Experiment No.03

PART A

(PART A: TO BE REFFERED BY STUDENTS)

A.1 Aim:

1. To study and implement DDL command with along with below constraints:
    - Primary key
    - Not Null
    - Unique key
    - Foreign key
    - Check constraint
2. To study and implement DML commands like Insert, Update, Delete and Select command.

A.2 Prerequisite:

Basic concepts of database and introduction to SQL, DDL and DMS commands.

A.3 Outcome:

After successful completion of this experiment students will be able to

1. Apply knowledge of relational algebra and structural query language to retrieve and manage data in relational databases.

A.4 Theory:

SQL (pronounced "ess-que-el") stands for Structured Query Language. SQL is used to communicate with a database. According to ANSI (American National Standards Institute), it is the standard language for relational database management systems. SQL statements are used to perform tasks such as update data on a database, or retrieve data from a database. Some common relational database management systems that use SQL are: Oracle, Sybase, Microsoft SQL Server, Access, Ingres, etc. Although most database systems use SQL, most of them also have their own additional proprietary extensions that are usually only used on their system. However, the standard SQL commands such as "Select", "Insert", "Update", "Delete", "Create", and "Drop" can be used to accomplish almost everything that one needs to do with a database.

A relational database system contains one or more objects called tables. The data or information for the database are stored in these tables. Tables are uniquely identified by their names and are comprised of columns and rows. Columns contain the column name, data type, and any other attributes for the column. Rows contain the records or data for the columns. Here is a sample table called "weather". City, state, high, and low are the columns. The rows contain the data for this table:

Weather			
city	**state**	**high**	**low**
Phoenix	Arizona	105	90
Tucson	Arizona	101	92

Flagstaff	Arizona	88	69
San Diego	California	77	60
Albuquerque	New Mexico	80	72

The **create table** statement is used to create a new table. Here is the format of a simple **create table**statement:

create table "tablename"("column1" "data type", "column2" "data type", "column3" "data type");

To create a new table, enter the keywords **create table** followed by the table name, followed by an open parenthesis, followed by the first column name, followed by the data type for that column, followed by any optional constraints, and followed by a closing parenthesis. It is important to make sure you use an open parenthesis before the beginning table, and a closing parenthesis after the end of the last column definition. Make sure you seperate each column definition with a comma. All SQL statements should end with a ";".

The table and column names must start with a letter and can be followed by letters, numbers, or underscores - not to exceed a total of 30 characters in length. Do not use any SQL reserved keywords as names for tables or column names (such as "select", "create", "insert", etc).

Data types specify what the type of data can be for that particular column. If a column called "Last_Name", is to be used to hold names, then that particular column should have a "varchar" (variable-length character) data type.

Here are the most common Data types:

char(size)	Fixed-length character string. Size is specified in parenthesis. Max 255 bytes.
varchar(size)	Variable-length character string. Max size is specified in parenthesis.
number(size)	Number value with a max number of column digits specified in parenthesis.
date	Date value
number(size,d)	Number value with a maximum number of digits of "size" total, with a maximum number of "d" digits to the right of the decimal.

Constraints:

What are constraints? When tables are created, it is common for one or more columns to have **constraints** associated with them. A constraint is basically a rule associated with a column that the data entered into that column must follow. For example, a "unique" constraint specifies that no two records can have the same value in a particular column. They must all be unique. The other two most popular constraints are "not null" which specifies that a column can't be left blank, and "primary key". A "primary key" constraint defines a unique identification of each record (or row) in a table.

Format of create table if you were to use optional constraints:

```
create table "tablename" ("column1" "data type"        [constraint],
                          "column2" "data type"        [constraint],
                          "column3" "data type"        [constraint]);
```

1. Primary key constraint:

The PRIMARY KEY constraint uniquely identifies each record in a database table. Primary keys must contain UNIQUE values. A primary key column cannot contain NULL values. Most tables should have a primary key, and each table can have only ONE primary key.

create table Accounts(acc_no number(5) primary key, balance number(8,2));

or

create table Accounts(acc_no number(5), balance number(8,2), primary key(acc_no));

or adding primary key constraint using alter command – can be used only if table has been already created and during creation time constraint was not given.

alter table Accounts add primary key(acc_no);

2. Unique constraint:

The UNIQUE constraint uniquely identifies each record in a database table. The UNIQUE and PRIMARY KEY constraints both provide a guarantee for uniqueness for a column or set of columns. A PRIMARY KEY constraint automatically has a UNIQUE constraint defined on it. Note that you can have many UNIQUE constraints per table, but only one PRIMARY KEY constraint per table.

create table person(p_id number(4) primary key, pname varchar(40),mobile_no number(10) unique, address varchar(100));

Or

create table person(p_id number(4) primary key, pname varchar(40), mobile_no number(10), address varchar(100), unique(mobile_no));

or adding unique key using alter command – can be used only if table is already created and at the time of creation of table constraint was not applied.

alter table person add unique(mobile_no);

3. NOT NULL constraint:

The NOT NULL constraint enforces a column to NOT accept NULL values. The NOT NULL constraint enforces a field to always contain a value. This means that you cannot insert a new record, or update a record without adding a value to this field.

create table person(p_id primary key, pname varchar(4) not null, mobile_no number(10), address varchar(100));

4. Foreign key:

A FOREIGN KEY in one table points to a PRIMARY KEY in another table. Let's illustrate the foreign key with an example. Look at the following two tables:

The "Persons" table:

P_Id	LastName	FirstName	Address	City
1	Hansen	Ola	Timoteivn 10	Sandnes
2	Svendson	Tove	Borgvn 23	Sandnes
3	Pettersen	Kari	Storgt 20	Stavanger

The "Orders" table:

O_Id	OrderNo	P_Id
1	77895	3
2	44678	3
3	22456	2
4	24562	1

Note that the "P_Id" column in the "Orders" table points to the "P_Id" column in the "Persons" table. The "P_Id" column in the "Persons" table is the PRIMARY KEY in the "Persons" table. The "P_Id" column in the "Orders" table is a FOREIGN KEY in the "Orders" table. The FOREIGN KEY constraint is used to prevent actions that would destroy links between tables. The FOREIGN KEY constraint also prevents invalid data from being inserted into the foreign key column, because it has to be one of the values contained in the table it points to.

```
create table orders(o_id number(3) primary key, orderno number not null, p_id foreign key references persons(p_id));
```

Adding foreign key with alter table command: Applicable only if table is already created and at the time of creation foreign key was not given.

```
alter table orders add foreign key(p_id) references persons(p_id);
```

Drop Table Command:

It is used to completely delete table along with all its rows from the memory.

Syntax:

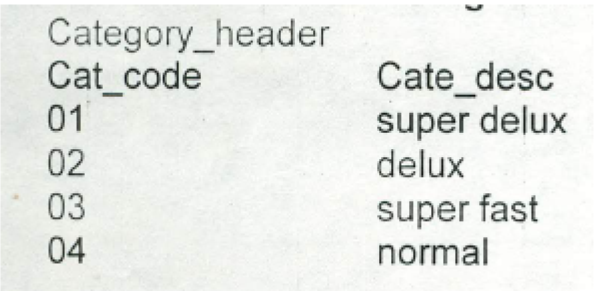
drop table table name;

drop table orders;

A.5 Task:

1. **Insert data into the tables as shown below**

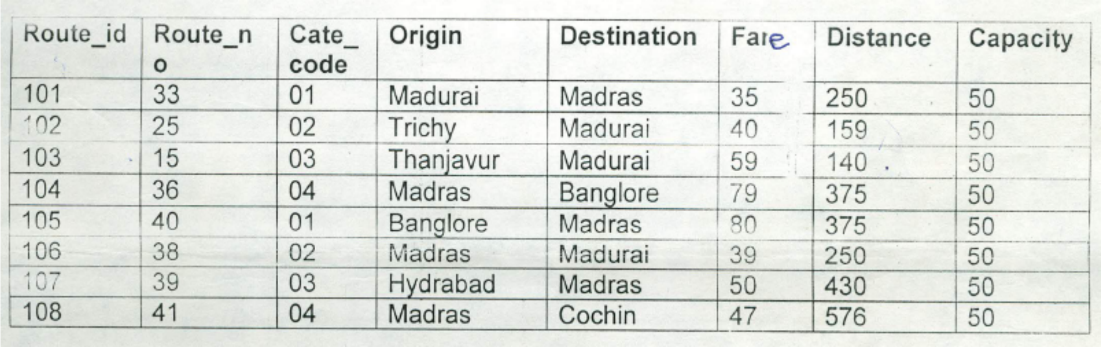
category_header



Category_header

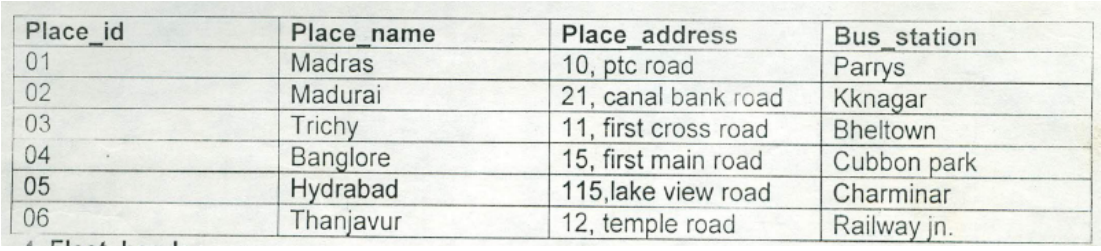
Cat_code	Cate_desc
01	super delux
02	delux
03	super fast
04	normal

route_header



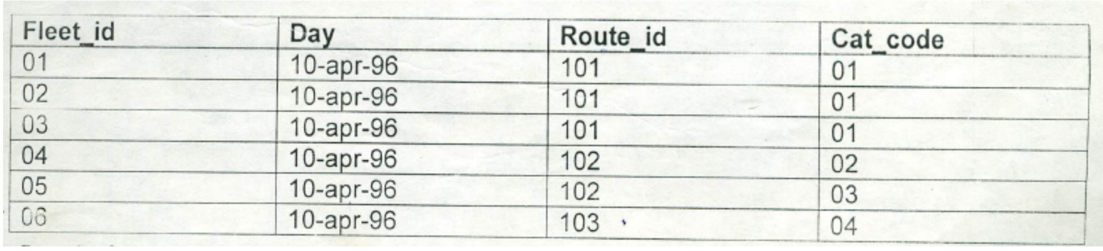
Route_id	Route_no	Cate_code	Origin	Destination	Fare	Distance	Capacity
101	33	01	Madurai	Madras	35	250	50
102	25	02	Trichy	Madurai	40	159	50
103	15	03	Thanjavur	Madurai	59	140	50
104	36	04	Madras	Banglore	79	375	50
105	40	01	Banglore	Madras	80	375	50
106	38	02	Madras	Madurai	39	250	50
107	39	03	Hydrabad	Madras	50	430	50
108	41	04	Madras	Cochin	47	576	50

Place Header:



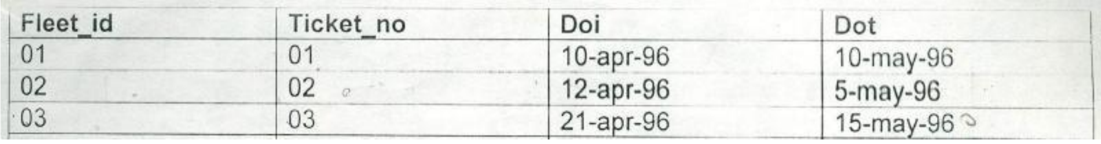
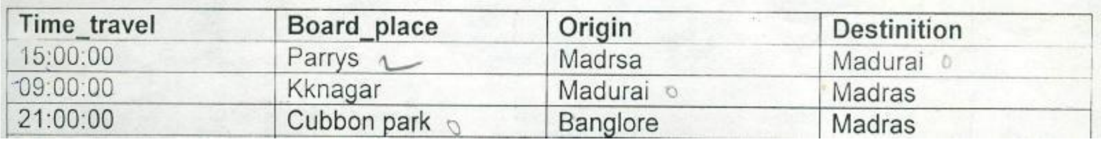
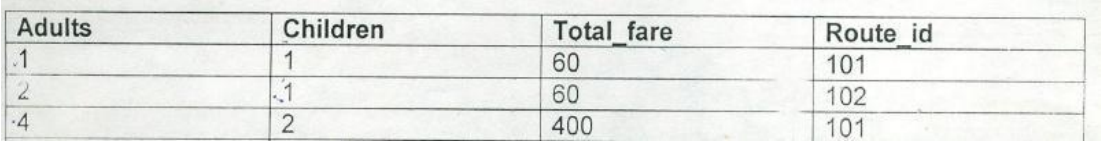
Place_id	Place_name	Place_address	Bus_station
01	Madras	10, ptc road	Parrys
02	Madurai	21, canal bank road	Kknagar
03	Trichy	11, first cross road	Bheltown
04	Banglore	15, first main road	Cubbon park
05	Hydrabad	115,lake view road	Charminar
06	Thanjavur	12, temple road	Railway jn.

Fleet Header:



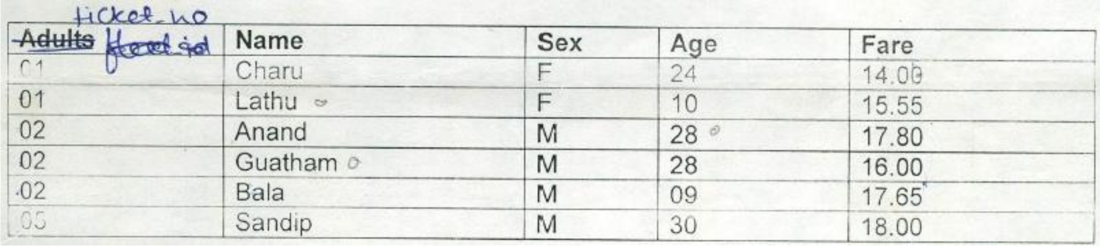
Fleet_id	Day	Route_id	Cat_code
01	10-apr-96	101	01
02	10-apr-96	101	01
03	10-apr-96	101	01
04	10-apr-96	102	02
05	10-apr-96	102	03
06	10-apr-96	103	04

Ticket Header:

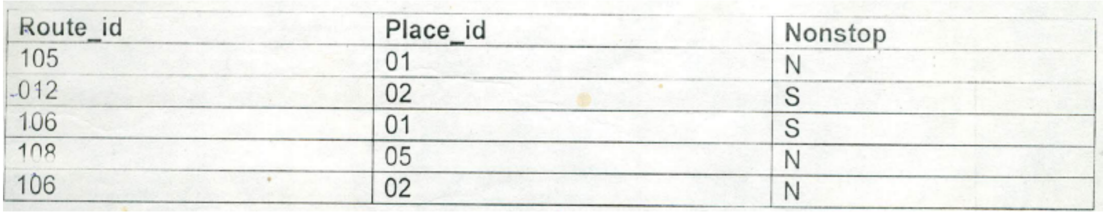
Fleet_id	Ticket_no	Doi	Dot	Time_travel	Board_place	Origin	Destinition	Adults	Children	Total_fare	Route_id
01	01	10-apr-96	10-may-96	15:00:00	Parrys	Madrsa	Madurai	1	1	60	101
02	02	12-apr-96	5-may-96	09:00:00	Kknagar	Madurai	Madras	2	1	60	102
03	03	21-apr-96	15-may-96	21:00:00	Cubbon park	Banglore	Madras	4	2	400	101

Ticket Detail:



~~Adults~~ ~~Fleet id~~ ticket_no	Name	Sex	Age	Fare
01	Charu	F	24	14.00
01	Lathu	F	10	15.55
02	Anand	M	28	17.80
02	Guatham	M	28	16.00
02	Bala	M	09	17.65
05	Sandip	M	30	18.00

Route Detail:



Route_id	Place_id	Nonstop
105	01	N
012	02	S
106	01	S
108	05	N
106	02	N

2. Display all records of the Category Header table.
3. Display place name and place address.
4. Display distinct destination from route header table.
5. Check what will be the fare if it is incremented by 10 rs for each route. (Give a new column alias as new_fare).
6. Write a query to change the fare of Cate-code 01 from 35 to 40.
7. Modify the distance by 300 and fare by 100 of cate_code 04
8. Display structure of table route detail.
9. Write a query to change the age of ‘Anand’ from 28 to 30.
10. Write a query to insert a new record into Route_Detail table with details as below:
    Route_id: 105 , Place_id : 01, NonStop: S
11. Write a query to delete rows inserted above from the Route_Detail table.

PART B

(PART B: TO BE COMPLETED BY STUDENTS)

(Students must submit the soft copy as per following segments within two hours of the practical. The soft copy must be uploaded on the Blackboard or emailed to the concerned lab in charge faculties at the end of the practical in case the there is no Black board access available)

Roll No. I066	Name: Srihari Thyagarajan
Program : B Tech Artificial Intelligence	Division: I
Batch: B3	Date of Experiment: 05/08/2022
Date of Submission: 05/08/2022	Grade :

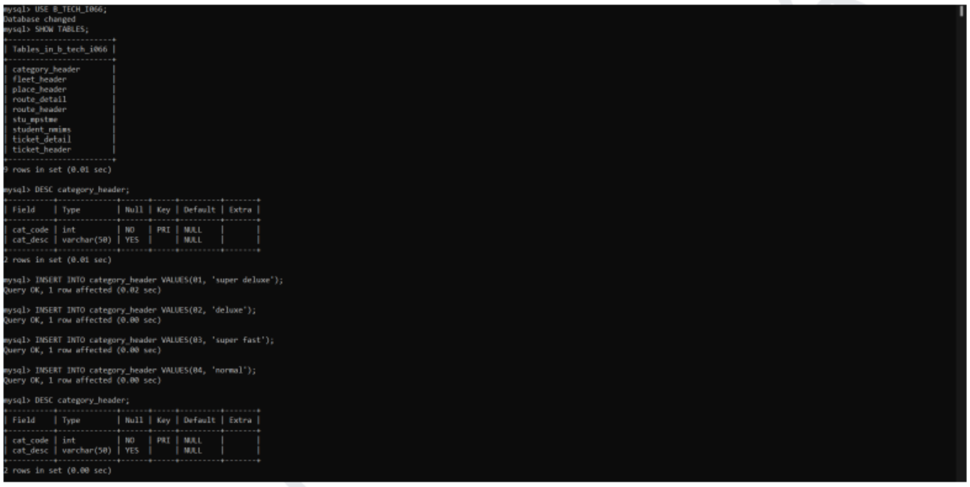
B.1 Table Screenshots:

1. **mysql> INSERT INTO category_header VALUES(01, 'super deluxe');**

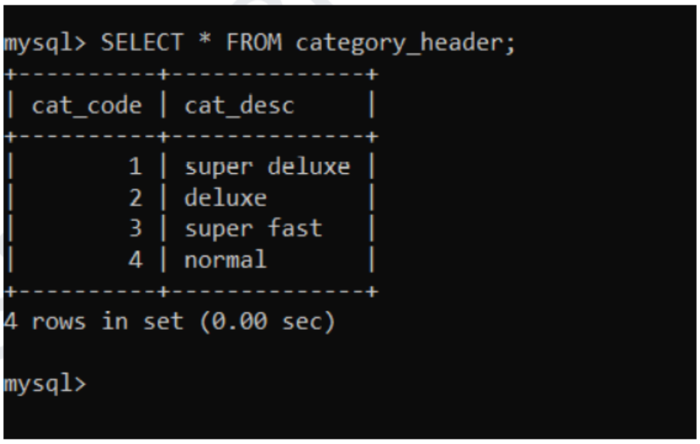
   mysql> INSERT INTO category_header VALUES(02, 'deluxe');

   mysql> INSERT INTO category_header VALUES(03, 'super fast');

   mysql> INSERT INTO category_header VALUES(04, 'normal');



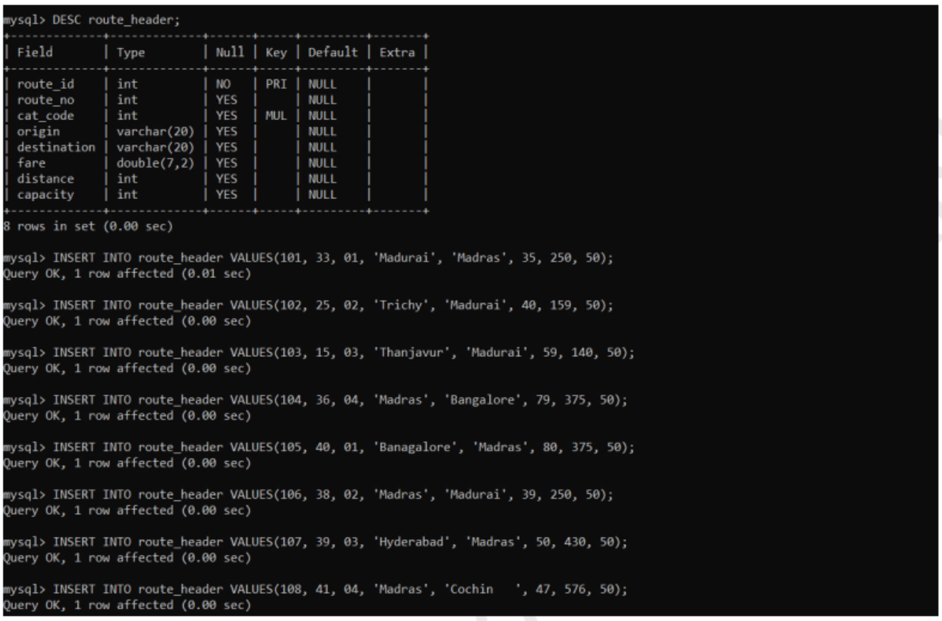
```
mysql> USE B_TECH_I066;
Database changed
mysql> SHOW TABLES;
+-----------------------+
| Tables_in_b_tech_i066 |
+-----------------------+
| category_header       |
| fleet_header          |
| place_header          |
| route_detail          |
| route_header          |
| stu_mpstme            |
| student_nmims         |
| ticket_detail         |
| ticket_header         |
+-----------------------+
9 rows in set (0.01 sec)

mysql> DESC category_header;
+----------+-------------+------+-----+---------+-------+
| Field    | Type        | Null | Key | Default | Extra |
+----------+-------------+------+-----+---------+-------+
| cat_code | int         | NO   | PRI | NULL    |       |
| cat_desc | varchar(50) | YES  |     | NULL    |       |
+----------+-------------+------+-----+---------+-------+
2 rows in set (0.01 sec)

mysql> INSERT INTO category_header VALUES(01, 'super deluxe');
Query OK, 1 row affected (0.02 sec)

mysql> INSERT INTO category_header VALUES(02, 'deluxe');
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO category_header VALUES(03, 'super fast');
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO category_header VALUES(04, 'normal');
Query OK, 1 row affected (0.00 sec)

mysql> DESC category_header;
+----------+-------------+------+-----+---------+-------+
| Field    | Type        | Null | Key | Default | Extra |
+----------+-------------+------+-----+---------+-------+
| cat_code | int         | NO   | PRI | NULL    |       |
| cat_desc | varchar(50) | YES  |     | NULL    |       |
+----------+-------------+------+-----+---------+-------+
2 rows in set (0.00 sec)
```



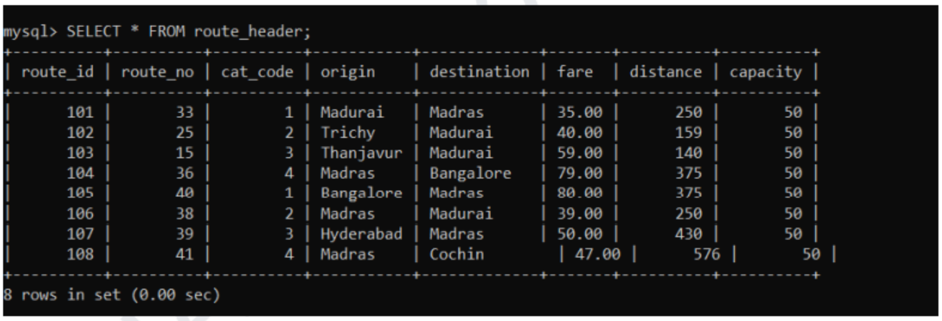
```
mysql> SELECT * FROM category_header;
+----------+--------------+
| cat_code | cat_desc     |
+----------+--------------+
|        1 | super deluxe |
|        2 | deluxe       |
|        3 | super fast   |
|        4 | normal       |
+----------+--------------+
4 rows in set (0.00 sec)

mysql>
```

2.

```
mysql> INSERT INTO route_header VALUES(101, 33, 01, 'Madurai', 'Madras', 35,
250, 50);

mysql> INSERT INTO route_header VALUES(102, 25, 02, 'Trichy', 'Madurai', 40, 159,
50);

mysql> INSERT INTO route_header VALUES(103, 15, 03, 'Thanjavur', 'Madurai', 59,
140, 50);

mysql> INSERT INTO route_header VALUES(104, 36, 04, 'Madras', 'Bangalore', 79,
375, 50);

mysql> INSERT INTO route_header VALUES(105, 40, 01, 'Bangalore', 'Madras', 80,
375, 50);

mysql> INSERT INTO route_header VALUES(106, 38, 02, 'Madras', 'Madurai', 39,
250, 50);


mysql> INSERT INTO route_header VALUES(107, 39, 03, 'Hyderabad', 'Madras', 50,
430, 50);

mysql> INSERT INTO route_header VALUES(108, 41, 04, 'Madras', 'Cochin ', 47, 576,
50);
```



```
mysql> DESC route_header;
+-------------+-------------+------+-----+---------+-------+
| Field       | Type        | Null | Key | Default | Extra |
+-------------+-------------+------+-----+---------+-------+
| route_id    | int         | NO   | PRI | NULL    |       |
| route_no    | int         | YES  |     | NULL    |       |
| cat_code    | int         | YES  | MUL | NULL    |       |
| origin      | varchar(20) | YES  |     | NULL    |       |
| destination | varchar(20) | YES  |     | NULL    |       |
| fare        | double(7,2) | YES  |     | NULL    |       |
| distance    | int         | YES  |     | NULL    |       |
| capacity    | int         | YES  |     | NULL    |       |
+-------------+-------------+------+-----+---------+-------+
8 rows in set (0.00 sec)

mysql> INSERT INTO route_header VALUES(101, 33, 01, 'Madurai', 'Madras', 35, 250, 50);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO route_header VALUES(102, 25, 02, 'Trichy', 'Madurai', 40, 159, 50);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO route_header VALUES(103, 15, 03, 'Thanjavur', 'Madurai', 59, 140, 50);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO route_header VALUES(104, 36, 04, 'Madras', 'Bangalore', 79, 375, 50);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO route_header VALUES(105, 40, 01, 'Banagalore', 'Madras', 80, 375, 50);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO route_header VALUES(106, 38, 02, 'Madras', 'Madurai', 39, 250, 50);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO route_header VALUES(107, 39, 03, 'Hyderabad', 'Madras', 50, 430, 50);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO route_header VALUES(108, 41, 04, 'Madras', 'Cochin   ', 47, 576, 50);
Query OK, 1 row affected (0.00 sec)
```



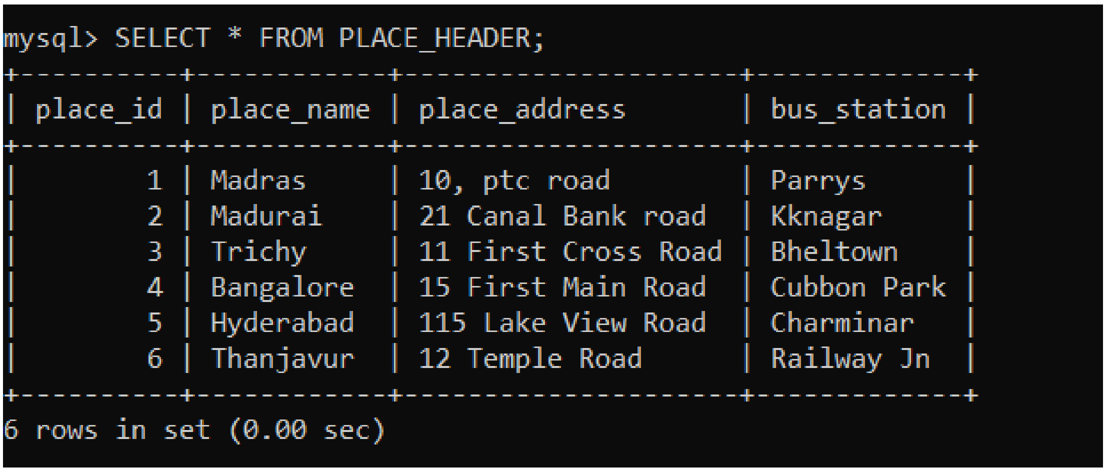
```
mysql> SELECT * FROM route_header;
+----------+----------+----------+-----------+-------------+-------+----------+----------+
| route_id | route_no | cat_code | origin    | destination | fare  | distance | capacity |
+----------+----------+----------+-----------+-------------+-------+----------+----------+
|      101 |       33 |        1 | Madurai   | Madras      | 35.00 |      250 |       50 |
|      102 |       25 |        2 | Trichy    | Madurai     | 40.00 |      159 |       50 |
|      103 |       15 |        3 | Thanjavur | Madurai     | 59.00 |      140 |       50 |
|      104 |       36 |        4 | Madras    | Bangalore   | 79.00 |      375 |       50 |
|      105 |       40 |        1 | Bangalore | Madras      | 80.00 |      375 |       50 |
|      106 |       38 |        2 | Madras    | Madurai     | 39.00 |      250 |       50 |
|      107 |       39 |        3 | Hyderabad | Madras      | 50.00 |      430 |       50 |
|      108 |       41 |        4 | Madras    | Cochin         | 47.00 |      576 |       50 |
+----------+----------+----------+-----------+-------------+-------+----------+----------+
8 rows in set (0.00 sec)
```

3. INSERT INTO place_header VALUES(02,"Madurai", "21 Canal Bank road", "Kknagar");

mysql> INSERT INTO place_header VALUES(03,"Trichy", "11 First Cross Road", "Bheltown");
INSERT INTO place_header VALUES(04, "Bangalore", "15 First Main Road", "Cubbon Park");

mysql> INSERT INTO place_header VALUES(05, "Hyderabad", "115 Lake View Road", "Charminar");

mysql> INSERT INTO place_header VALUES(06, "Thanjavur", "12 Temple Road", "Railway Jn");



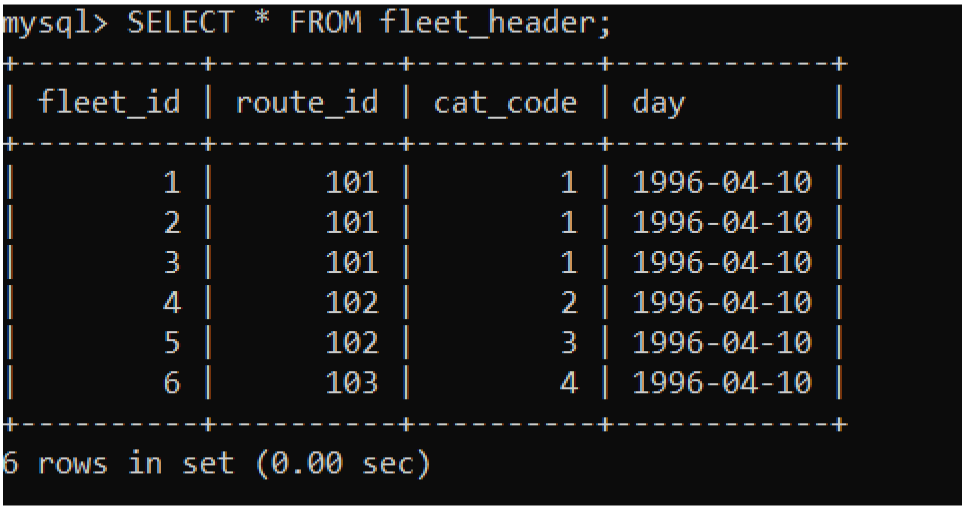
```
mysql> SELECT * FROM PLACE_HEADER;
+----------+------------+---------------------+-------------+
| place_id | place_name | place_address       | bus_station |
+----------+------------+---------------------+-------------+
|        1 | Madras     | 10, ptc road        | Parrys      |
|        2 | Madurai    | 21 Canal Bank road  | Kknagar     |
|        3 | Trichy     | 11 First Cross Road | Bheltown    |
|        4 | Bangalore  | 15 First Main Road  | Cubbon Park |
|        5 | Hyderabad  | 115 Lake View Road  | Charminar   |
|        6 | Thanjavur  | 12 Temple Road      | Railway Jn  |
+----------+------------+---------------------+-------------+
6 rows in set (0.00 sec)
```

4. **mysql> INSERT INTO fleet_header VALUES(02, 101, 01, '1996-04-10');**

   mysql> INSERT INTO fleet_header VALUES(03, 101, 01, '1996-04-10');
   mysql> INSERT INTO fleet_header VALUES(04, 102, 02, '1996-04-10');

   mysql> INSERT INTO fleet_header VALUES(05, 102, 03, '1996-04-10');

   mysql> INSERT INTO fleet_header VALUES(06, 103, 04, '1996-04-10');

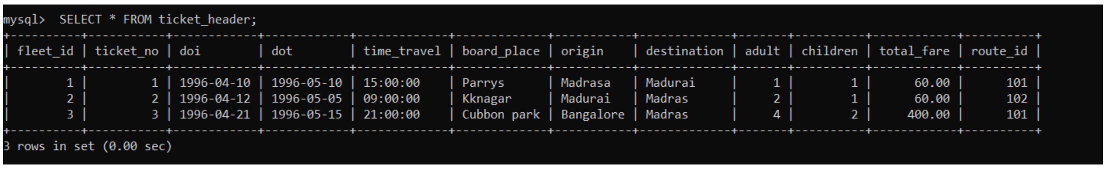


```
mysql> SELECT * FROM fleet_header;
+----------+----------+----------+------------+
| fleet_id | route_id | cat_code | day        |
+----------+----------+----------+------------+
|        1 |      101 |        1 | 1996-04-10 |
|        2 |      101 |        1 | 1996-04-10 |
|        3 |      101 |        1 | 1996-04-10 |
|        4 |      102 |        2 | 1996-04-10 |
|        5 |      102 |        3 | 1996-04-10 |
|        6 |      103 |        4 | 1996-04-10 |
+----------+----------+----------+------------+
6 rows in set (0.00 sec)
```

5. **mysql> INSERT INTO ticket_header VALUES(01, 01, '1996-04-10', '1996-05-10', '15:00:00', 'Parrys', 'Madrasa', 'Madurai', 1, 1, 60, 101);**

mysql> INSERT INTO ticket_header VALUES(02, 02, '1996-04-12', '1996-05-05', '09:00:00', 'Kknagar', 'Madurai', 'Madras', 2, 1, 60, 102);

mysql> INSERT INTO ticket_header VALUES(03, 03, '1996-04-21', '1996-05-15', '21:00:00', 'Cubbon park', 'Bangalore', 'Madras', 4, 2, 400, 101);



```
mysql>  SELECT * FROM ticket_header;
+----------+-----------+------------+------------+-------------+-------------+-----------+-------------+-------+----------+------------+----------+
| fleet_id | ticket_no | doi        | dot        | time_travel | board_place | origin    | destination | adult | children | total_fare | route_id |
+----------+-----------+------------+------------+-------------+-------------+-----------+-------------+-------+----------+------------+----------+
|        1 |         1 | 1996-04-10 | 1996-05-10 | 15:00:00    | Parrys      | Madrasa   | Madurai     |     1 |        1 |      60.00 |      101 |
|        2 |         2 | 1996-04-12 | 1996-05-05 | 09:00:00    | Kknagar     | Madurai   | Madras      |     2 |        1 |      60.00 |      102 |
|        3 |         3 | 1996-04-21 | 1996-05-15 | 21:00:00    | Cubbon park | Bangalore | Madras      |     4 |        2 |     400.00 |      101 |
+----------+-----------+------------+------------+-------------+-------------+-----------+-------------+-------+----------+------------+----------+
3 rows in set (0.00 sec)
```

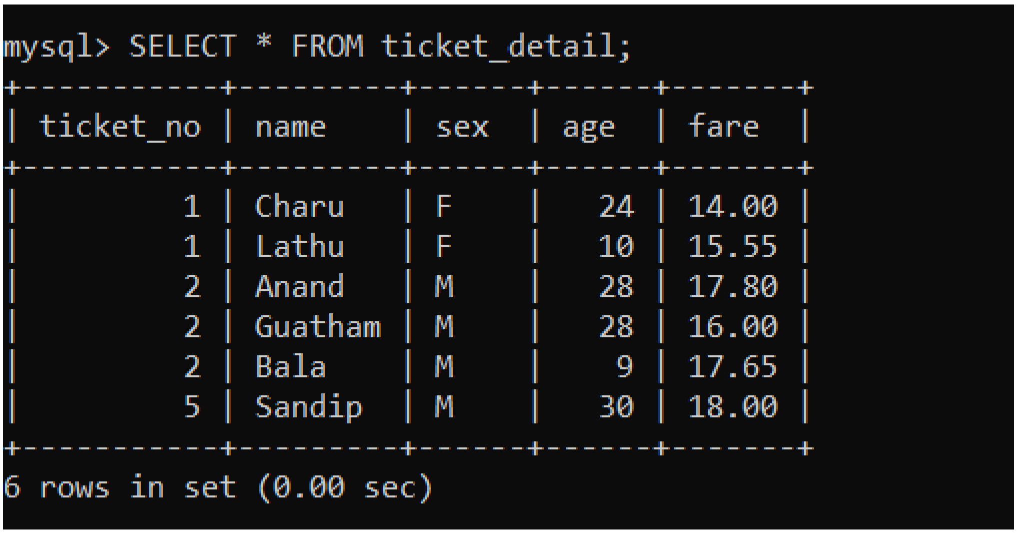
6. **mysql> INSERT INTO ticket_detail VALUES(01, 'Charu', 'F', 24, 14.00);**

   mysql> INSERT INTO ticket_detail VALUES(01, 'Lathu', 'F', 10, 15.55);

   mysql> INSERT INTO ticket_detail VALUES(02, 'Anand', 'M', 28, 17.80);

   mysql> INSERT INTO ticket_detail VALUES(02, 'Guatham', 'M', 28, 16.00);

   mysql> INSERT INTO ticket_detail VALUES(02, 'Bala', 'M', 09, 17.65);
   mysql> INSERT INTO ticket_detail VALUES(05, 'Sandip', 'M', 30, 18.00);



```
mysql> SELECT * FROM ticket_detail;
+-----------+---------+------+------+-------+
| ticket_no | name    | sex  | age  | fare  |
+-----------+---------+------+------+-------+
|         1 | Charu   | F    |   24 | 14.00 |
|         1 | Lathu   | F    |   10 | 15.55 |
|         2 | Anand   | M    |   28 | 17.80 |
|         2 | Guatham | M    |   28 | 16.00 |
|         2 | Bala    | M    |    9 | 17.65 |
|         5 | Sandip  | M    |   30 | 18.00 |
+-----------+---------+------+------+-------+
6 rows in set (0.00 sec)
```

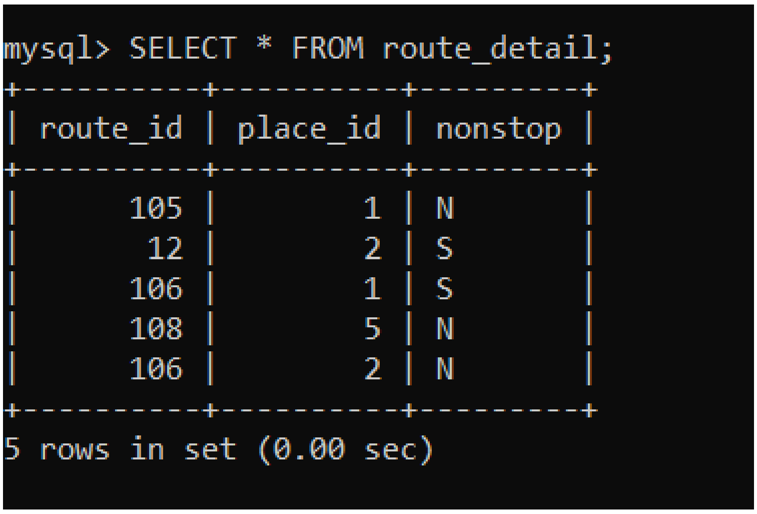
7. **mysql> INSERT INTO route_detail VALUES(105, 01, 'N');**

   mysql> INSERT INTO route_detail VALUES(012, 02, 'S');

   mysql> INSERT INTO route_detail VALUES(106, 01, 'S');

   mysql> INSERT INTO route_detail VALUES(108, 05, 'N');

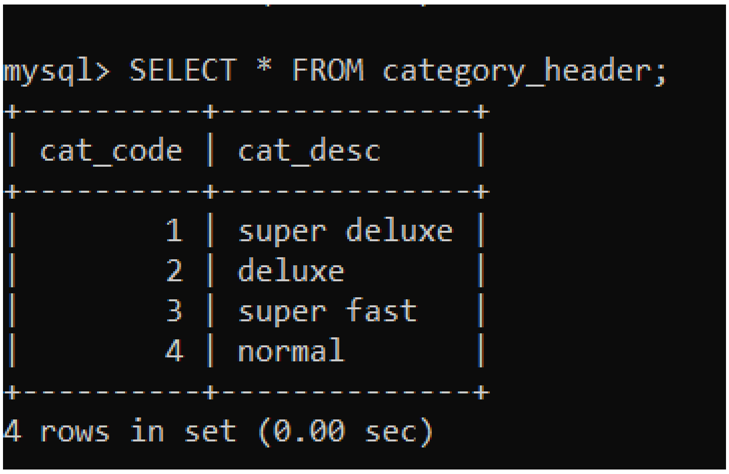
   mysql> INSERT INTO route_detail VALUES(106, 02, 'N');



```
mysql> SELECT * FROM route_detail;
+----------+----------+---------+
| route_id | place_id | nonstop |
+----------+----------+---------+
|      105 |        1 | N       |
|       12 |        2 | S       |
|      106 |        1 | S       |
|      108 |        5 | N       |
|      106 |        2 | N       |
+----------+----------+---------+
5 rows in set (0.00 sec)
```

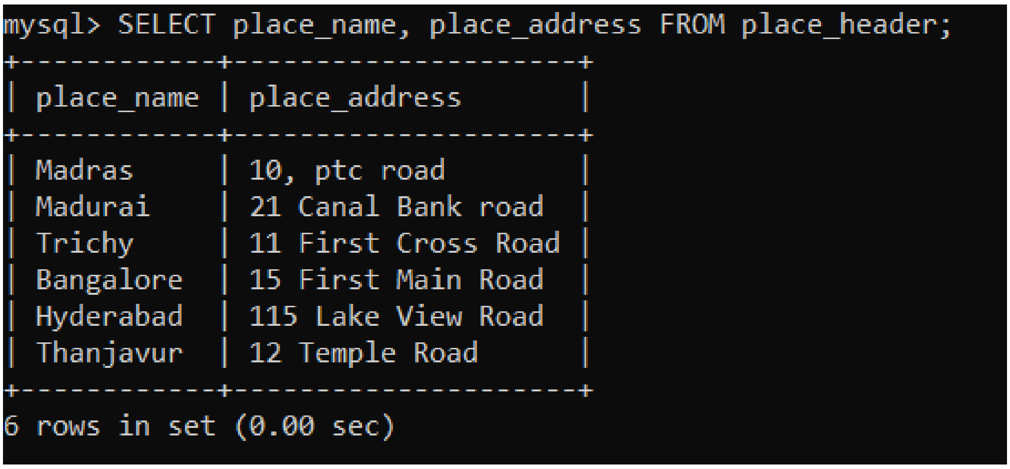
B.2 Queries:

1. **SELECT * FROM category_header;**



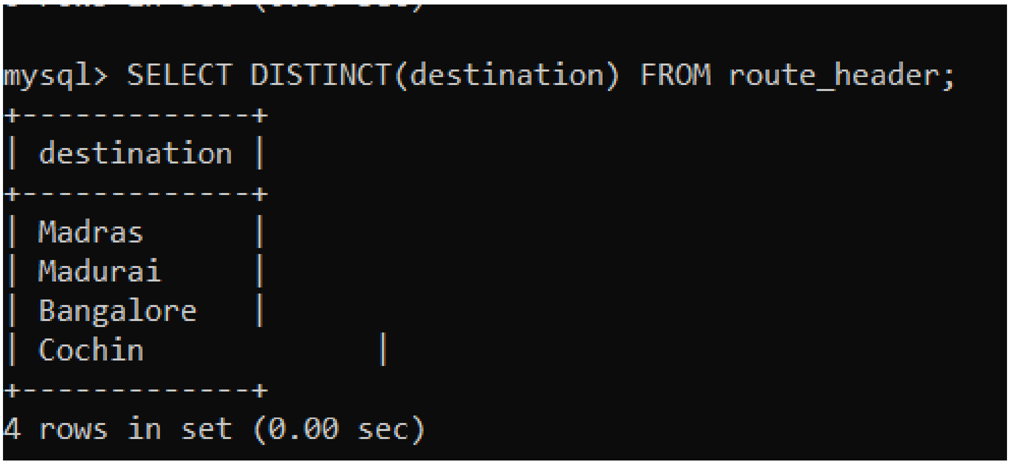
```
mysql> SELECT * FROM category_header;
+----------+--------------+
| cat_code | cat_desc     |
+----------+--------------+
|        1 | super deluxe |
|        2 | deluxe       |
|        3 | super fast   |
|        4 | normal       |
+----------+--------------+
4 rows in set (0.00 sec)
```

2. **SELECT place_name, place_address FROM place_header;**



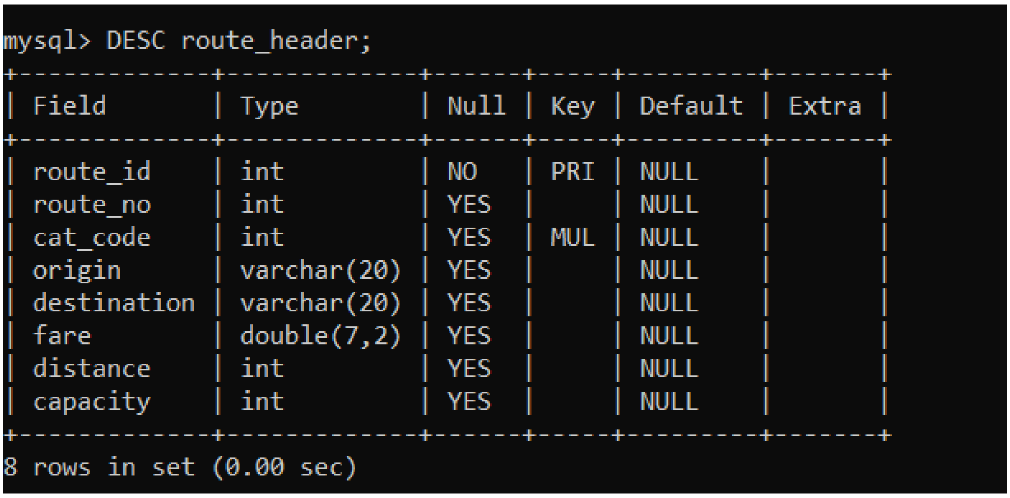
```
mysql> SELECT place_name, place_address FROM place_header;
+------------+--------------------+
| place_name | place_address      |
+------------+--------------------+
| Madras     | 10, ptc road       |
| Madurai    | 21 Canal Bank road |
| Trichy     | 11 First Cross Road |
| Bangalore  | 15 First Main Road |
| Hyderabad  | 115 Lake View Road |
| Thanjavur  | 12 Temple Road     |
+------------+--------------------+
6 rows in set (0.00 sec)
```

3. **SELECT DISTINCT(destination) FROM route_header;**



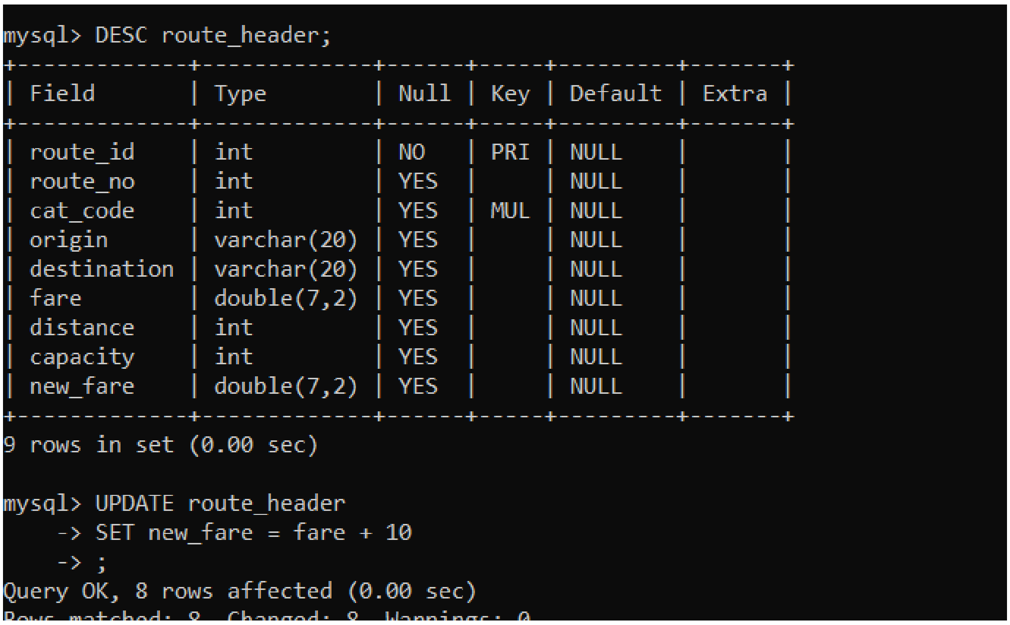
```
mysql> SELECT DISTINCT(destination) FROM route_header;
+-------------+
| destination |
+-------------+
| Madras      |
| Madurai     |
| Bangalore   |
| Cochin             |
+-------------+
4 rows in set (0.00 sec)
```

4. **DESC route_header;**



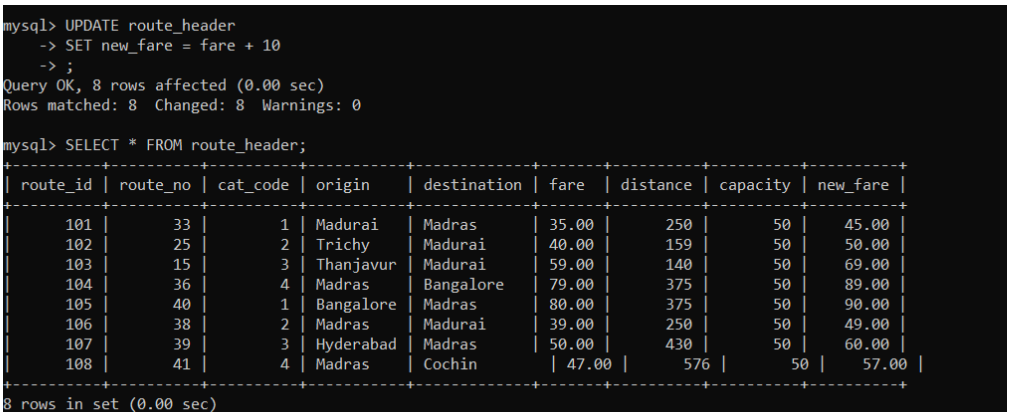
```
mysql> DESC route_header;
+-------------+-------------+------+-----+---------+-------+
| Field       | Type        | Null | Key | Default | Extra |
+-------------+-------------+------+-----+---------+-------+
| route_id    | int         | NO   | PRI | NULL    |       |
| route_no    | int         | YES  |     | NULL    |       |
| cat_code    | int         | YES  | MUL | NULL    |       |
| origin      | varchar(20) | YES  |     | NULL    |       |
| destination | varchar(20) | YES  |     | NULL    |       |
| fare        | double(7,2) | YES  |     | NULL    |       |
| distance    | int         | YES  |     | NULL    |       |
| capacity    | int         | YES  |     | NULL    |       |
+-------------+-------------+------+-----+---------+-------+
8 rows in set (0.00 sec)
```

5. **ALTER TABLE route_header**
   ADD COLUMN new_fare double(7, 2);



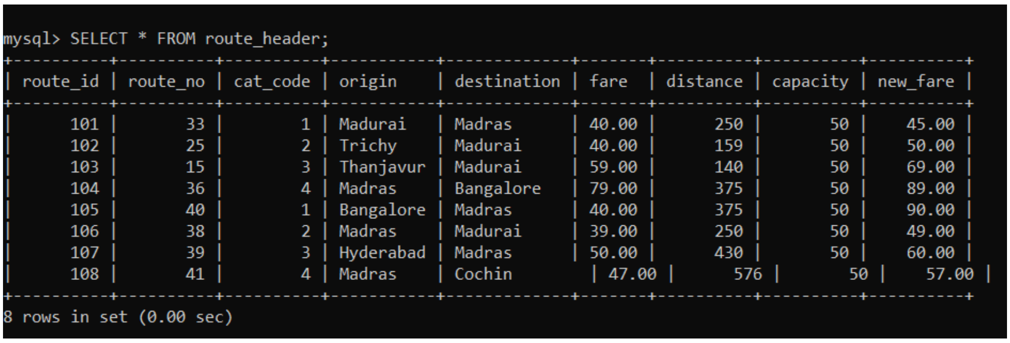
```
mysql> DESC route_header;
+-------------+-------------+------+-----+---------+-------+
| Field       | Type        | Null | Key | Default | Extra |
+-------------+-------------+------+-----+---------+-------+
| route_id    | int         | NO   | PRI | NULL    |       |
| route_no    | int         | YES  |     | NULL    |       |
| cat_code    | int         | YES  | MUL | NULL    |       |
| origin      | varchar(20) | YES  |     | NULL    |       |
| destination | varchar(20) | YES  |     | NULL    |       |
| fare        | double(7,2) | YES  |     | NULL    |       |
| distance    | int         | YES  |     | NULL    |       |
| capacity    | int         | YES  |     | NULL    |       |
| new_fare    | double(7,2) | YES  |     | NULL    |       |
+-------------+-------------+------+-----+---------+-------+
9 rows in set (0.00 sec)

mysql> UPDATE route_header
    -> SET new_fare = fare + 10
    -> ;
Query OK, 8 rows affected (0.00 sec)
Rows matched: 8  Changed: 8  Warnings: 0
```

UPDATE route_header
   -> SET new_fare = fare + 10;



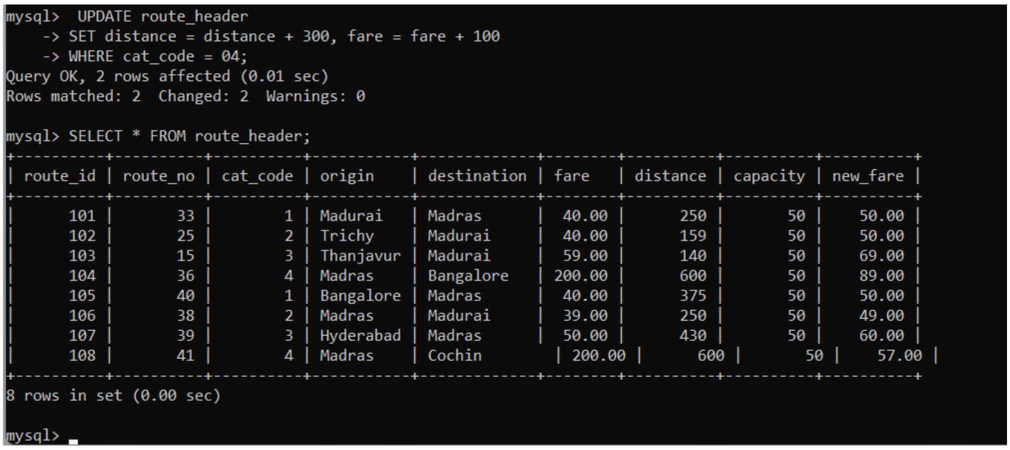
```
mysql> UPDATE route_header
    -> SET new_fare = fare + 10
    -> ;
Query OK, 8 rows affected (0.00 sec)
Rows matched: 8  Changed: 8  Warnings: 0

mysql> SELECT * FROM route_header;
+----------+----------+----------+-----------+-------------+-------+----------+----------+----------+
| route_id | route_no | cat_code | origin    | destination | fare  | distance | capacity | new_fare |
+----------+----------+----------+-----------+-------------+-------+----------+----------+----------+
|      101 |       33 |        1 | Madurai   | Madras      | 35.00 |      250 |       50 |    45.00 |
|      102 |       25 |        2 | Trichy    | Madurai     | 40.00 |      159 |       50 |    50.00 |
|      103 |       15 |        3 | Thanjavur | Madurai     | 59.00 |      140 |       50 |    69.00 |
|      104 |       36 |        4 | Madras    | Bangalore   | 79.00 |      375 |       50 |    89.00 |
|      105 |       40 |        1 | Bangalore | Madras      | 80.00 |      375 |       50 |    90.00 |
|      106 |       38 |        2 | Madras    | Madurai     | 39.00 |      250 |       50 |    49.00 |
|      107 |       39 |        3 | Hyderabad | Madras      | 50.00 |      430 |       50 |    60.00 |
|      108 |       41 |        4 | Madras    | Cochin        | 47.00 |      576 |       50 |    57.00 |
+----------+----------+----------+-----------+-------------+-------+----------+----------+----------+
8 rows in set (0.00 sec)
```

6. **UPDATE route_header**
   -> SET fare = 40
   -> WHERE cat_code = 1;



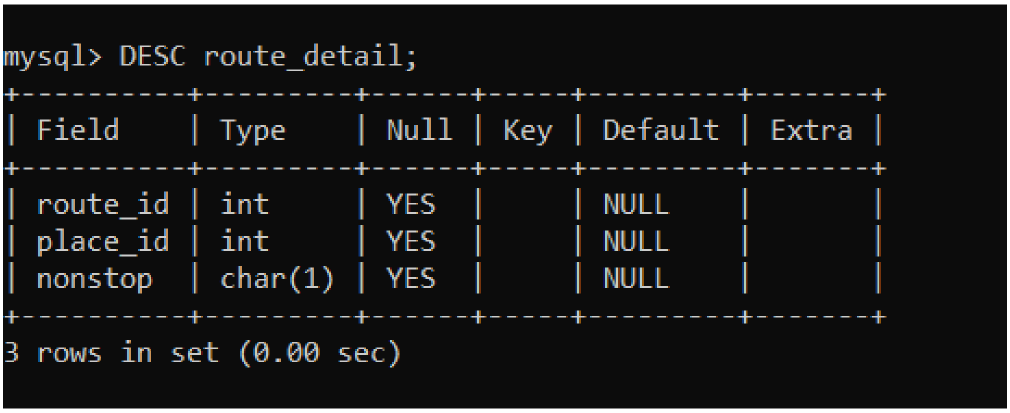
```
mysql> SELECT * FROM route_header;
+----------+----------+----------+-----------+-------------+-------+----------+----------+----------+
| route_id | route_no | cat_code | origin    | destination | fare  | distance | capacity | new_fare |
+----------+----------+----------+-----------+-------------+-------+----------+----------+----------+
|      101 |       33 |        1 | Madurai   | Madras      | 40.00 |      250 |       50 |    45.00 |
|      102 |       25 |        2 | Trichy    | Madurai     | 40.00 |      159 |       50 |    50.00 |
|      103 |       15 |        3 | Thanjavur | Madurai     | 59.00 |      140 |       50 |    69.00 |
|      104 |       36 |        4 | Madras    | Bangalore   | 79.00 |      375 |       50 |    89.00 |
|      105 |       40 |        1 | Bangalore | Madras      | 40.00 |      375 |       50 |    90.00 |
|      106 |       38 |        2 | Madras    | Madurai     | 39.00 |      250 |       50 |    49.00 |
|      107 |       39 |        3 | Hyderabad | Madras      | 50.00 |      430 |       50 |    60.00 |
|      108 |       41 |        4 | Madras    | Cochin        | 47.00 |      576 |       50 |    57.00 |
+----------+----------+----------+-----------+-------------+-------+----------+----------+----------+
8 rows in set (0.00 sec)
```

7. **mysql> UPDATE route_header**
   -> SET distance = 300, fare = 100
   -> WHERE cat_code = 04;



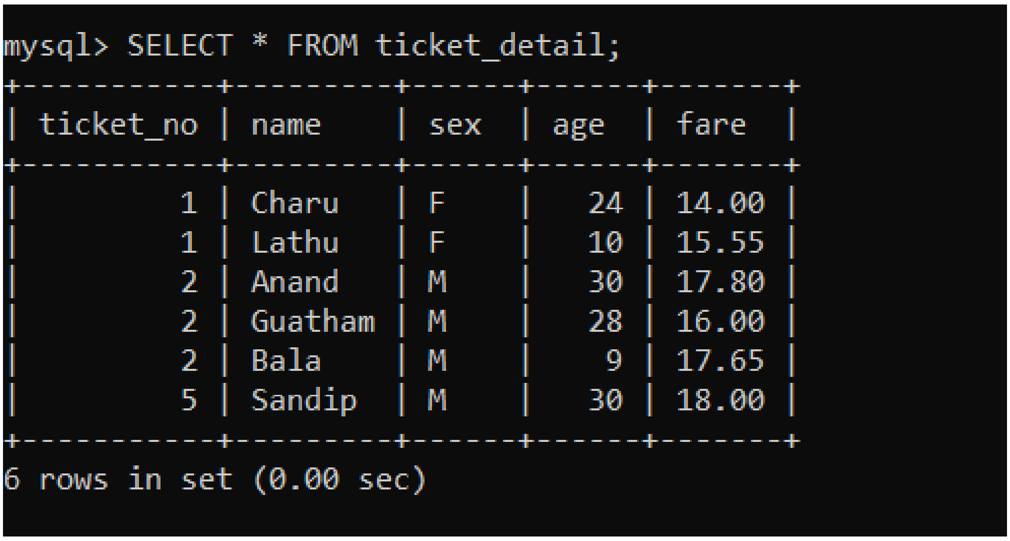
```
mysql>  UPDATE route_header
    -> SET distance = distance + 300, fare = fare + 100
    -> WHERE cat_code = 04;
Query OK, 2 rows affected (0.01 sec)
Rows matched: 2  Changed: 2  Warnings: 0

mysql> SELECT * FROM route_header;
+----------+----------+----------+-----------+-------------+--------+----------+----------+----------+
| route_id | route_no | cat_code | origin    | destination | fare   | distance | capacity | new_fare |
+----------+----------+----------+-----------+-------------+--------+----------+----------+----------+
|      101 |       33 |        1 | Madurai   | Madras      |  40.00 |      250 |       50 |    50.00 |
|      102 |       25 |        2 | Trichy    | Madurai     |  40.00 |      159 |       50 |    50.00 |
|      103 |       15 |        3 | Thanjavur | Madurai     |  59.00 |      140 |       50 |    69.00 |
|      104 |       36 |        4 | Madras    | Bangalore   | 200.00 |      600 |       50 |    89.00 |
|      105 |       40 |        1 | Bangalore | Madras      |  40.00 |      375 |       50 |    50.00 |
|      106 |       38 |        2 | Madras    | Madurai     |  39.00 |      250 |       50 |    49.00 |
|      107 |       39 |        3 | Hyderabad | Madras      |  50.00 |      430 |       50 |    60.00 |
|      108 |       41 |        4 | Madras    | Cochin        | 200.00 |      600 |       50 |    57.00 |
+----------+----------+----------+-----------+-------------+--------+----------+----------+----------+
8 rows in set (0.00 sec)

mysql> _
```

8. **DESC route_detail;**



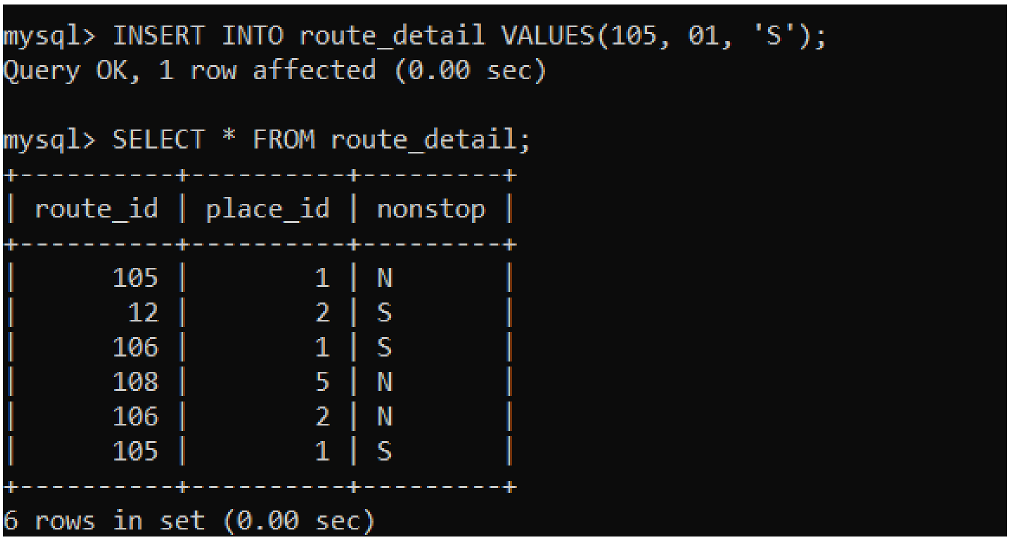
```
mysql> DESC route_detail;
+----------+---------+------+-----+---------+-------+
| Field    | Type    | Null | Key | Default | Extra |
+----------+---------+------+-----+---------+-------+
| route_id | int     | YES  |     | NULL    |       |
| place_id | int     | YES  |     | NULL    |       |
| nonstop  | char(1) | YES  |     | NULL    |       |
+----------+---------+------+-----+---------+-------+
3 rows in set (0.00 sec)
```

9. **UPDATE ticket_detail**
   -> SET age = 30
   -> WHERE name = 'Anand';



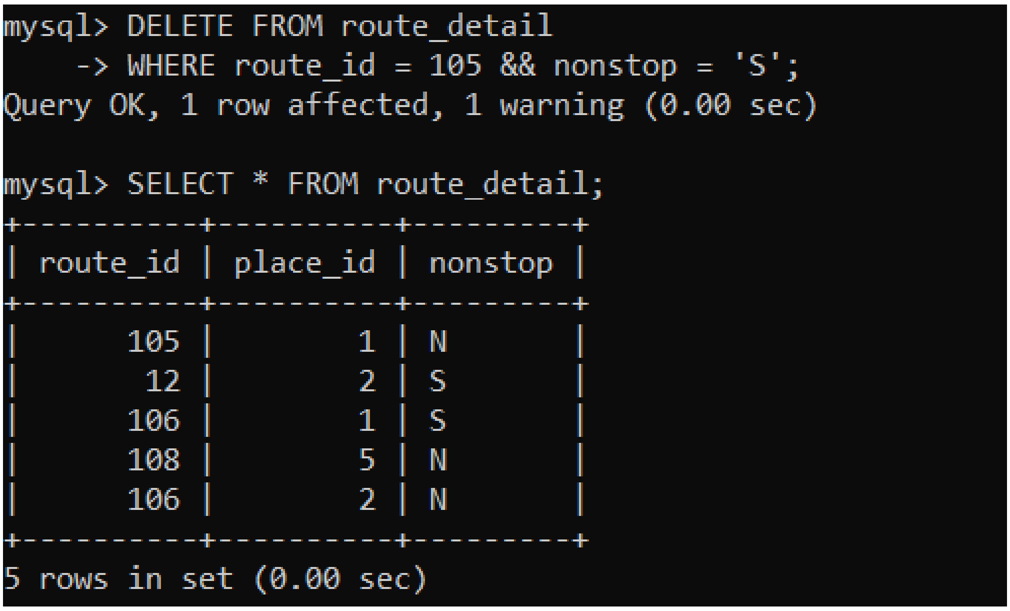
```
mysql> SELECT * FROM ticket_detail;
+-----------+---------+------+------+-------+
| ticket_no | name    | sex  | age  | fare  |
+-----------+---------+------+------+-------+
|         1 | Charu   | F    |   24 | 14.00 |
|         1 | Lathu   | F    |   10 | 15.55 |
|         2 | Anand   | M    |   30 | 17.80 |
|         2 | Guatham | M    |   28 | 16.00 |
|         2 | Bala    | M    |    9 | 17.65 |
|         5 | Sandip  | M    |   30 | 18.00 |
+-----------+---------+------+------+-------+
6 rows in set (0.00 sec)
```

10. **mysql> INSERT INTO route_detail VALUES(105, 01, 'S');**



```
mysql> INSERT INTO route_detail VALUES(105, 01, 'S');
Query OK, 1 row affected (0.00 sec)

mysql> SELECT * FROM route_detail;
+----------+----------+---------+
| route_id | place_id | nonstop |
+----------+----------+---------+
|      105 |        1 | N       |
|       12 |        2 | S       |
|      106 |        1 | S       |
|      108 |        5 | N       |
|      106 |        2 | N       |
|      105 |        1 | S       |
+----------+----------+---------+
6 rows in set (0.00 sec)
```

11. **DELETE FROM route_detail**
    -> WHERE route_id = 105 && nonstop = 'S';



```
mysql> DELETE FROM route_detail
    -> WHERE route_id = 105 && nonstop = 'S';
Query OK, 1 row affected, 1 warning (0.00 sec)

mysql> SELECT * FROM route_detail;
+----------+----------+---------+
| route_id | place_id | nonstop |
+----------+----------+---------+
|      105 |        1 | N       |
|       12 |        2 | S       |
|      106 |        1 | S       |
|      108 |        5 | N       |
|      106 |        2 | N       |
+----------+----------+---------+
5 rows in set (0.00 sec)
```

B.3 Curiosity Questions

State difference between primary key and super key:
Ans)

Primary Key	Super Key
It is an attribute that helps in identifying the attributes in a relation.	Super key is an attribute/ set of attributes that helps in distinguishing the attributes in the table.
It is a minimal super key.	Not all super keys can be primary keys.
Since it helps in distinguishing other attributes in the relation, it cannot contain values.	It can contain NULL values.
Number of Primary keys are less than super keys in a relation.	Supers keys are more in number than Primary Keys in a relation.

B.4 Conclusion:

From the above experiment, I was able to learn the following:

1. Implementing DDL commands with various constraints:
    - Primary key
    - Not Null
    - Unique key
    - Foreign key
    - Check constraint
2. Implementing DML commands like Insert, Update, Delete and Select/ DESC, etc.
3. Being able to retrieve various data stored in databases and modify them accordingly.