**Raiymbek Meirambek**

**Assignment 2**

**Executive Summary**

In this project, I used Python with the SQLAlchemy library to interact with a MySQL database for a caregiving service. Initially, I established a connection to the database and set up a session to execute SQL queries conveniently.

Throughout the code, I performed a variety of operations:

* Updating User Information:

The code updates the phone number for a specific user, "Askar Askarov," in the USER table. This ensures that the user's contact information is accurate.

Additionally, the code increases the hourly rate for caregivers who currently earn less than 9. This adjustment aims to reflect a fair compensation for their services.

* Deleting Jobs and Users:

The code deletes jobs posted by a user named "Bolat Bolatov" from the JOB table. This action might be necessary for various reasons, such as if the user has withdrawn the job postings or if they are no longer valid.

Furthermore, the code removes users who reside on Turan street and are classified as members. This operation is performed in the USER, MEMBER, and ADDRESS tables, ensuring that outdated or irrelevant user information is removed.

* Retrieving and Presenting Information:

Information about confirmed appointments is retrieved and presented. This includes details such as appointment IDs, caregiver names, member names, and the status of the appointments. The APPOINTMENT, CAREGIVER, MEMBER, and USER tables are likely involved in this operation.

The code lists job IDs for positions that have specific requirements, such as containing the term 'gentle' in their other requirements. This query is executed on the JOB table.

Work hours for babysitter positions are displayed by querying the JOB and APPOINTMENT tables based on the required caregiving type.

* Aggregation Queries:

The code performs several aggregation queries:

* It counts the number of applicants for each job posted by a member. This involves a left join between the JOB and JOB\_APPLICATION tables, counting the number of applications for each job.
* The total hours spent by caregivers for all confirmed appointments are calculated. This involves a join between the APPOINTMENT and CAREGIVER tables, summing the work hours for each caregiver.
* The average pay for caregivers based on accepted appointments is determined. This involves joining the APPOINTMENT and CAREGIVER tables, calculating the average hourly rate for caregivers with confirmed appointments.
* Caregivers earning above the calculated average pay based on accepted appointments are identified. This involves a nested query to compare individual caregivers' hourly rates with the calculated average.
* Data Analysis and Reporting:

The code conducts data analysis and reporting by calculating the total cost to pay caregivers for all accepted appointments. This involves a join between the APPOINTMENT and CAREGIVER tables, multiplying the work hours by the hourly rate for each caregiver, and summing the results to get the total cost.

One challenging aspect was formulating complex queries involving multiple joins and aggregations. It took about 70% of my time spent for this assignment. Including learning to create views, like job\_applications\_view, was an insightful part of the process. I gained a deeper understanding of database interactions, SQL query construction, and how to use SQLAlchemy effectively in a Python environment.

In conclusion, this project enhanced my skills in database manipulation, SQL querying, and working with SQLAlchemy. It presented challenges that required careful consideration of table relationships and effective query design.