DATABASE MANAGEMENT SYSTEM LAB

Payroll System Project

Team members

Aadarsh S. (1)

Akshay G. (4)

Bharath S. (18)

Lakshmi Narayanan (36)

Rohan Johnson (57)

Rohit Menon (58)

Royal Seby (60)

Shruti Maria Shibu (62)

Tessa Soji Cherian (65)

Yohan Jose (68)

THEORETICAL BACKGROUND

A payroll system is a computerized system that manages and processes employee compensation, including wages, salaries, bonuses, taxes, and deductions. It allows companies to automate the process of calculating employee payments and maintaining accurate financial records. A well-designed payroll system can help companies streamline their payroll processes, reduce errors, and improve the accuracy and timeliness of employee payments. It can also help companies avoid legal and financial penalties associated with payroll errors and non-compliance with tax laws.

A MySQL connector is a library or module that provides an interface for Python to interact with a MySQL database. There are different MySQL connectors available for Python, including the official MySQL Connector/Python, PyMySQL, and mysql-connector-python. The steps for connecting Python and MySQL using a MySQL connector typically involve the following:

- Install the MySQL connector for Python: This involves downloading and installing the MySQL connector module or library for Python, depending on the chosen MySQL connector.

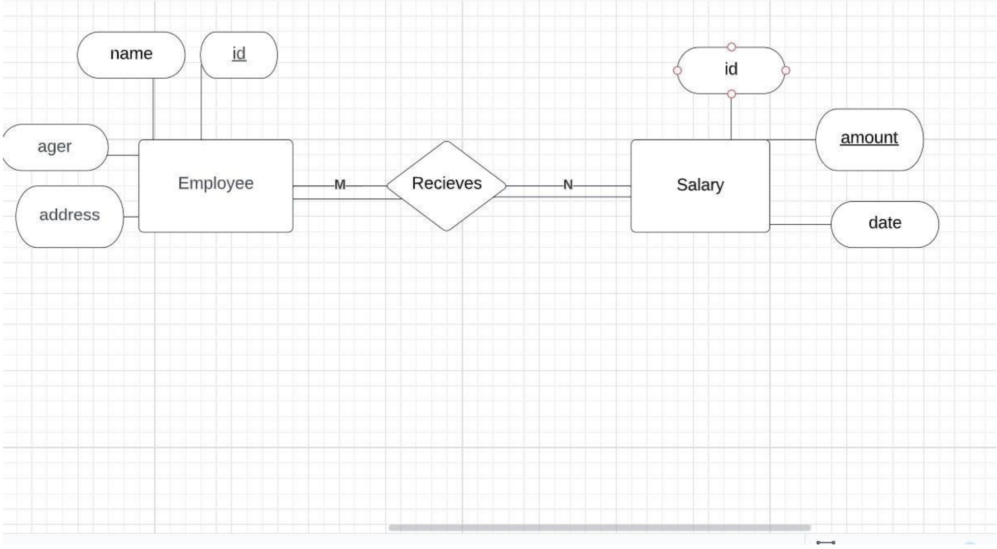
- Establish a connection: This involves creating a connection object that connects to the MySQL database server using the necessary credentials, including the hostname, username, password, and database name.

- Execute SQL queries: This involves using the connection object to execute SQL queries that retrieve, insert, update, or delete data from the MySQL database.

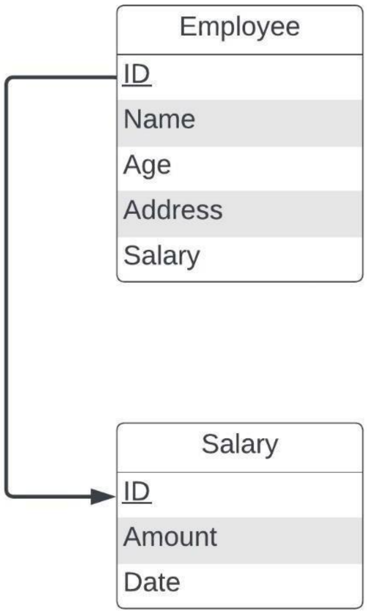
- Close the connection: This involves closing the connection object to free up system resources and ensure the security of the MySQL database.

The applications used in our project are Python and MySQL Version 8.0.32.

ER DIAGRAM



REATIONAL SCHEMA



SOURCE CODE

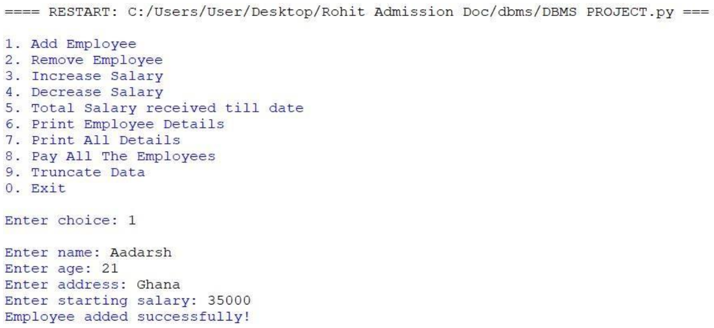
```
import mysql.connector as sqct mydb = sqct.connect (host="localhost", user="root",

password="12345", database="payroll") cur = mydb.cursor()

cur.execute("CREATE TABLE IF NOT EXISTS employee (id INT AUTO_INCREMENT PRIMARY KEY, name VARCHAR(255), age
INT, address VARCHAR(255), sal int)")

cur.execute("CREATE TABLE IF NOT EXISTS salary (id INT, amount INT, date DATE, FOREIGN KEY(id)

references employee(id))") # add a new employee

def add_employee():
   print()
   name = input("Enter name: ")     age
= input("Enter age: ")     address =
input("Enter address: ")
sal=int(input("Enter starting salary: "))
   sql = "INSERT INTO employee (name, age, address, sal) VALUES (%s, %s, %s,
%s)"     val = (name, age, address, sal)     cur.execute(sql, val)     mydb.commit
amount = sal
   cur.execute("SELECT LAST_INSERT_ID()")
   result = cur.fetchone()
emp_id = result[0]
   sql="INSERT INTO salary (id,amount,date) VALUES (%s, %s,
CURDATE())"   val  =  (emp_id,amount)   cur.execute(sql,   val)
mydb.commit()
   print("Employee added successfully!")
print()

# remove an employee
def remove_employee():
   print()
   id = input("Enter employee ID: ")
   cur.execute("DELETE FROM salary WHERE id = %s", (id,))
   sql = "DELETE FROM employee WHERE id =
%s"     val = (id,)     cur.execute(sql, val)
mydb.commit()
   print("Employee removed successfully!")
print()

# increase an employee's
salary def increase_salary():
print()
   id = input("Enter employee ID: ")     amount =
input("Enter amount to increase: ")
   sql = "INSERT INTO salary (id, amount, date) VALUES (%s, %s,
CURDATE())"     val = (id, amount)     cur.execute(sql, val)     mydb.commit()
   print("Salary increased successfully!")
# decrease an employee's salary
def decrease_salary():
   print()
   id = input("Enter employee ID: ")     amount = input("Enter amount to
decrease: ")     sql = "INSERT INTO salary (id, amount, date) VALUES (%s,
%s, CURDATE())"     val = (id, -int(amount))     cur.execute(sql, val)
mydb.commit()
   print("Salary decreased successfully!")
print()
```

```
# total salary received by an employee
def total_salary():
   print()
   id = input("Enter employee ID: ")
   sql = "SELECT SUM(amount) FROM salary WHERE id =
%s"     val = (id,)     cur.execute(sql, val)     result =
cur.fetchone()[0]     if result:
      print("Total salary received: " +
str(result))        print()     else:        print("No
salary received yet!")

# print an employee's details
def print_employee():
   id = input("Enter employee ID: ")     sql =
"SELECT * FROM employee WHERE id = %s"
   val = (id,)
cur.execute(sql, val)
result = cur.fetchone()
if result:        print()
      print("ID: " + str(result[0]))
print("Name: " + result[1])
print("Age: " + str(result[2]))
print("Address: " + result[3])
print("Salary: " + str(result[4]))
      print()     else:
print("Employee not found!")

# print all employee details
def print_all():
   cur.execute("SELECT * FROM
employee")     result = cur.fetchall()     for
row in result:
      print()        print("ID: " +
str(row[0]))        print("Name:
" + row[1])        print("Age: " +
str(row[2]))
print("Address: " + row[3])
      print("Salary: " + str(row[4]))
print()

# pay all the
employes def
pay_all():     print()
cur.execute("SELECT
id, sal FROM
employee")
   result = cur.fetchall()
for row in result:
emp_id = row[0]
sal = row[1]
      sql = "INSERT INTO salary (id, amount, date) VALUES (%s, %s, CURDATE())"
val = (emp_id, sal)
      cur.execute(sql, val)
mydb.commit()     print("All employees
paid successfully!")

# truncate all values in the
DB def trun():     print()
```

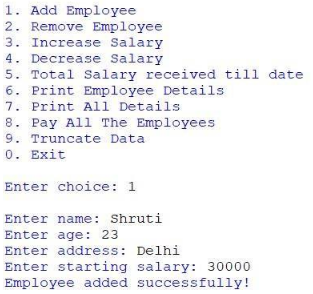
```
  x=input("Are You Sure to TRUNCATE All VALUES (Y/N) :")
if (x=="Y" or x=="y"):
    cur.execute("DROP table salary")
cur.execute("DROP table employee")
    cur.execute("CREATE TABLE IF NOT EXISTS employee (id INT AUTO_INCREMENT PRIMARY KEY, name
VARCHAR(255),    age INT, address VARCHAR(255), sal int)")        cur.execute("CREATE TABLE IF NOT EXISTS salary (id
INT, amount INT, date DATE, FOREIGN KEY(id) references employee(id))")        mydb.commit()        print("Tables
Truncated")    else:        print("Revoked by User")



while True:
  print()
  print("1. Add Employee")
print("2. Remove Employee")
print("3. Increase Salary")
print("4. Decrease Salary")
  print("5. Total Salary received till
date")    print("6. Print Employee
Details")    print("7. Print All Details")
print("8. Pay All The Employees")
print("9. Truncate Data")
  print("0. Exit")
  print()
  ch = input("Enter choice:
")    if ch == "1":
add_employee()    elif ch ==
"2":
remove_employee()    elif
ch == "3":
    increase_salary()
elif ch == "4":

decrease_salary()
elif ch == "5":
total_salary()    elif
ch == "6":

print_employee()
elif ch == "7":
print_all()    elif ch ==
"8":        pay_all()
  elif ch ==
"9":        trun()
elif ch == "0":
    print("Program Terminated By
User")        break    else:
    print("\nInvalid choice.\nPlease try again.")
```

OUTPUT TEST CASES

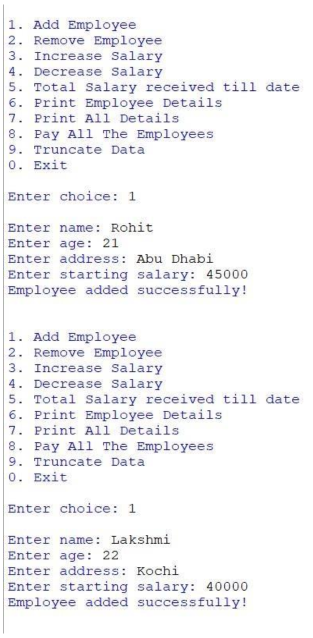
#Output for case 1



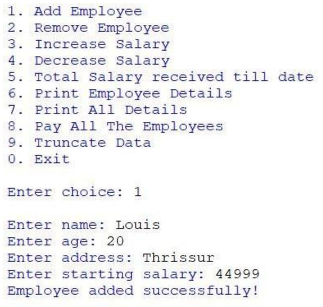
```
==== RESTART: C:/Users/User/Desktop/Rohit Admission Doc/dbms/DBMS PROJECT.py ===

1. Add Employee
2. Remove Employee
3. Increase Salary
4. Decrease Salary
5. Total Salary received till date
6. Print Employee Details
7. Print All Details
8. Pay All The Employees
9. Truncate Data
0. Exit

Enter choice: 1

Enter name: Aadarsh
Enter age: 21
Enter address: Ghana
Enter starting salary: 35000
Employee added successfully!
```



```
1. Add Employee
2. Remove Employee
3. Increase Salary
4. Decrease Salary
5. Total Salary received till date
6. Print Employee Details
7. Print All Details
8. Pay All The Employees
9. Truncate Data
0. Exit

Enter choice: 1

Enter name: Shruti
Enter age: 23
Enter address: Delhi
Enter starting salary: 30000
Employee added successfully!
```

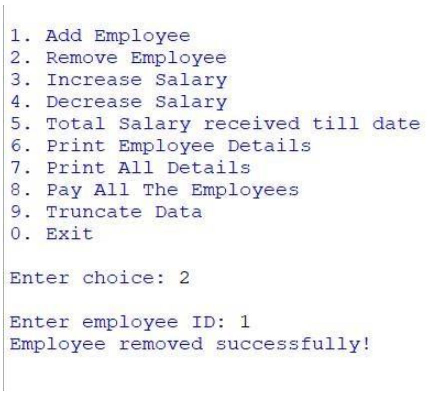


```
1. Add Employee
2. Remove Employee
3. Increase Salary
4. Decrease Salary
5. Total Salary received till date
6. Print Employee Details
7. Print All Details
8. Pay All The Employees
9. Truncate Data
0. Exit

Enter choice: 1

Enter name: Rohit
Enter age: 21
Enter address: Abu Dhabi
Enter starting salary: 45000
Employee added successfully!


1. Add Employee
2. Remove Employee
3. Increase Salary
4. Decrease Salary
5. Total Salary received till date
6. Print Employee Details
7. Print All Details
8. Pay All The Employees
9. Truncate Data
0. Exit

Enter choice: 1

Enter name: Lakshmi
Enter age: 22
Enter address: Kochi
Enter starting salary: 40000
Employee added successfully!
```



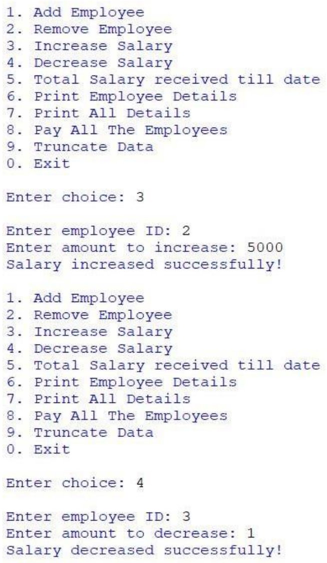
```
1. Add Employee
2. Remove Employee
3. Increase Salary
4. Decrease Salary
5. Total Salary received till date
6. Print Employee Details
7. Print All Details
8. Pay All The Employees
9. Truncate Data
0. Exit

Enter choice: 1

Enter name: Louis
Enter age: 20
Enter address: Thrissur
Enter starting salary: 44999
Employee added successfully!
```

#Output for case 2



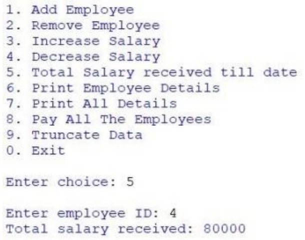
```
1. Add Employee
2. Remove Employee
3. Increase Salary
4. Decrease Salary
5. Total Salary received till date
6. Print Employee Details
7. Print All Details
8. Pay All The Employees
9. Truncate Data
0. Exit

Enter choice: 2

Enter employee ID: 1
Employee removed successfully!
```

#Output for case 3 and 4

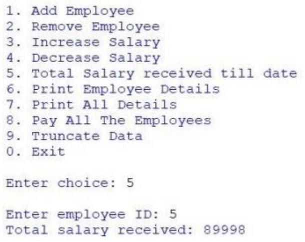


```
1. Add Employee
2. Remove Employee
3. Increase Salary
4. Decrease Salary
5. Total Salary received till date
6. Print Employee Details
7. Print All Details
8. Pay All The Employees
9. Truncate Data
0. Exit

Enter choice: 3

Enter employee ID: 2
Enter amount to increase: 5000
Salary increased successfully!

1. Add Employee
2. Remove Employee
3. Increase Salary
4. Decrease Salary
5. Total Salary received till date
6. Print Employee Details
7. Print All Details
8. Pay All The Employees
9. Truncate Data
0. Exit

Enter choice: 4

Enter employee ID: 3
Enter amount to decrease: 1
Salary decreased successfully!
```

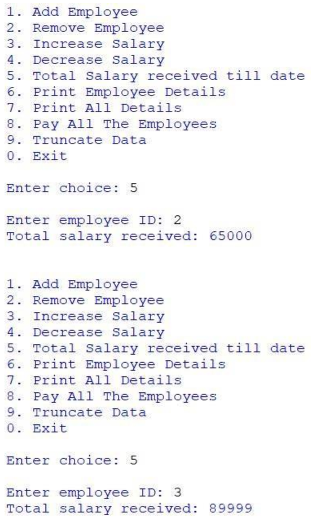
#Output for case 5



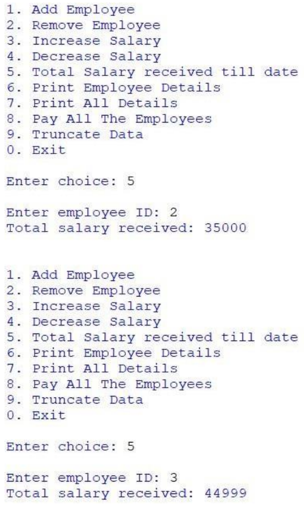
```
1. Add Employee
2. Remove Employee
3. Increase Salary
4. Decrease Salary
5. Total Salary received till date
6. Print Employee Details
7. Print All Details
8. Pay All The Employees
9. Truncate Data
0. Exit

Enter choice: 5

Enter employee ID: 4
Total salary received: 80000
```



```
1. Add Employee
2. Remove Employee
3. Increase Salary
4. Decrease Salary
5. Total Salary received till date
6. Print Employee Details
7. Print All Details
8. Pay All The Employees
9. Truncate Data
0. Exit

Enter choice: 5

Enter employee ID: 5
Total salary received: 89998
```

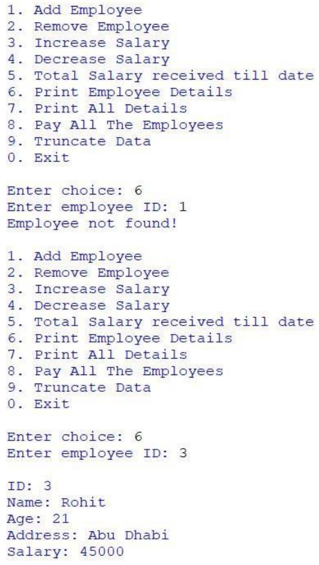


```
1. Add Employee
2. Remove Employee
3. Increase Salary
4. Decrease Salary
5. Total Salary received till date
6. Print Employee Details
7. Print All Details
8. Pay All The Employees
9. Truncate Data
0. Exit

Enter choice: 5

Enter employee ID: 2
Total salary received: 65000


1. Add Employee
2. Remove Employee
3. Increase Salary
4. Decrease Salary
5. Total Salary received till date
6. Print Employee Details
7. Print All Details
8. Pay All The Employees
9. Truncate Data
0. Exit

Enter choice: 5

Enter employee ID: 3
Total salary received: 89999
```



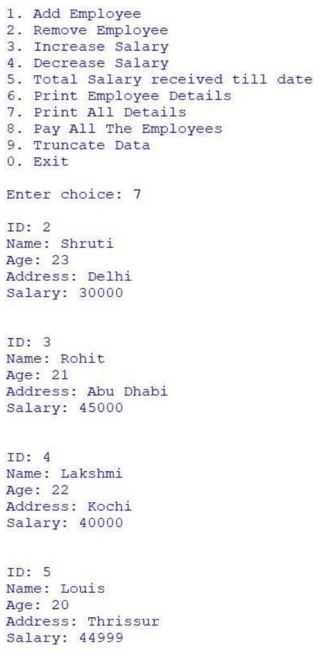
```
1. Add Employee
2. Remove Employee
3. Increase Salary
4. Decrease Salary
5. Total Salary received till date
6. Print Employee Details
7. Print All Details
8. Pay All The Employees
9. Truncate Data
0. Exit

Enter choice: 5

Enter employee ID: 2
Total salary received: 35000


1. Add Employee
2. Remove Employee
3. Increase Salary
4. Decrease Salary
5. Total Salary received till date
6. Print Employee Details
7. Print All Details
8. Pay All The Employees
9. Truncate Data
0. Exit

Enter choice: 5

Enter employee ID: 3
Total salary received: 44999
```

#Output for case 6



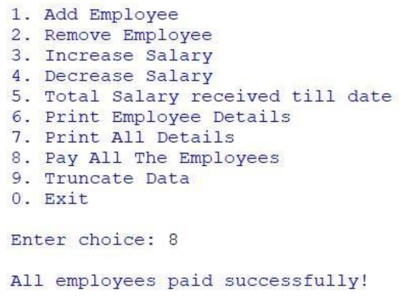
```
1. Add Employee
2. Remove Employee
3. Increase Salary
4. Decrease Salary
5. Total Salary received till date
6. Print Employee Details
7. Print All Details
8. Pay All The Employees
9. Truncate Data
0. Exit

Enter choice: 6
Enter employee ID: 1
Employee not found!

1. Add Employee
2. Remove Employee
3. Increase Salary
4. Decrease Salary
5. Total Salary received till date
6. Print Employee Details
7. Print All Details
8. Pay All The Employees
9. Truncate Data
0. Exit

Enter choice: 6
Enter employee ID: 3

ID: 3
Name: Rohit
Age: 21
Address: Abu Dhabi
Salary: 45000
```

#Output for case 7



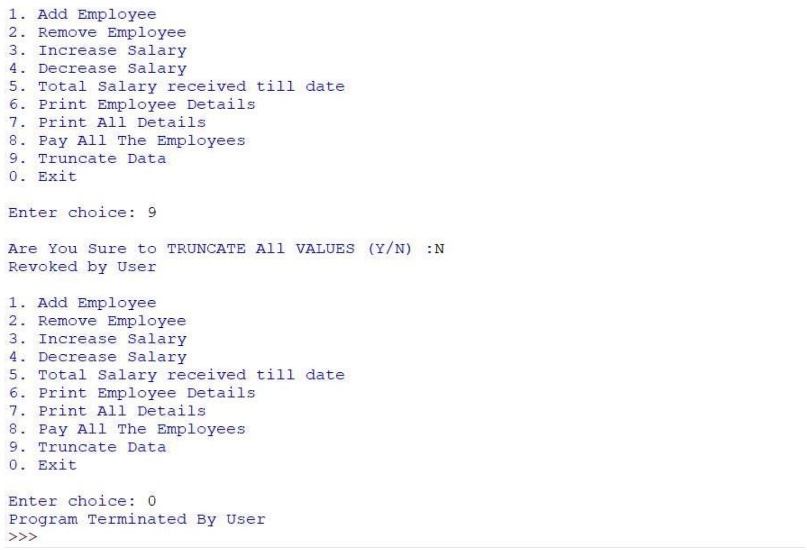
```
1. Add Employee
2. Remove Employee
3. Increase Salary
4. Decrease Salary
5. Total Salary received till date
6. Print Employee Details
7. Print All Details
8. Pay All The Employees
9. Truncate Data
0. Exit

Enter choice: 7

ID: 2
Name: Shruti
Age: 23
Address: Delhi
Salary: 30000


ID: 3
Name: Rohit
Age: 21
Address: Abu Dhabi
Salary: 45000


ID: 4
Name: Lakshmi
Age: 22
Address: Kochi
Salary: 40000


ID: 5
Name: Louis
Age: 20
Address: Thrissur
Salary: 44999
```

#Output for case 8



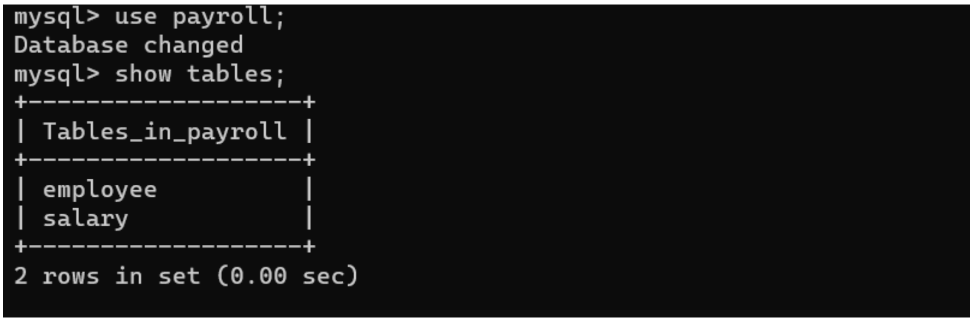
```
1. Add Employee
2. Remove Employee
3. Increase Salary
4. Decrease Salary
5. Total Salary received till date
6. Print Employee Details
7. Print All Details
8. Pay All The Employees
9. Truncate Data
0. Exit

Enter choice: 8

All employees paid successfully!
```

#Output for case 9

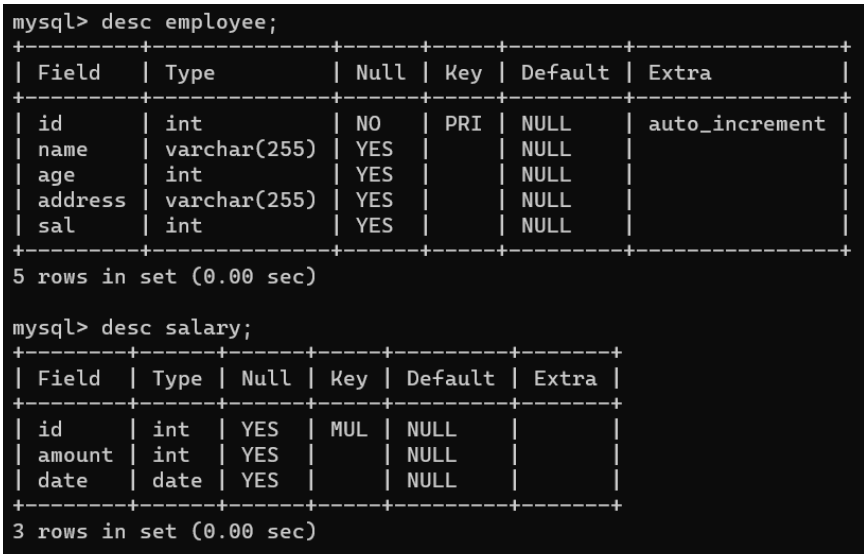


```
1. Add Employee
2. Remove Employee
3. Increase Salary
4. Decrease Salary
5. Total Salary received till date
6. Print Employee Details
7. Print All Details
8. Pay All The Employees
9. Truncate Data
0. Exit

Enter choice: 9

Are You Sure to TRUNCATE All VALUES (Y/N) :N
Revoked by User

1. Add Employee
2. Remove Employee
3. Increase Salary
4. Decrease Salary
5. Total Salary received till date
6. Print Employee Details
7. Print All Details
8. Pay All The Employees
9. Truncate Data
0. Exit

Enter choice: 0
Program Terminated By User
>>>
```

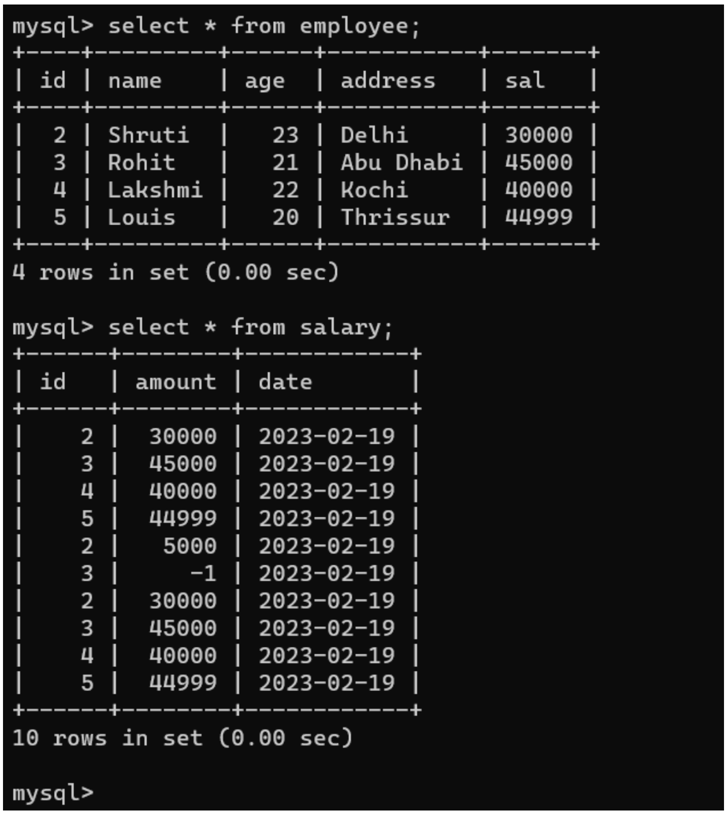
SQL TABLES



```
mysql> use payroll;
Database changed
mysql> show tables;
+-------------------+
| Tables_in_payroll |
+-------------------+
| employee          |
| salary            |
+-------------------+
2 rows in set (0.00 sec)
```



```
mysql> desc employee;
+---------+--------------+------+-----+---------+----------------+
| Field   | Type         | Null | Key | Default | Extra          |
+---------+--------------+------+-----+---------+----------------+
| id      | int          | NO   | PRI | NULL    | auto_increment |
| name    | varchar(255) | YES  |     | NULL    |                |
| age     | int          | YES  |     | NULL    |                |
| address | varchar(255) | YES  |     | NULL    |                |
| sal     | int          | YES  |     | NULL    |                |
+---------+--------------+------+-----+---------+----------------+
5 rows in set (0.00 sec)

mysql> desc salary;
+--------+------+------+-----+---------+-------+
| Field  | Type | Null | Key | Default | Extra |
+--------+------+------+-----+---------+-------+
| id     | int  | YES  | MUL | NULL    |       |
| amount | int  | YES  |     | NULL    |       |
| date   | date | YES  |     | NULL    |       |
+--------+------+------+-----+---------+-------+
3 rows in set (0.00 sec)
```



```
mysql> select * from employee;
+----+---------+------+------------+-------+
| id | name    | age  | address    | sal   |
+----+---------+------+------------+-------+
|  2 | Shruti  |   23 | Delhi      | 30000 |
|  3 | Rohit   |   21 | Abu Dhabi  | 45000 |
|  4 | Lakshmi |   22 | Kochi      | 40000 |
|  5 | Louis   |   20 | Thrissur   | 44999 |
+----+---------+------+------------+-------+
4 rows in set (0.00 sec)

mysql> select * from salary;
+------+--------+------------+
| id   | amount | date       |
+------+--------+------------+
|    2 |  30000 | 2023-02-19 |
|    3 |  45000 | 2023-02-19 |
|    4 |  40000 | 2023-02-19 |
|    5 |  44999 | 2023-02-19 |
|    2 |   5000 | 2023-02-19 |
|    3 |     -1 | 2023-02-19 |
|    2 |  30000 | 2023-02-19 |
|    3 |  45000 | 2023-02-19 |
|    4 |  40000 | 2023-02-19 |
|    5 |  44999 | 2023-02-19 |
+------+--------+------------+
10 rows in set (0.00 sec)

mysql>
```