Term Project

AML140830

Alex Lundin

# 1. Written answers to the questions in the Project Questions area.

**5 more rules**

1. Employee
   1. Primary key
      1. Employee\_ID
2. Potential Employee
   1. Primary key
      1. Potential\_Employee\_ID
3. Customer
   1. Primary key
      1. Customer\_ID
4. Catalog
   1. Combination of Part\_ID and vendor\_ID
5. Product
   1. Product\_ID
6. Shift
   1. Shift\_ID

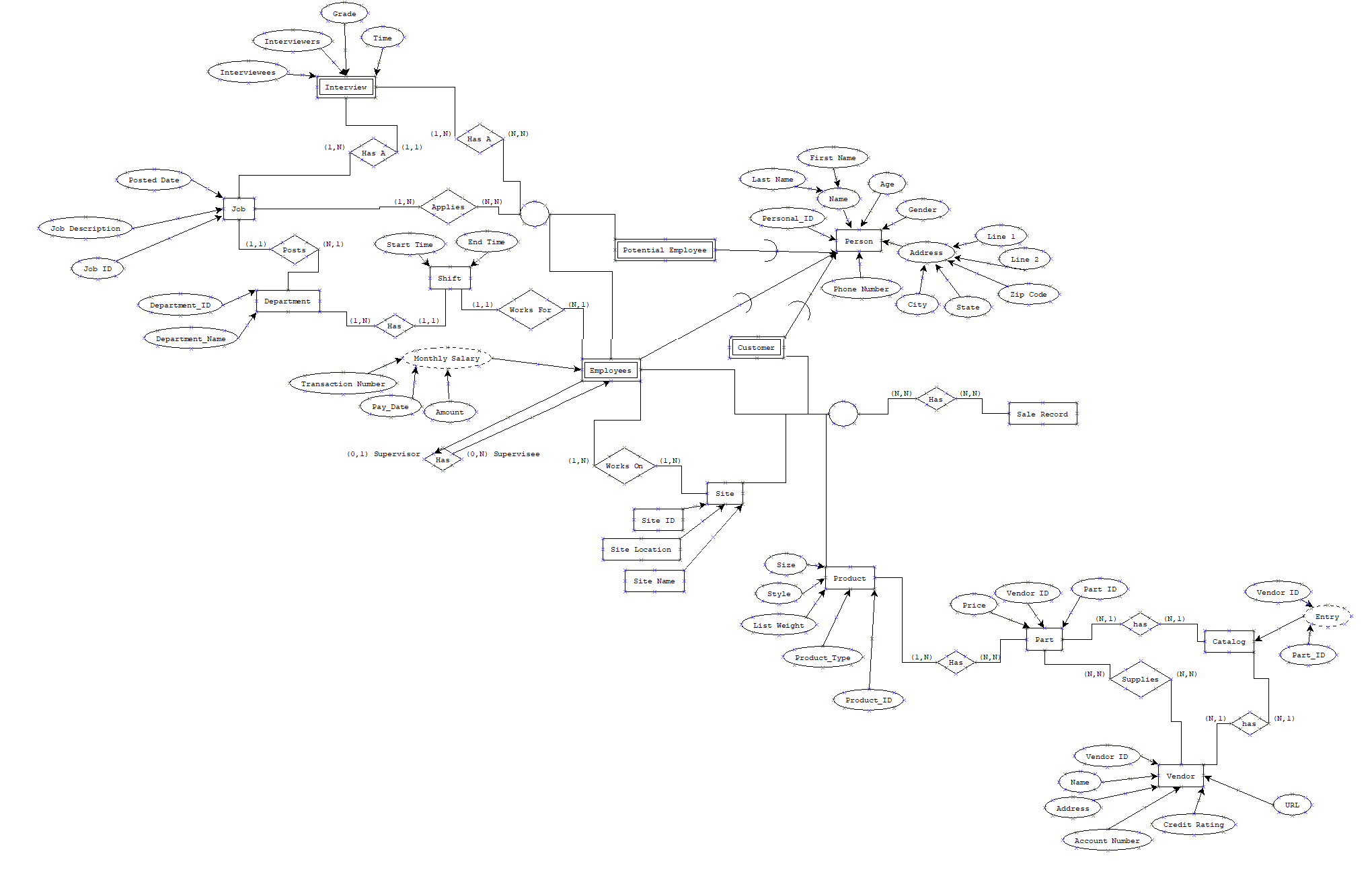
**Is the ability to model super class and sub class likely to be important in this type of environment? Why or why not**

Yes, because there are several types of person object in the database, employee, potential employee and customer. Each person type shares the 12 base attributes of the person object and then each type adds on their own unique attributes from there.

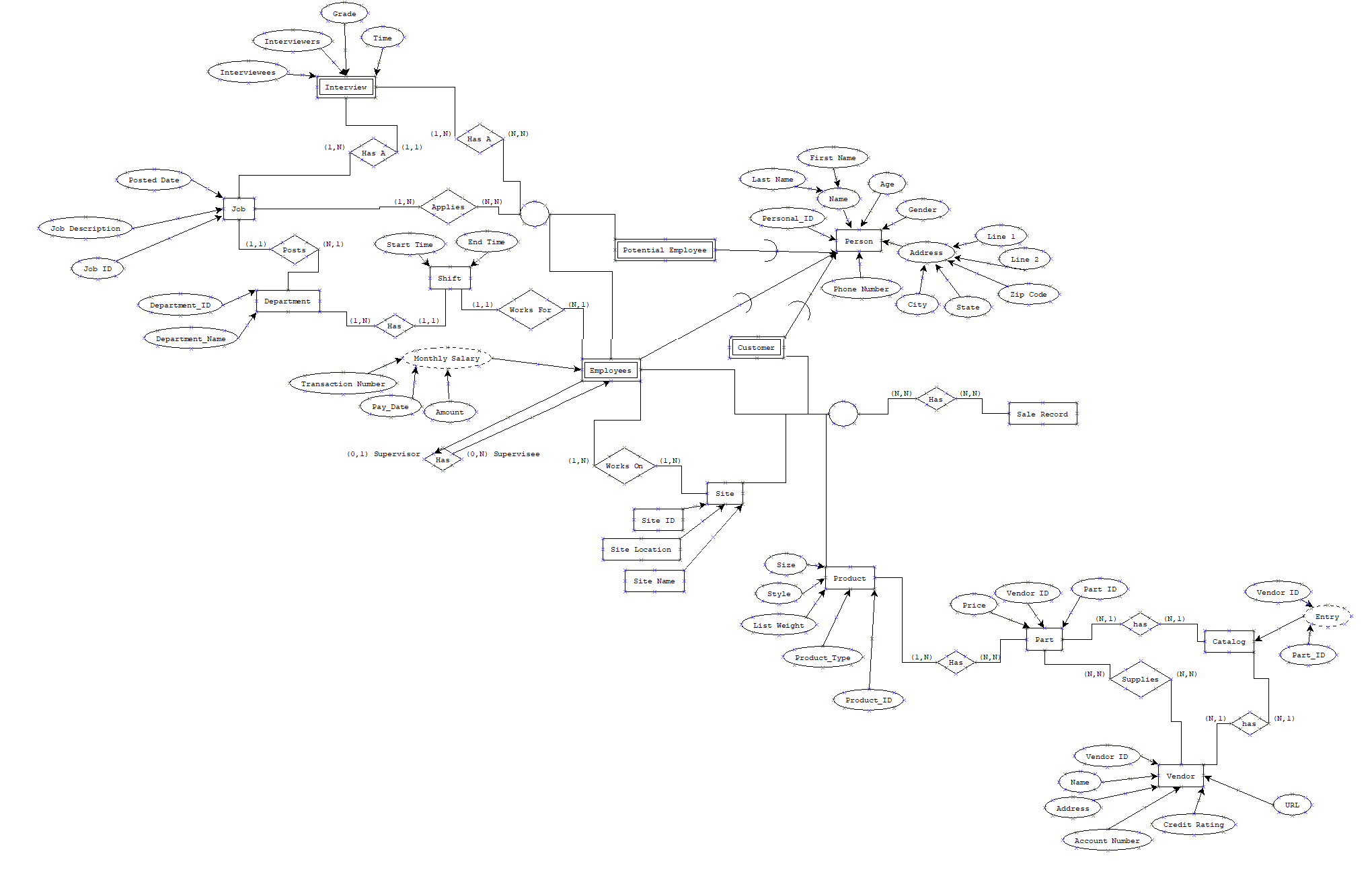
**Justify using a relational DBMS Software for this project**

A database management system software is required for this enterprise business logic. Computer systems are extremely ideal for this type of logging system with multiple entity types that relate to eachother.

# 2. EER diagram

Portrait 

Landscape



# 3. Relational schema

# The main relations stemmed from the personal\_ID and employee\_ID

# 

# 

# 

# 4. Database structure

# 

# 5. Database SQL setup script

CREATE TABLE person (

first\_name VARCHAR2(30) NOT NULL,

last\_name VARCHAR2(30),

personal\_id INTEGER NOT NULL,

age INTEGER,

gender VARCHAR2(1),

address\_line\_1 VARCHAR2(30),

city VARCHAR2(30),

address\_line\_2 VARCHAR2(30),

state VARCHAR2(30),

zip\_code INTEGER,

phone\_number INTEGER,

primary key (personal\_id)

);

CREATE TABLE customer (

prefered\_salesman VARCHAR2(30),

customer\_id INTEGER NOT NULL,

personal\_id INTEGER NOT NULL,

primary key (customer\_id),

foreign key (personal\_id) references person

);

CREATE TABLE monthly\_salary (

transaction\_numer INTEGER NOT NULL,

pay\_date DATE,

amount FLOAT,

monthly\_salary\_id NUMBER NOT NULL,

primary key (monthly\_salary\_id)

);

CREATE TABLE employee (

title VARCHAR2(30),

supervisor\_id INTEGER,

employee\_id INTEGER NOT NULL,

monthly\_salary\_id NUMBER,

transaction\_numer INTEGER,

personal\_id INTEGER NOT NULL,

primary key (employee\_id),

foreign key (personal\_id) references person,

foreign key (monthly\_salary\_id) references monthly\_salary

);

CREATE TABLE potential\_employee (

potential\_employee\_id INTEGER NOT NULL,

personal\_id INTEGER NOT NULL,

primary key (potential\_employee\_id),

foreign key (personal\_id) references person

);

CREATE TABLE interviewee (

interviewee\_id INTEGER NOT NULL,

potential\_employee\_id INTEGER NOT NULL,

primary key (interviewee\_id),

foreign key (potential\_employee\_id) references potential\_employee

);

CREATE TABLE interviewer (

job\_id INTEGER,

interviewer\_id INTEGER NOT NULL,

employee\_id INTEGER NOT NULL,

primary key (interviewer\_id),

foreign key (employee\_id) references employee

);

CREATE TABLE interview (

job\_position INTEGER,

interview\_time TIMESTAMP,

interviewee\_id INTEGER NOT NULL,

job\_id INTEGER NOT NULL,

interviewer\_id INTEGER NOT NULL,

interview\_id INTEGER NOT NULL,

grade NUMBER,

primary key (interview\_id),

foreign key (interviewer\_id) references interviewer,

foreign key (interviewee\_id) references interviewee

);

CREATE TABLE department (

department\_id INTEGER NOT NULL,

department\_name VARCHAR2(30 CHAR),

primary key (department\_id)

);

CREATE TABLE job\_position (

job\_id INTEGER NOT NULL,

job\_description VARCHAR2(30),

posted\_date TIMESTAMP,

department\_id INTEGER NOT NULL,

primary key (job\_id),

foreign key (department\_id) references department

);

CREATE TABLE product (

p\_size VARCHAR2(25),

p\_style VARCHAR2(25),

list\_weight VARCHAR2(25),

product\_id INTEGER NOT NULL,

product\_type VARCHAR2(25),

product\_name VARCHAR2(25),

part\_id INTEGER NOT NULL,

primary key (product\_id),

foreign key (part\_id) references part

);

CREATE TABLE part (

price NUMBER,

part\_id NUMBER NOT NULL,

primary key (part\_id)

);

CREATE TABLE vendor (

vendor\_id INTEGER NOT NULL,

name VARCHAR2(25),

address VARCHAR2(25),

account\_number INTEGER,

credit\_rating INTEGER,

url VARCHAR2(25),

primary key (vendor\_id)

);

CREATE TABLE catalog (

catalog\_id NUMBER NOT NULL,

part\_id NUMBER NOT NULL,

vendor\_id INTEGER NOT NULL,

primary key (catalog\_id),

foreign key (part\_id) references part,

foreign key (vendor\_id) references vendor

);

CREATE TABLE sales\_record (

sale\_time DATE,

sales\_record\_id NUMBER NOT NULL,

sales\_id INTEGER NOT NULL,

employee\_id INTEGER NOT NULL,

customer\_id INTEGER NOT NULL,

product\_id INTEGER NOT NULL,

primary key (sales\_id),

foreign key (employee\_id) references employee,

foreign key (customer\_id) references customer,

foreign key (product\_id) references product

);

CREATE TABLE shift (

end\_time TIMESTAMP,

start\_time TIMESTAMP,

employee\_id INTEGER NOT NULL,

shift\_id INTEGER NOT NULL,

department\_id INTEGER NOT NULL,

primary key (shift\_id ),

foreign key (department\_id) references department,

foreign key (employee\_id) references employee

);

CREATE TABLE site (

site\_id INTEGER NOT NULL,

site\_location VARCHAR2(20),

site\_name VARCHAR2(20),

primary key (site\_id)

);

CREATE SEQUENCE monthly\_salary\_id START WITH 1 NOCACHE ORDER;

CREATE OR REPLACE TRIGGER monthly\_salary\_id BEFORE

INSERT ON monthly\_salary

FOR EACH ROW

WHEN ( new.monthly\_salary\_id IS NULL )

BEGIN

:new.monthly\_salary\_id := monthly\_salary\_id.nextval;

END;

/

CREATE SEQUENCE part\_id\_seq START WITH 1 NOCACHE ORDER;

CREATE OR REPLACE TRIGGER part\_id\_trg BEFORE

INSERT ON part

FOR EACH ROW

WHEN ( new.part\_id IS NULL )

BEGIN

:new.part\_id := part\_id\_seq.nextval;

END;

/

CREATE SEQUENCE sales\_record\_id START WITH 1 NOCACHE ORDER;

CREATE OR REPLACE TRIGGER sales\_record\_id BEFORE

INSERT ON sales\_record

FOR EACH ROW

WHEN ( new.sales\_record\_id IS NULL )

BEGIN

:new.sales\_record\_id := sales\_record\_id.nextval;

END;

/

# 6. Sample populated tables

--- employees

INSERT INTO person (first\_name, last\_name, personal\_id, age, gender, address\_line\_1, city, address\_line\_2, state, zip\_code, phone\_number) VALUES ('Alex','Lundin',1, 26, 'M', '123 Street', 'McKinney', '456 Street', 'TX', 75070, 4663945275);

INSERT INTO person (first\_name, last\_name, personal\_id, age, gender, address\_line\_1, city, address\_line\_2, state, zip\_code, phone\_number) VALUES ('John','Smith',2, 26, 'M', '123 Street', 'McKinney', '456 Street', 'TX', 75070, 4663945275);

INSERT INTO person (first\_name, last\_name, personal\_id, age, gender, address\_line\_1, city, address\_line\_2, state, zip\_code, phone\_number) VALUES ('Suezy','Q',3, 26, 'F', '123 Street', 'McKinney', '456 Street', 'TX', 75070, 4663945275);

--- sales

INSERT INTO person (first\_name, last\_name, personal\_id, age, gender, address\_line\_1, city, address\_line\_2, state, zip\_code, phone\_number) VALUES ('ZZ','Top',5, 26, 'M', '123 Street', 'McKinney', '456 Street', 'TX', 75070, 4663945275);

INSERT INTO person (first\_name, last\_name, personal\_id, age, gender, address\_line\_1, city, address\_line\_2, state, zip\_code, phone\_number) VALUES ('James','Bond',6, 26, 'M', '123 Street', 'McKinney', '456 Street', 'TX', 75070, 4663945275);

INSERT INTO person (first\_name, last\_name, personal\_id, age, gender, address\_line\_1, city, address\_line\_2, state, zip\_code, phone\_number) VALUES ('Batman','Wayne',7, 26, 'M', '123 Street', 'McKinney', '456 Street', 'TX', 75070, 4663945275);

--- pot emp

INSERT INTO person (first\_name, last\_name, personal\_id, age, gender, address\_line\_1, city, address\_line\_2, state, zip\_code, phone\_number) VALUES ('Sarah','Smith',4, 26, 'F', '123 Street', 'McKinney', '456 Street', 'TX', 75070, 4663945275);

INSERT INTO person (first\_name, last\_name, personal\_id, age, gender, address\_line\_1, city, address\_line\_2, state, zip\_code, phone\_number) VALUES ('Hellen','Cole',8, 26, 'F', '123 Street', 'McKinney', '456 Street', 'TX', 75070, 4663945275);

--- customers

INSERT INTO CUSTOMER (PREFERED\_SALESMAN, CUSTOMER\_ID, PERSONAL\_ID) VALUES ('Batman', 1, 1);

INSERT INTO CUSTOMER (PREFERED\_SALESMAN, CUSTOMER\_ID, PERSONAL\_ID) VALUES ('James Bond', 2, 2);

INSERT INTO CUSTOMER (PREFERED\_SALESMAN, CUSTOMER\_ID, PERSONAL\_ID) VALUES ('ZZ Top', 3, 3);

--- salary

INSERT INTO MONTHLY\_SALARY (TRANSACTION\_NUMER, AMOUNT, monthly\_salary\_id) VALUES (1, 50000, 1);

INSERT INTO MONTHLY\_SALARY (TRANSACTION\_NUMER, AMOUNT, monthly\_salary\_id) VALUES (2, 60000, 2);

--- employees

INSERT INTO EMPLOYEE (EMPLOYEE\_ID, PERSONAL\_ID, MONTHLY\_SALARY\_ID) VALUES (1, 5, 1);

INSERT INTO EMPLOYEE (EMPLOYEE\_ID, PERSONAL\_ID, MONTHLY\_SALARY\_ID) VALUES (2, 6, 2);

--- pot employees

INSERT INTO POTENTIAL\_EMPLOYEE (potential\_employee\_id, PERSONAL\_ID) VALUES (1, 4);

INSERT INTO POTENTIAL\_EMPLOYEE (potential\_employee\_id, PERSONAL\_ID) VALUES (2, 8);

--- interviewee

INSERT INTO interviewee (interviewee\_id, potential\_employee\_id) VALUES (1, 1);

INSERT INTO interviewee (interviewee\_id, potential\_employee\_id) VALUES (2, 2);

--- interviewer

INSERT INTO interviewer (job\_id, interviewer\_id, employee\_id) VALUES (12345, 1, 1);

INSERT INTO interviewer (job\_id, interviewer\_id, employee\_id) VALUES (11111, 2, 2);

--- interview

INSERT INTO interview (job\_position, interview\_time, interviewee\_id, job\_id, interviewer\_id, interview\_id, grade) VALUES (job\_position, interview\_time, interviewee\_id,job\_id,interviewer\_id,interview\_id,grade);

# 7. Views from part 2d

# 8. Queries from part 2e.

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 PERSON.FIRST\_NAME, PERSON.LAST\_NAME, PERSON.PERSONAL\_ID

from person

where PERSON.FIRST\_NAME = 'Hellen'

AND PERSON.PERSONAL\_ID =

(select POTENTIAL\_EMPLOYEE.PERSONAL\_ID

FROM POTENTIAL\_EMPLOYEE

where PERSON.PERSONAL\_ID = POTENTIAL\_EMPLOYEE.PERSONAL\_ID

AND POTENTIAL\_EMPLOYEE.POTENTIAL\_EMPLOYEE\_ID =

(select INTERVIEWEE.POTENTIAL\_EMPLOYEE\_ID

FROM INTERVIEWEE

where INTERVIEWEE.POTENTIAL\_EMPLOYEE\_ID = POTENTIAL\_EMPLOYEE.POTENTIAL\_EMPLOYEE\_ID

)

);

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

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

select EMPLOYEE.PERSONAL\_ID, PERSON.FIRST\_NAME

from EMPLOYEE

INNER JOIN PERSON

ON EMPLOYEE.PERSONAL\_ID = PERSON.PERSONAL\_ID;

WHERE EMPLOYEE.SUPERVISOR\_ID = null

AND EMPLOYEE.PERSONAL\_ID =

(select PERSON.PERSONAL\_ID

from person

where PERSON.PERSONAL\_ID = EMPLOYEE.PERSONAL\_ID

);

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

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

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

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

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

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

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

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

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

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

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

# 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.

# 9. Dependency diagrams

