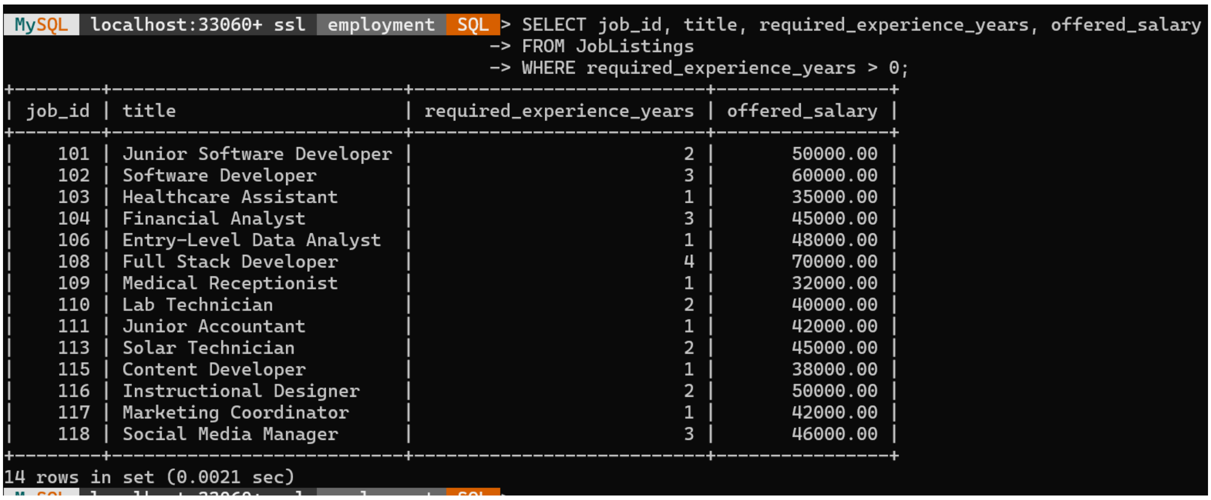
SQL Programming

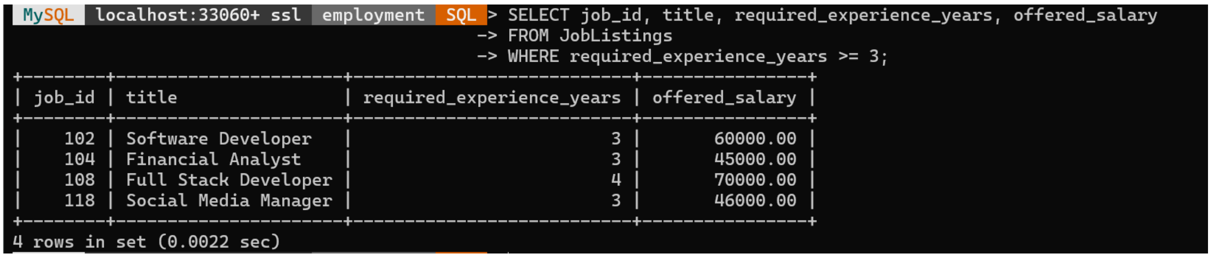
Data Retrieval Queries

1. **Find “Entry-Level” Job Listings That Require More Than 0 Years of Experience** (i.e., jobs that paradoxically require experience even though they are entry-level)



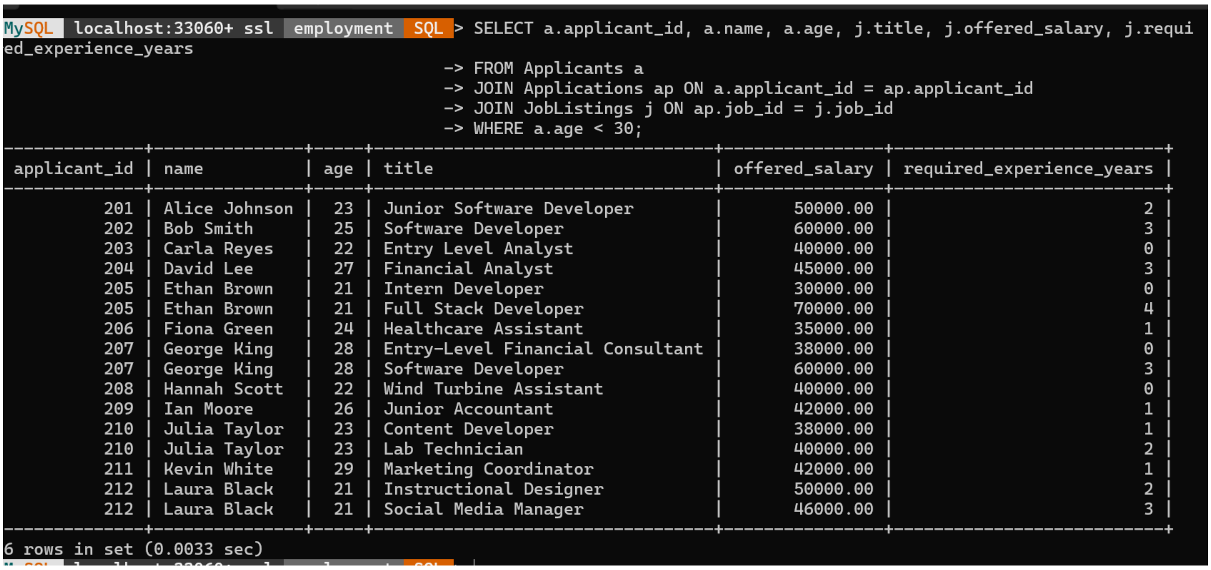
```
MySQL  localhost:33060+ ssl  employment  SQL > SELECT job_id, title, required_experience_years, offered_salary
                                            -> FROM JobListings
                                            -> WHERE required_experience_years > 0;
+--------+---------------------------+---------------------------+----------------+
| job_id | title                     | required_experience_years | offered_salary |
+--------+---------------------------+---------------------------+----------------+
|    101 | Junior Software Developer |                         2 |       50000.00 |
|    102 | Software Developer        |                         3 |       60000.00 |
|    103 | Healthcare Assistant      |                         1 |       35000.00 |
|    104 | Financial Analyst         |                         3 |       45000.00 |
|    106 | Entry-Level Data Analyst  |                         1 |       48000.00 |
|    108 | Full Stack Developer      |                         4 |       70000.00 |
|    109 | Medical Receptionist      |                         1 |       32000.00 |
|    110 | Lab Technician            |                         2 |       40000.00 |
|    111 | Junior Accountant         |                         1 |       42000.00 |
|    113 | Solar Technician          |                         2 |       45000.00 |
|    115 | Content Developer         |                         1 |       38000.00 |
|    116 | Instructional Designer    |                         2 |       50000.00 |
|    117 | Marketing Coordinator     |                         1 |       42000.00 |
|    118 | Social Media Manager      |                         3 |       46000.00 |
+--------+---------------------------+---------------------------+----------------+
14 rows in set (0.0021 sec)
```

2. Identify Jobs Requiring 3+ Years of Experience (Unrealistic for Youth)



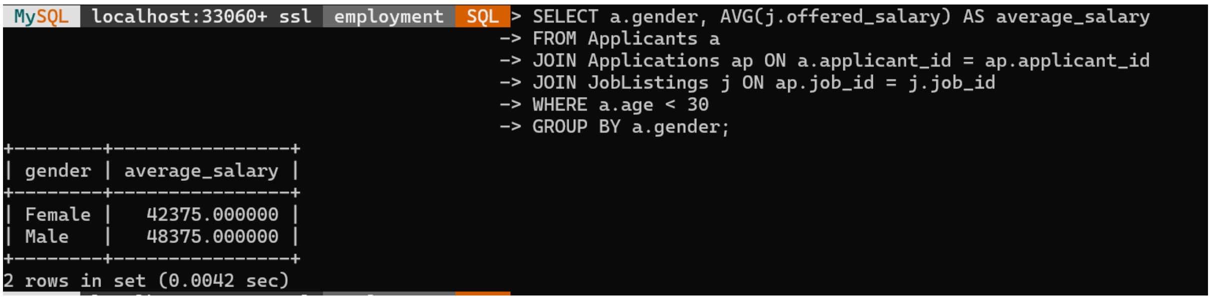
```
MySQL  localhost:33060+ ssl  employment  SQL > SELECT job_id, title, required_experience_years, offered_salary
                                            -> FROM JobListings
                                            -> WHERE required_experience_years >= 3;
+--------+----------------------+---------------------------+----------------+
| job_id | title                | required_experience_years | offered_salary |
+--------+----------------------+---------------------------+----------------+
|    102 | Software Developer   |                         3 |       60000.00 |
|    104 | Financial Analyst    |                         3 |       45000.00 |
|    108 | Full Stack Developer |                         4 |       70000.00 |
|    118 | Social Media Manager |                         3 |       46000.00 |
+--------+----------------------+---------------------------+----------------+
4 rows in set (0.0022 sec)
```

3. List All Applications by Young Applicants (Age < 30) with Their Job Titles and Offered Salaries



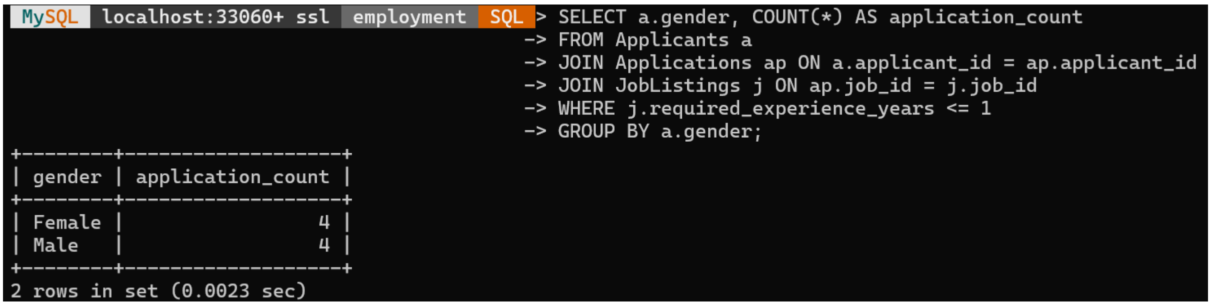
```
MySQL  localhost:33060+ ssl  employment  SQL > SELECT a.applicant_id, a.name, a.age, j.title, j.offered_salary, j.requi
ed_experience_years
                                                 -> FROM Applicants a
                                                 -> JOIN Applications ap ON a.applicant_id = ap.applicant_id
                                                 -> JOIN JobListings j ON ap.job_id = j.job_id
                                                 -> WHERE a.age < 30;
+--------------+---------------+-----+-----------------------------------+----------------+----------------------------+
| applicant_id | name          | age | title                             | offered_salary | required_experience_years |
+--------------+---------------+-----+-----------------------------------+----------------+----------------------------+
|          201 | Alice Johnson |  23 | Junior Software Developer         |       50000.00 |                          2 |
|          202 | Bob Smith     |  25 | Software Developer                |       60000.00 |                          3 |
|          203 | Carla Reyes   |  22 | Entry Level Analyst               |       40000.00 |                          0 |
|          204 | David Lee     |  27 | Financial Analyst                 |       45000.00 |                          3 |
|          205 | Ethan Brown   |  21 | Intern Developer                  |       30000.00 |                          0 |
|          205 | Ethan Brown   |  21 | Full Stack Developer              |       70000.00 |                          4 |
|          206 | Fiona Green   |  24 | Healthcare Assistant              |       35000.00 |                          1 |
|          207 | George King   |  28 | Entry-Level Financial Consultant  |       38000.00 |                          0 |
|          207 | George King   |  28 | Software Developer                |       60000.00 |                          3 |
|          208 | Hannah Scott  |  22 | Wind Turbine Assistant            |       40000.00 |                          0 |
|          209 | Ian Moore     |  26 | Junior Accountant                 |       42000.00 |                          1 |
|          210 | Julia Taylor  |  23 | Content Developer                 |       38000.00 |                          1 |
|          210 | Julia Taylor  |  23 | Lab Technician                    |       40000.00 |                          2 |
|          211 | Kevin White   |  29 | Marketing Coordinator             |       42000.00 |                          1 |
|          212 | Laura Black   |  21 | Instructional Designer            |       50000.00 |                          2 |
|          212 | Laura Black   |  21 | Social Media Manager              |       46000.00 |                          3 |
+--------------+---------------+-----+-----------------------------------+----------------+----------------------------+
6 rows in set (0.0033 sec)
```

4. Analyze Average Salary by Gender for Youth Applicants (Age < 30) *(Assuming that the salary from the job is the main indicator even though multiple applications may exist)*



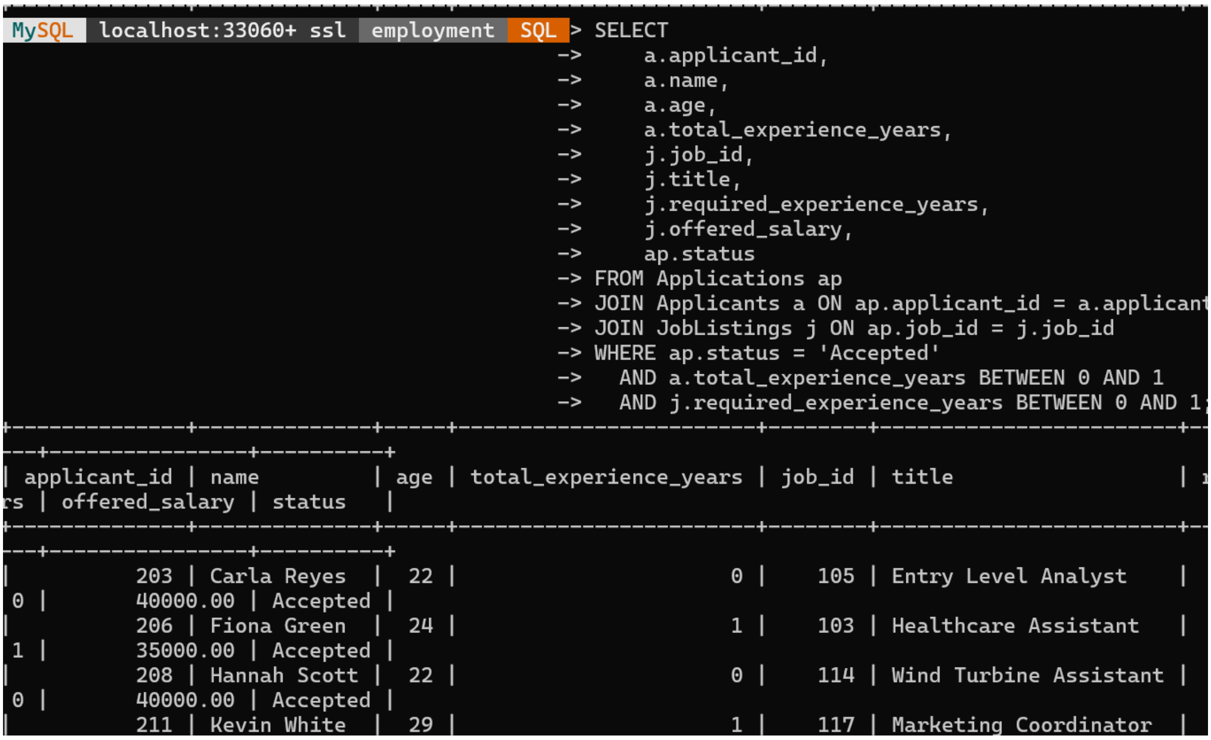
```
 MySQL  localhost:33060+ ssl  employment  SQL > SELECT a.gender, AVG(j.offered_salary) AS average_salary
                                             -> FROM Applicants a
                                             -> JOIN Applications ap ON a.applicant_id = ap.applicant_id
                                             -> JOIN JobListings j ON ap.job_id = j.job_id
                                             -> WHERE a.age < 30
                                             -> GROUP BY a.gender;
+--------+----------------+
| gender | average_salary |
+--------+----------------+
| Female |   42375.000000 |
| Male   |   48375.000000 |
+--------+----------------+
2 rows in set (0.0042 sec)
```

5. Count of Applications by Gender for Entry-Level Jobs Requiring Minimal Experience (0 or 1 year)



```
 MySQL  localhost:33060+ ssl  employment  SQL > SELECT a.gender, COUNT(*) AS application_count
                                             -> FROM Applicants a
                                             -> JOIN Applications ap ON a.applicant_id = ap.applicant_id
                                             -> JOIN JobListings j ON ap.job_id = j.job_id
                                             -> WHERE j.required_experience_years <= 1
                                             -> GROUP BY a.gender;
+--------+-------------------+
| gender | application_count |
+--------+-------------------+
| Female |                 4 |
| Male   |                 4 |
+--------+-------------------+
2 rows in set (0.0023 sec)
```

6. Accepted applications for entry-level jobs where the youth have minimal work experience.



```
 MySQL  localhost:33060+ ssl  employment  SQL > SELECT
                                            ->     a.applicant_id,
                                            ->     a.name,
                                            ->     a.age,
                                            ->     a.total_experience_years,
                                            ->     j.job_id,
                                            ->     j.title,
                                            ->     j.required_experience_years,
                                            ->     j.offered_salary,
                                            ->     ap.status
                                            -> FROM Applications ap
                                            -> JOIN Applicants a ON ap.applicant_id = a.applicant
                                            -> JOIN JobListings j ON ap.job_id = j.job_id
                                            -> WHERE ap.status = 'Accepted'
                                            ->   AND a.total_experience_years BETWEEN 0 AND 1
                                            ->   AND j.required_experience_years BETWEEN 0 AND 1;
+--------------+--------------+-----+------------------------+--------+------------------------+--
---+----------------+----------+
| applicant_id | name         | age | total_experience_years | job_id | title                  | r
rs | offered_salary | status   |
+--------------+--------------+-----+------------------------+--------+------------------------+--
---+----------------+----------+
|          203 | Carla Reyes  |  22 |                      0 |    105 | Entry Level Analyst    |
 0 |       40000.00 | Accepted |
|          206 | Fiona Green  |  24 |                      1 |    103 | Healthcare Assistant   |
 1 |       35000.00 | Accepted |
|          208 | Hannah Scott |  22 |                      0 |    114 | Wind Turbine Assistant |
 0 |       40000.00 | Accepted |
|          211 | Kevin White  |  29 |                      1 |    117 | Marketing Coordinator  |
```

Part 5: Integration and Testing

Integration: Importing Data into Excel

To ensure that my analysis was both comprehensive and accurate, I carefully integrated the data from my SQL database into Microsoft Excel. Here's how I did it:

1. **Data Export from the SQL Database:**
   - **I executed** several SQL queries to extract the necessary data (e.g., job listings, applicant details, and application outcomes) from my relational database.
   - **I exported** the query results to CSV files using my database management tool. This allowed me to work with a clean, flat-file format that could be easily imported into Excel.
2. **Importing Data into Excel:**
   - **I opened** Microsoft Excel and navigated to the **Data** tab, selecting **Get Data > From Text/CSV** to import the CSV files.
   - **During the import process,** I verified that each column (dates, numbers, text, etc.) was correctly recognized. I adjusted the data type settings where necessary to ensure that all data was imported accurately.
   - **I then loaded** the data into my Excel workbook, which served as the foundation for my pivot tables and dashboard visualizations.
3. **Establishing Data Connections and Refresh Settings:**
   - **I set up connections** between the Excel workbook and the CSV files so that my data remained linked to the source files.

    - **I configured** the refresh settings (under **Data > Queries & Connections**) to automatically update the workbook whenever the source data changed, ensuring that my analysis was always current.

Testing for Data Consistency

After importing the data, I performed several tests to ensure its consistency and accuracy:

1. **Data Validation Checks:**
    - **Row Count Comparison:** I compared the number of rows in the imported Excel tables against the row counts from my SQL queries to ensure no data was lost during the export/import process.
    - **Field Consistency:** I manually reviewed key columns—such as salary figures, required experience years, and applicant IDs—to confirm that data types and values remained consistent.
    - **Sample Verification:** I cross-checked a few sample records between the SQL output and the Excel data to confirm that the export/import process did not introduce any errors.
2. **Pivot Table and Chart Verification:**
    - **I created and refreshed pivot tables** in Excel to summarize the data (e.g., counting job listings by experience level, calculating average salaries, etc.).
    - **I verified** that the summarized figures matched my expectations and the original SQL query results.
    - **I tested** the interactivity of any slicers or filters on my Excel dashboard to ensure they correctly updated the visualizations based on the underlying data.
3. **User Acceptance Testing (UAT):**
    - **I invited feedback** from a peer to review both the imported data and the functionality of the Excel dashboard.
    - **I simulated various scenarios,** such as adding new job listings or updating applicant data, to ensure that the Excel workbook could accommodate changes and refresh accordingly.

Documentation

Throughout the integration and testing process, I maintained detailed documentation, including:

- Step-by-step screenshots of the import process in Excel.
- Notes on any data type adjustments made during the import.
- A summary of validation checks and any discrepancies identified, along with their resolutions.