**Final Project Phase 3 Yuqing (Brandy) Huang**

## Question1:

To convert the dataset into tables, I started by creating a new database in MySQL and then designed the necessary tables based on the ERD I had created during Phase 2. Once the tables were ready, I went back to the original dataset, which was in CSV format, and split it into eight smaller CSV files. Each file corresponded to a table in my database, following the relationships and structure outlined in the ERD.

Since some columns in the tables were created to let the data be in normalization and didn't exist in the original dataset, I found this method particularly helpful. By preparing the data in separate CSV files first, I could ensure that each file was properly aligned with the table structures in my database. After completing this step, I imported the CSV files one by one into their respective tables in MySQL, ensuring the data was correctly populated.

## Question2:

*Challenge 1: The Order of Data Import*
One of the first challenges I faced was deciding whether to split the dataset into multiple smaller files first or import the whole dataset into MySQL and separate it later. After considering my options, I decided to break the dataset into smaller CSV files based on my ERD. My original dataset had only 12 columns, which made it relatively simple to handle in Excel. Additionally, since some of the columns (e.g., ISO country codes) were created by me and didn't exist in the original dataset, splitting the data beforehand gave me more flexibility to prepare and organize it in Excel before importing it into MySQL.

*Challenge 2: Combining Country Names with Their ISO Codes*
Another challenge was linking countries with their corresponding ISO 3166-1 country codes. Initially, I planned to manually match and attach the country codes to their respective country names by filtering unique country names from both employee_residence and company_location However, I quickly realized that this approach would be time-consuming and prone to errors. To overcome this, I used an R package to generate a complete list of ISO 3166-1 country codes and their corresponding country names. I saved this data as a CSV file and imported it into the country table in my database, streamlining the process and ensuring accuracy.

*Challenge 3: Populating the Currency Table*
Similarly, for the currency table, I needed ISO 4217 currency codes. Instead of manually retrieving these, I utilized the R package **`rvest`** to scrape currency code data from Wikipedia. This allowed me to create a CSV file with the required currency information, which I then imported into the database.

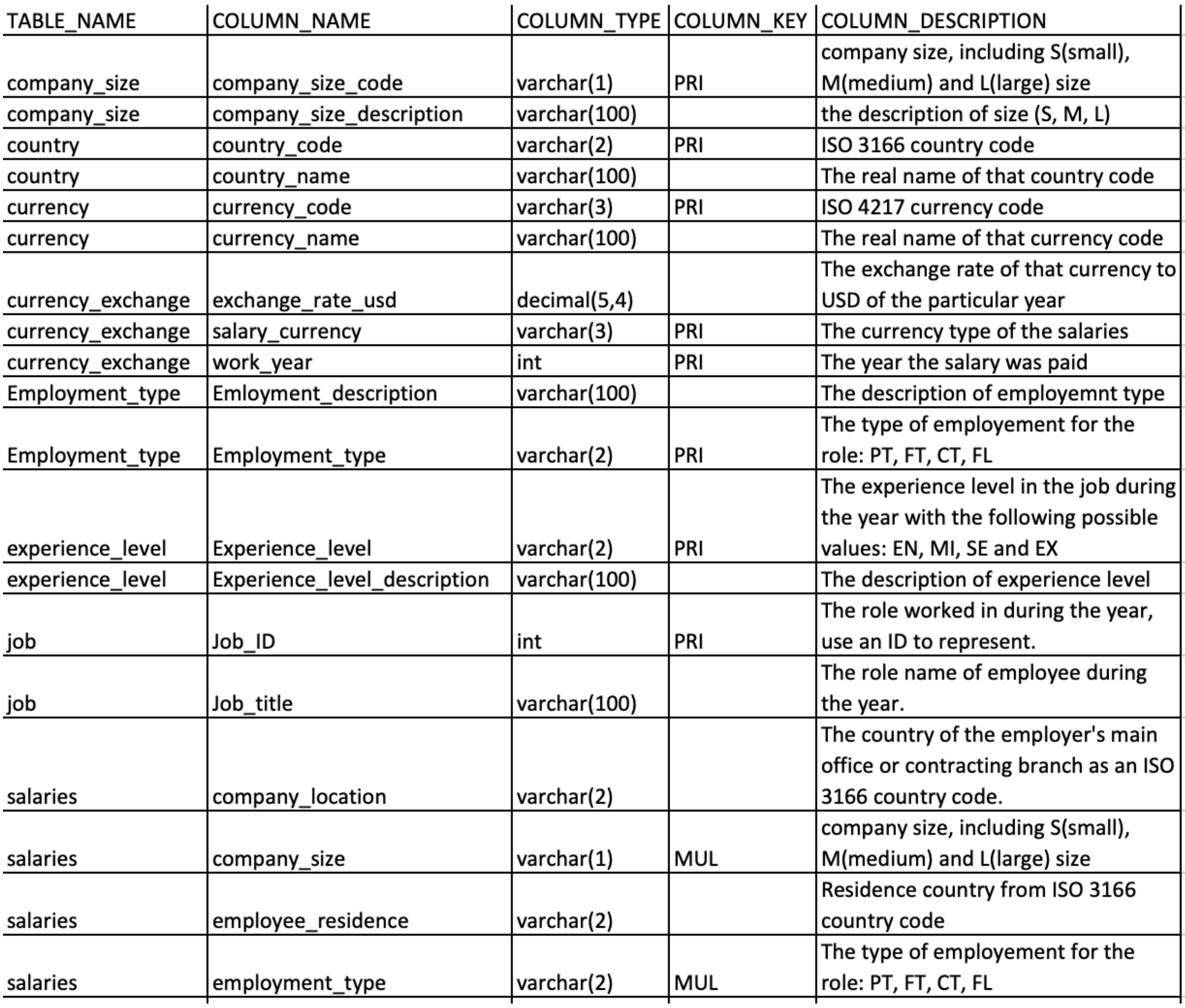
*Challenge 4: Data Format Mismatch*
While importing the country data, I encountered a "data format mismatch" error. The ASCII format MySQL was using only supported English characters, but some country names in my dataset contained non-English characters. To resolve this, I updated the MySQL table settings to a format of UTF8mb4 that supported international characters.

*Challenge 5: Data Import Failures*
At times, MySQL failed to load certain datasets. Through trial and error, I identified three common causes for these failures:

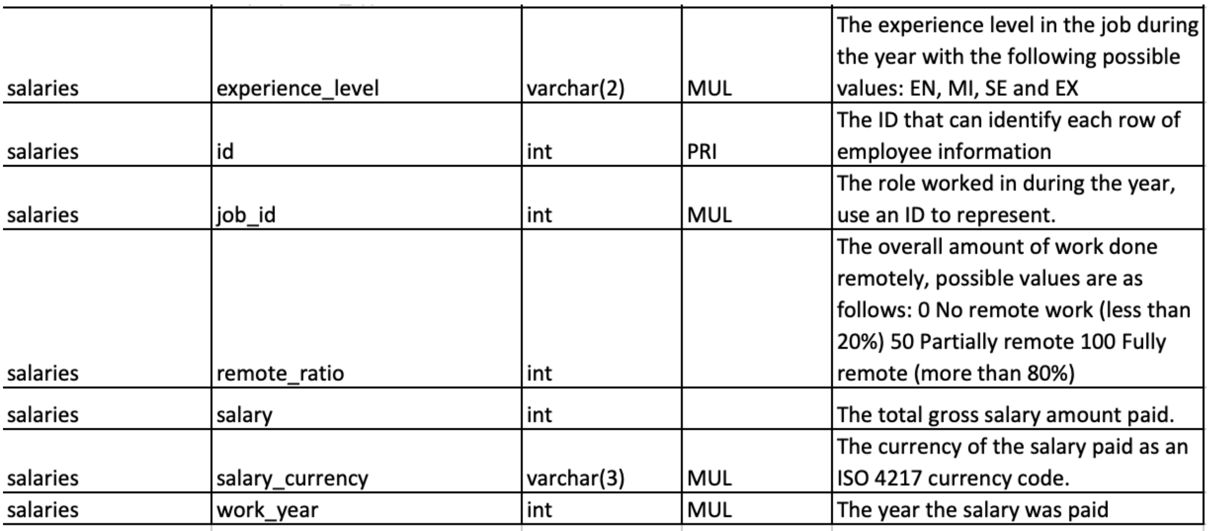
1. The ASCII format of the file didn’t match UTF-8.
2. The number of columns in the dataset didn’t match the table structure.
3. The data exceeded the defined limits for certain columns (e.g., a value was too large for a varchar column).
   To resolve these issues, I systematically checked for mismatches, corrected the dataset or table definitions as needed, and re-imported the data.

*Challenge 6: Adjusting Columns for Primary and Foreign Keys*
Another challenge was creating primary and foreign keys when the columns didn’t exist in the original dataset. For example, in my job table, I created a new job_id column to represent jobs instead of directly storing job_title in the main salaries table. This decision was based on normalization principles to handle duplicate job titles. However, transitioning from job_title in the main table to the new job_id required extra work. To address this, I used a temporary table in MySQL to map and transition the data. This process taught me how to use temporary tables effectively and required disabling safe mode temporarily to complete the operation successfully.

**Question 3**



| TABLE_NAME | COLUMN_NAME | COLUMN_TYPE | COLUMN_KEY | COLUMN_DESCRIPTION |
|---|---|---|---|---|
| company_size | company_size_code | varchar(1) | PRI | company size, including S(small), M(medium) and L(large) size |
| company_size | company_size_description | varchar(100) | | the description of size (S, M, L) |
| country | country_code | varchar(2) | PRI | ISO 3166 country code |
| country | country_name | varchar(100) | | The real name of that country code |
| currency | currency_code | varchar(3) | PRI | ISO 4217 currency code |
| currency | currency_name | varchar(100) | | The real name of that currency code |
| currency_exchange | exchange_rate_usd | decimal(5,4) | | The exchange rate of that currency to USD of the particular year |
| currency_exchange | salary_currency | varchar(3) | PRI | The currency type of the salaries |
| currency_exchange | work_year | int | PRI | The year the salary was paid |
| Employment_type | Emloyment_description | varchar(100) | | The description of employemnt type |
| Employment_type | Employment_type | varchar(2) | PRI | The type of employement for the role: PT, FT, CT, FL |
| experience_level | Experience_level | varchar(2) | PRI | The experience level in the job during the year with the following possible values: EN, MI, SE and EX |
| experience_level | Experience_level_description | varchar(100) | | The description of experience level |
| job | Job_ID | int | PRI | The role worked in during the year, use an ID to represent. |
| job | Job_title | varchar(100) | | The role name of employee during the year. |
| salaries | company_location | varchar(2) | | The country of the employer's main office or contracting branch as an ISO 3166 country code. |
| salaries | company_size | varchar(1) | MUL | company size, including S(small), M(medium) and L(large) size |
| salaries | employee_residence | varchar(2) | | Residence country from ISO 3166 country code |
| salaries | employment_type | varchar(2) | MUL | The type of employement for the role: PT, FT, CT, FL |



| | | | | |
|---|---|---|---|---|
| salaries | experience_level | varchar(2) | MUL | The experience level in the job during the year with the following possible values: EN, MI, SE and EX |
| salaries | id | int | PRI | The ID that can identify each row of employee information |
| salaries | job_id | int | MUL | The role worked in during the year, use an ID to represent. |
| salaries | remote_ratio | int | | The overall amount of work done remotely, possible values are as follows: 0 No remote work (less than 20%) 50 Partially remote 100 Fully remote (more than 80%) |
| salaries | salary | int | | The total gross salary amount paid. |
| salaries | salary_currency | varchar(3) | MUL | The currency of the salary paid as an ISO 4217 currency code. |
| salaries | work_year | int | MUL | The year the salary was paid |

## Question 4

Here are 8 business questions

1. What are the total salaries paid by each company size (small, medium, large) across different countries?

2. Which job titles have the highest average salaries for employees working fully remotely (remote_ratio = 100)?

3. What is the distribution of employment types (full-time, part-time, etc.) in different experience levels across various countries?

4. What is the average exchange rate of salary currencies to USD for each year in the currency_exchange table, and how does it vary across time?

5. For each country, which employment type has the highest percentage of senior-level employees (experience_level = 'SE')?

6. What are the top 5 job titles with the highest salaries in companies located in countries with a specific ISO currency code?

7. Which country has the most diverse company sizes (small, medium, large) and what are the average salaries for each size within that country?

8. Identify the job titles that consistently appear in the top 10 highest-paying jobs based on work_year.