# Lab2

**Nguyễn Phan Hoài Nam**

**MSSV: 2031200016**

**Link Github: https://github.com/kaito7love/CSE301.git**

**Problem1:**

**1: Finding Entities, Key Attributes, and Related Attributes**

1. **CUSTOMERS**
   - **Key Attribute:** CUSTOMERID
   - **Attributes:** FULLNAME, ADDRESS, PHONE, BIRTHDAY, SALE, REGISTRATIONDATE
2. **STAFF**
   - **Key Attribute:** STAFFID
   - **Attributes:** FULLNAME, DAYOFENTRY, PHONE
3. **PRODUCTS**
   - **Key Attribute:** PRODUCTID
   - **Attributes:** PRODUCTNAME, UNIT, NATION, PRICE
4. **INVOICE**
   - **Key Attribute:** INVOICEID
   - **Attributes:** PURCHASEDATE, CUSTOMERID, STAFFID, VALUE
5. **DETAILOFINVOICE**
   - **Key Attribute:** INVOICEID, PRODUCTID
   - **Attributes:** QUANTITY

**2: Finding Relationships**

1. **Customer-Invoice Relationship:**
   - A customer can have multiple invoices.
   - Each invoice is associated with one customer.
   - Relationship: One-to-Many
2. **Staff-Invoice Relationship:**
   - A staff member can handle multiple invoices.
   - Each invoice is handled by one staff member.
   - Relationship: One-to-Many
3. **Invoice-Detail of Invoice Relationship:**
   - Each invoice can contain multiple product details.
   - Each product detail is associated with one invoice.
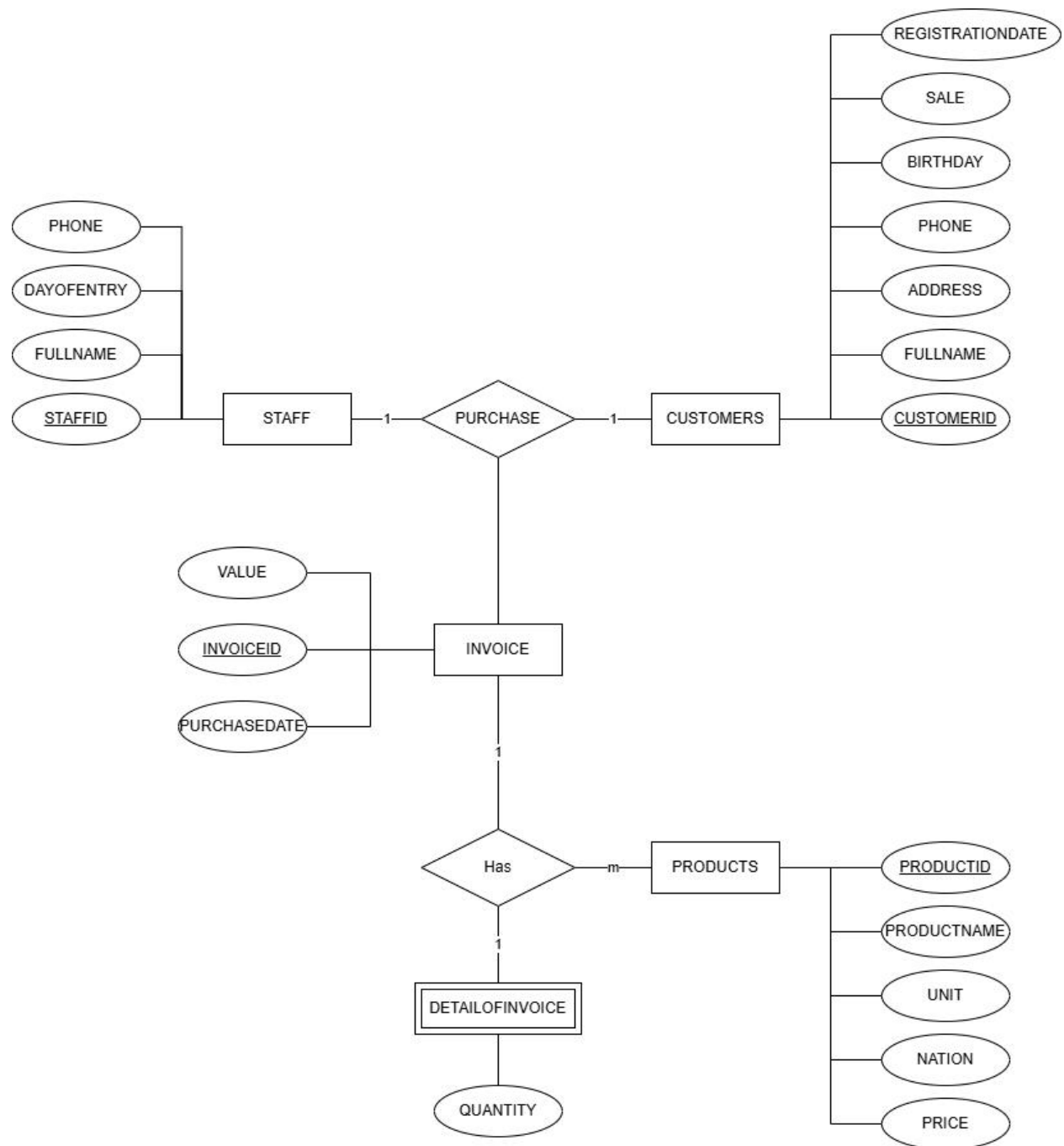   - Relationship: One-to-Many
4. **Product-Detail of Invoice Relationship:**

- A product can appear in multiple invoice details.
- Each invoice detail is associated with one product.
- Relationship: One-to-Many

## 3: Finding Weak Entities and Weak Relationships

DETAILOFINVOICE is week entity.

## 4: Draw an ER Model

## Problem2:

An industry company requires ==employee management== by computer. The company has many employees. Each ==employee== is recorded with <span style="color:red">Full Name: Middle Name and First Name, has a unique code, has date of birth, address, gender,</span> will be in ==charge== directly by a ==manager (also an employee)== and belong to a single department.

==Salaries== for employees are determined based on their working positions: ==full-time employees== receive a ==salary== according to ==company policies==, while ==part-time employees== are ==paid== based on their ==working hours==, according to rules pay rates.

In the company, there are many ==departments== to manage employees. Each ==department== <span style="color:red">has a unique code, with a room name, that oversees a certain employee, and records</span> the whole <span style="color:red">day that the manager starts in charge of the department.</span> Besides, each department can have one or more locations. The company will implement different ==projects==. Therefore, ==each project== records <span style="color:red">the code and name of the project, the location of the project implementation.</span>

The department will be assigned and perform the different projects. Each employee can participate in many different projects, and each project can be participated in by many employees.

When an ==employee joins a project==, that <span style="color:red">employee code, that scheme code, as well as the employee's working time</span> for the project in a week will be recorded.

To pay more attention to the employees, the company will record some information about the ==relatives of the employees==. <span style="color:red">The names, dates of birth and relationships of relatives will be recorded</span>

## 1: Finding Entities, Key Attributes, and Related Attributes

1. **EMPLOYEE**
   - **Key Attribute:** EMPLOYEEID (unique code)
   - **Attributes:** FULLNAME (Middle Name and First Name), DATEOFBIRTH, ADDRESS, GENDER
2. **MANAGER**
   - **Key Attribute:** MANAGERID (unique code)
   - **Attributes:** MANAGERSTARTDATE
3. **DEPARTMENT**
   - **Key Attribute:** DEPARTMENTCODE (unique code)
   - **Attributes:** DEPARTMENTNAME, LOCATIONS
4. **PROJECT**
   - **Key Attribute:** PROJECTID (unique code)
   - **Attributes:** PROJECTNAME, PROJECTLOCATION
5. **EMPLOYEE_PROJECT**
   - **Key Attribute:** (EMPLOYEEID, PROJECTID)

- o **Attributes:** HOURS_PER_WEEK
6. **RELATIVE**
   - o **Key Attribute:** EMPLOYEEID (unique code)
   - o **Attributes:** RELATIVENAME, RELATIVEDATEOFBIRTH, RELATIONSHIP

## 2: Finding Relationships

1. **Employee-Department Relationship:**
   - o An employee belongs to one department.
   - o A department oversees multiple employees.
   - o Relationship: One-to-Many
2. **Employee-Manager Relationship:**
   - o An employee is managed by one manager.
   - o A manager (employee) can manage multiple employees.
   - o Relationship: One-to-Many
3. **Department-Project Relationship:**
   - o A department can be assigned to multiple projects.
   - o A project can be assigned to one department.
   - o Relationship: Many-to-One
4. **Employee-Project Relationship:**
   - o An employee can participate in multiple projects.
   - o A project can have multiple employees.
   - o Relationship: Many-to-Many
5. **Employee-Relative Relationship:**
   - o An employee can have multiple relatives.
   - o Each relative is associated with one employee.
   - o Relationship: One-to-Many

## 3: Finding Weak Entities and Weak Relationships

**EMPLOYEE_PROJECT** can be considered a weak entity as it depends on both EMPLOYEE and PROJECT for its existence.

# 4: Draw an ER Model