CS 4347: DATABASE SYSTEMS

PROJECT

GROUP MEMBERS:

Al Wasee Mahmood (AXM190229)

Dail Kang (djk180004)

Muhammad Zubair(MAZ180006)

SECTION: CS 4347.503

PROFESSOR: Dr. Nidhiben Solanki

DATE: 12/01/2021

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:

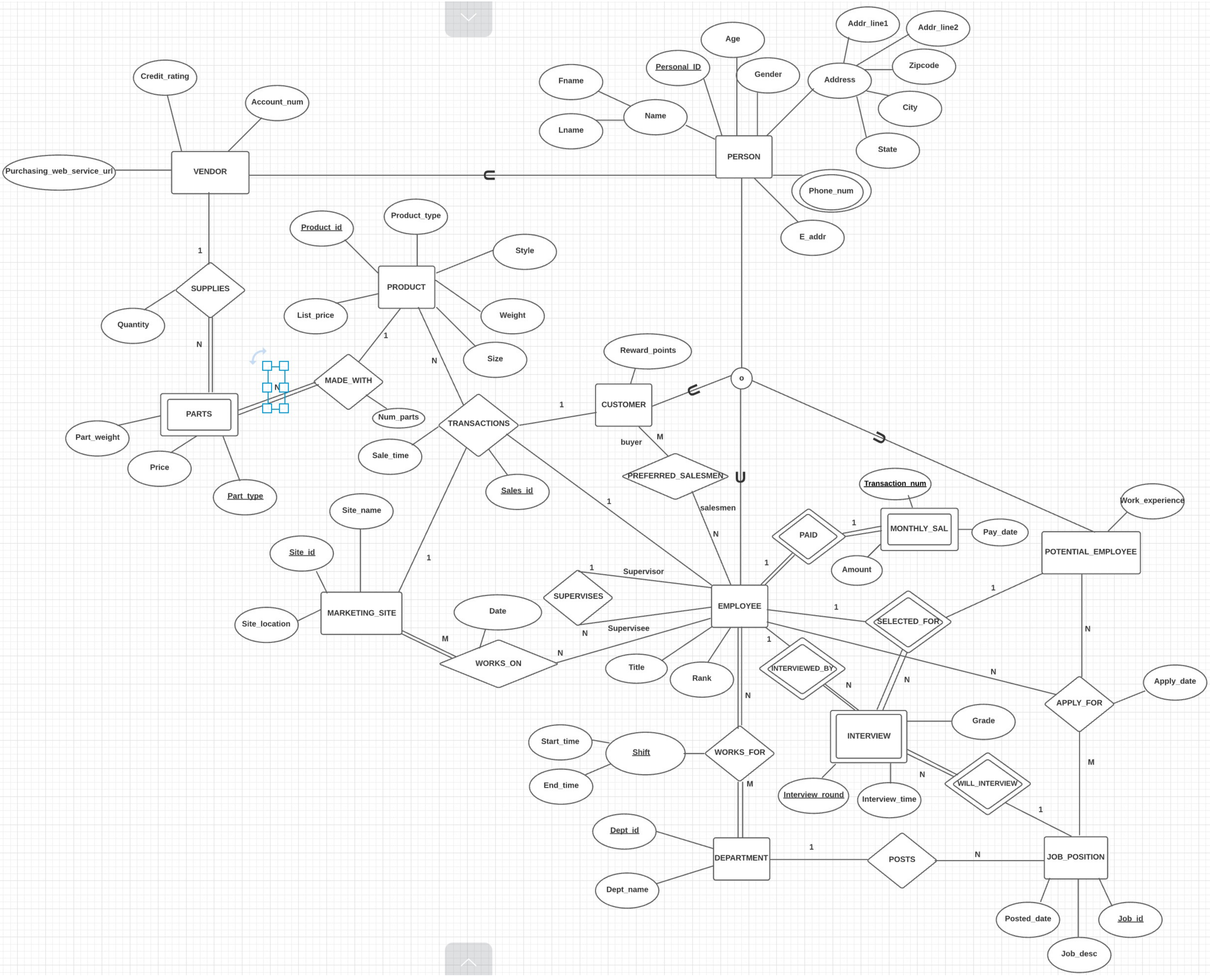
ASSUMPTIONS:

- 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


Credit_rating
Account_num
Purchasing_web_service_url
VENDOR
1
SUPPLIES
Quantity
N
PARTS
Part_weight
Price
Part_type
N
MADE_WITH
Num_parts
1
Product_id
Product_type
Style
PRODUCT
List_price
Weight
Size
N
TRANSACTIONS
Sale_time
Sales_id
1
Site_name
Site_id
Site_location
MARKETING_SITE
1
M
Date
WORKS_ON
N
Fname
Lname
Name
Age
Personal_ID
Gender
Addr_line1
Addr_line2
Zipcode
Address
City
State
PERSON
Phone_num
E_addr
Reward_points
CUSTOMER
1
buyer
M
PREFERRED_SALESMEN
salesmen
N
1
Supervisor
SUPERVISES
N
Supervisee
Title
Rank
EMPLOYEE
N
Start_time
Shift
End_time
WORKS_FOR
M
Dept_id
Dept_name
DEPARTMENT
1
POSTS
N
JOB_POSITION
Posted_date
Job_id
Job_desc
Transaction_num
PAID
1
MONTHLY_SAL
Pay_date
Amount
1
SELECTED_FOR
1
INTERVIEWED_BY
N
N
INTERVIEW
Grade
Interview_round
Interview_time
N
WILL_INTERVIEW
1
Work_experience
POTENTIAL_EMPLOYEE
1
N
APPLY_FOR
Apply_date
M
o

DATABASE SCHEMA BEFORW NORMALIZATION:

PERSON

Personal_id	Fname	Lname	Age	Gender	Addr_line1	Addr_line2	Zipcode	City	State	E_addr

EMPLOYEE

Employee_id	Rank	Title	Super_id

Foreign key: Employee_id references Personal_id {**PERSON}**

Super_id references Personal_id {**PERSON}**

CUSTOMER

Customer_id	Rewards_points

Foreign key: Customer_id references Personal_id {**PERSON}**

PHONE_NUMBER

Personal_id	Phone_num

Foreign key: Personal_id references Personal_id {**PERSON}**

POTENTIAL_EMPLOYEE

Personal_id	Work_experience

Foreign key: Personal_id references Personal_id {**PERSON}**

VENDOR

Vendor_id	Purchasing_web_service_url	Credit_rating	Account_num

Foreign key: Vendor_id references Personal_id {**PERSON}**

DEPARTMENT

Dept_id	Dept_name

JOB_POSITION

Job_id	Dept_id	Job_desc	Posted_date

Foreign key: Dept_id references Dept_id **{DEPARTMENT}**

MARKETING_SITE

Site_id	Site_location	Site_name

PRODUCT

Product_id	Product_type	List_price	Style	Weight	Size

PARTS

Part_type	Vendor_id	Quantity_supplied	Product_id	Num_for_product	Price	Part_weight

Foreign key: Vendor_id references Personal_id (**PERSON)**

Product_id references Product_id (**PRODUCT)**

INTERVIEW

Interview_round	Interviewer_id	Interviewee_id	Job_id	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

Employee_id	Transaction_num	Pay_date	Amount

Foreign key: Employee_id references Personal_id {**PERSON}**

WORKS_FOR

Dept_id	Employee_id	Start_time	End_time

Foreign key: Dept_id references Dept_id **{DEPARTMENT}**

Employee_id references Personal_id(**PERSON)**

WORKS_ON

Site_id (FK)	Employee_id	Date

Foreign key: Site_id references Site_id **{MARKETING_SITE}**

Employee_id references Personal_id {**PERSON}**

APPLY_FOR

Applicant_id	Job_id	Apply_date

Foreign key: Applicant_id references Personal _id (**PERSON)**

Job_id references Job_id (**JOB_POSITION**)

TRANSACTIONS

Sales_id	Product_id	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

Customer_id	Employee_id (FK)

Foreign key: Customer_id references Personal_id {**PERSON}**

Employee_id references Personal_id {**PERSON}**

Relation schema after normalization:

PERSON1A

Personal_id	Fname	Lname	Age	Gender	Addr_line1	Addr_line2	City	Zipcode	E_addr

PERSON1B

Zipcode	State

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

EMPLOYEE

Employee_id	Rank	Title	Super_id

Foreign key: Employee_id references Personal_id (**PERSON1A)**
Super_id references Personal_id (**PERSON1A)**

CUSTOMER

Customer_id	Rewards_points

Foreign key: Customer_id references Personal_id { **PERSON1A)**

PHONE_NUMBER

Personal_id	Phone_num

POTENTIAL_EMPLOYEE

Personal_id	Work_experience

Foreign key: Personal_id references Personal_id { **PERSON1A**)

VENDOR1

Vendor_id	Account_num

Foreign key: Vendor_id references Personal_id { **PERSON1A)**

VENDOR2

Account_num	Credit_rating	Purchasing_web_service_url

Foreign key: Account_num references Account_num (**VENDOR1)**

DEPARTMENT

Dept_id	Dept_name

JOB_POSITION

Job_id	Dept_id	Job_desc	Posted_date

Foreign key: Dept_id references Dept_id **{DEPARTMENT}**

MARKETING_SITE

Site_id	Site_location	Site_name

PRODUCT

Product_id	Product_type	List_price	Style	Weight	Size

PARTS1

Part_type	Vendor_id	Quantity_supplied	Price	Part_weight

Foreign key: Vendor_id references Personal_id (**PERSON1A)**

PARTS2

Part_type	Product_id	Num_for_product

Foreign key: Part_type references Part_type (**PARTS1)**
Product_id references Product_id (**PRODUCT)**

INTERVIEW

Interview_round	Interviewer_id	Interviewee_id	Job_id	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

Employee_id	Transaction_num	Pay_date	Amount

Foreign key: Employee_id references Personal_id {**PERSON1A}**

WORKS_FOR

Employee_id	Dept_id	Start_time	End_time

Foreign key: Dept_id references Dept_id **{DEPARTMENT}**
Employee_id references Personal_id(**PERSON1A)**

WORKS_ON

Site_id	Employee_id	Date

Foreign key: Site_id references Site_id **{MARKETING_SITE}**
Employee_id references Personal_id {**PERSON1A}**

APPLY_FOR

Applicant_id	Job_id	Apply_date

Foreign key: Applicant_id references Personal _id (**PERSON1A)**
Job_id references Job_id (**JOB_POSITION**)

PREFERRED_SALESMEN

Customer_id	Employee_id

Foreign key: Customer_id references Personal_id { **PERSON1A}**
Employee_id references Personal_id {**PERSON1A}**

TRANSACTIONS1

Product_id	Sales_id

Foreign key: Product_id references Product_id {**PRODUCT}**

TRANSACTIONS2

Sales_id	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 **NOT NULL,**

Fname VARCHAR(15) **NOT NULL,**

Lname VARCHAR(25) **NOT NULL,**

Age INT,

Gender CHAR,

Adrr_line1 VARCHAR(35),

Adrr_line2 VARCHAR(35),

City VARCHAR(35),

Zipcode INT **NOT NULL,**

E_addr VARCHAR(40),

PRIMARY KEY (Personal_id));

CREATE TABLE PERSON1B

(Zipcode INT **NOT NULL,**

State VARCHAR(15),

PRIMARY KEY (Zipcode**),**

FOREIGN KEY (Zipcode) **REFERENCES** PERSON1A(Zipcode)

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

CREATE TABLE EMPLOYEE

(Employee_id INT **NOT NULL,**

Rank VARCHAR(15),

Title VARCHAR(25),

Super_id INT **DEFAULT '**888665555'**,**

```
    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             NOT NULL,
         Site_location          VARCHAR(40)     NOT NULL,
         Site_name              VARCHAR(20)     NOT NULL,
         PRIMARY KEY (Site_id) );

CREATE TABLE PRODUCT
        (Product_id             INT             NOT NULL,
         Product_type           VARCHAR(20)     NOT NULL,
         List_price             INT             NOT NULL,
         Style                  VARCHAR(20),
         Weight                 DECIMAL(3,1),
         Size                   DECIMAL(3,1),
         PRIMARY KEY (Product_id) );

CREATE TABLE PARTS1
        (Part_type              VARCHAR(20)     NOT NULL,
         Vendor_id              INT             NOT NULL,
        Quantity_supplied       VARCHAR(60)     NOT NULL,
         Price                  INT             NOT NULL,
        Part_weight             INT,
         PRIMARY KEY (Part_type, Vendor_id),
         FOREIGN KEY (Vendor_id) REFERENCES VENDOR1(Vendor_id)
                        ON DELETE CASCADE       ON UPDATE CASCADE);
```

```
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 (interviwer_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          NOT NULL,
 Transaction_num      INT          NOT NULL,
 Pay_date             DATE         NOT NULL,
 Amount               FLOAT        NOT NULL,
 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          NOT NULL,
 Dept_id              INT          NOT NULL,
 Start_time           TIMESTAMP    NOT NULL,
 End_time             TIMESTAMP    NOT NULL,
 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          NOT NULL,
 Site_id              INT          NOT NULL,
 Date                 DATE         NOT NULL,
 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:

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

VIEW 2:

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

VIEW 3:

```
CREATE VIEW     Item_types_sold(Product_type, Number_sold)
  AS SELECT     DISTINCT P.Product_type, Count(*)
     FROM       PRODUCT P
     WHERE      P.Product_id IN
                (SELECT     Product_id
                 FROM       TRANSACTIONS1
                 WHERE      Product_id = P.Product_id)
     GROUP BY   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 RGHT 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)

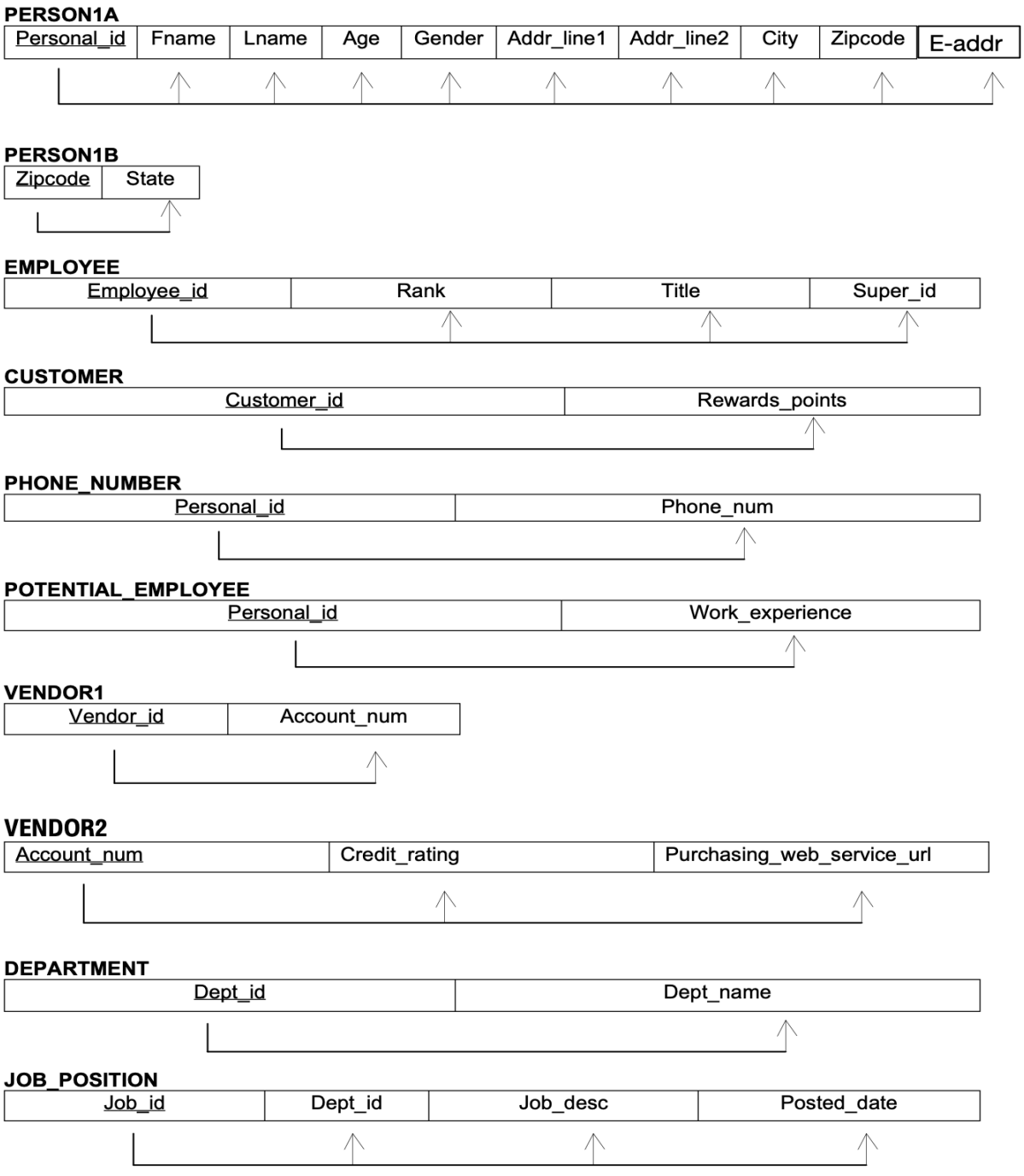
FROM AVG_EMP_SALARY);

(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:



MARKETING_SITE

Site_id	Site_location	Site_name

PRODUCT

Product_id	Product_type	List_price	Style	Weight	Size

PARTS1

Part_type	Vendor_id	Quantity_supplied	Price	Part_weight

PARTS2

Part_type	Product_id	Num_for_product

INTERVIEW

Interview_round	Interviewer_id	Interviewee_id	Job_id	Grade	Interview_time

MONTHLY_SAL

Employee_id	Transaction_num	Pay_date	Amount

WORKS_FOR

Employee_id	Dept_id	Start_time	End_time

WORKS_ON

Site_id	Employee_id	Date

APPLY_FOR

Applicant_id	Job_id	Apply_date

PREFERRED_SALESMEN

Customer_id	Employee_id

TRANSACTIONS1

Product_id	Sales_id

TRANSACTIONS2

Sales_id	Customer_id	Salesman_id	Site_id	Sale_time