**Onlinesales.ai**

## **Task-1 SQL**

In the attachment above, use each worksheet as a table in a relational database and write an SQL query that generates the output report.

Answer:

SELECT DEPT\_NAME, AVG(salary) AS "AVG\_MONTHLY\_SALARY (USD)"

FROM (

SELECT department\_name AS DEPT\_NAME, salary

FROM finance

UNION ALL

SELECT department\_name, salary

FROM it

UNION ALL

SELECT department\_name, salary

FROM sales

UNION ALL

SELECT department\_name, salary

FROM hr

UNION ALL

SELECT department\_name, salary

FROM ops

) AS combined\_tables

GROUP BY DEPT\_NAME

ORDER BY "AVG\_MONTHLY\_SALARY (USD)" DESC

LIMIT 3;

## **Task-2 Scripting**

With the same attachment, use each worksheet as a CSV file and write a script (Bash or Python) that generates the same report. Data is to be read from the CSV files not from a database.

Answer:

import pandas as pd

# Define the list of department names and their corresponding CSV files

departments = [

{'name': 'finance', 'file': 'finance.csv'},

{'name': 'it', 'file': 'it.csv'},

{'name': 'sales', 'file': 'sales.csv'},

{'name': 'hr', 'file': 'hr.csv'},

{'name': 'ops', 'file': 'ops.csv'}

]

# Function to calculate average monthly salary for a department

def calculate\_average\_salary(file):

df = pd.read\_csv(file)

return df['salary'].mean()

# Read CSV files, calculate average salaries, and store in a dictionary

department\_salaries = {}

for department in departments:

file = department['file']

average\_salary = calculate\_average\_salary(file)

department\_salaries[department['name']] = average\_salary

# Sort departments by average salary in descending order and get top 3

top\_departments = sorted(department\_salaries.items(), key=lambda x: x[1], reverse=True)[:3]

# Print the report

print("DEPT\_NAME")

print("AVG\_MONTHLY\_SALARY (USD)")

for department, salary in top\_departments:

print(department)

print(f"{salary:.2f}\n")

## **Task-3 Debugging**

Given below is a Bash / Python script that performs following computation on an integer input (n):

1. If n is less than 10: Calculate its Square
   1. Example: 4 => 16
2. If n is between 10 and 20: Calculate the factorial of (n-10)
   1. Example: 15 => 120
3. If n is greater than 20: Calculate the sum of all integers between 1 and (n-20)
   1. Example: 25 => 15

The task is to identify the bugs in the script, fix them and share the new script. Only one of the two scripts required Bash or Python. **Hint**: You can correct the script by only changing 3-4 characters.

Answer:

def compute(n):

if n < 10:

# For n less than 10, calculate its square

out = n \*\* 2

elif n < 20:

# For n between 10 and 20, calculate factorial of (n-10)

out = 1

for i in range(1, n - 9):

out \*= i

else:

# For n greater than 20, calculate the sum of all integers between 1 and (n-20)

out = sum(range(1, n - 19))

return out

n = int(input("Enter an integer: "))

result = compute(n)

print("Result:", result)

To Run this script install a python interpretor and click on RUN or press ctrl + f5 to run the above file.

All the above three tasks have been completed and tested successfully without any errors.