Programming Assignment 2 Report

Student(s):
William Issac Tesfagiorgish,

Philomina Ejegi Ede

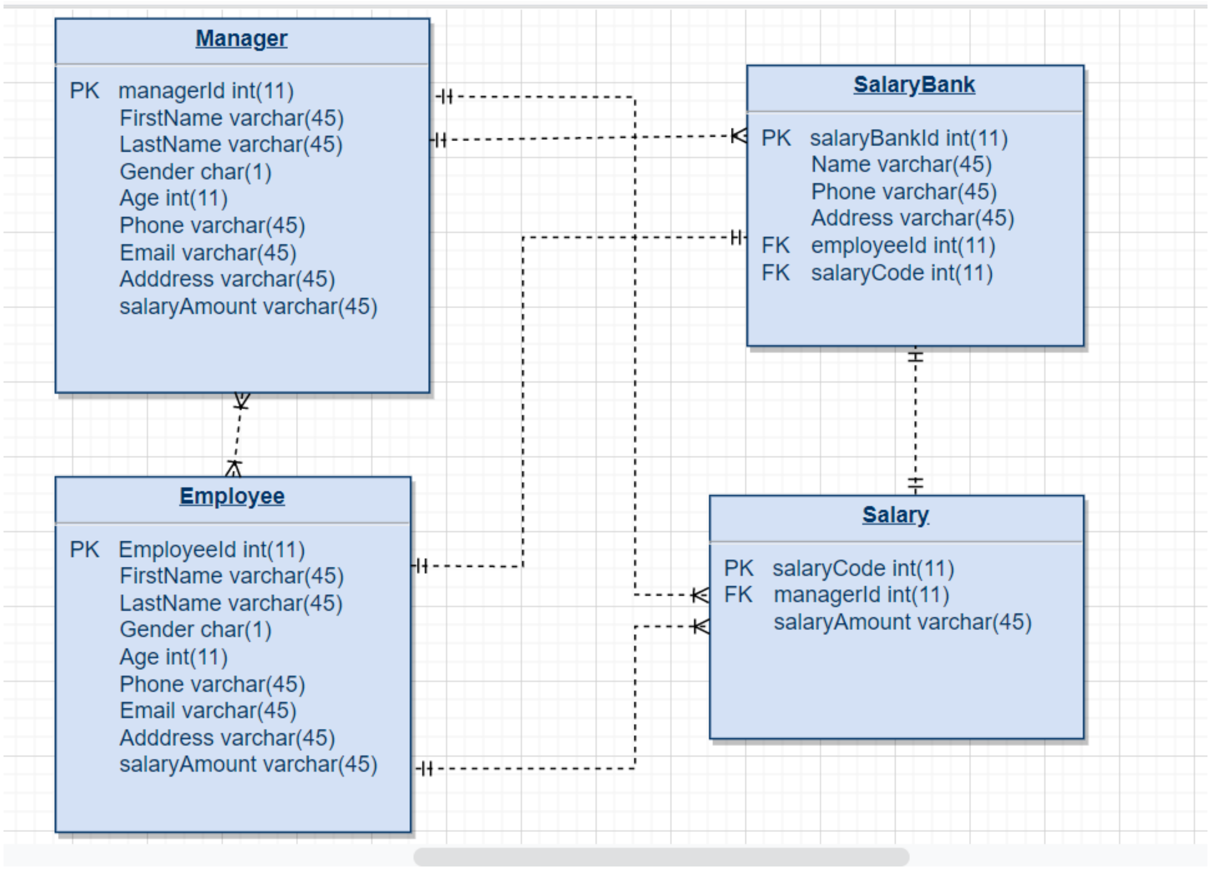
Ilir Jusufi - ilir.jusufi@lnu.se
Maria Ulan - maria.ulan@lnu.se

Project Idea

The idea of our system design and implementation is "A Salary Database". The System holds a list of Employees and the Salary amount that is being paid, their different Salary Bank and assigned unique Manager Id. This system makes it easy to keep track of employees' salary information, and to ascertain what an employee has earned over a period of time and when they are to be considered or due for a raise in salary or bonus. The dataset used was downloaded from the weblink, https://think.cs.vt.edu/corgis/csv/.

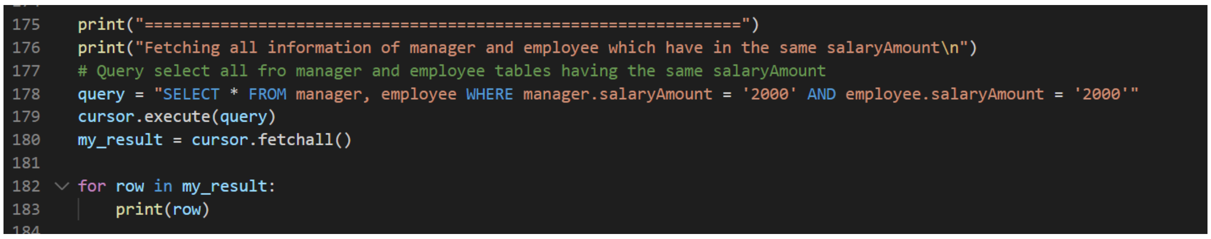
Schema Design

There four tables which are (Manager, employee, salary, and salary Bank). The manager is the one who hire an employee, and both are entities which have much similar attributes except the working position, and both have ID's as primary key. A Manager and can hire one or many employees and an employee can also be hired by one or many (two) managers in different place then gets his/her salary from one or two jobs. A manager pays a monthly salary to an employee through one and only one bank. On the salary bank or payslip exist the employee ID and salary code as foreign key. The relationship between salary bank and the salary is one-one because a salary can only be paid out through one bank at a time.

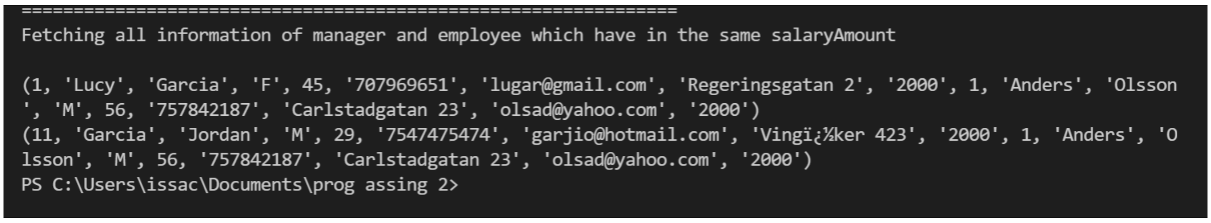


SQL Queries

This query prints all from the table (manager and employee). Where manager pays 2000 to the employee and the employee receive it.

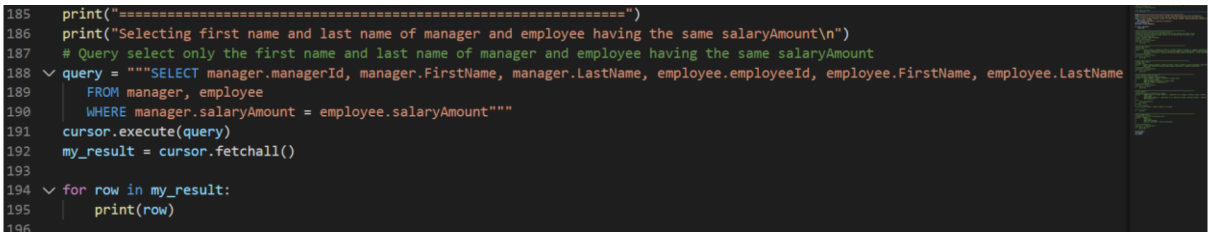


```python
print("==============================================================")
print("Fetching all information of manager and employee which have in the same salaryAmount\n")
# Query select all fro manager and employee tables having the same salaryAmount
query = "SELECT * FROM manager, employee WHERE manager.salaryAmount = '2000' AND employee.salaryAmount = '2000'"
cursor.execute(query)
my_result = cursor.fetchall()

for row in my_result:
    print(row)
```

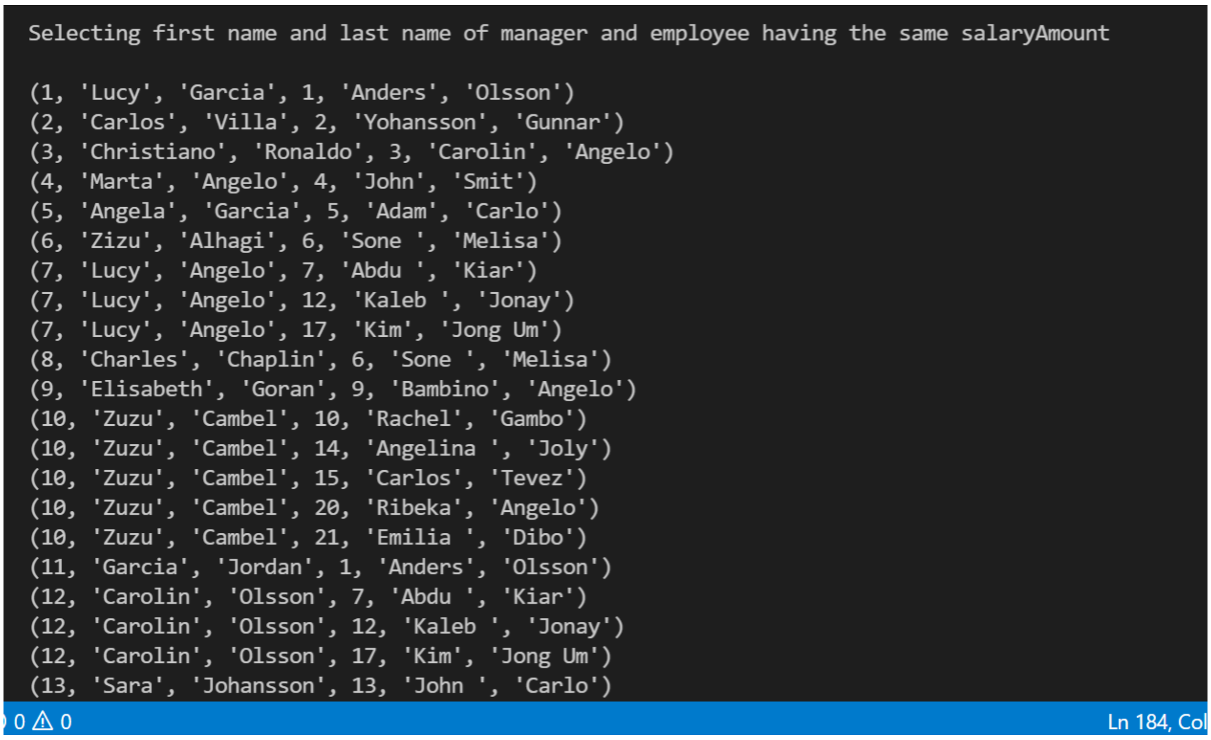


```
================================================================
Fetching all information of manager and employee which have in the same salaryAmount

(1, 'Lucy', 'Garcia', 'F', 45, '707969651', 'lugar@gmail.com', 'Regeringsgatan 2', '2000', 1, 'Anders', 'Olsson
', 'M', 56, '757842187', 'Carlstadgatan 23', 'olsad@yahoo.com', '2000')
(11, 'Garcia', 'Jordan', 'M', 29, '7547475474', 'garjio@hotmail.com', 'Vingï¿½ker 423', '2000', 1, 'Anders', 'O
lsson', 'M', 56, '757842187', 'Carlstadgatan 23', 'olsad@yahoo.com', '2000')
PS C:\Users\issac\Documents\prog assing 2>
```

Here it prints the manager ID & first name, and employee’s ID first name and last name from the table (manager and employee). Where the salary amount is the same on both (manager and employee).



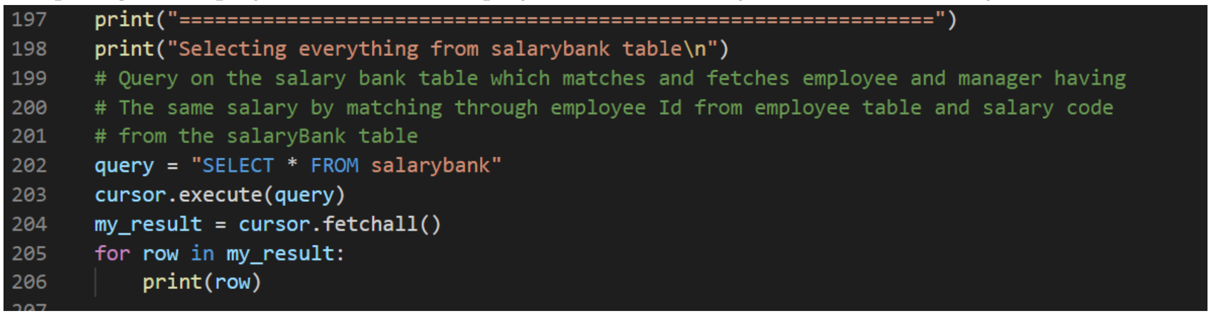
```python
print("================================================================")
print("Selecting first name and last name of manager and employee having the same salaryAmount\n")
# Query select only the first name and last name of manager and employee having the same salaryAmount
query = """SELECT manager.managerId, manager.FirstName, manager.LastName, employee.employeeId, employee.FirstName, employee.LastName
   FROM manager, employee
   WHERE manager.salaryAmount = employee.salaryAmount"""
cursor.execute(query)
my_result = cursor.fetchall()

for row in my_result:
    print(row)
```



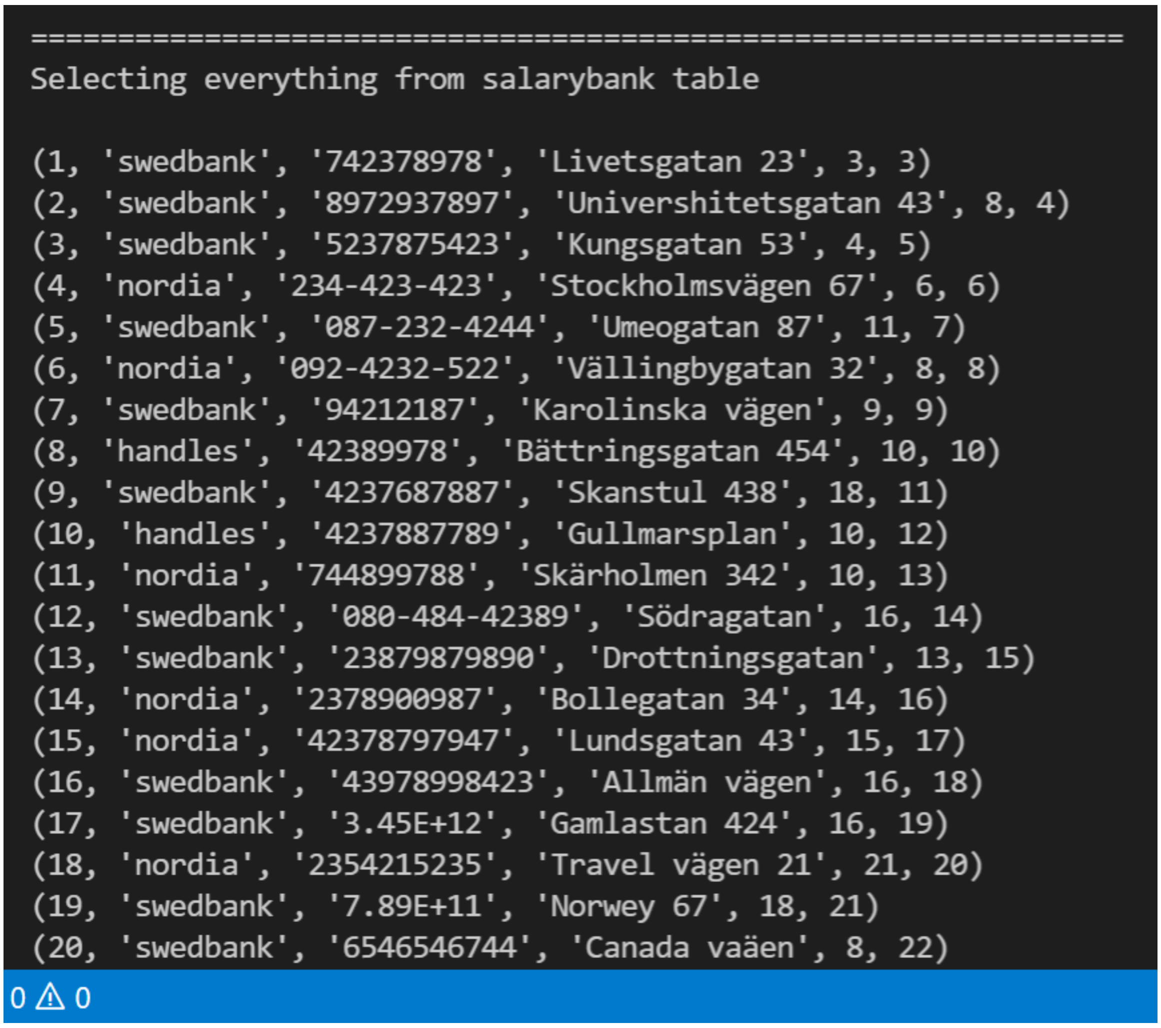
```
Selecting first name and last name of manager and employee having the same salaryAmount

(1, 'Lucy', 'Garcia', 1, 'Anders', 'Olsson')
(2, 'Carlos', 'Villa', 2, 'Yohansson', 'Gunnar')
(3, 'Christiano', 'Ronaldo', 3, 'Carolin', 'Angelo')
(4, 'Marta', 'Angelo', 4, 'John', 'Smit')
(5, 'Angela', 'Garcia', 5, 'Adam', 'Carlo')
(6, 'Zizu', 'Alhagi', 6, 'Sone ', 'Melisa')
(7, 'Lucy', 'Angelo', 7, 'Abdu ', 'Kiar')
(7, 'Lucy', 'Angelo', 12, 'Kaleb ', 'Jonay')
(7, 'Lucy', 'Angelo', 17, 'Kim', 'Jong Um')
(8, 'Charles', 'Chaplin', 6, 'Sone ', 'Melisa')
(9, 'Elisabeth', 'Goran', 9, 'Bambino', 'Angelo')
(10, 'Zuzu', 'Cambel', 10, 'Rachel', 'Gambo')
(10, 'Zuzu', 'Cambel', 14, 'Angelina ', 'Joly')
(10, 'Zuzu', 'Cambel', 15, 'Carlos', 'Tevez')
(10, 'Zuzu', 'Cambel', 20, 'Ribeka', 'Angelo')
(10, 'Zuzu', 'Cambel', 21, 'Emilia ', 'Dibo')
(11, 'Garcia', 'Jordan', 1, 'Anders', 'Olsson')
(12, 'Carolin', 'Olsson', 7, 'Abdu ', 'Kiar')
(12, 'Carolin', 'Olsson', 12, 'Kaleb ', 'Jonay')
(12, 'Carolin', 'Olsson', 17, 'Kim', 'Jong Um')
(13, 'Sara', 'Johansson', 13, 'John ', 'Carlo')
```

0 ⚠ 0 Ln 184, Col

Query on the salary bank table which matches and gets employee and manager have the same by comparing the employee ID from the employee table and salary code from the salary table.

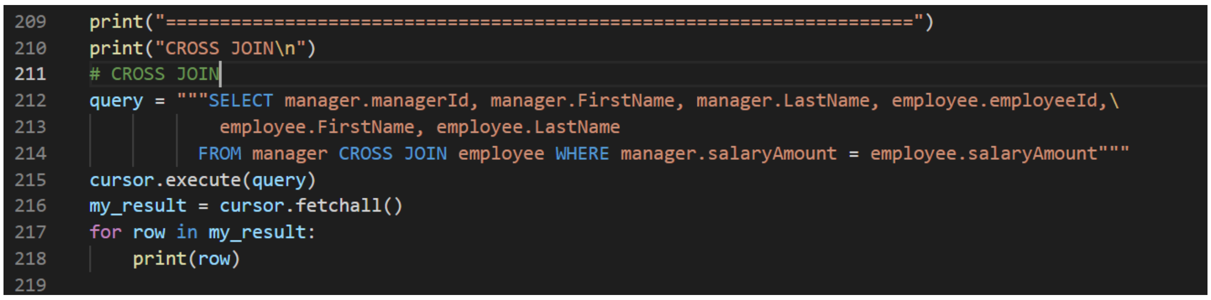


```python
print("================================================================")
print("Selecting everything from salarybank table\n")
# Query on the salary bank table which matches and fetches employee and manager having
# The same salary by matching through employee Id from employee table and salary code
# from the salaryBank table
query = "SELECT * FROM salarybank"
cursor.execute(query)
my_result = cursor.fetchall()
for row in my_result:
    print(row)
```

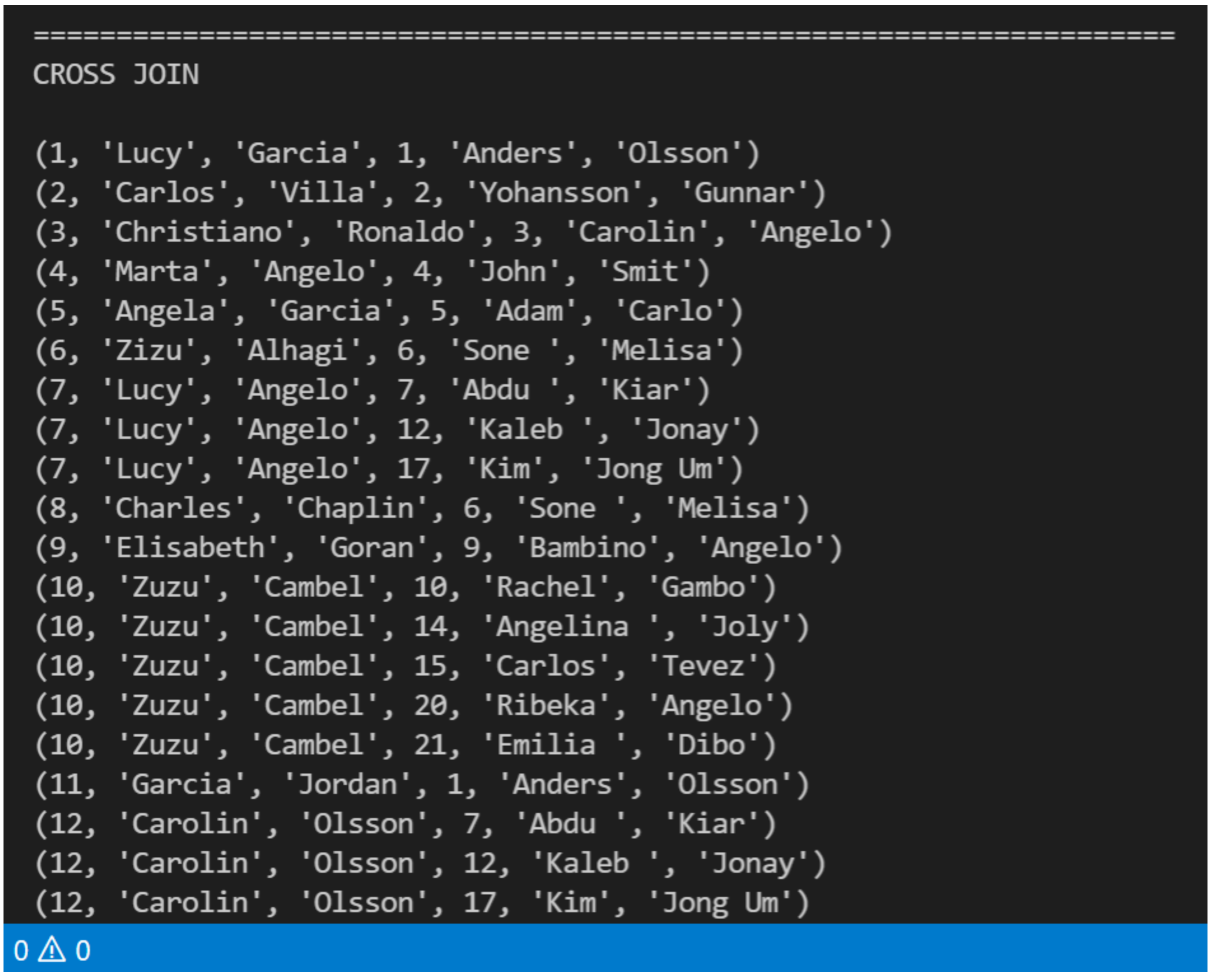


```
===============================================================
Selecting everything from salarybank table

(1, 'swedbank', '742378978', 'Livetsgatan 23', 3, 3)
(2, 'swedbank', '8972937897', 'Univershitetsgatan 43', 8, 4)
(3, 'swedbank', '5237875423', 'Kungsgatan 53', 4, 5)
(4, 'nordia', '234-423-423', 'Stockholmsvägen 67', 6, 6)
(5, 'swedbank', '087-232-4244', 'Umeogatan 87', 11, 7)
(6, 'nordia', '092-4232-522', 'Vällingbygatan 32', 8, 8)
(7, 'swedbank', '94212187', 'Karolinska vägen', 9, 9)
(8, 'handles', '42389978', 'Bättringsgatan 454', 10, 10)
(9, 'swedbank', '4237687887', 'Skanstul 438', 18, 11)
(10, 'handles', '4237887789', 'Gullmarsplan', 10, 12)
(11, 'nordia', '744899788', 'Skärholmen 342', 10, 13)
(12, 'swedbank', '080-484-42389', 'Södragatan', 16, 14)
(13, 'swedbank', '23879879890', 'Drottningsgatan', 13, 15)
(14, 'nordia', '2378900987', 'Bollegatan 34', 14, 16)
(15, 'nordia', '42378797947', 'Lundsgatan 43', 15, 17)
(16, 'swedbank', '43978998423', 'Allmän vägen', 16, 18)
(17, 'swedbank', '3.45E+12', 'Gamlastan 424', 16, 19)
(18, 'nordia', '2354215235', 'Travel vägen 21', 21, 20)
(19, 'swedbank', '7.89E+11', 'Norwey 67', 18, 21)
(20, 'swedbank', '6546546744', 'Canada vaäen', 8, 22)
```
0 ⚠ 0

Cross join, it selects the manager ID, first name, last name and employee's ID, first name and last name by crossing and join both manager and employee table, where salary amount is same on both.

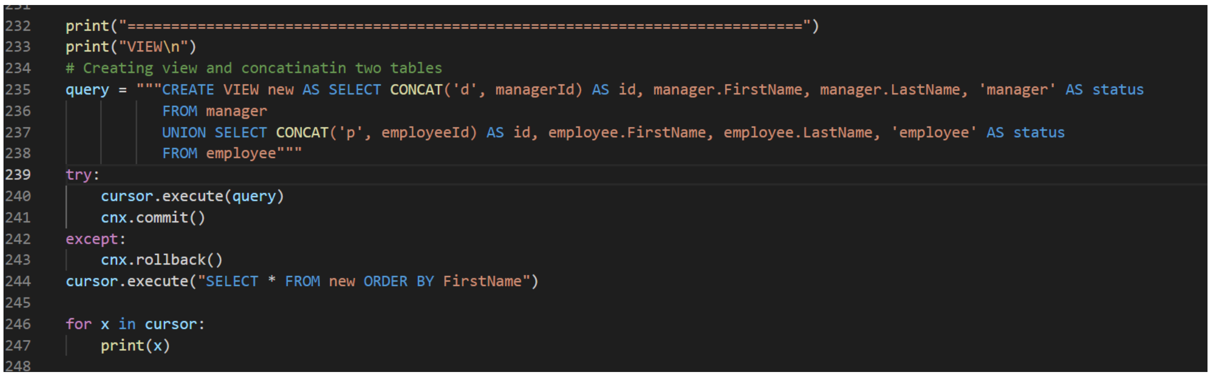


```python
print("===============================================================")
print("CROSS JOIN\n")
# CROSS JOIN
query = """SELECT manager.managerId, manager.FirstName, manager.LastName, employee.employeeId,\
            employee.FirstName, employee.LastName
          FROM manager CROSS JOIN employee WHERE manager.salaryAmount = employee.salaryAmount"""
cursor.execute(query)
my_result = cursor.fetchall()
for row in my_result:
    print(row)
```

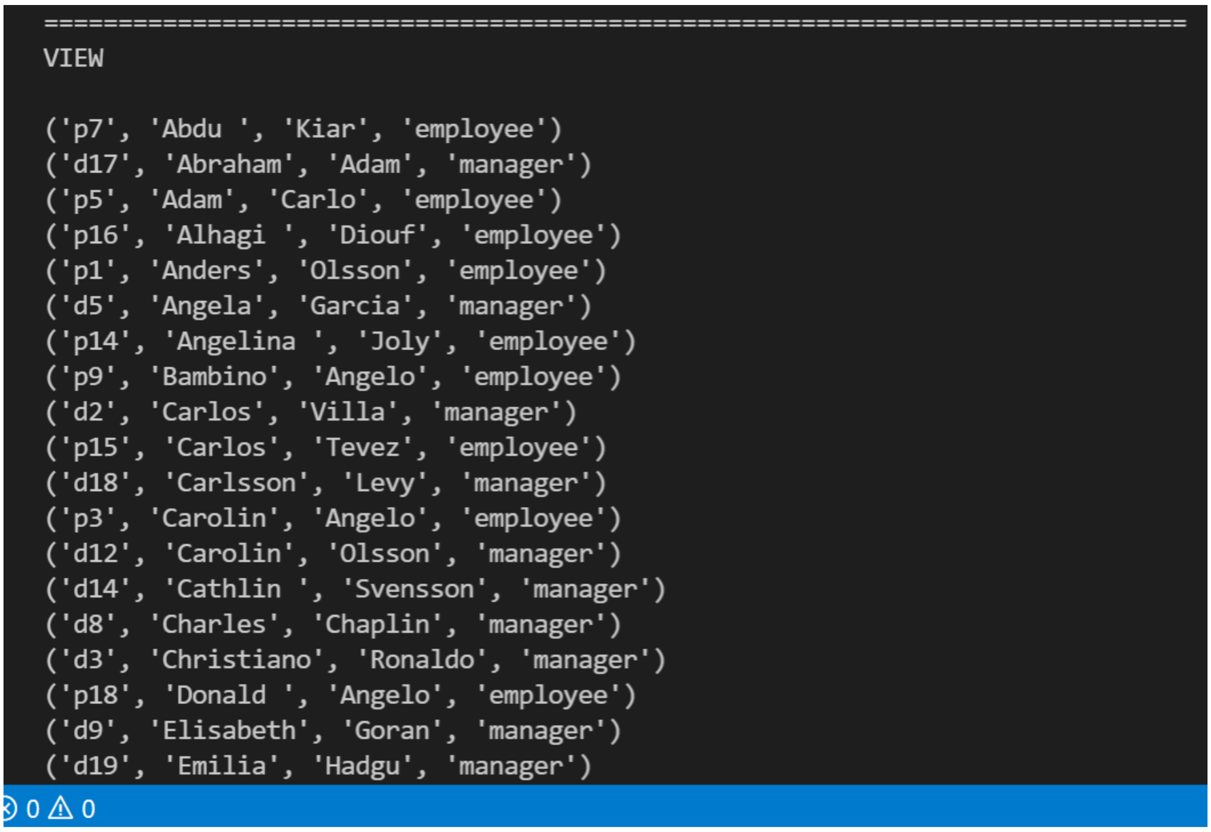


```
=============================================================================
CROSS JOIN

(1, 'Lucy', 'Garcia', 1, 'Anders', 'Olsson')
(2, 'Carlos', 'Villa', 2, 'Yohansson', 'Gunnar')
(3, 'Christiano', 'Ronaldo', 3, 'Carolin', 'Angelo')
(4, 'Marta', 'Angelo', 4, 'John', 'Smit')
(5, 'Angela', 'Garcia', 5, 'Adam', 'Carlo')
(6, 'Zizu', 'Alhagi', 6, 'Sone ', 'Melisa')
(7, 'Lucy', 'Angelo', 7, 'Abdu ', 'Kiar')
(7, 'Lucy', 'Angelo', 12, 'Kaleb ', 'Jonay')
(7, 'Lucy', 'Angelo', 17, 'Kim', 'Jong Um')
(8, 'Charles', 'Chaplin', 6, 'Sone ', 'Melisa')
(9, 'Elisabeth', 'Goran', 9, 'Bambino', 'Angelo')
(10, 'Zuzu', 'Cambel', 10, 'Rachel', 'Gambo')
(10, 'Zuzu', 'Cambel', 14, 'Angelina ', 'Joly')
(10, 'Zuzu', 'Cambel', 15, 'Carlos', 'Tevez')
(10, 'Zuzu', 'Cambel', 20, 'Ribeka', 'Angelo')
(10, 'Zuzu', 'Cambel', 21, 'Emilia ', 'Dibo')
(11, 'Garcia', 'Jordan', 1, 'Anders', 'Olsson')
(12, 'Carolin', 'Olsson', 7, 'Abdu ', 'Kiar')
(12, 'Carolin', 'Olsson', 12, 'Kaleb ', 'Jonay')
(12, 'Carolin', 'Olsson', 17, 'Kim', 'Jong Um')
0 ⚠ 0
```

View, this creates new view by concatenate “d”, manager as Id and first and last name of the manager from the manager table. And union the “p” as employee ID and first and last name, from employee table.

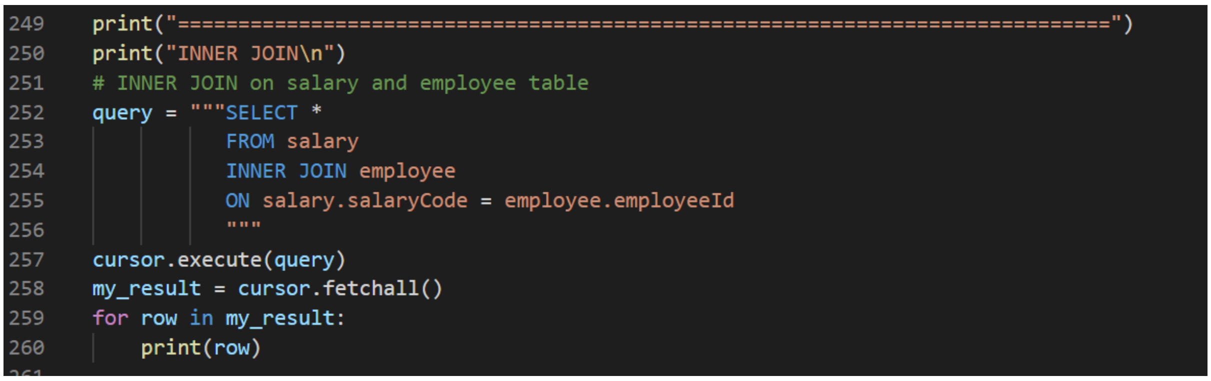


```python
print("=============================================================================")
print("VIEW\n")
# Creating view and concatinatin two tables
query = """CREATE VIEW new AS SELECT CONCAT('d', managerId) AS id, manager.FirstName, manager.LastName, 'manager' AS status
            FROM manager
            UNION SELECT CONCAT('p', employeeId) AS id, employee.FirstName, employee.LastName, 'employee' AS status
            FROM employee"""
try:
    cursor.execute(query)
    cnx.commit()
except:
    cnx.rollback()
cursor.execute("SELECT * FROM new ORDER BY FirstName")

for x in cursor:
    print(x)
```

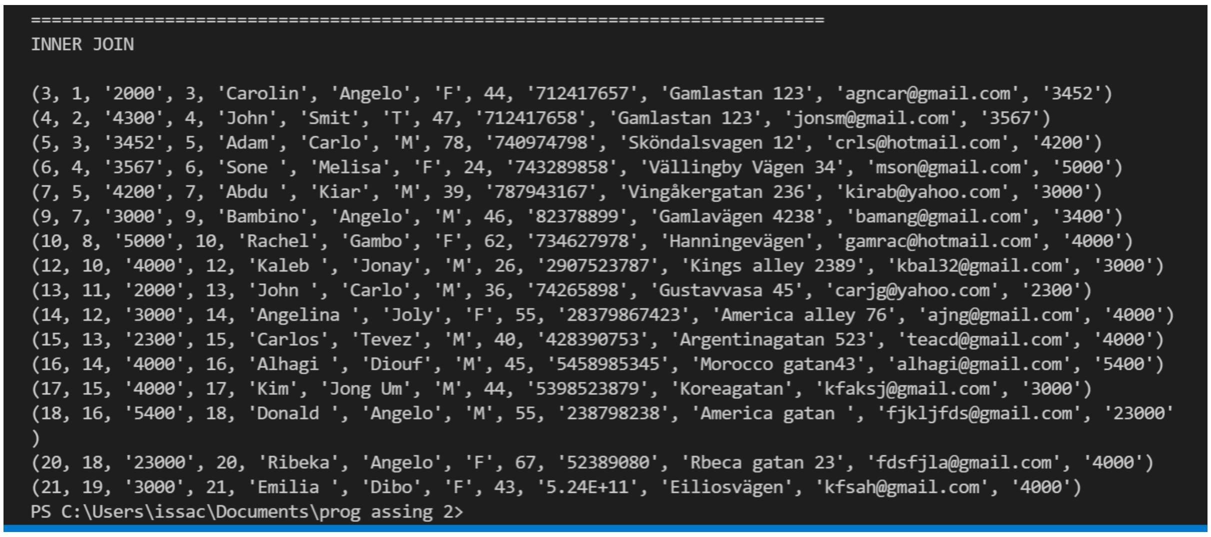


```
=============================================================================
VIEW

('p7', 'Abdu ', 'Kiar', 'employee')
('d17', 'Abraham', 'Adam', 'manager')
('p5', 'Adam', 'Carlo', 'employee')
('p16', 'Alhagi ', 'Diouf', 'employee')
('p1', 'Anders', 'Olsson', 'employee')
('d5', 'Angela', 'Garcia', 'manager')
('p14', 'Angelina ', 'Joly', 'employee')
('p9', 'Bambino', 'Angelo', 'employee')
('d2', 'Carlos', 'Villa', 'manager')
('p15', 'Carlos', 'Tevez', 'employee')
('d18', 'Carlsson', 'Levy', 'manager')
('p3', 'Carolin', 'Angelo', 'employee')
('d12', 'Carolin', 'Olsson', 'manager')
('d14', 'Cathlin ', 'Svensson', 'manager')
('d8', 'Charles', 'Chaplin', 'manager')
('d3', 'Christiano', 'Ronaldo', 'manager')
('p18', 'Donald ', 'Angelo', 'employee')
('d9', 'Elisabeth', 'Goran', 'manager')
('d19', 'Emilia', 'Hadgu', 'manager')
```

Inner Join, it select all from salary table and inner join it with employee on the salary code from salary table and employee ID from employee table.



```python
print("=============================================================================")
print("INNER JOIN\n")
# INNER JOIN on salary and employee table
query = """SELECT *
           FROM salary
           INNER JOIN employee
           ON salary.salaryCode = employee.employeeId
           """
cursor.execute(query)
my_result = cursor.fetchall()
for row in my_result:
    print(row)
```

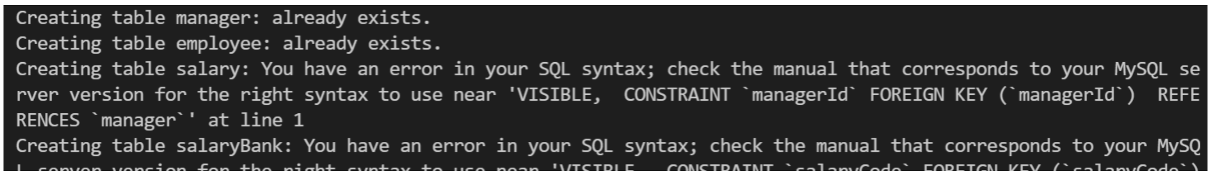


```
==============================================================================
INNER JOIN

(3, 1, '2000', 3, 'Carolin', 'Angelo', 'F', 44, '712417657', 'Gamlastan 123', 'agncar@gmail.com', '3452')
(4, 2, '4300', 4, 'John', 'Smit', 'T', 47, '712417658', 'Gamlastan 123', 'jonsm@gmail.com', '3567')
(5, 3, '3452', 5, 'Adam', 'Carlo', 'M', 78, '740974798', 'Sköndalsvagen 12', 'crls@hotmail.com', '4200')
(6, 4, '3567', 6, 'Sone ', 'Melisa', 'F', 24, '743289858', 'Vällingby Vägen 34', 'mson@gmail.com', '5000')
(7, 5, '4200', 7, 'Abdu ', 'Kiar', 'M', 39, '787943167', 'Vingåkergatan 236', 'kirab@yahoo.com', '3000')
(9, 7, '3000', 9, 'Bambino', 'Angelo', 'M', 46, '82378899', 'Gamlavägen 4238', 'bamang@gmail.com', '3400')
(10, 8, '5000', 10, 'Rachel', 'Gambo', 'F', 62, '734627978', 'Hanningevägen', 'gamrac@hotmail.com', '4000')
(12, 10, '4000', 12, 'Kaleb ', 'Jonay', 'M', 26, '2907523787', 'Kings alley 2389', 'kbal32@gmail.com', '3000')
(13, 11, '2000', 13, 'John ', 'Carlo', 'M', 36, '74265898', 'Gustavvasa 45', 'carjg@yahoo.com', '2300')
(14, 12, '3000', 14, 'Angelina ', 'Joly', 'F', 55, '28379867423', 'America alley 76', 'ajng@gmail.com', '4000')
(15, 13, '2300', 15, 'Carlos', 'Tevez', 'M', 40, '428390753', 'Argentinagatan 523', 'teacd@gmail.com', '4000')
(16, 14, '4000', 16, 'Alhagi ', 'Diouf', 'M', 45, '5458985345', 'Morocco gatan43', 'alhagi@gmail.com', '5400')
(17, 15, '4000', 17, 'Kim', 'Jong Um', 'M', 44, '5398523879', 'Koreagatan', 'kfaksj@gmail.com', '3000')
(18, 16, '5400', 18, 'Donald ', 'Angelo', 'M', 55, '238798238', 'America gatan ', 'fjkljfds@gmail.com', '23000'
)
(20, 18, '23000', 20, 'Ribeka', 'Angelo', 'F', 67, '52389080', 'Rbeca gatan 23', 'fdsfjla@gmail.com', '4000')
(21, 19, '3000', 21, 'Emilia ', 'Dibo', 'F', 43, '5.24E+11', 'Eiliosvägen', 'kfsah@gmail.com', '4000')
PS C:\Users\issac\Documents\prog assing 2>
```

1. Discussion and Resources

This illustration is based on an immaginary scenario, just for the purpose of this report.

In the course of this project we faced some issues. One of which was when inserting data into the various tables, repeatedly, we got the error message that “you have some error in your SQL syntax”. This posed some difficulties to import the data or doing any query for some reasons, until we identified where the source of the problem was and resolved it.



```
Creating table manager: already exists.
Creating table employee: already exists.
Creating table salary: You have an error in your SQL syntax; check the manual that corresponds to your MySQL se
rver version for the right syntax to use near 'VISIBLE,  CONSTRAINT `managerId` FOREIGN KEY (`managerId`)  REFE
RENCES `manager`' at line 1
Creating table salaryBank: You have an error in your SQL syntax; check the manual that corresponds to your MySQ
```

Another itche we encountered was error promptings when carrying out the query. The error message was:

The youtube videos

https://www.youtube.com/watch?v=hRsoq1rfb6I

Changelog

Person	Task	Date
William	Partial the code and report	2022-03-15
Philomina	Partial the code and report	2022-03-15