I. (32 marks; continued on next two pages) Answer each of the following using no more than a few sentences in each case.

(a) Explain the difference between logical and physical data independence.

5/6

Logical data independence indicates applications immune to changes of data organizations

Physical data independence indicates applications immune to changes of storage structure.

(b) Explain each of the following terms:

1. relation schema

1/3

Relation schema represents the logical structure of all data in database.

3/3

2. one-to-many relationship

One entity can has many relations with the other entity set while the other only can have one relation

Eg: 

Prof/se can teach many students, students can only teached by one prof.

3. foreign key constraint

2/3

The foreign key is referencing some other key from other table.

The foreign key might be on delete cascade which deletes with the referencing table is an example of foreign key constraint.

Foreign key constraint have the constraints on the key refering by foreign key upon modification and deletion to protect logical structure.

(c) Consider the relational operators: (a) rename, (b) selection, (c) projection, and (d) natural join.

1. Is this collection of operators relationally complete?
   If not, list a minimal collection of additional operators that can be added to yield a relationally complete query language.

No. Since it has to be as expressive as relation algebra. 3/4

We still need union (∪), ~~intersect (∩)~~, except (−), ~~divide (/)~~ to have a complete query language.

2. Consider the full collection of operators (a) to (d) and 1. Can any of the operators (a) to (d) be omitted and still have a relationally complete query language?
   If yes, list which operators and explain why.

4/7

With × product and a) rename, d) natural join can be replaced since it can be represented by other operators in full collection.

(d) Assuming a relation R has attributes $\{A, B, C\}$ and a relation S has attributes $\{D, E\}$, and consider the following relational algebra query:

$$\pi_A(\sigma_{D=B}(\sigma_{A=6}(R) \times \pi_D(\sigma_{E=2}(S))))$$

1. Rewrite the above query to an equivalent query with at most three occurrences of unary or binary operators.

$\pi(A(\delta_{B=D \wedge E=2 \wedge A=6}(R \times S))$

3.

2. Express the query you rewrote in point (a) in Tuple Relational Calculus.

TRC

$\{ W \mid \exists R_1, S_1 (R_1 \in R \wedge S_1 \in S \wedge S_1.E = 2 \wedge R_1.A = 6 \wedge R_1.B = S_1.D \wedge W.A = R_1.A) \}$

3. Express the query you rewrote in point (a) in Domain Relational Calculus.

DRC

$\{ \langle A \rangle \mid \exists A, B, D (R(A, B, C) \wedge S(D, 2) \wedge B = D \wedge A = 6) \}$

2

II. (40 marks; continued on next two pages) Consider the following SQL data definition for maintaining information about employees at a hypothetical company.

```
CREATE TABLE emp
   (  num INTEGER NOT NULL,
      name VARCHAR(20) NOT NULL,
      dept VARCHAR(20) NOT NULL,
      sal INTEGER NOT NULL,
      boss INTEGER NOT NULL,
   PRIMARY KEY (num),
   FOREIGN KEY (boss) REFERENCES emp (num) );
```

You can assume the following:

1. There is one person, the president, that has herself/himself as the boss;
2. All other employees have a boss they report to that is someone else;
3. There are no cycles in the boss hierarchy for anyone other than the president. (A cycle would exist if, for example, Fred was the boss of Mary and Mary was in turn the boss of Fred.)

Also, an employee is referred to as a *manager* if that employee is the boss of at least one other employee.

Write each of the following queries in the respective query language.

(10) **(a) Relational Algebra:** The number, name and salary of each employee whose boss is in a different department.

ρ_R(dept→deptNN, num→numNN ($\overline{\Pi}$ dept, num (emp))

$\overline{\Pi}$ num, name, salary ($\sigma_{R.deptNN \,!=\, dept}$

($\sigma_{R.numNN = boss}$ (emp, × R)))

(10) **(b) SQL:** The number, name and salary of each employee whose boss is in a different department.

select num, name, salary from emp E,
emp Em. where E.boss = Em.num
and E.dept <> Em.dept

(c) **SQL:** The departments that have an employee with a salary among the highest and for whom the boss of this employee is not the president.

```
select distinct E.dept from emp E where E.num = 1 ?
    select E.num from emp E1 where
        E1.salary = max(b.salary) and
    E1.boss <> ( select num from
        emp E2 where E2.num = E2.boss)
    ? group by E1.num)
```

6/10

(d) **SQL:** The percentage of employees for whom the salary of their boss is greater than the average salary of a manager.

5/10

```
select count(E.num) / count(E1.num) as percentage
    from emp E, emp E1, emp Em, where
        E.boss = E1.num and
        E1.salary > avg(Em.salary) and
        exists ( select * from emp E4 where
            E4.boss = Em.num)
    ? Group by Em.num, E1.num
```

III. (14 marks; continued on next page) Consider the following schema, and suppose you have two views defined as below:

Emp(*eid* : *integer*, *ename* : *string*, *age* : *integer*, *salary* : *real*)
Works(*eid* : *integer*, *did* : *integer*, *pct_time* : *integer*)
Dept(*did* : *integer*, *budget* : *real*, *managerid* : *integer*)

```
CREATE VIEW SeniorEmp (sname, sage, salary)
AS SELECT E.ename, E.age, E.salary
FROM Emp E
WHERE E.age > 50

CREATE VIEW AvgSalaryByAge (age, avgSalary)
AS SELECT E.eid,  AVG (E.salary)
FROM Emp E
GROUP BY E.age
```

Answer the following questions:

(a) Write an SQL query, **without using any views**, to produce the same results as the following query:

```
SELECT S.sname
FROM SeniorEmp S
WHERE S.salary > 100,000
```

2 select sname from SeniorEmp S, 2
2 where S1.salary > 100,000 2

0/6

(b) For each of the above two views: can you insert a new record into the view? If the answer is *Yes*, give the corresponding SQL statement. If the answer is *No*, give the reason for the failure.

SeniorEmp:

Yes ✓

```
insert into SeniorEmp
    values ('Kathy', 60, 60000)
```

2

AvgSalaryByAge:

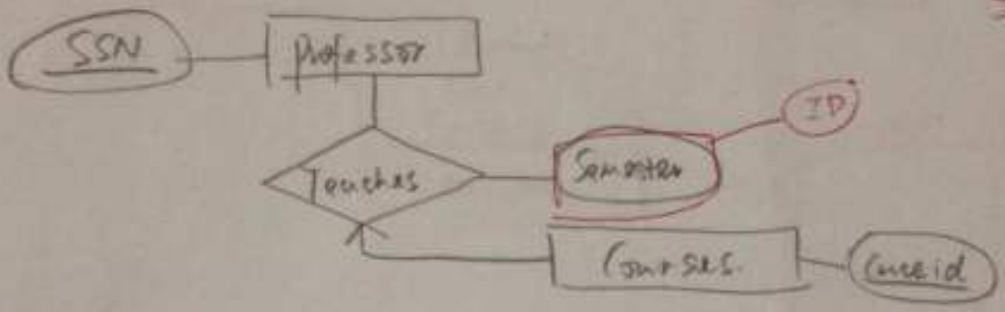
No. The view update is not supported when using an aggregation like AVG(E).

4

Answer the following questions:

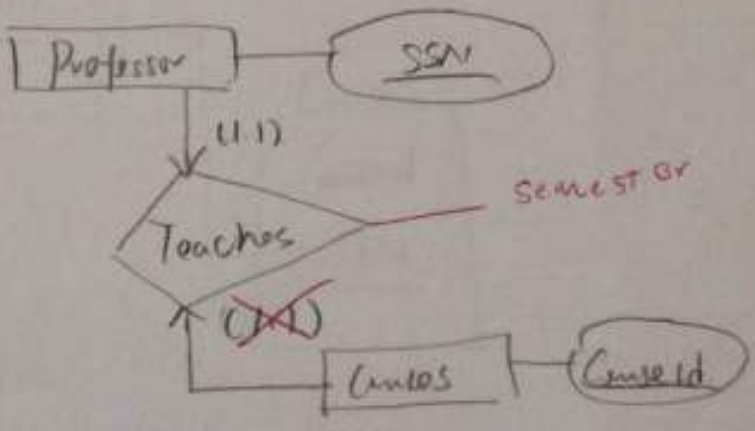
(a) For each situation, draw an ER diagram that describes the above database requirements (assuming no further constraints hold).

1.



5

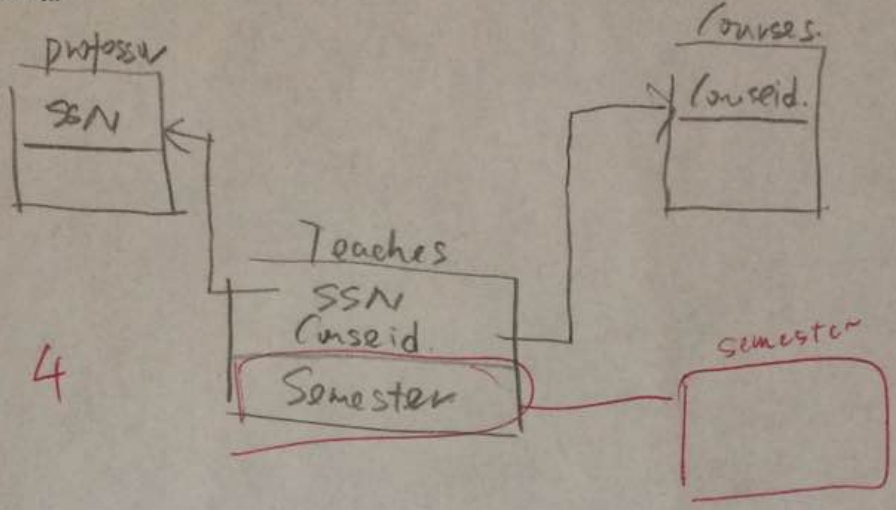
2.



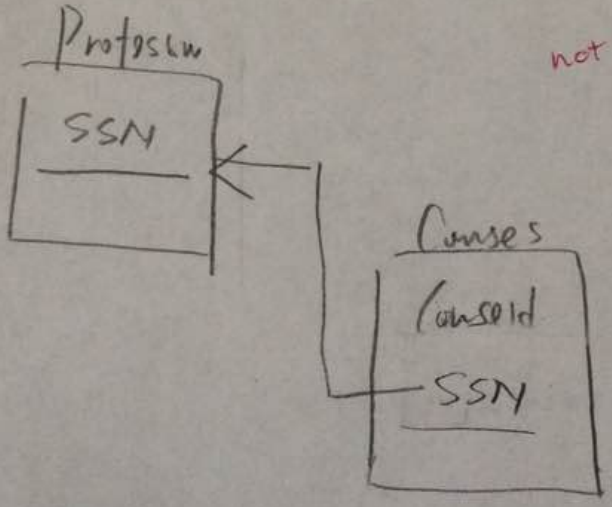
6

(b) **(Extra)** For each situation, translate your ER diagram into a relational database schema.

1.



2.



not enough

2