

CSE 344 Final Examination

March 20, 2013, 8:30am - 10:20am

Name: _____

Question	Points	Score
1	40	
2	10	
3	30	
4	10	
5	55	
6	55	
Total:	200	

- This exam is open book and open notes but NO laptops or other portable devices.
- 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 and Relational Languages

1. (40 points)

An online store uses a database with the following schema:

```
Product(pid, pname, price)
Customer(cid, cname, address)
Orders(oid, cid, date, total)    cid foreign key to Customer
LineItem(lid, oid, pid, quantity)  oid, pid foreign keys to Orders, Product
```

Products and **Customer** contains a set of products and of customers respectively. **Orders** represent orders placed by customers: each order has several lines, called **LineItems**, each line representing one product that the customer ordered. Each **Orders** record also stores the total value of the order, which is the sum of $\text{price} \times \text{quantity}$ for all line items in that order.

The following constraints must be enforced by the database system:

- All keys and foreign keys.
- All attributes are not null.
- $\text{price} > 0$

In addition, the business logic is such that the **total** in each order equals the sum of $\text{price} \times \text{quantity}$ for all line items in that order. This constraint is maintained by the application logic.

```
Product(pid, pname, price)
Customer(cid, cname, address)
Orders(oid, cid, date, total)
LineItem(lid, oid, pid, quantity)
```

- (a) (10 points) Write the CREATE TABLE statements. Include all constraints that must be enforced by the database system. All identifiers (pid, cid, oid, lid) are integers; quantity is an integer; price, total are float; all other fields are text.

Answer (write SQL CREATE TABLE statements):

```
Solution: All queries were tested on postgres.
create table product(pid int primary key,
                    pname text not null,
                    price float not null
                    check (price > 0.0));
create table customer(cid int primary key,
                    cname text not null,
                    address text not null);
create table orders(oid int primary key,
                   cid int not null references customer,
                   date text not null,
                   total float not null);
create table lineitem(lid int primary key,
                    oid int not null references orders,
                    pid int not null references product,
                    quantity int);

To test answers for the remaining questions, run this:
insert into product values(1, 'gadget', 100.00);
insert into product values(2, 'gizmo', 1000.00);

insert into customer values(1, 'alice', 'nome');

insert into orders values(1, 1, 'today', 5300.00);
insert into lineitem values(1, 1, 1, 3);
insert into lineitem values(2, 1, 2, 5);

insert into orders values(2, 1, 'yesterday', 3500);
insert into lineitem values(3, 2, 1, 5);
insert into lineitem values(4, 2, 2, 3);
```

```
Product(pid, pname, price)
Customer(cid, cname, address)
Orders(oid, cid, date, total)
LineItem(lid, oid, pid, quantity)
```

- (b) (10 points) In every order, the **total** must be equal to the sum of the line item quantity times the product price. If the total is incorrect, then that order must be reviewed by a human and the error must be corrected. Write a query that finds all inconsistent orders; your query should return a set of **oid**.

Answer (write a SQL query):

Solution:

```
select x.oid
from orders x, lineitem y, product z
where x.oid = y.oid and y.pid = z.pid
group by x.oid, x.total
having x.total != sum(y.quantity * z.price);
```

```
Product(pid, pname, price)
Customer(cid, cname, address)
Orders(oid, cid, date, total)
LineItem(lid, oid, pid, quantity)
```

- (c) (10 points) The company's *revenue* is the sum of all *totals*, from all orders. In this question you will write a *what-if* query: without updating the database, your query will compute how much the revenue would change, had the company changed some prices:

Write a query that computes how much the company's revenue would increase, if they raised the price of the 'gizmo' product by 10% (`pname = 'gizmo'`).

Answer Write a SQL query that returns one number (the revenue increase):

Solution:

```
select sum(y.quantity * 0.1 * z.price)
from lineitem y, product z
where y.pid = z.pid and z.pname='gizmo';
```

(d) For the queries below we will consider a simplified schema:

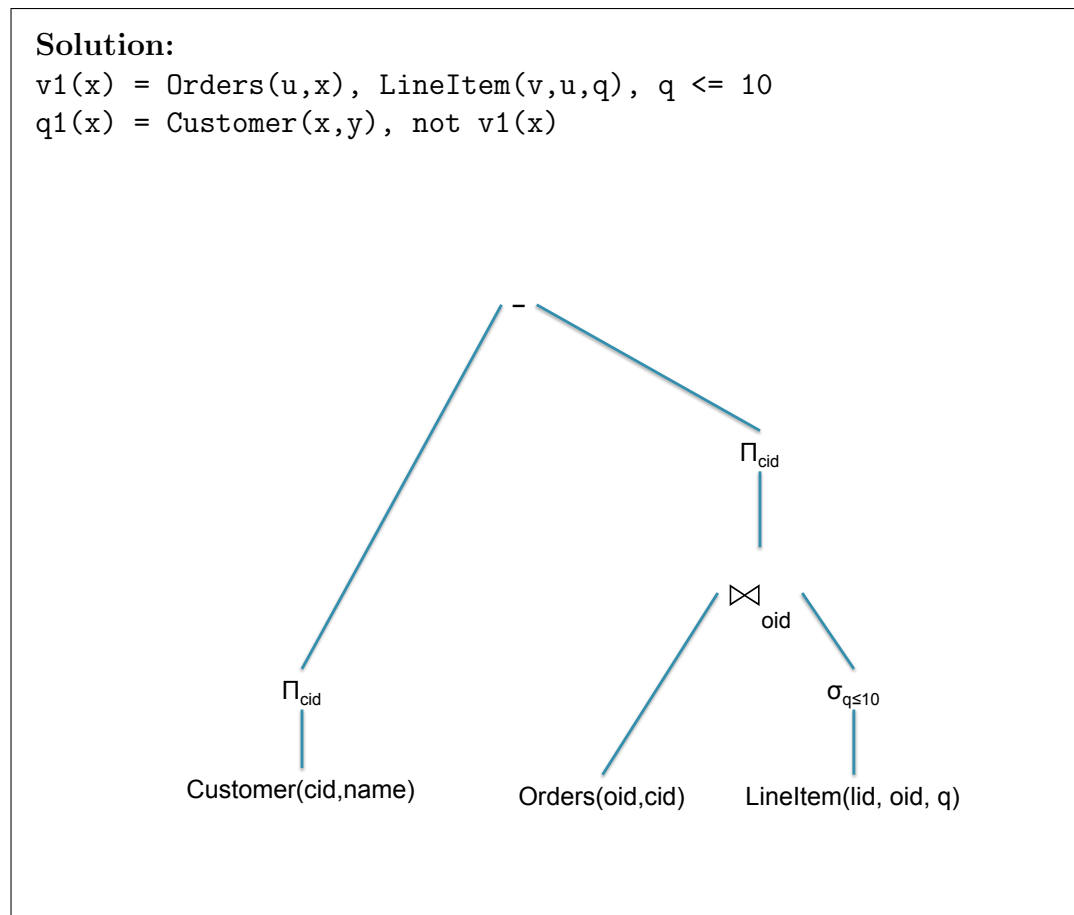
```
Customer(cid, name)
Orders(oid, cid)
LineItem(lid, oid, quantity)
```

For each of the three Relational Calculus expressions below, give an equivalent datalog query. In addition, give a relational algebra plan for the first expression only.

- i. (4 points) Find customers for which all their orders and all line items on those orders are for products with price > 10 :

$$q_1(x) = \exists y [\text{Customer}(x, y) \wedge \forall u \forall v \forall q (\text{Orders}(u, x) \wedge \text{LineItem}(v, u, q) \Rightarrow q > 10)]$$

Answer write a datalog query, AND draw a query plan:



- ii. (3 points) Find customers that placed some order where all line items are for products with price > 10 :

$$q_2(x) = \exists y \exists u [\text{Customer}(x, y) \wedge \text{Orders}(u, x) \wedge \forall v \forall q (\text{LineItem}(v, u, q) \Rightarrow q > 10)]$$

Answer write a datalog query (do NOT draw the query plan):

Solution:

$v_2(u) = \text{LineItem}(v, u, q), q \leq 10$

$q_2(x) = \text{Customer}(x, y), \text{Orders}(u, x), \text{not } v_2(u)$

- iii. (3 points) Find customers for which all their orders have some line item for a product with price > 10 :

$$q_3(x) = \exists y [\text{Customer}(x, y) \wedge \forall u (\text{Orders}(u, x) \Rightarrow \exists v \exists q (\text{LineItem}(v, u, q) \wedge q > 10))]$$

Answer write a datalog query (do NOT draw the query plan):

Solution:

$v_{32}(u) = \text{LineItem}(v, u, q), q > 10$

$v_{31}(x) = \text{Orders}(u, x), \text{not } v_{32}(u)$

$q_3(x) = \text{Customer}(x, y), \text{not } v_{31}(x)$

2 XML

2. (10 points)

Consider the following XML document:

```

<node>
  <node>
    <node> 1 </node>
    <node> 2 </node>
  </node>
  <node>
    <node> 3 </node>
    <node> 4 </node>
  </node>
  <node>
    <node> 2 </node>
    <node> 3 </node>
  </node>
  <node>
    <node> 4 </node>
    <node> 5 </node>
  </node>
</node>

```

Give the answer to each of the XPath expressions below.

(a) (2 points) `/node/node[3]/node/text()`

(a) 2 3

Answer:

(b) (3 points) `/node/node[node/text()=3]/node/text()`

(b) 3 4 2 3

Answer:

(c) (2 points) `/node/node[node/text()=3][node/text()=4]/node/text()`

(c) 3 4

Answer:

(d) (3 points) `/node[node/node/text()=3][node/node/text()=4]/node/node/text()`

(d) 1 2 3 4 2 3 4 5

Answer:

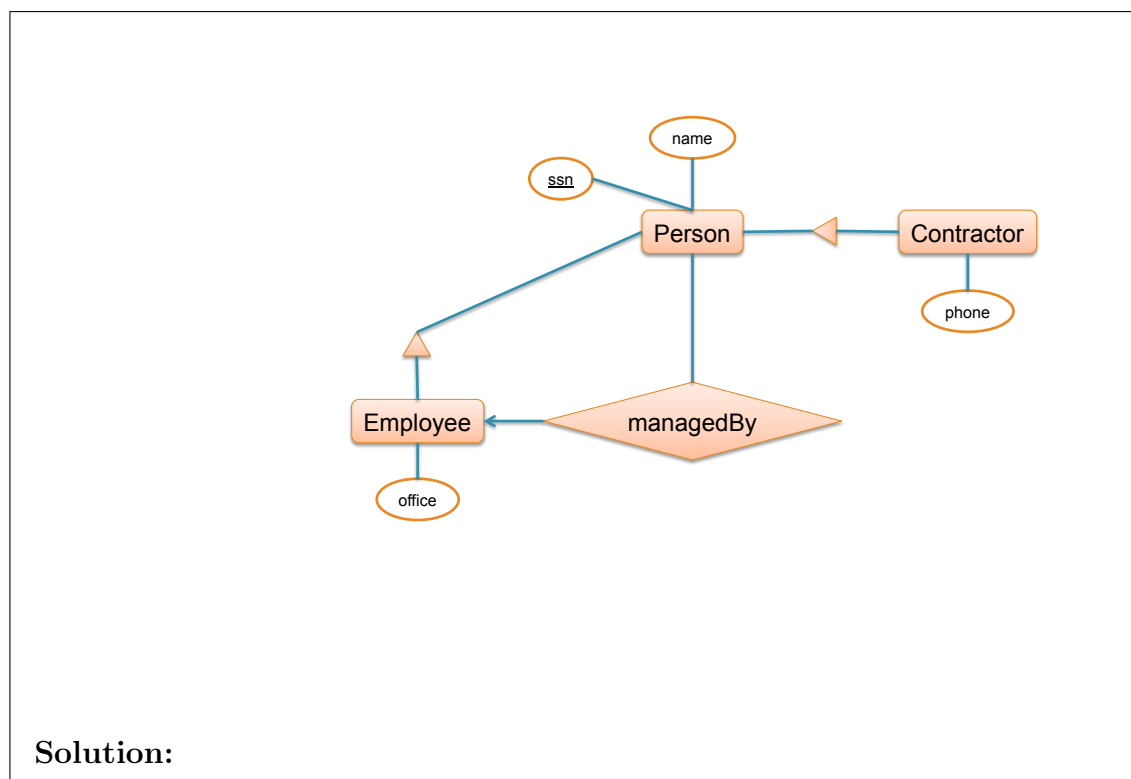
3 E/R Diagrams, Constraints, Conceptual Design

3. (30 points)

(a) (10 points) Design an E/R diagram for a database of employees about the following:

- Employee: they have ssn, name, office
- Contractor: they have ssn, name, phone
- ManagedBy: every Employee and every Contractor has a manager, who must be an Employee.

Answer Draw an E/R diagram; indicate all keys by underlining; mark all many-one relationships; show ISA relationships.



(b) Consider the following relational schemas and sets of functional dependencies:

$$\begin{array}{ll} R(A, B, C, D) : & AB \rightarrow D, \quad AD \rightarrow C \\ S(A, B, C, D) : & C \rightarrow A, \quad D \rightarrow B \end{array}$$

- i. (4 points) Show the keys for each relation. List all keys if there are more than one.

Answer Show the key(s) in R and the key(s) in S :

Solution:

The key in R is: AB .

The key in S is: CD .

- ii. (3 points) Consider the following two views:

$$\begin{array}{l} V_1(A, B, C, D) = \sigma_{B='99'}(R(A, B, C, D)) \\ V_2(A, B, C) = \Pi_{ABC}(R(A, B, C, D)) \end{array}$$

Show the keys in each view. List all keys if there are more than one.

Answer Show the key(s) in V_1 and the key(s) in V_2 :

Solution:

The key in V_1 is: A .

The key in V_2 is: AB . (The FD in V_2 is $AB \rightarrow C$.)

- iii. (3 points) Consider the following view:

$$V_3(A, B, C, D) = R(A, B, C, D) \cap S(A, B, C, D)$$

Compute the closure of BC in V_3 .

Answer Show $(BC)^+$.

Solution: $ABCD$

- (c) (10 points) Consider the following relational schema and set of functional dependencies.

$R(A,B,C,D,E)$ with functional dependencies:

$$\begin{aligned} A &\rightarrow B \\ C &\rightarrow B \\ DE &\rightarrow C \end{aligned}$$

Decompose R into BCNF. Show your work for partial credit. Your answer should consist of a list of table names and attributes and an indication of the keys in each table (underlined attributes).

Answer (Decompose R into BCNF):

Solution:

Solution 1 Start with the set A : $A^+ = AB$, decompose into \underline{AB} and $ACDE$.
In the latter use $DE^+ = CDE$ and decompose into \underline{CDE} and \underline{ADE} .

Final answer: $R_1(\underline{A}, B), R_2(C, \underline{D}, \underline{E}), R_3(\underline{A}, \underline{D}, \underline{E})$.

Solution 2 Start with the set C : $C^+ = BC$, decompose into \underline{BC} and $ACDE$.
The latter decomposes as above.

Final answer: $R_1(B, \underline{C}), R_2(C, \underline{D}, \underline{E}), R_3(\underline{A}, \underline{D}, \underline{E})$.

Solution 3 Start with the set DE : $DE^+ = BCDE$, decompose into \underline{BCDE} and ADE . In the former choose the set C : $C^+ = BC$ hence it decomposes into \underline{BC} and CDE .

Final answer: $R_1(B, \underline{C}), R_2(C, \underline{D}, \underline{E}), R_3(\underline{A}, \underline{D}, \underline{E})$ (same as Solution 2).

4 Views

4. (10 points)

We have a table `Follows(person1, person2)` storing followers in a social network. A data analyst issues the following commands:

```
CREATE VIEW V1 AS
  SELECT x.person1 AS person1, y.person2 AS person2
  FROM Follows x, Follows y
  WHERE x.person2=y.person1;
```

```
CREATE VIEW V2 AS
  SELECT x.person1 AS person1, y.person2 AS person2
  FROM V1 x, V1 y
  WHERE x.person2=y.person1;
```

```
CREATE VIEW V3 AS
  SELECT x.person1 AS person1, y.person2 AS person2
  FROM V2 x, V2 y
  WHERE x.person2=y.person1;
```

```
CREATE VIEW V4 AS
  SELECT x.person1 AS person1, y.person2 AS person2
  FROM V3 x, V3 y
  WHERE x.person2=y.person1;
```

```
SELECT x.person1 AS person1, y.person2 AS person2
FROM V4 x, V4 y
WHERE x.person2=y.person1;
```

How many join operations will the SQL engine perform in response to these five commands (four view definitions and one query), in each of the two cases below:

(a) (5 points) All views are virtual.

(a) 31

The total number of joins is:

(b) (5 points) All views are materialized.

(b) 5

The total number of joins is:

Solution: 31 joins and 5 joins respectively. Database systems do not use DAGs in query plans. To check this in postgres, run the following:

```
create table follows(person1 int, person2 int);

insert into follows values (0,1);
insert into follows values (1,2);
insert into follows values (2,3);
insert into follows values (3,4);
insert into follows values (4,5);
insert into follows values (5,6);
insert into follows values (6,7);
insert into follows values (7,8);
insert into follows values (8,9);
insert into follows values (9,10);
insert into follows values (10,11);
insert into follows values (11,12);
insert into follows values (12,13);
insert into follows values (13,14);
insert into follows values (14,15);
insert into follows values (15,16);
insert into follows values (16,17);
insert into follows values (17,18);
insert into follows values (18,19);
insert into follows values (19,20);
insert into follows values (20,21);
insert into follows values (21,22);
insert into follows values (22,23);
insert into follows values (23,24);
insert into follows values (24,25);
insert into follows values (25,26);
insert into follows values (26,27);
insert into follows values (27,28);
insert into follows values (28,29);
insert into follows values (29,30);
insert into follows values (30,31);
insert into follows values (31,32);

-- create the four views above

\o outputfile.txt

explain SELECT x.person1 AS person1, y.person2 AS person2 FROM V4 x, V4 y WHERE x.person2=y.person1;

At command prompt:
more outputfile.txt | grep Join | wc
```

5 Transactions

5. (55 points)

(a) Consider the following schedule:

T1	T2
BEGIN	
READ(A)	
$A = A + 50$	
WRITE(A)	
	BEGIN
	READ(A)
	$A = A + 200$
	WRITE(A)
	READ(B)
	$B = 3 * B$
	WRITE(B)
	COMMIT
READ(B)	
$B = 12 * B$	
WRITE(B)	
COMMIT	

Which of the following statements are true? Check all that apply.

i. (2 points) The schedule is serial.

i. **No**

Yes or no:

ii. (2 points) The schedule is serializable.

ii. **Yes**

Yes or no:

iii. (2 points) The schedule is conflict serializable.

iii. **No**

Yes or no:

iv. (2 points) This schedule is possible under SQL Lite. (Recall that SQL Lite uses a lock-based concurrency control manager, with a single lock for the entire database.)

iv. **No**

Yes or no:

v. (2 points) This schedule is possible under SQL Server. (Recall that SQL Server uses a lock-based concurrency control manager, with one lock for each database element.)

v. **No**

Yes or no:

- (b) The initial value of the database element A is $A = 0$. The following four transactions are started at approximately the same time:

T0	T1	T2	T3
BEGIN	BEGIN	BEGIN	BEGIN
READ(A)	A=1	A=10	A=100
$X = A$	WRITE(A)	WRITE(A)	WRITE(A)
READ(A)	ABORT	COMMIT	COMMIT
$Y = A$			
$B = X + Y$			
WRITE(B)			
COMMIT			

Both X and Y could be 0, 1, 10 or 100, hence B has 10 ($= \frac{5 \cdot 4}{2}$) possible values. Indicate what values are possible for B at the end of transaction T0. *Circle* the B values that are possible, and/or *cross out* those that are impossible. (The tables show only $B = X + Y$ where $X \geq Y$, since $X + Y = Y + X$.)

- i. (5 points) The isolation level is set to **READ UNCOMMITTED**

$X = 0$	$X = 1$	$X = 10$	$X = 100$	$B = X + Y$
$B = 0$	$B = 1$	$B = 10$	$B = 100$	$Y = 0$
	$B = 2$	$B = 11$	$B = 101$	$Y = 1$
		$B = 20$	$B = 110$	$Y = 10$
			$B = 200$	$Y = 100$

Solution: All values are possible.

- ii. (5 points) The isolation level is set to **READ COMMITTED**

$X = 0$	$X = 1$	$X = 10$	$X = 100$	$B = X + Y$
$B = 0$	$B = 1$	$B = 10$	$B = 100$	$Y = 0$
	$B = 2$	$B = 11$	$B = 101$	$Y = 1$
		$B = 20$	$B = 110$	$Y = 10$
			$B = 200$	$Y = 100$

Solution: The value $A = 1$ is never read. Hence, the possible outputs are 0, 10, 100, 20, 110, 200

- iii. (5 points) The isolation level is set to **REPEATABLE READ**

$X = 0$	$X = 1$	$X = 10$	$X = 100$	$B = X + Y$
$B = 0$	$B = 1$	$B = 10$	$B = 100$	$Y = 0$
	$B = 2$	$B = 11$	$B = 101$	$Y = 1$
		$B = 20$	$B = 110$	$Y = 10$
			$B = 200$	$Y = 100$

Solution: Possible values are 0, 20, 200.

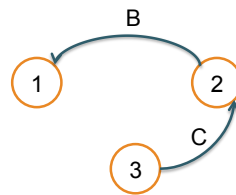
(c)

Consider the following transaction schedules. For each schedule, indicate if it is **conflict-serializable** or not. In each case show: the precedence graph; a cycle (if one exists) or an equivalent serial schedule (if one exists).

i. (5 points)

$r_1(A)$, $w_1(A)$, $r_2(B)$, $w_2(B)$, $r_3(C)$, $w_3(C)$, $r_2(C)$, $w_2(C)$, $r_1(B)$, $w_1(B)$, c_1 , c_2 , c_3

Answer (Draw the precedence graph and determine if it is serializable):



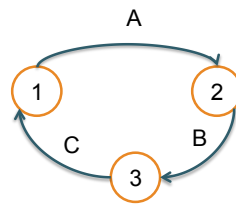
Solution:

Serializable. The serialization order is: T3, T2, T1.

ii. (5 points)

$r_1(A)$, $r_2(B)$, $r_3(C)$, $w_1(C)$, $w_2(A)$, $w_3(B)$, c_1 , c_2 , c_3

Answer (Draw the precedence graph and determine if it is serializable):

**Solution:**

Not serializable, because of the cycle $T1 \rightarrow T2 \rightarrow T3 \rightarrow T1$.

- iii. (5 points) Give an example of a schedule with two or more transaction with the following three properties:
- T1 commits before T2 starts.
 - The schedule is conflict serializable.
 - In any equivalent serial schedule, T2 must come before T1 (there may be other transactions between T2 and T1).

Solution:

$s_3, w_3(A), s_1, w_1(A), c_1, s_2, w_2(B), c_2, w_3(B), c_3$

T1 commits before T2 starts, yet the only serialization order is T2, T3, T1.

- (d) A database has 1000 elements, denote them A_1, \dots, A_{1000} , and needs to run a workload of transactions of the following kind:

```

BEGIN
READ( $A_i$ )
perform a non-parallelizable, CPU-intensive computation for 0.1seconds
WRITE( $A_i$ )
COMMIT

```

Each transaction reads one element, performs some intensive computation, then writes the same element back. Different transactions may update the same or different elements, for example five transactions may update elements A_3, A_1, A_3, A_3, A_2 . The system is shared-memory, has 100 CPUs, and uses inter-query parallelism (multiple transactions are run in parallel, but each transaction runs on a single CPU). For serializability, the system uses one lock per element (like SQL Server).

How many transactions per second can the system execute in each case below?

- i. (5 points) All transactions update element A_1 ; no other elements are read or updated. That is, the transactions update elements in this sequence: A_1, A_1, A_1, \dots
Answer Write the number of transactions per second (TPS):

Solution: 10 TPS. The transactions are executed serially.

- ii. (5 points) The transactions update only elements from the first 50 elements. More precisely, each transaction reads/writes an element A_i chosen uniformly at random from among A_1, \dots, A_{50} . For example, the transactions may update the elements in this sequence: $A_{44}, A_2, A_{13}, A_2, A_{36}, A_{11}, \dots$
Answer Write the number of transactions per second (TPS):

Solution: 500 TPS. For example, suppose the elements are updated round-robin: $A_1, A_2, \dots, A_{50}, A_1, A_2, \dots$ T51 can start only after T1 terminates, which is 0.1s after T1 starts. We run 50 transactions every 0.1s, or 500 TPS. We can sustain this rate because there are 100 CPUs and each can sustain 10 TPS max.

- iii. (5 points) The transactions update all elements. More precisely, each transaction reads/writes an element A_i , chosen uniformly at random from among A_1, \dots, A_{1000} .
Answer Write the number of transactions per second (TPS):

Solution: 1000 TPS, because the system becomes CPU bound. The max rate per CPU is 10 TPS and there are 100 CPUs.

6 Parallel Data Processing

6. (55 points)

- (a) There is a new search engine in town! To answer keyword queries, they compute an *inverted index*, with the following schema:

```
Inv(word, url, occ)
```

Each record consists of a word, a URL where that word occurs, and the number of occurrences of that word at that URL. When a user issues a keyword search, say she searches for `xyz`, then the search engine runs a query like this:

```
SELECT url FROM Inv WHERE word = 'xyz'
ORDER BY occ DESC
```

`Inv` has **1 Terabyte** ($= 10^{12}$ bytes).

The company needs to support **10,000 queries per second**.

They plan to use a cluster of servers, each server running an independent SQL Lite database, on a fragment of `Inv`. You are charged with managing their server cluster and databases.

You measure the performance of SQL Lite on a single server: on a subset of `Inv` and with an index on `Inv.word` SQL Lite answers about **10 queries per second**. This performance is largely independent on the size of the subset of `Inv` (this, of course, is due to the index); it is also independent on whether the queries return an empty or a nonempty answer.

- i. (5 points) How many servers P do you purchase in order to answer 10,000 queries per second on the 1 Terabyte database?

$P =$

i. 1000

```
Inv(word, url, occ)
SELECT url FROM Inv WHERE word = 'xyz'
ORDER BY occ DESC
```

- ii. (10 points) Next, you need to decide how to partition the table `Inv`. Assuming the value P you determined at the previous question, indicate how many queries per second your system can support if:

- `Inv` is block partition.

ii. 10

The number of queries per second is:

- `Inv` is hash partition on `word`.

ii. 10,000

The number of queries per second is:

- `Inv` is hash partition on `url`.

ii. 10

The number of queries per second is:

Lecture 27: counting number of words:

```
map(String key, String value):
for each word w in value:
EmitIntermediate(w, 1);
```

```
reduce(String key, Iterator values):
// key: a word, values: a list of counts
int result = 0;
for each v in values:
result += ParseInt(v);
Emit(AsString(result));
```

```
WebCrawl(url, fullText):
Inv(word, url, occ)
```

- (b) (20 points) Every week or so the company needs to recompute the entire inverted index, from a fresh crawl of the Web. The crawl is stored in a table with the following schema:

```
WebCrawl(url, fullText)
```

To generate the inverted index, you use a MapReduce job. Fill in the details of the Map and the Reduce functions below. Your answer may consist of pseudocode (they don't need to be perfect Java). Write your answers in the boxes provided:

```
function map(String url, String fullText):
for each word w in fullText:
```

```

/* EmitIntermediate(...) */
```

```
function reduce(String , Iterator ):
/* fill in the parameters */
```

```

/* compute word, url, occ */

/* Emit(word, url, occ) */
```

Solution:

Solution 1:

The intermediate key is a word, and its values are all the URLs where it occurs.

```
function map(String url, String fullText)
  for each word w in fullText
    EmitIntermediary(w, url)

function reduce(String w, Iterator urls)
  hashTable = empty
  /* group by url and count */
  for each url in urls
    if hashTable[url] == NULL
      hashTable[url] = 1
    else hashTable[url] = hashTable[url]+1
  for each url in hashTable
    Emit(w, url, hashTable[url])
```

Solution 2:

The intermediate key is a (word, url) pair, and the values are 1, 1, 1, ...

```
function map(String url, String fullText)
  for each word w in fullText
    EmitIntermediary(concat(w,',',url), 1)

function reduce(String wu, Iterator values)
  c = 0
  for each v in values
    c = c + ParseInt(v)
  Emit(wu[0], wu[1], c)
```

(c) Assume the following about your MapReduce job:

- The input (`WebCrawl`) has 1TB ($= 10^{12}$).
- The chunk size is 100MB ($= 10^8$).
- The number of servers is 1000.
- The number of reduce tasks is 10,000 ($= 10^4$).

Answer the following questions:

i. (5 points) How many map tasks are in your job?

i. $10^4 = 10,000$

Number of map tasks:

ii. (5 points) How many intermediate files are created?

ii. $10^8 = 100,000,000$

Number of intermediate files:

iii. (10 points) Both the map tasks and the reduce tasks are scheduled on the same 1000 servers. Each server is single-core, and runs only one map task or only one reduce task at any time. All map tasks and all reduce tasks are perfectly uniform, i.e. tasks that start at the same time will finish at the same time.

After the first 2,000 reduce tasks have completed, server number 333 crashes and never recovers. Which of the following happens? Circle exactly one:

1. The MR system continues running on 999 servers, without taking any other actions.
2. The MR system continues with 999 servers, as follows. All 999 reducers currently running will continue to run, and the system will reschedule reduce task number 2333 on a different server at a later time.
3. The MR system aborts all active reduce tasks currently running *that have not finished the copy phase*; next, it re-starts (on the remaining 999 servers) all map tasks that had been executed on server 333; finally, it continues to schedule all remaining reduce tasks (≤ 8000) on the 999 servers.
4. The MR system aborts all active reduce tasks currently running *that have not finished the sort phase*; next, it re-starts (on the remaining 999 servers) all map tasks that had been executed on server 333; finally, it continues to schedule all remaining reduce tasks (≤ 8000) on the 999 servers.
5. The MR system aborts *all* 999 reduce tasks currently running; next, it re-starts (on the remaining 999 servers) all map tasks that had been executed on server 333; finally, it continues to schedule all remaining 8000 reduce tasks on the 999 servers.
6. The MR system aborts the entire job, because it cannot recover in this case.
7. None of the above.

Solution: item 3