

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Practical | Date | Signature |
| 1 | Write a menu-driven python program to implement stack  operation. |  |  |
| 2 | Write a program to implement a stack for the employee details  (empno, name). |  |  |
| 3 | Write a python program to check whether a string is a  palindrome or not using stack. |  |  |
| 4 | Queries Set 1 (Database Fetching records) |  |  |
| 5 | Queries Set 2 (Based on Functions) |  |  |
| 6 | Queries Set 3 (DDL Commands) |  |  |
| 7 | Queries set 4 (Based on Two Tables) |  |  |
| 8 | Queries Set 5 (Group by , Order By) |  |  |
| 9 | 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:   + Insert two records in it   + Display the contents of the table |  |  |
| 10 | Perform all the operations with reference to table ‘students’  through MySQL-Python connectivity. |  |  |





### Write a menu-driven python program to implement stack operation. Code:

**PART A DATA STRUCTURE**

def check\_stack\_isEmpty(stk): if stk==[]:

return True else:

return False

# An empty list to store stack elements, initially empty s=[]

top = None # This is top pointer for push and pop

def main\_menu(): while True:

print("Stack Implementation") print("1 - Push")

print("2 - Pop")

print("3 - Peek") print("4 - Display") print("5 - Exit")

ch = int(input("Enter the your choice:")) if ch==1:

el = int(input("Enter the value to push an

element:"))

push(s,el)

elif ch==2:

e=pop\_stack(s)

if e=="UnderFlow":

print("Stack is underflow!") else:

print("Element popped:",e) elif ch==3:

e=pop\_stack(s)

if e=="UnderFlow":

print("Stack is underflow!") else:

print("The element on top is:",e) elif ch==4:

display(s) elif ch==5:



break else:

print("Sorry, You have entered invalid option")

def push(stk,e): stk.append(e) top = len(stk)-1

def display(stk):

if check\_stack\_isEmpty(stk): print("Stack is Empty")

else:

top = len(stk)-1 print(stk[top],"-Top")

for i in range(top-1,-1,-1): print(stk[i])

def pop\_stack(stk):

if check\_stack\_isEmpty(stk): return "UnderFlow"

else:

e = stk.pop() if len(stk)==0:

top = None else:

top = len(stk)-1 return e

def peek(stk):

if check\_stack\_isEmpty(stk): return "UnderFlow"

else:

top = len(stk)-1 return stk[top]

### Output:



|

### Write a program to implement a stack for the employee details (empno, name). Code:



stk=[] top=-1

def line(): print('~'\*100)

def isEmpty(): global stk if stk==[]:

print("Stack is empty!!!") else:

None

def push(): global stk global top

empno=int(input("Enter the employee number to push:")) ename=input("Enter the employee name to push:") stk.append([empno,ename])

top=len(stk)-1

def display(): global stk global top if top==-1:

isEmpty() else:

top=len(stk)-1 print(stk[top],"<-top")

for i in range(top-1,-1,-1): print(stk[i])

def pop\_ele(): global stk global top if top==-1:

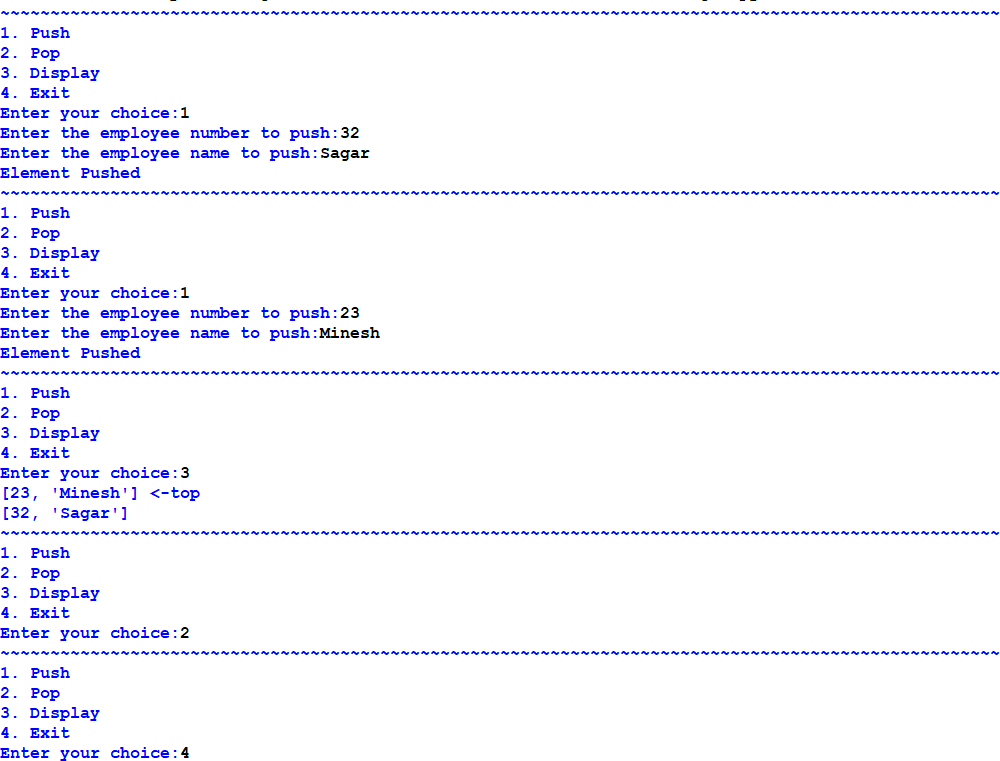
isEmpty() else:

stk.pop() top=top-1

def main(): while True:

line()

print("1. Push")



print("2. Pop") print("3. Display") print("4. Exit")

ch=int(input("Enter your choice:")) if ch==1:nm

push()

print("Element Pushed") elif ch==2:

pop\_ele() elif ch==3: display() elif ch==4:

break else:

print("Invalid Choice")

### Output:

### Write a python program to check whether a string is a palindrome or not using stack.



**Code:**

stack = [] top = -1

# push function def push(ele):

global top top += 1

stack[top] = ele

# pop function def pop():

global top

ele = stack[top] top -= 1

return ele

# Function that returns 1 if string is a palindrome def isPalindrome(string):

global stack

length = len(string)

# Allocating the memory for the stack stack = ['0'] \* (length + 1)

# Finding the mid mid = length // 2 i = 0

while i < mid:

push(string[i]) i += 1

# Checking if the length of the string is odd then neglect the middle character

if length % 2 != 0:

i += 1

# While not the end of the string while i < length:

ele = pop()

# If the characters differ then the given string is not a palindrome

if ele != string[i]: return False

i += 1

return True

string = input("Enter string to check:")

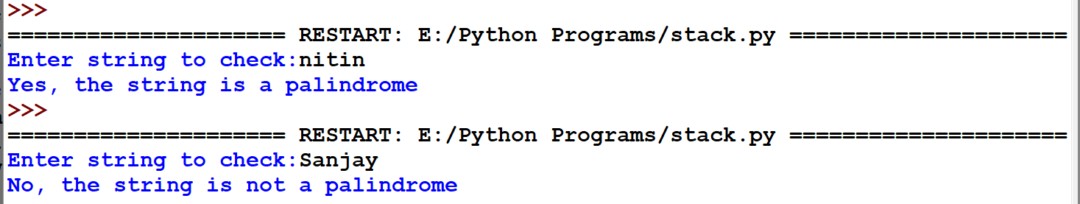
if isPalindrome(string):

print("Yes, the string is a palindrome")

else:

### Output:

print("No, the string is not a palindrome")







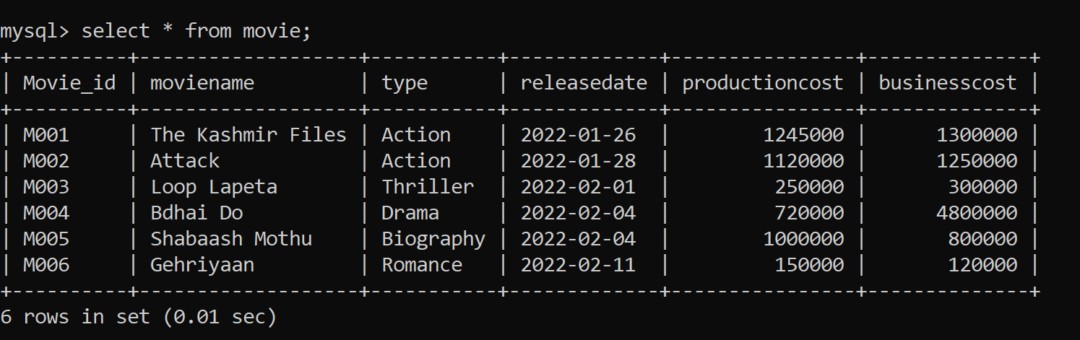
**PART B SQL QUERIES**

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

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Movie\_ID | MovieName | Type | ReleaseDate | ProductionCost | BusinessCost |
| M001 | The Kashmir Files | 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 | 800000 |
| M006 | Gehraiyaan | Romance | 2022/02/11 | 150000 | 120000 |

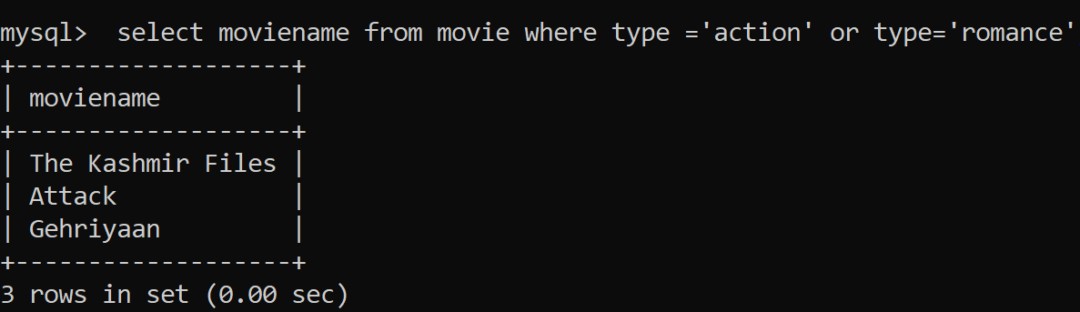
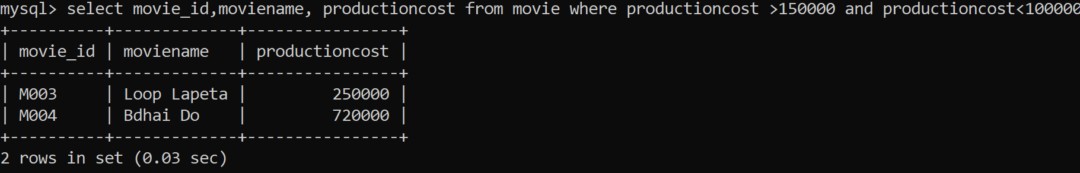
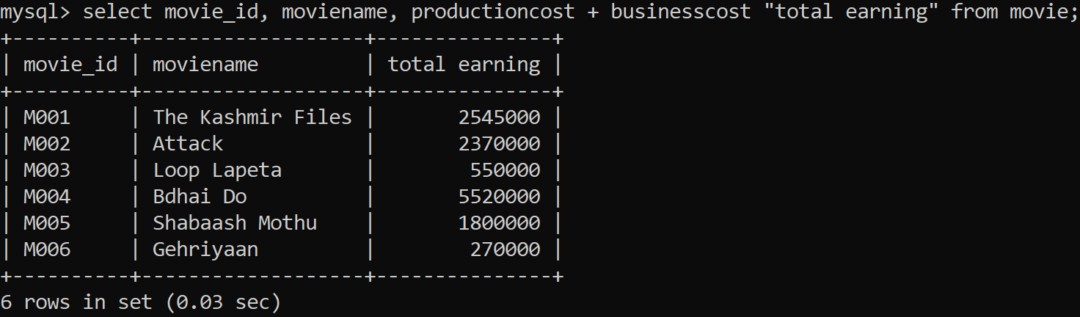
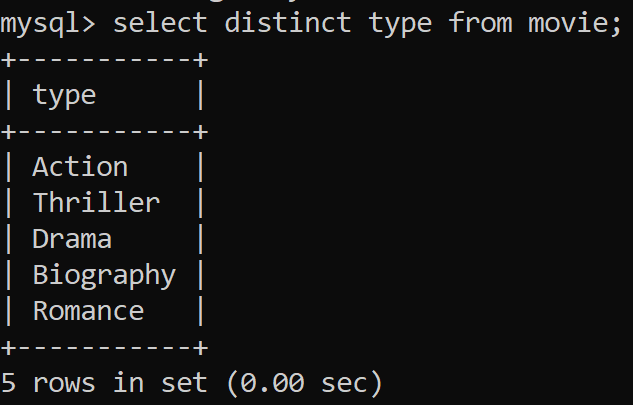
* 1. Display all information from movie.
  2. Display the type of movies.
  3. Display movieid, moviename, total\_eraning by showing the business done by the movies. Claculate the business done by movie using the sum of productioncost and businesscost.
  4. Display movieid, moviename and productioncost for all movies with productioncost greater thatn 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 movie;

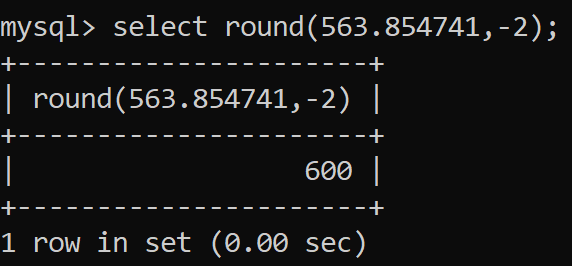
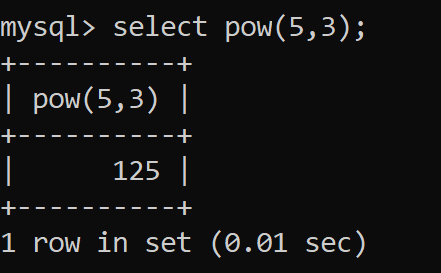
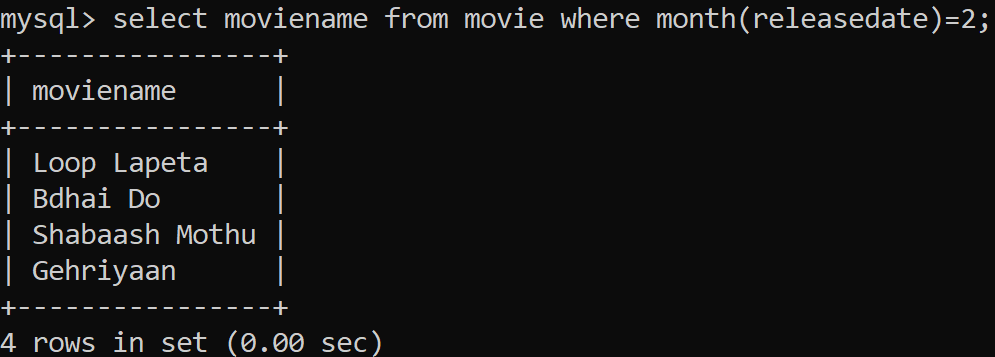




1. select distinct from a movie;



1. select movieid, moviename, productioncost + businesscost "total earning" from movie;
2. select movie\_id,moviename, productioncost from movie where producst is >150000 and <1000000;
3. select moviename from movie where type ='action' or type='romance';
4. select moviename from moview where month(releasedate)=2;



### Write following queries:

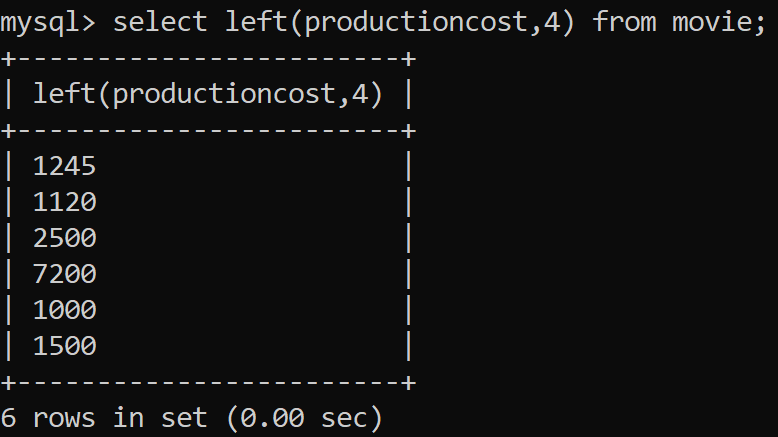
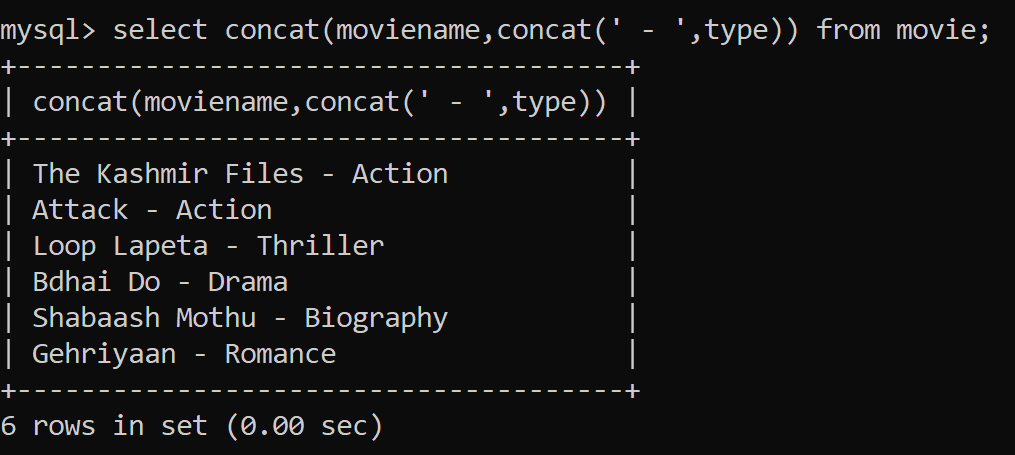
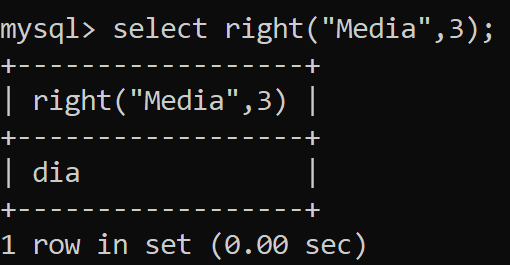
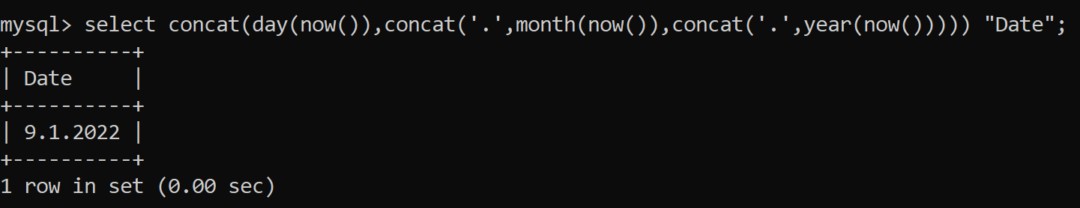
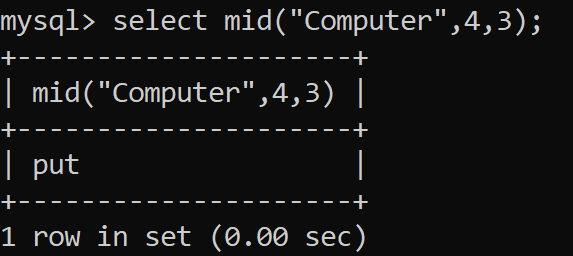
* 1. Write a query to display cube of 5.
  2. Write a query to display the number 563.854741 rounding off to the next hnudred.
  3. Write a query to display "put" from the word "Computer".
  4. Write a query to display today's date into DD.MM.YYYY format.
  5. Write a query to display 'DIA' from the word "MEDIA".
  6. Write a query to display moviename - type from the table movie.
  7. Write a query to display first four digits of productioncost.
  8. Write a query to display last four digits of businesscost.
  9. Write a query to display weekday of release dates.
  10. Write a query to display dayname on which movies are going to be released.

**Answers:**

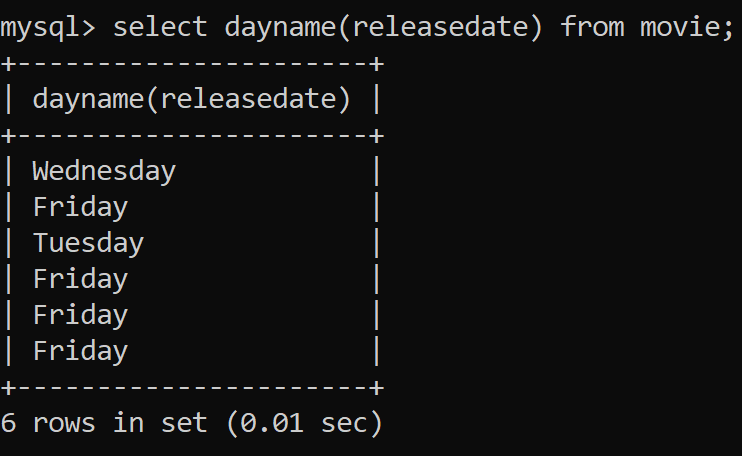
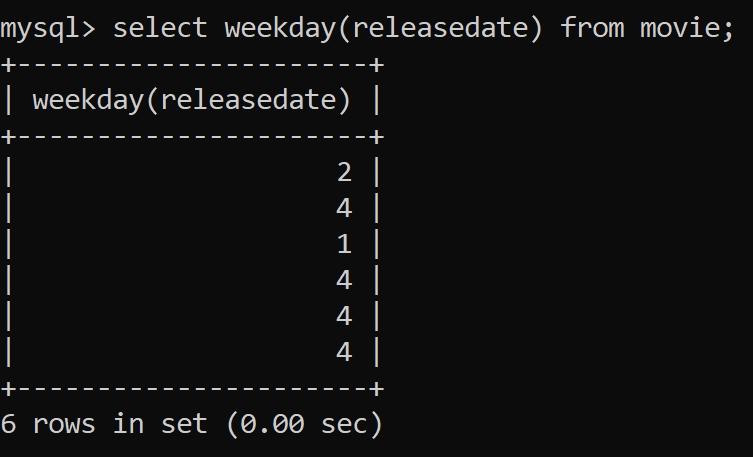
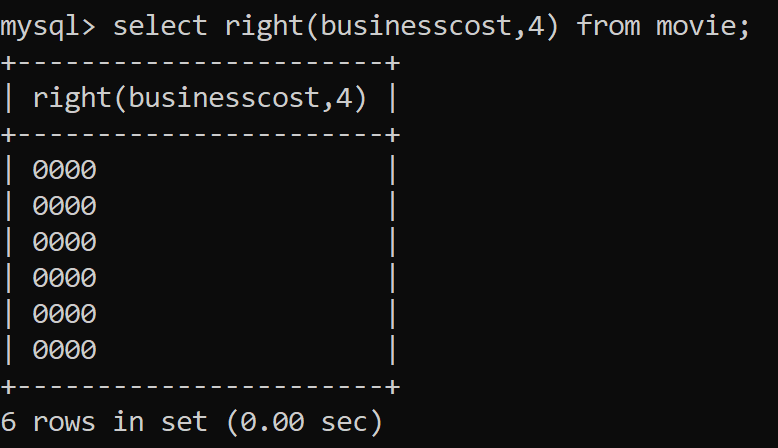
* + 1. select pow(5,3);

b) select round(563.854741,-2);

1. select mid("Computer",4,3);



1. select concat(day(now()), concat('.',month(now()), concat('.',year(now())))) "Date";
2. select right("Media",3);
3. select concat(moviename,concat(' - ',type)) from movie;
4. select left(productioncost,4) from movie;
5. select right(businesscost,4) from movie;



1. select weekday(releasedate) from movie;
2. select dayname(releasedate) from movie;

### 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.





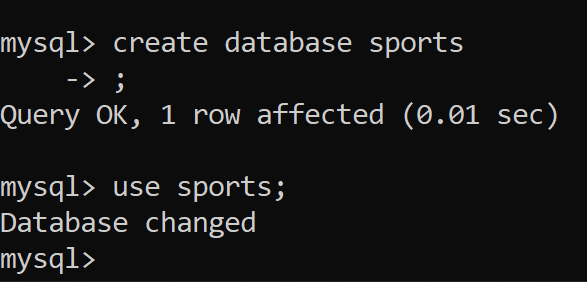
* 1. Show the structure of the table TEAM using a SQL statement.
  2. As per the preferences of the students four teams were formed as given below. Insert these four rows in TEAM table:

1. Row 1: (1, Tehlka)
2. Row 2: (2, Toofan)
3. Row 3: (3, Aandhi)
4. Row 3: (4, Shailab)
   1. Show the contents of the table TEAM using a DML statement.
   2. Now create another table MATCH\_DETAILS and insert data as shown below. Choose appropriate data types and constraints for each attribute.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| MatchID | MatchDate | FirstTeamID | SecondTeamID | FirstTeamScore | 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:

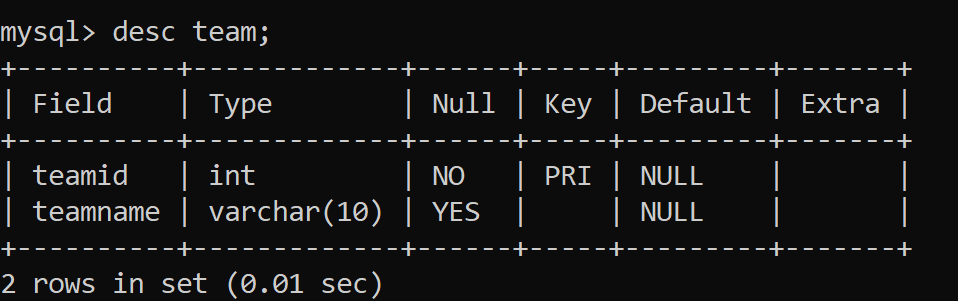
1. create database sports;



1. Creating table with the given specification

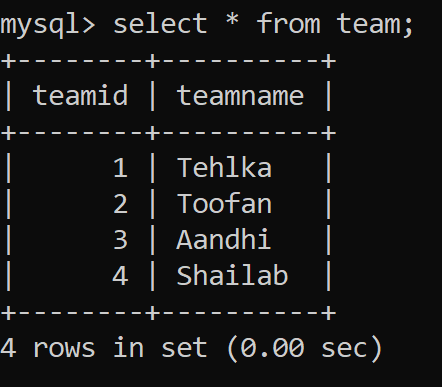
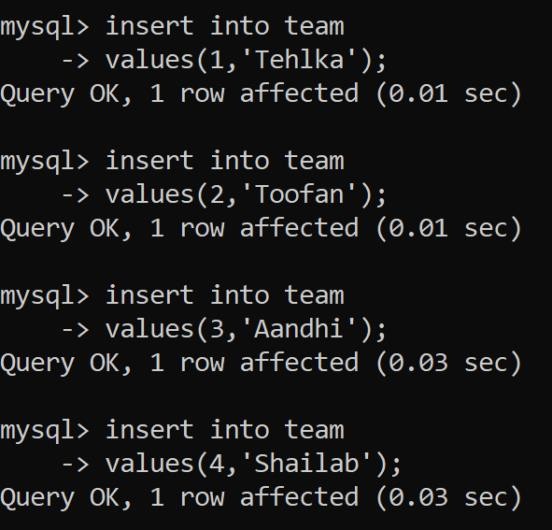
create table team -> (teamid int(1), -> teamname varchar(10), primary key(teamid));

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,

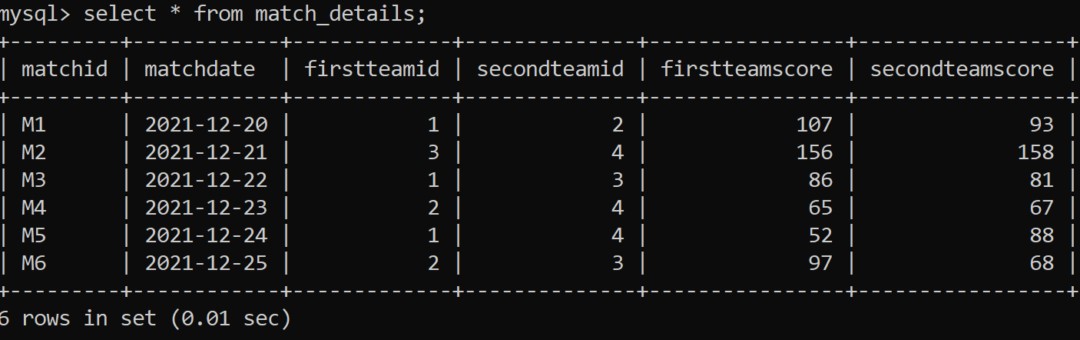
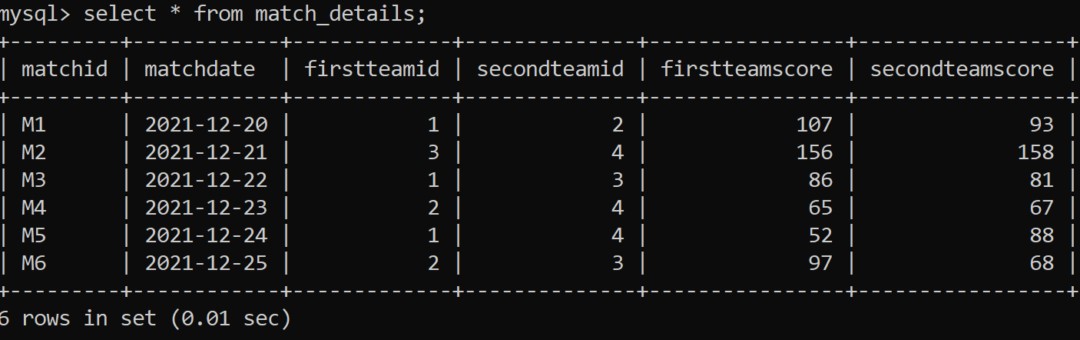
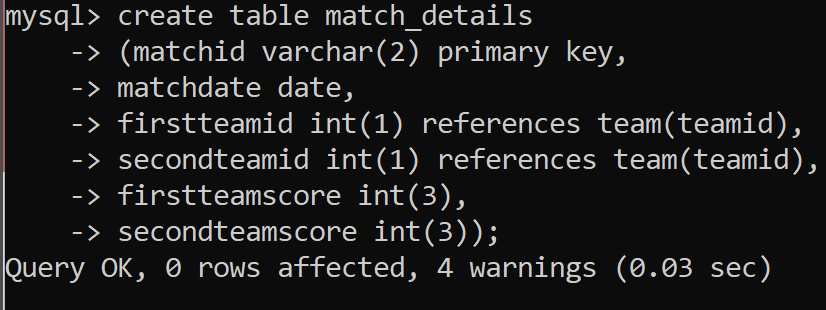
-> matchdate date,

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

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

-> firstteamscore int(3),

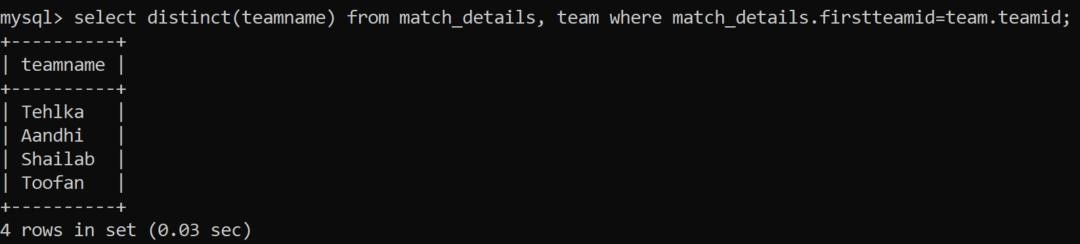
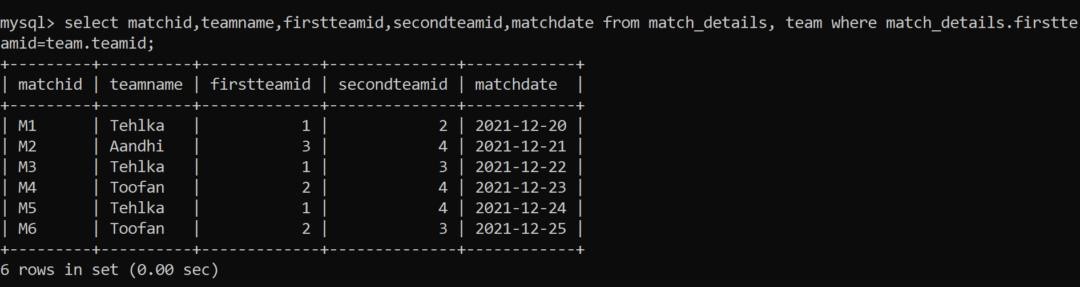
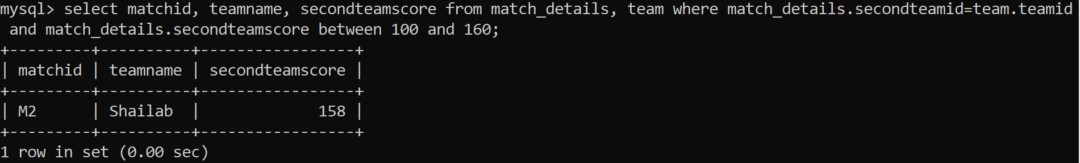
-> secondteamscore int(3));



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:

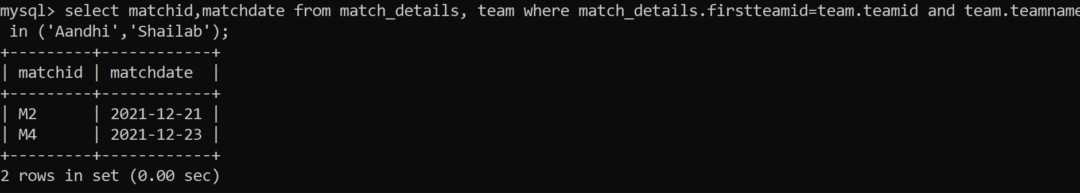
1. select match\_details.matchid, match\_details.firstteamid, team.teamname,match\_details.firstteamscore from match\_details, team where match\_details.firstteamid = team.teamid and match\_details.first
2. select match\_details.matchid, match\_details.firstteamid, team.teamname,match\_details.firstteamscore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







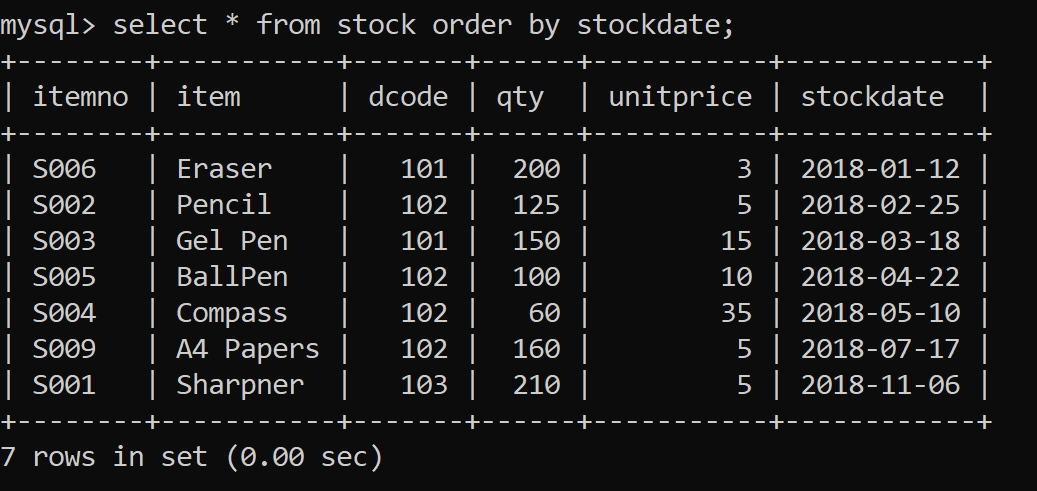
1. Consider the following table and write the queries:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| itemno | item | dcode | qty | unitprice | stockdate |
| S005 | Ballpen | 102 | 100 | 10 | 2018/04/22 |
| S003 | Gel Pen | 101 | 150 | 15 | 2018/03/18 |
| S002 | Pencil | 102 | 125 | 5 | 2018/02/25 |
| S006 | Eraser | 101 | 200 | 3 | 2018/01/12 |
| 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.

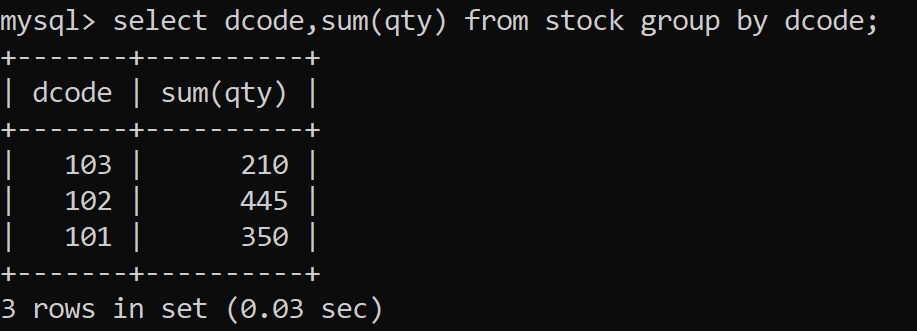
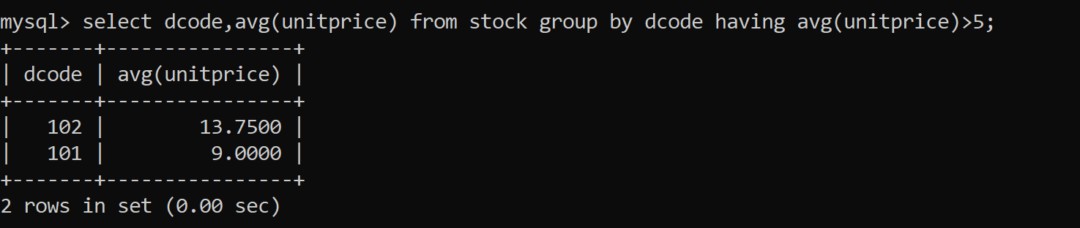
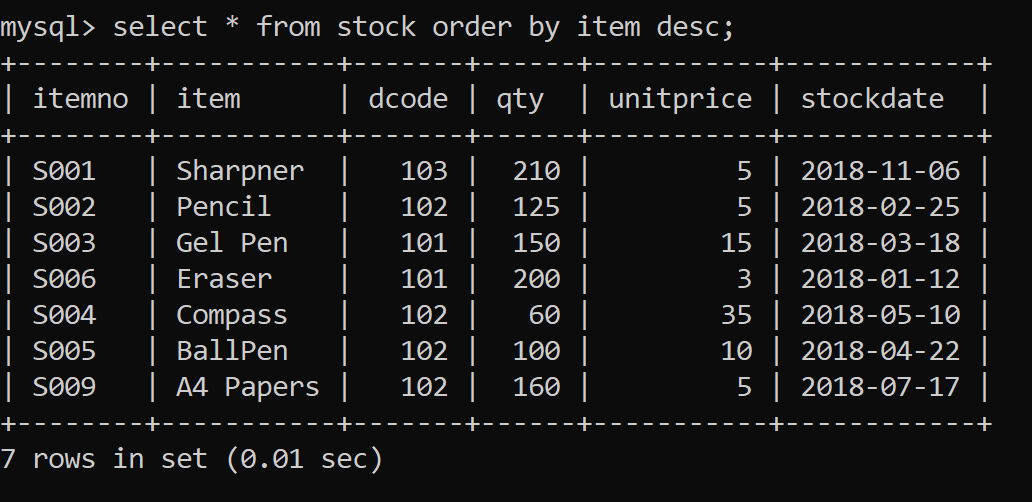
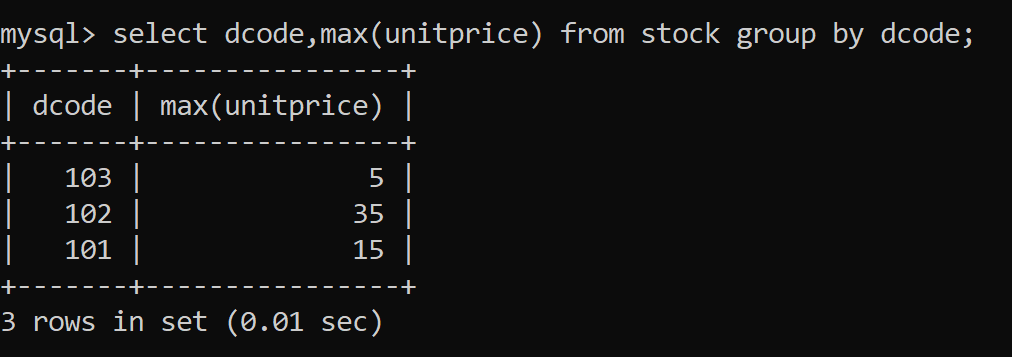
### Answers:

1. select \* from stock order by stockdate;





1. select dcode,max(unitprice) from stock group by code;



1. select \* from stock order by item desc;
2. select dcode,avg(unitprice) from stock group by dcode having avg(unitprice)>5;
3. select dcode,sum(qty) from stock group by dcode;

CS PRACTICAL RECORD FILE | Downloaded from [www.tutorialaicsip.com](http://www.tutorialaicsip.com/) | Page 22



1. Write a MySQL connectivity program in Python to

**PART C PYTHON DATABASE CONNECTIVITY**

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

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

### Answers:

1. using mysqlconnector



import mysql.connector as ms db=ms.connect(host="localhost",user="root",passwd="root",datab ase='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")

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")

### Output:

