CSE 344 Final Examination

March 15, 2016, 2:30pm - 4:20pm

Name: ____________________________________

Question	Points	Score
1	47	
2	17	
3	36	
4	54	
5	46	
Total:	200	

- This exam is CLOSED book and CLOSED devices.
- You are allowed TWO letter-size pages with notes (both sides).
- You have 1h:50 minutes; budget time carefully.
- Please read all questions carefully before answering them.
- Some questions are easier, others harder; if a question sounds hard, skip it and return later.
- Good luck!

1 SQL

1. (47 points)

(a) (7 points) We represent sparse matrices as tables with three attributes: row, colummn, value. Write a SQL query that computes the product of two sparse matrices called A and B. Recall that the product $C = AB$ of two matrices is defined as:

$$C_{ik} = \sum_j A_{ij} B_{jk}$$

For example, consider the matrix product below:

$$A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 3 & 0 \\ 2 & 0 & 1 \end{bmatrix} \qquad B = \begin{bmatrix} 0 & 2 & 1 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad AB = \begin{bmatrix} 0 & 2 & 1 \\ 3 & 0 & 0 \\ 0 & 4 & 3 \end{bmatrix}$$

Then the inputs and output to your query would be:

A

row	col	val
1	1	1
2	2	3
3	1	2
3	3	1

B

row	col	val
1	2	2
1	3	1
2	1	1
3	3	1

Answer

row	col	val
1	2	2
1	3	1
2	1	3
3	2	4
3	3	3

Write a SQL query:

Solution:

```
drop table if exists A;
drop table if exists B;
drop table if exists C;


create table A(row int, col int, val int);
create table B(row int, col int, val int);
create table C(row int, col int, val int);

insert into A values(1, 1, 1);
insert into A values(2, 2, 3);
insert into A values(3, 1, 2);
insert into A values(3, 3, 1);

insert into B values(1, 2, 2);
insert into B values(1, 3, 1);
insert into B values(2, 1, 1);
insert into B values(3, 3, 1);
```

```
insert into C values(1, 1, 1);
insert into C values(2, 2, 1);
insert into C values(3, 3, 1);
```

```
select A.row as row, B.col as col, sum(A.val * B.val) as val
from A, B
where A.col = B.row
group by A.row, B.col;
```

2 points off for missing sum

1 point off for $A.row = B.col$

2 points off for worse join conditions

2 points off for missing group by

1 point off for nested subquery

(b) (10 points) The traces of a matrix A is the sum of the elements on the diagonal: $Tr(A) = \sum_i A_{ii}$. Write a SQL query that computes $Tr(ABC)$, where A, B, C are three sparse matrices.

Write a SQL query:

Solution: Notice that $Tr(ABC) = \sum_{i,j,k} A_{ij} B_{jk} C_{ki}$. The SQL query is:

```
select sum(A.val * B.val * C.val)
from A,B,C
where A.col=B.row and B.col=C.row and C.col=A.row;
```

less than ten students got this right (why??)

2 points off for lightly nested but correct query

6 points off for deeply nested, messy query

I gave only 1 point partial credit for solutions of the form:

```
select sum(A.val * B.val * C.val) from A,B,C where [wrong condition]
```

where the wrong condition was typically $A.row = A.col$ etc, or $A.row = B.row$ etc.

(c) Consider three relations with schemas:

$$R(A,B), S(C,D), T(E,F)$$

All attributes are NOT NULL, and no attribute is a key. For each query below, write an equivalent SQL query that is unnested. For example, if the given query were

`select distinct x.A from (select y.A as A from R y where y.B = 2) x;`

then your answer would be:

`select x.A from R x where x.B = 2;`

If query cannot be unnested, then indicate so.

i. (5 points) Unnest this query:

```
select distinct x.A, (select sum(y.B) from R y where x.A = y.A) as S
from R x;
```

Write an unnested SQL query or say *impossible*:

Solution:

```
drop table if exists R;
drop table if exists S;
drop table if exists T;

create table R(A int, B int);
create table S(C int, D int);
create table T(E int, F int);

insert into R values(1,20);
insert into R values(2,30);
insert into S values(20,300);
insert into T values(300,4000);

select x.A, sum(x.B) as S
from R x
group by x.A;
```

2 points off for any join
1 point off for missing group by

ii. (5 points) Unnest this query:

```
select x.A, x.B, (select sum(y.D) from S y where x.B = y.C) as S
from R x;
```

Write an unnested SQL query or say *impossible*:

Solution:

```
select x.A, x.B, sum(y.D) as S
```

```
from R x left outer join S y on x.B=y.C
group by x.A, x.B;
```

Note that the original query does not return any duplicates, because R does not contain duplicates (is a set), and A, B are all the attributes in R. THis means that it is possible to unnest.

2 points off for missing outer join

1 point off for missing group by

iii. (5 points) Unnest this query:

```
select distinct x.A as A, y.F as F
from R x, (select u.C as C, v.F as F
           from S u, T v
           where u.d = v.E) y
where x.B = y.C;
```

Write an unnested SQL query or say *impossible*:

Solution:

```
select distinct x.A as A, v.F as F
from R x, S u, T v
where x.B = u.C and u.D = v.E;
```

1 point off for missing distinct

iv. (5 points) Unnest this query:

```
select x.A as A, y.F as F
from R x, (select distinct u.C as C, v.F as F
           from S u, T v
           where u.D = v.E) y
where x.B = y.C;
```

Write an unnested SQL query or say *impossible*:

Solution: Not possible to unnest

(d) Consider four relations with the following schemas:

$$R(A, B), R2(A, B), S(C, D), V(G, H, K)$$

For each of the identities in the Relational Algebra below, indicate whether they hold. Assume set semantics for all operators:

i. (2 points) Does this identity hold?

$$(R - R2) - R2 = R$$

i. **No**

Yes/No:

ii. (2 points) Does this identity hold?

$$((R - R2) - R2) - R2 = R - R2$$

ii. **Yes**

Yes/No:

iii. (2 points) Does this identity hold?

$$\Pi_A(R \bowtie_{B=C} S) = \Pi_A(R)$$

iii. **No**

Yes/No:

iv. (2 points) Does this identity hold?

$$\Pi_{AB}(R \bowtie_{B=C} S) - R2 = \Pi_{AB}((R - R2) \bowtie_{B=C} S)$$

iv. **Yes**

Yes/No:

v. (2 points) Does this identity hold?

$$\gamma_{G,\text{sum}(K) \to L}(V) = \gamma_{G,\text{sum}(M) \to L}(\gamma_{G,H,\text{sum}(K) \to M}(V))$$

v. **Yes**

Yes/No:

2 Semistructured Data and JSon

2. (17 points)

   (a) Answer the multi-choice questions below:

      i. (3 points) What do we mean when we say that the data is *not in First Normal Form*? Check all that apply.

         1. There exists a non-trivial functional dependency $X \rightarrow Y$ where X is not a superkey.
         2. The data is represented in a human-readable form, like JSon.
         3. The value of an attribute of a table is a collection, such as a table or an array.
         4. The table has no clustered index.

         i. **3**

         Select from 1,2,3, and/or 4:

      ii. (3 points) Check which of the following statements are true about the semistructured data model:

         1. JSon is semistructured data.
         2. In semistructured data the value of an attribute can be another collection.
         3. Semistructured data means that the data is compressed.
         4. There are no query languages for semistructured data.

         ii. **1,2**

         Select from 1,2,3, and/or 4:

(b) Consider the following database, given in JSon:

```
{"Course":
  [ {"title": "Math101",
     "room": "F777",
     "instructor": {"Name": "Bob", "Office": "E999"};
     "enrollment": [{"name": "David", "year": 2},
                    {"name": "Erol", "year": 3}
                   ]
    },
    {"title": "Phys202",
     "room": "H909",
     "instructor": {"name": "Alice", "office": "C222"};
     "enrollment": [{"name": "Carol", "year": 2},
                    {"name": "Erol", "year": 3},
                    {"name": "Fred", "year": 1}
                   ]
    },
    {"title": "CSE703",
     "room": "G080",
     "instructor": {"name": "Bob", "office": "E999"}
    }
  ]
}
```

Your task is to convert this data into a relational database.

i. (6 points) Design a schema for the a relational database capable of storing the database above. Turn in relation names, their attributes, and underline the key. For example, you may write:

`Room(roomnumber, instructorname, floorid)`, `Floor(floorid, level)`

(not a real answer). Write your answer below:

Solution:

```
Course(title,room,instructor)
Instructor(name,office)
Student(name,year)
Enrollment(courseTitle,studentName)
```

JSon file repeated from previous page:

```
{"Course":
  [ {"title": "Math101",
     "room": "F777",
     "instructor": {"Name": "Bob", "Office": "E999"};
     "enrollment": [{"name": "David", "year": 2},
                    {"name": "Erol", "year": 3}
                   ]
    },
    {"title": "Phys202",
     "room": "H909",
     "instructor": {"name": "Alice", "office": "C222"};
     "enrollment": [{"name": "Carol", "year": 2},
                    {"name": "Erol", "year": 3},
                    {"name": "Fred", "year": 1}
                   ]
    },
    {"title": "CSE703",
     "room": "G080",
     "instructor": {"name": "Bob", "office": "E999"}
    }
  ]
}
```

ii. (5 points) Show the content of your tables, representing the same data as the JSon file. For example, you may write:

Room

roomnumber	instructorname	floorid
E999	Carol	flr0521
C222	Alice	flr0521

Floor

floorid	level
flr0521	2

(not a real answer) Turn in several table instances:

Solution:

Professor

Name	Office
Alice	C222
Bob	E999

Student

Name	Year
Carol	2
David	2
Erol	3
Fred	1

Course

Title	Room	Instructor
Math101	F777	Bob
Phys202	H909	Alice
CS303	G080	Bob

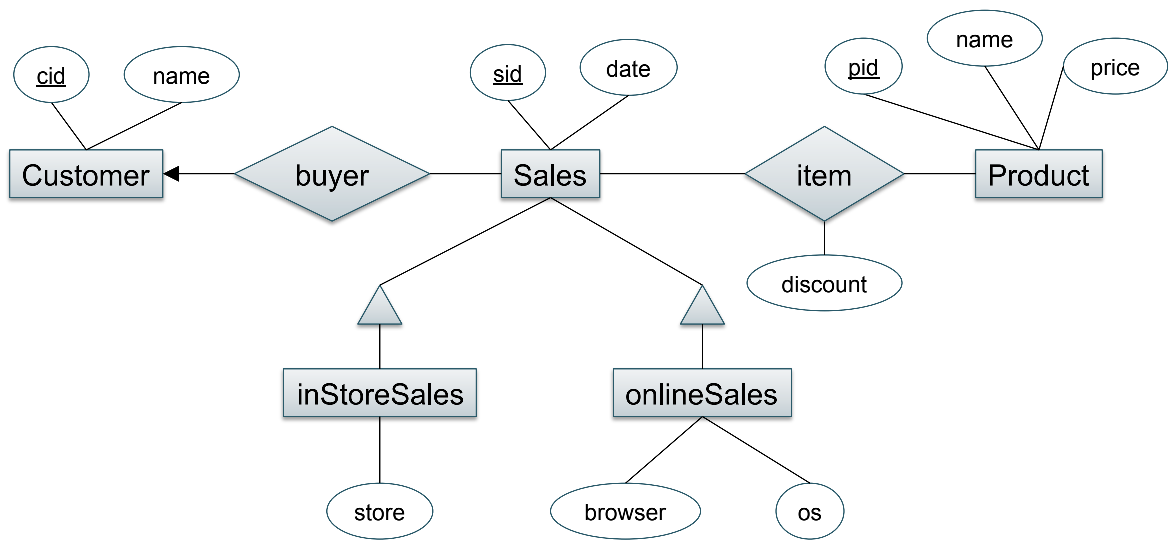
Enrollment

courseTitle	studentName
Math101	David
Math101	Erol
Phyis202	Carol
Phyis202	Erol
Phyis202	Fred

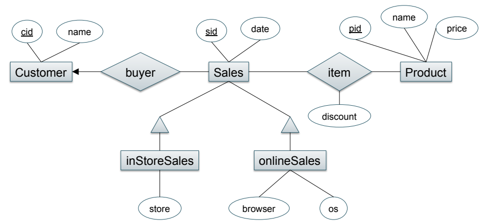
3 E/R Diagrams, Constraints, Conceptual Design

3. (36 points)

A company selling products both online and in their own stores has a database having a schema described by the E/R diagram below:



- `Sales` represents individual sales. One sale contains several products bought by one single customer. The `discount` is a real number representing a percent, for example `discount` = 33.33 means a discount of 1/3 from the original product price.
- `inStoreSales` consists of the sales in brick-and-mortar stores; each such sale contains the store name.
- `onlineSales` consists of online sales; each such sale includes the name of the browser and the operating system used during the purchase.
- `cid, sid, pid` are `int`, `price` and `discount` are `float`, the rest are `text`.
- No attributes may be null, except for `inStoreSales.store` and `onlineSales.os`.

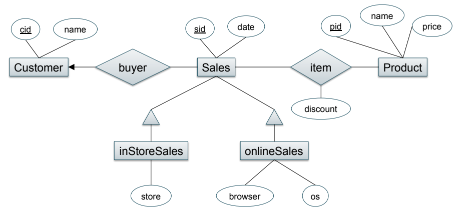


(a) (15 points) Create a database schema for the E/R diagram in the figure. You should turn an a set of CREATE TABLE statements. Your schema should include all constraints captured in the E/R diagrams.

Solution:

```
drop table if exists Item;
drop table if exists onlineSales;
drop table if exists instoreSales;
drop table if exists Sales;
drop table if exists Product;
drop table if exists Customer;

create table Customer (cid int primary key, name text not null);
create table Product
   (pid int primary key,
    name text not null,
    price float not null);
create table Sales (
     sid int primary key,
     date text not null,
     cid int not null references customer);
create table inStoreSales (
     sid int primary key references Sales,
     store text);
create table onlineSales (
     sid int primary key references Sales,
     browser text not null,
     os text);
create table item (
     pid int references Product,
     sid int references Sales,
     discount float not null,
     primary key (pid, sid));
```



(b) (10 points) Write a SQL query that computes, for each customer, the total amount that the customer spent online. Your query does not need to include customers who did not purchase anything oline.

Solution:

```
select v.cid, v.name, sum(u.price * (100-z.discount)/100)
from onlineSales x, Sales y, Item z, Product u, Customer v
where x.sid = y.sid
  and y.sid = z.sid
  and z.pid = u.pid
  and y.cid = v.cid
group by v.cid, v.name;
```

(c) Consider three relations:

$$R(\underline{A, B}, C), S(C, \underline{D}), T(D, A)$$

AB is a key in R, and D is a key in S. For each of the queries below, show the key of the query's answer, and compute $D+$:

i. (2 points) Query $Q1$:

```
select R.A, R.B, R.C, S.D
from R, S
where R.C = S.C and R.A = 20;
```

i. **Key=**BD**,** $D+ = ACD$

Key=? $D+$ =?

ii. (2 points) Query $Q2$:

```
select T.A, S.C, S.D
from S, T
where S.D = T.D;
```

ii. **Key=**AD**,** $D+ = CD$

Key=? $D+$ =?

iii. (2 points) Query $Q3$:

```
select R.A, R.B, R.C, S.D
from R, S, T
where R.A=T.A and R.C = S.C and S.D = T.D;
```

iii. **Key=**ABD**,** $D+ = CD$

Key=? $D+$ =?

(d) (5 points) Consider a relation $R(A_1, A_2, \ldots, A_n)$ satisfying the following functional dependencies:

$$
\begin{aligned}
A_1 &\rightarrow A_2 \\
A_2 &\rightarrow A_3 \\
A_3 &\rightarrow A_4 \\
&\ldots \\
A_{n-1} &\rightarrow A_n
\end{aligned}
$$

Decompose this relation into BCNF. You need to indicate only your answer by showing the relation names, their attributes, and their key, for example you may write:

$$R_1(\underline{A_2}, A_3, A_4 \ldots, A_n), R_2(A_1, \underline{A_3}, A_4 \ldots, A_n), R_2(A_1, A_2, \underline{A_4} \ldots, A_n), \ldots, R_n(\underline{A_1}, A_2, \ldots, A_{n-1})$$

(not the real answer).

Solution:

$$R_1(\underline{A_1}, A_2), R_2(\underline{A_2}, A_3), \ldots, R_{n-1}(\underline{A_{n-1}}, A_n)$$

4 Transactions

4. (54 points)

   (a) Consider a concurrency control manager that uses strict two phase locking that schedules three transactions:

   - $T1 : R_1(A), R_1(B), W_1(A), W_1(B), Co_1$
   - $T2 : R_2(B), W_2(B), R_2(C), W_2(C), Co_2$
   - $T3 : R_3(C), W_3(C), R_3(A), W_3(A), Co_3$

   Each transaction begins with its first read operation, and commits with the Co statement. Asnwer the following questions for each of the schedules below:

   - Is the schedule conflict-serializable? If yes, indicate a serialization order.
   - Is this schedule possible under a strict 2PL protocol?
   - If strict 2PL does not allow this schedule beause it denies a read or a write request, is the system in a deadlock at the time when the request is denied?

i. Schedule 1:

$R_2(B), W_2(B), R_3(C), W_3(C), R_3(A), W_3(A), Co_3, R_2(C), W_2(C), Co_2, R_1(A), R_1(B), W_1(A), W_1(B), Co_1$

α) (3 points) Is this schedule conflict-serializable? If yes, indicate a serialzation order.

Solution: yes: 3,2,1

β) (2 points) Is it possible under strict 2PL?

Solution: yes

γ) (2 points) Does strict 2PL lead to a deadlock?

Solution: no

ii. Schedule 2:

$R_2(B), W_2(B), R_3(C), W_3(C), R_1(A), R_1(B), W_1(A), W_1(B), Co_1, R_2(C), W_2(C), Co_2, R_3(A), W_3(A), Co_3$

α) (3 points) Is this schedule conflict-serializable? If yes, indicate a serialzation order.

Solution: no $L(C)$ and none can make progress.

β) (2 points) Is it possible under strict 2PL?

Solution: no $L(C)$ and none can make progress.

γ) (2 points) Does strict 2PL lead to a deadlock?

Solution: yes: T_1 holds $L(A)$, T_2 holds $L(B)$, T_3 holds $L(C)$ and none can make progress.

iii. Schedule 3:

$R_1(A), R_1(B), R_2(B), W_2(B), R_2(C), W_2(C), Co_2, R_3(C), W_3(C), R_3(A), W_3(A), Co_3, W_1(A), W_1(B), Co_1$

α) (3 points) Is this schedule conflict-serializable? If yes, indicate a serialzation order.

Solution: no:$R_2(B)$ is denited (or $W_2(B)$ is denied if shared locks are used)

β) (2 points) Is it possible under strict 2PL?

Solution: no

γ) (2 points) Does strict 2PL lead to a deadlock?

Solution: no

iv. Schedule 4:

$R_1(A), R_1(B), W_1(A), R_3(C), W_3(C), R_3(A), W_3(A), Co_3, W_1(B), R_2(B), W_2(B), Co_1, R_2(C), W_2(C), Co_2$

α) (3 points) Is this schedule conflict-serializable? If yes, indicate a serialzation order.

Solution: yes: 1,3,4;

β) (2 points) Is it possible under strict 2PL?

Solution: no: $W_2(B)$ is impossible since T_1 holds the lock on B

γ) (2 points) Does strict 2PL lead to a deadlock?

Solution: no

(b) (10 points) Consider the following three transactions:

- $T1 : R_1(A), W_1(B), Co_1$
- $T2 : R_2(B), W_2(C), Co_2$
- $T3 : R_3(C), W_3(D), Co_3$

Given an example of a conflict-serializable schedule that has the following properties: transaction $T1$ commits before transaction $T3$ starts, and the equivalent serial order is $T3, T2, T1$.

Solution: $R_1(A), R_2(B), W_1(B), Co_1, R_3(C), W_2(C), Co_2, W_3(D), Co_3$

Variations include: swap the first two reads (of A and B), and the last two writes (of C and D, together with the commit order)

(c) A read-only transaction is a transaction that only reads from the database, without writing/inserting deleting. Answer the questions below.

i. (2 points) If all transactions are read-only, then every schedule is serializable.

i. **True**

True or False?

ii. (2 points) If no transaction reads the same element twice, then the serialization level READ COMMITTED is equivalent to REPEATABLE READS.

ii. **False**

True or False?

Solution: A counterexample is: $R_1(A), W_2(B), W_2(A), Co_2, R_1(B)$. This schedule is possible under READ COMMITTED, but not under REPEATABLE READS (since the latter uses strict 2PL, which on a static database ensures conflict serializability, while this schedule is not conflict serializable).

iii. (2 points) If no transaction inserts or deletes records to/from the database, then the serialization level REPEATABLE READS is equivalent to SERIALIZABLE.

iii. **True**

True or False?

iv. (2 points) The reason why some applications use serialization levels other than SERIALIZABLE is because they would not be correct under the SERIALIZABLE isolation level.

iv. **False**

True or False?

v. (2 points) In Sqlite phantoms are not possible.

v. **True**

True or False?

vi. (2 points) The difference between Two Phase Locking and Strict Two Phase Locking is that the latter avoids deadlocks, while the former may allow deadlocks.

vi. **False**

True or False?

vii. (2 points) Only one transaction can hold a *shared lock* at any time.

vii. **False**

True or False?

viii. (2 points) Only one transaction can hold an *exclusive lock* at any time.

viii. **True**

True or False?

5 Parallel Data Processing

5. (46 points)

(a) Consider a social network database with two relations shown below:

`User(uid,name)`	`1 Million tuples`
`Follows(uid1,uid2)`	`10 Million tuples`

The table `User` contains user information, while `Follows` tells us that user1 follows user2. Suppose we are computing the following query:

```
select x.uid1,x.uid2,y.name
from Follows x, User y
where x.uid2 = y.uid
```

We use a distributed system with p servers, and compute the join using partitioned hash-join. In other words:

- The system partitions `User(uid,name)` by applying a hash function to `uid`,
- partitions `Follows` by applying a hash function to `uid2`,
- then each server computes a join of its local data.

On $p = 10$ servers, the query runs in 1000 seconds. Estimate the runtime of the system in each of the cases below, assuming the number of servers is increased as shown. Your numbers are only *estimates*: try to estimate within a factor of 2. For example, if the question were *what is the runtime on one server?* then you would answer 10,000 seconds, since one server must do the work of all the 10 servers that took 1000 seconds, although one server could run much faster than 10×1000 second.

i. (5 points) Assume that every user follows at most 5 users, and is followed by at most 5 users:

$p =$	10	100	1000	100000
Time=	1000s			

Solution:

$p =$	10	100	1000	100000
hline Time=	1000s	100s	10s	0.1s

ii. (5 points) As in item i, every user follows and is followed by at most 5 users, except for user `'JB'` who is followed by 10,000 users.

$p =$	10	100	1000	100000
Time=	1000s			

Solution: All the 10,000 `Follows` records with followers of `'JB'` will be sent to the same server. Notice that 10,000 is the average load per server when $p = 1000$, hence we do not have any speedup after that point:

$p =$	10	100	1000	100000
hline Time=	1000s	100s	20s	10.1s

Also OK: 1000,100,10,10, or numbers that are close enough.

iii. (5 points) As in item i, every user follows and is followed by at most 5 users, except for user `'JB'` who is followed by 100,000 users.

$p =$	10	100	1000	100000
Time=	1000s			

Solution: Now 100,000 records will be sent to one single server; this is the average load for $p = 100$ serers, hence there is more speedup beyond that point:

$p =$	10	100	1000	100000
hline Time=	1000s	200s	110s	100.1s

Also OK: 1000, 100, 100, 100

(b) We are running a MapReduce job over HDFS. Our input file has 10^{10} records, its size is 1TB= 10^{12}B. Hadoop's block size is configured at 100KB. Answer each of the questions below.

i. (2 points) How many map tasks will the MapReduce system create by default? If there is no default, then then indicate so.

i. 10^7

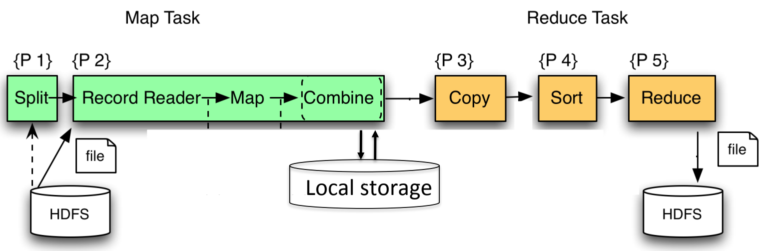
Number of map tasks:

ii. (2 points) How many reduce tasks will the MapReduce system create by default? If there is no default, then then indicate so.

ii. **No default**

Number of reduce tasks:

For the next few questions, recall the steps of a MapReduce job from the lecture notes:



iii. (2 points) The *Copy* phase of the reduce tasks may start immediately after the first map tasks finish without having to wait for all map tasks to finish.

iii. **Yes**

Yes or no?

iv. (2 points) The *Sort* phase of the reduce tasks may start immediately after the first map tasks finish without having to wait for all map tasks to finish.

iv. **No**

Yes or no?

v. (2 points) The *Reduce* phase of the reduce tasks may start immediately after the first map tasks finish without having to wait for all map tasks to finish.

v. **No**

Yes or no?

(c) (5 points) A MapReduce job runs on 100 workers and has 500 reduce tasks. At some point in time, all map tasks have finished, and 150 reduce tasks have finished too: the system is executing 100 reduce tasks, while another 250 reduce tasks are still waiting to be scheduled. At this point worker number 44 fails: the worker and its local disk are lost and not recoverable. Indicate which of the following will happen:

1. The system continues executing the 99 active tasks, then will schedule the remaining 250 tasks on the 99 remaining workers.
2. The system continues executing the 99 active tasks, then will schedule the remaining 251 tasks on the 99 remaining workers (including the reduce task that was running on worker 44).
3. The system reruns all 500 reduce tasks.
4. The system continues executing the 99 active tasks, then reruns the map tasks that had been ran on worker 44, and after that continues executing the remaining reduce tasks.
5. The system needs to restart the entire job (all map tasks and all reduce tasks) on 99 workers.

(c) ____**4**____

Your answer:

Solution: A partial credit of 2 points was given for answering item 2.

(d) The following questions compare MapReduce to Spark. For each statement indicate whether it is true or false.

i. (2 points) A program that involves iteration (such as page rank) requires the execution of several separate MapReduce jobs.

True or False?

i. **True**

ii. (2 points) A program that involves iteration (such as page rank) requires the execution of several separate Spark program.

True or False?

ii. **False**

iii. (2 points) In a MapReduce program, all intermediate results are stored on disk.

True or False?

iii. **True**

iv. (2 points) In a Spark program, all intermediate results are stored on disk.

True or False?

iv. **False**

v. (2 points) If a worker fails during the execution of a MapReduce program, then the entire program needs to be restarted.

True or False?

v. **False**

vi. (2 points) If a worker fails during the execution of a Spark program, then the entire program needs to be restarted.

True or False?

vi. **False**

vii. (2 points) MapReduce is ideally suited for OLTP applications.

True or False?

vii. **False**

viii. (2 points) Spark is ideally suited for OLTP applications.

True or False?

viii. **False**