

**VISVESVARAYA TECHNOLOGICAL UNIVERSITY  
JNANA SANGAMA, BELGAUM - 590014**



**A Report on  
“Travel Agency Management System”**

**Submitted in the partial fulfillment for Database Management Laboratory: An  
Open Ended Problem.**

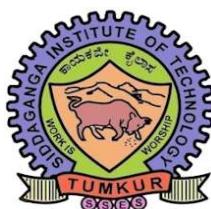
**Submitted By :**

**Kaisar Shabir                  (1SI17CS045)  
Kasthuri C M                  (1SI17CS047)**

**Under the guidance of :**

**Mrs. Thejaswini S  
Assistant Professor**

**Mrs. Shwetha A N  
Assistant Professor**



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING  
SIDDAGANGA INSTITUTE OF TECHNOLOGY, TUMKUR-572103  
(An Autonomous Institute, Affiliated to Visvesvaraya Technological University, Belgaum, Recognized  
by AICTE and Accredited by NBA, New Delhi)**

**2019-2020**

# **Table Of Contents**

<b>Chapters</b>	<b>Page. No.</b>
Chapter 1: Introduction	01
Chapter 2: Requirement specification	03
Chapter 3: Constraints	06
Chapter 4: Design	07
Chapter 5: Implementation	09
Chapter 6: Conclusion	43
Chapter 7: References	44

## CHAPTER 1

# INTRODUCTION

We are living in the dynamic world where technology is growing vigorously, and everything has been digitalized. By this, people are generating data, on an average an individual will generate 1.7MB data every second [1]. This generated data has to be stored in proper manner as it has a lot of applications like in machine learning, business, marketing etc., One of the method to store data systematically is using “Databases”. A database is an organized collection of data, generally stored and accessed electronically from a computer system [2].

One such database collected is “Travel Agency Management System”. A travel agency is a private retailer or public service that provides travel and tourism-related services to the general public on behalf of accommodation or travel suppliers [3]. It is necessary for the travel agencies to keep the record of customer travelling details intact. It also helps them to understand customer behavior and provide various benefits to increase their profit margin.

The Travel agency provides facilities for its customers like tourist packages, hotel bookings, reservations, promotions etc., which a customer can buy in a discount. They help the customers to get the overview about the places that can be visited. They also maintain the payment history of the customers and reservation details.

This project is totally built at administrative end and thus only the administrator is guaranteed the access to it. The purpose of the project is to build a database which stores the customer travelling details and thus reduce the manual work for managing the travel agency, customer and their bookings. It tracks all the details about the customers and provides the searching facilities based on various factors.

The editing, adding and updating of records are improved which results in proper resource management of travel agency data. Here the administrator can manage the customers, their bookings, packages taken, means of transport used and their payments. The reservation can be either a cab or a bus or a flight or a hotel for which the payment details will be recorded in payment table. The package is also related to the above entities. It also store the promotion details

---

## 1.1 Motivation

This project gives a complete insight to the travel agency management dataset. It shows how the tables are created for different entities and their attributes, relationship types etc., It give us an idea about how to identify entities and their attributes, and how different entities are related. It also helps us to learn various functions and operations used to store and maintain database.

The advantages are as follows:

1. We will be able to store the records easily without any effort or manual work.
2. We can generate various reports related to the customer travelling details.
3. It facilitates easy search of a particular customer travelling record.
4. It will also provide insight about the places that are frequently visited.
5. It is useful when there is natural disasters in visited places, i.e., it helps to track the details of the tourists who visited to that particular place.
6. It also helps to keep track of immigrants or tourists who entered a particular country without any proper legal documents.

## 1.2 Aim of the Project

This project aims to build a travel agency management dataset in a structured and refined manner. It provides functionalities like editing, updating of tables and also records. It enables to perform various queries on this dataset to obtain the customer travelling history, total profit of the agency, packages offered by the agencies and also helps to keep track of customer reservation. It is used to generate reports and also helps to form promotion of packages. It helps to record customer travelling pattern which in turn helps to recommend packages according to that patterns. It makes the storing of data easier and avoids confusions.

## CHAPTER 2

# REQUIREMENT SPECIFICATION

## 2.1 Database Requirements

### ❖ MySQL

MySQL is an open-source relational database management system (RDBMS). Its name is a combination of "My", the name of co-founder Michael Widenius's daughter, and "SQL", the abbreviation for Structured Query Language.

MySQL is free and open-source software under the terms of the GNU General Public License, and is also available under a variety of proprietary licenses. MySQL was owned and sponsored by the Swedish company MySQL AB, which was bought by Sun Microsystems (now Oracle Corporation). In 2010, when Oracle acquired Sun, Widenius forked the open-source MySQL project to create MariaDB [4].

### ❖ Travel Agencies Database Requirements

- The TRAVEL\_AGENCY are located in various areas and contains id, name, email, phone numbers and address and it maintains various functionalities.
- Each travel agency has various CUSTOMERS with id, name, email, phone number and address and he makes payment, and also reserves.
- Each travel agencies provide various PACKAGEs and are identified by id, location, hotel details, price, transport details, expiry date, price and no. of days.
- It maintains RESERVATION details with id, start date and end date.
- It will have PAYMENT details with price, customer, package details and date paid.
- It must have PROMOTION to promote particular package with id, conditions, price, and discount. It is given with the package.
- It must also have RECOMMENDED\_PACKAGE\_TOUR to recommend specific packages to a specific customers.

## 2.2 Entities and their Attributes

Entities	Attributes
TRAVEL_AGENCY	TA_ID, TA_EMAIL, TA_NAME, TA_NUM, TA_ADDRESS
CUSTOMER	C_ID, C_FNAME, C_LNAME, C_NUM, C_EMAIL, C_ADDRESS
RECOMMENDED_PACKAGE_TOUR	RPK_ID, RPK_LOCATION, RPK_HOTEL, RPK_PRICE, RPK_TRANSPORT, RPK_EXP_DATE, RPK_DAYS
PACKAGE	PCK_ID, PCK_LOCATION, PCK_HOTEL, PCK_PRICE, PCK_TRANSPORT, PCK_EXP_DATE, PCK_DAYS
RESERVATION	R_ID, R_PEOPLE, E_START_DATE, R_END_DATE
PAYMENT	P_ID, P_TYPE, P_DATE, P_PRICE
PROMOTION	PR_ID, PR_CONDITIONS, PR_DETAILS

## 2.1 Relationship Types

Relationship	From Entity	To Entity
RECOMENDED	RECOMMENDED_TOUR_PACKAGE	CUSTOMER
IS_MAINTAINED_BY	TRAVEL_AGENCY	RECOMMENDED_TOUR_PACKAGE
IS_CREATED_BY	CUSTOMER	RESERVATION
IS_VIEWED_BY	PACKAGE	CUSTOMER
CONTAIN	RESERVATION	PACKAGE
MAINTAINED_BY	TRAVEL_AGENCY	PACKAGE
MAKE	CUSTOMER	PAYMENT
VIEWED_BY	CUSTOMER	PROMOTION
MAINTAIN_BY	TRAVEL_AGENCY	PROMOTION
VISITS	CUSTOMER	TRAVEL_AGENCY
DEFINE	RESERVATION	PAYMENT
RESPAID	PAYMENT	RESERVATION
HAS	RECOMMENDED_TOUR_PACKAGE	PROMOTION
AGENCY_CUSTOMER	TRAVEL_AGENCY	CUSTOMER

## CHAPTER 3

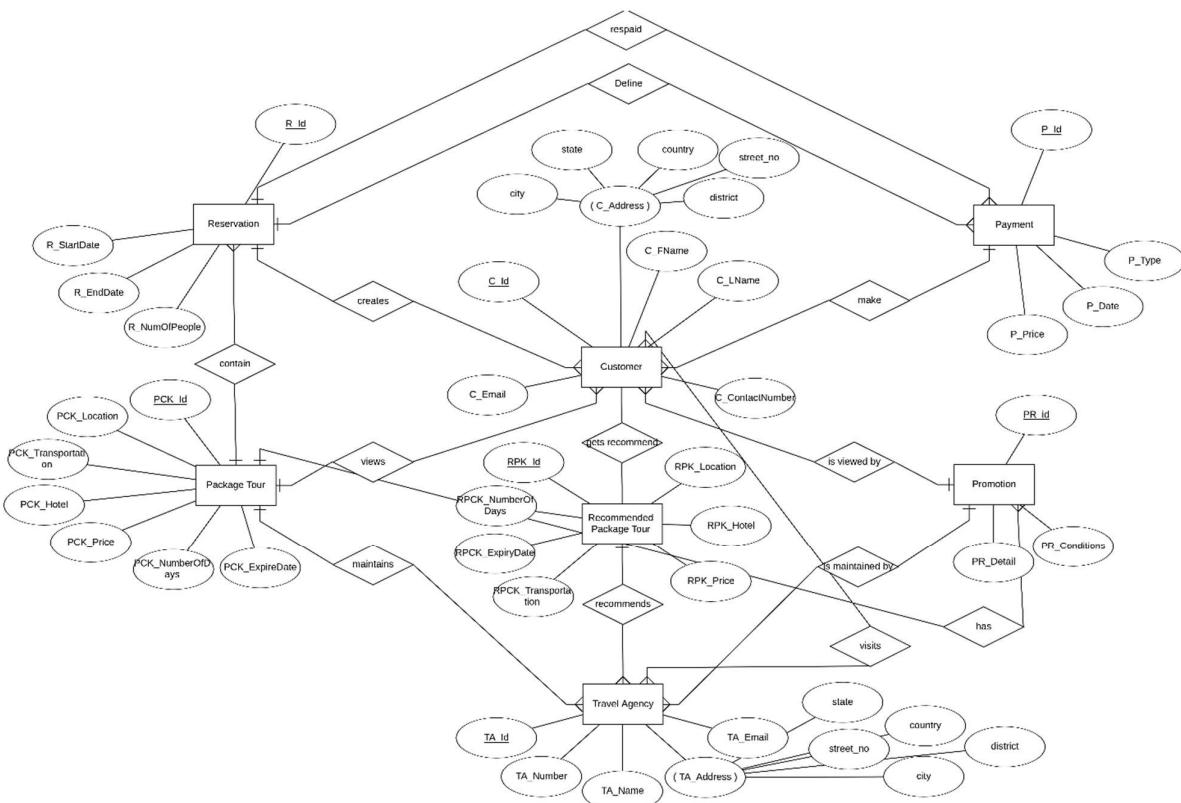
# CONSTRAINTS

- The details of Travel Agency Management System is stored into the Travel Agency tables with each table containing its respective attributes and data.
- Each entity ( TRAVEL\_AGENCY, CUSTOMER, PACKAGE, PAYMENT, PROMOTION, RECOMMENDED\_PACKAGE\_TOUR, RESERVATION ) contains primary key and unique keys.
- The various relationship types are:
  1. One travel agency maintains many packages (1 : n).
  2. One travel agency recommends many packages (1 : n).
  3. One travel agency maintains many promotions (1 : n).
  4. Many customers can visit many travel agencies (m : n).
  5. One customer gets many recommends (1 : n).
  6. One customer views many packages (1 : n).
  7. One customer creates many reservation (1 : n).
  8. One customer makes many payments (1 : n).
  9. One reservation contains one package (1 : 1).
  10. One reservation has one payment (1 : 1).
  11. One payment can define many reservations (1 : n).
  12. Many promotions viewed by one customer (n : 1).
  13. One recommended package has one promotion (1 : 1).
  14. One travel agency has many payments (1 : n).
- The entities contain foreign keys according to the relationship types.
- All the tables have been normalized.

## CHAPTER 4

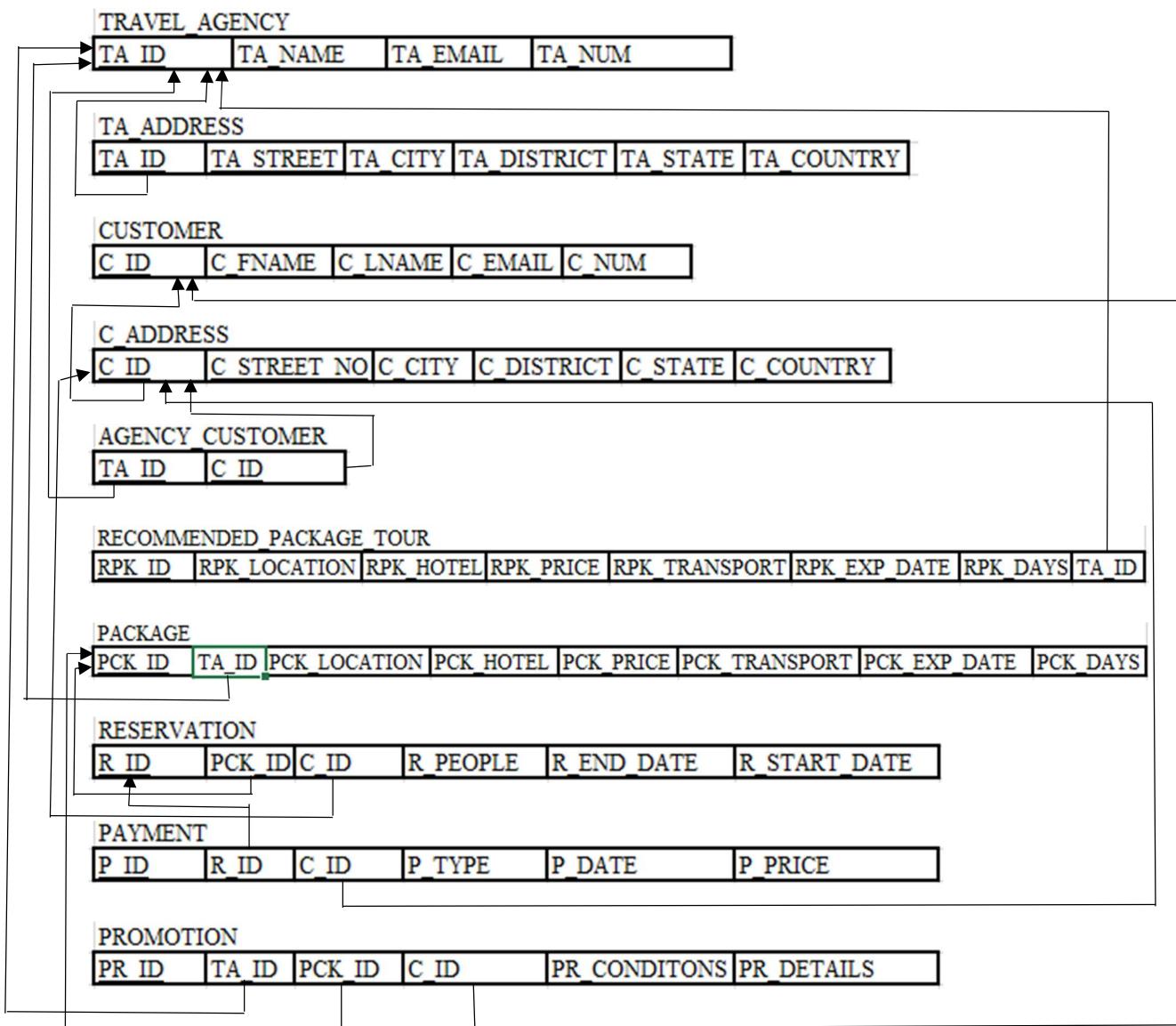
# DESIGN

### 4.1 ER Diagram



[5]

## 4.2 Relational Schema Diagram



## CHAPTER 5

# IMPLEMENTATION

The Travel Agency Management System is a travel agencies database. It consists of entities like TRAVEL\_AGENCY, CUSTOMER, PAYMENT, PACKAGE, RESERVATION, RECOMMENDED\_PACKAGE\_TOUR and PROMOTION.

- TRAVEL\_AGENCY has attributes like TA\_ID, TA\_NAME, TA\_EMAIL, TA\_NUM and TA\_ADDRESS. Each agency offers PACKAGE, PROMOTION and RECOMMENDED\_PACKAGE\_TOUR to CUSTOMERS. It maintains PAYMENT and RESERVATION details. TA\_ID is the primary key.
- CUSTOMERS visits the agency. It has attributes like C\_ID, C\_FNAME, C\_LNAME, C\_NUM, C\_EMAIL and C\_ADDRESS. The customers views the RECOMMENDED\_PACKAGE\_TOUR and select the PACKAGE and make RESERVATION and PAYMENT. C\_ID is the primary key.
- PACKAGE contains all the details like PCK\_ID, PCK\_LOCATION, PCK\_HOTEL, PCK\_TRANSPORT, PCK\_PRICE, PCK\_EXP\_DATE and PCK\_DAYS. The different agencies offers different packages. PCK\_ID is the primary key.
- RECOMMENDED\_TOUR\_PACKAGE is viewed by the customer when he visits the travel agency and selects best suitable package. It contains RPK\_ID, RPK\_LOCATION, RPK\_HOTEL, RPK\_TRANSPORT, RPK\_TRANSPORT, RPK\_EXP\_DATE and RPK\_DAYS. RPK\_ID is the primary key.
- RESERVATION contains R\_ID, R\_START\_DATE AND R\_END\_DATE. The reservation is made by customer based on package selected. It also defines payment details. R\_ID is the primary key.
- PAYMENT is made by the customer based on the price of the package selected. It also contains the reservation details. It has attributes like P\_ID, P\_TYPE, P\_PRICE and P\_DATE. P\_ID is the primary key.
- PROMOTION is the discount given to the customer and also offer made by the agencies. Different agencies gives different promotions. It contains PR\_ID, PR\_CONDITIONS, and PR\_DETAILS. PR\_ID is the primary key.

- Creating database TRAVEL\_AGENCIES in mySQL.

```
mysql> CREATE DATABASE TRAVEL_AGENCIES;
Query OK, 1 row affected (0.10 sec)
```

- Switching to TRAVEL\_AGENCIES database.

```
mysql> USE TRAVEL_AGENCIES;
Database changed
```

## 5.1 Queries

### 5.1.1 Table Creation

#### T1: TRAVEL\_AGENCY

```
CREATE TABLE TRAVEL_AGENCY(
    TA_ID VARCHAR(10) NOT NULL,
    TA_EMAIL VARCHAR(26),
    TA_NAME VARCHAR(26),
    TA_NUM BIGINT(11),
    CONSTRAINT TAGPK1 PRIMARY KEY(TA_ID));
```

```
mysql> CREATE TABLE TRAVEL_AGENCY(
    -> TA_ID VARCHAR(10) NOT NULL,
    -> TA_EMAIL VARCHAR(26),
    -> TA_NAME VARCHAR(26),
    -> TA_NUM BIGINT(11),
    -> CONSTRAINT TAGPK1 PRIMARY KEY(TA_ID)
    -> );
Query OK, 0 rows affected, 1 warning (0.48 sec)
```

```
mysql> DESC TRAVEL_AGENCY;
+-----+-----+-----+-----+-----+
| Field | Type   | Null | Key  | Default | Extra |
+-----+-----+-----+-----+-----+
| TA_ID | varchar(10) | NO   | PRI  | NULL    |       |
| TA_EMAIL | varchar(26) | YES  |      | NULL    |       |
| TA_NAME | varchar(26) | YES  |      | NULL    |       |
| TA_NUM | bigint   | YES  |      | NULL    |       |
+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

## T2: TA\_ADDRESS

```
CREATE TABLE TA_ADDRESS(
    TA_STREET_NO INT NOT NULL,
    TA_ID VARCHAR(10),
    TA_CITY VARCHAR(26),
    TA_DISTRICT VARCHAR(26),
    TA_STATE VARCHAR(26),
    TA_COUNTRY VARCHAR(26),
    CONSTRAINT TADPK1 PRIMARY KEY(TA_STREET_NO, TA_ID),
    CONSTRAINT TADFK1 FOREIGN KEY(TA_ID) REFERENCES
    TRAVEL_AGENCY(TA_ID));
```

```
mysql> CREATE TABLE TA_ADDRESS(
-> TA_STREET_NO INT NOT NULL,
-> TA_ID VARCHAR(10),
-> TA_CITY VARCHAR(26),
-> TA_DISTRICT VARCHAR(26),
-> TA_STATE VARCHAR(26),
-> TA_COUNTRY VARCHAR(26),
-> CONSTRAINT TADPK1 PRIMARY KEY(TA_STREET_NO, TA_ID),
-> CONSTRAINT TADFK1 FOREIGN KEY(TA_ID) REFERENCES TRAVEL_AGENCY(TA_ID)
-> );
Query OK, 0 rows affected (1.00 sec)

mysql> DESC TA_ADDRESS;
+-----+-----+-----+-----+-----+-----+
| Field      | Type       | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| TA_STREET_NO | int        | NO   | PRI | NULL    |       |
| TA_ID        | varchar(10) | NO   | PRI | NULL    |       |
| TA_CITY       | varchar(26) | YES  |     | NULL    |       |
| TA_DISTRICT   | varchar(26) | YES  |     | NULL    |       |
| TA_STATE      | varchar(26) | YES  |     | NULL    |       |
| TA_COUNTRY    | varchar(26) | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.01 sec)
```

### T3: CUSTOMER

```
CREATE TABLE CUSTOMER(
    C_ID VARCHAR(10) NOT NULL,
    C_FNAME VARCHAR(26),
    C_LNAME VARCHAR(26),
    C_NUM BIGINT(11),
    C_EMAIL VARCHAR(32),
    CONSTRAINT CPK1 PRIMARY KEY(C_ID));
```

```
mysql> CREATE TABLE CUSTOMER(
-> C_ID VARCHAR(10) NOT NULL,
-> C_FNAME VARCHAR(26),
-> C_LNAME VARCHAR(26),
-> C_NUM BIGINT(11),
-> C_EMAIL VARCHAR(32),
-> CONSTRAINT CPK1 PRIMARY KEY(C_ID)
-> );
Query OK, 0 rows affected, 1 warning (0.51 sec)
```

```
mysql> DESC CUSTOMER;
+-----+-----+-----+-----+-----+
| Field | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| C_ID  | varchar(10) | NO   | PRI | NULL    |       |
| C_FNAME | varchar(26) | YES  |     | NULL    |       |
| C_LNAME | varchar(26) | YES  |     | NULL    |       |
| C_NUM  | bigint     | YES  |     | NULL    |       |
| C_EMAIL | varchar(32) | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

#### T4: C\_ADDRESS

```
CREATE TABLE C_ADDRESS(
    C_STREET_NO VARCHAR(10) NOT NULL,
    C_CITY VARCHAR(26),
    C_DISTRICT VARCHAR(26),
    C_STATE VARCHAR(26),
    C_COUNTRY VARCHAR(26),
    C_ID VARCHAR(10),
    CONSTRAINT CAPK1 PRIMARY KEY(C_STREET_NO, C_ID),
    CONSTRAINT CAFK1 FOREIGN KEY(C_ID) REFERENCES
    CUSTOMER(C_ID));
```

```
mysql> CREATE TABLE C_ADDRESS(
-> C_STREET_NO VARCHAR(10) NOT NULL,
-> C_CITY VARCHAR(26),
-> C_DISTRICT VARCHAR(26),
-> C_STATE VARCHAR(26),
-> C_COUNTRY VARCHAR(26),
-> C_ID VARCHAR(10),
-> CONSTRAINT CAPK1 PRIMARY KEY(C_STREET_NO, C_ID),
-> CONSTRAINT CAFK1 FOREIGN KEY(C_ID) REFERENCES CUSTOMER(C_ID)
-> );
Query OK, 0 rows affected (1.41 sec)

mysql> DESC C_ADDRESS;
+-----+-----+-----+-----+-----+
| Field      | Type       | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| C_STREET_NO | varchar(10) | NO   | PRI | NULL    |       |
| C_CITY      | varchar(26)  | YES  |     | NULL    |       |
| C_DISTRICT   | varchar(26)  | YES  |     | NULL    |       |
| C_STATE      | varchar(26)  | YES  |     | NULL    |       |
| C_COUNTRY    | varchar(26)  | YES  |     | NULL    |       |
| C_ID         | varchar(10) | NO   | PRI | NULL    |       |
+-----+-----+-----+-----+-----+
6 rows in set (0.01 sec)
```

#### T5: AGENCY\_CUSTOMER

```
CREATE TABLE AGENCY_CUSTOMER(
    TA_ID VARCHAR(10),
    C_ID VARCHAR(10),
    CONSTRAINT ACUK1 UNIQUE(TA_ID, C_ID),
    CONSTRAINT FOREIGN KEY(TA_ID) REFERENCES
    TRAVEL_AGENCY(TA_ID),
    CONSTRAINT ACFK1 FOREIGN KEY(C_ID) REFERENCES
    CUSTOMER(C_ID));
```

```
mysql> CREATE TABLE AGENCY_CUSTOMER(
->     TA_ID VARCHAR(10),
->     C_ID VARCHAR(10),
->     CONSTRAINT ACUK1 UNIQUE(TA_ID, C_ID),
->     CONSTRAINT FOREIGN KEY(TA_ID) REFERENCES TRAVEL_AGENCY(TA_ID),
->     CONSTRAINT ACFK1 FOREIGN KEY(C_ID) REFERENCES CUSTOMER(C_ID)
-> );
Query OK, 0 rows affected (0.73 sec)

mysql> DESC AGENCY_CUSTOMER;
+-----+-----+-----+-----+-----+
| Field | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| TA_ID | varchar(10) | YES  | MUL | NULL    |       |
| C_ID | varchar(10) | YES  | MUL | NULL    |       |
+-----+-----+-----+-----+-----+
2 rows in set (0.01 sec)
```

## T6: RECOMMENDED\_PACKAGE\_TOUR

```
CREATE TABLE RECOMMENDED_PACKAGE_TOUR(
    RPK_ID VARCHAR(10) NOT NULL,
    RPK_LOCATION VARCHAR(26),
    RPK_HOTEL VARCHAR(26),
    RPK_PRICE INT,
    RPK_TRANSPORT VARCHAR(26),
    RPK_EXP_DATE DATE,
    RPK_DAYS INT,
    TA_ID VARCHAR(10),
    CONSTRAINT RTPK1 PRIMARY KEY(RPK_ID),
    CONSTRAINT RPTFK1 FOREIGN KEY(TA_ID) REFERENCES
    TRAVEL_AGENCY(TA_ID);
```

```
mysql> CREATE TABLE RECOMMENDED_PACKAGE_TOUR(
->     RPK_ID VARCHAR(10) NOT NULL,
->     RPK_LOCATION VARCHAR(26),
->     RPK_HOTEL VARCHAR(26),
->     RPK_PRICE INT,
->     RPK_TRANSPORT VARCHAR(26),
->     RPK_EXP_DATE DATE,
->     RPK_DAYS INT,
->     TA_ID VARCHAR(10),
->     CONSTRAINT RTPK1 PRIMARY KEY(RPK_ID),
->     CONSTRAINT RPTFK1 FOREIGN KEY(TA_ID) REFERENCES TRAVEL_AGENCY(TA_ID)
-> );
Query OK, 0 rows affected (0.65 sec)

mysql> DESC RECOMMENDED_PACKAGE_TOUR;
+-----+-----+-----+-----+-----+-----+
| Field      | Type       | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| RPK_ID      | varchar(10) | NO   | PRI | NULL    |       |
| RPK_LOCATION | varchar(26) | YES  |     | NULL    |       |
| RPK_HOTEL    | varchar(26) | YES  |     | NULL    |       |
| RPK_PRICE    | int         | YES  |     | NULL    |       |
| RPK_TRANSPORT | varchar(26) | YES  |     | NULL    |       |
| RPK_EXP_DATE | date        | YES  |     | NULL    |       |
| RPK_DAYS     | int         | YES  |     | NULL    |       |
| TA_ID        | varchar(10) | YES  | MUL | NULL    |       |
+-----+-----+-----+-----+-----+-----+
8 rows in set (0.00 sec)
```

## T7: PACKAGE

```
CREATE TABLE PACKAGE(
    PCK_ID varchar(10) NOT NULL,
    TA_ID varchar(10),
    PCK_LOCATION varchar(30),
    PCK_HOTEL varchar(40),
    PCK_PRICE float(10,2),
    PCK_TRANSPORT varchar(20),
    PCK_EXP_DATE DATE,
    PCK_DAYS int(10),
    constraint PCKPK1 primary key(PCK_ID),
    constraint PCKFK1 foreign key(TA_ID) references TRAVEL_AGENCY(TA_ID)
);
```

```
mysql> CREATE TABLE PACKAGE(
->     PCK_ID varchar(10) NOT NULL,
->     TA_ID varchar(10),
->     PCK_LOCATION varchar(30),
->     PCK_HOTEL varchar(40),
->     PCK_PRICE float(10,2),
->     PCK_TRANSPORT varchar(20),
->     PCK_EXP_DATE DATE,
->     PCK_DAYS int(10),
->     constraint PCKPK1 primary key(PCK_ID),
->     constraint PCKFK1 foreign key(TA_ID) references TRAVEL_AGENCY(TA_ID)
-> );
Query OK, 0 rows affected, 2 warnings (0.59 sec)
```

```
mysql> DESC PACKAGE;
```

Field	Type	Null	Key	Default	Extra
PCK_ID	varchar(10)	NO	PRI	NULL	
TA_ID	varchar(10)	YES	MUL	NULL	
PCK_LOCATION	varchar(30)	YES		NULL	
PCK_HOTEL	varchar(40)	YES		NULL	
PCK_PRICE	float(10,2)	YES		NULL	
PCK_TRANSPORT	varchar(20)	YES		NULL	
PCK_EXP_DATE	date	YES		NULL	
PCK_DAYS	int	YES		NULL	

8 rows in set (0.01 sec)

## T8: RESERVATION

```
CREATE TABLE RESERVATION(
    R_ID varchar(10) NOT NULL,
    PCK_ID varchar(10),
    C_ID varchar(10),
    R_PEOPLE int(60),
    R_END_DATE DATE,
    R_START_DATE DATE,
    constraint RPK1 primary key(R_ID),
    constraint RFK1 foreign key(PCK_ID) references PACKAGE(PCK_ID),
    constraint RFK2 foreign key(C_ID) references CUSTOMER(C_ID));
```

```
mysql> CREATE TABLE RESERVATION(
-> R_ID varchar(10) NOT NULL,
-> PCK_ID varchar(10),
-> C_ID varchar(10),
-> R_PEOPLE int(60),
-> R_END_DATE DATE,
-> R_START_DATE DATE,
-> constraint RPK1 primary key(R_ID),
-> constraint RFK1 foreign key(PCK_ID) references PACKAGE(PCK_ID),
-> constraint RFK2 foreign key(C_ID) references CUSTOMER(C_ID)
-> );
Query OK, 0 rows affected, 1 warning (1.15 sec)

mysql> DESC RESERVATION;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| R_ID | varchar(10) | NO | PRI | NULL | |
| PCK_ID | varchar(10) | YES | MUL | NULL | |
| C_ID | varchar(10) | YES | MUL | NULL | |
| R_PEOPLE | int | YES | | NULL | |
| R_END_DATE | date | YES | | NULL | |
| R_START_DATE | date | YES | | NULL | |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.01 sec)
```

## T9: PAYMENT

```
CREATE TABLE PAYMENT(
    P_ID varchar(10) NOT NULL,
    R_ID varchar(10),
    C_ID varchar(10),
    P_TYPE varchar(10),
    P_DATE DATE,
    P_PRICE float(10,2),
    constraint PPK1 primary key(P_ID),
    constraint PFK1 foreign key(R_ID) references RESERVATION(R_ID),
    constraint PFK2 foreign key(C_ID) references CUSTOMER(C_ID);
```

```
mysql> CREATE TABLE PAYMENT(
-> P_ID varchar(10) NOT NULL,
-> R_ID varchar(10),
-> C_ID varchar(10),
-> P_TYPE varchar(10),
-> P_DATE DATE,
-> P_PRICE float(10,2),
-> constraint PPK1 primary key(P_ID),
-> constraint PFK1 foreign key(R_ID) references RESERVATION(R_ID),
-> constraint PFK2 foreign key(C_ID) references CUSTOMER(C_ID)
-> );
Query OK, 0 rows affected, 1 warning (0.96 sec)

mysql> DESC PAYMENT;
+-----+-----+-----+-----+-----+
| Field | Type   | Null | Key  | Default | Extra |
+-----+-----+-----+-----+-----+
| P_ID  | varchar(10) | NO   | PRI  | NULL    |       |
| R_ID  | varchar(10) | YES  | MUL  | NULL    |       |
| C_ID  | varchar(10) | YES  | MUL  | NULL    |       |
| P_TYPE | varchar(10) | YES  |      | NULL    |       |
| P_DATE | date     | YES  |      | NULL    |       |
| P_PRICE | float(10,2) | YES  |      | NULL    |       |
+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)
```

## T10: PROMOTION

```
CREATE TABLE PROMOTION(
    PR_ID varchar(10) NOT NULL,
    TA_ID varchar(10),
    PCK_ID varchar(10),
    C_ID varchar(10),
    PR_CONDITIONS varchar(60),
    PR_DETAILS varchar(40),
    constraint PRPK1 primary key(PR_ID),
    constraint PRPF1 foreign key(TA_ID) references TRAVEL_AGENCY(TA_ID),
    constraint PRPF2 foreign key(PCK_ID) references PACKAGE(PCK_ID),
    constraint PRPF3 foreign key(C_ID) references CUSTOMER(C_ID));
```

```
mysql> CREATE TABLE PROMOTION(
-> PR_ID varchar(10) NOT NULL,
-> TA_ID varchar(10),
-> PCK_ID varchar(10),
-> C_ID varchar(10),
-> PR_CONDITIONS varchar(60),
-> PR_DETAILS varchar(40),
-> constraint PRPK1 primary key(PR_ID),
-> constraint PRPF1 foreign key(TA_ID) references TRAVEL_AGENCY(TA_ID),
-> constraint PRPF2 foreign key(PCK_ID) references PACKAGE(PCK_ID),
-> constraint PRPF3 foreign key(C_ID) references CUSTOMER(C_ID)
-> );
Query OK, 0 rows affected (1.34 sec)

mysql> DESC PROMOTION;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| PR_ID | varchar(10) | NO | PRI | NULL |       |
| TA_ID | varchar(10) | YES | MUL | NULL |       |
| PCK_ID | varchar(10) | YES | MUL | NULL |       |
| C_ID | varchar(10) | YES | MUL | NULL |       |
| PR_CONDITIONS | varchar(60) | YES |   | NULL |       |
| PR_DETAILS | varchar(40) | YES |   | NULL |       |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.01 sec)
```

## 5.1.2 Inserting values in tuples

### T1: TRAVEL\_AGENCY

```
mysql> INSERT INTO TRAVEL_AGENCY VALUES('1A00000001', 'MAKEMYTRIP@GMAIL.COM', 'MAKEMYTRIP', 9900213122);
Query OK, 1 row affected (0.06 sec)

mysql> INSERT INTO TRAVEL_AGENCY VALUES('1A00000002', 'YATRA@GMAIL.COM', 'YATRA', 8809292921);
Query OK, 1 row affected (0.10 sec)

mysql> INSERT INTO TRAVEL_AGENCY VALUES('1A00000003', 'IXIGO@GMAIL.COM', 'IXIGO', 4304392956);
Query OK, 1 row affected (0.10 sec)

mysql> INSERT INTO TRAVEL_AGENCY VALUES('1A00000004', 'EASYGO@GMAIL.COM', 'EASYGO', 9876553523);
Query OK, 1 row affected (0.07 sec)

mysql> INSERT INTO TRAVEL_AGENCY VALUES('1A00000005', 'GOIBIBO@GMAIL.COM', 'GOIBIBO', 9906752301);
Query OK, 1 row affected (0.08 sec)

mysql> INSERT INTO TRAVEL_AGENCY VALUES('1A00000006', 'CLEARTRIP@GMAIL.COM', 'CLEARTRIP', 9796338668);
Query OK, 1 row affected (0.09 sec)

mysql> INSERT INTO TRAVEL_AGENCY VALUES('1A00000007', 'PAYTM@GMAIL.COM', 'PAYTM', 9768987898);
Query OK, 1 row affected (0.10 sec)

mysql> INSERT INTO TRAVEL_AGENCY VALUES('1A00000008', 'HOLIDAY@GMAIL.COM', 'HOLIDAY', 6767262673);
Query OK, 1 row affected (0.12 sec)

mysql> INSERT INTO TRAVEL_AGENCY VALUES('1A00000009', 'SHEHAR_GHOMO@GMAIL.COM', 'SHEHAR_GHOMO', 7700665352);
Query OK, 1 row affected (0.16 sec)

mysql> INSERT INTO TRAVEL_AGENCY VALUES('1A00000010', 'VACATION@GMAIL.COM', 'VACATION', 8803465465);
Query OK, 1 row affected (0.13 sec)
```

```
mysql> SELECT * FROM TRAVEL_AGENCY;
+-----+-----+-----+-----+
| TA_ID | TA_EMAIL        | TA_NAME    | TA_NUM   |
+-----+-----+-----+-----+
| 1A00000001 | MAKEMYTRIP@GMAIL.COM | MAKEMYTRIP | 9900213122 |
| 1A00000002 | YATRA@GMAIL.COM    | YATRA      | 8809292921 |
| 1A00000003 | IXIGO@GMAIL.COM   | IXIGO      | 4304392956 |
| 1A00000004 | EASYGO@GMAIL.COM  | EASYGO     | 9876553523 |
| 1A00000005 | GOIBIBO@GMAIL.COM | GOIBIBO    | 9906752301 |
| 1A00000006 | CLEARTRIP@GMAIL.COM | CLEARTRIP  | 9796338668 |
| 1A00000007 | PAYTM@GMAIL.COM   | PAYTM      | 9768987898 |
| 1A00000008 | HOLIDAY@GMAIL.COM | HOLIDAY    | 6767262673 |
| 1A00000009 | SHEHAR_GHOMO@GMAIL.COM | SHEHAR_GHOMO | 7700665352 |
| 1A00000010 | VACATION@GMAIL.COM | VACATION   | 8803465465 |
+-----+-----+-----+-----+
10 rows in set (0.01 sec)
```

## T2: TA\_ADDRESS

```
mysql> INSERT INTO TA_ADDRESS VALUES( 23, '1A00000001', 'ANANTNAG', 'ANANTNAG', 'J AND K', 'INDIA');
Query OK, 1 row affected (0.11 sec)

mysql> INSERT INTO TA_ADDRESS VALUES( 253, '1A00000002', 'RANGRETH', 'SRINAGAR', 'J AND K', 'INDIA');
Query OK, 1 row affected (0.12 sec)

mysql> INSERT INTO TA_ADDRESS VALUES( 76, '1A00000003', 'HYDERPORA', 'SRINAGAR', 'J AND K', 'INDIA');
Query OK, 1 row affected (0.08 sec)

mysql> INSERT INTO TA_ADDRESS VALUES( 89, '1A00000004', 'TUMKUR', 'TUMAKURU', 'KARNATAKA', 'INDIA');
Query OK, 1 row affected (0.08 sec)

mysql> INSERT INTO TA_ADDRESS VALUES( 535, '1A00000005', 'BTM LAYOUT', 'BANGALORE', 'KARNATAKA', 'INDIA');
Query OK, 1 row affected (0.08 sec)

mysql> INSERT INTO TA_ADDRESS VALUES( 83, '1A00000006', 'VIDYANAGAR', 'BALLARI', 'KARNATAKA', 'INDIA');
Query OK, 1 row affected (0.10 sec)

mysql> INSERT INTO TA_ADDRESS VALUES( 83, '1A00000007', 'MARATHALLI', 'BANGALORE', 'KARNATAKA', 'INDIA');
Query OK, 1 row affected (0.13 sec)

mysql> INSERT INTO TA_ADDRESS VALUES( 83, '1A00000008', 'GHANDHI NAGAR', 'NEW DELHI', 'DELHI', 'INDIA');
Query OK, 1 row affected (0.12 sec)

mysql> INSERT INTO TA_ADDRESS VALUES( 83, '1A00000009', 'GHANDHI NAGAR', 'NEW DELHI', 'DELHI', 'INDIA');
Query OK, 1 row affected (0.17 sec)

mysql> INSERT INTO TA_ADDRESS VALUES( 813, '1A00000010', 'ANANTNAG', 'ANANTNAG', 'J AND K', 'INDIA');
Query OK, 1 row affected (0.18 sec)
```

```
mysql> SELECT * FROM TA_ADDRESS;
+-----+-----+-----+-----+-----+-----+
| TA_STREET_NO | TA_ID | TA_CITY | TA_DISTRICT | TA_STATE | TA_COUNTRY |
+-----+-----+-----+-----+-----+-----+
| 23 | 1A00000001 | ANANTNAG | ANANTNAG | J AND K | INDIA
| 76 | 1A00000003 | HYDERPORA | SRINAGAR | J AND K | INDIA
| 83 | 1A00000006 | VIDYANAGAR | BALLARI | KARNATAKA | INDIA
| 83 | 1A00000007 | MARATHALLI | BANGALORE | KARNATAKA | INDIA
| 83 | 1A00000008 | GHANDHI NAGAR | NEW DELHI | DELHI | INDIA
| 83 | 1A00000009 | GHANDHI NAGAR | NEW DELHI | DELHI | INDIA
| 89 | 1A00000004 | TUMKUR | TUMAKURU | KARNATAKA | INDIA
| 253 | 1A00000002 | RANGRETH | SRINAGAR | J AND K | INDIA
| 535 | 1A00000005 | BTM LAYOUT | BANGALORE | KARNATAKA | INDIA
| 813 | 1A00000010 | ANANTNAG | ANANTNAG | J AND K | INDIA
+-----+-----+-----+-----+-----+
10 rows in set (0.00 sec)
```

## T3: CUSTOMER

```
mysql> INSERT INTO CUSTOMER VALUES('1C00000001', 'KAISAR', 'DAR', 7006605443, 'KAISARSHABIRDAR@GMAIL.COM');
Query OK, 1 row affected (0.12 sec)

mysql> INSERT INTO CUSTOMER VALUES('1C00000002', 'KASTHURI', 'CM', 7892748768, 'KASTHURIC907@GMAIL.COM');
Query OK, 1 row affected (0.09 sec)

mysql> INSERT INTO CUSTOMER VALUES('1C00000003', 'DAWOOD', 'DAR', 1231231231, 'DAWOOD9055@GMAIL.COM');
Query OK, 1 row affected (0.11 sec)

mysql> INSERT INTO CUSTOMER VALUES('1C00000004', 'AMAAN', '', 9898989898, 'AMAAN@GMAIL.COM');
Query OK, 1 row affected (0.08 sec)

mysql> INSERT INTO CUSTOMER VALUES('1C00000005', 'SAHIL', '', 9696969696, 'SAHIL@GMAIL.COM');
Query OK, 1 row affected (0.29 sec)

mysql> INSERT INTO CUSTOMER VALUES('1C00000006', 'AISHA', 'SHABIR', 9797979797, 'AISHA@GMAIL.COM');
Query OK, 1 row affected (0.17 sec)

mysql> INSERT INTO CUSTOMER VALUES('1C00000007', 'RUFFYDA', 'KHAN', 9999999999, 'RUFFYDA@GMAIL.COM');
Query OK, 1 row affected (0.14 sec)

mysql> INSERT INTO CUSTOMER VALUES('1C00000008', 'FAHEEM', 'KHANDAY', 9595959595, 'FAHEEM@GMAIL.COM');
Query OK, 1 row affected (0.14 sec)

mysql> INSERT INTO CUSTOMER VALUES('1C00000009', 'MAHRUKH', 'KHANDAY', 9494949494, 'MAHRUKH@GMAIL.COM');
Query OK, 1 row affected (0.07 sec)

mysql> INSERT INTO CUSTOMER VALUES('1C00000010', 'DIYA', '', 9393939393, 'DIYA@GMAIL.COM');
Query OK, 1 row affected (0.11 sec)
```

```
mysql> SELECT * FROM CUSTOMER;
+-----+-----+-----+-----+-----+
| C_ID      | C_FNAME | C_LNAME | C_NUM      | C_EMAIL        |
+-----+-----+-----+-----+-----+
| 1C00000001 | KAISAR   | DAR     | 7006605443 | KAISARSHABIRDAR@GMAIL.COM
| 1C00000002 | KASTHURI | CM      | 7892748768 | KASTHURIC907@GMAIL.COM
| 1C00000003 | DAWOOD   | DAR     | 1231231231 | DAWOOD9055@GMAIL.COM
| 1C00000004 | AMAAN    |         | 9898989898 | AMAAN@GMAIL.COM
| 1C00000005 | SAHIL    |         | 9696969696 | SAHIL@GMAIL.COM
| 1C00000006 | AISHA    | SHABIR  | 9797979797 | AISHA@GMAIL.COM
| 1C00000007 | RUFFYDA  | KHAN    | 9999999999 | RUFFYDA@GMAIL.COM
| 1C00000008 | FAHEEM   | KHANDAY | 9595959595 | FAHEEM@GMAIL.COM
| 1C00000009 | MAHRUKH  | KHANDAY | 9494949494 | MAHRUKH@GMAIL.COM
| 1C00000010 | DIYA     |         | 9393939393 | DIYA@GMAIL.COM
+-----+-----+-----+-----+-----+
10 rows in set (0.00 sec)
```

## T4: C\_ADDRESS

```
mysql> INSERT INTO C_ADDRESS VALUES(254, 'ANANTNAG', 'ANANTNAG', 'J AND K', 'INDIA', '1C00000001');
Query OK, 1 row affected (0.10 sec)

mysql> INSERT INTO C_ADDRESS VALUES(254, 'ANANTNAG', 'ANANTNAG', 'J AND K', 'INDIA', '1C00000003');
Query OK, 1 row affected (0.09 sec)

mysql> INSERT INTO C_ADDRESS VALUES(254, 'ANANTNAG', 'ANANTNAG', 'J AND K', 'INDIA', '1C00000006');
Query OK, 1 row affected (0.09 sec)

mysql> INSERT INTO C_ADDRESS VALUES(354, 'BARSOO', 'PULWAMA', 'J AND K', 'INDIA', '1C00000008');
Query OK, 1 row affected (0.11 sec)

mysql> INSERT INTO C_ADDRESS VALUES(397, 'BARSOO', 'PULWAMA', 'J AND K', 'INDIA', '1C00000009');
Query OK, 1 row affected (0.10 sec)

mysql> INSERT INTO C_ADDRESS VALUES(385, 'SANATNAGAR', 'SRINAGAR', 'J AND K', 'INDIA', '1C00000005');
Query OK, 1 row affected (0.09 sec)

mysql> INSERT INTO C_ADDRESS VALUES(676, 'KALKATA', 'WEST BENGAL', 'WEST BENGAL', 'INDIA', '1C00000004');
Query OK, 1 row affected (0.10 sec)

mysql> INSERT INTO C_ADDRESS VALUES(764, 'GHANDHI NAGAR', 'BANGALORE', 'KARNATAKA', 'INDIA', '1C00000002');
Query OK, 1 row affected (0.10 sec)

mysql> INSERT INTO C_ADDRESS VALUES(678, 'UTRASOO', 'ANANTNAG', 'J AND K', 'INDIA', '1C00000007');
Query OK, 1 row affected (0.12 sec)

mysql> INSERT INTO C_ADDRESS VALUES(679, 'GHANDHI NAGAR', 'NEW DELHI', 'DELHI', 'INDIA', '1C00000010');
Query OK, 1 row affected (0.09 sec)
```

```
mysql> SELECT * FROM C_ADDRESS;
+-----+-----+-----+-----+-----+-----+
| C_STREET_NO | C_CITY | C_DISTRICT | C_STATE | C_COUNTRY | C_ID |
+-----+-----+-----+-----+-----+-----+
| 254 | ANANTNAG | ANANTNAG | J AND K | INDIA | 1C00000001 |
| 254 | ANANTNAG | ANANTNAG | J AND K | INDIA | 1C00000003 |
| 254 | ANANTNAG | ANANTNAG | J AND K | INDIA | 1C00000006 |
| 354 | BARSOO | PULWAMA | J AND K | INDIA | 1C00000008 |
| 385 | SANATNAGAR | SRINAGAR | J AND K | INDIA | 1C00000005 |
| 397 | BARSOO | PULWAMA | J AND K | INDIA | 1C00000009 |
| 676 | KALKATA | WEST BENGAL | WEST BENGAL | INDIA | 1C00000004 |
| 678 | UTRASOO | ANANTNAG | J AND K | INDIA | 1C00000007 |
| 679 | GHANDHI NAGAR | NEW DELHI | DELHI | INDIA | 1C00000010 |
| 764 | GHANDHI NAGAR | BANGALORE | KARNATAKA | INDIA | 1C00000002 |
+-----+-----+-----+-----+-----+-----+
10 rows in set (0.00 sec)
```

## T5: AGENCY\_CUSTOMER

```
mysql> INSERT INTO AGENCY_CUSTOMER VALUES('1A00000001', '1C00000001');
Query OK, 1 row affected (0.14 sec)

mysql> INSERT INTO AGENCY_CUSTOMER VALUES('1A00000007', '1C00000002');
Query OK, 1 row affected (0.10 sec)

mysql> INSERT INTO AGENCY_CUSTOMER VALUES('1A00000002', '1C00000003');
Query OK, 1 row affected (0.08 sec)

mysql> INSERT INTO AGENCY_CUSTOMER VALUES('1A00000004', '1C00000004');
Query OK, 1 row affected (0.07 sec)

mysql> INSERT INTO AGENCY_CUSTOMER VALUES('1A00000006', '1C00000005');
Query OK, 1 row affected (0.15 sec)

mysql> INSERT INTO AGENCY_CUSTOMER VALUES('1A00000006', '1C00000006');
Query OK, 1 row affected (0.24 sec)

mysql> INSERT INTO AGENCY_CUSTOMER VALUES('1A00000008', '1C00000007');
Query OK, 1 row affected (0.17 sec)

mysql> INSERT INTO AGENCY_CUSTOMER VALUES('1A00000009', '1C00000008');
Query OK, 1 row affected (0.15 sec)

mysql> INSERT INTO AGENCY_CUSTOMER VALUES('1A00000009', '1C00000009');
Query OK, 1 row affected (0.06 sec)

mysql> INSERT INTO AGENCY_CUSTOMER VALUES('1A00000009', '1C00000010');
Query OK, 1 row affected (0.06 sec)

mysql> INSERT INTO AGENCY_CUSTOMER VALUES('1A00000001', '1C00000010');
Query OK, 1 row affected (0.09 sec)
```

```
mysql> SELECT * FROM AGENCY_CUSTOMER;
+-----+-----+
| TA_ID | C_ID   |
+-----+-----+
| 1A00000001 | 1C00000001 |
| 1A00000001 | 1C00000010 |
| 1A00000002 | 1C00000003 |
| 1A00000004 | 1C00000004 |
| 1A00000006 | 1C00000005 |
| 1A00000006 | 1C00000006 |
| 1A00000007 | 1C00000002 |
| 1A00000008 | 1C00000007 |
| 1A00000009 | 1C00000008 |
| 1A00000009 | 1C00000009 |
| 1A00000009 | 1C00000010 |
+-----+-----+
11 rows in set (0.00 sec)
```

## T6: RECOMMENDED\_PACKAGE\_TOUR

```

mysql> INSERT INTO RECOMMENDED_PACKAGE_TOUR VALUES('1RP0000001', 'GULMARGH', 'VIDHANTHA', '80000', 'ROAD/SUV', '2020-09-20', 7, '1A00000001');
Query OK, 1 row affected (0.09 sec)

mysql> INSERT INTO RECOMMENDED_PACKAGE_TOUR VALUES('1RP0000002', 'DUDH PATHRI', 'FORTUNE', '66000', 'ROAD/SUV', '2020-10-21', 7, '1A00000001');
;
Query OK, 1 row affected (0.11 sec)

mysql> INSERT INTO RECOMMENDED_PACKAGE_TOUR VALUES('1RP0000003', 'HUBLI', 'NAVEEN HUBLI', '66000', 'ROAD/SUV', '2020-10-21', 7, '1A00000006');
Query OK, 1 row affected (0.11 sec)

mysql> INSERT INTO RECOMMENDED_PACKAGE_TOUR VALUES('1RP0000004', 'BALLARY', 'POLA PARADISE', '66000', 'ROAD/SUV', '2020-10-21', 7, '1A00000006');
;
Query OK, 1 row affected (0.15 sec)

mysql> INSERT INTO RECOMMENDED_PACKAGE_TOUR VALUES('1RP0000005', 'GHANDHI NAGAR', 'IBIS NEW DELHI AEROCITY', '110000', 'ROAD/SUV', '2020-10-25',
', 7, '1A00000008');
Query OK, 1 row affected (0.16 sec)

mysql> INSERT INTO RECOMMENDED_PACKAGE_TOUR VALUES('1RP0000006', 'GHANDHI NAGAR', 'IBIS NEW DELHI AEROCITY', '100000', 'ROAD/SUV', '2020-10-25',
', 7, '1A00000009');
Query OK, 1 row affected (0.12 sec)

mysql> INSERT INTO RECOMMENDED_PACKAGE_TOUR VALUES('1RP0000007', 'PAHALGHAM', 'FORTUNE', '150000', 'ROAD/SUV', '2021-01-01', 7, '1A00000008');
Query OK, 1 row affected (0.07 sec)

mysql> INSERT INTO RECOMMENDED_PACKAGE_TOUR VALUES('1RP0000008', 'SHIMLA', 'TAJ', '170000', 'ROAD/SUV', '2021-01-01', 10, '1A00000009');
Query OK, 1 row affected (0.08 sec)

mysql> INSERT INTO RECOMMENDED_PACKAGE_TOUR VALUES('1RP0000009', 'GULMARGH', 'VIDHANTHA', '50000', 'ROAD/SUV', '2020-09-20', 5, '1A00000001');
Query OK, 1 row affected (0.08 sec)

mysql> INSERT INTO RECOMMENDED_PACKAGE_TOUR VALUES('1RP0000010', 'GULMARGH', 'VIDHANTHA', '30000', 'ROAD/SUV', '2020-09-20', 2, '1A00000001');
Query OK, 1 row affected (0.08 sec)

```

```

mysql> SELECT * FROM RECOMMENDED_PACKAGE_TOUR;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| RPK_ID | RPK_LOCATION | RPK_HOTEL | RPK_PRICE | RPK_TRANSPORT | RPK_EXP_DATE | RPK_DAYS | TA_ID |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 1RP0000001 | GULMARGH | VIDHANTHA | 80000 | ROAD/SUV | 2020-09-20 | 7 | 1A00000001 |
| 1RP0000002 | DUDH PATHRI | FORTUNE | 66000 | ROAD/SUV | 2020-10-21 | 7 | 1A00000001 |
| 1RP0000003 | HUBLI | NAVEEN HUBLI | 66000 | ROAD/SUV | 2020-10-21 | 7 | 1A00000006 |
| 1RP0000004 | BALLARY | POLA PARADISE | 66000 | ROAD/SUV | 2020-10-21 | 7 | 1A00000006 |
| 1RP0000005 | GHANDHI NAGAR | IBIS NEW DELHI AEROCITY | 110000 | ROAD/SUV | 2020-10-25 | 7 | 1A00000008 |
| 1RP0000006 | GHANDHI NAGAR | IBIS NEW DELHI AEROCITY | 100000 | ROAD/SUV | 2020-10-25 | 7 | 1A00000009 |
| 1RP0000007 | PAHALGHAM | FORTUNE | 150000 | ROAD/SUV | 2021-01-01 | 7 | 1A00000008 |
| 1RP0000008 | SHIMLA | TAJ | 170000 | ROAD/SUV | 2021-01-01 | 10 | 1A00000009 |
| 1RP0000009 | GULMARGH | VIDHANTHA | 50000 | ROAD/SUV | 2020-09-20 | 5 | 1A00000001 |
| 1RP0000010 | GULMARGH | VIDHANTHA | 30000 | ROAD/SUV | 2020-09-20 | 2 | 1A00000001 |
+-----+-----+-----+-----+-----+-----+-----+-----+
10 rows in set (0.00 sec)

```

# Travel Agency Management System

## T7: PACKAGE

```
mysql> INSERT INTO PACKAGE VALUES('1PK0000001', '1A00000001', 'KARNATAKA', 'HOTEL VIDHANTA', 18998.99, 'ROAD/SUV', '2020-10-19', 3);
Query OK, 1 row affected (0.12 sec)

mysql> INSERT INTO PACKAGE VALUES('1PK0000002', '1A00000002', 'J AND K', 'FORTUNE', 188888, 'ROAD/SUV', '2021-12-01', 9);
Query OK, 1 row affected (0.10 sec)

mysql> INSERT INTO PACKAGE VALUES('1PK0000003', '1A00000002', 'HIMACHAL', 'HOTEL PARADISE', 139999.99, 'ROAD/SUV', '2020-05-03', 10);
Query OK, 1 row affected (0.08 sec)

mysql> INSERT INTO PACKAGE VALUES('1PK0000004', '1A00000003', 'KARNATAKA', 'HOTEL INN', 20000.00, 'ROAD/SUV', '2020-12-12', 4);
Query OK, 1 row affected (0.10 sec)

mysql> INSERT INTO PACKAGE VALUES('1PK0000005', '1A00000004', 'DUBAI', 'HAYATT REGENCY DUBAI CREEK HEIGHTS', 70000, 'ROAD/SUV', '2020-03-09',
3);
Query OK, 1 row affected (0.10 sec)

mysql> INSERT INTO PACKAGE VALUES('1PK0000006', '1A00000006', 'DUBAI', 'RODA AL MUROOJ HOTEL', 80000, 'ROAD/SUV', '2020-03-09', 3);
Query OK, 1 row affected (0.12 sec)

mysql> INSERT INTO PACKAGE VALUES('1PK0000007', '1A00000007', 'DELHI', 'HOTEL TAJ', 70000, 'ROAD/SUV', '2020-09-09', 3);
Query OK, 1 row affected (0.07 sec)

mysql> INSERT INTO PACKAGE VALUES('1PK0000008', '1A00000008', 'J AND K', 'KHYBER RESORT', 90000, 'ROAD/SUV', '2021-01-01', 3);
Query OK, 1 row affected (0.07 sec)

mysql> INSERT INTO PACKAGE VALUES('1PK0000009', '1A00000009', 'J AND K', 'HOTEL KANWAL', 12000, 'ROAD/SUV', '2021-01-01', 2);
Query OK, 1 row affected (0.10 sec)

mysql> INSERT INTO PACKAGE VALUES('1PK0000010', '1A00000005', 'KARNATAKA', 'TAJ HOTEL', 50000, 'ROAD/SUV', '2021-01-01', 3);
Query OK, 1 row affected (0.05 sec)
```

```
mysql> SELECT * FROM PACKAGE;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| PCK_ID | TA_ID | PCK_LOCATION | PCK_HOTEL | PCK_PRICE | PCK_TRANSPORT | PCK_EXP_DATE | PCK_DAYS |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 1PK0000001 | 1A00000001 | KARNATAKA | HOTEL VIDHANTA | 18998.99 | ROAD/SUV | 2020-10-19 | 3 |
| 1PK0000002 | 1A00000002 | J AND K | FORTUNE | 188888.00 | ROAD/SUV | 2021-12-01 | 9 |
| 1PK0000003 | 1A00000002 | HIMACHAL | HOTEL PARADISE | 139999.99 | ROAD/SUV | 2020-05-03 | 10 |
| 1PK0000004 | 1A00000003 | KARNATAKA | HOTEL INN | 20000.00 | ROAD/SUV | 2020-12-12 | 4 |
| 1PK0000005 | 1A00000004 | DUBAI | HAYATT REGENCY DUBAI CREEK HEIGHTS | 70000.00 | ROAD/SUV | 2020-03-09 | 3 |
| 1PK0000006 | 1A00000006 | DUBAI | RODA AL MUROOJ HOTEL | 80000.00 | ROAD/SUV | 2020-03-09 | 3 |
| 1PK0000007 | 1A00000007 | DELHI | HOTEL TAJ | 70000.00 | ROAD/SUV | 2020-09-09 | 3 |
| 1PK0000008 | 1A00000008 | J AND K | KHYBER RESORT | 90000.00 | ROAD/SUV | 2021-01-01 | 3 |
| 1PK0000009 | 1A00000009 | J AND K | HOTEL KANWAL | 12000.00 | ROAD/SUV | 2021-01-01 | 2 |
| 1PK0000010 | 1A00000005 | KARNATAKA | TAJ HOTEL | 50000.00 | ROAD/SUV | 2021-01-01 | 3 |
+-----+-----+-----+-----+-----+-----+-----+-----+
10 rows in set (0.00 sec)
```

## T8: RESERVATION

```
mysql> INSERT INTO RESERVATION VALUES('1R00000001', '1PK0000001', '1C00000001', 3, '2020-12-30', '2020-11-30');
Query OK, 1 row affected (0.13 sec)

mysql> INSERT INTO RESERVATION VALUES('1R00000002', '1PK0000002', '1C00000003', 3, '2020-11-30', '2020-10-30');
Query OK, 1 row affected (0.07 sec)

mysql> INSERT INTO RESERVATION VALUES('1R00000003', '1PK0000003', '1C00000003', 3, '2020-03-30', '2020-04-30');
Query OK, 1 row affected (0.09 sec)

mysql> INSERT INTO RESERVATION VALUES('1R00000004', '1PK0000004', '1C00000004', 3, '2020-11-30', '2020-10-30');
Query OK, 1 row affected (0.07 sec)

mysql> INSERT INTO RESERVATION VALUES('1R00000005', '1PK0000006', '1C00000006', 3, '2020-11-30', '2020-10-30');
Query OK, 1 row affected (0.08 sec)

mysql> INSERT INTO RESERVATION VALUES('1R00000006', '1PK0000008', '1C00000002', 9, '2020-07-21', '2020-07-19');
Query OK, 1 row affected (0.10 sec)

mysql> INSERT INTO RESERVATION VALUES('1R00000007', '1PK0000009', '1C00000008', 2, '2020-08-21', '2020-08-20');
Query OK, 1 row affected (0.10 sec)

mysql> INSERT INTO RESERVATION VALUES('1R00000008', '1PK0000009', '1C00000010', 5, '2020-09-06', '2020-09-04');
Query OK, 1 row affected (0.10 sec)

mysql> INSERT INTO RESERVATION VALUES('1R00000009', '1PK0000006', '1C00000009', 5, '2020-10-14', '2020-10-12');
Query OK, 1 row affected (0.24 sec)

mysql> INSERT INTO RESERVATION VALUES('1R00000010', '1PK0000005', '1C00000002', 1, '2020-12-20', '2020-12-18');
Query OK, 1 row affected (0.08 sec)
```

```
mysql> SELECT * FROM RESERVATION;
+-----+-----+-----+-----+-----+-----+
| R_ID | PCK_ID | C_ID | R_PEOPLE | R_END_DATE | R_START_DATE |
+-----+-----+-----+-----+-----+-----+
| 1R00000001 | 1PK0000001 | 1C00000001 | 3 | 2020-12-30 | 2020-11-30 |
| 1R00000002 | 1PK0000002 | 1C00000003 | 3 | 2020-11-30 | 2020-10-30 |
| 1R00000003 | 1PK0000003 | 1C00000003 | 3 | 2020-03-30 | 2020-04-30 |
| 1R00000004 | 1PK0000004 | 1C00000004 | 3 | 2020-11-30 | 2020-10-30 |
| 1R00000005 | 1PK0000006 | 1C00000006 | 3 | 2020-11-30 | 2020-10-30 |
| 1R00000006 | 1PK0000008 | 1C00000002 | 9 | 2020-07-21 | 2020-07-19 |
| 1R00000007 | 1PK0000009 | 1C00000008 | 2 | 2020-08-21 | 2020-08-20 |
| 1R00000008 | 1PK0000009 | 1C00000010 | 5 | 2020-09-06 | 2020-09-04 |
| 1R00000009 | 1PK0000006 | 1C00000009 | 5 | 2020-10-14 | 2020-10-12 |
| 1R00000010 | 1PK0000005 | 1C00000002 | 1 | 2020-12-20 | 2020-12-18 |
+-----+-----+-----+-----+-----+-----+
10 rows in set (0.00 sec)
```

## T9: PAYMENT

```
mysql> INSERT INTO PAYMENT VALUES('1PY0000001', '1R00000001', '1C00000001','CHEQUE', '2020-03-01', 18000.00);
Query OK, 1 row affected (0.10 sec)

mysql> INSERT INTO PAYMENT VALUES('1PY0000002', '1R00000002', '1C00000003', 'UPI', '2020-03-02', 180000.00);
Query OK, 1 row affected (0.10 sec)

mysql> INSERT INTO PAYMENT VALUES('1PY0000003', '1R00000003', '1C00000003', 'CHEQUE', '2020-02-29', 130000.00);
Query OK, 1 row affected (0.09 sec)

mysql> INSERT INTO PAYMENT VALUES('1PY0000004', '1R00000004', '1C00000004', 'CASH', '2020-02-28', 68000.00);
Query OK, 1 row affected (0.11 sec)

mysql> INSERT INTO PAYMENT VALUES('1PY0000005', '1R00000005', '1C00000006', 'CASH', '2020-01-01', 80000.00);
Query OK, 1 row affected (0.10 sec)

mysql> INSERT INTO PAYMENT VALUES('1PY0000006', '1R00000006', '1C00000002','CHEQUE', '2020-07-15', 270000.00);
Query OK, 1 row affected (0.09 sec)

mysql> INSERT INTO PAYMENT VALUES('1PY0000007', '1R00000007', '1C00000008', 'UPI', '2020-08-02', 12000.00);
Query OK, 1 row affected (0.21 sec)

mysql> INSERT INTO PAYMENT VALUES('1PY0000008', '1R00000008', '1C00000010', 'CHEQUE', '2020-08-29', 20000.00);
Query OK, 1 row affected (0.18 sec)

mysql> INSERT INTO PAYMENT VALUES('1PY0000009', '1R00000009', '1C00000009', 'CASH', '2020-09-28', 180000.00);
Query OK, 1 row affected (0.08 sec)

mysql> INSERT INTO PAYMENT VALUES('1PY0000010', '1R00000010', '1C00000002', 'CASH', '2020-12-01', 70000.00);
Query OK, 1 row affected (0.08 sec)
```

```
mysql> SELECT * FROM PAYMENT;
```

P_ID	R_ID	C_ID	P_TYPE	P_DATE	P_PRICE
1PY0000001	1R00000001	1C00000001	CHEQUE	2020-03-01	18000.00
1PY0000002	1R00000002	1C00000003	UPI	2020-03-02	180000.00
1PY0000003	1R00000003	1C00000003	CHEQUE	2020-02-29	130000.00
1PY0000004	1R00000004	1C00000004	CASH	2020-02-28	68000.00
1PY0000005	1R00000005	1C00000006	CASH	2020-01-01	80000.00
1PY0000006	1R00000006	1C00000002	CHEQUE	2020-07-15	270000.00
1PY0000007	1R00000007	1C00000008	UPI	2020-08-02	12000.00
1PY0000008	1R00000008	1C00000010	CHEQUE	2020-08-29	20000.00
1PY0000009	1R00000009	1C00000009	CASH	2020-09-28	180000.00
1PY0000010	1R00000010	1C00000002	CASH	2020-12-01	70000.00

```
10 rows in set (0.00 sec)
```

## T10: PROMOTION

```
mysql> INSERT INTO PROMOTION VALUES('1PR0000001', '1A00000001', '1PK0000001', '1C00000003', 'valid for next trip with more than 4 members', '30% off');
Query OK, 1 row affected (0.10 sec)

mysql> INSERT INTO PROMOTION VALUES('1PR0000002', '1A00000002', '1PK0000002', '1C00000004', 'valid for 2 members', 'free dessert');
Query OK, 1 row affected (0.09 sec)

mysql> INSERT INTO PROMOTION VALUES('1PR0000003', '1A00000003', '1PK0000003', '1C00000003', 'valid for bill above 500', '50% off');
Query OK, 1 row affected (0.10 sec)

mysql> INSERT INTO PROMOTION VALUES('1PR0000004', '1A00000001', '1PK0000004', '1C00000003', 'valid before december', 'free trip to maldives');
Query OK, 1 row affected (0.32 sec)

mysql> INSERT INTO PROMOTION VALUES('1PR0000005', '1A00000002', '1PK0000005', '1C00000003', 'valid for 4 members above', 'free dinner');
Query OK, 1 row affected (0.09 sec)

mysql> SELECT * FROM PROMOTION;
+----+----+----+----+-----+
| PR_ID | TA_ID | PCK_ID | C_ID | PR_CONDITIONS | PR_DETAILS |
+----+----+----+----+-----+
| 1PR0000001 | 1A00000001 | 1PK0000001 | 1C00000003 | valid for next trip with more than 4 members | 30% off
| 1PR0000002 | 1A00000002 | 1PK0000002 | 1C00000004 | valid for 2 members | free dessert
| 1PR0000003 | 1A00000003 | 1PK0000003 | 1C00000003 | valid for bill above 500 | 50% off
| 1PR0000004 | 1A00000001 | 1PK0000004 | 1C00000003 | valid before december | free trip to maldives
| 1PR0000005 | 1A00000002 | 1PK0000005 | 1C00000003 | valid for 4 members above | free dinner
+----+----+----+----+-----+
5 rows in set (0.00 sec)
```

### 5.1.3 Simple Queries

#### Q1: SELECT OPERATION

SELECT CUSTOMER ID AND FIRST NAME FROM CUSTOMER.

```
mysql> SELECT C_ID,C_FNAME FROM CUSTOMER;
+----+----+
| C_ID | C_FNAME |
+----+----+
| 1C00000001 | KAISAR |
| 1C00000002 | KASTHURI |
| 1C00000003 | DAWOOD |
| 1C00000004 | AMAAN |
| 1C00000005 | SAHIL |
| 1C00000006 | AISHA |
| 1C00000007 | RUFFYDA |
| 1C00000008 | FAHEEM |
| 1C00000009 | MAHRUKH |
| 1C00000010 | DIYA |
+----+----+
10 rows in set (0.00 sec)
```

## Q2: ALTER OPERATION

### Q2.1 ADD CHECK CONSTRAINT TO PAYMENT PRICE, IT SHOULD ALWAYS BE POSITIVE.

```
mysql> ALTER TABLE PAYMENT ADD CHECK(P_PRICE > 0);
Query OK, 10 rows affected (1.49 sec)
Records: 10  Duplicates: 0  Warnings: 0

mysql> INSERT INTO RESERVATION VALUES('1R00000011', '1PK0000005', '1C00000002', 1, '2020-12-20', '2020-12-18');
Query OK, 1 row affected (0.07 sec)

mysql> INSERT INTO PAYMENT VALUES('1PY0000011', '1R00000011', '1C00000002', 'CASH', '2020-12-01', -70000.00);
ERROR 3819 (HY000): Check constraint 'PAYMENT_chk_1' is violated.
```

### Q2.2 ADD NUMBER OF PEOPLE ATTRIBUTE TO PACKAGE TABLE.

```
mysql> ALTER TABLE PACKAGE ADD PCK_PEOPLE INT;
Query OK, 0 rows affected (0.42 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> DESC PACKAGE;
+-----+-----+-----+-----+-----+-----+
| Field      | Type       | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| PCK_ID     | varchar(10) | NO   | PRI | NULL    |       |
| TA_ID      | varchar(10) | YES  | MUL | NULL    |       |
| PCK_LOCATION | varchar(30) | YES  |     | NULL    |       |
| PCK_HOTEL   | varchar(40) | YES  |     | NULL    |       |
| PCK_PRICE   | float(10,2) | YES  |     | NULL    |       |
| PCK_TRANSPORT | varchar(20) | YES  |     | NULL    |       |
| PCK_EXP_DATE | date       | YES  |     | NULL    |       |
| PCK_DAYS    | int        | YES  |     | NULL    |       |
| PCK_PEOPLE   | int        | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
9 rows in set (0.01 sec)
```

## Q3: UPDATE OPERATION

Q3.1 CHANGE PRICE=120000 FOR THE CUSTOMER WITH PAYMENT ID='1PY0000011'.

BEFORE UPDATING:

```
mysql> SELECT * FROM PAYMENT;
+-----+-----+-----+-----+-----+-----+
| P_ID | R_ID | C_ID | P_TYPE | P_DATE | P_PRICE |
+-----+-----+-----+-----+-----+-----+
| 1PY0000001 | 1R00000001 | 1C00000001 | CHEQUE | 2020-03-01 | 18000.00 |
| 1PY0000002 | 1R00000002 | 1C00000003 | UPI | 2020-03-02 | 180000.00 |
| 1PY0000003 | 1R00000003 | 1C00000003 | CHEQUE | 2020-02-29 | 130000.00 |
| 1PY0000004 | 1R00000004 | 1C00000004 | CASH | 2020-02-28 | 68000.00 |
| 1PY0000005 | 1R00000005 | 1C00000006 | CASH | 2020-01-01 | 80000.00 |
| 1PY0000006 | 1R00000006 | 1C00000002 | CHEQUE | 2020-07-15 | 270000.00 |
| 1PY0000007 | 1R00000007 | 1C00000008 | UPI | 2020-08-02 | 12000.00 |
| 1PY0000008 | 1R00000008 | 1C00000010 | CHEQUE | 2020-08-29 | 20000.00 |
| 1PY0000009 | 1R00000009 | 1C00000009 | CASH | 2020-09-28 | 180000.00 |
| 1PY0000010 | 1R00000010 | 1C00000002 | CASH | 2020-12-01 | 70000.00 |
| 1PY0000011 | 1R00000011 | 1C00000002 | CASH | 2020-12-01 | 70000.00 |
+-----+-----+-----+-----+-----+
11 rows in set (0.00 sec)
```

AFTER UPDATING:

```
mysql> UPDATE PAYMENT SET P_PRICE=120000 WHERE P_ID='1PY0000011';
Query OK, 1 row affected (0.12 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> SELECT * FROM PAYMENT;
+-----+-----+-----+-----+-----+-----+
| P_ID | R_ID | C_ID | P_TYPE | P_DATE | P_PRICE |
+-----+-----+-----+-----+-----+-----+
| 1PY0000001 | 1R00000001 | 1C00000001 | CHEQUE | 2020-03-01 | 18000.00 |
| 1PY0000002 | 1R00000002 | 1C00000003 | UPI | 2020-03-02 | 180000.00 |
| 1PY0000003 | 1R00000003 | 1C00000003 | CHEQUE | 2020-02-29 | 130000.00 |
| 1PY0000004 | 1R00000004 | 1C00000004 | CASH | 2020-02-28 | 68000.00 |
| 1PY0000005 | 1R00000005 | 1C00000006 | CASH | 2020-01-01 | 80000.00 |
| 1PY0000006 | 1R00000006 | 1C00000002 | CHEQUE | 2020-07-15 | 270000.00 |
| 1PY0000007 | 1R00000007 | 1C00000008 | UPI | 2020-08-02 | 12000.00 |
| 1PY0000008 | 1R00000008 | 1C00000010 | CHEQUE | 2020-08-29 | 20000.00 |
| 1PY0000009 | 1R00000009 | 1C00000009 | CASH | 2020-09-28 | 180000.00 |
| 1PY0000010 | 1R00000010 | 1C00000002 | CASH | 2020-12-01 | 70000.00 |
| 1PY0000011 | 1R00000011 | 1C00000002 | CASH | 2020-12-01 | 120000.00 |
+-----+-----+-----+-----+-----+
11 rows in set (0.00 sec)
```

Q3.2 CHANGE NUMBER\_OF\_PEOPLE=4 FOR THE CUSTOMER WITH RESERVATION ID='1R00000011'.

```
mysql> SELECT * FROM RESERVATION;
+-----+-----+-----+-----+-----+-----+
| R_ID | PCK_ID | C_ID | R_PEOPLE | R_END_DATE | R_START_DATE |
+-----+-----+-----+-----+-----+-----+
| 1R00000001 | 1PK0000001 | 1C00000001 | 3 | 2020-12-30 | 2020-11-30 |
| 1R00000002 | 1PK0000002 | 1C00000003 | 3 | 2020-11-30 | 2020-10-30 |
| 1R00000003 | 1PK0000003 | 1C00000003 | 3 | 2020-03-30 | 2020-04-30 |
| 1R00000004 | 1PK0000004 | 1C00000004 | 3 | 2020-11-30 | 2020-10-30 |
| 1R00000005 | 1PK0000006 | 1C00000006 | 3 | 2020-11-30 | 2020-10-30 |
| 1R00000006 | 1PK0000008 | 1C00000002 | 9 | 2020-07-21 | 2020-07-19 |
| 1R00000007 | 1PK0000009 | 1C00000008 | 2 | 2020-08-21 | 2020-08-20 |
| 1R00000008 | 1PK0000009 | 1C00000010 | 5 | 2020-09-06 | 2020-09-04 |
| 1R00000009 | 1PK0000006 | 1C00000009 | 5 | 2020-10-14 | 2020-10-12 |
| 1R00000010 | 1PK0000005 | 1C00000002 | 1 | 2020-12-20 | 2020-12-18 |
| 1R00000011 | 1PK0000005 | 1C00000002 | 1 | 2020-12-20 | 2020-12-18 |
+-----+-----+-----+-----+-----+-----+
11 rows in set (0.00 sec)

mysql> UPDATE RESERVATION SET R_PEOPLE=4 WHERE R_ID='1R00000011';
Query OK, 1 row affected (0.28 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> SELECT * FROM RESERVATION;
+-----+-----+-----+-----+-----+-----+
| R_ID | PCK_ID | C_ID | R_PEOPLE | R_END_DATE | R_START_DATE |
+-----+-----+-----+-----+-----+-----+
| 1R00000001 | 1PK0000001 | 1C00000001 | 3 | 2020-12-30 | 2020-11-30 |
| 1R00000002 | 1PK0000002 | 1C00000003 | 3 | 2020-11-30 | 2020-10-30 |
| 1R00000003 | 1PK0000003 | 1C00000003 | 3 | 2020-03-30 | 2020-04-30 |
| 1R00000004 | 1PK0000004 | 1C00000004 | 3 | 2020-11-30 | 2020-10-30 |
| 1R00000005 | 1PK0000006 | 1C00000006 | 3 | 2020-11-30 | 2020-10-30 |
| 1R00000006 | 1PK0000008 | 1C00000002 | 9 | 2020-07-21 | 2020-07-19 |
| 1R00000007 | 1PK0000009 | 1C00000008 | 2 | 2020-08-21 | 2020-08-20 |
| 1R00000008 | 1PK0000009 | 1C00000010 | 5 | 2020-09-06 | 2020-09-04 |
| 1R00000009 | 1PK0000006 | 1C00000009 | 5 | 2020-10-14 | 2020-10-12 |
| 1R00000010 | 1PK0000005 | 1C00000002 | 1 | 2020-12-20 | 2020-12-18 |
| 1R00000011 | 1PK0000005 | 1C00000002 | 4 | 2020-12-20 | 2020-12-18 |
+-----+-----+-----+-----+-----+-----+
11 rows in set (0.00 sec)
```

## Q3: SET OPERATIONS

### Q3.1 UNION OPERATION

LIST ALL THE AGENCIES AND CUSTOMERS WITH THEIR IDS.

```
1 SELECT TA_ID AS ID, TA_NAME AS NAME
2 FROM TRAVEL_AGENCY
3 UNION
4 SELECT C_ID AS ID, C_FNAME AS NAME
5 FROM CUSTOMER|
```

ID	NAME
1A00000001	MAKEMYTRIP
1A00000002	YATRA
1A00000003	IXIGO
1A00000004	EASYGO
1A00000005	GOIBIBO
1A00000006	CLEARTRIP
1A00000007	PAYTM
1A00000008	HOLIDAY
1A00000009	SHEHAR_GHOMO
1A00000010	VACATION
1C00000001	KAISAR
1C00000002	KASTHURI
1C00000003	DAWOOD
1C00000004	AMAAN
1C00000005	SAHIL
1C00000006	AISHA
1C00000007	RUFFYDA
1C00000008	FAHEEM
1C00000009	MAHRUKH
1C00000010	DIYA

[Download CSV](#)

20 rows selected.

### Q3.2 INTERSECTION OPERATION

LIST THE PACKAGES WHICH ATLEAST BOUGHT ONCE BY THE CUSTOMER.

```
1 SELECT PCK_ID AS PACKAGE_ID FROM RESERVATION
2 INTERSECT
3 SELECT PCK_ID AS PACKAGE_ID FROM PACKAGE;|
```

PACKAGE_ID
1PK0000001
1PK0000002
1PK0000003
1PK0000004
1PK0000006

[Download CSV](#)

5 rows selected.

Q3.3 DIFFERENCE OPERATION

LIST ALL THE PACKAGES NOT BOUGHT BY THE CUSTOMERS.

**SQL Worksheet**

```
1 SELECT PCK_ID AS PACKAGE_ID FROM PACKAGE
2 MINUS
3 SELECT PCK_ID AS PACKAGE_ID FROM RESERVATION;
```

**PACKAGE\_ID**

1PK0000005

1PK0000007

1PK0000008

1PK0000009

1PK0000010

[Download CSV](#)

5 rows selected.

Q4: AGGREGATE OPERATIONS

Q4.1 COUNT THE TOTAL NUMBER OF PACKAGES AVAILABLE.

```
mysql> SELECT COUNT(*) AS TOTAL_PACKAGES FROM PACKAGE;
+-----+
| TOTAL_PACKAGES |
+-----+
|          10 |
+-----+
1 row in set (0.05 sec)
```

Q4.2 CALCULATE AVERAGE COST OF ALL THE PACKAGES.

```
mysql> SELECT AVG(PCK_PRICE) AS AVERAGE_PACKAGE_PRICE
    -> FROM PACKAGE;
+-----+
| AVERAGE_PACKAGE_PRICE |
+-----+
|      73988.697461 |
+-----+
1 row in set (0.00 sec)
```

Q4.3 FIND THE COSTLIEST PACKAGE AMONG THE AVAILABLE PACKAGES.

```
mysql> SELECT MAX(PCK_PRICE) AS MAXIMUM_PACKAGE_PRICE  
      -> FROM PACKAGE;  
+-----+  
| MAXIMUM_PACKAGE_PRICE |  
+-----+  
|          188888.00 |  
+-----+  
1 row in set (0.00 sec)
```

Q4.4 FIND THE CHEAPEST PACKAGE AMONG THE AVAILABLE PACKAGES.

```
mysql> SELECT MIN(PCK_PRICE) AS MINIMUM_PACKAGE_PRICE  
      -> FROM PACKAGE;  
+-----+  
| MINIMUM_PACKAGE_PRICE |  
+-----+  
|          12000.00 |  
+-----+  
1 row in set (0.00 sec)
```

Q4.5 FIND THE SUM OF PRICE OF ALL THE AVAILABLE PACKAGES.

```
mysql> SELECT SUM(PCK_PRICE) AS TOTAL_PACKAGE_PRICE  
      -> FROM PACKAGE;  
+-----+  
| TOTAL_PACKAGE_PRICE |  
+-----+  
|          739886.97 |  
+-----+  
1 row in set (0.00 sec)
```

## Q5: REPORTED QUERIES

**Q5.1 PRODUCE A REPORT TO THE CUSTOMER ABOUT THE RESERVATION HE MADE. THE REPORT SHOULD CONTAIN (BILL).**

```
mysql> SELECT R.C_ID,C.C_FNAME,C.C_LNAME,T.TA_NAME,R.R_ID,R.R_PEOPLE,R.R_START_DATE,R.R_END_DATE,P.P_PRICE
-> FROM RESERVATION R,PAYMENT P,CUSTOMER C,AGENCY_CUSTOMER A,TRAVEL_AGENCY T
-> WHERE R.C_ID=C.C_ID AND C.C_ID=A.C_ID AND A.TA_ID=T.TA_ID AND R.R_ID=P.R_ID;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| C_ID | C_FNAME | C_LNAME | TA_NAME | R_ID | R_PEOPLE | R_START_DATE | R_END_DATE | P_PRICE |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1C00000001 | KATSAR | DAR | MAKEMYTRIP | 1R00000001 | 3 | 2020-11-30 | 2020-12-30 | 18000.00 |
| 1C00000003 | DAWOOD | DAR | YATRA | 1R00000002 | 3 | 2020-10-30 | 2020-11-30 | 180000.00 |
| 1C00000003 | DAWOOD | DAR | YATRA | 1R00000003 | 3 | 2020-04-30 | 2020-03-30 | 130000.00 |
| 1C00000004 | AMAAN | EASYGO | EASYGO | 1R00000004 | 3 | 2020-10-30 | 2020-11-30 | 68000.00 |
| 1C00000006 | AISHA | SHABIR | CLEARTRIP | 1R00000005 | 3 | 2020-10-30 | 2020-11-30 | 80000.00 |
| 1C00000002 | KASTHURI | CM | PAYTM | 1R00000006 | 9 | 2020-07-19 | 2020-07-21 | 270000.00 |
| 1C00000008 | FAHEEM | KHNDAY | SHEHAR_GHOMO | 1R00000007 | 2 | 2020-08-20 | 2020-08-21 | 12000.00 |
| 1C00000010 | DIYA | DIYA | SHEHAR_GHOMO | 1R00000008 | 5 | 2020-09-04 | 2020-09-06 | 20000.00 |
| 1C00000010 | DIYA | DIYA | MAKEMYTRIP | 1R00000008 | 5 | 2020-09-04 | 2020-09-06 | 20000.00 |
| 1C00000009 | MAHRUKH | KHNDAY | SHEHAR_GHOMO | 1R00000009 | 5 | 2020-10-12 | 2020-10-14 | 180000.00 |
| 1C00000002 | KASTHURI | CM | PAYTM | 1R00000010 | 1 | 2020-12-18 | 2020-12-20 | 70000.00 |
| 1C00000002 | KASTHURI | CM | PAYTM | 1R00000011 | 4 | 2020-12-18 | 2020-12-20 | 120000.00 |
+-----+-----+-----+-----+-----+-----+-----+-----+
12 rows in set (0.00 sec)
```

**Q5.2 PRODUCE A REPORT CONTAINING CUSTOMER'S ID, TOTAL AMOUNT SPENT BY HIM AND NO. OF BOOKINGS MADE BY HIM.**

```
mysql> SELECT P.C_ID,SUM(P.P_PRICE) AS TOTAL_AMOUNT,COUNT(*) AS NUM_OF_RESERVATION
-> FROM PAYMENT P
-> GROUP BY P.C_ID
-> ORDER BY P.C_ID;
+-----+-----+-----+
| C_ID | TOTAL_AMOUNT | NUM_OF_RESERVATION |
+-----+-----+-----+
| 1C00000001 | 18000.00 | 1 |
| 1C00000002 | 460000.00 | 3 |
| 1C00000003 | 310000.00 | 2 |
| 1C00000004 | 68000.00 | 1 |
| 1C00000006 | 80000.00 | 1 |
| 1C00000008 | 12000.00 | 1 |
| 1C00000009 | 180000.00 | 1 |
| 1C00000010 | 20000.00 | 1 |
+-----+-----+-----+
8 rows in set (0.01 sec)
```

Q5.3 PRODUCE A REPORT CONTAINING TRAVEL AGENCY'S ID, TOTAL PROFIT EARNED AND TOTAL NO. OF CUSTOMERS.

```
mysql> SELECT A.TA_ID,COUNT(*) AS NUM_OF_CUSTOMERS,SUM(P.P_PRICE) AS TOTAL_PROFIT
-> FROM AGENCY_CUSTOMER A,PAYMENT P
-> WHERE A.C_ID=P.C_ID
-> GROUP BY A.TA_ID;
+-----+-----+-----+
| TA_ID | NUM_OF_CUSTOMERS | TOTAL_PROFIT |
+-----+-----+-----+
| 1A00000001 | 2 | 38000.00 |
| 1A00000002 | 2 | 310000.00 |
| 1A00000004 | 1 | 68000.00 |
| 1A00000006 | 1 | 80000.00 |
| 1A00000007 | 3 | 460000.00 |
| 1A00000009 | 3 | 212000.00 |
+-----+-----+-----+
6 rows in set (0.00 sec)
```

Q5.4 LIST THE CUSTOMERS WHO GOT PROMOTIONS AND DISPLAY THE DETAILS.

```
mysql> SELECT DISTINCT R.C_ID,A.TA_ID,P.PCK_ID,P.PR_DETAILS
-> FROM RESERVATION R,AGENCY_CUSTOMER A,PROMOTION P
-> WHERE R.C_ID=P.C_ID AND P.TA_ID=A.TA_ID;
+-----+-----+-----+-----+
| C_ID | TA_ID | PCK_ID | PR_DETAILS |
+-----+-----+-----+-----+
| 1C00000003 | 1A00000001 | 1PK0000001 | 30% off |
| 1C00000004 | 1A00000002 | 1PK0000002 | free dessert |
| 1C00000003 | 1A00000001 | 1PK0000004 | free trip to maldives |
| 1C00000003 | 1A00000002 | 1PK0000005 | free dinner |
+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

#### 5.1.4 Nested Queries

Q1: LIST THE CUSTOMERS OF MAKE TRIP AND PAYTM.

```
mysql> SELECT * FROM CUSTOMER WHERE C_ID NOT IN(
->     SELECT C_ID FROM AGENCY_CUSTOMER WHERE TA_ID IN(
->         SELECT TA_ID FROM TRAVEL_AGENCY WHERE TA_NAME = 'MAKEMYTRIP' OR TA_NAME
->         = 'PAYTM'));
+-----+-----+-----+-----+-----+
| C_ID      | C_FNAME | C_LNAME | C_NUM      | C_EMAIL        |
+-----+-----+-----+-----+-----+
| 1C00000003 | DAWOOD   | DAR      | 1231231231 | DAWOOD9055@GMAIL.COM
| 1C00000004 | AMAAN    |          | 9898989898 | AMAAN@GMAIL.COM
| 1C00000005 | SAHIL    |          | 9696969696 | SAHIL@GMAIL.COM
| 1C00000006 | AISHA    | SHABIR   | 9797979797 | AISHA@GMAIL.COM
| 1C00000007 | RUFFYDA  | KHAN     | 9999999999 | RUFFYDA@GMAIL.COM
| 1C00000008 | FAHEEM   | KHANDAY  | 9595959595 | FAHEEM@GMAIL.COM
| 1C00000009 | MAHRUKH  | KHANDAY  | 9494949494 | MAHRUKH@GMAIL.COM
+-----+-----+-----+-----+-----+
7 rows in set (0.04 sec)
```

Q2: LIST THE CUSTOMERS HAVING ACCOUNT IN DIFFERENT AGENCIES.

```
mysql> SELECT TRAVEL_AGENCY.TA_NAME
->     FROM TRAVEL_AGENCY
->     WHERE TRAVEL_AGENCY.TA_ID IN(
->         SELECT AGENCY_CUSTOMER.TA_ID
->             FROM AGENCY_CUSTOMER
->             WHERE AGENCY_CUSTOMER.C_ID = '1C00000010');
+-----+
| TA_NAME      |
+-----+
| SHEHAR_GHOMO |
| MAKEMYTRIP   |
+-----+
2 rows in set (0.00 sec)
```

## 5.1.5 Triggers

Q1: TRIGGER TO CHECK THE PAYMENT MADE IS NOT NEGATIVE.

### SQL Worksheet

```
1 CREATE OR REPLACE TRIGGER TRIGGER_PAYMENT_CHECK
2   BEFORE INSERT ON PAYMENT FOR EACH ROW
3   DECLARE
4   BEGIN
5     IF(:NEW.P_PRICE < 0) THEN
6       RAISE_APPLICATION_ERROR(-20010, 'YOU CAN NOT PAY NEGATIVE AMOUNT');
7     END IF;
8   END;
9 /
10 SELECT * FROM PAYMENT;
11 INSERT INTO PAYMENT VALUES('1PY0000006', '1R00000003', '1C00000002', 'CASH', DATE '2020-04-04', -6.00);
12 SELECT * FROM PAYMENT;
```

Trigger created.

P_ID	R_ID	C_ID	P_TYPE	P_DATE	P_PRICE
1PY0000001	1R00000001	1C00000001	CHEQUE	01-MAR-20	18000
1PY0000002	1R00000002	1C00000003	UPI	02-MAR-20	180000
1PY0000003	1R00000003	1C00000003	CHEQUE	29-FEB-20	130000
1PY0000004	1R00000004	1C00000004	CASH	28-FEB-20	68000
1PY0000005	1R00000005	1C00000006	CASH	01-JAN-20	80000

[Download CSV](#)

5 rows selected.

ORA-20010: YOU CAN NOT PAY NEGATIVE AMOUNT ORA-06512: at "SQL\_MVR0FRHNDGZVTPVSFK0EQVPUF.TRIGGER\_PAYMENT\_CHECK", line 4  
ORA-06512: at "SYS.DBMS\_SQL", line 1721

P_ID	R_ID	C_ID	P_TYPE	P_DATE	P_PRICE
1PY0000001	1R00000001	1C00000001	CHEQUE	01-MAR-20	18000
1PY0000002	1R00000002	1C00000003	UPI	02-MAR-20	180000
1PY0000003	1R00000003	1C00000003	CHEQUE	29-FEB-20	130000
1PY0000004	1R00000004	1C00000004	CASH	28-FEB-20	68000
1PY0000005	1R00000005	1C00000006	CASH	01-JAN-20	80000

[Download CSV](#)

5 rows selected.

# Travel Agency Management System

Q2: TRIGGER TO MAKE SURE THAT THE TOUR DOES NOT START AFTER THE END DATE.

## SQL Worksheet

```
1 CREATE OR REPLACE TRIGGER RESERVATION_PACKAGE_DATE_CHECK
2   BEFORE INSERT ON RESERVATION FOR EACH ROW
3 DECLARE
4 BEGIN
5   IF(:NEW.R_END_DATE < :NEW.R_START_DATE) THEN
6     RAISE_APPLICATION_ERROR(-20010, 'YOU CAN NOT PAY NEGATIVE AMOUNT');
7   END IF;
8 END;
9 /
10 SELECT * FROM RESERVATION;
11 INSERT INTO RESERVATION VALUES('1R00000006', '1PK0000007', '1C00000005', 3, DATE '2020-09-30', DATE '2020-10-30');
12 SELECT * FROM RESERVATION;
```

Trigger created.

R_ID	PCK_ID	C_ID	R_PEOPLE	R_END_DATE	R_START_DATE
1R00000001	1PK0000001	1C00000001	3	30-DEC-20	30-NOV-20
1R00000002	1PK0000002	1C00000003	3	30-NOV-20	30-OCT-20
1R00000003	1PK0000003	1C00000003	3	30-MAR-20	30-APR-20
1R00000004	1PK0000004	1C00000004	3	30-NOV-20	30-OCT-20
1R00000005	1PK0000006	1C00000006	3	30-NOV-20	30-OCT-20

[Download CSV](#)

5 rows selected.

ORA-20010: YOU CAN NOT PAY NEGATIVE AMOUNT ORA-06512: at "SQL\_MVROFRHNDGZTPVSFKOEQVPUF.RESERVATION\_PACKAGE\_DATE\_CHECK", line 4  
ORA-06512: at "SYS.DBMS\_SQL", line 1721

R_ID	PCK_ID	C_ID	R_PEOPLE	R_END_DATE	R_START_DATE
1R00000001	1PK0000001	1C00000001	3	30-DEC-20	30-NOV-20
1R00000002	1PK0000002	1C00000003	3	30-NOV-20	30-OCT-20
1R00000003	1PK0000003	1C00000003	3	30-MAR-20	30-APR-20
1R00000004	1PK0000004	1C00000004	3	30-NOV-20	30-OCT-20
1R00000005	1PK0000006	1C00000006	3	30-NOV-20	30-OCT-20

[Download CSV](#)

5 rows selected.

### 5.1.6 Stored Procedure

Q1: PROCEDURE TO GET ALL THE AGENCIES A PARTICULAR CUSTOMER IS RELATED TO.

---

#### SQL Worksheet

```
1 CREATE OR REPLACE PROCEDURE PROCEDURE_FIND_AGENCIES_RELATED_TO(CUSTOMER_ID IN VARCHAR) IS
2 CURSOR A1 IS
3 SELECT TRAVEL_AGENCY.TA_NAME
4 FROM TRAVEL_AGENCY
5 WHERE TRAVEL_AGENCY.TA_ID IN(
6     SELECT AGENCY_CUSTOMER.TA_ID
7     FROM AGENCY_CUSTOMER
8     WHERE AGENCY_CUSTOMER.C_ID = CUSTOMER_ID);
9 C1 TRAVEL_AGENCY.TA_NAME%TYPE;
10 BEGIN
11     OPEN A1;
12     FETCH A1 INTO C1;
13     LOOP
14         EXIT WHEN A1%NOTFOUND;
15         DBMS_OUTPUT.PUT_LINE('TRAVEL AGENCY NAME ' || C1);
16         FETCH A1 INTO C1;
17     END LOOP;
18     CLOSE A1;
19 END;
20 /
21 EXEC PROCEDURE_FIND_AGENCIES_RELATED_TO('1C00000010');
```

Procedure created.

Statement processed.

TRAVEL AGENCY NAME MAKEMYTRIP  
TRAVEL AGENCY NAME SHEHAR\_GHOMO

Q2: PROCEDURE TO GET LIST OF CUSTOMERS IN AGENCY.

## SQL Worksheet

```
1 CREATE OR REPLACE PROCEDURE PROCEDURE_FIND_CUSTOMERS_RELATED_TO(AGENCY_ID IN VARCHAR) IS
2 CURSOR A1 IS
3 SELECT CUSTOMER.C_FNAME, CUSTOMER.C_LNAME, CUSTOMER.C_ID
4 FROM CUSTOMER, AGENCY_CUSTOMER
5 WHERE CUSTOMER.C_ID = AGENCY_CUSTOMER.C_ID AND AGENCY_CUSTOMER.TA_ID = AGENCY_ID;
6 C1 CUSTOMER.C_FNAME%TYPE;
7 C2 CUSTOMER.C_LNAME%TYPE;
8 C3 CUSTOMER.C_ID%TYPE;
9 BEGIN
10   OPEN A1;
11   FETCH A1 INTO C1, C2, C3;
12   LOOP
13     EXIT WHEN A1%NOTFOUND;
14     DBMS_OUTPUT.PUT_LINE('NAME: ' || C1 || ' ' || C2 || ', CUSTOMER ID: ' || C3);
15     FETCH A1 INTO C1, C2, C3;
16   END LOOP;
17   CLOSE A1;
18 END;
19 /
20 EXEC PROCEDURE_FIND_CUSTOMERS_RELATED_TO('1A00000009');
```

Procedure created.

Statement processed.

NAME: FAHEEM KHANDAY, CUSTOMER ID: 1C00000008  
NAME: MAHRUKH KHANDAY, CUSTOMER ID: 1C00000009  
NAME: DIYA , CUSTOMER ID: 1C00000010

## CHAPTER 6

# CONCLUSION

In today's world data is generated everywhere and every second. It is a great responsibility to collect this data systematically and store it in an organised manner. So that it can be used properly in the future. One such method of storing the data is by using databases. There are various kinds of database and variety of applications. One simple example is "Travel Agency Management System".

The Travel Agency Management System is used to manage the travel agencies data. It identifies various entities and their attributes, relationship types etc., associated with travel agencies. The entities of travel agency management system are TRAVEL\_AGENCY, CUSTOMER, RECOMMENDED\_PACKAGE\_TOUR, PACKAGE, RESERVATION, PAYMENT and PROMOTION. It contains relationships like maintains by, visits, defines, contains, views etc.,

It is created in order to make storing of data hustle free and easier. It eliminates the traditional file storing mechanism and also manual entering of data. It also enables us to perform various queries and retrieve the required information. It also generates report which is like a summary for the customer and also travel agency. It helps us to calculate overall profit of the agencies. It helps us to identify the customer travelling pattern and by this we can recommend the packages of the customer choice. We can also give discounts based on that. It has various added on benefits.

The project can further be extended to be incorporate into web applications or mobile applications where we can create forms which enables data entry and retrieval more easy. It can be used to track the customer or tourist details efficiently, by this we can search for a person if he is missing. It can also be used in machine learning and extract various details which can be used by the travel agency to provide offers which brings more profit along with customer satisfaction.

Travel Agency Management System is the database management system which stores data systematically and makes the retrieval of information easier and comfortable.

## CHAPTER 7

# REFERENCES

- [1] “How Much Data Is Generated Every Minute? [Infographic],” [Online]. Available: <https://www.socialmediatoday.com/news/how-much-data-is-generated-every-minute-infographic-1/525692/>.
- [2] “Database - Wikipedia,” [Online]. Available: <https://en.wikipedia.org/wiki/Database>.
- [3] “Travel agency - Wikipedia,” [Online]. Available: [https://en.wikipedia.org/wiki/Travel\\_agency](https://en.wikipedia.org/wiki/Travel_agency).
- [4] “MySQL - Wikipedia,” [Online]. Available: <https://en.wikipedia.org/wiki/MySQL>.
- [5] “ERD PLUS,” [Online]. Available: <https://erdplus.com/>.