

## **Problem Description:**

There is an ABC company that wants to build a database system to store information about their employees and their monthly salary, departments and its job positions, and interviews with future candidates. The company also wants to store data about their vendors supplying parts to make a product that sells out to customers in different marketing locations.

- **Individual Contribution Breakdown:**
- EER Diagram and assumptions: Dail Kang
- Convert EER to conceptual design: Dail Kang
- Normalizing the tables: Al Wasee Mahmood
- Dependency diagram: Al Wasee Mahmood
- SQL statements to create database: Muhammad Zubair
- Creating Views: Muhammad Zubair and Al Wasee Mahmood
- Retrieval queries: 1/3 by each person in the group
- Documenting the final project: Muhammad Zubair
- Forming group and setting up group meetings: Dail Kang

## **Project questions:**

### **EER diagram with all assumptions:**

#### **ASSUMPTIONS:**

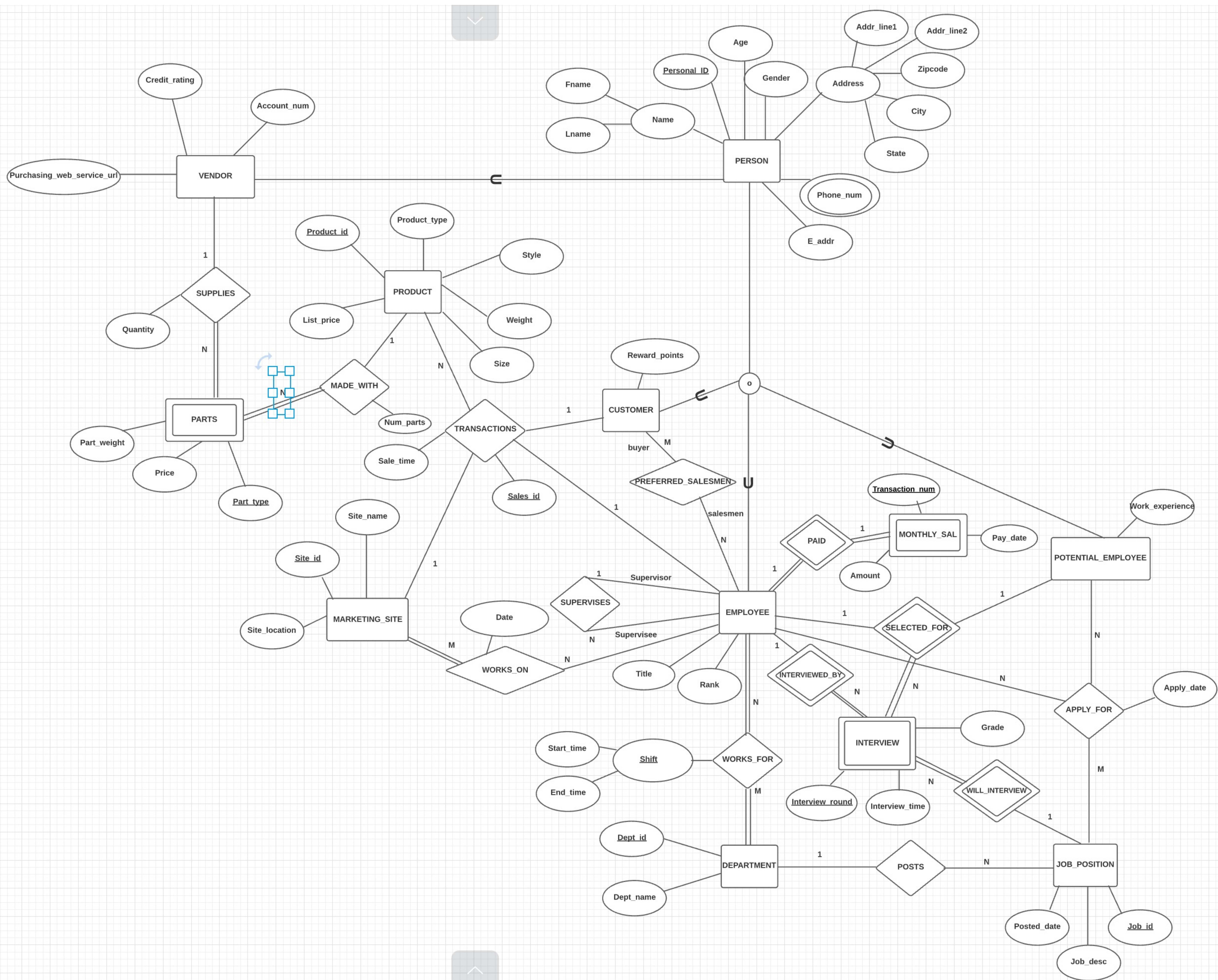
- VENDOR, EMPLOYEE, POTENTIAL\_EMPLOYEE being subclasses of a PERSON superclass as they may share common attributes and this PERSON superclass corresponds to all people related to the company.
- Start and end time of each employee shift describes both exact date and time.
- The database keeps track of each person's email address
- A certain job position can be posted by only one Department.
- An employee can conduct multiple interviews, but an interview has only one interviewer employee.
- Potential\_employees' work experience is recorded, and it should be in months.

- The rewards points of all customers are recorded, and each new customer starts off with zero rewards points.
- Each sale transaction made by the company has to have a unique sales\_id to uniquely identify each sale.
- An employee may work on multiple marketing sites and the date they work on each site is recorded.
- The number of a certain part type to make a product is recorded.
- The quantity of a certain part type supplied by a vendor is recorded.

Link to EER:

[https://lucid.app/lucidchart/31802d89-2234-43d7-9592-d01a32b3eb12/edit?invitationId=inv\\_372fd719-9a31-4fa3-a8c6-0ab0929bd558](https://lucid.app/lucidchart/31802d89-2234-43d7-9592-d01a32b3eb12/edit?invitationId=inv_372fd719-9a31-4fa3-a8c6-0ab0929bd558)







## **DATABASE SCHEMA BEFORW NORMALIZATION:**

### **PERSON**

<u>Personal_id</u>	Fname	Lname	Age	Gender	Addr_line1	Addr_line2	Zipcode	City	State	E_addr
--------------------	-------	-------	-----	--------	------------	------------	---------	------	-------	--------

### **EMPLOYEE**

<u>Employee_id</u>	Rank	Title	Super_id
--------------------	------	-------	----------

Foreign key: Employee\_id references Personal\_id {**PERSON**}

Super\_id references Personal\_id {**PERSON**}

### **CUSTOMER**

<u>Customer_id</u>	Rewards_points
--------------------	----------------

Foreign key: Customer\_id references Personal\_id {**PERSON**}

### **PHONE\_NUMBER**

<u>Personal_id</u>	Phone_num
--------------------	-----------

Foreign key: Personal\_id references Personal\_id {**PERSON**}

### **POTENTIAL\_EMPLOYEE**

<u>Personal_id</u>	Work_experience
--------------------	-----------------

Foreign key: Personal\_id references Personal\_id {**PERSON**}

### **VENDOR**

<u>Vendor_id</u>	Purchasing_web_service_url	Credit_rating	Account_num
------------------	----------------------------	---------------	-------------

Foreign key: Vendor\_id references Personal\_id {**PERSON**}

## DEPARTMENT

<u>Dept_id</u>	Dept_name
----------------	-----------

## JOB\_POSITION

<u>Job_id</u>	Dept_id	Job_desc	Posted_date
---------------	---------	----------	-------------

Foreign key: Dept\_id references Dept\_id {**DEPARTMENT**}

## MARKETING\_SITE

<u>Site_id</u>	Site_location	Site_name
----------------	---------------	-----------

## PRODUCT

<u>Product_id</u>	Product_type	List_price	Style	Weight	Size
-------------------	--------------	------------	-------	--------	------

## PARTS

<u>Part_type</u>	<u>Vendor_id</u>	Quantity_supplied	<u>Product_id</u>	Num_for_product	Price	Part_weight
------------------	------------------	-------------------	-------------------	-----------------	-------	-------------

Foreign key: Vendor\_id references Personal\_id (**PERSON**)

Product\_id references Product\_id (**PRODUCT**)

## INTERVIEW

<u>Interview_round</u>	<u>Interviewer_id</u>	<u>Interviewee_id</u>	<u>Job_id</u>	Grade	Interview_time
------------------------	-----------------------	-----------------------	---------------	-------	----------------

Foreign key: Interviewer\_id references Personal\_id (**PERSON**)

Interviewee\_id references Personal\_id (**PERSON**)

Job\_id references Job\_id (JOB\_POSITION)

## MONTHLY\_SAL

<u>Employee_id</u>	<u>Transaction_num</u>	Pay_date	Amount
--------------------	------------------------	----------	--------

Foreign key: Employee\_id references Personal\_id {**PERSON**}

## WORKS\_FOR

<u>Dept_id</u>	<u>Employee_id</u>	<u>Start_time</u>	<u>End_time</u>
----------------	--------------------	-------------------	-----------------

Foreign key: Dept\_id references Dept\_id {**DEPARTMENT**}

Employee\_id references Personal\_id(**PERSON**)

### **WORKS\_ON**

<u>Site_id (FK)</u>	<u>Employee_id</u>	Date
---------------------	--------------------	------

Foreign key: Site\_id references Site\_id {**MARKETING\_SITE**}

Employee\_id references Personal\_id {**PERSON**}

### **APPLY\_FOR**

<u>Applicant_id</u>	<u>Job_id</u>	Apply_date
---------------------	---------------	------------

Foreign key: Applicant\_id references Personal\_id (**PERSON**)

Job\_id references Job\_id (**JOB\_POSITION**)

### **TRANSACTIONS**

<u>Sales_id</u>	<u>Product_id</u>	Customer_id	Salesman_id	Site_id	Sale_time
-----------------	-------------------	-------------	-------------	---------	-----------

Foreign key: Site\_id references Site\_id {**MARKETING\_SITE**}

Salesman\_id references Personal\_id {**PERSON**}

Customer\_id references Personal\_id {**PERSON**}

Product\_id references Product\_id {**PRODUCT**}

### **PREFERRED\_SALESMEN**

<u>Customer_id</u>	<u>Employee_id (FK)</u>
--------------------	-------------------------

Foreign key: Customer\_id references Personal\_id {**PERSON**}

Employee\_id references Personal\_id {**PERSON**}

## **Relation schema after normalization:**

**PERSON1A**

<u>Personal_id</u>	Fname	Lname	Age	Gender	Addr_line1	Addr_line2	City	Zipcode	E_addr
--------------------	-------	-------	-----	--------	------------	------------	------	---------	--------

#### PERSON1B

<u>Zipcode</u>	State
----------------	-------

Foreign key: Zipcode references Zipcode (**PERSON1A**)

#### EMPLOYEE

<u>Employee_id</u>	Rank	Title	Super_id
--------------------	------	-------	----------

Foreign key: Employee\_id references Personal\_id (**PERSON1A**)

Super\_id references Personal\_id (**PERSON1A**)

#### CUSTOMER

<u>Customer_id</u>	Rewards_points
--------------------	----------------

Foreign key: Customer\_id references Personal\_id { **PERSON1A** }

#### PHONE\_NUMBER

<u>Personal_id</u>	Phone_num
--------------------	-----------

#### POTENTIAL\_EMPLOYEE

<u>Personal_id</u>	Work_experience
--------------------	-----------------

Foreign key: Personal\_id references Personal\_id { **PERSON1A** }

#### VENDOR1

<u>Vendor_id</u>	Account_num
------------------	-------------

Foreign key: Vendor\_id references Personal\_id { **PERSON1A** }

#### VENDOR2

<u>Account_num</u>	Credit_rating	Purchasing_web_service_url
--------------------	---------------	----------------------------

Foreign key: Account\_num references Account\_num (**VENDOR1**)

#### DEPARTMENT

<u>Dept_id</u>	Dept_name
----------------	-----------

#### JOB\_POSITION

<u>Job_id</u>	Dept_id	Job_desc	Posted_date
---------------	---------	----------	-------------

Foreign key: Dept\_id references Dept\_id {**DEPARTMENT**}

#### MARKETING\_SITE

<u>Site_id</u>	Site_location	Site_name
----------------	---------------	-----------

**PRODUCT**

<u>Product_id</u>	Product_type	List_price	Style	Weight	Size
-------------------	--------------	------------	-------	--------	------

**PARTS1**

<u>Part_type</u>	<u>Vendor_id</u>	Quantity_supplied	Price	Part_weight
------------------	------------------	-------------------	-------	-------------

Foreign key: Vendor\_id references Personal\_id (**PERSON1A**)

**PARTS2**

<u>Part_type</u>	<u>Product_id</u>	Num_for_product
------------------	-------------------	-----------------

Foreign key: Part\_type references Part\_type (**PARTS1**)  
 Product\_id references Product\_id (**PRODUCT**)

**INTERVIEW**

<u>Interview_round</u>	<u>Interviewer_id</u>	<u>Interviewee_id</u>	<u>Job_id</u>	Grade	Interview_time
------------------------	-----------------------	-----------------------	---------------	-------	----------------

Foreign key: Interviewer\_id references Personal\_id (**PERSON1A**)  
 Interviewee\_id references Personal\_id (**PERSON1A**)  
 Job\_id references Job\_id (**JOB\_POSITION**)

**MONTHLY\_SAL**

<u>Employee_id</u>	<u>Transaction_num</u>	Pay_date	Amount
--------------------	------------------------	----------	--------

Foreign key: Employee\_id references Personal\_id {**PERSON1A**}

**WORKS\_FOR**

<u>Employee_id</u>	<u>Dept_id</u>	<u>Start_time</u>	<u>End_time</u>
--------------------	----------------	-------------------	-----------------

Foreign key: Dept\_id references Dept\_id {**DEPARTMENT**}  
 Employee\_id references Personal\_id(**PERSON1A**)

**WORKS\_ON**

<u>Site_id</u>	<u>Employee_id</u>	Date
----------------	--------------------	------

Foreign key: Site\_id references Site\_id {**MARKETING\_SITE**}  
 Employee\_id references Personal\_id {**PERSON1A**}

**APPLY\_FOR**

<u>Applicant_id</u>	<u>Job_id</u>	Apply_date
---------------------	---------------	------------

Foreign key: Applicant\_id references Personal\_id (**PERSON1A**)  
 Job\_id references Job\_id (**JOB\_POSITION**)

**PREFERRED\_SALESMEN**

<u>Customer_id</u>	<u>Employee_id</u>
--------------------	--------------------



Foreign key: Customer\_id references Personal\_id { **PERSON1A** }  
Employee\_id references Personal\_id { **PERSON1A** }

#### **TRANSACTIONS1**

<u>Product_id</u>	Sales_id
-------------------	----------

Foreign key: Product\_id references Product\_id { **PRODUCT** }

#### **TRANSACTIONS2**

<u>Sales_id</u>	Customer_id	Salesman_id	Site_id	Sale_time
-----------------	-------------	-------------	---------	-----------

Foreign key: Site\_id references Site\_id { **MARKETING\_SITE** }  
Salesman\_id references Personal\_id { **PERSON1A** }  
Customer\_id references Personal\_id { **PERSON1A** }  
Sales\_id references Sales\_id { **TRANSACTION1** }

**All SQL Statements:**

**CREATE TABLE PERSON1A**

(Personal_id	INT	<b>NOT NULL,</b>
Fname	VARCHAR(15)	<b>NOT NULL,</b>
Lname	VARCHAR(25)	<b>NOT NULL,</b>
Age	INT,	
Gender	CHAR,	
Addr_line1	VARCHAR(35),	
Addr_line2	VARCHAR(35),	
City	VARCHAR(35),	
Zipcode	INT	<b>NOT NULL,</b>
E_addr	VARCHAR(40),	
<b>PRIMARY KEY (Personal_id) );</b>		

**CREATE TABLE PERSON1B**

(Zipcode	INT	<b>NOT NULL,</b>
State	VARCHAR(15),	
<b>PRIMARY KEY (Zipcode),</b>		
<b>FOREIGN KEY (Zipcode) REFERENCES PERSON1A(Zipcode)</b>		
<b>ON DELETE CASCADE</b>	<b>ON UPDATE CASCADE);</b>	

**CREATE TABLE EMPLOYEE**

(Employee_id	INT	<b>NOT NULL,</b>
Rank	VARCHAR(15),	
Title	VARCHAR(25),	
Super_id	INT	<b>DEFAULT '888665555',</b>

**PRIMARY KEY** (Employee\_id) ,

**FOREIGN KEY** (Employee\_id) **REFERENCES** PERSON1A(personal\_id)

**ON DELETE** CASCADE

**ON UPDATE** CASCADE,

**FOREIGN KEY** (Super\_id) **REFERENCES** PERSON1A(personal\_id)

**ON DELETE** SET DEFAULT

**ON UPDATE** CASCADE);

**CREATE TABLE** CUSTOMER

(Customer\_id INT **NOTNULL**,

Rewards\_points INT **DEFAULT 0**,

**PRIMARY KEY** (Customer\_id),

**FOREIGN KEY** (Customer\_id) **REFERENCES** PERSON1A(Personal\_id)

**ON DELETE** CASCADE

**ON UPDATE** CASCADE);

**CREATE TABLE** PHONE\_NUMBER

(Personal\_id INT **NOTNULL**,

Phone\_num INT,

**PRIMARY KEY** (Personal\_id) );

**CREATE TABLE** POTENTIAL\_EMPLOYEE

(Personal\_id INT **NOTNULL**,

Work\_experience INT,

**PRIMARY KEY** (Personal\_id),

**FOREIGN KEY** (Personal\_id) **REFERENCES** PERSON1A(Personal\_id)

**ON DELETE** CASCADE

**ON UPDATE** CASCADE);

**CREATE TABLE** VENDOR1

(Vendor\_id                    INT                    **NOT NULL,**  
 Account\_num                INT                    **NOT NULL,**  
**PRIMARY KEY (Vendor\_id),**  
**FOREIGN KEY (Vendor\_id) REFERENCES PERSON1A(Personal\_id)**  
**ON DELETE CASCADE                    ON UPDATE CASCADE);**

#### **CREATE TABLE VENDOR2**

(Account\_num                    INT                    **NOT NULL,**  
 Credit\_rating                    INT                    **NOT NULL,**  
 Purchasing\_web\_service\_url    VARCHAR(60)        **NOT NULL,**  
**PRIMARY KEY (Account\_num),**  
**FOREIGN KEY (Account\_num) REFERENCES VENDOR1(Account\_num)**  
**ON DELETE CASCADE                    ON UPDATE CASCADE);**

#### **CREATE TABLE DEPARTMENT**

(Dept\_id                    INT                    **NOTNULL,**  
 Dept\_name                    VARCHAR(20)        **NOTNULL,**  
**PRIMARY KEY (Dept\_id) );**

#### **CREATE TABLE JOB\_POSITION**

(Job\_id                    INT                    **NOT NULL,**  
 Dept\_id                    INT                    **NOT NULL,**  
 Job\_desc                    VARCHAR(15)        **NOT NULL,**  
 Posted\_date                DATE,  
**PRIMARY KEY (Job\_id),**  
**FOREIGN KEY (Dept\_id) REFERENCES DEPARTMENT(Dept\_id)**  
**ON DELETE CASCADE                    ON UPDATE CASCADE);**



**CREATE TABLE MARKETING\_SITE**

(Site_id	INT	<b>NOT NULL,</b>
Site_location	VARCHAR(40)	<b>NOT NULL,</b>
Site_name	VARCHAR(20)	<b>NOT NULL,</b>
<b>PRIMARY KEY (Site_id) );</b>		

**CREATE TABLE PRODUCT**

(Product_id	INT	<b>NOT NULL,</b>
Product_type	VARCHAR(20)	<b>NOT NULL,</b>
List_price	INT	<b>NOT NULL,</b>
Style	VARCHAR(20),	
Weight	DECIMAL(3,1),	
Size	DECIMAL(3,1),	
<b>PRIMARY KEY (Product_id) );</b>		

**CREATE TABLE PARTS1**

(Part_type	VARCHAR(20)	<b>NOT NULL,</b>
Vendor_id	INT	<b>NOT NULL,</b>
Quantity_supplied	VARCHAR(60)	<b>NOT NULL,</b>
Price	INT	<b>NOT NULL,</b>
Part_weight	INT,	
<b>PRIMARY KEY (Part_type, Vendor_id),</b>		
<b>FOREIGN KEY (Vendor_id) REFERENCES VENDOR1(Vendor_id)</b>		
<b>ON DELETE CASCADE</b>		<b>ON UPDATE CASCADE);</b>

## CREATE TABLE PARTS2

```
(Part_type          VARCHAR(20)      NOT NULL,
Product_id         INT              NOT NULL,
Num_for_product    INT              NOT NULL,
PRIMARY KEY (Part_type, Product_id),
FOREIGN KEY (Product_id) REFERENCES PRODUCT(Product_id),
ON DELETE CASCADE ON UPDATE CASCADE
FOREIGN KEY (Part_type) REFERENCES PARTS1(Part_type)
ON DELETE CASCADE ON UPDATE CASCADE);
```

## CREATE TABLE INTERVIEW

```

(Interview_round          INT          NOT NULL,
 Interviewer_id           INT          NOT NULL,
 Interviewee_id           INT          NOT NULL,
 Job_id                   INT          NOT NULL,
 Grade                    CHAR         NOT NULL,
 Interview_time            TIMESTAMP,
PRIMARY KEY (interviewer_id, interviewee_id, interview_round, job_id),
FOREIGN KEY (Interviewer_id) REFERENCES PERSON1A(Personal_id)
ON DELETE CASCADE ON UPDATE CASCADE,
FOREIGN KEY (Interviewee_id) REFERENCES PERSON1A(Personal_id)
ON DELETE CASCADE ON UPDATE CASCADE,
FOREIGN KEY (Job_id) REFERENCES JOB_Position(Job_id)
ON DELETE CASCADE ON UPDATE CASCADE);

```

```
CREATE TABLE Monthly SAL
```

(Employee_id	INT	<b>NOT NULL,</b>
Transaction_num	INT	<b>NOT NULL,</b>
Pay_date	DATE	<b>NOT NULL,</b>
Amount	FLOAT	<b>NOT NULL,</b>

**PRIMARY KEY (Transaction\_num, Employee\_id),**

**FOREIGN KEY (Employee\_id) REFERENCES PERSON1A(Personal\_id)**

**ON DELETE CASCADE ON UPDATE CASCADE);**

**CREATE TABLE WORKS\_FOR**

(Employee_id	INT	<b>NOT NULL,</b>
Dept_id	INT	<b>NOT NULL,</b>
Start_time	TIMESTAMP	<b>NOT NULL,</b>
End_time	TIMESTAMP	<b>NOT NULL,</b>

**PRIMARY KEY (Dept\_id, Employee\_id),**

**FOREIGN KEY (Dept\_id) REFERENCES DEPARTMENT(Dept\_id)**

**ON DELETE CASCADE ON UPDATE CASCADE,**

**FOREIGN KEY (Employee\_id) REFERENCES PERSON1A(Personal\_id)**

**ON DELETE CASCADE ON UPDATE CASCADE);**

**CREATE TABLE WORKS\_ON**

(Employee_id	INT	<b>NOT NULL,</b>
Site_id	INT	<b>NOT NULL,</b>
Date	DATE	<b>NOT NULL,</b>

**PRIMARY KEY (Site\_id, Employee\_id),**

**FOREIGN KEY (Site\_id) REFERENCES MARKETING\_SITE(Dept\_id)**

**ON DELETE CASCADE ON UPDATE CASCADE,**

**FOREIGN KEY (Employee\_id) REFERENCES PERSON1A(Site\_id)**

```

                ON DELETE CASCADE                ON UPDATE CASCADE);

CREATE TABLE APPLY_FOR
    (Applicant_id            INT            NOT NULL,
     Job_id                  INT            NOT NULL,
     Apply_date              DATE           NOT NULL,
     PRIMARY KEY (Applicant_id, Job_id),
     FOREIGN KEY (Job_id) REFERENCES JOB_POSITION(Job_id)
                ON DELETE CASCADE                ON UPDATE CASCADE,
     FOREIGN KEY (Applicant_id) REFERENCES PERSON1A(Personal_id)
                ON DELETE CASCADE                ON UPDATE CASCADE);

CREATE TABLE PREFERRED_SALESMAN
    (Employee_id            INT            NOT NULL,
     Customer_id            INT            NOT NULL,
     PRIMARY KEY (Customer_id, Employee_id),
     FOREIGN KEY (Customer_id) REFERENCES PERSON1A(Personal_id)
                ON DELETE CASCADE                ON UPDATE CASCADE,
     FOREIGN KEY (Employee_id) REFERENCES PERSON1A(Personal_id)
                ON DELETE CASCADE                ON UPDATE CASCADE);

CREATE TABLE TRANSACTIONS1
    (Product_id            INT            NOT NULL,
     Sales_id              INT            NOT NULL,
     PRIMARY KEY (Product_id),
     FOREIGN KEY (Product_id) REFERENCES PRODUCT(Product_id)
                ON DELETE CASCADE                ON UPDATE CASCADE);

CREATE TABLE TRANSACTIONS2

```



(Sales\_id INT NOT NULL,  
 Customer\_id INT,  
 Salesman\_id INT,  
 Site\_id INT,  
 Sale\_time DATE,  
**PRIMARY KEY** (Sales\_id),  
**FOREIGN KEY** (Site\_id) **REFERENCES** MARKETING\_SITE(Site\_id)  
**ON DELETE** SET NULL **ON UPDATE** CASCADE,  
**FOREIGN KEY** (Sales\_id) **REFERENCES** TRANSACTION1(Sales\_id)  
**ON DELETE** CASCADE **ON UPDATE** CASCADE,  
**FOREIGN KEY** (Customer\_id) **REFERENCES** PERSON1A(Personal\_id)  
**ON DELETE** SET NULL **ON UPDATE** CASCADE,  
**FOREIGN KEY** (Salesman\_id) **REFERENCES** PERSON1A(Personal\_id)  
**ON DELETE** SET NULL **ON UPDATE** CASCADE);

## **VIEW STATEMENTS**

### **VIEW 1:**

<b>CREATE VIEW</b>	AVG_EMP_SALARY(Employee_id, avg_salary)
<b>AS SELECT</b>	M.Employee_id, AVG(M.Amount)
<b>FROM</b>	EMPLOYEE E, MONTHLY_SAL M
<b>WHERE</b>	E.Employee_id = M.Employee_id
<b>GROUP BY</b>	M.Employee_id;

### VIEW 2:

<b>CREATE VIEW</b>	ROUNDS_PASSED(Interviewee_id, Job_id,
passed_rounds)	
<b>AS SELECT</b>	I.Interviewee_id, I.Job_id, COUNT(*)
<b>FROM</b>	INTREVIEW I, JOB_POSITION J, PERSON1A P
<b>WHERE</b>	I.Job_id = J.Job_id AND I.interviewee_id=P.personal_id <b>AND</b> I.Grade > 60
<b>GROUP BY</b>	I.Interviewee_id, I.Job_id;

### VIEW 3:

<b>CREATE VIEW</b>	Item_types_sold(Product_type, Number_sold)
<b>AS SELECT</b>	<b>DISTINCT</b> P.Product_type, Count(*)
<b>FROM</b>	PRODUCT P
<b>WHERE</b>	P.Product_id <b>IN</b> <b>(SELECT</b> Product_id <b>FROM</b> TRANSACTIONS1 <b>WHERE</b> Product_id = P.Product_id)
<b>GROUP BY</b>	P.Product_type;

#### **VIEW 4:**

```
CREATE VIEW          PART_COST_FOR_PRODUCT(Product_id,  
Total_parts_cost)  
  
  AS SELECT          Product_id, SUM(Price)  
  
    FROM              (PARTS2 NATURAL LEFT OUTER JOIN PARTS1)  
  
    GROUP BY          Product_id;
```

#### **QUERIES**

(1)

```
SELECT      DISTINCT Interviewer_id, Fname, Lname  
  
FROM        INTERVIEW LEFT OUTER JOIN PERSON1A ON Interviewer_id = Personal_id  
  
WHERE        Interviewer_id IN (SELECT          DISTINCT Interviewer_id  
  
              FROM          INTERVIEW LEFT OUTER JOIN PERSON1A  
ON              Interviewee_id = Personal_id  
  
              WHERE        Fname = 'Hellen' AND Lname = 'Cole' AND  
                          Job_id = 11111);
```

(2)

```
SELECT      Job_id  
  
FROM        JOB_POSITION NATURAL LEFT OUTER JOIN DEPARTMENT  
  
WHERE        WHERE Dept_name = "Marketing"  
  
              AND DATEPART(mm, Posted_date) = 01  
  
              AND DATEPART(yy, Posted_date) = 2011;
```

(3)

```
SELECT      Personal_id, Fname, Lname
```

```

FROM PERSON1A
WHERE Personal_id NOT IN (SELECT DISTINCT Personal_id
FROM PERSON1A RIGHT OUTER JOIN EMPLOYEE ON
Personal_id = Super_id);

```

(4)

```

SELECT DISTINCT Site_id, Site_location
FROM TRANSACTIONS2 LEFT OUTER JOIN MARKETING_SITE
WHERE AND DATEPART(mm, Sale_time) <> 03
AND DATEPART(yy, Sale_time) <> 2011;

```

(5)

Assumed that employees have not been hired one month after it is posted and in the same year.

```

WITH EMPLOYEES_HIRED(Job_id, , Interview_round, Interview_time, Posted_date,
Job_desc) AS
(SELECT Job_id, Interview_round, Interview_time, Posted_date, Job_desc
FROM ((INTERVIEW LEFT OUTER JOIN PERSON1A ON Interviewee_id = Personal_id)
NATURAL LEFT OUTER JOIN JOB_POSITION)
WHERE Interviewee_id, Job_id IN
(SELECT Interviewee_id, Job_id
FROM ROUNDS_PASSED
WHERE passed_rounds >= 5)
AND
Interviewee_id, Job_id IN
(SELECT Interviewee_id, Job_id, AVG(Grade)
FROM INTERVIEW

```



```

        GROUP BY Interviewee_id, Job_id
        HAVING AVG(Grade) > 70))

SELECT DISTINCT Job_id, Job_desc
FROM EMPLOYEES_HIRED
WHERE Interview_round > 5 AND
      DATEPART(dd, Posted_date) <= DATEPART(dd, Interview_time)
      AND (DATEPART(mm, Posted_date) - DATEPART(mm, Interview_time) ) < 1
      AND DATEPART(yy, Posted_date) = DATEPART(dd, Interview_time)

UNION

SELECT DISTINCT Job_id, Job_desc
FROM EMPLOYEES_HIRED
WHERE Interview_round > 5 AND
      DATEPART(dd, Posted_date) <= DATEPART(dd, Interview_time)
      AND (DATEPART(mm, Posted_date) - DATEPART(mm, Interview_time) ) = 1
      AND DATEPART(yy, Posted_date) = DATEPART(dd, Interview_time)

```

(6)

```

SELECT DISTINCT Salesman_id, Fname, Lname
FROM (((TRANSACTIONS1 NATURAL LEFT OUTER JOIN PRODUCT)
      NATURAL JOIN TRANSACTIONS2)
      LEFT OUTER JOIN PERSON1A ON Salesman_id = Personal_id)
WHERE List_Price > 200;

```

(7)

```
SELECT      Dept_id, Dept_name
FROM        (JOB_POSITION NATURAL JOIN DEPARTMENT)
WHERE       DATEPART(dd, Posted_date) <> 01
           AND DATEPART(mm, Posted_date) <> 02
           AND DATEPART(mm, Posted_date) <> 01
           AND DATEPART(yy, Posted_date) <> 2011;
```

(8)

```
SELECT      Employee_id, Fname, Lname, Dept_id
FROM        ((PERSON1A RIGHT OUTER JOIN WORKS_FOR ON Personal_id = Employee_id)
           JOIN APPLY_FOR ON Employee_id = Applicant_id)
WHERE       Job_id = 12345;
```

(9)

```
WITH        PRODUCT_SELL_COUNT(Product_type, Sell_count) AS
           (SELECT      DISTINCT Product_type, COUNT(*)
            FROM        (TRANSACTIONS1 NATURAL LEFT OUTER JOIN PRODUCT)
            GROUP BY    Product_type)

SELECT      Product_type
FROM        PRODUCT_SELL_COUNT
WHERE       Sell_count IN (SELECT      MAX(Sell_count)
                           FROM        PRODUCT_SELL_COUNT);
```

(10)

Using the previous view table PART\_COST\_FOR\_PRODUCT

```
WITH PRODUCT_PROFIT(Product_type, Profit) AS
    (SELECT DISTINCT Product_type, SUM(List_price –
Total_parts_cost)
    FROM ((TRANSACTIONS1 NATURAL LEFT OUTER JOIN PRODUCT)
    NATURAL LEFT OUTER JOIN PART_COST_FOR_PRODUCT)
    GROUP BY Product_type)
SELECT Product_type
FROM PRODUCT_PROFIT
WHERE Profit IN (SELECT MAX(Profit) AS Profit
    FROM PRODUCT_PROFIT);
```

(11)

```
SELECT DISTINCT Employee_id, Fname, Lname, COUNT(*)
FROM (WORKS_FOR LEFT OUTER JOIN PERSON1A ON Employee_id = Personal_id)
WHERE COUNT(*) IN (SELECT COUNT(*)
    FROM DEPARTMENT)
GROUP BY Employee_id;
```

(12)

Using view table ROUNDS\_PASSED(Interviewee\_id, Job\_id, passed\_rounds)

```
SELECT Fname, Lname, E_addr, Job_id
FROM (INTERVIEW LEFT OUTER JOIN PERSON1A ON Interviewee_id = Personal_id)
WHERE Interviewee_id, Job_id IN
    (SELECT Interviewee_id, Job_id
    FROM ROUNDS_PASSED
```

```

WHERE      passed_rounds >= 5)
AND
Interviewee_id, Job_id IN
(SELECT      Interviewee_id, Job_id, AVG(Grade)
FROM        INTERVIEW
GROUP BY    Interviewee_id, Job_id
HAVING      AVG(Grade) > 70);

```

(13)

Assumed that the query finds name, phone num and email address of job application candidates selected as interviewees for interview

```

WITH      CANDIDATE INFO(Personal_id, Fname, Lname, Phone number,
E_addr)
SELECT      Personal_id, Fname, Lname, Phone number, E_addr
FROM        (PHONE_NUMBER NATURAL LEFT OUTER JOIN PERSON1A),
INTERVIEW
WHERE      Personal_id = Interviewee_id

```

(14)

Using the view table AVG\_EMP\_SALARY(Employee\_id, avg\_salary)

```

SELECT      Fname, Lname, Employee_id
FROM        (AVG_EMP_SALARY LEFT OUTER JOIN PERSON1A ON Employee_id =
Personal_id)
WHERE      avg_salary IN (SELECT      MAX(avg_salary)

```

**FROM** AVG\_EMP\_SALARY);

(15)


- Part name is assumed to be the part tye.

```
SELECT      Vendor_id, Fname, Lname
FROM        (PARTS1 LEFT OUTER JOIN PERSON1A ON Vendor_id = Personal_id)
WHERE        Part_weight < 4 AND Part_type = 'Cup' AND
              Price IN (SELECT      MIN(Price)
                        FROM        PARTS1
                        WHERE        Part_weight < 4 AND Part_type = 'Cup');
```

## DEPENDENCY DIAGRAM:


### PERSON1A

Personal_id	Fname	Lname	Age	Gender	Addr_line1	Addr_line2	City	Zipcode	E-addr
-------------	-------	-------	-----	--------	------------	------------	------	---------	--------




### PERSON1B

Zipcode	State
---------	-------




### EMPLOYEE

Employee_id	Rank	Title	Super_id
-------------	------	-------	----------




### CUSTOMER

Customer_id	Rewards_points
-------------	----------------




### PHONE\_NUMBER

Personal_id	Phone_num
-------------	-----------




### POTENTIAL\_EMPLOYEE

Personal_id	Work_experience
-------------	-----------------




### VENDOR1

Vendor_id	Account_num
-----------	-------------




### VENDOR2

Account_num	Credit_rating	Purchasing_web_service_url
-------------	---------------	----------------------------




### DEPARTMENT

Dept_id	Dept_name
---------	-----------




### JOB\_POSITION

Job_id	Dept_id	Job_desc	Posted_date
--------	---------	----------	-------------



**MARKETING\_SITE**


<u>Site_id</u>	Site_location	Site_name
----------------	---------------	-----------

**PRODUCT**


<u>Product_id</u>	Product_type	List_price	Style	Weight	Size
-------------------	--------------	------------	-------	--------	------

**PARTS1**


<u>Part_type</u>	<u>Vendor_id</u>	Quantity_supplied	Price	<u>Part_weight</u>
------------------	------------------	-------------------	-------	--------------------

**PARTS2**


<u>Part_type</u>	<u>Product_id</u>	<u>Num_for_product</u>
------------------	-------------------	------------------------

**INTERVIEW**


<u>Interview_round</u>	<u>Interviewer_id</u>	<u>Interviewee_id</u>	<u>Job_id</u>	Grade	<u>Interview_time</u>
------------------------	-----------------------	-----------------------	---------------	-------	-----------------------

**MONTHLY\_SAL**


<u>Employee_id</u>	<u>Transaction_num</u>	<u>Pay_date</u>	<u>Amount</u>
--------------------	------------------------	-----------------	---------------

**WORKS\_FOR**


<u>Employee_id</u>	<u>Dept_id</u>	<u>Start_time</u>	<u>End_time</u>
--------------------	----------------	-------------------	-----------------

**WORKS\_ON**

<u>Site_id</u>	<u>Employee_id</u>	<u>Date</u>
----------------	--------------------	-------------

**APPLY\_FOR**


<u>Applicant_id</u>	<u>Job_id</u>	<u>Apply_date</u>
---------------------	---------------	-------------------

**PREFERRED\_SALESMEN**

<u>Customer_id</u>	<u>Employee_id</u>
--------------------	--------------------

**TRANSACTIONS1**

<u>Product_id</u>	<u>Sales_id</u>
-------------------	-----------------





**TRANSACTIONS2**

<u>Sales_id</u>	Customer_id	Salesman_id	Site_id	Sale_time
-----------------	-------------	-------------	---------	-----------

