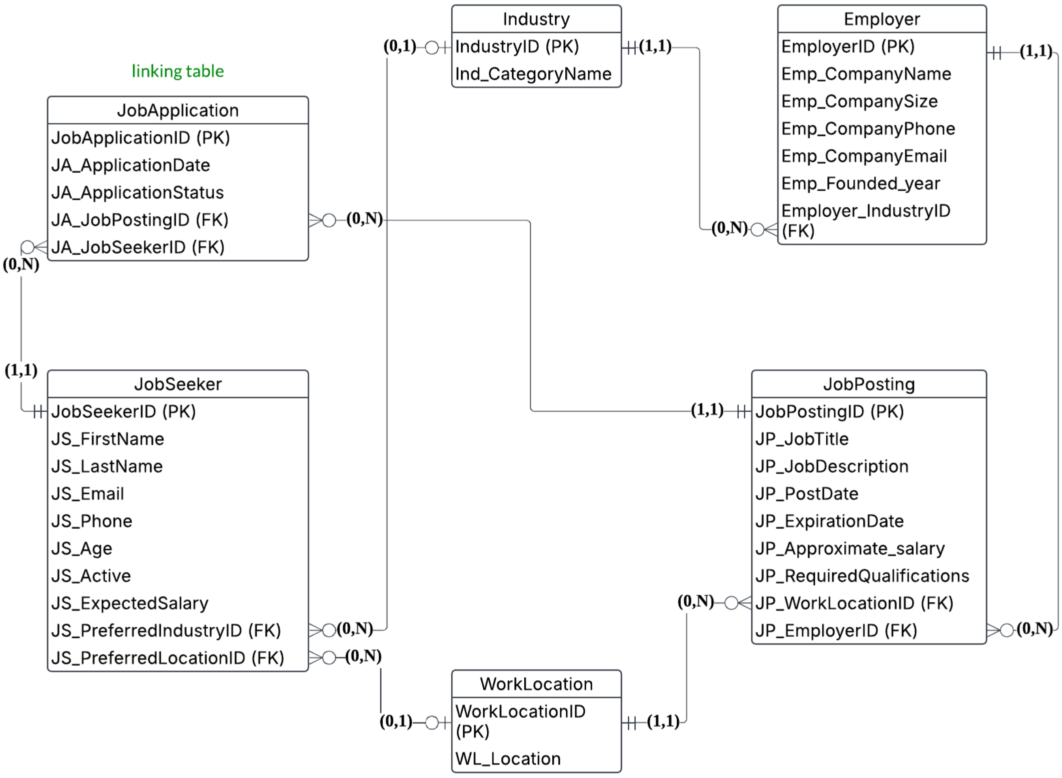
< SQL Relational Database Project >

Project: Job Search Database

- **Introduction** :
  1. Brief introduction to the project:
     The purpose of the Job Search Database is to maintain the data that supports job seekers in finding employment opportunities and facilitates employer recruitment efforts.
  2. Team member: Angel Huang
  3. Location of the database on Akira: Angel Huang(under DB3)

- **Mission Statement and Mission Objectives:**
  1. Maintain complete employer information, including company name, industry category, company size, and location.
  2. Classify employers by industry category to support industry-specific job searches.
  3. Maintain job posting details, including job title, job description, salary range, work location, and required qualifications.
  4. Maintain job location details to support regional and remote job searches.
  5. Maintain job seeker information, including name, contact details, preferred job industries, and desired work location, to facilitate personalized job matching.

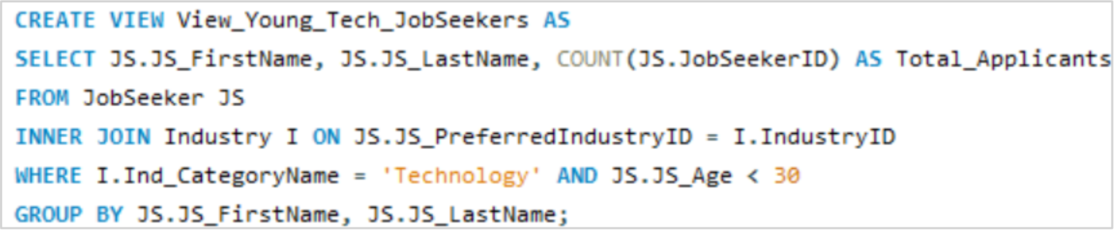
- **Conceptual Design:**
  - ERD:
    https://docs.google.com/document/d/15vEyA7_h99XOShBOnLxSTGYiwPy6MAx9eIlyE5kQs4g/edit?usp=sharing



- **Logical design**:
  - Data Dictionary Table: https://docs.google.com/document/d/1TP7lpekZ9EwtzlVPLMpRIlkXGFesL_J_TNlllW2F1Ms/edit?usp=sharing
  - Business rules: https://docs.google.com/document/d/12aY8GnXByMpV0OQv0sMbbRMb7ojnQlHtVwElLJwk7I4/edit?usp=sharing

- **Queries:**

1. Find job seekers whose preferred industry is 'Technology' and age is below 30, along with their count:

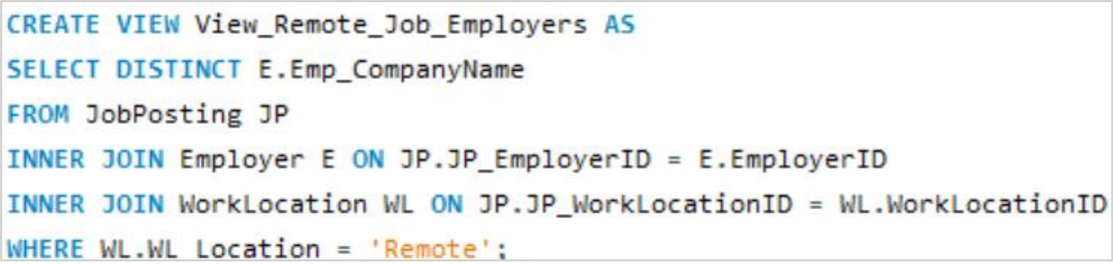


```
CREATE VIEW View_Young_Tech_JobSeekers AS
SELECT JS.JS_FirstName, JS.JS_LastName, COUNT(JS.JobSeekerID) AS Total_Applicants
FROM JobSeeker JS
INNER JOIN Industry I ON JS.JS_PreferredIndustryID = I.IndustryID
WHERE I.Ind_CategoryName = 'Technology' AND JS.JS_Age < 30
GROUP BY JS.JS_FirstName, JS.JS_LastName;
```

```
CREATE VIEW View_Young_Tech_JobSeekers AS
SELECT JS.JS_FirstName, JS.JS_LastName, COUNT(JS.JobSeekerID)
```

```
AS Total_Applicants
FROM JobSeeker JS
INNER JOIN Industry I ON JS.JS_PreferredIndustryID = I.IndustryID
WHERE I.Ind_CategoryName = 'Technology' AND JS.JS_Age < 30
GROUP BY JS.JS_FirstName, JS.JS_LastName;
```

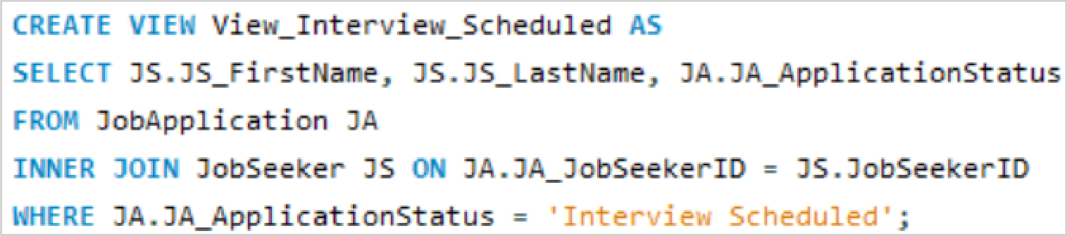
2. Retrieve employer names for job postings that require 'Remote' work location:



```
CREATE VIEW View_Remote_Job_Employers AS
SELECT DISTINCT E.Emp_CompanyName
FROM JobPosting JP
INNER JOIN Employer E ON JP.JP_EmployerID = E.EmployerID
INNER JOIN WorkLocation WL ON JP.JP_WorkLocationID = WL.WorkLocationID
WHERE WL.WL_Location = 'Remote';
```

```
CREATE VIEW View_Remote_Job_Employers AS
SELECT DISTINCT E.Emp_CompanyName
FROM JobPosting JP
INNER JOIN Employer E ON JP.JP_EmployerID = E.EmployerID
INNER JOIN WorkLocation WL ON JP.JP_WorkLocationID =
WL.WorkLocationID
WHERE WL.WL_Location = 'Remote';
```

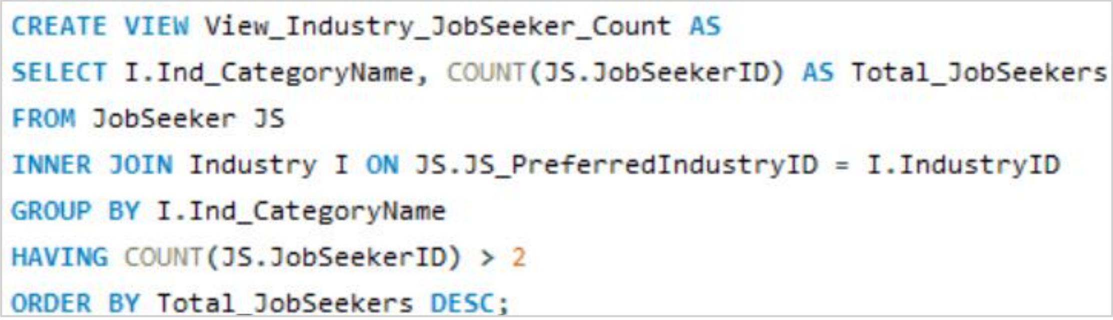
3. Retrieve job seekers whose application status is 'Interview Scheduled', including application status:



```
CREATE VIEW View_Interview_Scheduled AS
SELECT JS.JS_FirstName, JS.JS_LastName, JA.JA_ApplicationStatus
FROM JobApplication JA
INNER JOIN JobSeeker JS ON JA.JA_JobSeekerID = JS.JobSeekerID
WHERE JA.JA_ApplicationStatus = 'Interview Scheduled';
```

```
CREATE VIEW View_Interview_Scheduled AS
SELECT JS.JS_FirstName, JS.JS_LastName, JA.JA_ApplicationStatus
FROM JobApplication JA
INNER JOIN JobSeeker JS ON JA.JA_JobSeekerID = JS.JobSeekerID
WHERE JA.JA_ApplicationStatus = 'Interview Scheduled';
```

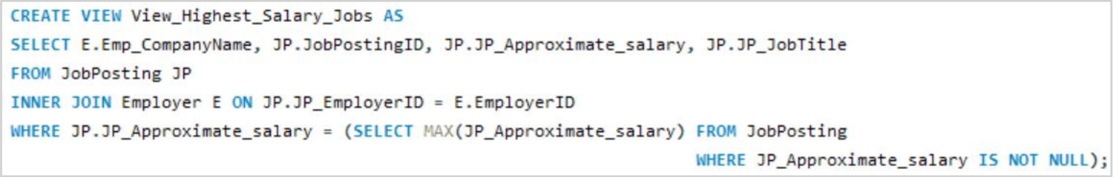
4. Count job seekers in each industry, only showing industries with more than 2 job seekers:



```
CREATE VIEW View_Industry_JobSeeker_Count AS
SELECT I.Ind_CategoryName, COUNT(JS.JobSeekerID) AS Total_JobSeekers
FROM JobSeeker JS
INNER JOIN Industry I ON JS.JS_PreferredIndustryID = I.IndustryID
GROUP BY I.Ind_CategoryName
HAVING COUNT(JS.JobSeekerID) > 2
ORDER BY Total_JobSeekers DESC;
```

```
CREATE VIEW View_Industry_JobSeeker_Count AS
SELECT I.Ind_CategoryName, COUNT(JS.JobSeekerID) AS
Total_JobSeekers
FROM JobSeeker JS
INNER JOIN Industry I ON JS.JS_PreferredIndustryID = I.IndustryID
GROUP BY I.Ind_CategoryName
HAVING COUNT(JS.JobSeekerID) > 2
ORDER BY Total_JobSeekers DESC;
```

5. Retrieve the highest salary job in Job Posting and the corresponding job title, Company Name and salary:



```
CREATE VIEW View_Highest_Salary_Jobs AS
SELECT E.Emp_CompanyName, JP.JobPostingID, JP.JP_Approximate_salary, JP.JP_JobTitle
FROM JobPosting JP
INNER JOIN Employer E ON JP.JP_EmployerID = E.EmployerID
WHERE JP.JP_Approximate_salary = (SELECT MAX(JP_Approximate_salary) FROM JobPosting
                                                                  WHERE JP_Approximate_salary IS NOT NULL);
```

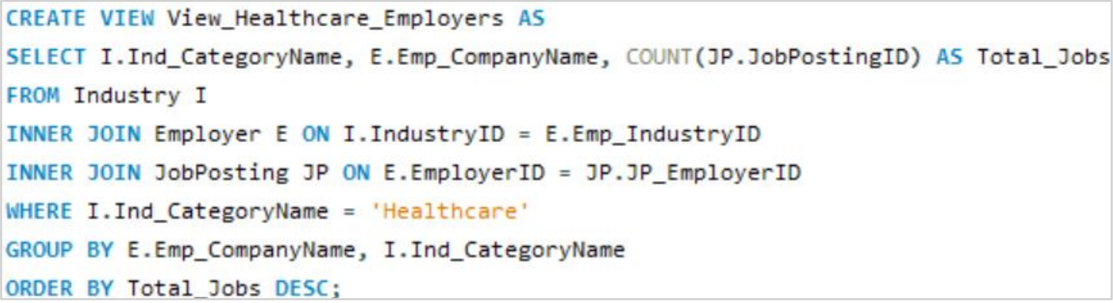
```
CREATE VIEW View_Highest_Salary_Jobs AS
SELECT E.Emp_CompanyName, JP.JobPostingID,
JP.JP_Approximate_salary, JP.JP_JobTitle
FROM JobPosting JP
INNER JOIN Employer E ON JP.JP_EmployerID = E.EmployerID
WHERE JP.JP_Approximate_salary = (SELECT
MAX(JP_Approximate_salary) FROM JobPosting WHERE
JP_Approximate_salary IS NOT NULL);
```

6. Retrieve employers with more than 1 job posting:

```
CREATE VIEW View_Employers_With_Multiple_Jobs AS
SELECT E.Emp_CompanyName, COUNT(JP.JobPostingID) AS Job_Count
FROM Employer E
LEFT JOIN JobPosting JP ON E.EmployerID = JP.JP_EmployerID
GROUP BY E.Emp_CompanyName
HAVING COUNT(JP.JobPostingID) > 1
ORDER BY Job_Count DESC;
```

```
CREATE VIEW View_Employers_With_Multiple_Jobs AS
SELECT E.Emp_CompanyName, COUNT(JP.JobPostingID) AS
Job_Count
FROM Employer E
LEFT JOIN JobPosting JP ON E.EmployerID = JP.JP_EmployerID
GROUP BY E.Emp_CompanyName
HAVING COUNT(JP.JobPostingID) > 1
ORDER BY Job_Count DESC;
```

7. Retrieve all employers in the 'Healthcare' industry and the number of job postings they have:



```
CREATE VIEW View_Healthcare_Employers AS
SELECT I.Ind_CategoryName, E.Emp_CompanyName, COUNT(JP.JobPostingID) AS Total_Jobs
FROM Industry I
INNER JOIN Employer E ON I.IndustryID = E.Emp_IndustryID
INNER JOIN JobPosting JP ON E.EmployerID = JP.JP_EmployerID
WHERE I.Ind_CategoryName = 'Healthcare'
GROUP BY E.Emp_CompanyName, I.Ind_CategoryName
ORDER BY Total_Jobs DESC;
```

```
CREATE VIEW View_Healthcare_Employers AS
SELECT I.Ind_CategoryName, E.Emp_CompanyName,
COUNT(JP.JobPostingID) AS Total_Jobs
FROM Industry I
INNER JOIN Employer E ON I.IndustryID = E.Emp_IndustryID
INNER JOIN JobPosting JP ON E.EmployerID = JP.JP_EmployerID
WHERE I.Ind_CategoryName = 'Healthcare'
GROUP BY E.Emp_CompanyName, I.Ind_CategoryName
ORDER BY Total_Jobs DESC;
```