* **Employee Management System SQL Data analysis**

The task is to create a Database-driven Employee Management System in Python that will store the information in the MySQL Database. The script will contain the following operations :

* Add Employee
* Remove Employee
* Promote Employee
* Display Employees

The idea is that we perform different changes in our Employee Record by using different functions for example the Add\_Employee will insert a new row in our Employee, also, we will create a Remove Employee Function which will delete the record of any particular existing employee in our Employee table. This System works on the concepts of taking the information from the database making required changes in the fetched data and applying the changes in the record which we will see in our Promote Employee System. We can also have the information about all the existing employees by using the Display Employee function. The main advantage of connecting our program to the database is that the information becomes lossless even after closing our program a number of times.

**Getting Started**

For creating the Employee Management System in Python that uses MySQL database we need to connect Python with MySQL.

For making a connection we need to install **[mysqlconnector](https://www.geeksforgeeks.org/mysql-connector-python-module-in-python/)** which can be done by writing the following command in the command prompt on Windows.

**import** pyodbc  
  
  
*# SQL Server connection details*sql\_server\_config = {  
 **'Driver'**: **'{SQL Server}'**,  
 **'Server'**: **'DESKTOP-V9NJ13A'**,  
 **'Database'**: **'employee\_management\_system'**}  
  
**def** fetch\_data\_from\_sql\_server():  
 **try**:  
 *# Connect to SQL Server* sql\_conn = pyodbc.connect(\*\*sql\_server\_config)  
 cursor = sql\_conn.cursor()  
  
 *# Fetch data from SQL Server* cursor.execute(**'SELECT \* FROM dbo.Employee'**)  
 data = cursor.fetchall()  
  
 **return** data  
  
 **except** pyodbc.Error **as** ex:  
 print(**f"Error connecting to SQL Server: {**ex**}"**)  
 **return None**result = fetch\_data\_from\_sql\_server()  
print(result)

Now we are Done with the connections, so we can focus on our Employee Management System

The idea is that we keep all the information about theEmployee in the above table and manipulate the table whenever required. So now we will look at the working of each operation in detail.

**Check Employee Function**

The check employee function takes employee id as a parameter and checks whether any employee with given id exists in the employee details record or not. For checking this it uses **cursor.rowcount()** function which counts the number of rows that match with given details. It is a utility function, and we will see its use in later operations like Add employee function, etc.

Program :-

# Function To Check if Employee with

# given Id Exist or Not

**def** check\_employee(employee\_id):

    # Query to select all Rows f

    # rom employee Table

    sql **=** 'select \* from empd where id=?'

    # making cursor buffered to make

    # rowcount method work properly

    c **=** con.cursor(buffered**=**True)

    data **=** (employee\_id,)

    # Executing the SQL Query

    c.execute(sql, data)

    # rowcount method to find

    # number of rows with given values

    r **=** c.fetchone()

**if** r **is not None**:

**return** True

**else**:

**return** False

**Add Employee Function**

The Add Employee function will ask for the Employee Id and uses the check employee function to check whether the employee to be added already exist in our record or not if employee details do not already exist then it asks for details of the employee to be added like Employee Name, Post of Employee and Salary of the employee. Now after getting all such details from the user of that system it simply inserts the information in our Employee details table.

Program :-

# Function to mAdd\_Employee

**def** Add\_Employ():

    Id **=** input("Enter Employee Id : ")

    # Checking if Employee with given Id

    # Already Exist or Not

**if**(check\_employee(Id) **==** True):

        print("Employee already exists\nTry Again\n")

        menu()

**else**:

        Name **=** input("Enter Employee Name : ")

        Post **=** input("Enter Employee Post : ")

        Salary **=** input("Enter Employee Salary : ")

        data **=** (Id, Name, Post, Salary)

        # Inserting Employee details in the Employee

        # Table

        sql **=** 'insert into empd values(?,?,?,?)'

        c **=** con.cursor()

        # Executing the SQL Query

        c.execute(sql, data)

        # commit() method to make changes in the table

        con.commit()

        print("Employee Added Successfully ")

        menu()

**Remove Employee Function**

The Remove Employee Function will simply ask for Id of the employee to be removed because Id is [Primary key](https://www.geeksforgeeks.org/primary-key-constraint-in-sql/) in our Employee Details Record as there can be two employees with the same name, but they must have a unique id. The Remove Employee function uses the check employee function to check whether the employee to be removed exists in our record or not if employee details exist then after getting a valid employee id it deletes the record corresponding to that employee id.

Program :-

# Function to Remove Employee with given Id

**def** Remove\_Employ():

    Id **=** input("Enter Employee Id : ")

    # Checking if Employee with given Id

    # Exist or Not

**if**(check\_employee(Id) **==** False):

**print**("Employee does not  exists\nTry Again\n")

        menu()

**else**:

        # Query to Delete Employee from Table

        sql **=** 'delete from empd where id=?'

        data **=** (Id,)

        c **=** con.cursor()

        # Executing the SQL Query

        c.execute(sql, data)

        # commit() method to make changes in

        # the table

        con.commit()

        print("Employee Removed")

        menu()

**Promote Employee Function**

The Promote Employee function will ask for the Employee Id and uses the check employee function to check whether the employee to be Promoted exist in our record or not if employee details exist then it will ask for the amount by which his salary is to be increased. After getting the valid details it increases the salary of the employee with the given id by the given amount.

**Program:-**

# Function to Promote Employee

**def** Promote\_Employee():

    Id **=** int(input("Enter Employ's Id"))

    # Checking if Employee with given Id

    # Exist or Not

**if**(check\_employee(Id) **==** False):

        print("Employee does not  exists\nTry Again\n")

        menu()

**else**:

        Amount **=** int(input("Enter increase in Salary"))

        # Query to Fetch Salary of Employee with

        # given Id

        sql **=** 'select salary from empd where id=?'

        data **=** (Id,)

        c **=** con.cursor()

        # Executing the SQL Query

        c.execute(sql, data)

        # Fetching Salary of Employee with given Id

        r **=** c.fetchone()

        t **=** r[0]**+**Amount

        # Query to Update Salary of Employee with

        # given Id

        sql **=** 'update empd set salary=? where id=?'

        d **=** (t, Id)

        # Executing the SQL Query

        c.execute(sql, d)

        # commit() method to make changes in the table

        con.commit()

        print("Employee Promoted")

        menu()

**Display Employees Function**

The Display Employees function is simply a select query of SQL which fetches all the records stored in the employee details table and prints them line by line.

**Program:**

# Function to Display All Employees

# from Employee Table

**def** Display\_Employees():

    # query to select all rows from

    # Employee Table

    sql **=** 'select \* from empd'

    c **=** con.cursor()

    # Executing the SQL Query

    c.execute(sql)

    # Fetching all details of all the

    # Employees

    r **=** c.fetchall()

**for** i **in** r:

**print**("Employee Id : ", i[0])

**print**("Employee Name : ", i[1])

**print**("Employee Post : ", i[2])

        print("Employee Salary : ", i[3])

**print**("**-----------------------------**\

**-------------------------------------**\

**-----------------------------------**")

    menu()

**Menu Function**

The Menu function displays the menu to the user and asks the user to enter his choice for performing operations like Add employee, Remove employee, etc.

**Program :-**

# menu function to display the menu

**def** menu():

**print**("Welcome to Employee Management Record")

**print**("Press ")

**print**("1 to Add Employee")

**print**("2 to Remove Employee ")

    print("3 to Promote Employee")

    print("4 to Display Employees")

**print**("5 to Exit")

    # Taking choice from user

    ch **=** int(input("Enter your Choice "))

**if** ch **==** 1:

        Add\_Employ()

**elif** ch **==** 2:

        Remove\_Employ()

**elif** ch **==** 3:

        Promote\_Employee()

**elif** ch **==** 4:

        Display\_Employees()

**elif** ch **==** 5:

        exit(0)

**else**:

        print("Invalid Choice")

        menu()

**Complete Code:**

*# importing sql connector***import** pyodbc  
  
*# SQL Server connection details*sql\_server\_config = {  
 **'Driver'**: **'{SQL Server}'**,  
 **'Server'**: **'DESKTOP-V9NJ13A'**,  
 **'Database'**: **'employee\_management\_system'**}  
  
*# making Connection*con = sql\_conn = pyodbc.connect(\*\*sql\_server\_config)  
  
*# Function to Add\_Employee***def** Add\_Employ():  
  
 Id = int(input(**"Enter Employee Id : "**))  
  
 *# Checking if Employee with given Id  
 # Already Exist or Not* **if**(check\_employee(Id) == **True**):  
 print(**"Employee already exists\nTry Again\n"**)  
 menu()  
  
 **else**:  
 Name = input(**"Enter Employee Name : "**)  
 Post = input(**"Enter Employee Post : "**)  
 Salary = int(input(**"Enter Employee Salary : "**))  
 data = (Id, Name, Post, Salary)  
  
 *# Inserting Employee details in  
 # the Employee Table* sql = **'insert into dbo.Employee values(?,?,?,?)'** c = con.cursor()  
  
 *# Executing the SQL Query* c.execute(sql, data)  
  
 *# commit() method to make changes in  
 # the table* con.commit()  
 print(**"Employee Added Successfully "**)  
 menu()  
  
*# Function to Promote Employee***def** Promote\_Employee():  
 Id = int(input(**"Enter Employ's Id"**))  
  
 *# Checking if Employee with given Id  
 # Exist or Not* **if**(check\_employee(Id) == **False**):  
 print(**"Employee does not exists\nTry Again\n"**)  
 menu()  
 **else**:  
 Amount = int(input(**"Enter increase in Salary"**))  
  
 *# Query to Fetch Salary of Employee  
 # with given Id* sql = **'select salary from dbo.Employee where id=?'** data = (Id,)  
 c = con.cursor()  
  
 *# Executing the SQL Query* c.execute(sql, data)  
  
 *# Fetching Salary of Employee with given Id* r = c.fetchone()  
 t = r[0]+Amount  
  
 *# Query to Update Salary of Employee with  
 # given Id* sql = **'update dbo.Employee set salary=? where id=?'** d = (t, Id)  
  
 *# Executing the SQL Query* c.execute(sql, d)  
  
 *# commit() method to make changes in the table* con.commit()  
 print(**"Employee Promoted"**)  
 menu()  
  
*# Function to Remove Employee with given Id***def** Remove\_Employ():  
 Id = input(**"Enter Employee Id : "**)  
  
 *# Checking if Employee with given Id Exist  
 # or Not* **if**(check\_employee(Id) == **False**):  
 print(**"Employee does not exists\nTry Again\n"**)  
 menu()  
 **else**:  
  
 *# Query to Delete Employee from Table* sql = **'delete from dbo.Employee where id=?'** data = (Id,)  
 c = con.cursor()  
  
 *# Executing the SQL Query* c.execute(sql, data)  
  
 *# commit() method to make changes in  
 # the table* con.commit()  
 print(**"Employee Removed"**)  
 menu()  
  
  
*# Function To Check if Employee with  
# given Id Exist or Not***def** check\_employee(employee\_id):  
  
 *# Query to select all Rows f  
 # rom employee Table* sql = **'select \* from dbo.Employee where id=?'** *# making cursor buffered to make  
 # rowcount method work properly* c = con.cursor()  
 data = (employee\_id,)  
  
 *# Executing the SQL Query* c.execute(sql, data)  
  
 *# rowcount method to find  
 # number of rows with given values* r = c.fetchone()  
 **if** r **is not None**:  
 **return True  
 else**:  
 **return False***# Function to Display All Employees  
# from Employee Table***def** Display\_Employees():  
  
 *# query to select all rows from  
 # Employee Table* sql = **'select \* from dbo.Employee'** c = con.cursor()  
  
 *# Executing the SQL Query* c.execute(sql)  
  
 *# Fetching all details of all the  
 # Employees* r = c.fetchall()  
 **for** i **in** r:  
 print(**"Employee Id : "**, i[0])  
 print(**"Employee Name : "**, i[1])  
 print(**"Employee Post : "**, i[2])  
 print(**"Employee Salary : "**, i[3])  
 print(**"---------------------\  
 -----------------------------\  
 ------------------------------\  
 ---------------------"**)  
  
 menu()  
  
*# menu function to display menu***def** menu():  
 print(**"Welcome to Employee Management Record"**)  
 print(**"Press "**)  
 print(**"1 to Add Employee"**)  
 print(**"2 to Remove Employee "**)  
 print(**"3 to Promote Employee"**)  
 print(**"4 to Display Employees"**)  
 print(**"5 to Exit"**)  
  
 ch = int(input(**"Enter your Choice "**))  
 **if** ch == 1:  
 Add\_Employ()  
 **elif** ch == 2:  
 Remove\_Employ()  
 **elif** ch == 3:  
 Promote\_Employee()  
 **elif** ch == 4:  
 Display\_Employees()  
 **elif** ch == 5:  
 exit(0)  
 **else**:  
 print(**"Invalid Choice"**)  
 menu()  
  
  
*# Calling menu function*menu()