

DBMS Project

SUBMITTED BY:

Vishal Mathur (61)

Rahul Jain (69)

Rishabh Bajaj (72)

Muskaan Dua (92)

Akshat Bisht (93)

Class : CSE – B , 4th Sem.

SUBMITTED TO:

Dr. Achal Kaushik

Associate Professor,

Bhagwan Parshuram Institute of

Technology (BPIT)

A Java based project, linked with oracle SQL. It helps an individual to know about places near a particular metro station and also helps in knowing the reverse of it, i.e., knowing nearest metro to a location.

A mini project which helps the user to:

- FETCH the nearby places to a metro station along with the distance between them.
- FETCH the nearest metro station to a prime location along with the distance between them.
- INSERT a new location as a prime location to a metro station.
- UPDATE existing data in our prime location relation.
- DELETE record(s) from our prime location relation.

The user is required to login to the application. Only the authorized users are allowed to use the application. If not an authorized user, one must sign in to access the application. All the necessary user information is recorded and kept secured. After authorization, user interacts with the application. Now the user can either enter the prime location to find the near by metro stations from it; or he can search for different attractions around a particular metro station. The user is provided with the nearest metro station to a prime location. His source and destination places are tracked efficiently to provide him the best ride. He can also view the metro map for interchanging between two or more line.

TABLE DESCRIPTION

The project uses the following Database schema:

Tables used:

- USERS: records contains user related information. It stores the information of authorized users. Users whose information are stored in the USER table are only allowed to login and use the application.

Table 1 Table USER

S. No.	Name	Type	Constraint
1.	FName	Varchar2(15)	Not null
2.	LName	Varchar2(10)	Not null
3.	Username	Varchar2(20)	Primary key
4.	Password	Varchar2(30)	Not null
5.	Mail	Varchar2(30)	
6.	Gender	Varchar2(6)	Not null

- LOCATION: contains the basic details of metro station along with its prime location and distance between them. It is used to fetch all the related information of a metro including its colour, etc.

Table 2 Table LOCATION

S. No.	Name	Type	Constraint
1.	PrimeLocation	Varchar2(30)	
2.	MetroStation	Varchar2(30)	Foreign Key, references table station - column MetroStation.
3.	Distance(km)	Decimal(4,2)	

- STATION: stores the station names of all metro tracks and their relative metro information which can be easily used to find the color of the metro line available in that station.

Table 3 Table STATION

S. No.	Name	Type	Constraint
1.	MetroStation	Varchar2(20)	Primary Key
2.	MetroID	Number(4)	Foreign Key, references table metro - column MetroID.

- METRO: stores information about different metro ID .

Table 4 Table METRO

S. No.	Name	Type	Constraint
1.	MetroID	Number(4)	Primary Key
2.	MetroColor	Varchar2(10)	Not null

Normalization is a process of organizing the data in database to avoid data redundancy, insertion anomaly, update anomaly & deletion anomaly.

- As per the rule of first normal form, an attribute (column) of a table cannot hold multiple values. It should hold only atomic values. All our tables hold only atomic values.
- A table is said to be in 2NF if both the following conditions hold: a) Table is in 1NF and b) No non-prime attributes is dependent on the proper subset of any candidate key of table. The metro station have different color metro lines. If a particular station appears more than once in the table location, then we have normalized the table and stored the value of color lines available only once for a given metro station.

PrimeLocation	MetroStation	Distance	MetroID
Adventure Island	Rohini West	1.5	1024
Unity One	Rohini West	0.2	1024
Car Market	Karol Bagh	2.2	1240
Swarn Jayanti P.	Rohini West	1	1024
Nai Sarak	Chandini Chowk	3	1204
Sis Gany Sahib	Chandini Chowk	3.5	1204
Red Fort	New Delhi	7.8	1204
Majnu Ka Tilla	Civil Lines	4.8	1204
Majnu Ka Tilla	Vidhan Sabha	4	1204
Majnu Ka Tilla	Model Town	4.3	1204
:	:		

There are 3 tuples containing Rohini-West & 1024 as metro-station & MetroID, there are 2 tuples for Chandini Chowk. Change in MetroID for Rohini West would require in all the three tuples. If by mistake any one of the tuple is left out and the data remains unchanged, there will be a redundancy. So, for normalizing the table for achieving 2NF which in turn reduces redundancy in data is given below:

PrimeLocation	MetroStation	Distance	MetroStation	MetroID
Same as above	Same as above	Same as above	Rohini West	1024
			Karol Bagh	1240
			Chandini Chowk	1204
			New Delhi	1204
			:	:
			:	:
			:	:

- A table design is said to be in 3NF if both the following conditions hold: a) Table is in 2NF and b) Transitive functional dependency of non-prime attribute on any super key should be removed. The concept of 3NF is applied as shown below :

MetroStation	MetroID	MetroColor
Rohini West	1024	Red
Karol Bagh	1240	Blue
Chandini Ch.	1204	Yellow
New Delhi	1204	Yellow
Netaji Subhash..	1024	Red
Rajiv Chowk	1204	Yellow
⋮	⋮	⋮

If we see in this case $\text{MetroID} \rightarrow \text{MetroColor}$, there is a complete dependency of MetroColor on MetroID, which on updation of some tuples may result in redundancy of data. So for normalizing the table & for achieving 3NF we can do the following \rightarrow

MetroStation	MetroID
Same as above	Same as above

MetroID	Metro Color
1024	Red
1240	Blue
1204	Yellow
⋮	⋮



ER DIAGRAM

