation and storage, the DBMS provides an abstract view of the data that hides such details.
- **Efficient Data Access:** A DBMS utilizes a variety of sophisticated techniques to store and retrieve data efficiently. This feature is especially important for large databases.
- **Data integrity and security:** if data is always accessed through the DBMS, it can enforce integrity constraints. For example, before inserting information for an employee, the DBMS can check that the department number does not exceed. Also, it can enforce access controls that govern what data is available to different classes of users.
- **Data Administration:** when several users share data, centralized administration of data can offer significant improvements. DBMS professionals who understand the nature of the data being managed can define different groups of users use it. It can be responsible for organizing representation to minimize redundancy and for fine-tuning the storage to make retrieval efficient.
- **Concurrent Access and Crash Recovery:** A DBMS schedules access to the data in such a manner that there can think of the data as being accessed by only one user at a time. Further, the DBMS prevents data loss effects of system failures.
- **Reduced Application Development Time:** clearly, the DBMS implemented functions that are common to most applications according to DBMS. Thus, in comparison with the high-level interface to data, full-fledged application development DBMS applications are also likely to be more than a similar stand-alone application because many important tasks are handled by the DBMS.

ER-Diagram

SQL Queries

Query 1 : Find total number of business class seats available on any Metro that reaches Jahangirpuri before 7AM on Monday

```

1 /* Find total number of business class seats available on any Metro that reaches Delhi Main before 7AM on Monday*/
2 drop view A;
3 drop view B;
4 drop view C;
5 Regarding a metro system. Since it is very difficult to maintain a record of data
6 create view A (station_code,metro_no,arrival_time) AS
7     SELECT stoppage.station_code, metro_no, arrival_time
8     FROM station inner join stoppage ON station.station_code=stoppage.station_code
9     WHERE station.station_name='Jahangirpuri';
10
11 -- This view will be used to maintain a record of all the stops made by the metro
12 -- without cluttering with proper size interface and some random
13 select * from A;
14
15 create view B (station_code,metro_no,arrival_time) AS
16     SELECT station_code, metro_no, arrival_time
17     FROM A
18     WHERE EXTRACT(HOUR FROM arrival_time) < 7;
19
20 create view C (station_code, metro_no, arrival_time, business_class_seats, run_on_monday) AS
21     SELECT station_code, metro.metro_no, arrival_time, seat.business, run_on_monday
22     FROM metro inner join B on metro.metro_no = B.Metro_No
23     WHERE metro.run_on_monday='Y' AND metro.seat_business > 0;
24
25 select * from C;
26
27 SELECT SUM(business_class_seats)
28 FROM C;

```

Using a DBMS to manage data has many advantages.

- **Data Independence:** Application programs should not ideally be referred to directly for data representation and storage. The DBMS provides an abstract view of the data that hides such details.
- **Efficient Data Access:** A DBMS utilizes a variety of sophisticated techniques to store and retrieve data efficiently. This feature is especially important if the data is to be stored on an external device.
- **Data Integrity and Security:** If data is always accessed through DBMS, the DBMS can enforce integrity constraints. For example, before inserting salary information for an employee, the DBMS can check whether department budget is not exceeded. Also, it can enforce access controls that prevent certain data from visible to different classes of users.
- **Data Administration:** When several users share data, controlling the administration of data can offer significant improvements. Experienced professionals who understand the nature of the data being managed, and how different groups of users use it, it can be responsible for organizing the data representations to minimize redundancy and for fine-tuning the storage of the data to make retrieval efficient.
- **Concurrent Access and Crash Recovery:** A DBMS schedules concurrent access to the data in such a manner that users can think of the data as being accessed by only one user at a time. Further, the DBMS protects users from the effects of system failures.
- **Reduced Application Development Time:** clearly, the DBMS supports important functions that are common to many applications accessing data of the DBMS. Thus, in conjunction with the high-level interface to data, facilitates quick application development. DBMS applications are also likely to be more robust than a similar stand-alone application because many important truly are handled by the DBMS.

View A created.																			
>>Query Run In:Query Result 7																			
View B created.																			
>>Query Run In:Query Result 8																			
View C created.																			
>>Query Run In:Query Result 9																			
>>Query Run In:Query Result 10																			
query result - 7																			
<table border="1"> <thead> <tr> <th>STATION_CODE</th> <th>METRO_NO</th> <th>ARRIVAL_TIME</th> </tr> </thead> <tbody> <tr> <td>1 JAH</td> <td>RI01D</td> <td>05-01-21 12:05:00.000000000 AM</td> </tr> <tr> <td>2 JAH</td> <td>YO02N</td> <td>06-01-21 03:45:00.000000000 PM</td> </tr> <tr> <td>3 JAH</td> <td>BI03D</td> <td>07-01-21 01:20:00.000000000 AM</td> </tr> </tbody> </table>					STATION_CODE	METRO_NO	ARRIVAL_TIME	1 JAH	RI01D	05-01-21 12:05:00.000000000 AM	2 JAH	YO02N	06-01-21 03:45:00.000000000 PM	3 JAH	BI03D	07-01-21 01:20:00.000000000 AM			
STATION_CODE	METRO_NO	ARRIVAL_TIME																	
1 JAH	RI01D	05-01-21 12:05:00.000000000 AM																	
2 JAH	YO02N	06-01-21 03:45:00.000000000 PM																	
3 JAH	BI03D	07-01-21 01:20:00.000000000 AM																	
query result - 8																			
<table border="1"> <thead> <tr> <th>STATION_CODE</th> <th>METRO_NO</th> <th>ARRIVAL_TIME</th> </tr> </thead> <tbody> <tr> <td>1 JAH</td> <td>RI01D</td> <td>05-01-21 12:05:00.000000000 AM</td> </tr> <tr> <td>2 JAH</td> <td>BI03D</td> <td>07-01-21 01:20:00.000000000 AM</td> </tr> </tbody> </table>					STATION_CODE	METRO_NO	ARRIVAL_TIME	1 JAH	RI01D	05-01-21 12:05:00.000000000 AM	2 JAH	BI03D	07-01-21 01:20:00.000000000 AM						
STATION_CODE	METRO_NO	ARRIVAL_TIME																	
1 JAH	RI01D	05-01-21 12:05:00.000000000 AM																	
2 JAH	BI03D	07-01-21 01:20:00.000000000 AM																	
query result - 9																			
<table border="1"> <thead> <tr> <th>STATION_CODE</th> <th>METRO_NO</th> <th>ARRIVAL_TIME</th> <th>BUSINESS_CLASS_SEATS</th> <th>RUN_ON_MONDAY</th> </tr> </thead> <tbody> <tr> <td>1 JAH</td> <td>RI01D</td> <td>05-01-21 12:05:00.000000000 AM</td> <td>192</td> <td>Y</td> </tr> <tr> <td>2 JAH</td> <td>BI03D</td> <td>07-01-21 01:20:00.000000000 AM</td> <td>144</td> <td>Y</td> </tr> </tbody> </table>					STATION_CODE	METRO_NO	ARRIVAL_TIME	BUSINESS_CLASS_SEATS	RUN_ON_MONDAY	1 JAH	RI01D	05-01-21 12:05:00.000000000 AM	192	Y	2 JAH	BI03D	07-01-21 01:20:00.000000000 AM	144	Y
STATION_CODE	METRO_NO	ARRIVAL_TIME	BUSINESS_CLASS_SEATS	RUN_ON_MONDAY															
1 JAH	RI01D	05-01-21 12:05:00.000000000 AM	192	Y															
2 JAH	BI03D	07-01-21 01:20:00.000000000 AM	144	Y															
query result - 10																			
<table border="1"> <thead> <tr> <th>SUM(BUSINESS_CLASS_SEATS)</th> </tr> </thead> <tbody> <tr> <td>1 336</td> </tr> </tbody> </table>					SUM(BUSINESS_CLASS_SEATS)	1 336													
SUM(BUSINESS_CLASS_SEATS)																			
1 336																			

Query 2 : Find the time at which last Metro leaves Adarsh Nagar station

```
/* Find the time at which last Metro leaves Adarsh Nagar station */

create view F (departure_time)as
    SELECT departure_time
    FROM stoppage
    WHERE station_code IN ( SELECT station_code
                                FROM station
                                WHERE station_name='Adarsh Nagar') ;
select * from F ;
select MAX(Departure_time) from F;
```

The screenshot shows a database interface with two main sections. The top section displays the creation of a view named F:

```
View F created.

>>Query Run In:Query Result
>>Query Run In:Query Result 1
```

The bottom section shows the results of the query:

query result

	DEPARTURE_TIME
1	05-01-21 01:38:00.000000000 AM
2	06-01-21 01:05:00.000000000 AM
3	07-01-21 04:20:00.000000000 AM
4	06-01-21 05:00:00.000000000 AM

query result 1

	MAX(DEPARTURE_TIME)
1	07-01-21 04:20:00.000000000 AM

Query 3 : Find the phone number of the user whose email id is jack@gmail.com

```
/* Find the phone number of the user whose email id is jack@gmail.com */

Select phone_no
from contact
where username IN (Select username
                    from account
                    where email_id='jack@gmail.com');
```

The screenshot shows the results of the query:

query_-3.sql 1 KB

PHONE_NO

1	9650367698
---	------------

TESTING

Procedure

The procedure of testing of SQL codes used is quite simple, standard, effective. The steps used in complete testing procedure is shown below-

1. Checking for SQL Syntax by directly running the queries
2. Checking for broader logic of code: Is code working according to logic or not?
3. White box Checking
- 4.1 Structural testing which checks outer structure of logic
5. Black box checking
- 6.1 Functional testing which checks if code working according to functionality or not.
7. Coding standards which is not a testing step but we check for some standard code including good formatting, alignments, procedure calls, etc.

Outcome

Testing provides us to minimize sort of error prone functionality as per our Implementation. Here, following simple procedure for testing we get desired hassle free functionality. Project is currently working fine and since testing procedure is available, we can further apply that testing procedure in case of malfunction or bug findings.

Conclusion and Future Scope

Conclusions:

Usage of database today is of utmost importance for an enterprise or an institution because it eliminates all the major drawbacks of file system via Programming language or offline file saving.

It helps in improving the whole procedure of data/information storage and retrieval. There are also some risks such as injection techniques, Reliability issues, data inconsistency risks, etc. Hence it requires to construct a proper Risk Mitigation, Monitoring, and Management plan (RMMM Plan) for specific usage of databases practically.

We keep our project small and simple which lead to some major improvements in costs, efforts, and time usage in complete life cycle of project development. We've tried to include all major functionalist which must be there in a metro/railway system database but still it is far behind than a practical working software. It requires large number of Attribute their manipulate and maintenance.

Future Scope and improvements:

In near future the usage of database will probably be decreased due to huge sources of data and new technologies such as Big data analytic which will further improve the drawbacks of database management system.

The project output can be used as raw input to a bigger project which may lead to Practical implementation of various concepts and debugging. This can be enhanced much on following points-

Implementation:

- It can be implemented using Oracle database 12C, DB2, PostgreSQL, MongoDB. LibreOffice is currently using HSQL Embedded as embedded Database, instead of which other databases can be used as per requirements.
- The relational schema for Metro System can also be implemented on Object Oriented database which has its own pros and cons.

References

Database System concepts. Abraham Silberschatz, Henry F. Korth, S. Sudarshan. 6th International editon. Mcraw-Hill, New York.
ISBN 978-0-07-352332-3. MHID 0-07-352332-1.

Database Systems The Complete Book. Hector arcia-Molina, Jefrey D. Ullman, Jennifer Widom. 2ne editon. Prentce Hall, New Jersey.
ISBN 0-13-606701-8, 978-0-13-606701-6.

Database Management Systems. Raghu Ramakrishnan, Johannes Gehrke. 2 nd editon. Mcraw Hill higher education. ISBN 0-07-246535-2.

Database Management System (DMS) tutorial. Tutorials Point.
SQL tutorial. Tutorials Point. <https://www.tutorialspoint.com/>.