Project Questions

1) Can you think 5 more rules (other than the one explicitly described above) that are likely to be used in a company.

a) Employee Performance Reviews: Each employee might have periodic performance reviews that are conducted by their supervisor. These reviews could include ratings in various categories and overall feedback.

b) Product Inventory: There might be a need to track the inventory levels of each product. This could involve rules for when to reorder products, minimum inventory levels, etc.

c) Customer Orders: Customers might place orders for products. These orders could have rules around shipping, payment, order fulfillment, etc.

d) Employee Benefits: The company might offer various benefits to employees, such as health insurance, retirement plans, etc. There could be rules around eligibility for these benefits.

e) Customer purchase-info: customer should have a subclass like employee has (salary). Which can used to keep track of the customer's purchase history (purchase id, purchase_type, payment_info, etc)

2) Is the ability to model super-class/subclass relationships likely to be important in such environment? Why or why not?

The ability to model super-class/subclass relationships is indeed important in such an environment.
This is because it allows for the representation of 'is-a' relationships, which are common in real-world scenarios.
For example, in the context of this project, an Employee, a Customer, and a Potential Employee are all types of People.
By modeling this as a super-class (People) / subclass (Employee, Customer, Potential Employee) relationship, we can avoid redundancy and maintain consistency in the database.
It also allows for easier querying and data manipulation.

3) Justify using a Relational DBMS like Oracle for this project.

Using a Relational DBMS like Oracle for this project would be beneficial for several reasons:

a) Structured Data: Oracle is excellent at handling structured data, which is what we have in this scenario. The data can be easily organized into tables, making it easy to manage and query.

b) ACID Properties: Oracle ensures Atomicity, Consistency, Isolation, and Durability (ACID) of transactions, which is crucial for maintaining data integrity in a business environment.

c) Scalability: Oracle can handle large volumes of data and many concurrent users, making it suitable for a company setting.

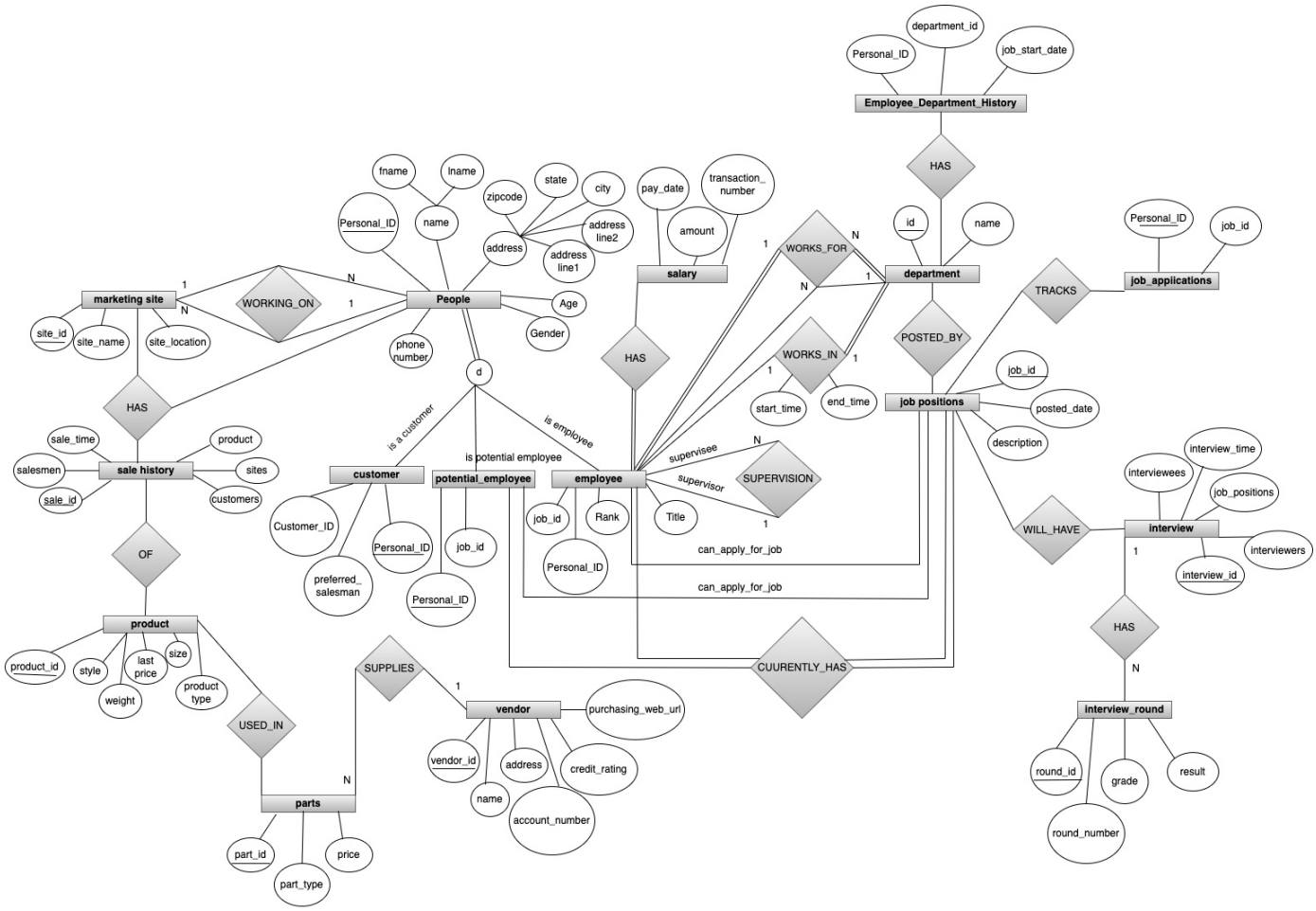
d) Security: Oracle provides robust security features, including access control, data encryption, and auditing capabilities.

e) Advanced Features: Oracle supports advanced features such as stored procedures, triggers, and views, which can be useful for implementing complex business logic.

Project Exercises

I. Draw an EER to accurately represent this set of requirements. This will be your Conceptual Design. Clearly specify any assumption that you are making. You can use any tools (software) to draw the EER. You don't need describe the value constraints of the attributions in the EER diagram. (25%)

EER



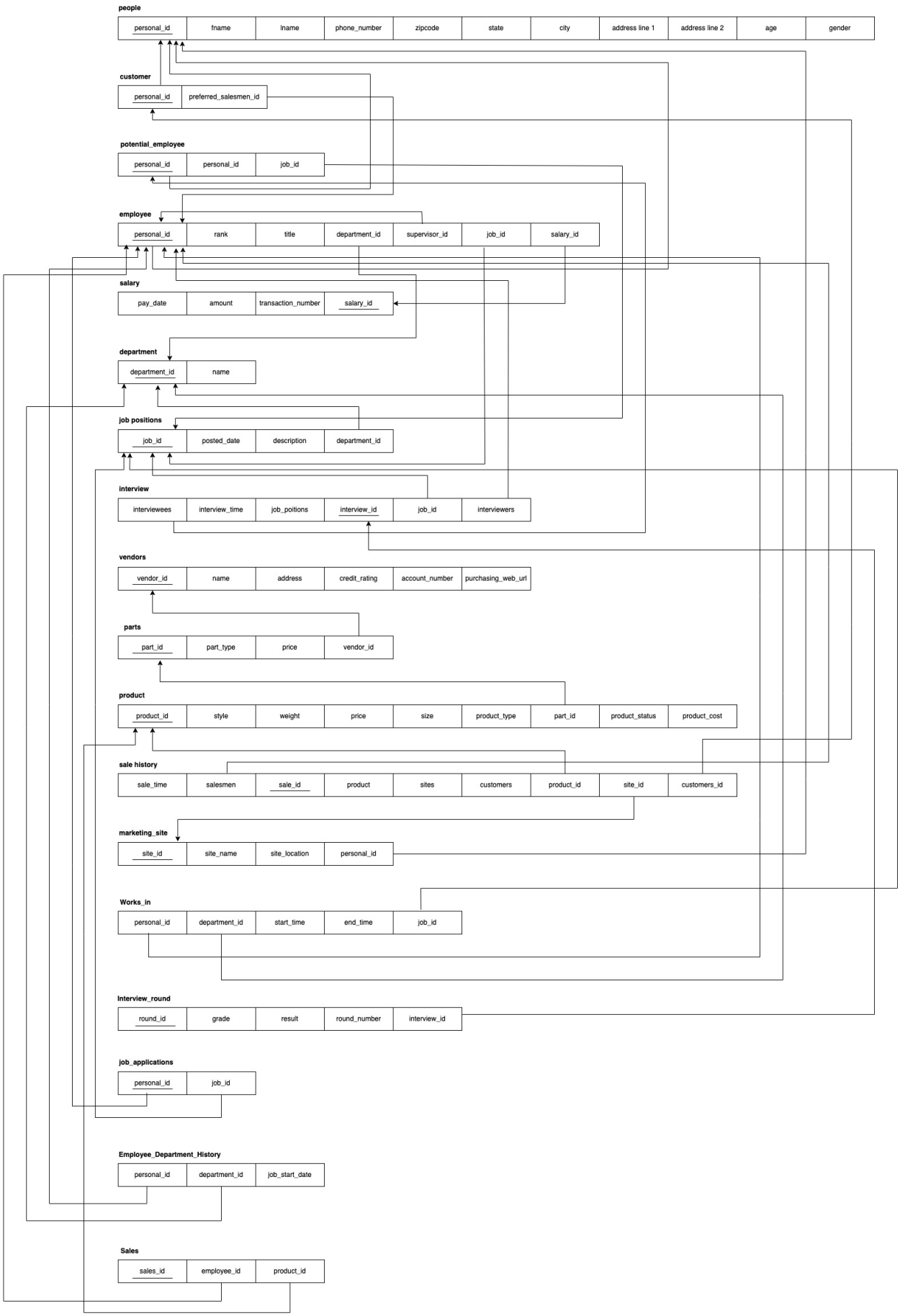
Assumption

we can have a junctions table name "sales" to connect two tables - "employee" with "products"

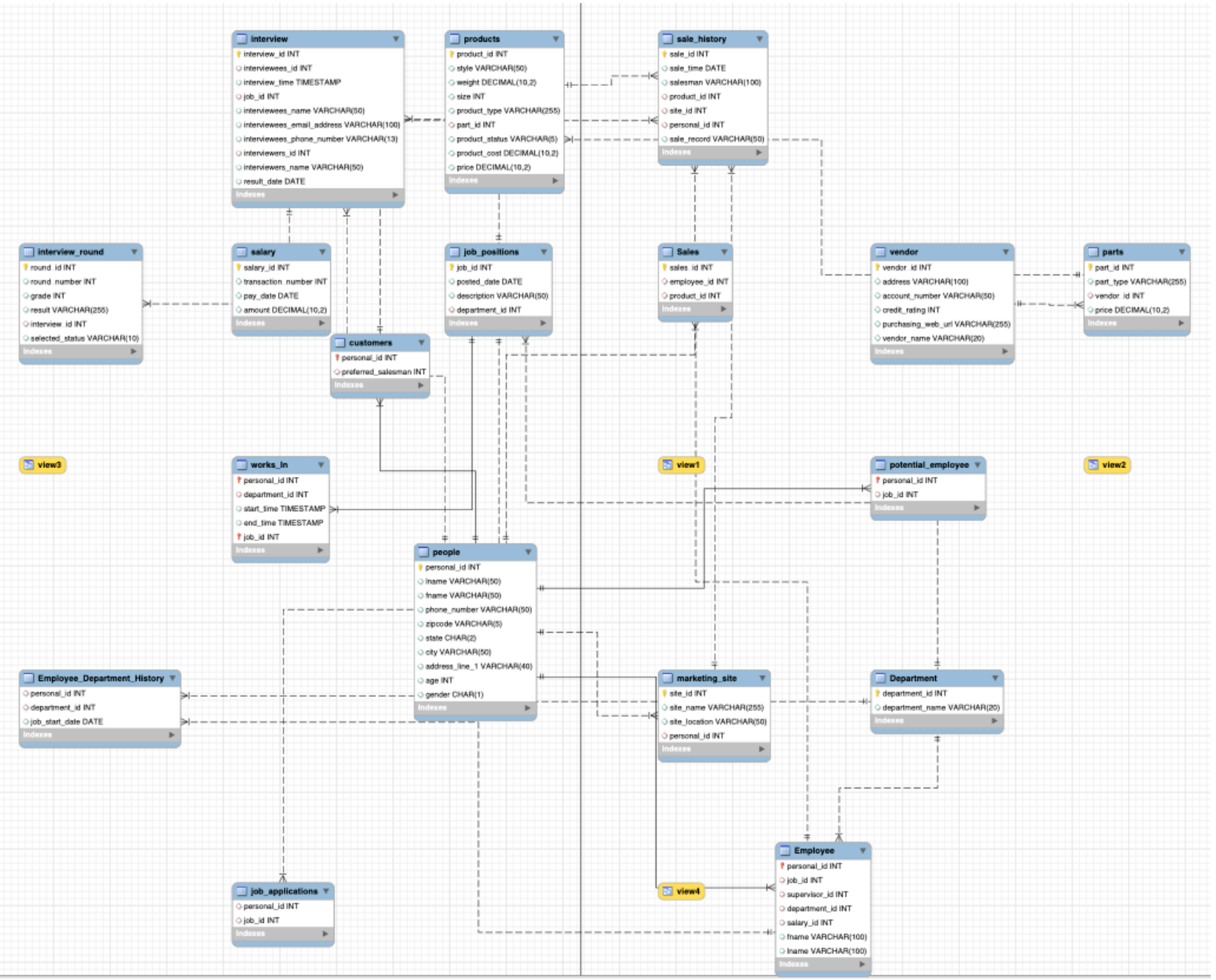
II. Use a relational DBMS to implement the database. Perform the following steps. (20%)

a) Convert your Conceptual model to a Logical model that can be implemented in a relational DBMS like Oracle. During this process you replace M-N relationships and multi-valued attributes with constructs that can be implemented in the relational DBMS. Draw EER for the logical model after your modifications. Feel free to change your conceptual model (first delivery) if needed.

relational schema



b) Convert the EER to a database design. Document your design in Database Schema format like the one we discussed in the class.



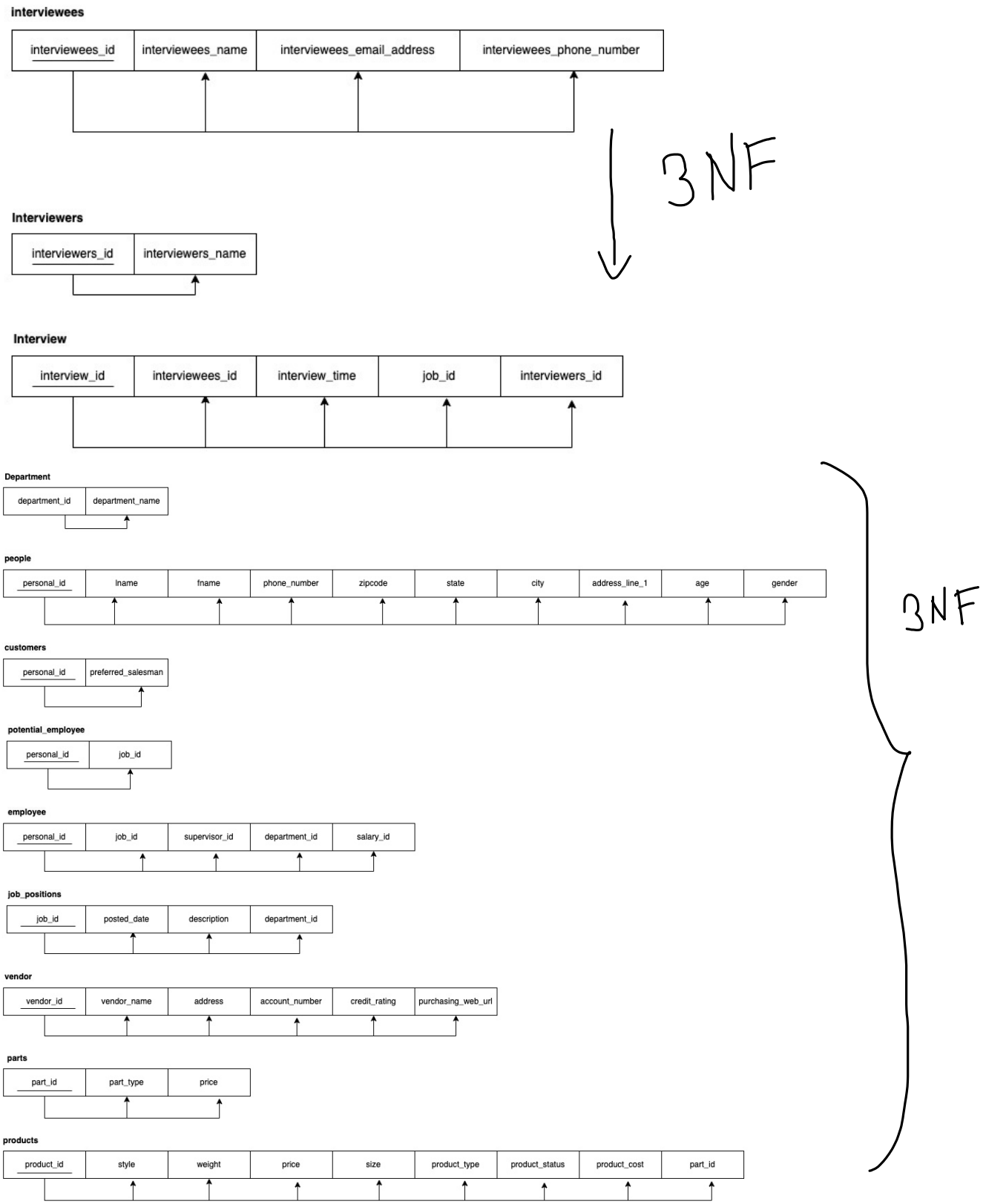
III. Use appropriate naming conventions for all your tables and attributes. (40%)

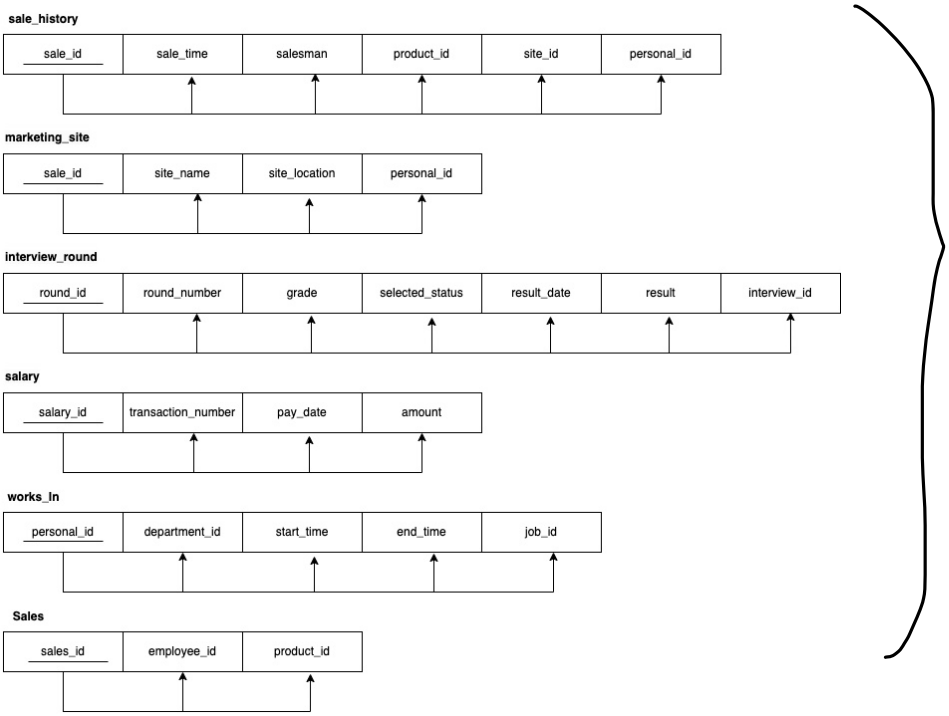
a) Normalize all your tables to third normal form. Make any necessary changes to the EER. Explain why these changes needed to be made.

One of my entity "Interview" was not in 3NF (Third Normal Form) because it had transitive dependencies. In this case, the attributes interviewees_name, interviewees_email_address, interviewees_phone_number, and interviewers_name were dependent on interviewees_id and interviewers_id, which are non-prime attributes (not part of the primary key).

To normalize this table into 3NF, I removed these transitive dependencies. I did this by creating separate tables for Interviewees and Interviewers where their id is the primary key and their other attributes are dependent on this primary key.

b) Draw a dependency diagram for each table.





c) Write SQL statements to create database, tables, and all other structures.
Primary keys and foreign keys must be defined appropriately.
The quantity constraints of the relation between the entities, which should be described in EER diagram, are not required.

```
CREATE TABLE people (
  personal_id          INT PRIMARY KEY,
  lname                VARCHAR(50),
  fname                VARCHAR(50),
  phone_number         VARCHAR(50),
  zipcode              VARCHAR(5),
  state                CHAR(2),
  city                 VARCHAR(50),
  address_line_1       VARCHAR(40),
  age                  INT,
  gender               CHAR(1)
);

CREATE TABLE Department (
  department_id                INT PRIMARY KEY,
  department_name              VARCHAR(20)
);

CREATE TABLE customers (
  personal_id INT PRIMARY KEY,
  preferred_salesman INT,
  FOREIGN KEY (personal_id) REFERENCES people(personal_id),
  FOREIGN KEY (preferred_salesman) REFERENCES employee(Personal_id)
);

CREATE TABLE potential_employee (
  personal_id INT PRIMARY KEY,
  job_id INT,
  FOREIGN KEY (personal_id) REFERENCES people(personal_id),
  FOREIGN KEY (job_id) REFERENCES Job_positions(job_id)
);
```

```
CREATE TABLE employee (
  personal_id INT PRIMARY KEY,
  job_id INT,
  supervisor_id INT,
  department_id INT,
  salary_id INT,
  FOREIGN KEY (personal_id) REFERENCES people(personal_id),
  FOREIGN KEY (job_id) REFERENCES Job_positions(job_id),
  FOREIGN KEY (supervisor_id) REFERENCES employee(personal_id),
  FOREIGN KEY (department_id) REFERENCES Department(department_id),
  FOREIGN KEY (salary_id) REFERENCES Salary(salary_id)
);

CREATE TABLE job_positions (
  job_id INT PRIMARY KEY,
  posted_date DATE, # YYYY-MM-DD
  description VARCHAR(50),
  department_id INT,
  FOREIGN KEY (department_id) REFERENCES department(department_id)
);

CREATE TABLE Interviewees (
  interviewees_id INT PRIMARY KEY,
  interviewees_name VARCHAR(50),
  interviewees_email_address VARCHAR(100),
  interviewees_phone_number VARCHAR(13),
  FOREIGN KEY (interviewees_id) REFERENCES People(Personal_id)
);

CREATE TABLE Interviewers (
  interviewers_id INT PRIMARY KEY,
  interviewers_name VARCHAR(50),
  FOREIGN KEY (interviewers_id) REFERENCES Employee(Personal_id)
);

CREATE TABLE Interview (
  interview_id INT PRIMARY KEY,
  interviewees_id INT,
  interview_time TIMESTAMP,
  job_id INT,
  interviewers_id INT,
  FOREIGN KEY (interviewees_id) REFERENCES Interviewees(interviewees_id),
  FOREIGN KEY (job_id) REFERENCES job_positions(job_id),
  FOREIGN KEY (interviewers_id) REFERENCES Interviewers(interviewers_id)
);

CREATE TABLE vendor (
  vendor_id INT PRIMARY KEY,
  vendor_name VARCHAR(50),
  address VARCHAR(100),
  account_number VARCHAR(50),
  credit_rating INT,
  purchasing_web_url VARCHAR(255)
);

CREATE TABLE parts (
  part_id INT PRIMARY KEY,
  part_type VARCHAR(255),
  price DECIMAL(10,2),
  vendor_id INT,
  FOREIGN KEY (vendor_id) REFERENCES vendor(vendor_id)
);
```

```
CREATE TABLE products (
  product_id INT PRIMARY KEY,
  style VARCHAR(50),
  weight DECIMAL(10,2),
  price DECIMAL(10,2),
  size INT,
  product_type VARCHAR(255),
  product_status VARCHAR(5),
  product_cost DECIMAL(10,2),
  part_id INT,
  FOREIGN KEY (part_id) REFERENCES parts(part_id)
);

CREATE TABLE sale_history (
  sale_id INT PRIMARY KEY,
  sale_time TIMESTAMP,
  salesman VARCHAR(100),
  product_id INT,
  site_id INT,
  personal_id INT,
  FOREIGN KEY (personal_id) REFERENCES people(personal_id),
  FOREIGN KEY (product_id) REFERENCES products(product_id),
  FOREIGN KEY (site_id) REFERENCES marketing_site(site_id),
  FOREIGN KEY (customer_id) REFERENCES customers(personal_id),
  FOREIGN KEY (personal_id) REFERENCES employee(personal_id)
);

CREATE TABLE marketing_site (
  site_id INT PRIMARY KEY,
  site_name VARCHAR(255),
  site_location VARCHAR(50),
  personal_id INT,
  FOREIGN KEY (personal_id) REFERENCES people(personal_id)
);

CREATE TABLE interview_round (
  round_id INT PRIMARY KEY,
  round_number INT,
  grade INT,
  selected_status VARCHAR(10),
  result_date DATE,
  result VARCHAR(255),
  interview_id INT,
  FOREIGN KEY (interview_id) REFERENCES interview(interview_id)
);

CREATE TABLE salary (
  salary_id INT PRIMARY KEY,
  transaction_number INT,
  pay_date DATE,
  amount DECIMAL(10,2)
);

CREATE TABLE works_In (
  personal_id INT,
  department_id INT,
  start_time TIMESTAMP,
  end_time TIMESTAMP,
  job_id INT,
  PRIMARY KEY (personal_id, job_id),
  FOREIGN KEY (personal_id) REFERENCES employee(personal_id),
  FOREIGN KEY (department_id) REFERENCES department(department_id),
  FOREIGN KEY (job_id) REFERENCES job_positions(job_id)
);

CREATE TABLE job_applications (
  personal_id INT,
  job_id INT,
  FOREIGN KEY (personal_id) REFERENCES employee(personal_id),
  FOREIGN KEY (job_id) REFERENCES job_positions(job_id)
);
```

```
CREATE TABLE Employee_Department_History (
  personal_id INT,
  department_id INT,
  job_start_date DATE,
  FOREIGN KEY (personal_id) REFERENCES Employee(personal_id),
  FOREIGN KEY (department_id) REFERENCES Department(department_id)
);

# junctions to connect two tables
CREATE TABLE Sales (
  sales_id INT PRIMARY KEY,
  employee_id INT,
  product_id INT,
  FOREIGN KEY (employee_id) REFERENCES Employee(personal_id),
  FOREIGN KEY (product_id) REFERENCES Products(product_id)
);
```

d) Use the Create View statement to create the following views:

 # View1: This view returns the average salary each employee has earned from the company monthly after she/he becomes an employee in the company.

```
CREATE VIEW View1 AS
SELECT e.personal_id, ROUND(AVG(s.amount)/12, 2) AS average_monthly_salary
FROM Employee e
JOIN Salary s ON e.salary_id = s.salary_id
GROUP BY e.personal_id;
```

personal_id	average_monthly_salary
1	4166.67
2	6666.67
3	9166.67
8	9166.67
9	9166.67
10	9166.67
11	9166.67
4	5416.67
5	5833.33

 # View2: This view returns the number of interviews rounds each interviewee pass for each job position.

```
CREATE VIEW View2 AS
SELECT i.interviewees_id, i.interviewees_name, jp.job_id, COUNT(ir.round_id) AS passed_rounds
FROM Interview i
LEFT JOIN Interview_round ir ON i.interview_id = ir.interview_id
LEFT JOIN Job_positions jp ON i.job_id = jp.job_id
WHERE ir.selected_status = 'YES'
GROUP BY i.interviewees_id, i.interviewees_name, jp.job_id;
```

interviewees_id	interviewees_name	job_id	passed_rounds
5	Hellen Cole	11111	1
3	Jim Williams	33333	1
2	Jane Johnson	44444	1
5	Hellen Cole	12345	1
5	Hellen Cole	33333	1
5	Hellen Cole	44444	1
5	Hellen Cole	55555	1
3	Jim Williams	11111	1
3	Jim Williams	22222	1
3	Jim Williams	44444	1
3	Jim Williams	55555	1

 # View3: This view returns the number of items of each product type sold.

```
CREATE VIEW View3 AS
SELECT p.product_type, COUNT(sh.sale_id) AS items_sold
FROM Products p
JOIN Sale_history sh ON p.product_id = sh.product_id
WHERE p.product_status = 'Sold'
GROUP BY p.product_type;
```

product_type	items_sold
cup	4

 # View4: This view returns the part purchase cost for each product.

```
CREATE VIEW View4 AS
SELECT p.product_id, SUM(pt.price) AS part_cost
FROM Products p
JOIN Parts pt ON p.part_id = pt.part_id
GROUP BY p.product_id;
```

product_id	part_cost
1	100.00
5	100.00
6	100.00
2	200.00
3	400.00
4	900.00

e) Answer the following Queries. Feel free to use any of the views that you created in part (d).

1) Return the ID and Name of interviewers who participate in interviews where the interviewee's name is "Hellen Cole" arranged for job "11111".

interviewer_id	interviewer_name
1	John Smith

2) Return the ID of all jobs which are posted by department "Marketing" in January 2011.

job_id
12345
77777
NULL

3) Return the ID and Name of the employees having no supervisees.

personal_id	fname	lname
1	John	Smith
8	Jim	Williams

4) Return the Id and Location of the marketing sites which have no sale records during March, 2011.

site_id	site_location
1	Location 1
2	Location 2
NULL	NULL

5) Return the job's id and description which does not hire a suitable person one month after it is posted.

job_id	description
55555	IT Support

6) Return the ID and Name of the salesmen who have sold all product type whose price is above $200.

Salesman_ID	Salesman_Name
3	Jim Williams
4	June Jones

7) Return the department's id and name which has no job post during 1/1/2011 and 2/1/2011.

department_id	department_name
2	Sales
3	HR
4	Finance
5	IT
NULL	NULL

8) Return the ID, Name, and Department ID of the existing employees who apply job "12345".

personal_id	fname	lname	department_id
1	John	Smith	2
2	Jane	Johnson	1
3	Jim	Williams	3

9) Return the best seller's type in the company (sold the most items).

salesman	items_sold
John Smith	3

10) Return the product type whose net profit is highest in the company (money earned minus the part cost).

product_id	product_type	net_profit
6	Product Type 6	599.00

11) Return the name and id of the employees who has worked in all departments after hired by the company.

Employee_Name	Number_Of_Departments_Worked
Jim	5

12) Return the name and email address of the interviewee who is selected.

interviewees_name	interviewees_email_address	selected_status
Hellen Cole	hellen.cole@example.com	YES
Jim Williams	jim.williams@example.com	YES
Jane Johnson	jane.johnson@example.com	YES

13) Retrieve the name, phone number, email address of the interviewees selected for all the jobs they apply.

interviewees_name	interviewees_phone_number	interviewees_email_address	Number_of_Departments_Applied	Number_of_Departments_Selected
Jim Williams	1122334455	jim.williams@example.com	5	5
Hellen Cole	3344556677	hellen.cole@example.com	5	5

14) Return the employee's name and id whose average monthly salary is highest in the company.

Employee_Name	Average_Monthly_Salary
Jim	9166.666667

15) Return the ID and Name of the vendor who supply part whose name is "Cup" and weight is smaller than 4 pound and the price is lowest among all vendors.

Vendor_ID	Vendor_Name	Product_Type	Lowest_Price
1	Vendor 1	cup	100.00