CSE 414 Final Examination

Monday, June 10, 2019, 2:30-4:20

Name: ______________________________

Question	Points	Score
1	25	
2	15	
3	20	
4	40	
5	20	
6	40	
7	40	
Total:	200	

- This exam is CLOSED book and CLOSED devices.
- You are allowed TWO, HAND-WRITTEN letter-size sheets with notes (both sides).
- You have 110 minutes;
- Answer the easy questions before you spend too much time on the more difficult ones.
- Good luck!

1 Relational Data Model

1. (25 points)

A marine biologist collects samples of micro-organisms from the ocean. She stores her data in a relational database with the following schema:

```
Sample(sid, lat, long, depth, technician)
Analysis(sid, oid)
Organism(oid, name)
```

- `Sample` represents a small amount of water taken from the ocean. The data records the latitude, longitude, and depth where the sample was taken (real numbers), and the name of the technician (of type text) who collected and analyzed the sample.
- Each sample is analyzed for organisms. `Analysis` lists all organisms found in that sample. `sid` and `oid` are foreign keys to `Sample` and `Organism` respectively (integers).
- `Organism` is a collection of names of organisms (e.g. marine microbes). The name is a text.
- `sid`, `oid` are integers.

Solution:

```
-- FOR TESTING ONLY!!!
drop table if exists Analysis;
drop table if exists Sample;
drop table if exists Organism;
create table Sample (sid int primary key,
                     lat real, long real, depth int,
                     technician text);
create table Organism (oid int primary key, name text);
create table Analysis (sid int references Sample,
                       oid int references Organism);
```

(a) (5 points) Technicians are rewarded by the total number of organisms that they identify in all samples. Write a SQL query that computes, for each technician, the total number of organisms that they found in all samples. (If Alice finds, say, Trichodesmium, in two different samples, then you count this as 2.) Your query should return the technician's name, the total number of organisms they identified, ordered decreasingly by this number.
Write your answer below:

Solution:

```
select x.technician, count(*)
from Sample x, Analysis y
where x.sid = y.sid
```

```
group by x.technician
order by count(*) desc;
```

```
Sample(sid, lat, long, depth, technician)
Analysis(sid, oid)
Organism(oid, name)
```

(b) (5 points) The biologists is particularly interested in the geographical area with latitude from 20.5 to 22.3 and longitude from −158.3 to −156.7. (Yes, this is close to Hawaii!) Write a SQL query that returns the `oid`s and names all organisms that were found in more than 100 samples from this area.

Write your answer below:

Solution:

```
select z.oid, z.name
from Sample x, Analysis y, Organism z
where x.sid = y.sid and y.oid = z.oid
  and 20.5 < x.lat and x.lat < 22.3
  and -158.3 < x.long and x.long < -156.7
group by z.oid, z.name
having count(*) > 100;
```

```
Sample(sid, lat, long, depth, technician)
Analysis(sid, oid)
Organism(oid, name)
```

(c) (15 points) A deep-see organism is one that lives only below a depth of 1000m. Write a SQL query that returns the names of all organisms that were found only at a depth greater than 1000m. (The `depth` is an integer and is represented in meters.) **Write your answer below:**

Solution:

```
select x.oid, x.name
from Organism x
where not exists
  (select *
   from Analysis y, Sample z
   where x.oid = y.oid and y.sid = z.sid and z.depth < 1000);
```

```
Sample(sid, lat, long, depth, technician)
Analysis(sid, oid)
Organism(oid, name)
```

2 NoSQL, JSON, SQL++

2. (15 points)

The raw data that biologist used to populate her database came in JSon format directly from the ship where the samples were collected and analyzed. It had the following structure:

```
{"samples": [
    { "sid": "s001",
      "lat": "21.2",
      "long": "-157.9",
      "depth": "300",
      "analysis":
        { "technician": "Alice",
          "organisms": [ {"oid": "o252", "name": "Trichodesmium"},
                         {"oid": "o301", "name": "Crocosphaera"},
                         ...
                       ]
        }
    },
    { "sid": "s002",
      "lat": "25.0",
      "long": "-150.0",
      "depth": "400",
      "analysis":
        { "technician": "Bob",
          "organisms": [ {"oid": "o301", "name": "Crocosphaera"},
                         {"oid": "o999", "name": "Prochlorococcus"},
                         {"oid": "o777", "name": "Synechococcus"},
                         ...
                       ]
        }
    },
    ...
    ]
}
```

Write tree SQL++ queries to convert the JSon data above into the relational schema.

(a) (5 points) Write the SQL++ query to construct the `Sample` table. Your query should return an output like:

```
[ { "sid": "s001", "lat": "21.2", "long": "-157.9", "depth": "300", "technician": "Alice"},
  { "sid": "s002", "lat": "25.0", "long": "-150.0", "depth": "400", "technician": "Bob" },
  ...
]
```

Solution:

```
select x.sid, x.lat, x.long, x.depth, x.analysis.technician
from samples x;
```

(b) (5 points) Write the SQL++ query to construct the `Analysis` table. Your query should return an output like this:

```
[ { "sid": "s001", "oid": "o252"},
  { "sid": "s001", "oid": "o301"},
  ...
  { "sid": "s002", "oid": "o301"},
  ...
]
```

Solution:

```
select x.sid, y.oid
from samples x, x.analysis.organisms y;
```

(c) (5 points) Write the SQL query to construct the `Organisms` table. Your query should return an output like this:

```
[ { "oid": "o252", "name": "Trichodesmium"},
  { "oid": "o301", "name": "Crocosphaera},
  { "oid": "o999", "name": "Prochlorococcus},
  ...
]
```

Solution:

```
select distinct y.oid, y.name
from samples x, x.analysis.organisms y;
```

3 Datalog

3. (20 points)

At Gotham University[1] each course has two prerequisites: for some courses both prerequisites are required, for other courses only one of the two prerequisites is required. The only exceptions are introductory courses, which don't have prerequisites. Students can only take one course every quarter.

The schema is:

```
Course(cid, name, noQuarters)
NoPrereq(cid)
PrereqOneOfTwo(cid, cid1, cid2)
PrereqTwoOfTwo(cid, cid1, cid2)
```

(a) (5 points) The university requires that every course appear under either `NoPrereq` or under `PrereqOneOfTwo` or under `PrereqTwoOfTwo`. Write a datalog program that checks this constraint. Your program should return the `cid`'s and `name`'s of courses that do not occur in `NoPrereq` or `PrereqOneOfTwo` or `PrereqTwoOfTwo`. **Write your answer below:**

Solution:

```
Listed(cid) :- NoPrereq(cid)
Listed(cid) :- PrereqOneOfTwo(cid, -, -)
Listed(cid) :- PrereqTwoOfTwo(cid, -, -)
Answer(cid,name :- Course(cid, name, -), not Listed(cid)
```

[1] The *Wise Men of Gotham* are supposed to have feigned idiocy to avoid a Royal visit by King John. Gotham is also a fictional city in the comics of Batman.

```
Course(cid, name, noQuarters)
NoPrereq(cid)
PrereqOneOfTwo(cid, cid1, cid2)
PrereqTwoOfTwo(cid, cid1, cid2)
```

(b) (15 points) Write a datalog program that computes, for each course, the smallest number of quarters need for a student to take sufficient prerequisites for that course, and the course itself. Your query should return pairs `oid`, `number-of-quarters`. For example, if the database is the following:

Course:

cid	name	noQuarters
Math101	Math	3
Java102	Java	2
DB414	DB	3
ML446	ML	5

NoPrereq

cid
Math101
Java102

PrereqOneOfTwo

cid	cid1	cid2
DB414	Math101	Java102

PrereqTwoOfTwo

cid	cid1	cid2
ML446	Math101	Java102

then your answer should be:

cid	c	
Math101	3	– because has no prereq
Java102	2	– because has no prereq
DB414	5	– can take either Math101 (3qtr) or Java102 (2qtr) plus DB414 (3qtr)
ML446	10	– must take Math101 (3qtr) then Java102 (2qtr) plus ML446 (5qtr)

Hint if you were to compute the number of quarters that a student needs to take “Math101” and “Java102”, which don’t have any prerequisites, you could write:

```
Q(c1+c2) :- Course("Math101", -, c1), Course("Java102", -, c2)
```

Write your answer below:

Solution:

```
Q(cid, q) :- NoPrereq(cid), Course(cid, -, q)
Q(cid, q+q1) :- PrereqOneOfTwo(cid, cid1, cid2),
                Q(cid1,q1), Q(cid2,q2), q1<q2,
                Course(cid, -, q)
Q(cid, q+q2) :- PrereqOneOfTwo(cid, cid1, cid2),
                Q(cid1,q1), Q(cid2,q2), q1>=q2,
                Course(cid, -, q)
Q(cid, q+q1+q2) :- PrereqTwoOfTwo(cid, cid1, cid2),
                   Q(cid1,q1), Q(cid2,q2),
                   Course(cid, -, q)
```

```
Sample(sid, lat, long, depth, technician)
Analysis(sid, oid)
Organism(oid, name)
```

4 Query Execution and Optimization

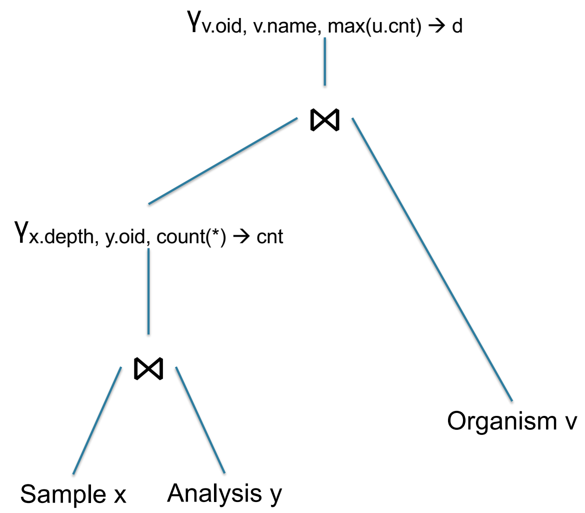
4. (40 points)

   (a) (10 points) Write a logical plan for the following query

```
with  DepthCnt as
   (select x.depth, y.oid, count(*) as cnt
    from Sample x, Analysis y
    where x.sid = y.sid
    group by x.depth, y.oid)
select v.oid, v.name, max(u.cnt) as d
from DepthCnt u, Organism v
where u.oid = v.oid
group by v.oid, v.name;
```

You should turn in a relational algebra tree.

Write your answer below:

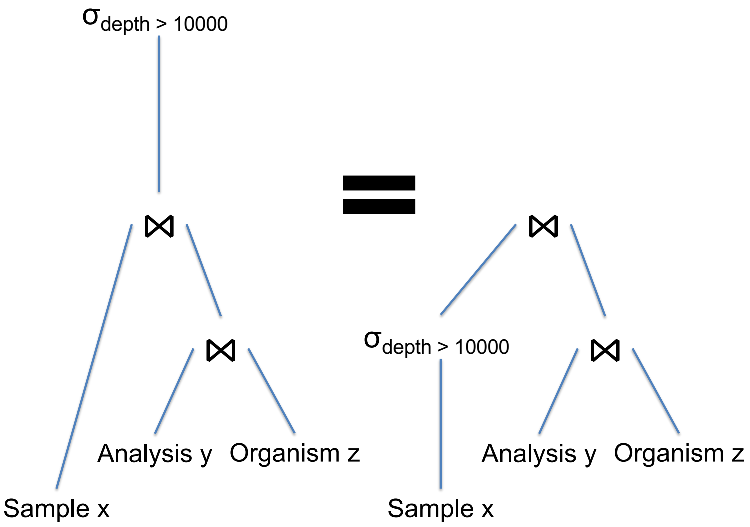

$\gamma_{v.oid, v.name, max(u.cnt) \rightarrow d}$
⋈
$\gamma_{x.depth, y.oid, count(*) \rightarrow cnt}$
⋈
Sample x
Analysis y
Organism v

Solution:

```
Sample(sid, lat, long, depth, technician)
Analysis(sid, oid)
Organism(oid, name)
```

(b) Indicate which of the optimization rules below are correct. All joins are natural joins:

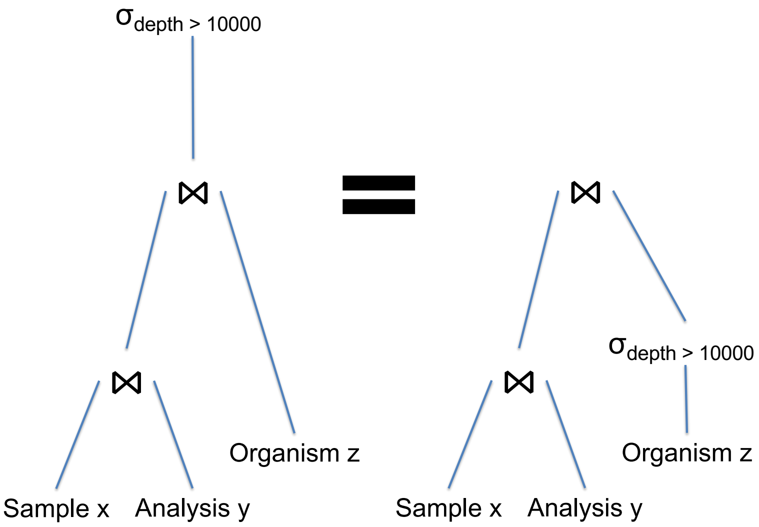
i. (2 points)



i. **Yes**

Correct? [Yes/no]:

ii. (2 points)

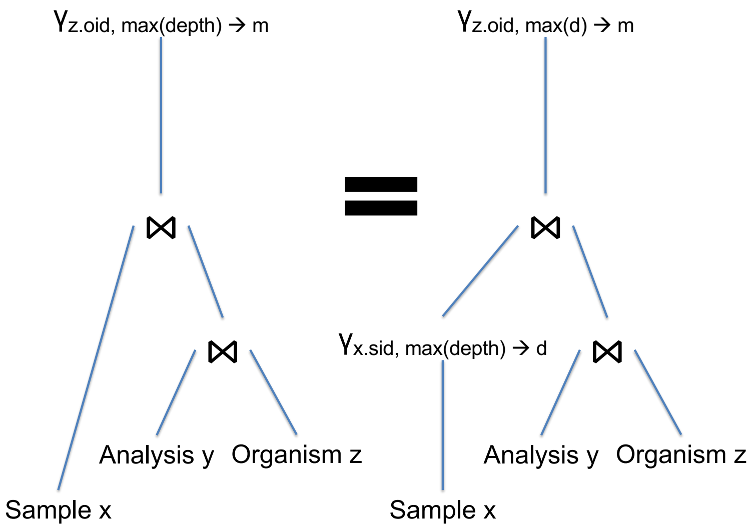


ii. **No**

Correct? [Yes/no]:

```
Sample(sid, lat, long, depth, technician)
Analysis(sid, oid)
Organism(oid, name)
```

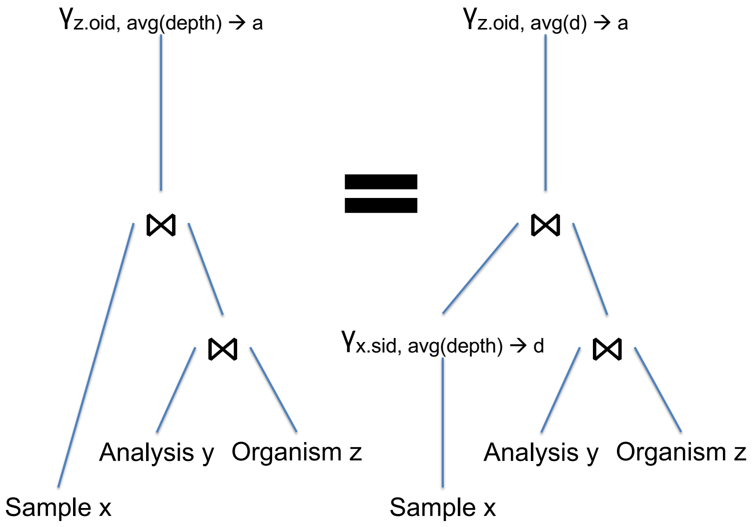
iii. (2 points)



iii. **yes**

Correct? [Yes/no]:

iv. (2 points)



iv. **YES(see note)**

Correct? [Yes/no]:

Solution:

Note The previous two questions were designed wrongly. The lower group-by on the right is applied to a key, so it is trivally a no-op. As stated, both equalities hold trivially, because the extra group by does nothing, so the answer is "YES" for both.
The correct design should have applied the new group on a join:

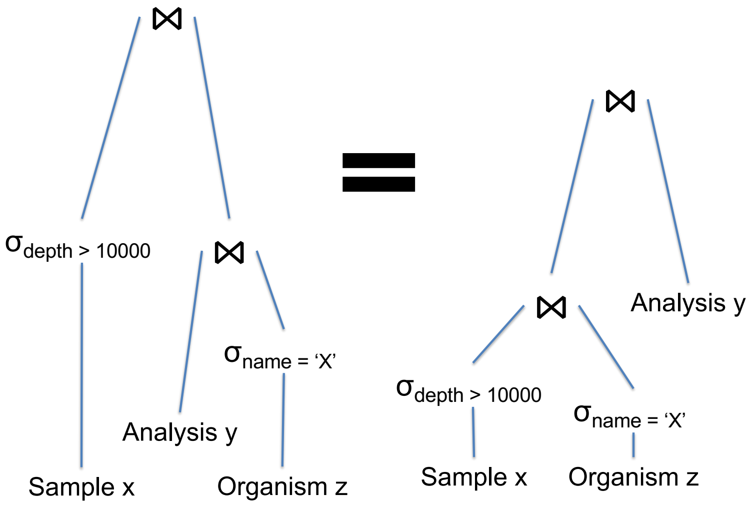
$$\gamma_{z.oid,\max(d)\to m}\Big(\gamma_{x.sid,\max(depth)\to d}((\texttt{Sample}\ x) \bowtie (\texttt{Analysis}\ y)) \bowtie \texttt{Organism}\ z\Big)$$
$$\gamma_{z.oid,avg(d)\to a}\Big(\gamma_{x.sid,avg(depth)\to d}((\texttt{Sample}\ x) \bowtie (\texttt{Analysis}\ y)) \bowtie \texttt{Organism}\ z\Big)$$

The the first answer is "yes", while the second is "no", because the average of averages is not always the average. For example:

$$avg(1,1,9,11,13) = 7 \neq avg(avg(1,1), avg(9,11,13)) = avg(1,11) = 6$$

```
Sample(sid, lat, long, depth, technician)
Analysis(sid, oid)
Organism(oid, name)
```

v. (2 points)



v. yes

Correct? [Yes/no]:

```
Sample(sid, lat, long, depth, technician)
Analysis(sid, oid)
Organism(oid, name)
```

(c) (10 points) Assume the following statistics on the database:

$T(\texttt{Sample}) = 100,000$	$T(\texttt{Analysis}) = 20,000,000$	$T(\texttt{Organism}) = 60,000$
$V(\texttt{Sample}, \texttt{technician}) = 100$	$V(\texttt{Analysis}, \texttt{sid}) = 10,000$	$V(\texttt{Organism}, \texttt{name}) = 15,000$
	$V(\texttt{Analysis}, \texttt{oid}) = 45,000$	

Estimate the size of the answer to the following SQL query. You should make the usual uniformity, independence, and preservation of values assumption that we used in class:

```
select *
from Sample x, Analysis y, Organism z
where x.sid = y.sid and y.oid = z.oid
  and x.technician = 'Alice'
  and z.name = 'Synechococcus';
```

Solution:

$$\frac{T(\texttt{Sample}) \cdot T(\texttt{Analysis}) \cdot T(\texttt{Organism})}{\underbrace{V(\texttt{Sample}, \texttt{sid})}_{>V(\texttt{Analysis,sid})} \cdot \underbrace{V(\texttt{Organism}, \texttt{oid})}_{>V(\texttt{Analysis,oid})} \cdot V(\texttt{Sample}, \texttt{technician}) \cdot V(\texttt{Organism}, \texttt{name})}$$

$$= \frac{T(\texttt{Analysis})}{V(\texttt{Sample}, \texttt{technician}) \cdot V(\texttt{Organism}, \texttt{name})}$$

$$= \frac{20,000,000}{100 \cdot 15,000} = 13.3$$

```
Sample(sid, lat, long, depth, technician)
Analysis(sid, oid)
Organism(oid, name)
```

(d) The depths of the samples are highly skewed: for each additional 100m, the number of samples is reduced by half. (That is, half of the samples have depth < 100m; half of the rest have depth < 200m; half of the rest have depth < 300m, etc.). There is an unclustered, B^+ index on `Sample.depth`. Indicate the optimal physical plan that the optimizer should choose for each of the two queries below. Choose between *Table-scan and on-the-fly selection,* or *Index selection.*

i. (5 points)

```
    Q1: select * from Sample where depth < 100;
```

i. **Table scan**

Table scan or index selection?

ii. (5 points)

```
    Q1: select * from Sample where depth > 8000;
```

ii. **Index Selection**

Table scan or index selection?

```
Sample(sid, lat, long, depth, technician)
Analysis(sid, oid)
Organism(oid, name)
```

5 Parallel Query Processing

5. (20 points)

Consider the same relational database, and the same statistics:

$$T(\texttt{Sample}) = 100,000 \qquad T(\texttt{Analysis}) = 20,000,000$$

We are storing and processing the data on $P = 100$ servers. The data is initially block partitioned, and we compute the following query:

```
    select *
    from Sample x, Analysis y
    where x.sid = y.sid;
```

The query is computed distributively, on the 100 servers. You will be asked to estimate the number of answer tuples returned by each server. You only need to estimate the number of tuples in the **final answer**, not in any intermediate results. In case this estimate differs among servers, indicate the maximum number.

(a) **Plan 1:** `Sample` is broadcast to all 100 servers, then each server joins it with its local fragment of `Analysis`.

i. (5 points) Estimate the number of tuples/server assuming the data is uniform.

Solution: 200,000

ii. (5 points) Estimate the number of tuples/server assuming the data is skewed.

Solution: 200,000

(b) **Plan 2:** `Sample` and `Analysis` are hash-partitioned on the `sid` attribute on the 100 servers, then each server joins its local fragments.

i. (5 points) Estimate the number of tuples/server assuming the data is uniform.

Solution: 200,000

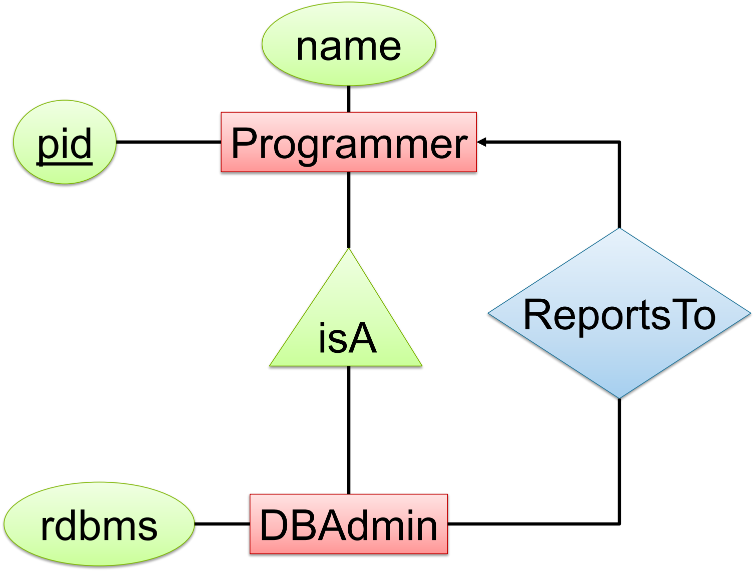
ii. (5 points) Estimate the number of tuples/server assuming the data is skewed.

Solution: 20,000,000

6 Conceptual Design

6. (40 points)

(a) (10 points) A large software company maintains a database of all its programmers. Some programmers are database administrators, and they have to report to some other programmer. The E/R diagram is given below. Write the CREATE TABLE statements for this E/R diagram, where `name` and `rdbms` are of type text, and `pid` is of type int. You answer should include keys and foreign keys where necessary.



Write your answer below:

Solution:

```
create table Programmer
    (pid int primary key,
     name text);
create table DBAdmin
    (pid int primary key references Programmer,
     rdbms text,
     reportsTo int references Programmer);
```

(b) (10 points) Consider a relation $R(A, B, C, D, E)$ satisfying the following FD's:

$$A \rightarrow B \qquad C \rightarrow B \qquad BD \rightarrow E$$

Decompose R into BCNF. In your final answer indicate the key of each relation.

Solution: Solution 1:

- in $R(ABCDE)$: $A+ = AB$ split into $R_1(AB), R_2(ACDE)$.
- in $R_2(ACDE)$: $CD+ = CDE$ split into $R_3(CDE), R_4(ACD)$.

Final answer: $R_1(\underline{A}B), R_3(\underline{CD}E), R_4(ACD)$.
Solution 2:

- in $R(ABCDE)$: $C+ = BC$ split into $R_1(BC), R_2(ACDE)$.
- in $R_2(ACDE)$: $CD+ = CDE$ split into $R_3(CDE), R_4(ACD)$.

Final answer: $R_1(B\underline{C})$, $R_3(\underline{CD}E)$, $R_4(ACD)$.
Solution 3:

- in $R(ABCDE)$: $BD+ = BDE$ split into $R_1(BDE)$, $R_2(ABCD)$.
- in $R_2(ABCD)$: $A+ = AB$ split into $R_3(AB)$, $R_4(ACD)$.

Final answer: $R_1(\underline{BD}E)$, $R_3(\underline{A}B)$, $R_4(ACD)$.
Solution 4:

- in $R(ABCDE)$: $BD+ = BDE$ split into $R_1(BDE)$, $R_2(ABCD)$.
- in $R_2(ABCD)$: $C+ = BC$ split into $R_3(BC)$, $R_4(ACD)$.

Final answer: $R_1(\underline{BD}E)$, $R_3(B\underline{C})$, $R_4(ACD)$.

(c) Consider a relation $R(\underline{A}, B, C, D, E, F, G)$ where A is a key. We create a new table S by running this query:

```
S:    select *  from R  where (B=C+2) and (D+E=F) and (G=7);
```

Indicate which of the following FDs are guaranteed to hold on S:

i. (2 points) $B \to C$

i. **Yes**

Yes or no?

ii. (2 points) $C \to B$

ii. **Yes**

Yes or no?

iii. (2 points) $D \to F$

iii. **No**

Yes or no?

iv. (2 points) $F \to D$

iv. **No**

Yes or no?

v. (2 points) $DE \to F$

v. **Yes**

Yes or no?

vi. (2 points) $F \to DE$

vi. **No**

Yes or no?

vii. (2 points) $F \to G$

vii. **Yes**

Yes or no?

viii. (2 points) $G \to F$

viii. **No**

Yes or no?

ix. (2 points) $CDE \to A$

ix. **No**

Yes or no?

x. (2 points) $A \to CDE$

x. **Yes**

Yes or no?

7 Transactions

7. (40 points)

   (a) For each schedule below indicate whether it is conflict serializable and, if it is, indicate the equivalent serial schedule. Show your work by drawing the precedence graph.

      i. (5 points)

$$R_2(A), R_4(C), W_2(B), W_1(D), W_3(A), R_4(D), R_1(B), W_3(C)$$

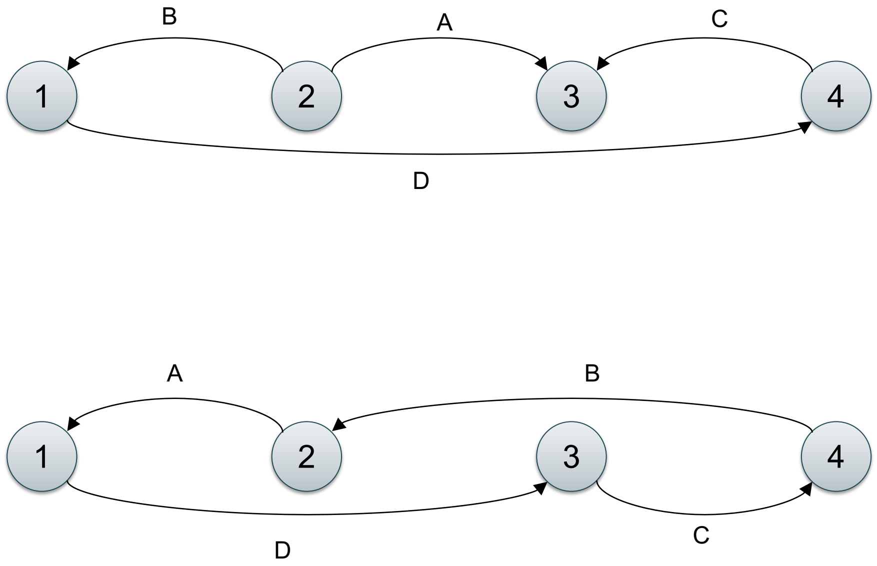
Solution: Conflict serializable, in the unique order $2, 1, 4, 3$.

      ii. (5 points)

$$R_2(A), R_1(D), W_4(B), R_3(C), W_3(D), R_2(B), W_4(C), W_1(A)$$

Solution: Not conflict serializable.

Solution: Both graphs:



(b) (10 points) Consider a database about the happiness status of Alice and Bob. Each morning Alice and Bob wake up happy or sad. Today their morning status is:

```
          A = happy          B = sad
```

Every day at noon we run these two transactions, at about the same time:

```
T1: START
    READ(A,x) - read Alice
    WRITE(B,x) - update Bob
    COMMIT
```

```
T2: START
    READ(B,y)
    if y='sad' then WRITE(A,'sad')
    COMMIT
```

Indicate which of the following outcomes are possible if we run the transactions (1) under the REPEATABLE READ isolation level[2], or (2) under the READ UNCOMMITTED isolation level. Write **yes** or **no** in each entry below.

Alice	Bob	REPEATABLE READS	READ UNCOMMITTED
		Possible?	Possible?
sad	sad	**Solution:** yes	**Solution:** yes
sad	happy	**Solution:** no	**Solution:** yes
happy	sad	**Solution:** no	**Solution:** no
happy	happy	**Solution:** yes	**Solution:** yes
DEADLOCK		**Solution:** yes	**Solution:** no

[2] This is the same as SERIALIZABLE on our static database.

(c) For each of the following statements indicate whether it is true or false:

i. (2 points) In a static database, every serializable schedule is conflict serializable.

i. **False**

True or false?

ii. (2 points) In a dynamic database, every serializable schedule is conflict serializable.

ii. **False**

True or false?

iii. (2 points) In a static database, every conflict serializable schedule is serializable.

iii. **True**

True or false?

iv. (2 points) In a dynamic database, every conflict serializable schedule is serializable.

iv. **False**

True or false?

v. (2 points) The two-phase locking protocol guarantees conflict serializability.

v. **True**

True or false?

vi. (2 points) The strict two-phase locking protocol guarantees conflict serializability.

vi. **True**

True or false?

vii. (2 points) Deadlocks can occur under the READ UNCOMMITTED isolation level.

vii. **True**

True or false?

viii. (2 points) Deadlocks can occur under the REPEATBLE READ isolation level.

viii. **True**

True or false?

ix. (2 points) Two transactions can hold the same SHARED LOCK at the the same time.

ix. **True**

True or false?

x. (2 points) Two transactions can hold the same EXCLUSIVE LOCK at the the same time.

x. **False**

True or false?