KENDRIYA VIDYALAYA NO.2 ITANAGAR (A.P)

Computer science project report

SUBMITTED BY:

Lungsom Lamnio

Class: XII' A

CBSE Roll no: 1665839

SUMITTED TO:

Saroj K. Mishra

(subject teacher)

CERTIFICATE

This is to certify that **LUNGSOM LAMNIO** of class: **XII A** of **KENDRIYA VIDYALAYA NO.2** has done his project on **FLIGHT MANAGEMENT SYSTEM** under my supervision. He has taken interest and has shown at most sincerity in completion of this project.

I certify this project up to my expectation & as per guidelines issued by **CBSE, NEW DELHI.**

Internal Examiner   *ExternalExaminer*

principal

ACKNOWLEDGMENT

It is with pleasure that I acknowledge my sincere gratitude to our teacher, ***MR. S.K. MISRA*** who taught and undertook the responsibility of teaching the subject computer science. I have been greatly benefited from his classes.

I am especially indebted to our Principal ***MR. K R MEENA*** who has always been a source of encouragement and support and without whose inspiration this project would not have been a successful I would like to place on record heartfelt thanks to him.

Finally, I would like to express my sincere appreciation for all the other students for my batch their friendship &

the fine time that we all shared together

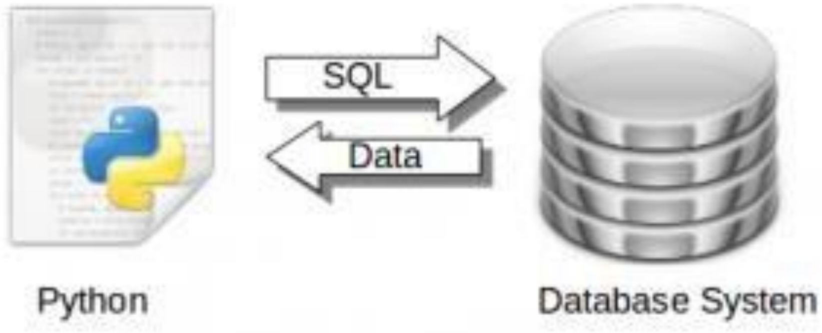
HARDWARE AND SOFTWARE REQUIRED

HARDWARE

1. PC
2. MOBILE PHONE

SOFTWARE

- PYTHON (latest version)
- MYSQL
- PYTHON CONNECTOR



FLIGHT

The following are the main advantages:

- High Speed: It is the fast speed means of transport. ...
- Minimum Cost: ...
- Strategic Importance: ...
- Easy transport of costly and light goods: ...
- Free from physical barriers: ...
- Useful for Agriculture: ...
- Useful in natural calamities:

FLIGHT MANAGEMENT SYSTEM

A **flight management system** (**FMS**) is a fundamental component of a modern airliner's avionics. An FMS is a specialized computer system that automates a wide variety of in-flight tasks, reducing the workload on the flight crew to the point that modern civilian aircraft no longer carry flight engineers or navigators. A primary function is in-flight management of the flight plan. Using various sensors (such as GPS and INS often backed up by radio navigation) to determine the aircraft's position, the FMS can guide the aircraft along the flight plan. From the cockpit, the FMS is normally controlled through a Control Display Unit (CDU) which incorporates a small screen and keyboard or touchscreen. The FMS sends the flight plan for display to the Electronic Flight Instrument System (EFIS), Navigation Display (ND), or Multifunction Display (MFD). The FMS can be summarised as being a dual system consisting of the **Flight Management Computer (FMC)**, CDU and a cross talk bus.

The modern FMS was introduced on the Boeing 767, though earlier navigation computers did exist.[1] Now, systems similar to FMS exist on aircraft as small as the Cessna 182. In its evolution an FMS has had many different sizes, capabilities and controls. However certain characteristics are common to all FMSs.

NEED OF FMS

1. Minimized documentation and no duplication of records.
2. Reduced paper work.
3. Improved patient care
4. Better Administration Control
5. Faster information flow between various departments
6. Smart Revenue Management
7. Effective billing of various services
8. Exact stock information

#CODE:

```
                    #FLIGHT MANAGEMENT SYSTEM
                        #USING PYTHON+MYSQL
          #CREATED BY LUNGSOM LAMNIO, LAKHI MILI AND ANKIT DAS
                   #SUBMITTED TO SAROJ K. MISHRA

import mysql.connector                  #_______mysql connector package



obj=mysql.connector.connect(host="localhost",user="root",passwd="admin")
#here obj is connection object




#CREATING DATABASE & TABLE

mycursor=obj.cursor()

#________cursor is used to row by row processing of record in the
resultset


mycursor.execute("create database if not exists airlines")
                                   #______mycursor is cursor object


#CREATING DATABASE
Mycursor.execute(“create database if not exists airlines”)
mycursor.execute("use airlines")


#CREATING TABLE FOR ODER FOOD
mycursor.execute("create table if not exists food_items(sl_no int(4)
auto_increment primary key,food_name varchar(40)not null,price int(4) not
null)")
mycursor.execute("insert into food_items
values({},'{}',{})".format('null','pepsi',150))
mycursor.execute("insert into food_items
values({},'{}',{})".format('null','coffee',70))
mycursor.execute("insert into food_items
values({},'{}',{})".format('null','tea',50))
mycursor.execute("insert into food_items
values({},'{}',{})".format('null','water',60))
mycursor.execute("insert into food_items
values({},'{}',{})".format('null','milk shake',80))
mycursor.execute("insert into food_items
values({},'{}',{})".format('null','chicken burger',160))
```

```
mycursor.execute("insert into food_items
values({},'{}',{})".format('null','cheese pizza',70))
mycursor.execute("insert into food_items
values({},'{}',{})".format('null','chicken biryani',300))
mycursor.execute("insert into food_items
values({},'{}',{})".format('null','plane rice',80))
mycursor.execute("insert into food_items
values({},'{}',{})".format('null','aloo paratha',120))
mycursor.execute("insert into food_items
values({},'{}',{})".format('null','roti sabji',100))
mycursor.execute("insert into food_items
values({},'{}',{})".format('null','omelette',50))



#CREATING TABLE FOR LUGGAGE ENTRY
mycursor.execute("create table if not exists luggage(luggage_id int(4)
auto_increment primary key,weight int(3)not null,price int(4) not null)")



#CREATING TABLE FOR CUSTOMER DETAILS
mycursor.execute("create table if not exists cust_details(cust_id int(4)
auto_increment primary key,cust_name varchar(40)not null,cont_no
bigint(10) not null)")



#CREATING TABLE FOR CUSTOMER'S FLIGHT DETAILS
mycursor.execute("create table if not exists flight_details(cus_id
int(4),cus_name varchar(40)not null,flight_id)")


obj.commit()




#TO ENTER THE DETAILS OF LUGGAGE
def luggage():
    print("what do you want to do?")
    print("1. add luggage")
    print("2. delete luggage")
    x=int(input("enter your choice: "))
    if x==1:
        lname=input("enter luggage type: ")
        mycursor.execute("insert into luggage
values({},'{}'".format('null',lname))
    elif x==2:
        lid=int(input("enter your luggage id: "))
        mycursor.execute("delete from luggage where
luggage_id={}".format(lid))

    else:
        print(" **************** PLEASE ENTER A VALID OPTION
**************************** ")
        food()
    obj.commit()
```

```
#TO UPDATE THE INFORMATION OF FOOD DETAILS
def food():
    print("what do you want to do?")
    print("1. add new items")
    print("2. update price")
    print("3. delete items")
    x=int(input("enter your choice: "))
          if x==1:
              fname=input("enter food name: ")
              fprice=int(input("enter food price: "))
              mycursor.execute("insert into food_items
values({},'{}',{}".format('null',fname,fprice))
          elif x==2:
              fid=int(input("enter food id: "))
              fprice=int(input("enter new price: "))
              mycursor.execute("update food_items set price={} where
food_id={}".format(fid,fprice))
          elif x==3:
              fid=int(input("enter food id: "))
              mycursor.execute("delete from food_items where where
food_id={}".format(fid))
          else:
              print(" **************** PLEASE ENTER A VALID OPTION
**************************** ")
              food()
          obj.commit()




#TO UPDATE THE INFORMATION OF CLASSTYPE
def classtype():
    print("what do you wat to do? ")
    print("1. change the classtype name")
    print("2. change the price of classtype")
    x=int(input("enter your choice: "))
    if x==1:
        oname=input("enter old name: ")
        nname=input("enter new name: ")
        mycursor.execute("update classtype set
'{}'='{}'".format(oname,nname))


def fooditems():
    print(" ")
    print(" ")

    print("            THE AVAILABLE FOODS ARE:              ")
    print(" ")
    print(" ")
    mycursor.execute("select * from food_items")
    x=mycursor.fetchall()
    for i in x:
        print(" FOOD ID: ",i[0])
        print(" FOOD Name: ",i[1])
        print(" PRICE: ",i[2])

print("_________________________________________________________________
________________________")

    user()
```

```
#Admin Interface after verifying password
def admin1():
    print("************ WHAT'S YOUR TODAYS GOAL?  ****************")
    print("1. update details")
    print("2. show details")
    print("3. job approval")
    x=int(input("select your choice: "))
    while True:
        if x==1:
            print("1. classtype")
            print("2. food")
            print("3. luggage")
            x1=int(input("enter your choice"))
            if x1==1:
                classtype()
            elif x1==2:
                food()
            elif x1==3:
                luggage()
            else:
                print(" ********************* PLEASE ENTER A VALID
OPTION ******************************** ")
                admin1()


        elif x==2:
            print("1. classtype")
            print("2. food")
            print("3. luggage")
            print("4. records")
            y=int(input("from which table: "))
            if y==1:
                mycursor.execute("select * from classtype")
            else:
                mycursor.execute("select * from customer_details")
                z=mycursor.fetchall()
                for i in z:
                    print(i)
                print("*************** THESE ABOVE PEOPLE HAVE BOOKED
TICKET ****************************")
        break
```

```
#Admin Interface
def admin():
     while True:
            sec=input("enter the password: ")
            if sec=="admin":
                admin1()
            else:
                print("************YOUR PASSWORD IS INCORRECT*********")
                print("***********PLEASE TRY AGAIN*****************")
                admin()
            break
```

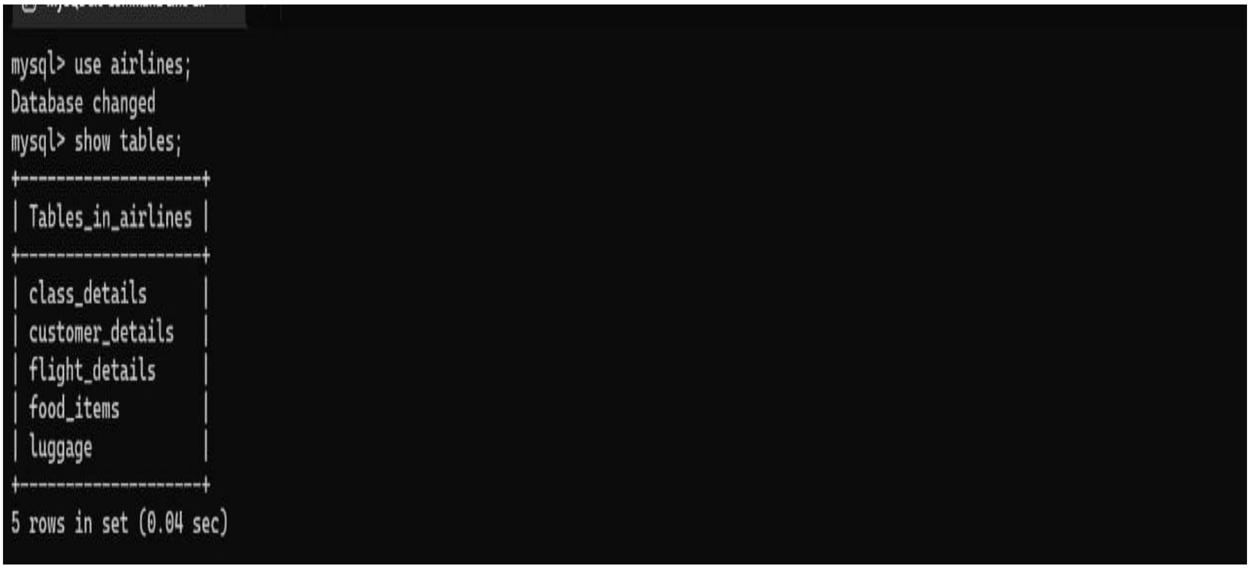
```
#TO SEE THE RECORDS OF THE CUSTOMER
def record():
    cid=int(input("enter your customer id: "))
    mycursor.execute(" select * from customer_details where
cus_id={}".format(cid))
    d=mycursor.fetchall()

    print("YOUR DETAILS ARE HERE...........")
    print("customer id: ",d[0])
    print("name: ",d[1])
    print("mobile number: ",d[2])
    print("flight id: ",d[3])
    print("flight name",d[4])
    print("classtype: ",d[5])
    print("departure place",d[6])
    print("destination",d[7])
    print("flight day: ",d[8])
    print("flight time: ",d[9])
    print("price of ticket: ",d[10])
    print("date of booking ticket: ",d[11])
```

```
#TO BOOK THE TICKETS
def ticketbooking():
    cname=input("enter your name: ")
    cmob=int(input("enter your mobileno: "))
    if cmob==0000000000:
        print(" MOBILE NUMBER CANT BE NULL ")
        ticketbooking()
    fid=int(input("enter flight id: "))
    fcl=input("enter your class: ")
    fname=input("enter your flight name")
    dept=input("enter departure place: ")
    dest=input("enter destination: ")
    fday=input("enter flight day: ")
    ftime=input("enter flight time: ")
    fprice=input("enter ticket rate: ")
    mycursor.execute("insert into customer_details
values({},'{}',{},{},'{}','{}','{}','{}','{}','{}',{}".format('null',cnam
e,cmob,fid,fname,fcl,dept,dest,fday,ftime,"curdate()"))

    obj.commit()




#TO SEE THE AVAILABLE FLIGHTS
def flightavailable():
    print(" ")
    print(" ")

    print("            THE AVAILABLE FLIGHTS ARE:              ")
    print(" ")
    print(" ")
    mycursor.execute("select * from flight_details")
    x=mycursor.fetchall()
    for i in x:
        print(" ")
        print(" Flight ID: ",i[0])
        print(" Flight Name: ",i[1])
        print(" departure: ",i[2])
        print(" Destination: ",i[3])
        print(" Take off Day: ",i[4])
        print(" Take off time : ",i[5])
        print(" bussiness : ",i[6])
        print(" middle : ",i[7])
        print(" economic : ",i[8])

print("__________________________________________________________________
________________________")

    user()




#USER INTERFACE
def user():
    while True:
        print("************** MAY I HELP YOU? *****************")
        print("1. flight details")
        print("2. food details")
        print("3. book ticket")
        print("4. my details")
        print("5. exit")
        x=int(input("enter your choice: "))
        if x==1:
            flightavailable()
        elif x==2:
            fooditems()
        elif x==3:
            ticketbooking()
        elif x==4:
            records()
        else:
            print("************ PLEASE CHOOSE A CORRECT OPTION
************")
            user()
        break




print("****************** WELCOME TO LAMNIO AIRLINES
**********************")
print("************  MAKE YOUR JOURNEY SUCCESS WITH US!
*****************")
print(" ")
print(" ")
print("  ")
print(" ")
print(" ")
```

```
#Main Interface
def menu1():
    print("****************    YOUR DESIGNATION?    *******************")
    print("1. admin")
    print("2. user")
    print("3. exit")
    x=int(input("choose a option: "))
    while True:
        if x==1:
            admin()
        elif x==2:
            user()
        else:
            print("************PLEASE CHOOSE A CORRECT
OPTION******************")
            menu1()
        break




menu1()
```

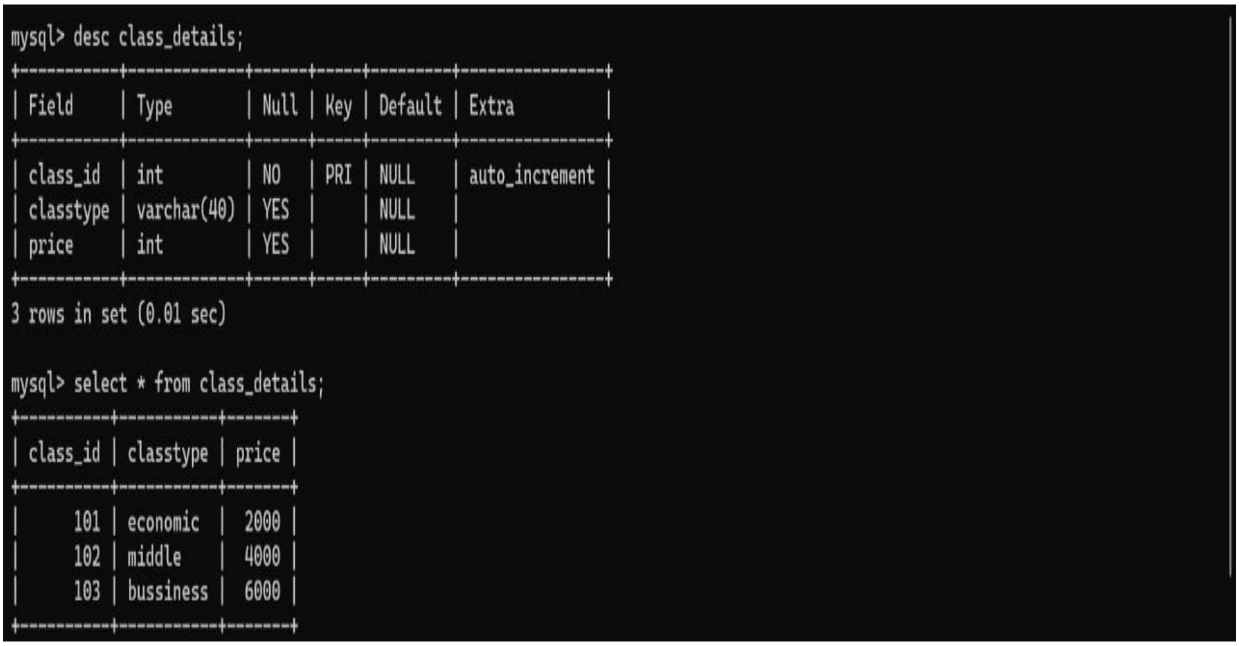
#mysql

- **Tables in database airlines:**



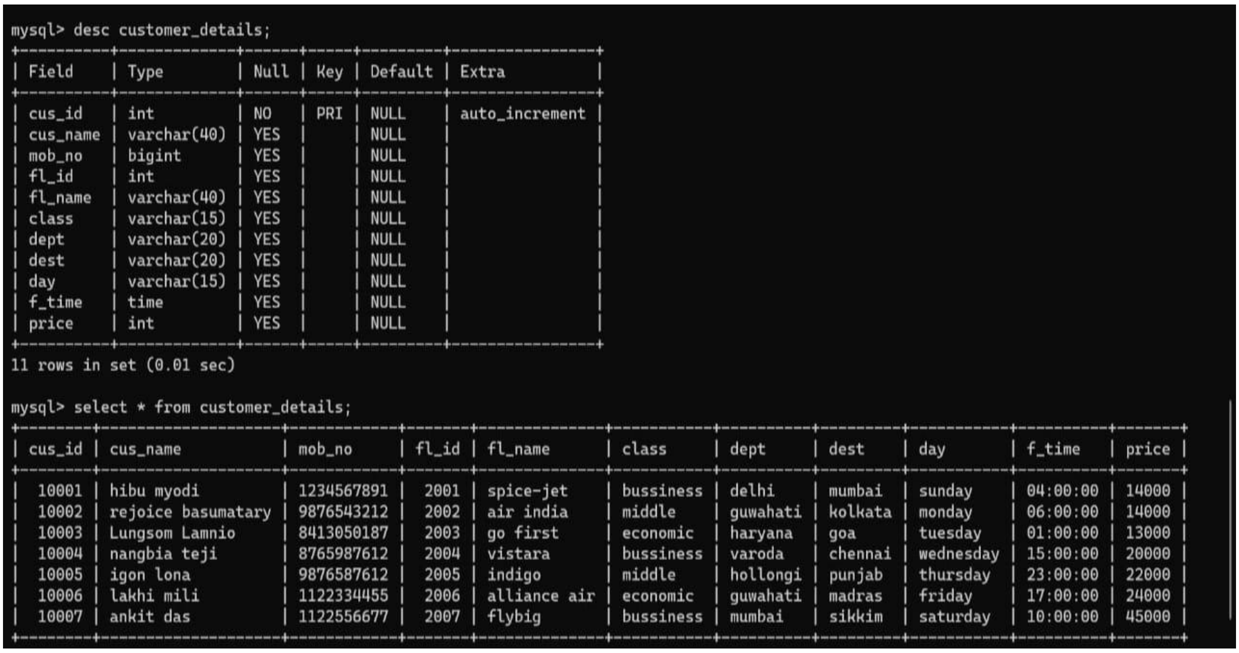
```
mysql> use airlines;
Database changed
mysql> show tables;
+--------------------+
| Tables_in_airlines |
+--------------------+
| class_details      |
| customer_details   |
| flight_details     |
| food_items         |
| luggage            |
+--------------------+
5 rows in set (0.04 sec)
```

- **Description and datas in table class_details:**



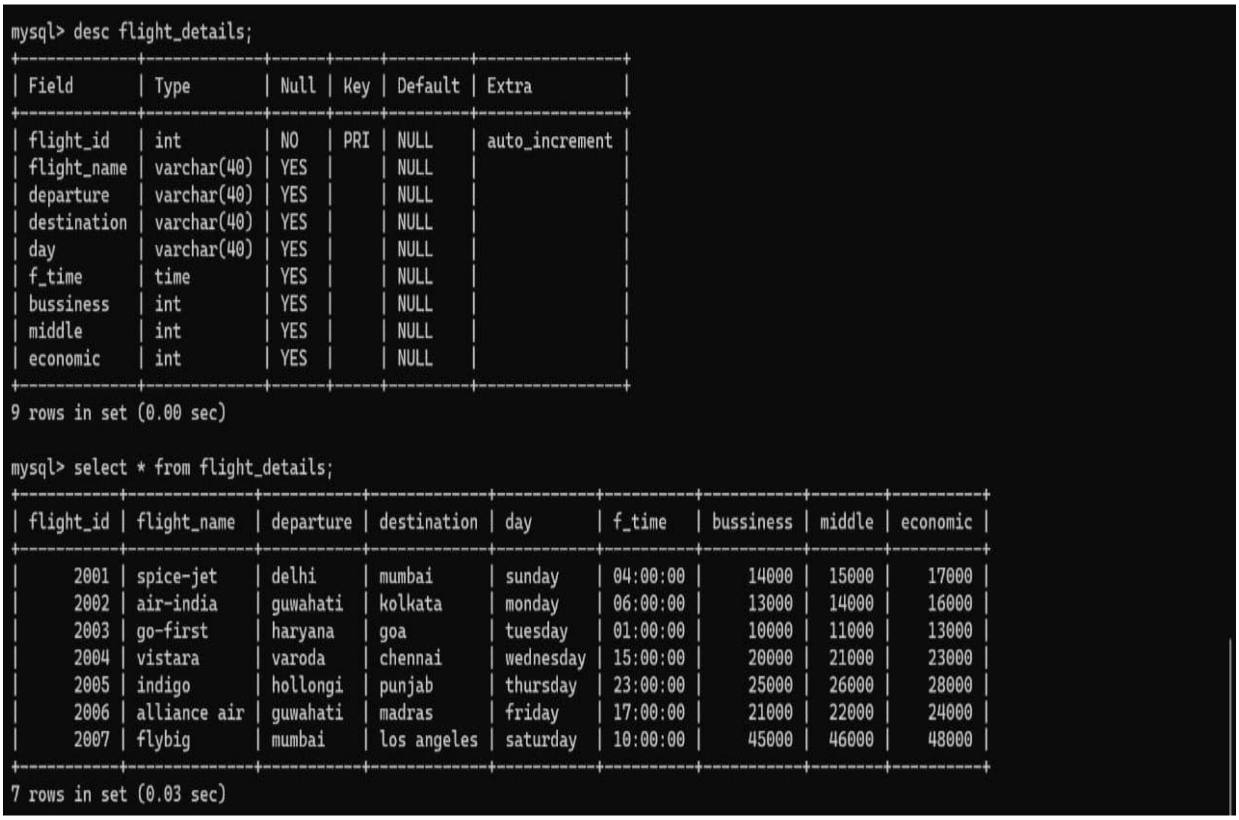
```
mysql> desc class_details;
+-----------+-------------+------+-----+---------+----------------+
| Field     | Type        | Null | Key | Default | Extra          |
+-----------+-------------+------+-----+---------+----------------+
| class_id  | int         | NO   | PRI | NULL    | auto_increment |
| classtype | varchar(40) | YES  |     | NULL    |                |
| price     | int         | YES  |     | NULL    |                |
+-----------+-------------+------+-----+---------+----------------+
3 rows in set (0.01 sec)

mysql> select * from class_details;
+----------+-----------+-------+
| class_id | classtype | price |
+----------+-----------+-------+
|      101 | economic  |  2000 |
|      102 | middle    |  4000 |
|      103 | bussiness |  6000 |
+----------+-----------+-------+
```

- **Description and datas in table customer_details:**



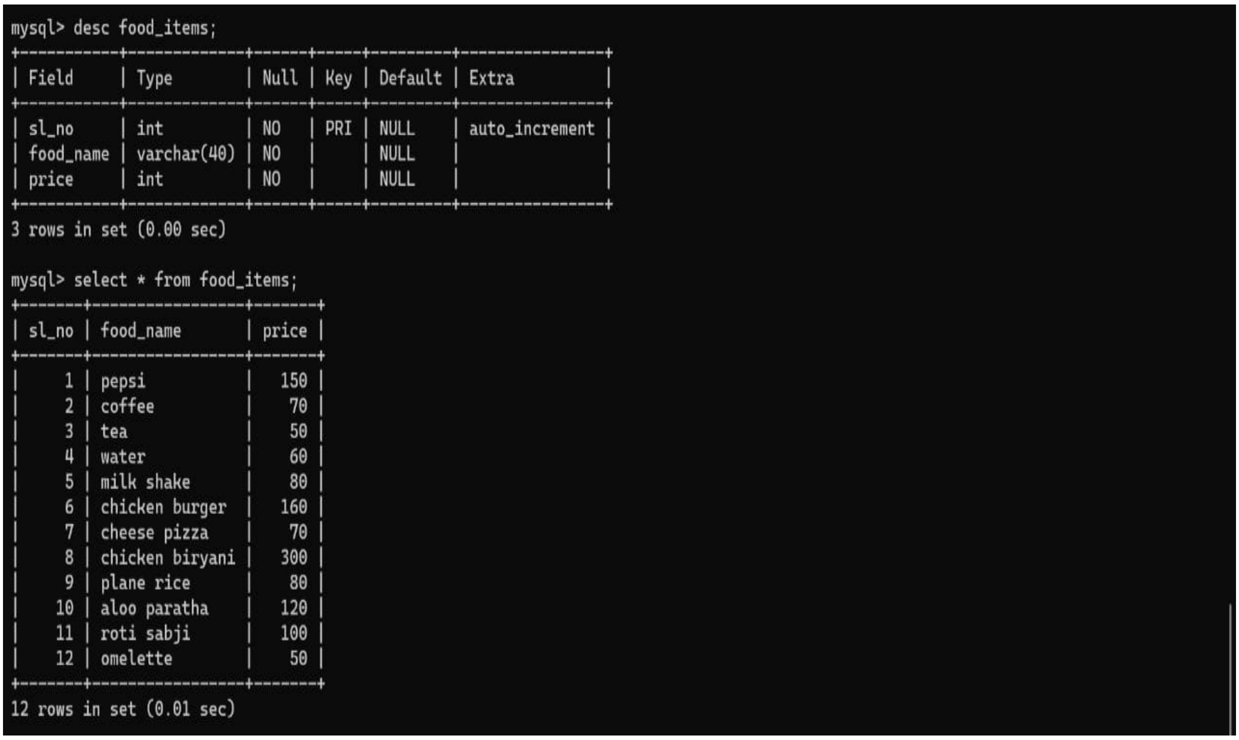
```
mysql> desc customer_details;
+----------+-------------+------+-----+---------+----------------+
| Field    | Type        | Null | Key | Default | Extra          |
+----------+-------------+------+-----+---------+----------------+
| cus_id   | int         | NO   | PRI | NULL    | auto_increment |
| cus_name | varchar(40) | YES  |     | NULL    |                |
| mob_no   | bigint      | YES  |     | NULL    |                |
| fl_id    | int         | YES  |     | NULL    |                |
| fl_name  | varchar(40) | YES  |     | NULL    |                |
| class    | varchar(15) | YES  |     | NULL    |                |
| dept     | varchar(20) | YES  |     | NULL    |                |
| dest     | varchar(20) | YES  |     | NULL    |                |
| day      | varchar(15) | YES  |     | NULL    |                |
| f_time   | time        | YES  |     | NULL    |                |
| price    | int         | YES  |     | NULL    |                |
+----------+-------------+------+-----+---------+----------------+
11 rows in set (0.01 sec)

mysql> select * from customer_details;
+--------+--------------------+------------+-------+--------------+-----------+----------+---------+-----------+----------+-------+
| cus_id | cus_name           | mob_no     | fl_id | fl_name      | class     | dept     | dest    | day       | f_time   | price |
+--------+--------------------+------------+-------+--------------+-----------+----------+---------+-----------+----------+-------+
|  10001 | hibu myodi         | 1234567891 |  2001 | spice-jet    | bussiness | delhi    | mumbai  | sunday    | 04:00:00 | 14000 |
|  10002 | rejoice basumatary | 9876543212 |  2002 | air india    | middle    | guwahati | kolkata | monday    | 06:00:00 | 14000 |
|  10003 | Lungsom Lamnio     | 8413050187 |  2003 | go first     | economic  | haryana  | goa     | tuesday   | 01:00:00 | 13000 |
|  10004 | nangbia teji       | 8765987612 |  2004 | vistara      | bussiness | varoda   | chennai | wednesday | 15:00:00 | 20000 |
|  10005 | igon lona          | 9876587612 |  2005 | indigo       | middle    | hollongi | punjab  | thursday  | 23:00:00 | 22000 |
|  10006 | lakhi mili         | 1122334455 |  2006 | alliance air | economic  | guwahati | madras  | friday    | 17:00:00 | 24000 |
|  10007 | ankit das          | 1122556677 |  2007 | flybig       | bussiness | mumbai   | sikkim  | saturday  | 10:00:00 | 45000 |
+--------+--------------------+------------+-------+--------------+-----------+----------+---------+-----------+----------+-------+
```

- **Description and datas in table flight_details:**



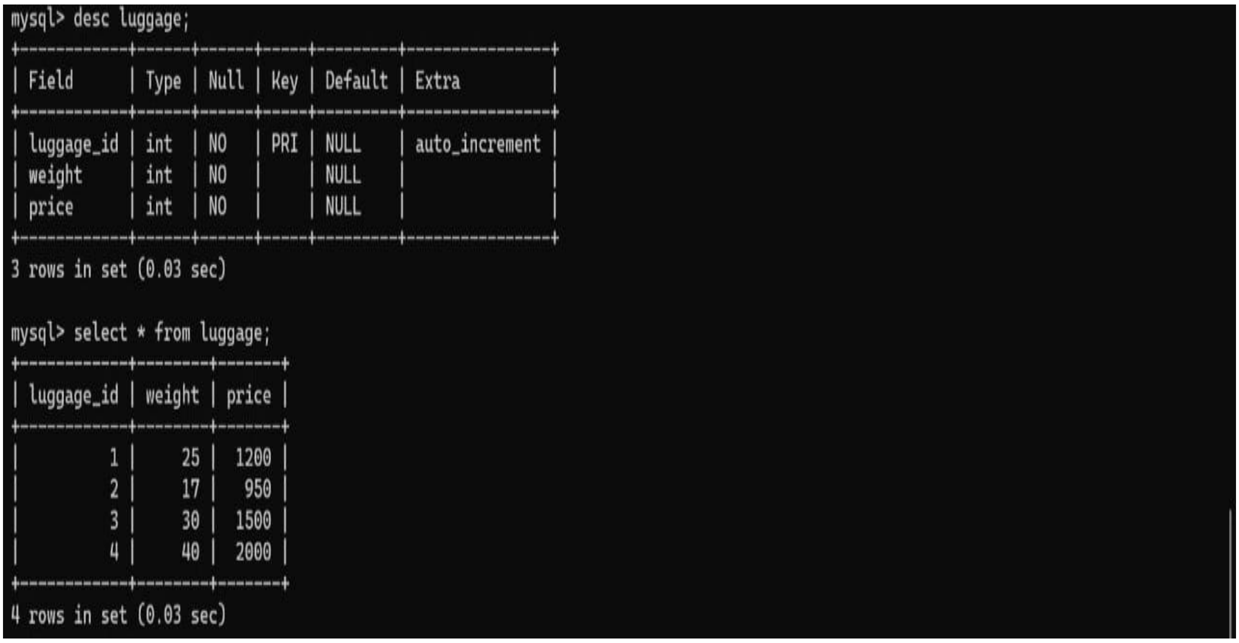
```
mysql> desc flight_details;
+-------------+-------------+------+-----+---------+----------------+
| Field       | Type        | Null | Key | Default | Extra          |
+-------------+-------------+------+-----+---------+----------------+
| flight_id   | int         | NO   | PRI | NULL    | auto_increment |
| flight_name | varchar(40) | YES  |     | NULL    |                |
| departure   | varchar(40) | YES  |     | NULL    |                |
| destination | varchar(40) | YES  |     | NULL    |                |
| day         | varchar(40) | YES  |     | NULL    |                |
| f_time      | time        | YES  |     | NULL    |                |
| bussiness   | int         | YES  |     | NULL    |                |
| middle      | int         | YES  |     | NULL    |                |
| economic    | int         | YES  |     | NULL    |                |
+-------------+-------------+------+-----+---------+----------------+
9 rows in set (0.00 sec)

mysql> select * from flight_details;
+-----------+--------------+-----------+-------------+-----------+----------+-----------+--------+----------+
| flight_id | flight_name  | departure | destination | day       | f_time   | bussiness | middle | economic |
+-----------+--------------+-----------+-------------+-----------+----------+-----------+--------+----------+
|      2001 | spice-jet    | delhi     | mumbai      | sunday    | 04:00:00 |     14000 |  15000 |    17000 |
|      2002 | air-india    | guwahati  | kolkata     | monday    | 06:00:00 |     13000 |  14000 |    16000 |
|      2003 | go-first     | haryana   | goa         | tuesday   | 01:00:00 |     10000 |  11000 |    13000 |
|      2004 | vistara      | varoda    | chennai     | wednesday | 15:00:00 |     20000 |  21000 |    23000 |
|      2005 | indigo       | hollongi  | punjab      | thursday  | 23:00:00 |     25000 |  26000 |    28000 |
|      2006 | alliance air | guwahati  | madras      | friday    | 17:00:00 |     21000 |  22000 |    24000 |
|      2007 | flybig       | mumbai    | los angeles | saturday  | 10:00:00 |     45000 |  46000 |    48000 |
+-----------+--------------+-----------+-------------+-----------+----------+-----------+--------+----------+
7 rows in set (0.03 sec)
```

Description and datas in table food_items:



```
mysql> desc food_items;
+-----------+-------------+------+-----+---------+----------------+
| Field     | Type        | Null | Key | Default | Extra          |
+-----------+-------------+------+-----+---------+----------------+
| sl_no     | int         | NO   | PRI | NULL    | auto_increment |
| food_name | varchar(40) | NO   |     | NULL    |                |
| price     | int         | NO   |     | NULL    |                |
+-----------+-------------+------+-----+---------+----------------+
3 rows in set (0.00 sec)

mysql> select * from food_items;
+-------+----------------+-------+
| sl_no | food_name      | price |
+-------+----------------+-------+
|     1 | pepsi          |   150 |
|     2 | coffee         |    70 |
|     3 | tea            |    50 |
|     4 | water          |    60 |
|     5 | milk shake     |    80 |
|     6 | chicken burger |   160 |
|     7 | cheese pizza   |    70 |
|     8 | chicken biryani|   300 |
|     9 | plane rice     |    80 |
|    10 | aloo paratha   |   120 |
|    11 | roti sabji     |   100 |
|    12 | omelette       |    50 |
+-------+----------------+-------+
12 rows in set (0.01 sec)
```

- **Description and datas in table luggage:**



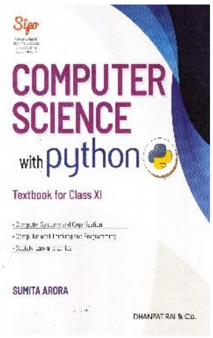
```
mysql> desc luggage;
+------------+------+------+-----+---------+----------------+
| Field      | Type | Null | Key | Default | Extra          |
+------------+------+------+-----+---------+----------------+
| luggage_id | int  | NO   | PRI | NULL    | auto_increment |
| weight     | int  | NO   |     | NULL    |                |
| price      | int  | NO   |     | NULL    |                |
+------------+------+------+-----+---------+----------------+
3 rows in set (0.03 sec)

mysql> select * from luggage;
+------------+--------+-------+
| luggage_id | weight | price |
+------------+--------+-------+
|          1 |     25 |  1200 |
|          2 |     17 |   950 |
|          3 |     30 |  1500 |
|          4 |     40 |  2000 |
+------------+--------+-------+
4 rows in set (0.03 sec)
```

BIBLOGRAPHY

- www.google.com

- www.wikipedia.org

- www.yahoo.com



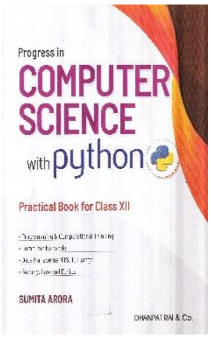
Sipo

COMPUTER SCIENCE with python

Textbook for Class XI

SUMITA ARORA

DHANPAT RAI & Co.



Progress in

COMPUTER SCIENCE with python

Practical Book for Class XII

SUMITA ARORA

DHANPAT RAI & Co.

- Class XI & XII NCERT Book