Ansheng Li
CS4347

Semester Project

I. Project Description

There is an XYZ Company, which purchases some parts from vendors to produce some products. It has several departments, marketing sites, and parts supply vendors in the company.

Assumptions will be marked with bullet points.

1) For each department, department id and department name will be recorded.
    - Department id should be unique for each department

2) People in the company can be divided into three types -- employees, customers, and potential employees. Each person can belong to more than one type. Each person in the company has the following attributes: Personal_ID, Name (Last Name, First Name), Age (below 65), Gender, Address (address line 1, address line 2, city, state, zip code), and Phone number (one individual may have more than one phone number). For customers, his/her preferred salesmen were recorded in the system. For employees, Rank and Title (e.g. CEO, Principle, Partner, etc.) will be recorded for them.
    - Personal ID should be unique for each Person entity
    - Rank should be unique for each Employee entity with ranking such as 1, 2, 3, …,n.

3) Each employee of the company must have only one direct supervisor, while each supervisor can have several supervisees. One employee can work for one or more departments at different times. But at one time, one employee can only work for one department. The system needs to record start time and end time for each shift among different departments for one employee.
    - An employee can work for one department at a time with their given start time and end time. However, they can work for different departments at different times.
    - We will show a N:1 from Employee entity to Department entity for this EER diagram
    - This would show each Employee can only work for one Department at a time

4) Each job position's information is recorded to hire new people. It contains the Job ID, job description, and posted date in the system.

5) The job positions are posted by the departments. Both existing employees and potential employees can apply for each job post by any department. The company will select some candidates from the applications and make interviews.
    - A Department does not have to post job position assuming the Department has preselected employees, and they can choose to hire new people
    - There may or may not be any potential employees or current employees applying for a job position
    - Interviews may not happen at all if no one applies so it should be a weak entity

- A person can apply to several job positions
- A department can post several job positions

6) For each job position, several interviews will be made to select a suitable person.

7) For each interview, candidates (interviewees), interviewers, job position and interview time are recorded. After each round interview, the interviewers give a grade to it ranging from 0 to 100. The grade over 60 represents that the interviewee pass the interview. One person is selected when her/his average grade is over 70 and she/he passes at least 5 rounds of interviews.

- There can be multiple Interviews for each candidate separated by rounds
- A job position has many interviews
- A weak relation between job position entity and interviews entity as interviews entity is dependent on job position, which may not produce any interviews

8) For each product in the company, the system needs to record Product ID, Product Type, Size, List Price, Weight, and Style.

- A Department must produce the product, 1:N

9) There are many marketing sites for the company. For each site, Site ID, Site Name, and Site Location are recorded.

- A Department must manage the marketing sites, 1:N

10) There are several people working for each site, and meanwhile, one person can work on different sites. It is able to track the details of each sale history --- salesmen, customers, product, sale time, and sites.

- Multivalued for salesmen and customers
- There can be many sale histories for each site
- Each sale histories should have unique salesman, customer, and sale time
- Since a salesman can sell to the same customer but each time is unique

11) Part purchase is also a vital activity in the company. The system needs to record each vendor's Vendor ID, Name, Address, Account Number, Credit Rating, and Purchasing Web Service URL.

- The department is the entity that is purchasing the parts from the vendor
- Vendor ID and account number should be unique per vendor

12) One vendor may supply many types of parts. The price of the same part type may vary from different vendors but the price of one part type of one vendor will keep the same. It is able to track which part types used in each product and the number of each type of part used for the product.

- A product can be supplied by many vendors, but a vendor should supply a part to a product
- Having the same part being supplied by different vendors is counter-intuitive in business

13) In addition, the system maintains the information of each employee's monthly salary which includes transaction number, pay_date, and amount (Note: transaction number could be the same among different employees. However, for each employee, the transaction number is unique).

II. Project Questions

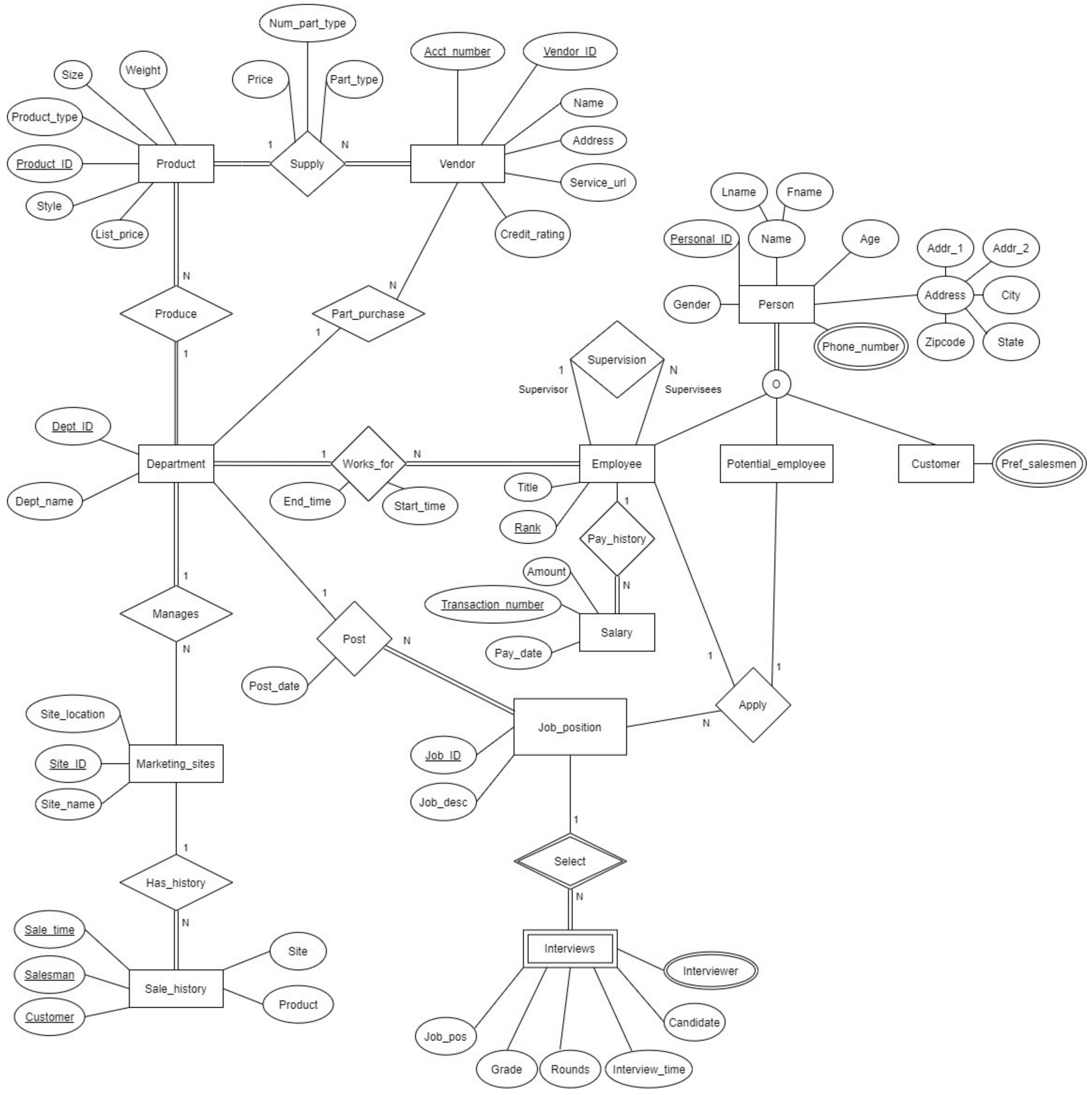
1. Can you think of 5 more rules (other than the one explicitly described above) that are likely to be used in a company.
    a) In regards to the customer entity, there is not a way for a customer to keep track of their orders. This should be an important aspect for the company to keep track of so they can associate a product or similar products to a customer. It can also be used by the customer to keep a history of their purchases from the company.
    b) Another relation that could be added is a department manager. The above listing ruled for supervisors and supervisees amongst the employees, but there is not one for management of a department. It could be the case that a supervisor is a manager but if the company is large enough there could be several supervisors but there should be a head of the department.
    c) Regarding the department entity, it is lacking in other attributes that would make sense for a department within a company. Other attributes such as department location, address, or other branches within that department could be added. Perhaps each of the branches could link to specific marketing sites.
    d) Another attribute the company should keep track of is inventory. There should be an entity that keeps a record of all the finished products they produced monthly, and it should be tracked by the department.
    e) Another aspect we should consider is that there should be a link between the customer attribute in the sales history and the customer entity. They both should be the same and should be connected when referring to one or the other.

2. Is the ability to model superclass/subclass relationships likely to be important in such an environment? Why or why not?
   Modeling superclasses and subclasses are important to the company perspective because a company has a hierarchy in practice. From the described listing above, we can tell that every person that is involved has a role to play such as being an employee, potential employee, and a customer. Even though they are all people, they all play a different role within the company environment. We can use common attributes for all people and then give more to them depending on the subclasses they fall under. Furthermore, we can derive more subclasses within each of the subclasses to create greater specialization. We can make different roles for the employee if we wanted to, but keeping a simple rank attribute is just as effective.

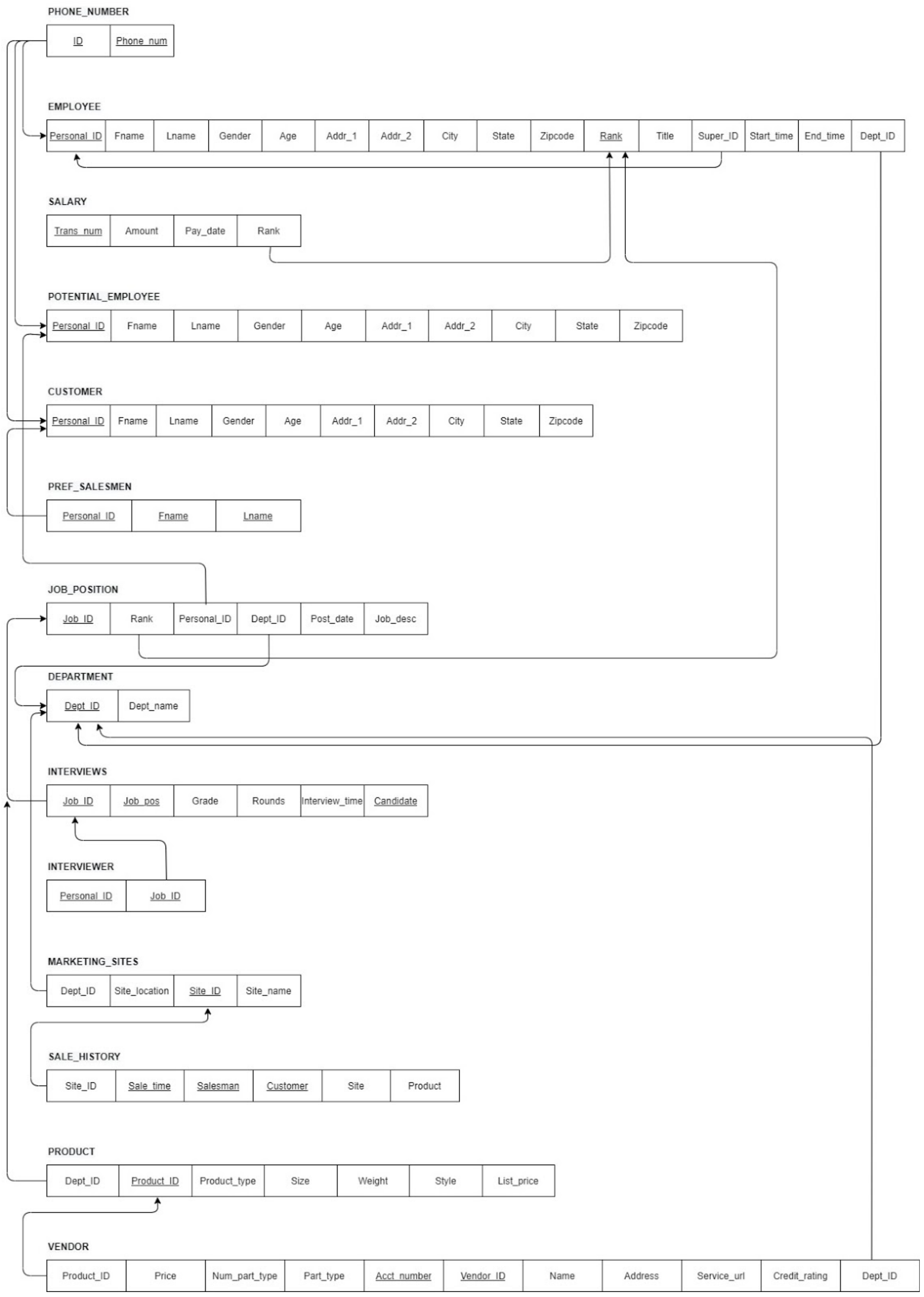
3. Justify using a Relational DBMS like Oracle for this project.
   A relational DBMS is very efficient to use and is scalable when a more complex structure is needed by the database designer. By using primary keys as the identifier for a given

table, organization of data is swift and more robust. In addition, using the foreign keys can hold the data accountable so that redundancy can be handled properly as Oracle denies duplication of primary keys.In addition, relational databases possess physical data independence. This refers to a system's capacity to make changes to the inner schema without altering the external schemas or application programs.

III. EER Diagram



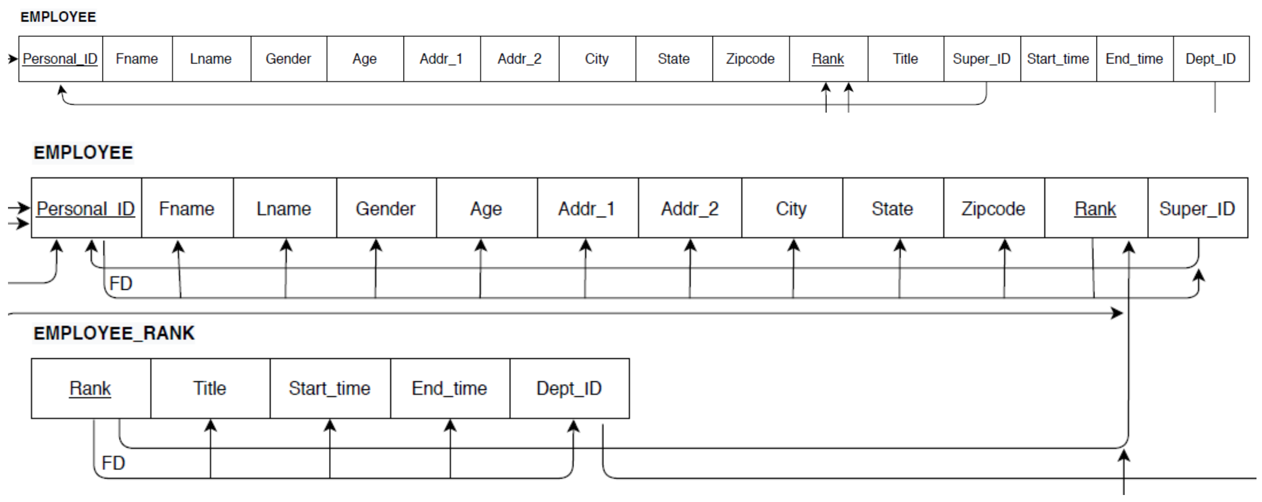
IV. Logical Schema



V. Normalization

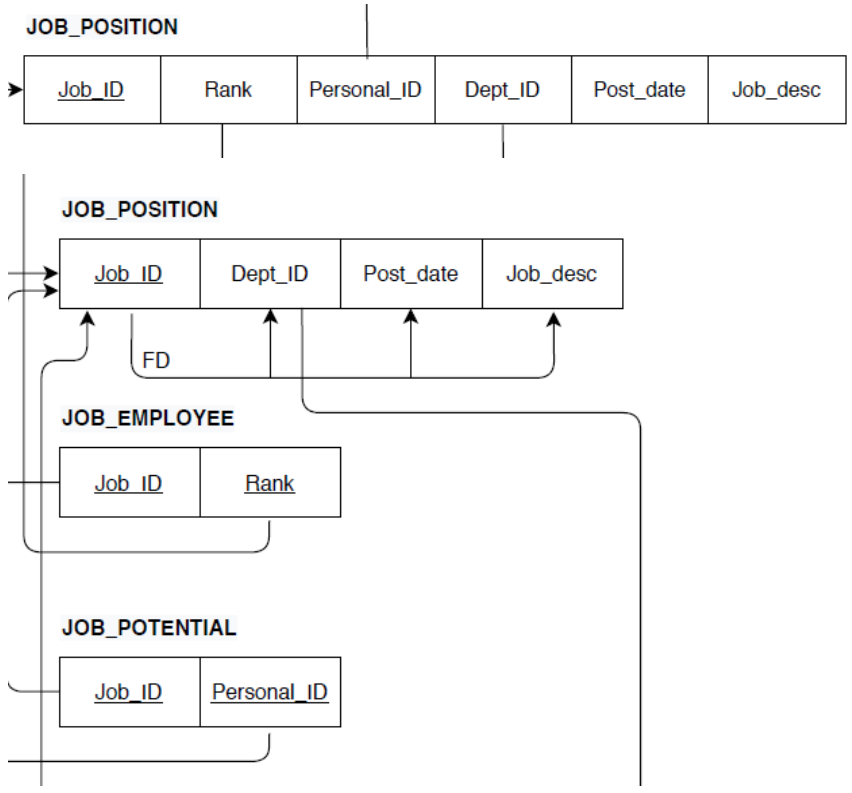
EMPLOYEE

We want to separate PK rank into a new table EMPLOYEE_RANK because the rank determines the title, start time, end time and department ID. The personal ID has functional dependencies on the names and address and such.



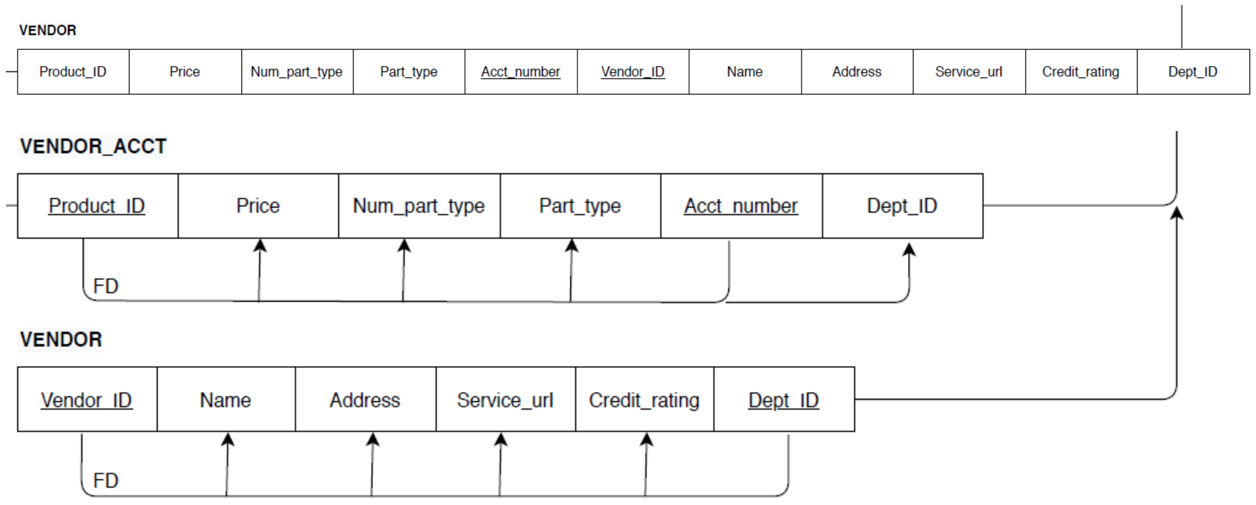
JOB_POSITION

We want to separate the table for employees that apply and potential new employees. Since an employee can apply any number of times, we will need a composite key of job ID and the employee rank. This will be a lookup table so we can associate an employee to many job IDs or a job ID to many employees. We shall do the same for potential employees. This would be 3NF.



VENDOR

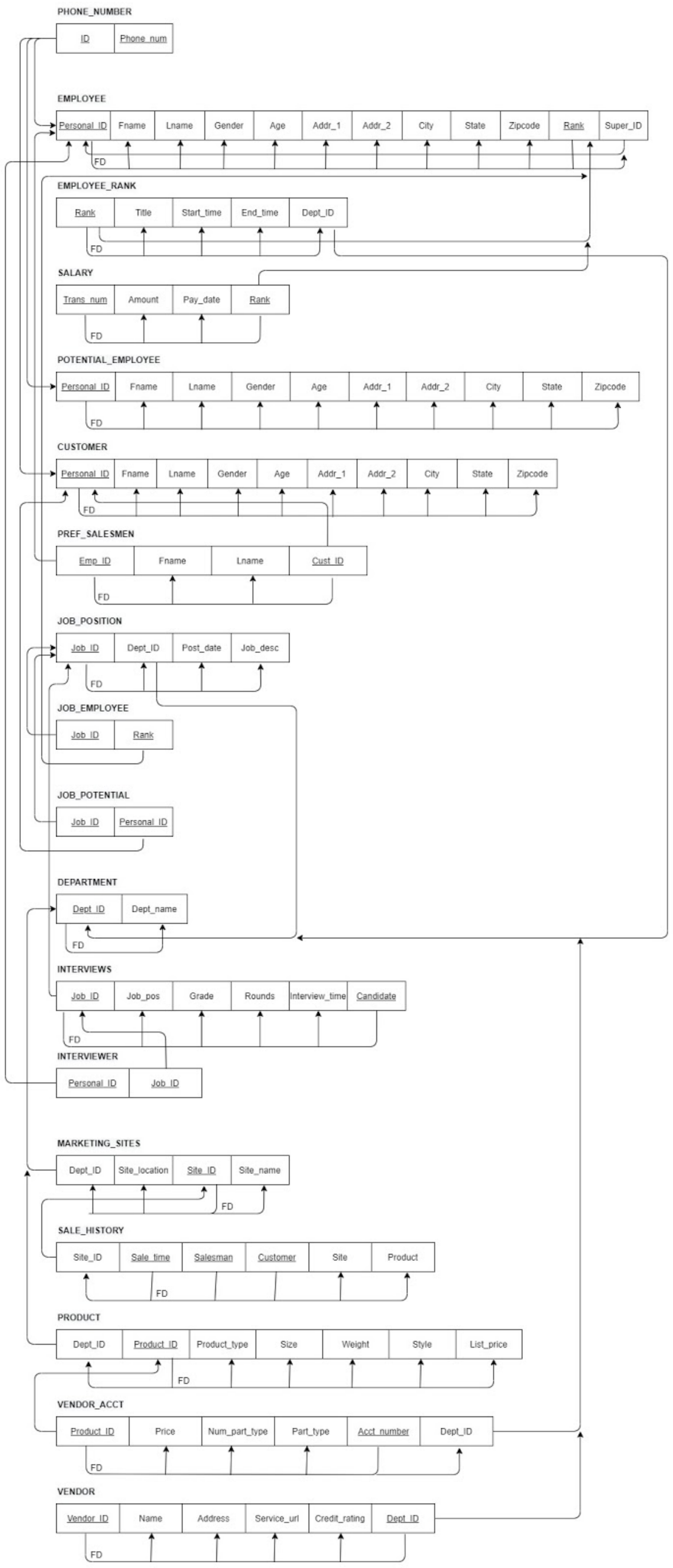
The vendor can be decomposed to two new tables because the department can keep track of which vendor is supplying them. The product ID and account number should be the primary keys that act as functional dependencies for price, number of part type, part type, and department ID. This means they identify for every part and for every department. The vendor ID and department ID gives us the information relating to the vendor such as name, address, service URL, and credit rating.



Every other table cannot be decomposed further as it would make keeping track of data unmanageable. The other tables are small enough or make common sense based on our assumption to be kept as it is. All multivalued and composite values have been separated into columns, all assumed partial dependencies and transitive dependencies have been accounted for based on our assumption.

VI. Dependency Diagram

See the next page. Also included in file *Project_Dependency.jpg.*


PHONE_NUMBER
ID | Phone_num
EMPLOYEE
Personal_ID | Fname | Lname | Gender | Age | Addr_1 | Addr_2 | City | State | Zipcode | Rank | Super_ID
FD
EMPLOYEE_RANK
Rank | Title | Start_time | End_time | Dept_ID
FD
SALARY
Trans_num | Amount | Pay_date | Rank
FD
POTENTIAL_EMPLOYEE
Personal_ID | Fname | Lname | Gender | Age | Addr_1 | Addr_2 | City | State | Zipcode
FD
CUSTOMER
Personal_ID | Fname | Lname | Gender | Age | Addr_1 | Addr_2 | City | State | Zipcode
FD
PREF_SALESMEN
Emp_ID | Fname | Lname | Cust_ID
FD
JOB_POSITION
Job_ID | Dept_ID | Post_date | Job_desc
FD
JOB_EMPLOYEE
Job_ID | Rank
JOB_POTENTIAL
Job_ID | Personal_ID
DEPARTMENT
Dept_ID | Dept_name
FD
INTERVIEWS
Job_ID | Job_pos | Grade | Rounds | Interview_time | Candidate
FD
INTERVIEWER
Personal_ID | Job_ID
MARKETING_SITES
Dept_ID | Site_location | Site_ID | Site_name
FD
SALE_HISTORY
Site_ID | Sale_time | Salesman | Customer | Site | Product
FD
PRODUCT
Dept_ID | Product_ID | Product_type | Size | Weight | Style | List_price
FD
VENDOR_ACCT
Product_ID | Price | Num_part_type | Part_type | Acct_number | Dept_ID
FD
VENDOR
Vendor_ID | Name | Address | Service_url | Credit_rating | Dept_ID
FD

VII. Physical Schema

Write SQL statements to create databases, 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.

See file *xyz_company_tables.sql*

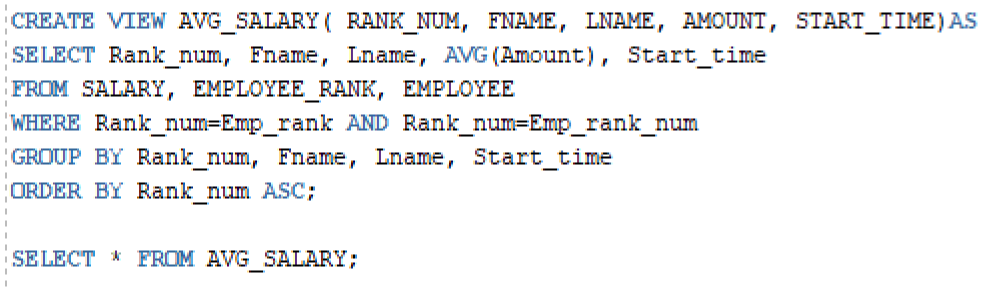
VIII. Sample Data Insertion

See file *data_insertion.sql*

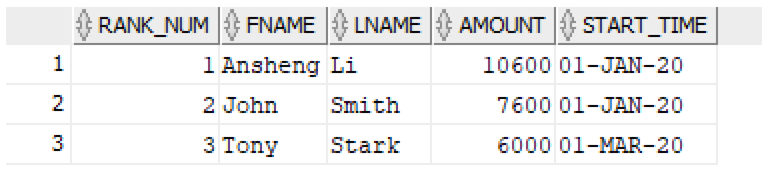
IX. Views

The following views shown below are captured from SQLDeveloper and the file can be seen in *views.sql*.

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

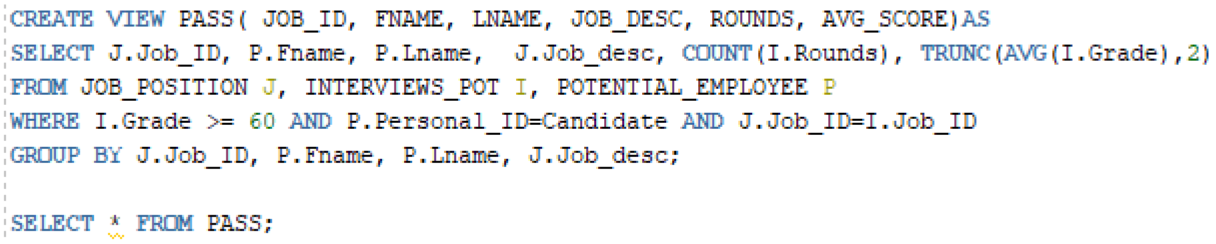


```
CREATE VIEW AVG_SALARY( RANK_NUM, FNAME, LNAME, AMOUNT, START_TIME)AS
SELECT Rank_num, Fname, Lname, AVG(Amount), Start_time
FROM SALARY, EMPLOYEE_RANK, EMPLOYEE
WHERE Rank_num=Emp_rank AND Rank_num=Emp_rank_num
GROUP BY Rank_num, Fname, Lname, Start_time
ORDER BY Rank_num ASC;

SELECT * FROM AVG_SALARY;
```

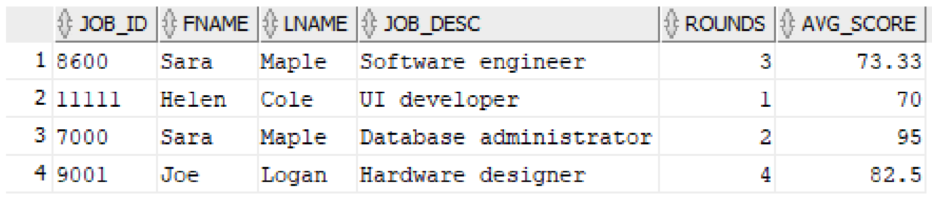


	RANK_NUM	FNAME	LNAME	AMOUNT	START_TIME
1	1	Ansheng	Li	10600	01-JAN-20
2	2	John	Smith	7600	01-JAN-20
3	3	Tony	Stark	6000	01-MAR-20

2) View2: This view returns the number of interview rounds each interviewee pass for each job position.

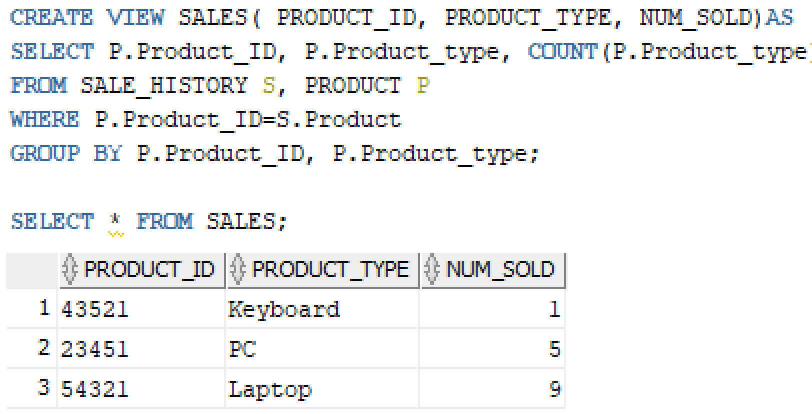


```
CREATE VIEW PASS( JOB_ID, FNAME, LNAME, JOB_DESC, ROUNDS, AVG_SCORE)AS
SELECT J.Job_ID, P.Fname, P.Lname,  J.Job_desc, COUNT(I.Rounds), TRUNC(AVG(I.Grade),2)
FROM JOB_POSITION J, INTERVIEWS_POT I, POTENTIAL_EMPLOYEE P
WHERE I.Grade >= 60 AND P.Personal_ID=Candidate AND J.Job_ID=I.Job_ID
GROUP BY J.Job_ID, P.Fname, P.Lname, J.Job_desc;

SELECT * FROM PASS;
```



	JOB_ID	FNAME	LNAME	JOB_DESC	ROUNDS	AVG_SCORE
1	8600	Sara	Maple	Software engineer	3	73.33
2	11111	Helen	Cole	UI developer	1	70
3	7000	Sara	Maple	Database administrator	2	95
4	9001	Joe	Logan	Hardware designer	4	82.5

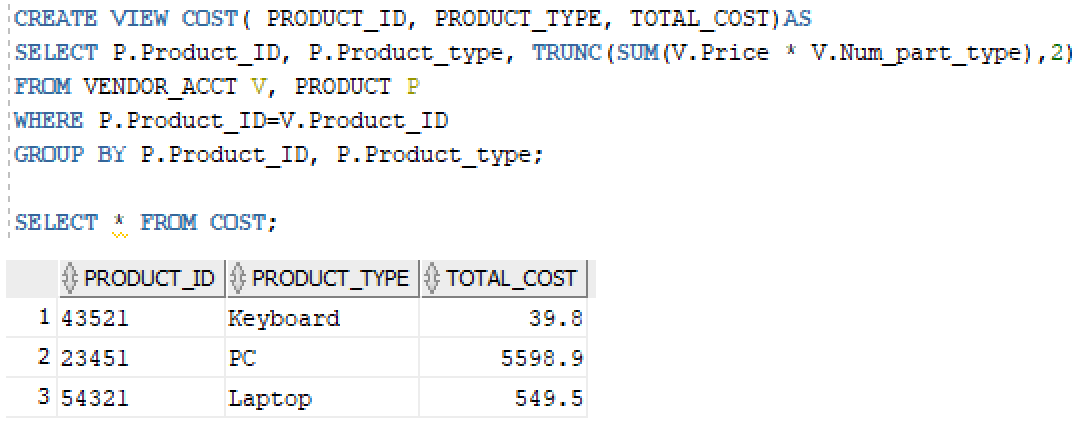
3) View3: This view returns the number of items of each product type sold.



```sql
CREATE VIEW SALES( PRODUCT_ID, PRODUCT_TYPE, NUM_SOLD)AS
SELECT P.Product_ID, P.Product_type, COUNT(P.Product_type)
FROM SALE_HISTORY S, PRODUCT P
WHERE P.Product_ID=S.Product
GROUP BY P.Product_ID, P.Product_type;

SELECT * FROM SALES;
```

	PRODUCT_ID	PRODUCT_TYPE	NUM_SOLD
1	43521	Keyboard	1
2	23451	PC	5
3	54321	Laptop	9

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



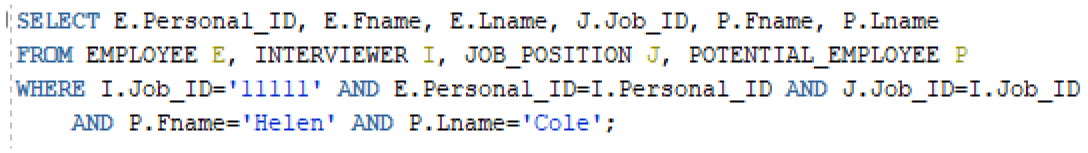
```sql
CREATE VIEW COST( PRODUCT_ID, PRODUCT_TYPE, TOTAL_COST)AS
SELECT P.Product_ID, P.Product_type, TRUNC(SUM(V.Price * V.Num_part_type),2)
FROM VENDOR_ACCT V, PRODUCT P
WHERE P.Product_ID=V.Product_ID
GROUP BY P.Product_ID, P.Product_type;

SELECT * FROM COST;
```

	PRODUCT_ID	PRODUCT_TYPE	TOTAL_COST
1	43521	Keyboard	39.8
2	23451	PC	5598.9
3	54321	Laptop	549.5

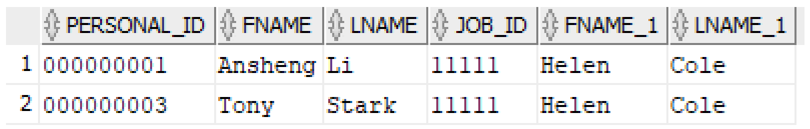
X. Additional Queries

The following query results shown below are captured from SQLDeveloper and the file can be seen in *queries.sql*.

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

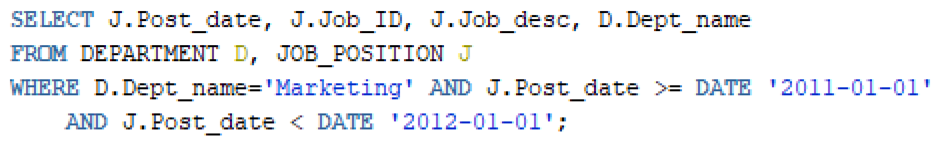


```
SELECT E.Personal_ID, E.Fname, E.Lname, J.Job_ID, P.Fname, P.Lname
FROM EMPLOYEE E, INTERVIEWER I, JOB_POSITION J, POTENTIAL_EMPLOYEE P
WHERE I.Job_ID='11111' AND E.Personal_ID=I.Personal_ID AND J.Job_ID=I.Job_ID
    AND P.Fname='Helen' AND P.Lname='Cole';
```

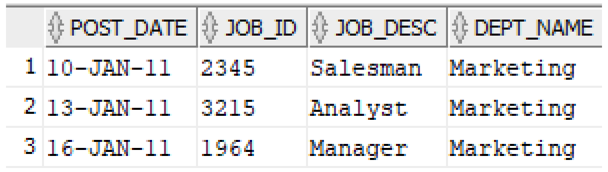


	PERSONAL_ID	FNAME	LNAME	JOB_ID	FNAME_1	LNAME_1
1	000000001	Ansheng	Li	11111	Helen	Cole
2	000000003	Tony	Stark	11111	Helen	Cole

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

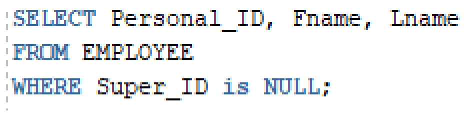


```
SELECT J.Post_date, J.Job_ID, J.Job_desc, D.Dept_name
FROM DEPARTMENT D, JOB_POSITION J
WHERE D.Dept_name='Marketing' AND J.Post_date >= DATE '2011-01-01'
    AND J.Post_date < DATE '2012-01-01';
```

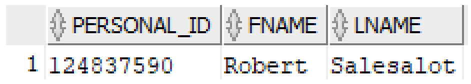


	POST_DATE	JOB_ID	JOB_DESC	DEPT_NAME
1	10-JAN-11	2345	Salesman	Marketing
2	13-JAN-11	3215	Analyst	Marketing
3	16-JAN-11	1964	Manager	Marketing

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

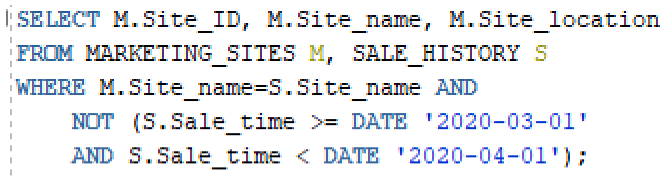


```
SELECT Personal_ID, Fname, Lname
FROM EMPLOYEE
WHERE Super_ID is NULL;
```

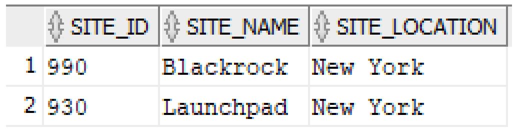


	PERSONAL_ID	FNAME	LNAME
1	124837590	Robert	Salesalot

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

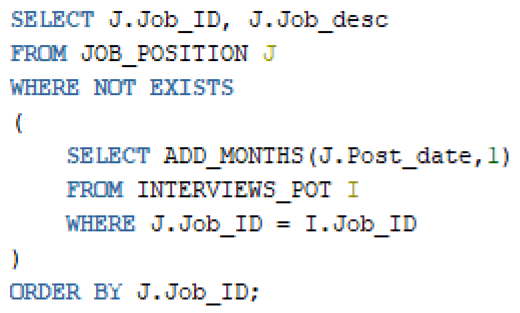


```
SELECT M.Site_ID, M.Site_name, M.Site_location
FROM MARKETING_SITES M, SALE_HISTORY S
WHERE M.Site_name=S.Site_name AND
    NOT (S.Sale_time >= DATE '2020-03-01'
    AND S.Sale_time < DATE '2020-04-01');
```

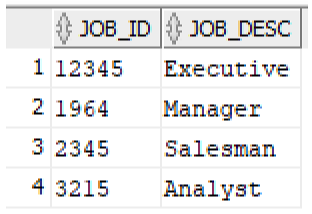


	SITE_ID	SITE_NAME	SITE_LOCATION
1	990	Blackrock	New York
2	930	Launchpad	New York

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

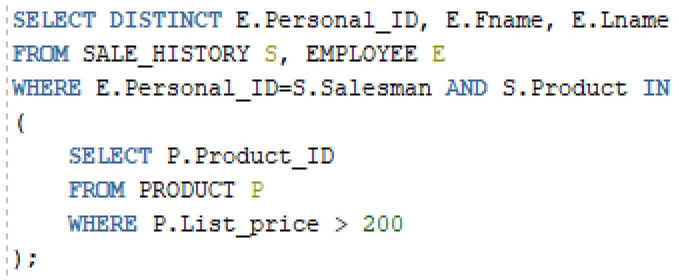


```
SELECT J.Job_ID, J.Job_desc
FROM JOB_POSITION J
WHERE NOT EXISTS
(
    SELECT ADD_MONTHS(J.Post_date,1)
    FROM INTERVIEWS_POT I
    WHERE J.Job_ID = I.Job_ID
)
ORDER BY J.Job_ID;
```

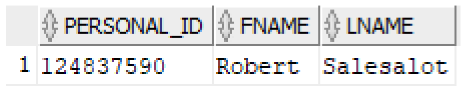


	JOB_ID	JOB_DESC
1	12345	Executive
2	1964	Manager
3	2345	Salesman
4	3215	Analyst

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

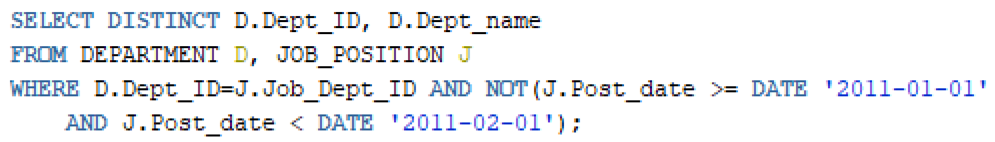


```
SELECT DISTINCT E.Personal_ID, E.Fname, E.Lname
FROM SALE_HISTORY S, EMPLOYEE E
WHERE E.Personal_ID=S.Salesman AND S.Product IN
(
    SELECT P.Product_ID
    FROM PRODUCT P
    WHERE P.List_price > 200
);
```

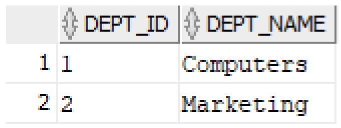


	PERSONAL_ID	FNAME	LNAME
1	124837590	Robert	Salesalot

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

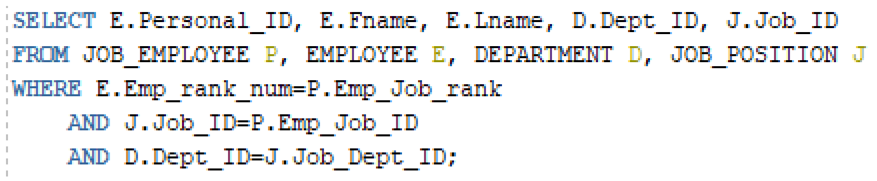


```
SELECT DISTINCT D.Dept_ID, D.Dept_name
FROM DEPARTMENT D, JOB_POSITION J
WHERE D.Dept_ID=J.Job_Dept_ID AND NOT(J.Post_date >= DATE '2011-01-01'
    AND J.Post_date < DATE '2011-02-01');
```

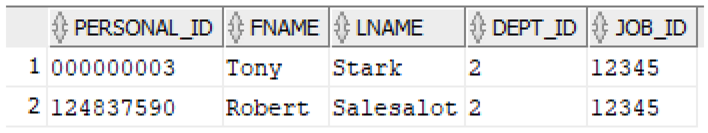


	DEPT_ID	DEPT_NAME
1	1	Computers
2	2	Marketing

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

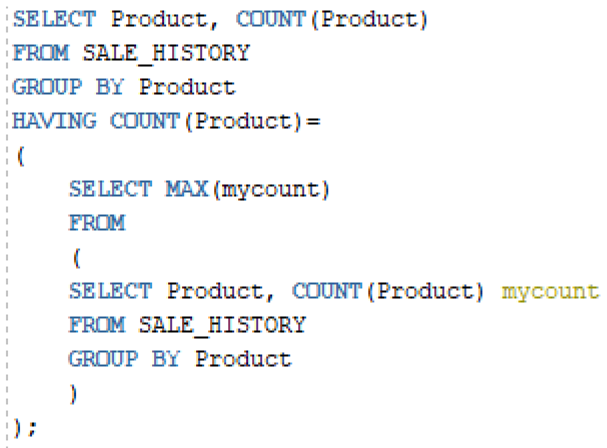


```
SELECT E.Personal_ID, E.Fname, E.Lname, D.Dept_ID, J.Job_ID
FROM JOB_EMPLOYEE P, EMPLOYEE E, DEPARTMENT D, JOB_POSITION J
WHERE E.Emp_rank_num=P.Emp_Job_rank
    AND J.Job_ID=P.Emp_Job_ID
    AND D.Dept_ID=J.Job_Dept_ID;
```

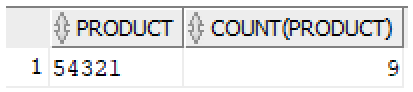


	PERSONAL_ID	FNAME	LNAME	DEPT_ID	JOB_ID
1	000000003	Tony	Stark	2	12345
2	124837590	Robert	Salesalot	2	12345

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

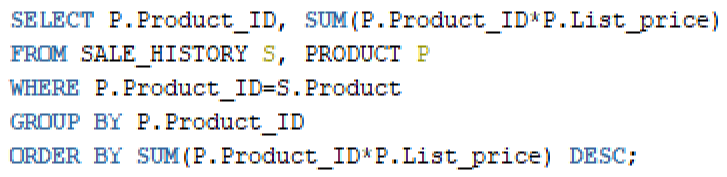


```
SELECT Product, COUNT(Product)
FROM SALE_HISTORY
GROUP BY Product
HAVING COUNT(Product)=
(
    SELECT MAX(mycount)
    FROM
    (
    SELECT Product, COUNT(Product) mycount
    FROM SALE_HISTORY
    GROUP BY Product
    )
);
```

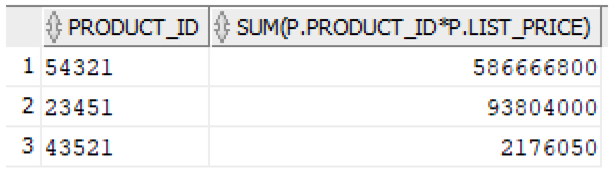


	PRODUCT	COUNT(PRODUCT)
1	54321	9

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

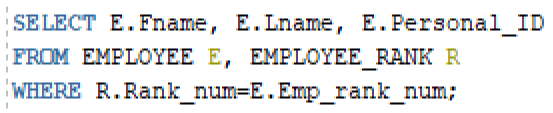


```
SELECT P.Product_ID, SUM(P.Product_ID*P.List_price)
FROM SALE_HISTORY S, PRODUCT P
WHERE P.Product_ID=S.Product
GROUP BY P.Product_ID
ORDER BY SUM(P.Product_ID*P.List_price) DESC;
```

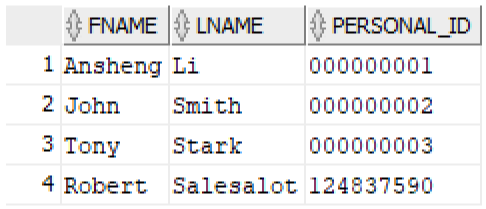


	PRODUCT_ID	SUM(P.PRODUCT_ID*P.LIST_PRICE)
1	54321	586666800
2	23451	93804000
3	43521	2176050

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

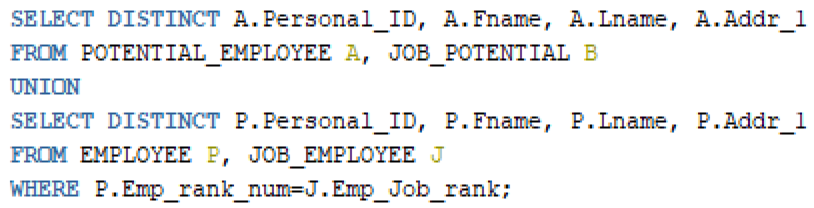


```sql
SELECT E.Fname, E.Lname, E.Personal_ID
FROM EMPLOYEE E, EMPLOYEE_RANK R
WHERE R.Rank_num=E.Emp_rank_num;
```

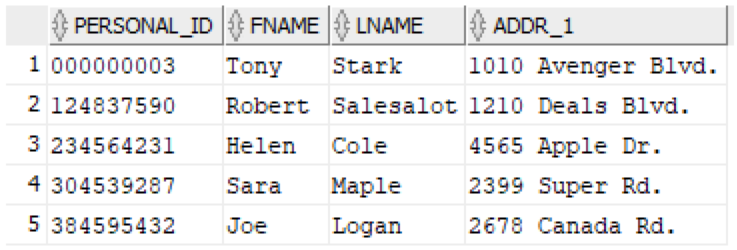


	FNAME	LNAME	PERSONAL_ID
1	Ansheng	Li	000000001
2	John	Smith	000000002
3	Tony	Stark	000000003
4	Robert	Salesalot	124837590

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

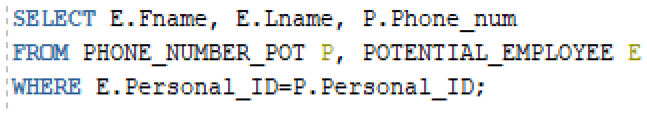


```sql
SELECT DISTINCT A.Personal_ID, A.Fname, A.Lname, A.Addr_1
FROM POTENTIAL_EMPLOYEE A, JOB_POTENTIAL B
UNION
SELECT DISTINCT P.Personal_ID, P.Fname, P.Lname, P.Addr_1
FROM EMPLOYEE P, JOB_EMPLOYEE J
WHERE P.Emp_rank_num=J.Emp_Job_rank;
```

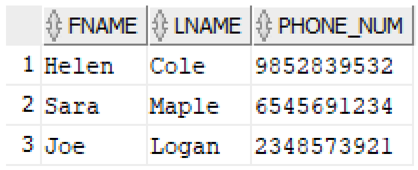


	PERSONAL_ID	FNAME	LNAME	ADDR_1
1	000000003	Tony	Stark	1010 Avenger Blvd.
2	124837590	Robert	Salesalot	1210 Deals Blvd.
3	234564231	Helen	Cole	4565 Apple Dr.
4	304539287	Sara	Maple	2399 Super Rd.
5	384595432	Joe	Logan	2678 Canada Rd.

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

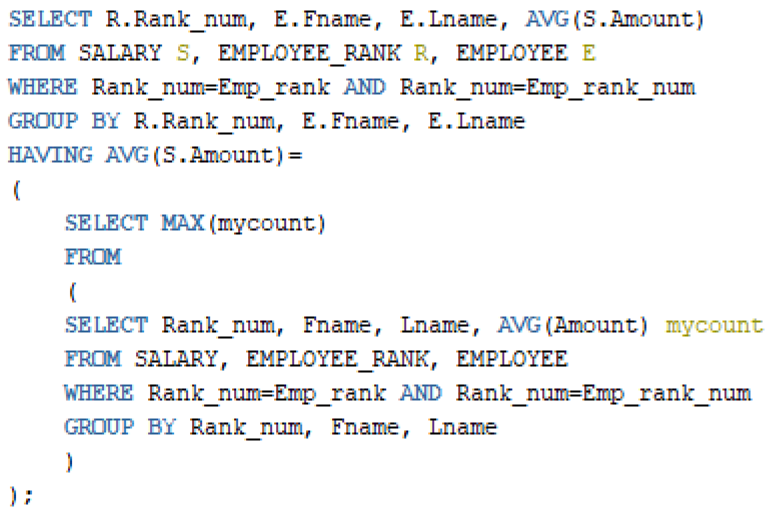


```sql
SELECT E.Fname, E.Lname, P.Phone_num
FROM PHONE_NUMBER_POT P, POTENTIAL_EMPLOYEE E
WHERE E.Personal_ID=P.Personal_ID;
```

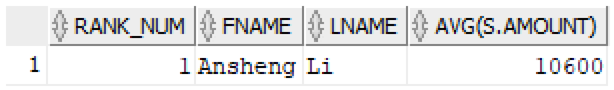


	FNAME	LNAME	PHONE_NUM
1	Helen	Cole	9852839532
2	Sara	Maple	6545691234
3	Joe	Logan	2348573921

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

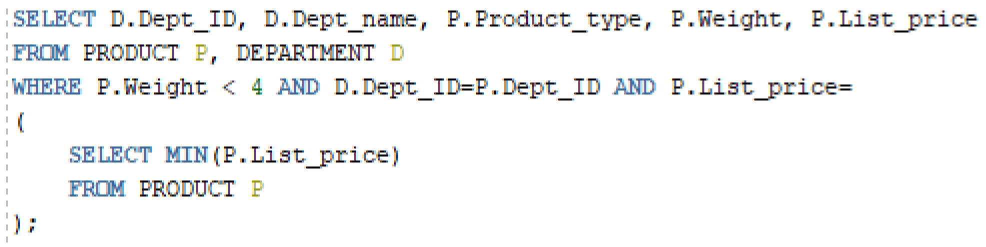


```
SELECT R.Rank_num, E.Fname, E.Lname, AVG(S.Amount)
FROM SALARY S, EMPLOYEE_RANK R, EMPLOYEE E
WHERE Rank_num=Emp_rank AND Rank_num=Emp_rank_num
GROUP BY R.Rank_num, E.Fname, E.Lname
HAVING AVG(S.Amount)=
(
    SELECT MAX(mycount)
    FROM
    (
    SELECT Rank_num, Fname, Lname, AVG(Amount) mycount
    FROM SALARY, EMPLOYEE_RANK, EMPLOYEE
    WHERE Rank_num=Emp_rank AND Rank_num=Emp_rank_num
    GROUP BY Rank_num, Fname, Lname
    )
);
```

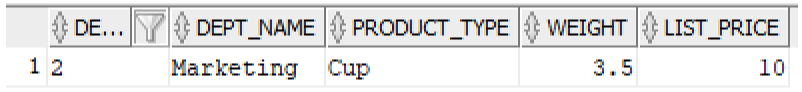


	RANK_NUM	FNAME	LNAME	AVG(S.AMOUNT)
1	1	Ansheng	Li	10600

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.



```
SELECT D.Dept_ID, D.Dept_name, P.Product_type, P.Weight, P.List_price
FROM PRODUCT P, DEPARTMENT D
WHERE P.Weight < 4 AND D.Dept_ID=P.Dept_ID AND P.List_price=
(
    SELECT MIN(P.List_price)
    FROM PRODUCT P
);
```



	DE...	DEPT_NAME	PRODUCT_TYPE	WEIGHT	LIST_PRICE
1	2	Marketing	Cup	3.5	10