INTERNATIONAL UNIVERSITY

School of Computer Science and Engineering

|  |  |
| --- | --- |
| PROJECT REPORT  Instructor: Dr.Nguyen Thi Thuy Loan | TOPIC 17: EMPLOYMENT AGENCY  Author:  Võ Văn Việt,  Lê Trần Minh Hoàng,  Lê Bảo Phúc.  Principles of Database Management |

**Contribution of members**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **ID** | **Proportion** | **Work** |
| Võ Văn Việt | ITDSIU18043 | 50% | Relational Model, JDBC. |
| Lê Trần Minh Hoàng | ITDSIU18042 | 30% | Query. |
| Lê Bảo Phúc | ITDSIU18033 | 20% | Relational Algebra. |

**Table of Content**

[**1. Overview - Motivation:**](#_heading=h.gc1az4rsj7fn) **4**

[**2. Analyzing the requirement:**](#_heading=h.xb6t39e8p8hl) **4**

[**3. ER Diagram:**](#_heading=h.u8ldy5q94xi8) **5**

[**4. Relational Model:**](#_heading=h.a2dg6zqylr6y) **7**

[**5. Conceptual Schema:**](#_heading=h.1qk9wtifs3up) **11**

[**6. Create Database:**](#_heading=h.gv9d3zhlnbl5) **12**

[**7. Relational Algebra, SQL Queries**](#_heading=h.aqcy7l34kh6m) **14**

[**8. Querying Using JDBC:**](#_heading=h.r5w1wrlg0r7k) **15**

[**9. Result:**](#_heading=h.1jvt1wzfp94l) **16**

[**10. Conclusion:**](#_heading=h.65fdwkl54fa) **19**

[**11. References:**](#_heading=h.xi5v4hnkdmi) **19**

**TOPIC 17: EMPLOYMENT AGENCY**

**Group member**: Võ Văn Việt - ITDSIU18043

Lê Trần Minh Hoàng - ITDSIU18042

Lê Bảo Phúc - ITDSIU18033

*One of the most important things which graduates care about is to look for a suitable and well-paid job. Have you ever heard about or browsed the web site of employment agencies, such as www.vietnamworks.com? Your topic is to build a system where employers can post their vacancies, employees can browse or search suitable vacancies based on their attributes (age, degree, experience, required salary…)*

*Typical users and their activities in the required system:*

*• Employer: registry, post vacancies, browse or search available resumes*

*• Employee: registry, browse or search suitable vacancies*

*• Staff: manage vacancies (erase outdated vacancies, remind employees with the availability of suitable vacancies…)*

*• Manager: manage staffs*

# **1. Overview - Motivation:**

Having a job that suits a high salary, good compensation and professional environment is a big dream of every student after graduation. Some websites like Linkedin, Vietnamworks,... are where they can post their CVs to seek a suitable job. Employers also can post their interested job to find talented employees. Our topic is to build a system where employers can post their vacancies, employees can browse or search suitable vacancies based on their attributes. Employees can find the job that have expected salary, interested filed, and promote all their strengths. On the other hand, the company can get the employees who meet the requirement such as degree, age, experience.

# **2. Analyzing the requirement:**

**Our database has:**

*Employee*: who want to browse or search suitable job, post their resume to have attention from employers. If there are suitable vacancy, and they like it, they can apply to. Employees provide their basic information such as name, address, phone, email… Employees will be provided an ID by the system.

*Resume:* have been posted from employee, that contains some employee information like expected salary, their background, academic level, year of experience, and their skills. Skills is a multivalued attribute since an employee can have more than one skill. Each resume will be provided an ID by the system.

*Staff:* who post vacancy to recruit new employee for the company, they also looking for resume from employee and if they like it, they can invite them for an interview. Staff can delete the outdate vacancy base on the publish day. Each staff will be provided an ID by the system. Staffs are managed by the manager. The manager will have the manager ID.

*Vacancy*: have been posted from staff and supervise from the company. That contains the job position, field, required academic level, experience, start salary, and upload day. Each vacancy will be provided an ID by the system.

*Company:* contains company name, address, email, rating from client, employee, field.

**Some relationships between entities:**

One employee can post many resumes, and many resumes can be posted by one employee.

Many employees can like many vacancies, and many vacancies can be liked by many employees. If a employee like a job, they can mark the vacancy as apply, interest, like.

One staff can post many vacancies, and many vacancies can be posted by one staff.

Many staffs can like many resumes, and many resumes can be liked by many staffs. If a staff likes a resume, they can mark the resume as invited, consider, ignore the resume.

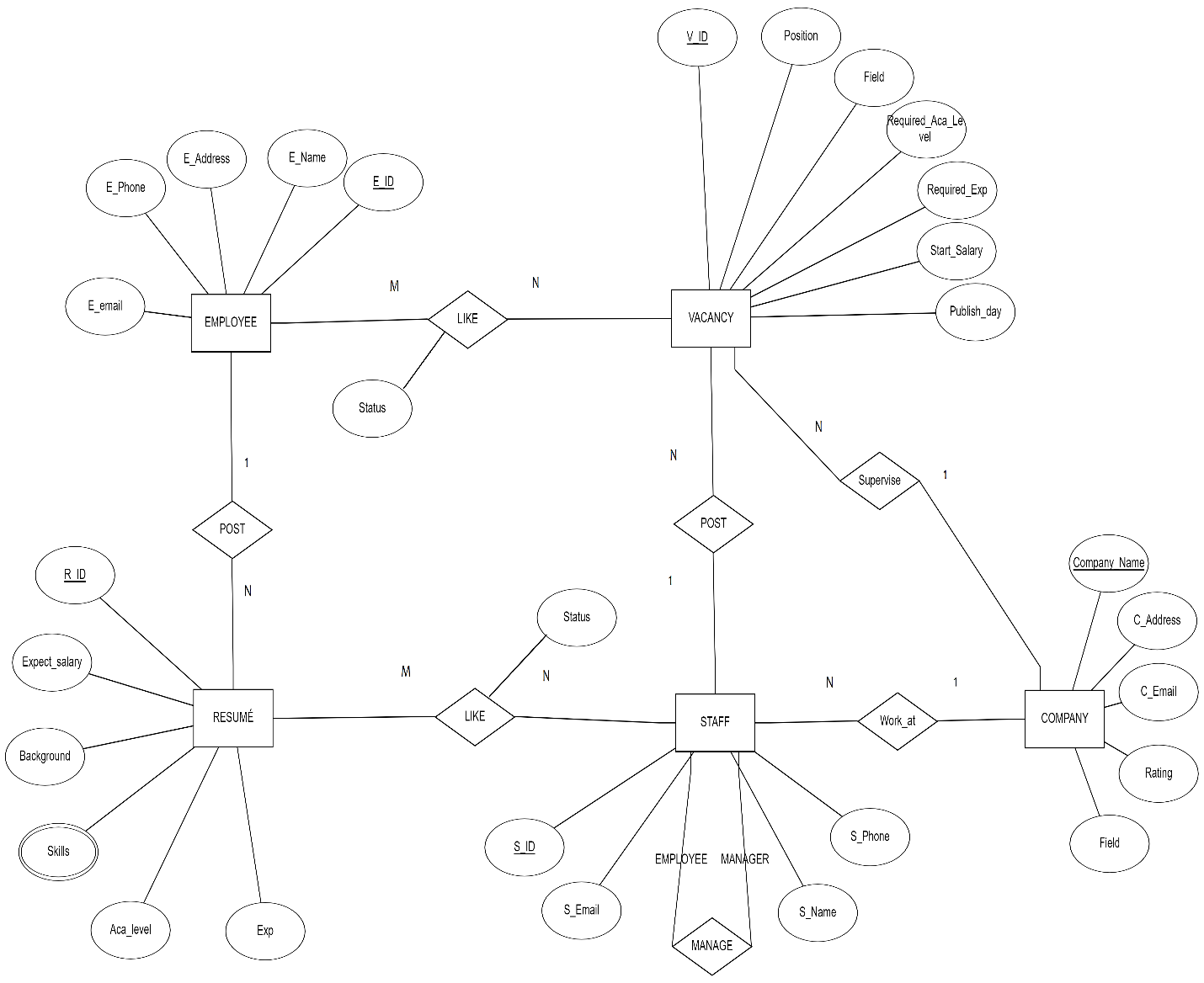
One manager can manage many staffs, and many staffs can be managed by one manager.

Many staffs can work at one company, and one company can have many staffs

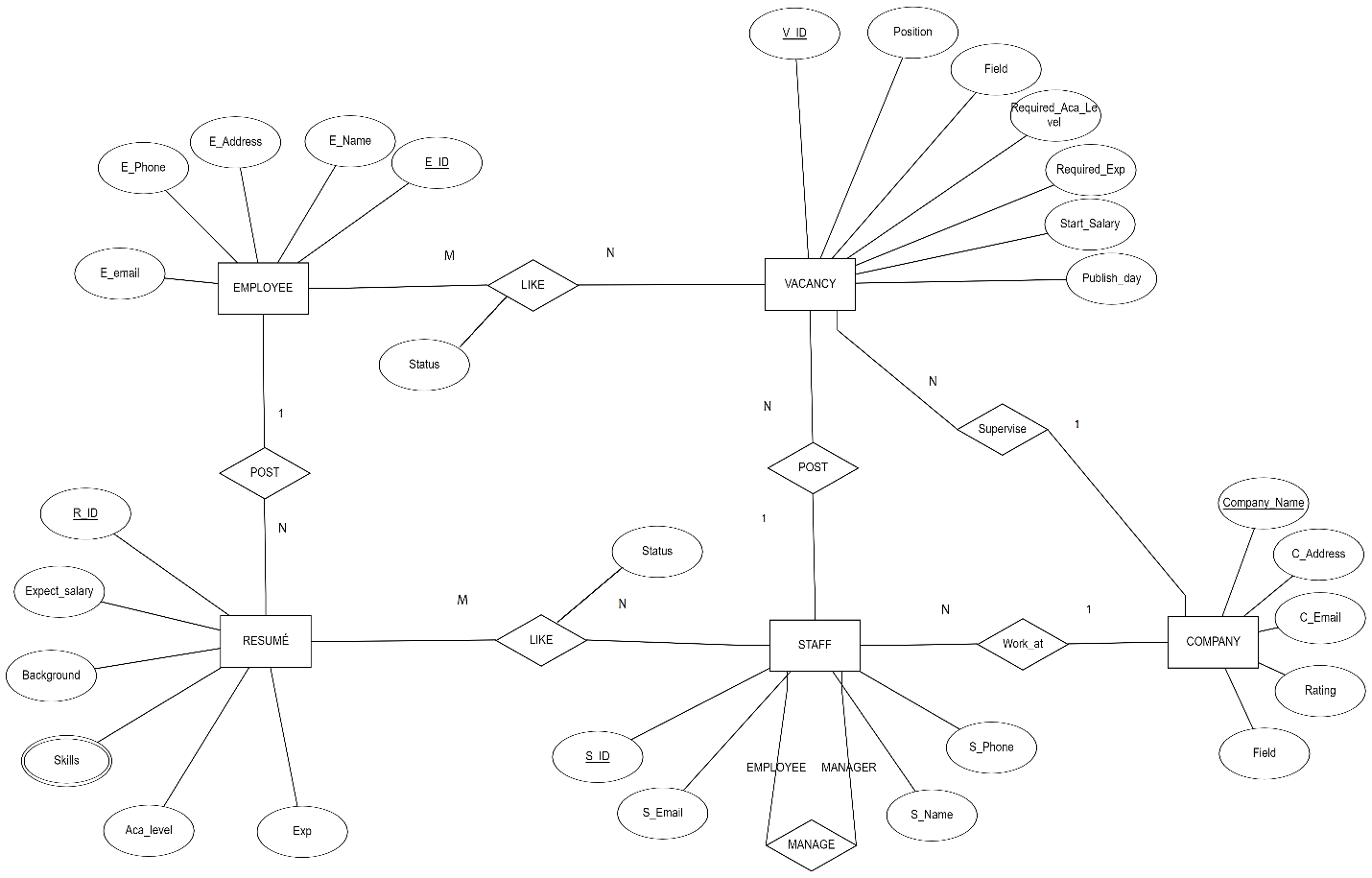
One company can supervise many vacancies, many vacancies can be supervised by one company.

# **3. ER Diagram:**

By discussing and analyzing the requirement, we have completed the ERD below



# **4. Relational Model:**

**Step 1: Regular Entities Type**

**EMPLOYEE**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| E\_ID | E\_Name | E\_Address | E\_Phone | E\_email |

**RESUME**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| R\_ID | Expect\_salary | Background | Skills | Aca\_level | Exp |

**COMPANY**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Company\_Name | C\_Address | C\_Email | Rating | Field |

**STAFF**

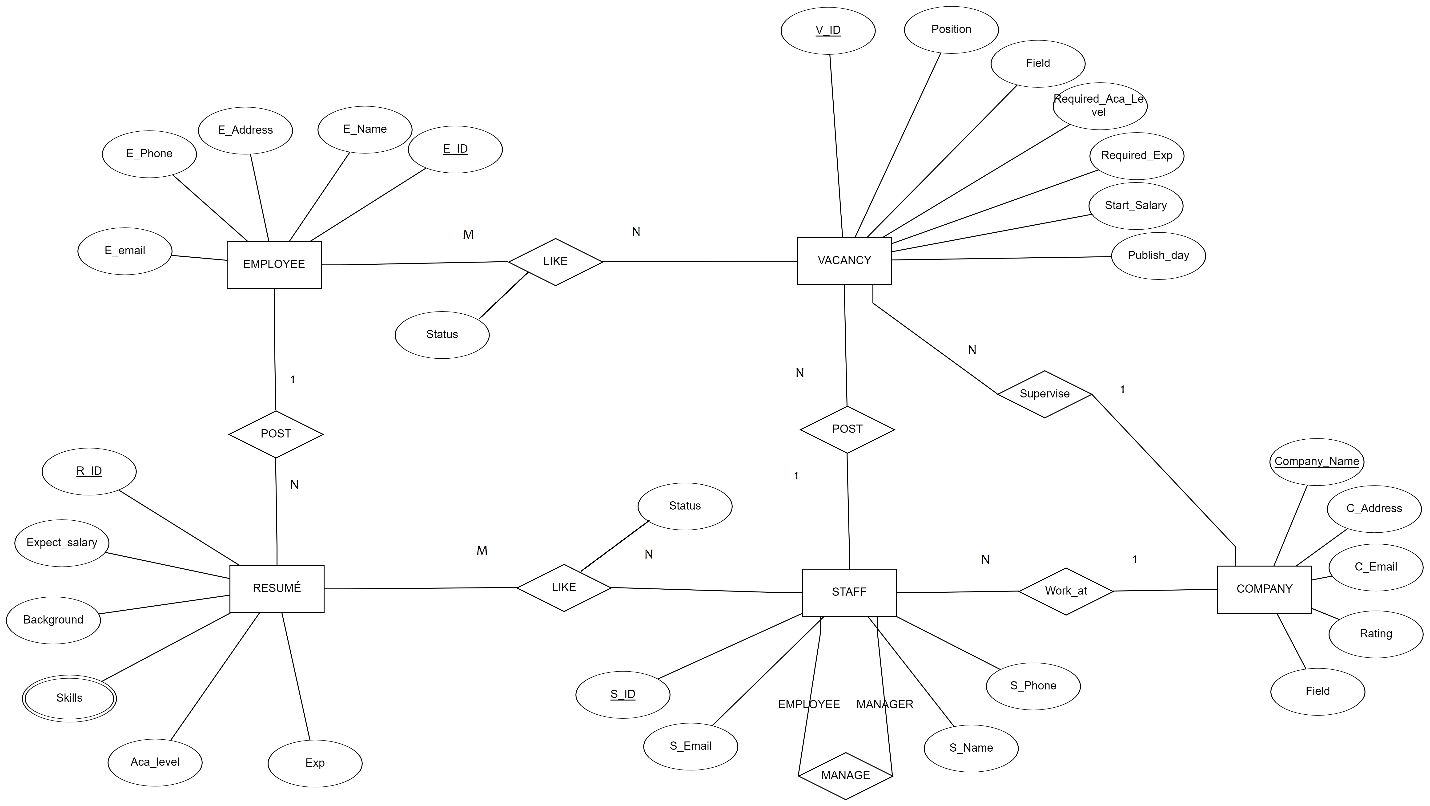
|  |  |  |  |
| --- | --- | --- | --- |
| S\_ID | S\_Email | S\_Name | S\_Phone |

**VACANCY**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| V\_ID | Position | Field | Required\_Aca\_Level | Required\_Exp | Start\_Salary | Publish\_day |

**Step 2: Weak Entity Types**

**Step 3: Mapping Binary 1-to-1**



**Step 4: Binary N-to-1**

**EMPLOYEE**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| E\_ID | E\_Name | E\_Address | E\_Phone | E\_email |



**RESUME**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| R\_ID | Expect\_salary | Background | Skills | Aca\_level | Exp | E\_ID |



**COMPANY**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Company\_Name | C\_Address | C\_Email | Rating | Field |



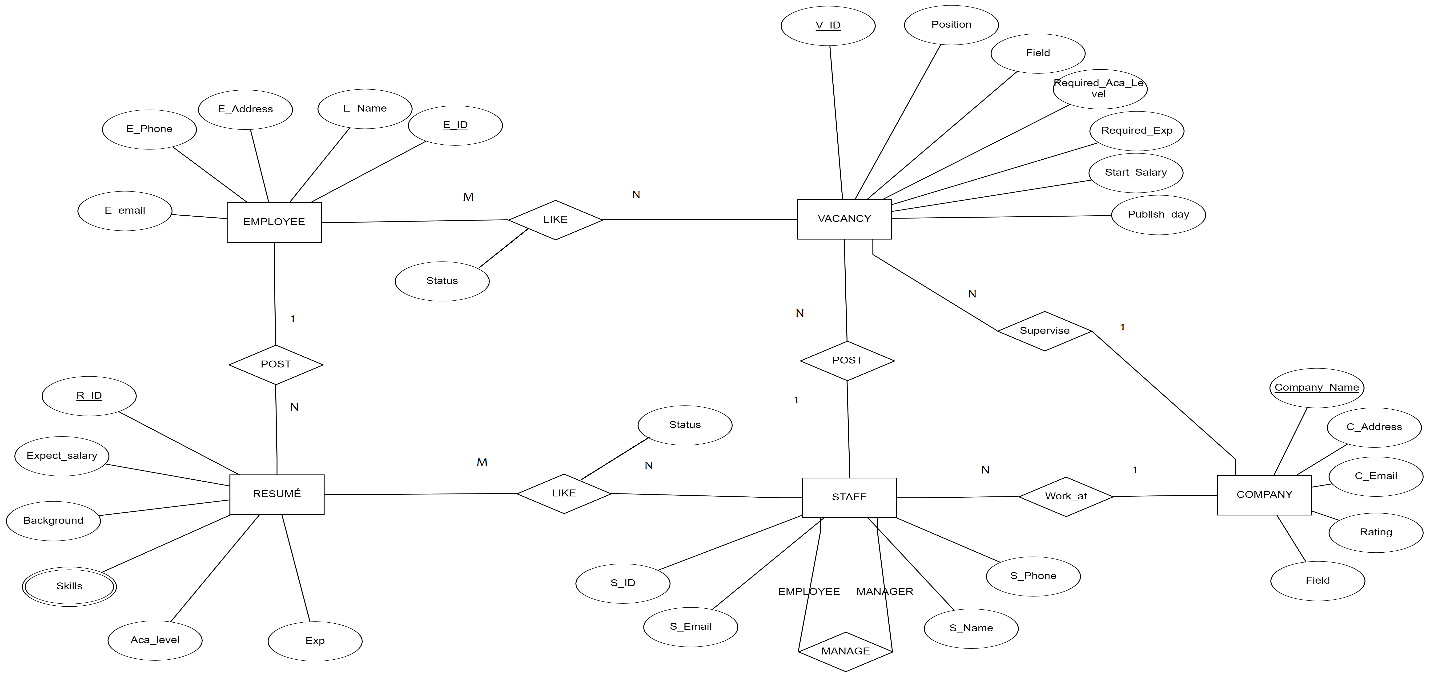
**STAFF**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| S\_ID | S\_Email | S\_Name | S\_Phone | Company\_Name | M\_ID |



**VACANCY**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| V\_ID | Position | Field | Required\_Aca\_Level | Required\_Exp | Start\_Salary | Publish\_day | S\_ID | Company\_Name |

**Step 5: Binary M-to-N**

**EMPLOYEE**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| E\_ID | E\_Name | E\_Address | E\_Phone | E\_email |



**RESUME**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| R\_ID | Expect\_salary | Background | Skills | Aca\_level | Exp | E\_ID |



**R\_Like**

|  |  |  |
| --- | --- | --- |
| R\_ID | S\_ID | Status |



**COMPANY**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Company\_Name | C\_Address | C\_Email | Rating | Field |



**STAFF**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| S\_ID | S\_Email | S\_Name | S\_Phone | Company\_Name | M\_ID |



**VACANCY**

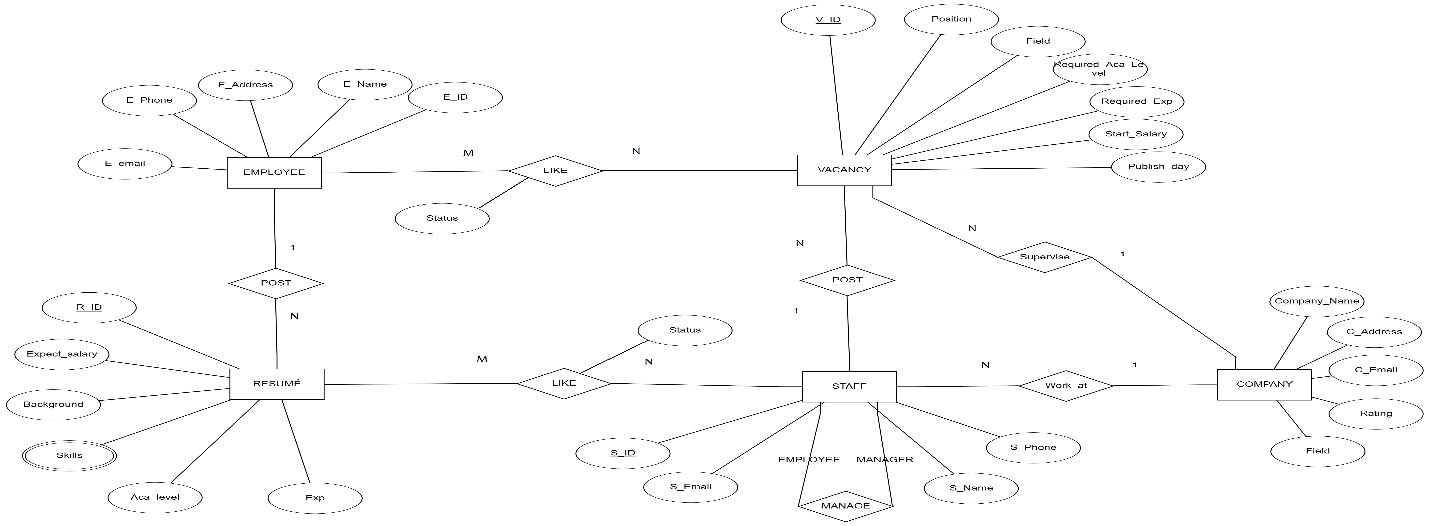
|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| V\_ID | Position | Field | Required\_Aca\_Level | Required\_Exp | Start\_Salary | Publish\_day | S\_ID | Company\_Name |



**V\_Like**

|  |  |  |
| --- | --- | --- |
| E\_ID | V\_ID | Status |



**Step 6: Multivalued Attribute**

**EMPLOYEE**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| E\_ID | E\_Name | E\_Address | E\_Phone | E\_email |



**RESUME**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| R\_ID | Expect\_salary | Background | Aca\_level | Exp | E\_ID |



**Skills**

|  |  |
| --- | --- |
| R\_ID | Skill |



**R\_Like**

|  |  |  |
| --- | --- | --- |
| R\_ID | S\_ID | Status |



**COMPANY**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Company\_Name | C\_Address | C\_Email | Rating | Field |



**STAFF**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| S\_ID | S\_Email | S\_Name | S\_Phone | Company\_Name | M\_ID |



**VACANCY**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| V\_ID | Position | Field | Required\_Aca\_Level | Required\_Exp | Start\_Salary | Publish\_day | S\_ID | Company\_Name |

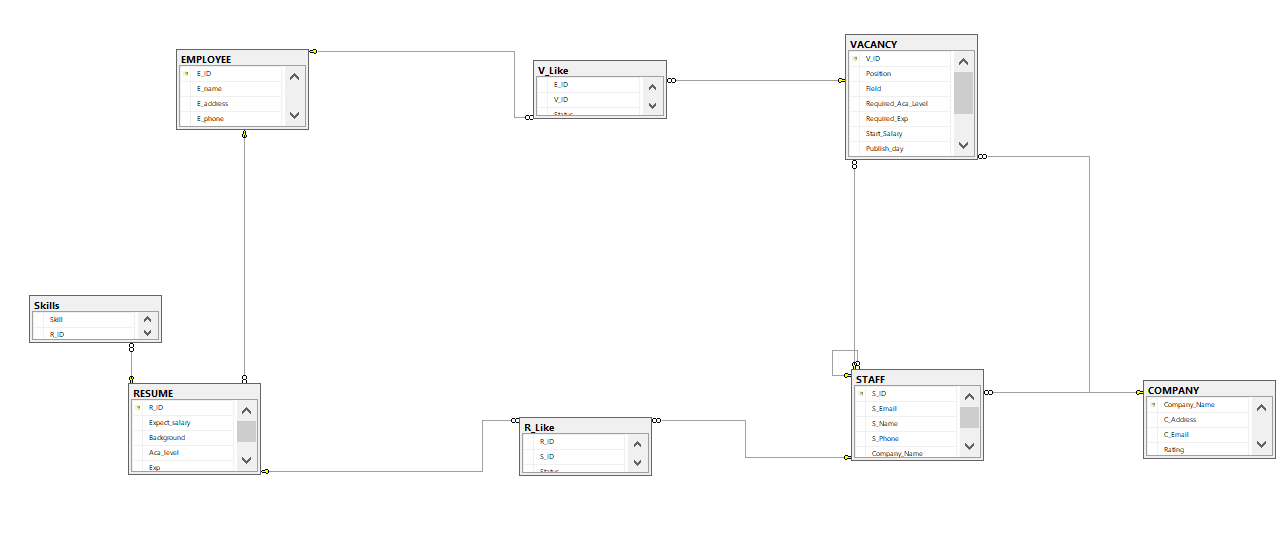


V\_Like

|  |  |  |
| --- | --- | --- |
| E\_ID | V\_ID | Status |



# **5. Conceptual Schema:**

**Database Diagram:** 

**Employee** = (E\_ID, E\_Name, E\_Address, E\_Phone, E\_email)

**Resume** = (R\_ID, Expected\_Salary, Background, Aca\_level, Exp, *E\_ID*)

* From Resume.E\_ID to Employee.E\_ID

**Skills** = *(R\_ID*, Skill)

* From Skills.R\_ID to Resume.R\_ID

**R\_Like** = (*R\_ID, S\_ID*, Status)

* From R\_Like.R\_ID to Resume.R\_ID
* From R\_Like.S\_ID to S\_ID

**Company** = (Company\_Name, C\_Address, C\_Email, Rating, Field)

**Staff** = (S\_ID, S\_Email, S\_Name, S\_Phone*, Company\_Name, M\_ID*)

* From Staff.Company\_Name to Company.Company\_Name
* From Staff.M\_ID to Staff.S\_ID

**Vacancy** = (V\_ID, Position, Field, Required\_Ace\_Level, Required\_Exp, Start\_Salary, Publish\_day, *S\_ID, Company\_Name*)

* From Vacancy.S\_ID to Staff.S\_ID
* From Vacancy.Company\_Name to Company.Company\_Name

**V\_Like** = (*E\_ID, V\_ID*, Status)

* From V\_Like.E\_ID to Employee.E\_ID

- From V\_ID to Vacancy.V\_ID

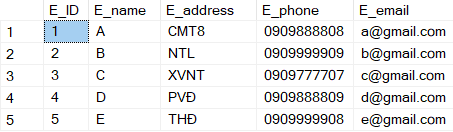
# **6. Create Database:**

The .bak and .sql files in zip file

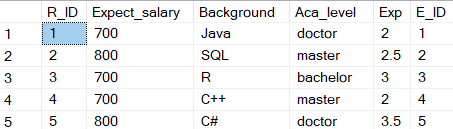
Database Name: EmploymentAgency\_17

Sample Data from Tables:

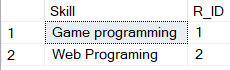
Employee:



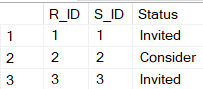
Resume:



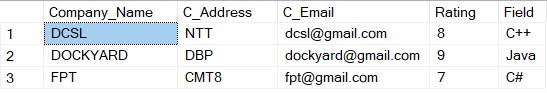
Skills:



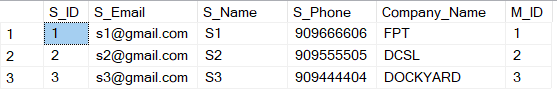
R\_Like:



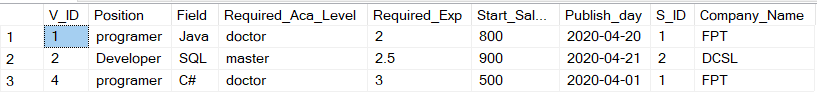
Company:



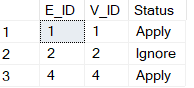
Staff:



Vacancy:



V\_Like:



# **7. Relational Algebra, SQL Queries**

***Select the employee and the company that have the same street address***

*Query*

--Select the employee and the company that have the same street address

SELECT em.E\_name,em.E\_address, co.Company\_Name

FROM dbo.EMPLOYEE AS em, dbo.COMPANY AS co

WHERE em.E\_address = co.C\_Address

*Relational* *Algebra*

π E\_name,E\_address,Company\_Name (σ E\_address = C\_Address (EMPLOYEE X COMPANY))

***Select the suitable job***

*Query*

--Select the suitable job

SELECT va.Company\_Name, va.Position, va.Field, va.Required\_Aca\_Level, va.Required\_Exp, va.Start\_Salary

FROM dbo.RESUME AS re, dbo.VACANCY AS va

WHERE re.Expect\_salary<=va.Start\_Salary AND re.Background = va.Field AND re.Aca\_level=va.Required\_Aca\_Level AND re.Exp>=va.Required\_Exp

*Relational Algebra*

π Company\_Name,Position,Field,Required\_Aca\_Level,Required\_Exp (σ Expect\_salary<=Start\_Salary ∧ Background = Field ∧ Aca\_level=Required\_Aca\_Level ∧ Exp>=Required\_Exp (RESUME X VACANCY))

***Select the jobs that have salary larger than 800***

*Query*

--Select the jobs that have salary larger than 800

SELECT va.Position,va.Start\_Salary, va.Company\_Name

FROM dbo.VACANCY AS va

WHERE va.Start\_Salary > 800

*Relational* *Algebra*

π Position,Start\_Salary,Company\_Name (σ Start\_Salary > 800 (VACANCY))

***Select the employees who have background in SQL***

*Query*

--Select the employees who have background in SQL

SELECT re.R\_ID, em.E\_Name

FROM dbo.RESUME AS re, dbo.EMPLOYEE AS em

WHERE re.Background = 'SQL' AND re.E\_ID = em.E\_ID

*Relational* *Algebra*

π R\_ID,E\_Name (σ Background = 'SQL' (RESUME ⨝ EMPLOYEE))

***Select the employees who have bachelor degree and work experience > 2.5 years***

*Query*

--Select the employees who have bachelor degree and work experience > 2.5 years

SELECT re.R\_ID, em.E\_Name

FROM dbo.RESUME AS re, dbo.EMPLOYEE AS em

WHERE re.E\_ID = em.E\_ID AND re.Aca\_level = 'bachelor' AND re.Exp > 2.5

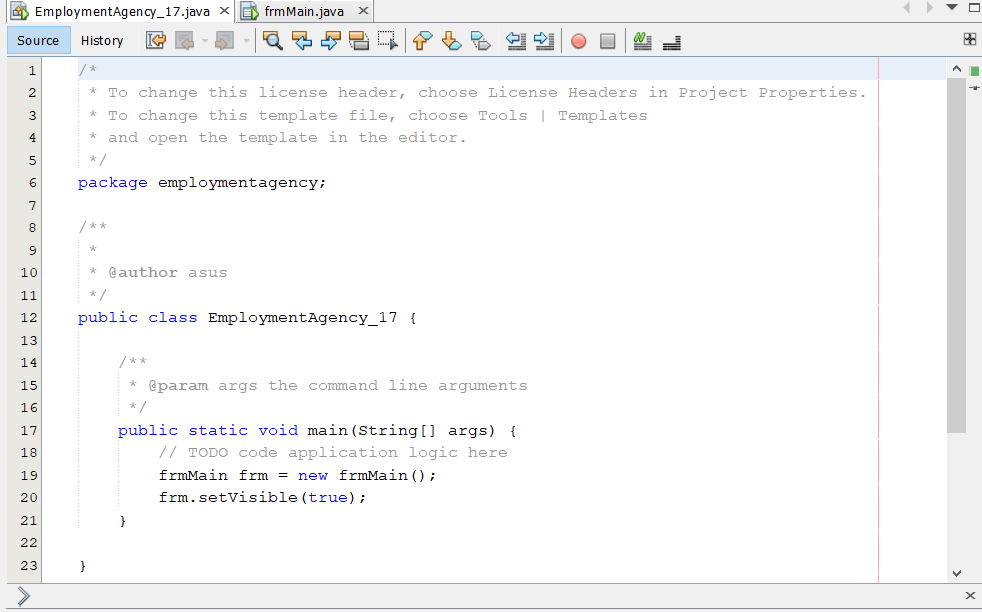
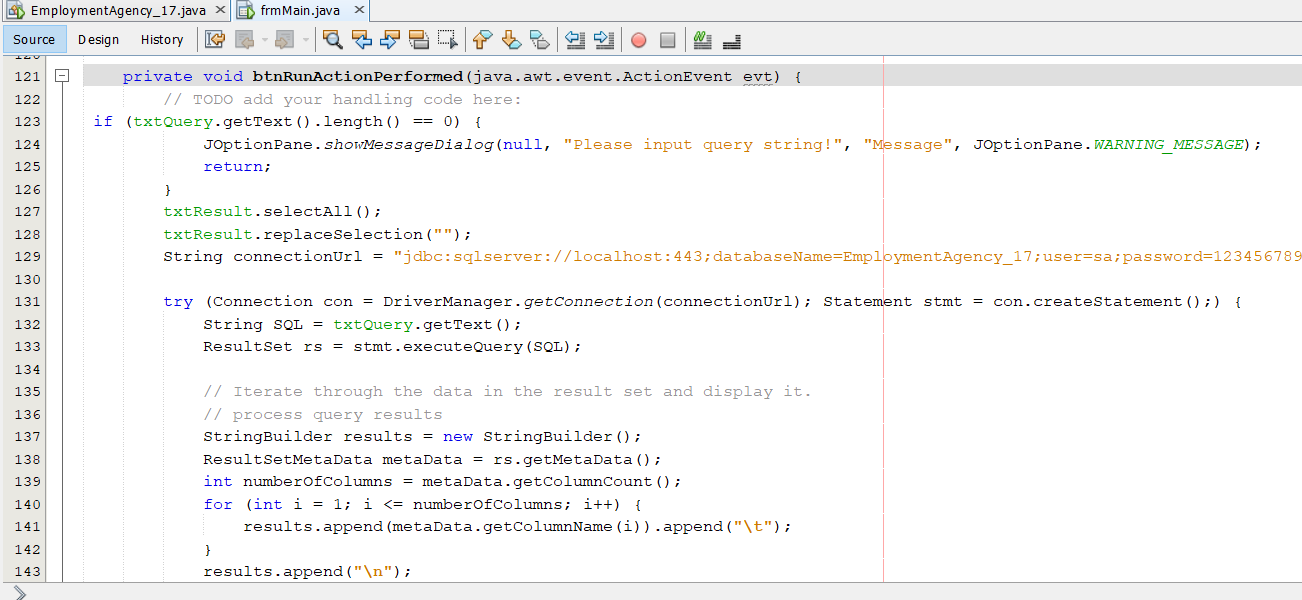
*Relational* *Algebra*

π R\_ID,E\_Name (σ Aca\_level = 'bachelor' ∧ Exp > 2.5 (RESUME ⨝ EMPLOYEE))

# **8. Querying Using JDBC:**

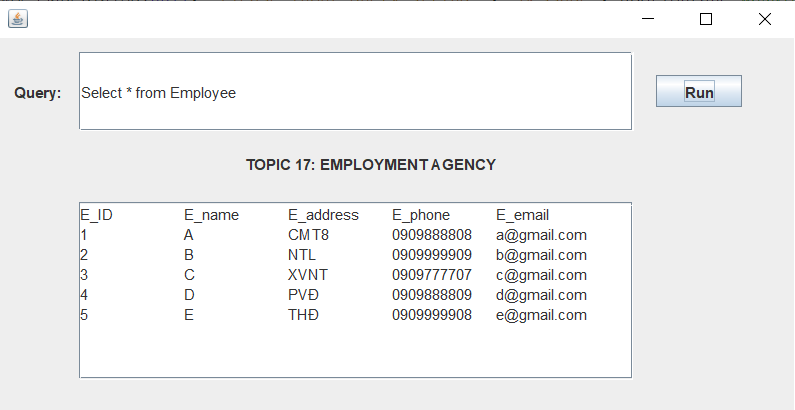
Create the Java Project to connect open database connection from Java to SQL Server

The application has classes below:

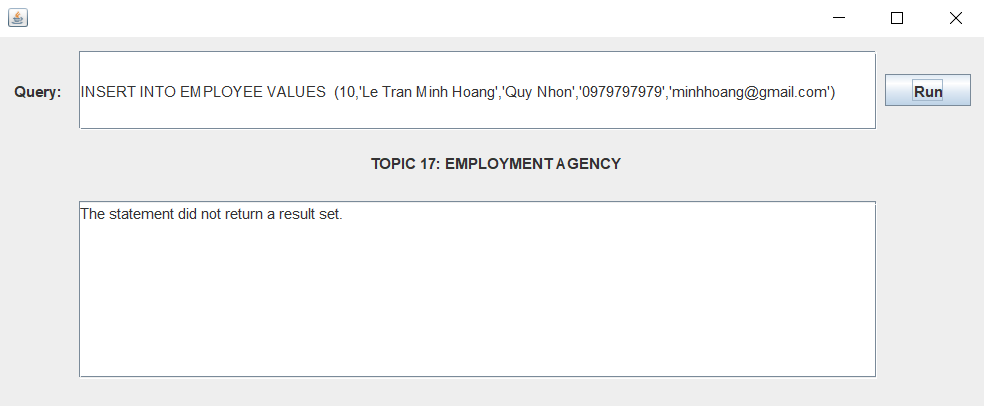
# **9. Result:**

The java application that we can input the query to the box, click run, and the system will show the query result.

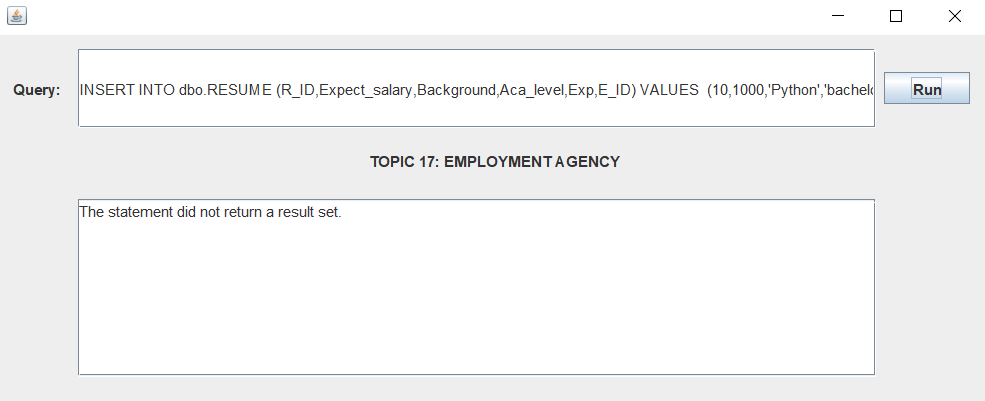


Let’s do more:

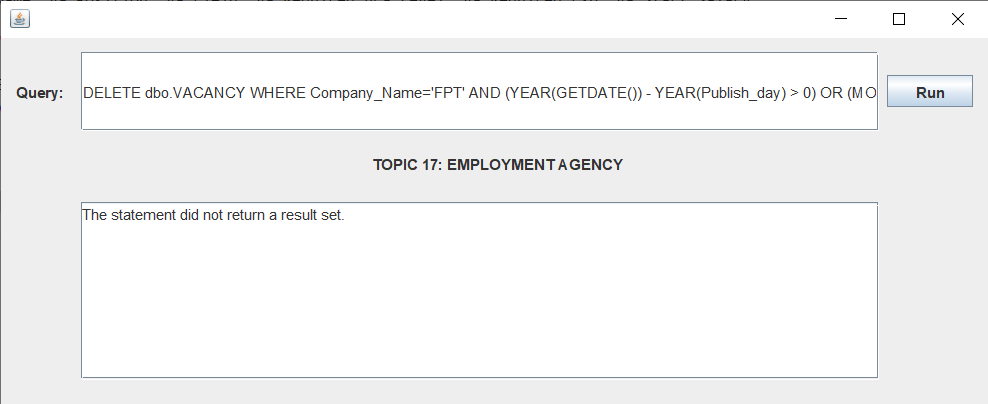
For example, a fresher who is seeking for job. Firstly, he should provide some basic information in the system:



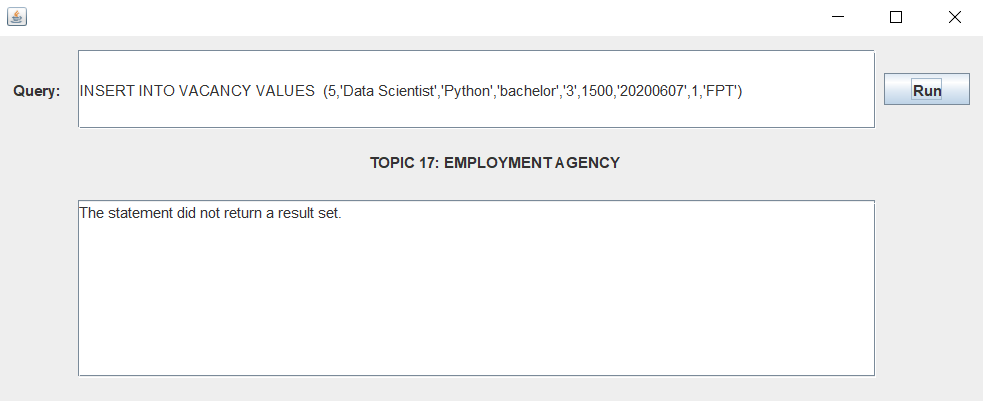
Then, we are going to post the resumé:



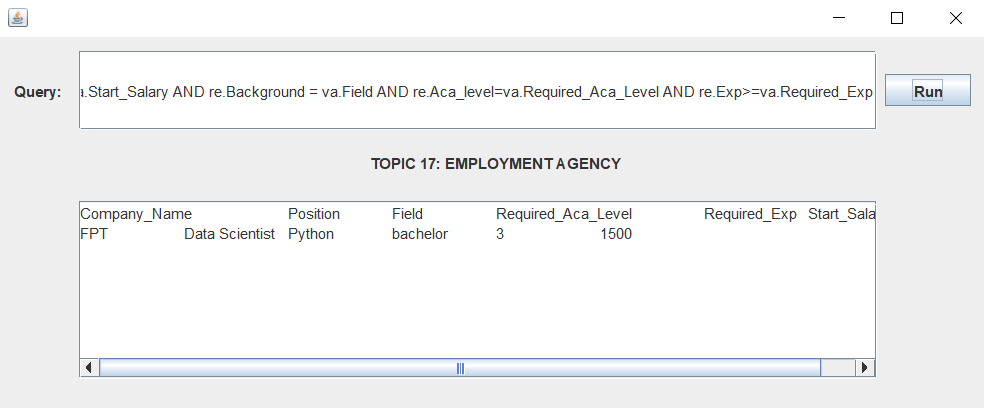
Besides, the staff from the company FPT is going to delete the outdate vacancy:



The, they post the vacancy for the recruitment:



The fresher can search for the suitable job with expected salary and the field that he interested in:



# **10. Conclusion:**

In this project, we have built a system of Employment Agency where employers can post their vacancies, employees can browse or search suitable vacancies based on their attributes. The system is created by using SQL Server database, JDBC, and the knowledge that we archived from the Principle of Database Management course.

# **11. References:**

*The Course: Principles of Database Management, Lab 6-7*

[*https://www.vietnamworks.com/*](https://www.vietnamworks.com/)

[*https://www.linkedin.com/*](https://www.linkedin.com/)