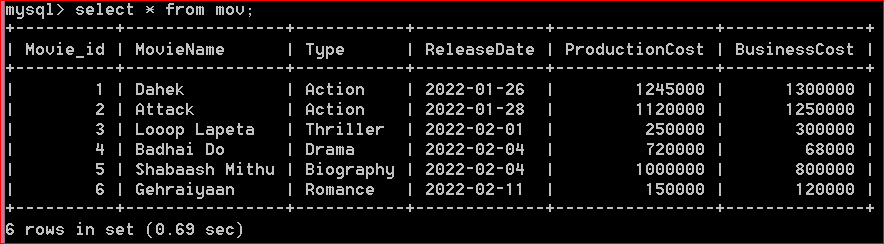
1. **Consider the following MOVIE table and write the SQL queries based on it.**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Movie\_id** | **MovieName** | **Type** | **ReleaseDate** | **ProductionCost** | **BusinessCost** |
| M001 | Dahek | Action | 2022/01/26 | 1245000 | 1300000 |
| M002 | Attack | Action | 2022/01/28 | 1120000 | 1250000 |
| M003 | Looop Lapeta | Thriller | 2022/02/01 | 250000 | 300000 |
| M004 | Badhai Do | Drama | 2022/02/04 | 720000 | 68000 |
| M005 | Shabaash Mithu | Biography | 2022/02/04 | 1000000 | 8000000 |
| M006 | Gehraiyaan | Romance | 2022/02/11 | 150000 | 120000 |

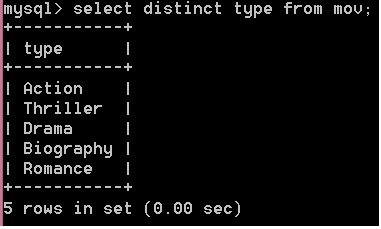
1. Display all information from movie.
2. Display the type of movies.
3. Display movieid, moviename, total\_eraning by showing the business done by the movies. Calculate the business done by movie using the sum of productioncost and businesscost.
4. Display movieid, moviename and productioncost for all movies with productioncost greater than 150000 and less than 1000000.
5. Display the movie of type action and romance.
6. Display the list of movies which are going to release in February, 2022.

**Answers:**

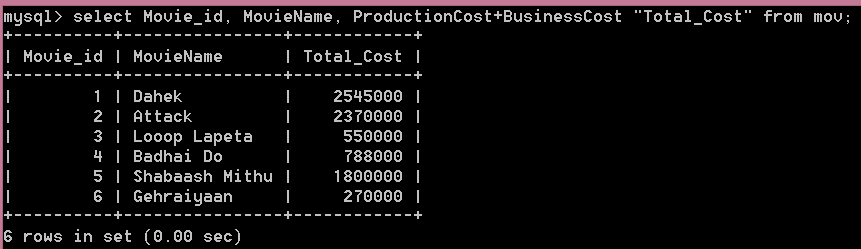
1. select \* from mov;



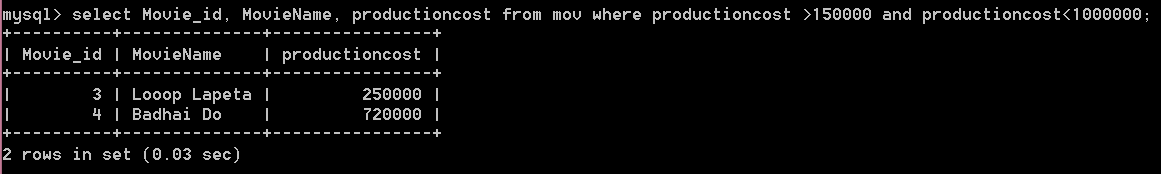
1. select distinct type from a mov;



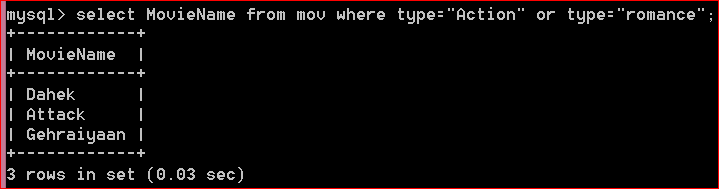
1. select movieid, moviename, productioncost + businesscost "total earning" from mov;



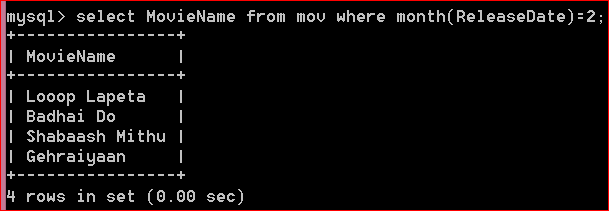
1. select movie\_id,moviename, productioncost from mov where producst is >150000 and <1000000;



1. select MovieName from mov where type ='action' or type='romance';



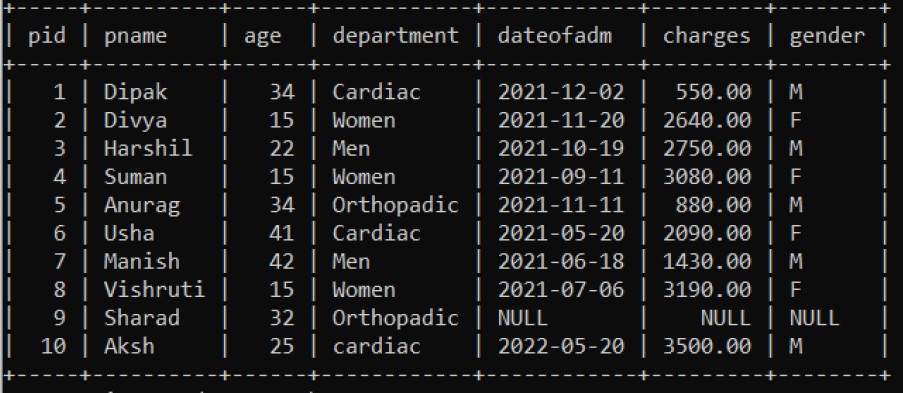
1. select moviename from movie where month(releasedate)=2;



1. **Consider the given table patient and Write following queries:**

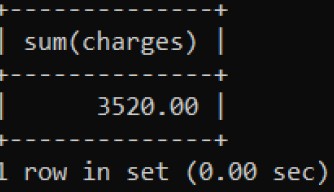
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Pid** | **Pname** | **Age** | **Department** | **Dataofdm** | **Charges** | **Gender** |
| 1 | Dipak | 34 | Cardiac | 2021-12-02 | 550.00 | M |
| 2 | Divya | 15 | Women | 2021-11-20 | 2640.00 | F |
| 3 | Harshil | 22 | Men | 2021-10-19 | 2750.00 | M |
| 4 | Suman | 15 | Women | 2021-09-11 | 3080.00 | F |
| 5 | Anurag | 34 | Orthopedic | 2021-11-11 | 880.00 | M |
| 6 | Usha | 41 | Cardiac | 2021-05-20 | 2090.00 | F |
| 7 | Manish | 42 | Men | 2021-06-18 | 1430.00 | M |
| 8 | Vishuruti | 15 | Women | 2021-07-06 | 3190.00 | F |
| 9 | Sharad | 32 | Orthopedic | NULL | NULL | NULL |
| 10 | Aksh | 25 | Cardiac | 2022-05-20 | 3500.00 | M |

1. Display the total charges of patient admitted in the month of November.
2. Display the eldest patient with name and age.
3. Count the unique departments.
4. Display an average charges.

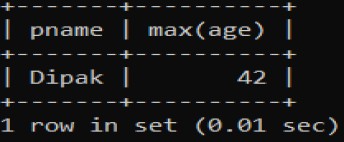


**Answers:**

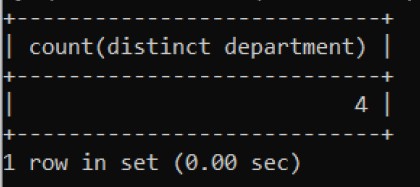
1. select sum(charges) from patient where dateofadm like '%-11-%';



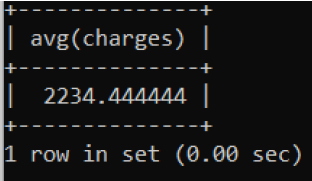
1. select pname,max(age) from patient;



1. Select count(distinct department) from patient;



1. Select avg(charges) from patient;

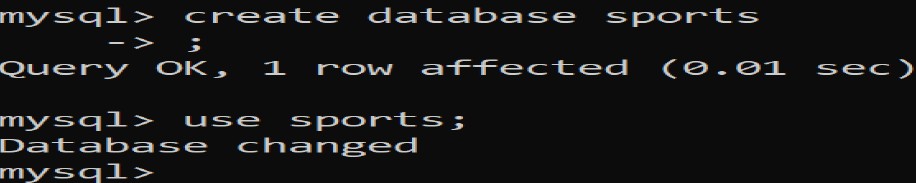


1. **Suppose your school management has decided to conduct cricket matches between students of Class XI and Class XII. Students of each class are asked to join any one of the four teams – Team Titan, Team Rockers, Team Magnet and Team Hurricane. During summer vacations, various matches will be conducted between these teams. Help your sports teacher to do the following:**
   1. Create a database “Sports”.
   2. Create a table “TEAM” with following considerations:
      1. It should have a column TeamID for storing an integer value between 1 to 9, which refers to unique identification of a team.
      2. Each TeamID should have its associated name (TeamName), which should be a string of length not less than 10 characters.
      3. Using table level constraint, make TeamID as the primary key.
   3. Show the structure of the table TEAM using a SQL statement.
   4. As per the preferences of the students four teams were formed as given below. Insert these four rows in TEAM table:
      1. Row 1: (1, Tehlka)
      2. Row 2: (2, Toofan)
      3. Row 3: (3, Aandhi)
      4. Row 3: (4, Shailab)
   5. Show the contents of the table TEAM using a DML statement.
   6. Now create another table MATCH\_DETAILS and insert data as shown below. Choose appropriate data types and constraints for each attribute.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| MatchId | MatchDate | FirstTeamID | SecondTemID | FristTeamScore | SecondTeamScore |
| M1 | 2021/12/20 | 1 | 2 | 107 | 93 |
| M2 | 2021/12/21 | 3 | 4 | 156 | 158 |
| M3 | 2021/12/22 | 1 | 3 | 86 | 81 |
| M4 | 2021/12/23 | 2 | 4 | 65 | 67 |
| M5 | 2021/12/24 | 1 | 4 | 52 | 88 |
| M6 | 2021/12/25 | 2 | 3 | 97 | 68 |

**Answers:**

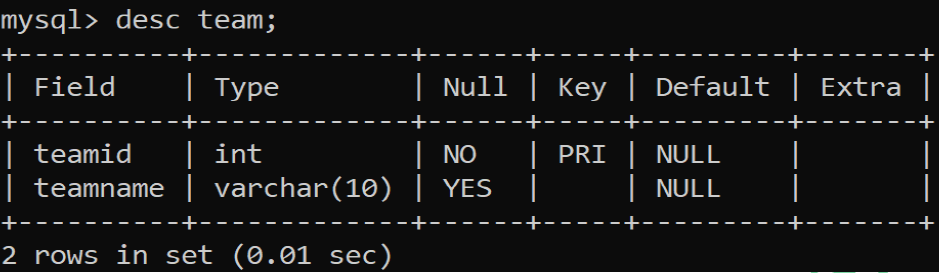
1. create database sports;



1. Creating table with the given specification

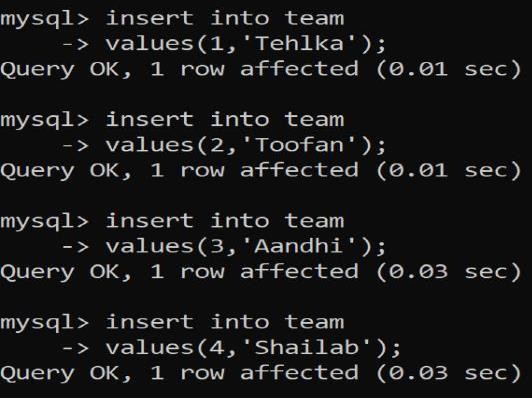
create table team (teamid int(1),teamname varchar(10), primary key(teamid));

1. desc team;



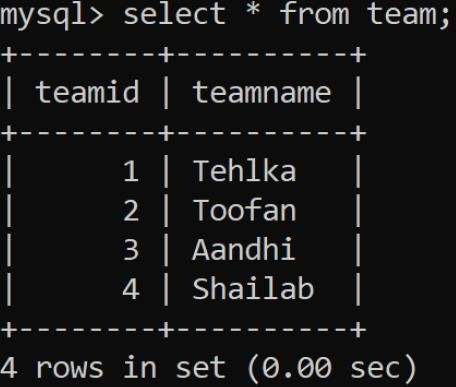
* **Inserting data:**

mqsql> insert into team values(1,'Tehlka');



* **Show the content of table - team:**

select \* from team;



* **Creating another table:**

create table match\_details

-> (matchid varchar(2) primary key,

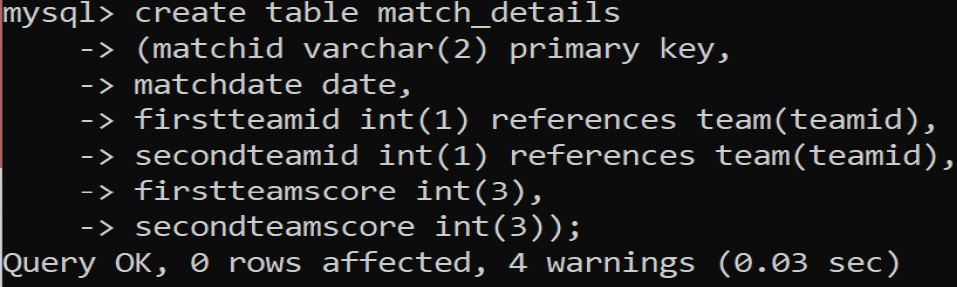
-> matchdate date,

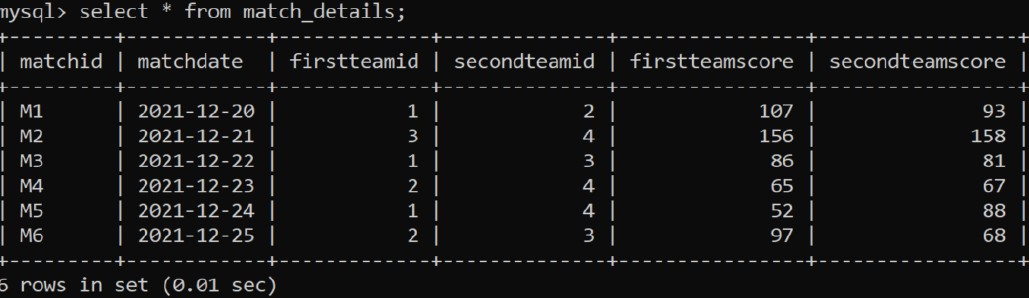
-> firstteamid int(1) references team(teamid),

-> secondteamid int(1) references team(teamid),

-> firstteamscore int(3),

-> secondteamscore int(3));

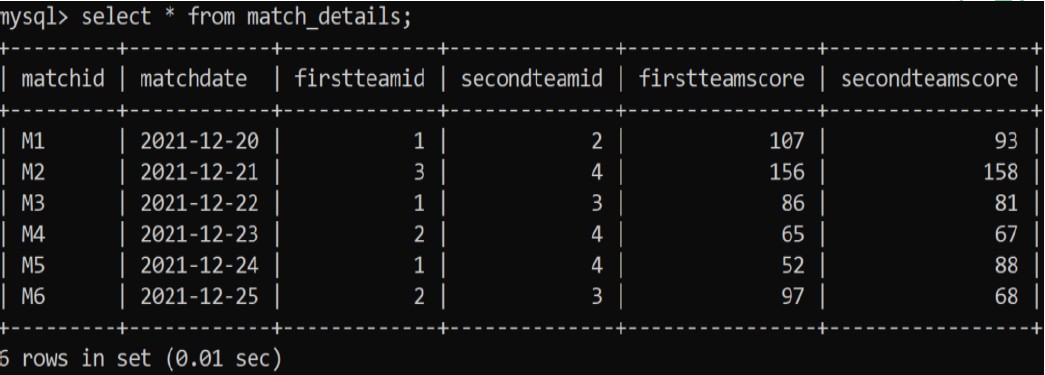




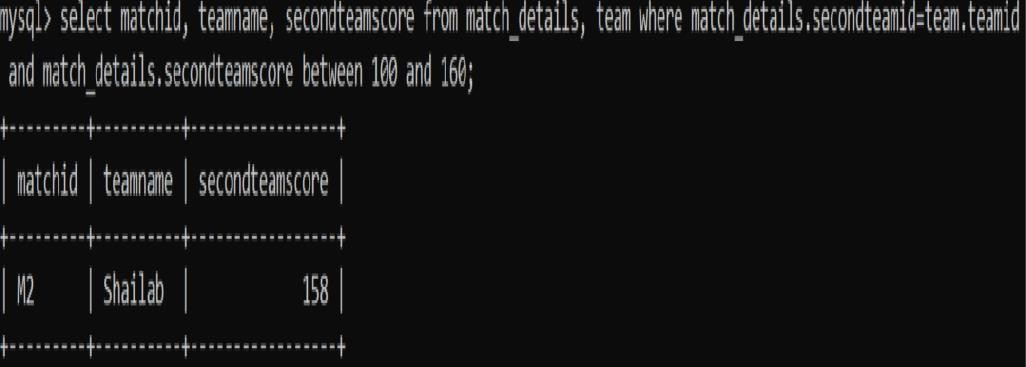
1. Write following queries:
   1. Display the matchid, teamid, teamscore whoscored more than 70 in first ining along with team name.
   2. Display matchid, teamname and secondteamscore between 100 to 160.
   3. Display matchid, teamnames along with matchdates.
   4. Display unique team names
   5. Display matchid and matchdate played by Anadhi and Shailab.

**Answers:**

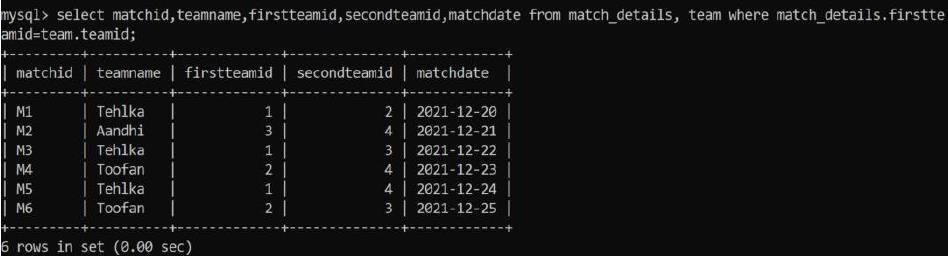
1. select match\_details.matchid, match\_details.firstteamid, team.teamname,match\_details.firstteamscore from match\_details, team where match\_details.firstteamid = team.teamid and match\_details.first



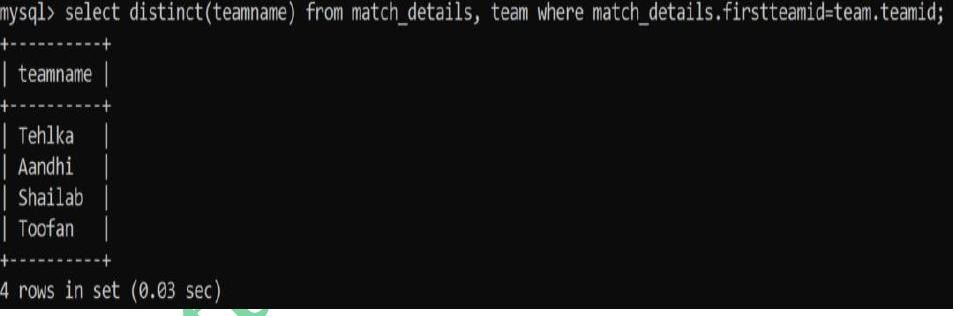
1. select match\_details.matchid, match\_details.firstteamid, team.teamname,match\_details.firstteamscore from match\_details, team where match\_details.firstteamid = team.teamid and match\_details.firstteamscore>70;



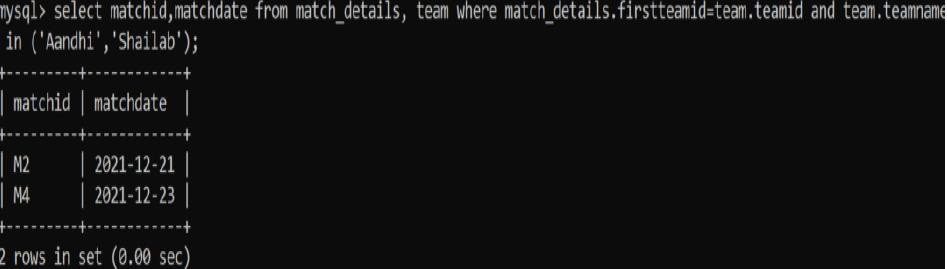
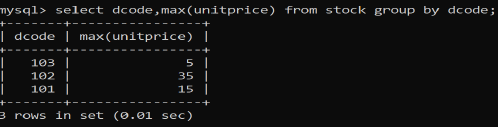
1. select matchid, teamname, firstteamid, secondteamid, matchdate from match\_details, team where match\_details.firstteamid = team.teamid;



1. select distinct(teamname) from match\_details, team where match\_details.firstteamid = team.teamid;



1. select matchid,matchdate from match\_details, team where match\_details.firstteamid = team.teamid and team.teamname in ('Aandhi','Shailab');



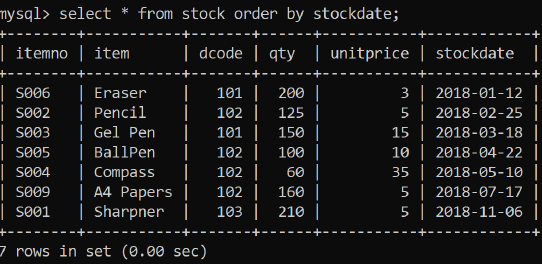
1. Consider the following table and write the queries:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Itemno | Item | Dcode | Qty | Unitprice | Stockdate |
| S005 | Ballpen | 102 | 100 | 10 | 2018/04/22 |
| S003 | GelPen | 101 | 150 | 15 | 2018/03/18 |
| S002 | Pencil | 102 | 125 | 5 | 2018/02/25 |
| S006 | Eraser | 101 | 200 | 3 | 2018/02/25 |
| S001 | Sharpner | 103 | 210 | 5 | 2018/06/11 |
| S004 | Compass | 102 | 60 | 35 | 2018/05/10 |
| S009 | A4 Papers | 102 | 160 | 5 | 2018/07/17 |

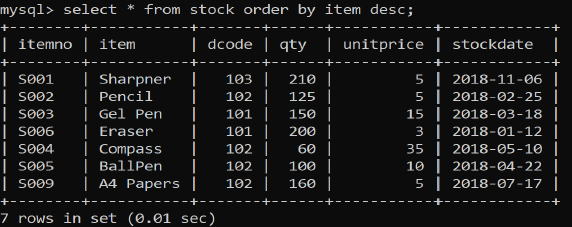
1. Display all the items in the ascending order of stockdate.
2. Display maximum price of items for each dealer individually as per dcode from stock.
3. Display all the items in descending orders of itemnames.
4. Display average price of items for each dealer individually as per doce from stock which avergae price is more than 5.
5. Diisplay the sum of quantity for each dcode.

**Answers:**

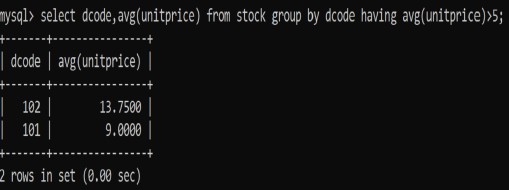
1. select \* from stock order by stockdate;



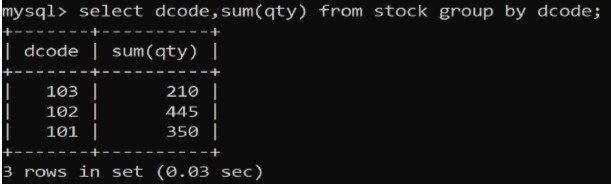
1. select dcode,max(unitprice) from stock group by code;
2. select \* from stock order by item desc;



1. select dcode,avg(unitprice) from stock group by dcode having avg(unitprice)>5;



1. select dcode,sum(qty) from stock group by dcode;



**PART 5 - PYTHON DATABASE CONNECTIVITY**

1. **Write a MySQL connectivity program in Python to**

* **Create a database school**
* **Create a table students with the specifications - ROLLNO integer, STNAME character(10) in MySQL and perform the following operations:** o **Insert two records in it**

o **Display the contents of the table**

1. **Perform all the operations with reference to table ‘students’ through MySQL-Python connectivity.**

**Answers:**

1. **Using pymysql - Code:**

|  |
| --- |
| **PROGRAM** |
| import pymysql as ms  #Function to create Database as per users choice def c\_database():  try:  dn=input("Enter Database Name=") c.execute("create database {}".format(dn)) c.execute("use {}".format(dn)) print("Database created successfully")  except Exception as a: print("Database Error",a)  #Function to Drop Database as per users choice def d\_database():  try:  dn=input("Enter Database Name to be dropped=") c.execute("drop database {}".format(dn)) print("Database deleted sucessfully")  except Exception as a: print("Database Drop Error",a)  #Function to create Table def c\_table():  try:  c.execute('''create table students (rollno int(3),stname varchar(20));''') print("Table created successfully")  except Exception as a: print("Create Table Error",a)  #Function to Insert Data def e\_data():  try: |

while True:

rno=int(input("Enter student rollno=")) name=input("Enter student name=") c.execute("use {}".format('school'))

c.execute("insert into students values({},'{}');".format(rno,name)) db.commit()

choice=input("Do you want to add more record<y/n>=") if choice in "Nn":

break

except Exception as a: print("Insert Record Error",a)

#Function to Display Data def d\_data():

try:

c.execute("select \* from students") data=c.fetchall()

for i in data: print(i)

except Exception as a: print("Display Record Error",a)

db=ms.connect(host="localhost",user="root",password="root") c=db.cursor()

while True:

print("MENU\n1. Create Database\n2. Drop Database \n3. Create Table\n4. Insert Record \n5.

Display Entire Data\n6. Exit") choice=int(input("Enter your choice<1-6>=")) if choice==1:

c\_database() elif choice==2: d\_database()

elif choice==3: c\_table()

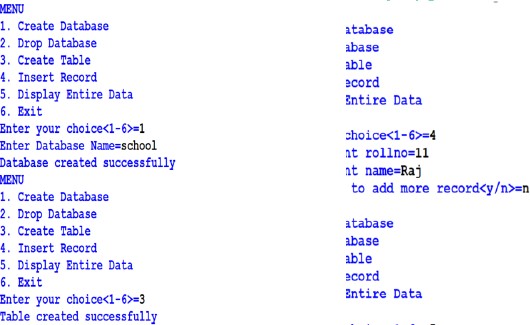
elif choice==4: e\_data()

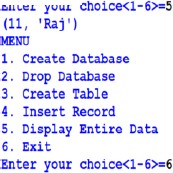
elif choice==5: d\_data()

elif choice==6: break

else:

print("Wrong option selected")





1. **using mysqlconnector**

import mysql.connector as ms db=ms.connect(host="localhost",user="root",passwd="root",database='school') cn=db.cursor()

def insert\_rec():

try:

while True:

rn=int(input("Enter roll number:")) sname=input("Enter name:") marks=float(input("Enter marks:")) gr=input("Enter grade:")

cn.execute("insert into students values({},'{}',{},'{}')".format(rn,sname,marks,gr)) db.commit()

ch=input("Want more records? Press (N/n) to stop entry:") if ch in 'Nn':

break

except Exception as e: print("Error", e)

def update\_rec():

try:

rn=int(input("Enter rollno to update:")) marks=float(input("Enter new marks:"))

gr=input("Enter Grade:")

cn.execute("update students set marks={},grade='{}' where rno={}".format(marks,gr,rn)) db.commit()

except Exception as e:

print("Error",e) def delete\_rec():

try:

rn=int(input("Enter rollno to delete:"))

cn.execute("delete from students where rno={}".format(rn)) db.commit()

except Exception as e:

print("Error",e) def view\_rec():

try:

cn.execute("select \* from students") import mysql.connector as ms

db=ms.connect(host="localhost",user="root",passwd="root",database='school') cn=db.cursor()

def insert\_rec():

try:

while True:

rn=int(input("Enter roll number:")) sname=input("Enter name:") marks=float(input("Enter marks:")) gr=input("Enter grade:")

cn.execute("insert into students values({},'{}',{},'{}')".format(rn,sname,marks,gr)) db.commit()

ch=input("Want more records? Press (N/n) to stop entry:") if ch in 'Nn':

break

except Exception as e:

print("Error", e) def update\_rec():

try:

rn=int(input("Enter rollno to update:")) marks=float(input("Enter new marks:")) gr=input("Enter Grade:")

cn.execute("update students set marks={},grade='{}' where rno={}".format(marks,gr,rn)) db.commit()

except Exception as e: print("Error",e)

def delete\_rec():

try:

rn=int(input("Enter rollno to delete:"))

cn.execute("delete from students where rno={}".format(rn)) db.commit()

except Exception as e:

print("Error",e) def view\_rec():

try:

cn.execute("select \* from students") data=c.fetchall()

for i in data:

print(i)

except Exception as e:

print("Error",e) while True:

print("MENU\n1. Insert Record\n2. Update Record \n3. Delete Record\n4. Display Record \n5. Exit")

ch=int(input("Enter your choice<1-4>=")) if ch==1:

insert\_rec() elif ch==2:

update\_rec() elif ch==3:

delete\_rec() elif ch==4:

view\_rec() elif ch==5:

break else:

print("Wrong option selected") except Exception as e:

print("Error",e) while True:

print("MENU\n1. Insert Record\n2. Update Record \n3. Delete Record\n4. Display Record \n5. Exit")

ch=int(input("Enter your choice<1-4>=")) if ch==1:

insert\_rec() elif ch==2: update\_rec() elif ch==3: delete\_rec() elif ch==4: view\_rec() elif ch==5: break

else:

print("Wrong option selected")



1. **Write a menu-driven program to store data into a MySQL database named shop and table customer as following:**
2. Add customer details
3. Update customer details
4. Delete customer details
5. Display all customer details

import mysql.connector as ms db=ms.connect(host="localhost",user="root",passwd="root",database='mydb') cn=db.cursor()

def insert\_rec():

try:

while True:

cid=int(input("Enter customer id:")) cname=input("Enter name:") city=input("Enter city:") bill\_amt=float(input("Enter bill amount:")) cat=input("Enter category:")

cn.execute("insert into customer values({},'{}','{}',{},'{}')".format(cid,cname,city,bill\_amt,cat)) db.commit()

ch=input("Want more records? Press (N/n) to stop entry:") if ch in 'Nn':

break

except Exception as e:

print("Error", e) def update\_rec():

try:

cn.execute("select \* from customer") data=cn.fetchall()

for i in data:

ci=i[0] cna=i[1] ct=i[2] b=i[3]

c=i[4]

cid=int(input("Enter customer id to update:")) if cid==ci:

ch\_cname=input("Want to update Name, Press 'Y':") if ch\_cname.lower()=='y':

cname=input("Enter new name:") else:

cname=cna

ch\_city=input("Want to update city, Press 'Y':") if ch\_city.lower()=='y':

city=input("Enter new city:") else:

city=ct

ch=input("Want to update bill amount, Press 'Y':") if ch.lower()=='y':

bill\_amt=float(input("Enter new bill amount:")) else:

bill\_amt=b

ch\_cat=input("Want to update Category, Press 'Y':") if ch\_cat.lower()=='y':

cat=input("Enter new category:") else:

cat=c

cn.execute("update customer set cname='{}', city='{}', bill\_amt={},category='{}' where cust\_id={}".format(cname,city,bill\_amt,cat,cid))

db.commit() else:

print("Record Not Found...") except Exception as e:

print("Error",e) def delete\_rec():

try:

cid=int(input("Enter rollno to delete:"))

cn.execute("delete from customer where cust\_id={}".format(cid)) db.commit()

except Exception as e:

print("Error",e) def view\_rec():

try:

cn.execute("select \* from customer") data=cn.fetchall()

cnt=0

for i in data:

cnt=cnt+1 print("Record:",cnt) print('~'\*50) print("Customer ID:",i[0])

print("Customer Name:",i[1]) print("City:",i[2])

print("Bill Amount:",i[3])

print("Category:",i[4]) print('~'\*50)

except Exception as e:

print("Error",e) while True:

print("MENU\n1. Insert Record\n2. Update Record \n3. Delete Record\n4. Display Record \n5. Exit")

ch=int(input("Enter your choice<1-4>=")) if ch==1:

insert\_rec() elif ch==2: update\_rec() elif ch==3: delete\_rec()

elif ch==4:

view\_rec() elif ch==5:

break else:

print("Wrong option selected")

|  |  |
| --- | --- |
|  |  |

1. **Modify the above program and display the customer details based on the following menu:**
   1. Display customer details by city
   2. Display customer details by bill amount
   3. Display customer details by name
   4. Display customer details by category

import mysql.connector as ms db=ms.connect(host="localhost",user="root",passwd="root",database='mydb') cn=db.cursor()

def byCity(): try:

city=input("Enter city to search:")

cn.execute("select \* from customer where city='{}'".format(city)) data=cn.fetchall()

if data!=[]:

cnt=0

for i in data:

cnt=cnt+1 print('~'\*100) print("Record:",cnt) print('~'\*100)

print("Customer ID:",i[0]) print("Customer Name:",i[1]) print("City:",i[2])

print("Bill Amount:",i[3])

print("Category:",i[4]) else:

print("No records found for city ", city) except Exception as e:

print("Error",e)

def byBillAmt():

try:

ba=input("Enter the bill amount:")

cn.execute("select \* from customer where bill\_amt={}".format(ba)) data=cn.fetchall()

if data!=[]:

cnt=0

for i in data:

cnt=cnt+1 print('~'\*100) print("Record:",cnt) print('~'\*100)

print("Customer ID:",i[0]) print("Customer Name:",i[1]) print("City:",i[2])

print("Bill Amount:",i[3])

print("Category:",i[4]) else:

print("No records found for bill amount ", ba) except Exception as e:

print("Error",e) def byName():

try:

name=input("Enter the name:")

cn.execute("select \* from customer where cname='{}'".format(name)) data=cn.fetchall()

if data!=[]:

cnt=0

for i in data:

cnt=cnt+1 print('~'\*100) print("Record:",cnt) print('~'\*100)

print("Customer ID:",i[0]) print("Customer Name:",i[1]) print("City:",i[2])

print("Bill Amount:",i[3])

print("Category:",i[4]) else:

print("No records found for ", name) except Exception as e:

print("Error",e) def byCat():

try:

cat=input("Enter the cat:")

cn.execute("select \* from customer where category='{}'".format(cat)) data=cn.fetchall()

if data!=[]:

cnt=0

for i in data:

cnt=cnt+1 print('~'\*100) print("Record:",cnt) print('~'\*100)

print("Customer ID:",i[0]) print("Customer Name:",i[1]) print("City:",i[2])

print("Bill Amount:",i[3])

print("Category:",i[4]) else:

print("No records found for category ", cat) except Exception as e:

print("Error",e) while True:

print(''' MENU

1.Display customer details by city 2.Display customer details by bill amount 3.Display customer details by name

4.Display customer details by category 5.Exit ''')

ch=int(input("Enter your choice<1-4>=")) if ch==1:

byCity() elif ch==2: byBillAmt() elif ch==3:

byName() elif ch==4:

byCat() elif ch==5:

break else:

print("Wrong option selected")

|  |  |
| --- | --- |
|  |  |