**ACKNOWLEDGEMENT**

**I extend my heartfelt gratitude to Mr. Christy Thomas for their invaluable guidance and unwavering support throughout the completion of this computer science practical file. Mr. Thomas’s expertise, patience, and dedication to imparting knowledge have been instrumental in shaping my understanding of the subject.**

**Their commitment to our learning journey is evident through their tireless efforts in explaining intricate concepts, offering additional resources, and providing constructive feedback that has immensely enriched the quality of this practical file. Mr. Thomas's encouragement and willingness to address our queries have significantly contributed to our academic growth.**

**Moreover, their mentorship has transcended the confines of this practical file, leaving a lasting impact on my approach to problem-solving and critical thinking. Their guidance has not only aided in fulfilling the requirements of this assignment but has also ignited a deeper interest in exploring the diverse realms of computer science.**

**I am profoundly grateful for Mr. Christy Thomas’s unwavering support and dedication to our academic development. Their commitment to fostering a stimulating learning environment has been truly inspirational and has left an indelible mark on my educational journey. I am privileged to have had the opportunity to learn under their guidance.**

**INDEX**

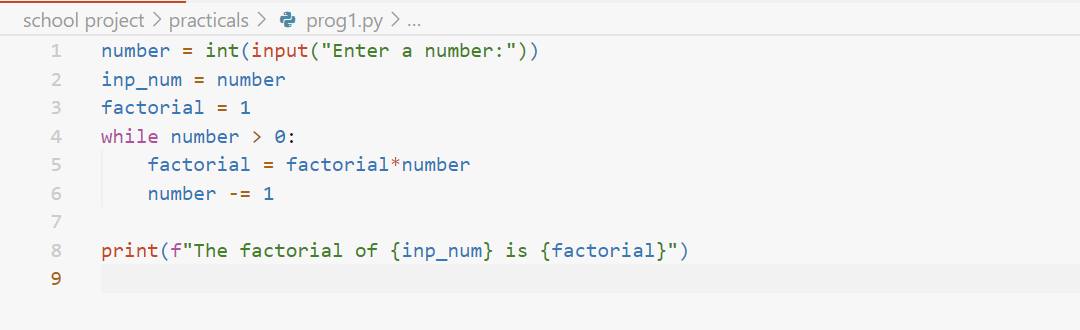
|  |  |  |  |
| --- | --- | --- | --- |
| Sr. No. | Exercise Title | Date | Teacher’s  signature |
| 1 | Program to find the factorial. |  |  |
| 2 | Program to check if a number is prime or not. |  |  |
| 3 | 1. Program to find sum of a list recursively. 2. Program to calculate nth term of Fibonacci series. |  |  |
| 4 | Program to search any word in given string or sentence. |  |  |
| 5 | Program to read and display file content line by line with each word  separated by #. |  |  |
| 6 | Program to read content of file and display the total number of  consonants, uppercase, vowels and lower case characters |  |  |
| 7 | Program to create file file to add Roll No and Name and search for  the given roll number. |  |  |
| 8 | Program to create binary file to store Roll No, Name and Marks and  update marks of entered roll number |  |  |
| 9 | Program to create a CSV file and store empno, name, salary and  search any empno and display name, salary and print appropriate message. |  |  |
| 10 | Program to generate random number 1-6, simulating a dice. |  |  |
| 11 | Write a function that takes amount in dollars and dollar to rupee  conversion price and returns the amount converted to rupees. |  |  |
| 12 | Program to get item details for multiple item from the user and  create a csv file by writing all the item details in one go. |  |  |
| 13 | Creating a simple username password checker using CSV file. |  |  |
| 14 | Program to implement Stack in Python using list. |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| 15 | Program to take 10 sample phishing emails and find the most  common occurring words. |  |  |
| 16 | Program to create 21 Stick game so that computer always wins. |  |  |
| 17 | Program to connect to a database and store record of employee  and display records |  |  |
| 18 | Program to connect with database and search employee number in  table and display record, if employee number is not found display appropriate message. |  |  |
| 19 | Program to connect with database and update the employee record  of entered emp. No. |  |  |
| 20 | Program to connect with database and delete the record of entered employee number. |  |  |
| 21 | MySQL Query Set 1 |  |  |
| 22 | MySQL Query Set 2 |  |  |
| 23 | MySQL Query Set 3 |  |  |
| 24 | MySQL Query Set 4 |  |  |
| 25 | MySQL Query Set 5 |  |  |

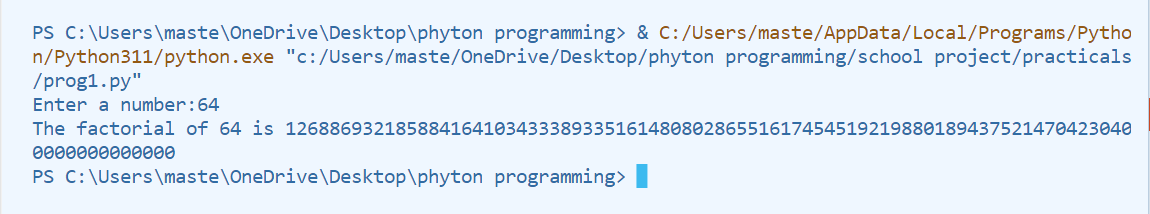
**PRACTICAL EXERCISE-1A**

**Q: Write a program to find the factorial**

**PROGRAM: -**

****

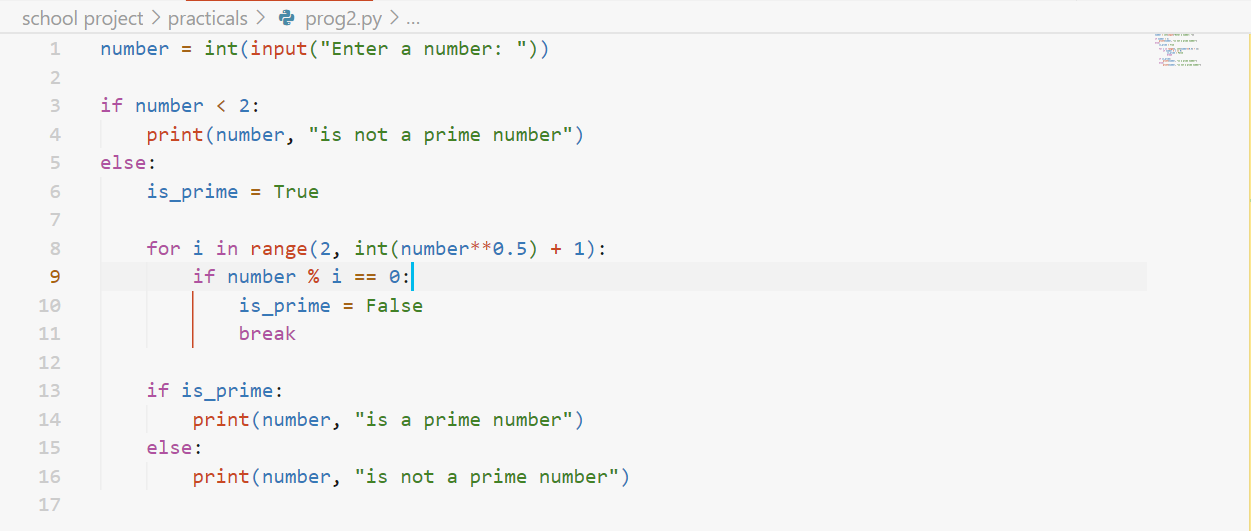
**OUTPUT: -**

****

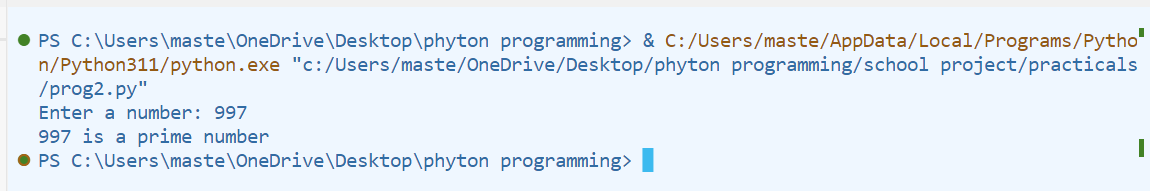
**PRACTICAL EXERCISE-1B**

**Q: Write a program to check if a number is prime or not prime**

**PROGRAM: -**

****

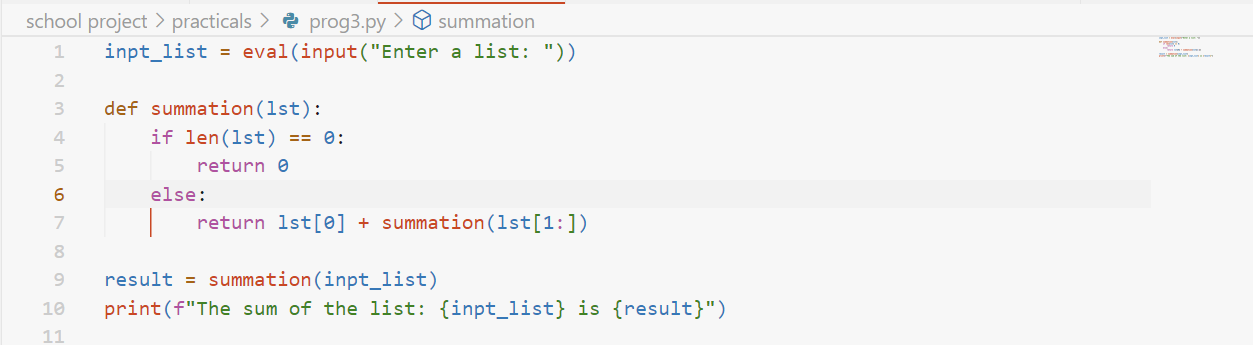
**OUTPUT: -**

****

**PRACTICAL EXERCISE-2**

**Q: Write a program to find the sum of a list recursively**

**PROGRAM: -**

****

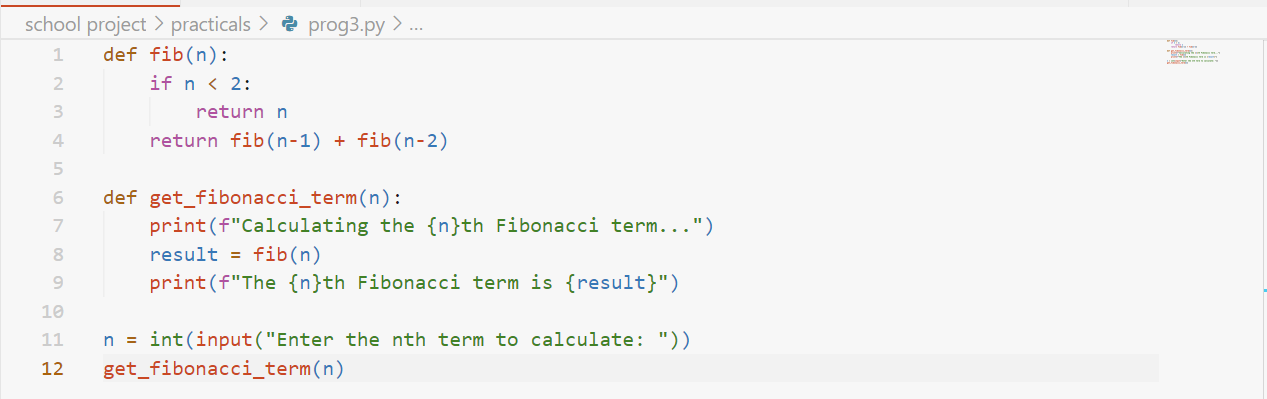
**OUTPUT: -**

****

**PRACTICAL EXERCISE-3**

**Q: Write a program to calculate the nth term of Fibonacci series.**

**PROGRAM: -**

****

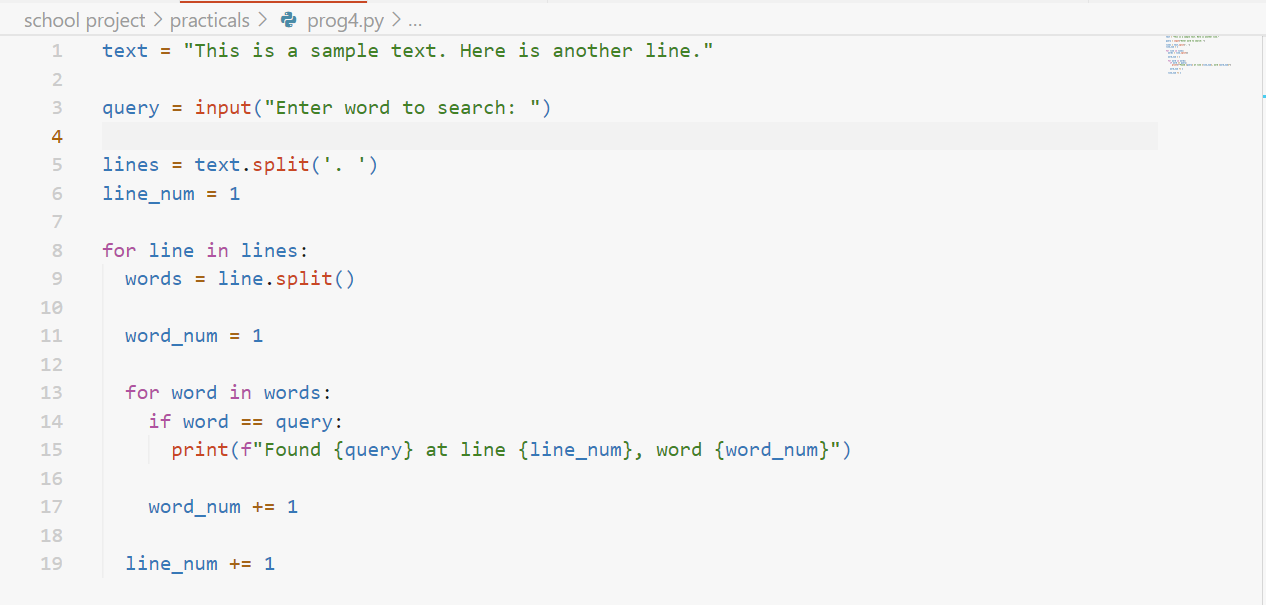
**OUTPUT: -**

****

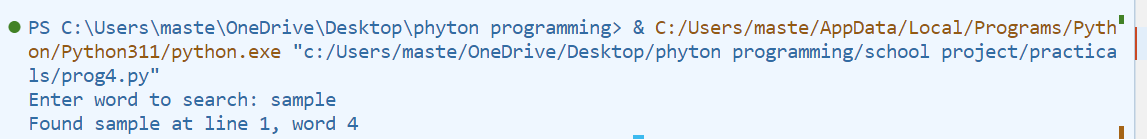
**PRACTICAL EXERCISE-4**

**Q: Write a program to search any word in given string or sentence**

**PROGRAM: -**

****

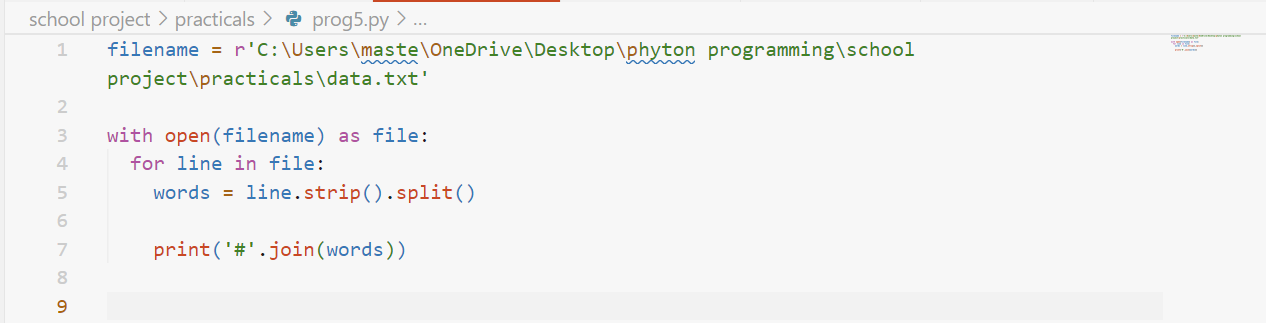
**OUTPUT: -**

****

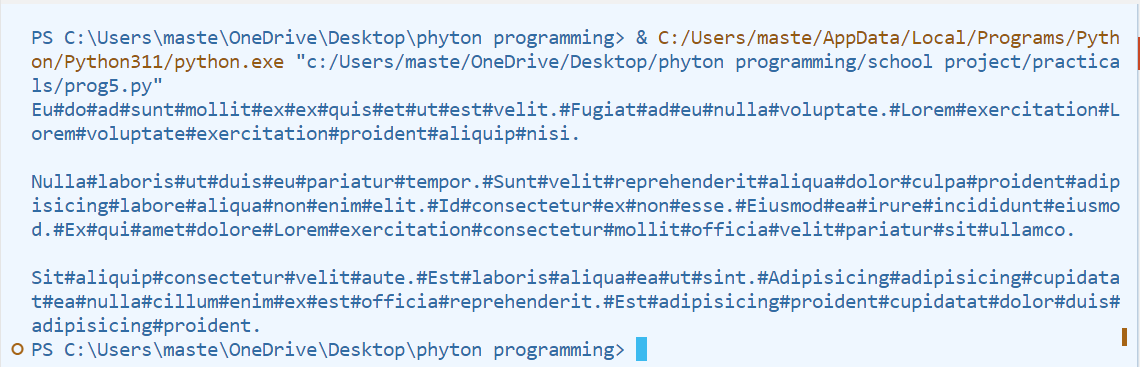
**PRACTICAL EXERCISE-5**

**Q: Write a program to read and display file content line by line with each word separated by #**

**PROGRAM: -**

****

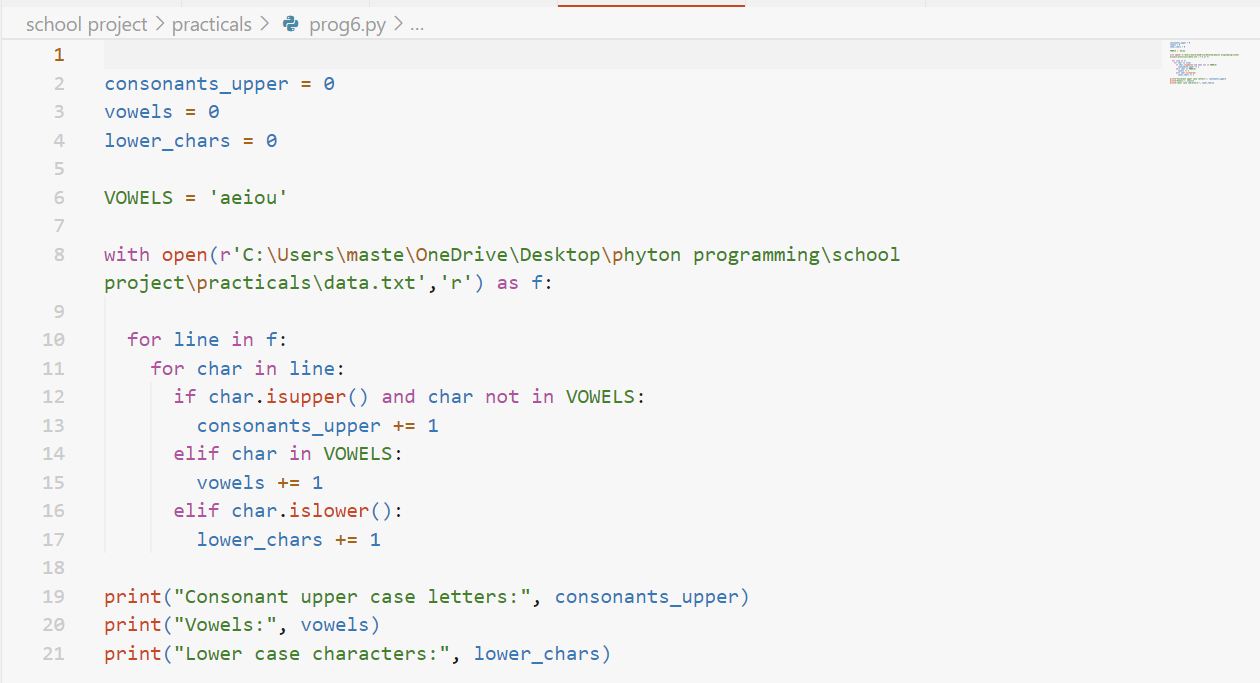
**OUTPUT: -**

****

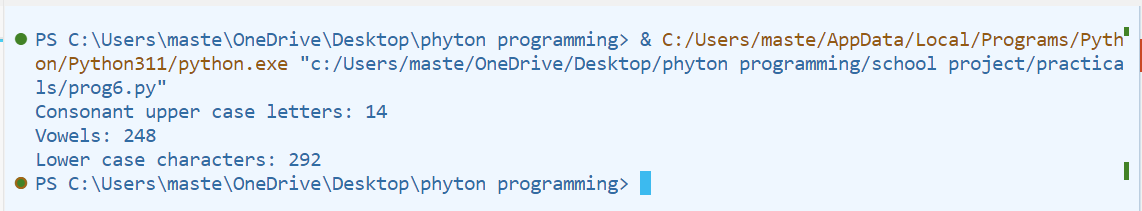
**PRACTICAL EXERCISE-6**

**Q: Write a program to read the content of a file and display the total number of consonants, uppercase, vowels and lowercase characters.**

**PROGRAM: -**

****

**OUTPUT: -**

****

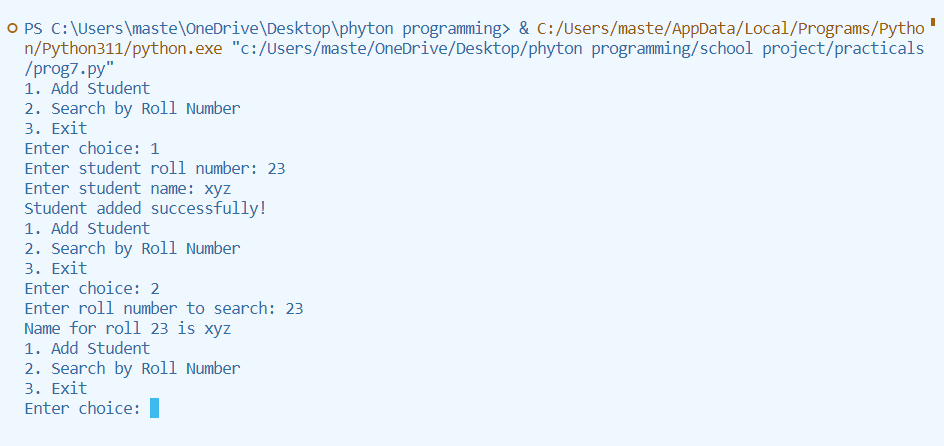
**PRACTICAL EXERCISE-7**

**Q: Create a binary list to store roll number and name and create a search function for searching name using role number.**

**PROGRAM: -**

****

**OUTPUT: -**

****

**PRACTICAL EXERCISE-8**

**Q Write a program to create a binary file to store roll number marks and name and write a function to update marks.**

**PROGRAM: -**

****

****

**OUTPUT: -**

****

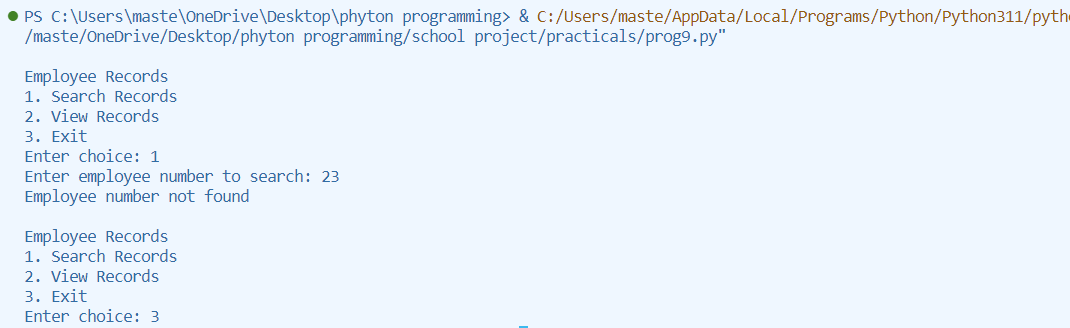
**PRACTICAL EXERCISE-9**

**Q: Write a program to create a csv file, storing employee number and salary, and also write a search function.**

**PROGRAM: -**

****

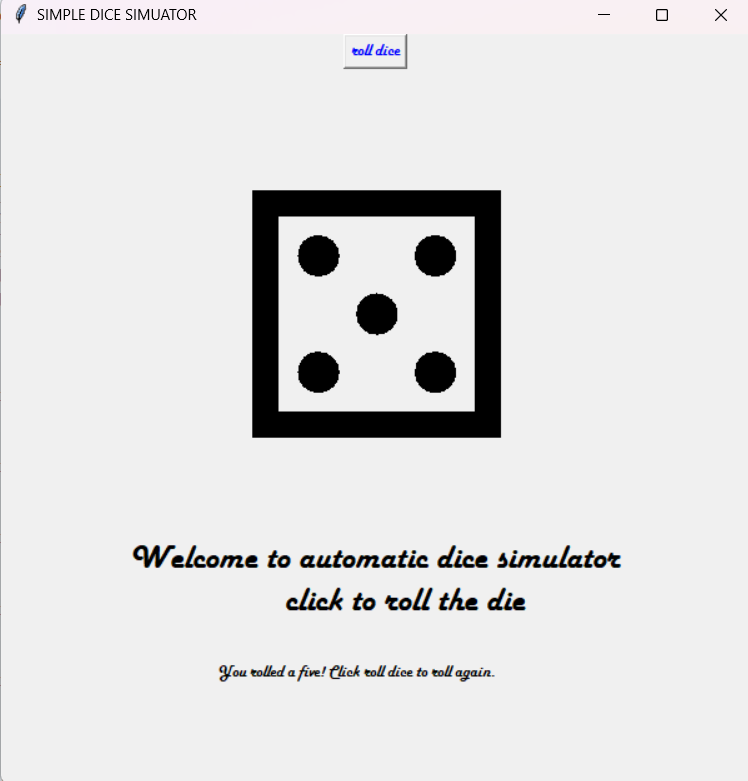
**OUTPUT: -**

****

**PRACTICAL EXERCISE-10**

**Q: Write a program to generate random number one to 6 and simulator dice.**

**PROGRAM: -**

**OUTPUT: -**

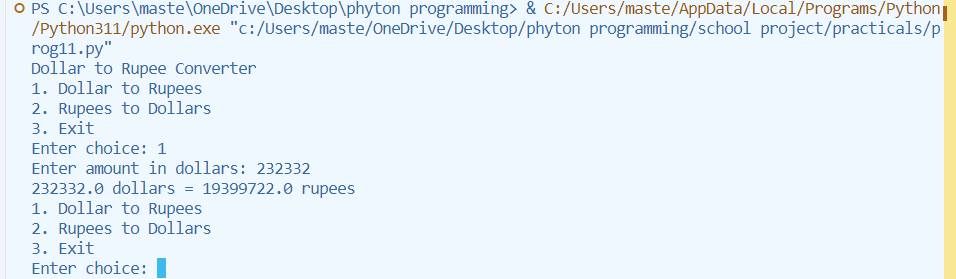
**PRACTICAL EXERCISE-11**

**Q: Write a program for currency conversion from dollar to rupee.**

**PROGRAM: -**

****

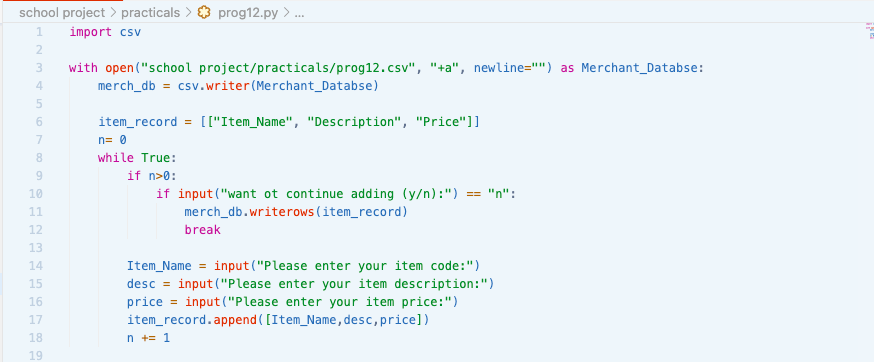
**OUTPUT: -**

****

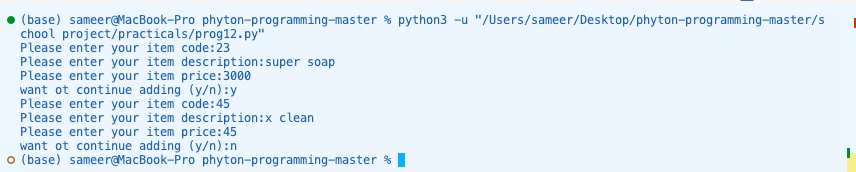
**PRACTICAL EXERCISE-12**

**Q: Program to get item details for multiple item from the user and create a csv file by writing all the item details in one go.**

**PROGRAM: -**

****

**OUTPUT: -**

****

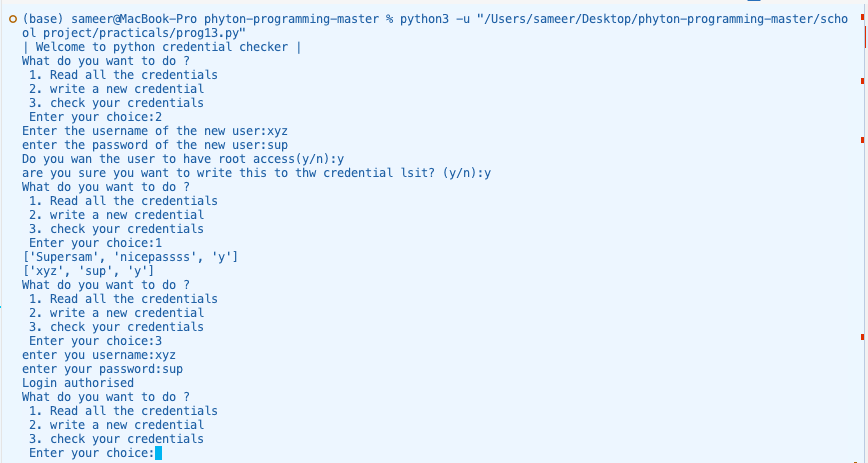
**PRACTICAL EXERCISE-13**

**Q: Creating a simple username password checker using CSV file.**

**PROGRAM: -**

****

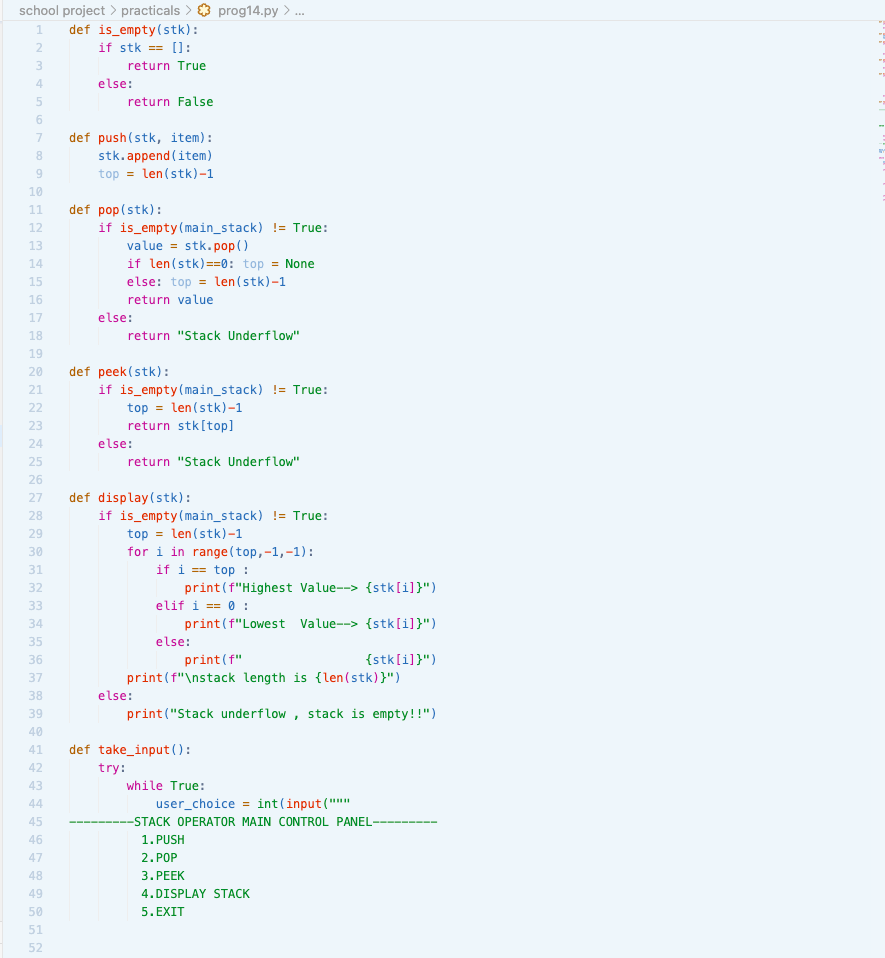
**OUTPUT: -**

****

**PRACTICAL EXERCISE-14**

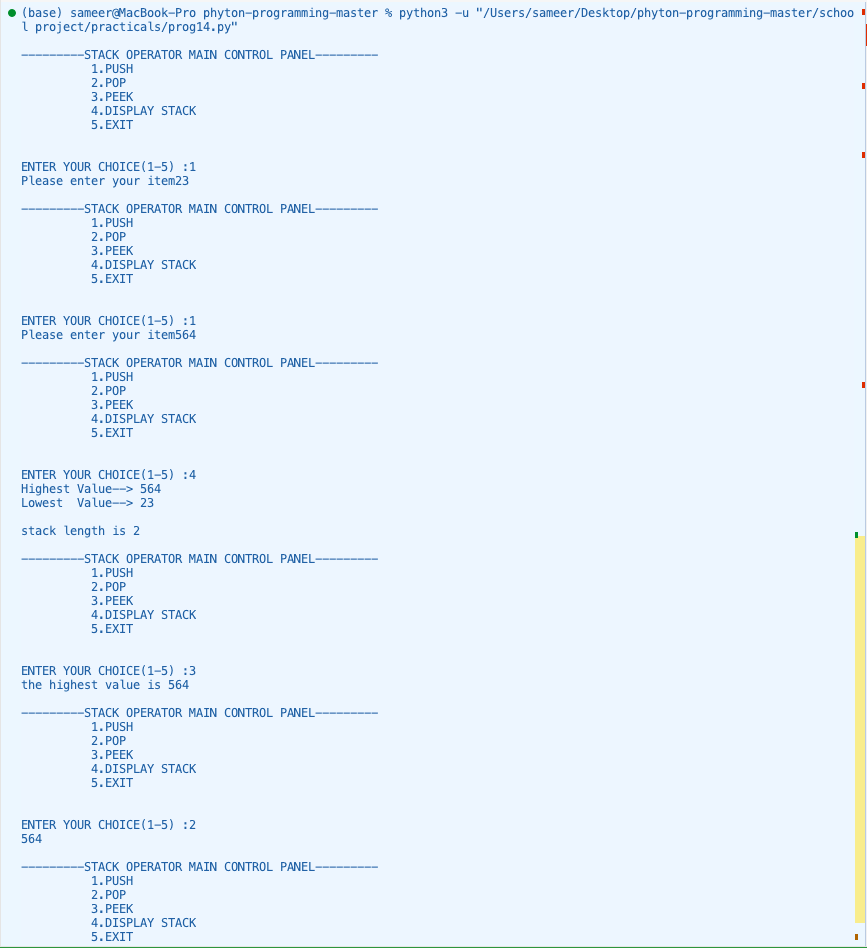
**Q: Program to implement Stack in Python using list.**

**PROGRAM: -**

****

****

**OUTPUT: -**

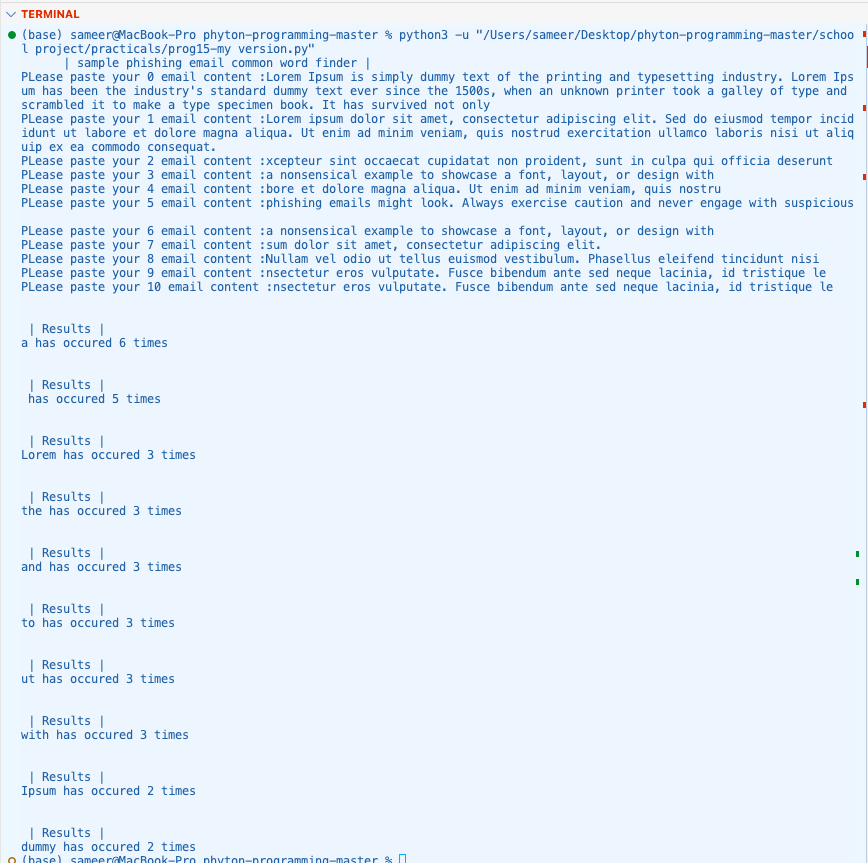
****

**PRACTICAL EXERCISE-15**

**Q: Program to take 10 sample phishing emails and find the most common occurring words.**

**PROGRAM: -**

****

**OUTPUT: -**

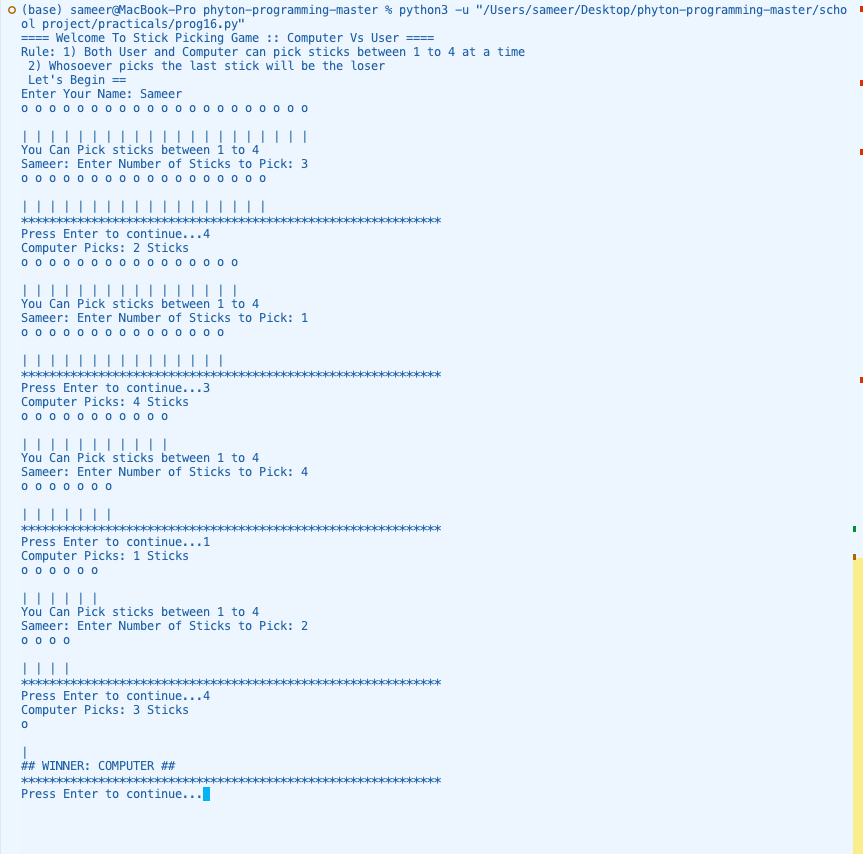
**PRACTICAL EXERCISE-16**

**Q: Program to create 21 Stick game so that computer always wins.**

**PROGRAM: -**

****

**OUTPUT: -**

****

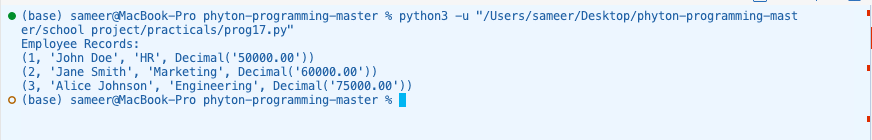
**PRACTICAL EXERCISE-17**

**Q: Program to connect to a database and store record of employee and display records**

**PROGRAM: -**

****

**OUTPUT: -**

****

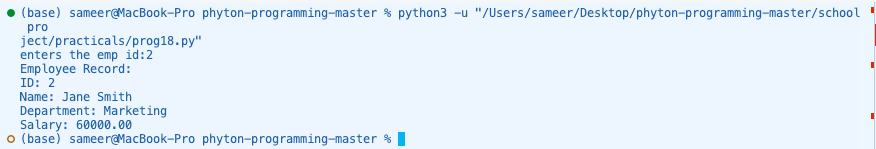
**PRACTICAL EXERCISE-18**

**Q: Program to connect with database and search employee number in table and display record, if employee number is not found display appropriate message.**

**PROGRAM: -**

****

**OUTPUT: -**

****

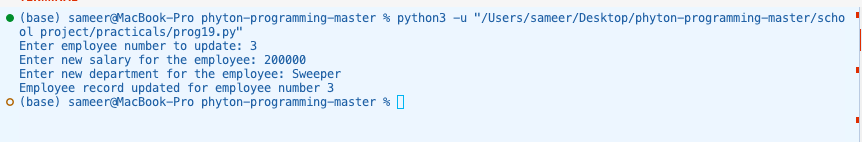
**PRACTICAL EXERCISE-19**

**Q: Program to connect with database and update the employee record of entered emp. No.**

**PROGRAM: -**

****

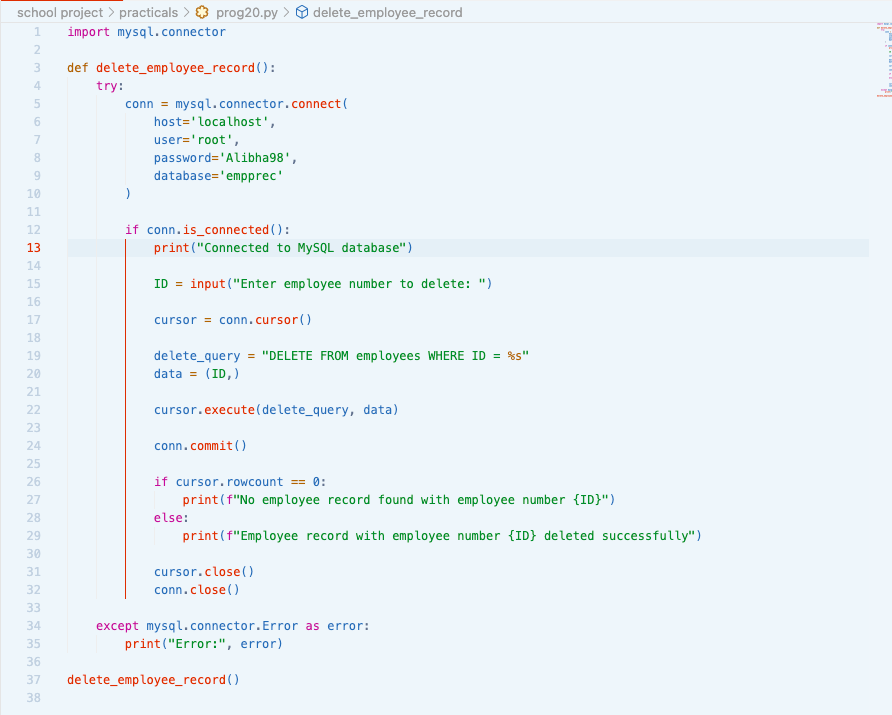
**OUTPUT: -**

****

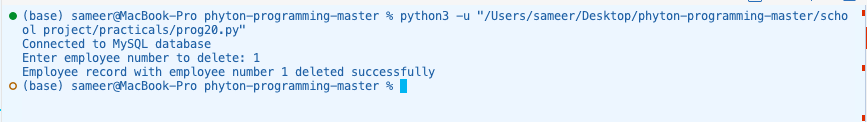
**PRACTICAL EXERCISE-20**

**Q: Program to connect with database and delete the record of entered employee number.**

**PROGRAM: -**

****

**OUTPUT: -**

****

**PRACTICAL EXERCISE-21**

**Q: Write SQL commands for the following on the basis of given table MOV:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| No | Title | Type | Rating | Stars | Qty | Price |
| 1 | Gone with the Wind | Drama | G | Gable | 4 | 39.95 |
| 2 | Friday the 13th | Horror | R | Jason | 2 | 69.95 |
| 3 | Top Gun | Drama | PG | Cruise | 7 | 49.95 |
| 4 | Splash | Comedy | PG13 | Hanks | 3 | 29.95 |
| 5 | Independence Day | Drama | R | Turner | 3 | 19.95 |
| 6 | Risky Business | Comedy | R | Cruise | 2 | 44.95 |
| 7 | Cocoon | Scifi | PG | Ameche | 2 | 31.95 |
| 8 | Crocodile Dundee | Comedy | PG13 | Harris | 23 | 69.95 |
| 9 | 101 Dalmatians | Comedy | G |  | 3 | 59.95 |
| 10 | Tootsie | Comedy | PG | Hoffman | 1 | 29.95 |

(a) Display a list of all movies with Price over 20 and sorted by Price.

>> SELECT \* FROM MOV WHERE Price > 20 ORDER BY Price;

(b) Display all the movies sorted by QTY in decreasing order.

>>> SELECT \* FROM MOV ORDER BY Qty DESC;

(c) Display a report listing a movie number, current value and replacement value for each movie in the above table. Calculate the replacement value for all movies as: QTY Price 1.15.

>>> SELECT

No AS Movie\_Number,

Qty \* Price AS Current\_Value,

Qty \* Price \* 1.15 AS Replacement\_Value

FROM MOV;

**PRACTICAL EXERCISE-22**

**Q: Write SQL commands for the following on the basis of given table Teacher:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No.** | **Name** | **Age** | **Department** | **Date of join** | **Salary** | **Sex** |
| 1 | Jugal | 34 | Computer | 10/01/97 | 12000 | M |
| 2 . | Sharmila | 31 | History | 24/03/98 | 20000 | F |
| 3 . | Sandeep | 32 | Maths | 12/12/96 | 30000 | M |
| 4 . | Sangeeta | 35 | History | 01/07/99 | 40000 | F |
| 5 . | Rakesh | 42 | Maths | 05/09/97 | 25000 | M |
| 6 . | Shyam | 50 | History | 27/06/98 | 30000 | M |
| 7 . | Shiv Om | 44 | Computer | 25/02/97 | 21000 | M |
| 8 . | Shalakha | 33 | Maths | 31/07/97 | 20000 | F |

(a) To show all information about the teacher of history department

>>> SELECT \* FROM Teacher WHERE Department = 'History';

(b) To list the names of female teachers who are in Hindi department

>>> SELECT Name FROM Teacher WHERE Department = 'Hindi' AND Sex = 'F';

(c) To list names of all teachers with their date of joining in ascending order.

>>> SELECT Name, [Date of join] FROM Teacher ORDER BY [Date of join] ASC;

**PRACTICAL EXERCISE-23**

**Q: Write the SQL queries for the given two tables job and employee:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **JOBID** | **JOBTITLE** | **SALARY** |  | **EMPLOYEEID** | **NAME** | **SALES** | **JOBID** |
| 101 | President | 200000 |  | E1 | SUMIT SINHA | 1100000 | 102 |
| 102 | Vice President | 125000 |  | E2 | VIJAY SINGH TOMAR | 1300000 | 101 |
| 103 | Administration Assistant | 80000 |  | E3 | AJAY RAJPAL | 1400000 | 103 |
| 104 | Accounting Manager | 70000 |  | E4 | MOHIT RAMNANI | 1250000 | 102 |
| 105 | Accountant | 65000 |  | E5 | SHAILJA SINGH | 1450000 | 103 |
| 106 | Sales Manager | 80000 |  |  |  |  |  |

(i) To display employee ids, names of employees, job ids with corresponding job titles.

>>> SELECT EMPLOYEEID, NAME, EMPLOYEE.JOBID, JOBTITLE

FROM employee

JOIN job ON EMPLOYEE.JOBID = JOB.JOBID;

(ii) To display names of employees, sales and corresponding job titles who have achieved sales more than 1300000.

>>> SELECT NAME, SALES, JOBTITLE

FROM employee

JOIN job ON EMPLOYEE.JOBID = JOB.JOBID

WHERE SALES > 1300000;

(iii) To display names and corresponding job titles of those employees who have 'SINGH' (anywhere) in their names.

>>> SELECT NAME, JOBTITLE

FROM employee

JOIN job ON EMPLOYEE.JOBID =JOB.JOBID

WHERE NAME LIKE '%SINGH%';

(iv) Identify foreign key in the table EMPLOYEE.

The foreign key in the table 'EMPLOYEE' is the column **JOBID** as it references the **JOBID** column in the 'job' table.

(e) Write SQL. command to change the JOBID to 104 of the EMPLOYEE with ID as E4 in the table 'EMPLOYEE’

UPDATE employee

SET JOBID = '104'

WHERE EMPLOYEEID = 'E4';

**PRACTICAL EXERCISE-24**

**Q: Consider the following tables Employee and Salary, Write SQL commands for the statements (i) to (iv) and give outputs for SQL. queries (v) to (vii)**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Eid | Name | Depid | Qualification | Sec |  | Eid | Basic | D.A. | HRA | Bonus |
| 1 | Deepali Gupta | 101 | MCA | F |  | 1 | 6000 | 2000 | 2300 | 200 |
| 2 | Rajat Tyagi | 101 | BCA | M |  | 2 | 2000 | 300 | 300 | 30 |
| 3 | Hari Mohan | 102 | B.A. | M |  | 3 | 1000 | 300 | 300 | 40 |
| 4 | Harry | 102 | M.A. | M |  | 4 | 1500 | 390 | 490 | 30 |
| 5 | Sumit Mittal | 103 | B.Tech . | M |  | 5 | 8000 | 900 | 900 | 80 |
| 6 | Jyoti | 101 | M.Tech . | F |  | 6 | 10000 | 300 | 490 | 89 |

**Table: Employee Table: Salary**

(i) To display the frequency of employees department wise.

>>> SELECT Depid, COUNT(\*) AS Frequency

FROM Employee

GROUP BY Depid;

(ii) To list the names of those employees only whose name starts with 'H'

>>> SELECT Name

FROM Employee

WHERE Name LIKE 'H%';

(iii) To add a new column in salary table. The column name is Total Sal.

>>>ALTER TABLE Salary

ADD COLUMN TotalSal DECIMAL(10, 2);

(iv) To store the corresponding values in the Total Sal column.

UPDATE Salary

SET TotalSal = Basic + "D.A." + HRA + Bonus;

(v) Select max(Basic) from Salary where Bonus > 40:

Output:

|  |  |
| --- | --- |
| MAX(Basic) ---------- 10000 |  |

(vi) Select count(\*) from Employee group by Sex;

Output:

|  |  |
| --- | --- |
| Sex | Count |
| F | 2 |
| M | 4 |

(vii) Select Distinct Depid from Employee:

|  |
| --- |
| Depid |
| 101 |
| 102 |
| 103 |

Output:

**PRACTICAL EXERCISE-25**

**Q: With reference to following relations PERSONAL. and JOB answer the questions that follow: Create following tables such that Empro and Sno are not null and unique, date of birth is after "12-Jan-1960', name is never blank, Area and Native place is valid, hobby, dept is not empty, salary is between 4000 and 10000.**

**Table : Personal**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Empno | Name | Dobirth | Native - place | Hobby |
| 123 | Amit | 23 - Jan - 1965 | Delhi | Music |
| 127 | Manoj | 12 - dec - 1976 | Mumbai | Writing |
| 124 | Abhai | 11 - aug - 1975 | Allahabad | Music |
| 125 | Vinod | 04 - apr - 1977 | Delhi | Sports |
| 128 | Abhay | 10 - mar - 1974 | Mumbai | Gardening |
| 129 | Ramesh | 28 - oct - 1981 | Pune | Sports |

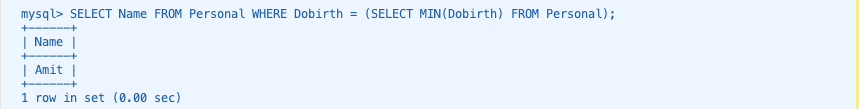
**Table : Job**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sno | Area | App date | Salary | Retd date | Dept |
| 123 | Agra | 25 - jan - 2006 | 5000 | 25 - jan - 2026 | Marketing |
| 127 | Mathura | 22 - dec - 2006 | 6000 | 22 - dec - 2026 | Finance |
| 124 | Agra | 19 - aug - 2007 | 5500 | 19 - aug - 2027 | Marketing |
| 125 | Delhi | 14 - apr - 2004 | 8500 | 14 - apr - 2018 | Sales |
| 128 | Pune | 13 - mar - 2008 | 7500 | 13 - mar - 2028 | Sales |

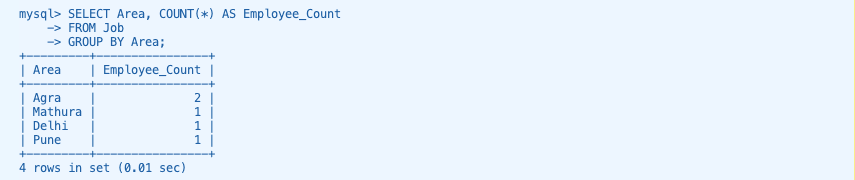
(a) Show empno, name, and salary of those who have Sports as a hobby.



(b) Show the name of the eldest employee.



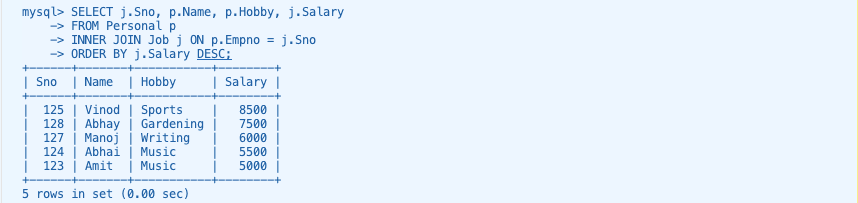
(c) Show the number of employees area-wise.



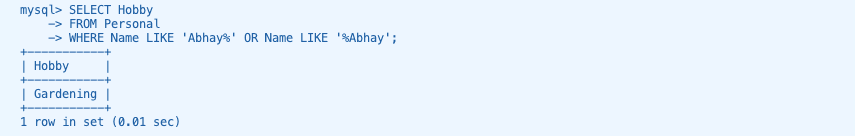
(d) Show the youngest employees from each Native place.



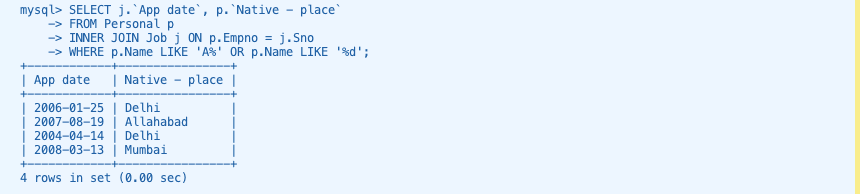
(e) Show Sno, Name, Hobby, and Salary in descending order of Salary.



(f) Show the hobbies of those whose name pronounces as "Abhay".



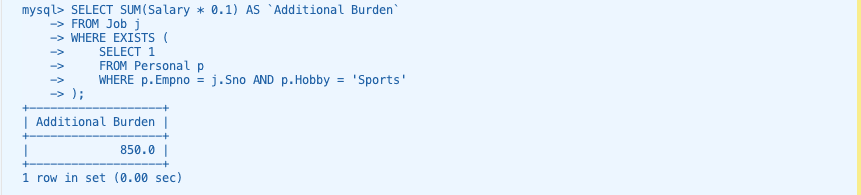
(g) Show the appointment date and native place of those whose name starts with 'A' or ends in 'd'.



(h) Show the salary expense with suitable column heading of those who shall retire after 20-Jan-2006.



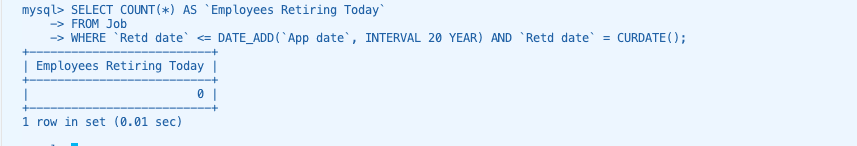
(i) Show the additional burden on the company in case the salary of employees having a hobby as sports is increased by 10%.



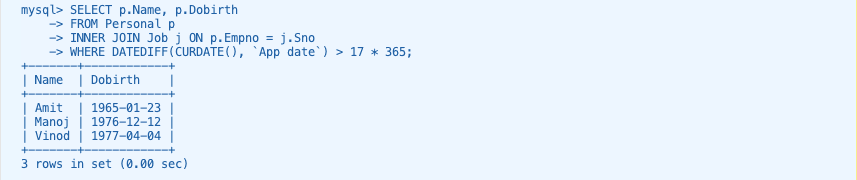
(j) Show the hobby of which there are 2 or more employees.



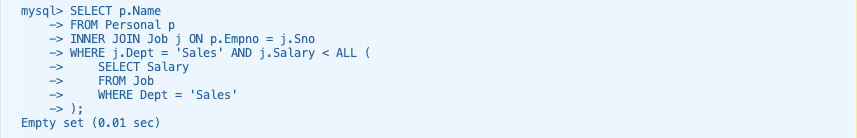
(k) Show how many employees shall retire today if the maximum length of service is 20 years.



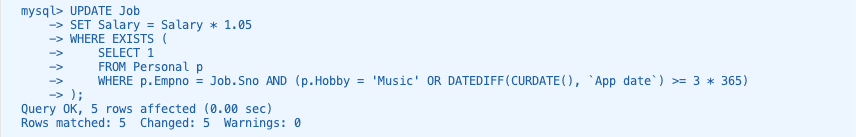
(l) Show those employees' names and dates of birth who have served more than 17 years as of the date.



(m) Show names of those who earn more than all of the employees of the Sales dept.

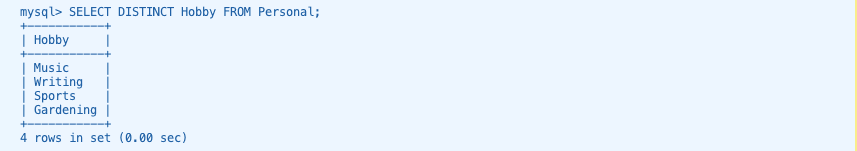


(n) Increase the salary of the employees by 5% of their present salary with the hobby as Music or who have completed at least 3 years of service.



(o) Write the output of:

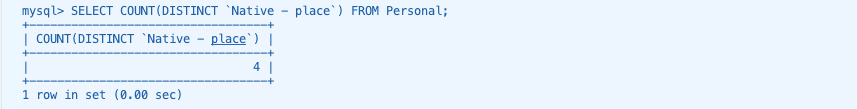
* Select distinct hobby from personal;



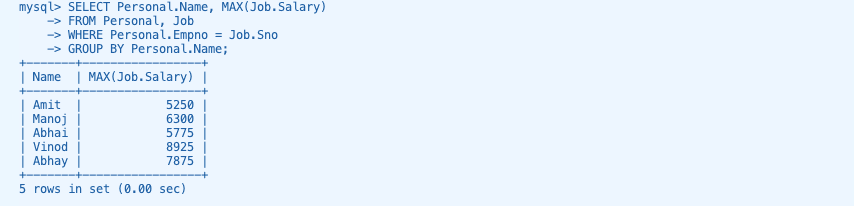
* Select avg(salary) from personal, job where Personal Empno Job Sno and Area in ('Agra', 'Delhi');



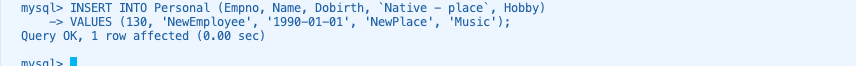
* Select count(distinct Native\_place) from personal.



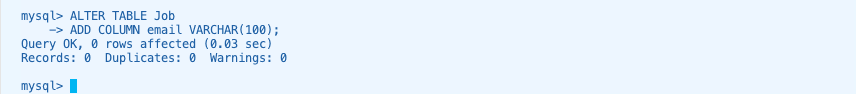
* Select name, max(salary) from Personal, Job where Personal. Empno Job Sno;



(p) Add a new tuple in the table Personal essentially with a hobby as Music.



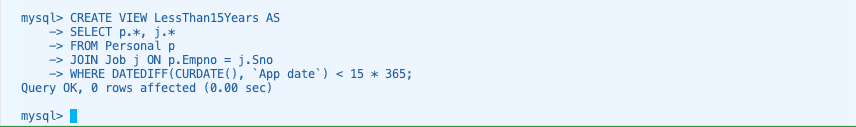
(q) Insert a new column email in the Job table.



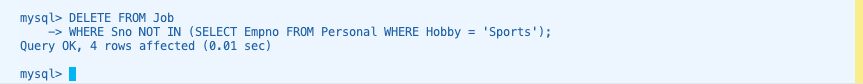
(r) Create a table with values of columns empno, name, and hobby.



(s) Create a view of Personal and Job details of those who have served less than 15 years.



(t) Erase the records of employees from the Job table whose hobby is not Sports.



(u) Remove the table Personal.

