Final Exam

Prof. Yanlei Diao

March 21, 2016

NAME: _____

This is a **take-home** examination. You have 24 hours to answer as many questions as possible. Partial credit will be given. There are 100 points in total.

Please type your answers clearly and submit a **pdf** document through Moodle.

GOOD LUCK!

Problem	**Maximum**	**Score**
1. Relational DBMS	**10**	
2. Query Processing and Optimization	**30**	
3. Answering Queries using Views	**10**	
4. Transaction Management	**16**	
5. OLTP/OLAP and Parallel DBMSs	**34**	
TOTAL	**100**	

Question 1 [10 points]: Relational DBMS

(1) [4 Points] Please state **the year** that the relational model was originally proposed and explain **three major features** of relational DBMSs that distinguish them from earlier data management systems, such as file systems and navigational systems used in the 60's.

Please limit your writing to one paragraph per feature.

(2) [6 Points] Consider the following relational schema (with primary keys underlined):

> **STUDENTS** (*sid*, *s_name*, *street*, *city*, *age*)
> **COURSES** (*cid*, *c_name*, *prof_name*)
> **REGISTERED** (*sid*, *cid*, *credits*)

where *sid* and *cid* are foreign keys referencing STUDENTS and COURSES respectively.

Write the following query in both **relational algebra** and **SQL**:

> "Find the names of students who have taken all the courses taught by Prof. "Franklin".

Question 2 [4 parts, 24 points total] Query Processing and Optimization

Consider the following relational schema (with primary keys underlined):

> **STUDENTS** (*sid*, *s_name*, *street*, *city*, *age*)
> **COURSES** (*cid*, *c_name*, *prof_name*)
> **REGISTERED** (*sid*, *cid*, *credits*)

where *sid* and *cid* are foreign keys referencing STUDENTS and COURSES respectively. These relations have the following characteristics:

	STUDENTS	**COURSES**	**REGISTERED**
Number of tuples	10,000	500	40,000
Number of tuples per page	20	40	50
Number of pages	500	13	800

and the following indexes:

- A **clustered B+Tree** index (Alternative 2) is built on **STUDENTS**.*city*, with 200 distinct values of *city*.
- An **unclustered B+Tree** index (Alternative 2) is on **STUDENTS**.*age*, with 50 distinct values of *age*.

Assume that the height of each index is, **H=2**, and the number of leaf nodes is, **L=250**.

(1) [9 points] Consider the following query.

```
Query 1:   SELECT *
           FROM STUDENTS
           WHERE city = 'Amherst' and age = 25;
```

Please name 3 available access methods for this query, analyze their I/O costs, and state the most efficient method. Please mark any assumption you make, e.g., uniform distribution of values in an attribute.

	STUDENTS	COURSES	REGISTERED
Number of tuples	10,000	500	40,000
Number of tuples per page	20	40	50
Number of pages	500	13	800

(2) [9 points] Now consider the natural join:

$\rhd\lhd$ STUDENTS $_{sid}$ REGISTERED

Given available memory of 32 buffer pages, state the I/O costs of **block nested loops join**, **sort-merge join**, and **hash join**. Please ignore the I/O cost of producing the join output.

(3) [4 points] Consider the following query.

```
Query 2:   SELECT *
           FROM STUDENTS S, REGISTERED R, COURSES C
           WHERE S.sid = R.sid and R.cid = C.cid and C.cid = "445";
```

Draw all **left-deep query plans** that the query optimizer considers for Query 2. Make sure to avoid cross products between two relations before exhausting the join predicates, and place the selection in the most effective place.

(4) [8 points] Consider Query 3.

```
Query 3:   SELECT     *
           FROM       COURSES C1, REGISTERED R1, STUDENTS S,
                      REGISTERED R2, COURSES C2
           WHERE      C1.cname = 'DB' and C2.cname = 'OS' and
                      S.sid = R1.sid and R1.cid = C1.cid and
                      S.sid = R2.sid and R2.cid = C2.cid;
```

Assume that the query optimizer decides to push the selections to the file scans of C1 and C2 relations. Then it uses dynamic programming (System R style) to enumerate ***k*-relation subplans** for the query (k=2, 3, 4, 5). How many **plans** are considered in passes *k*=2, *k*=3, *k*=4, and *k*=5, respectively? There is no need to consider particular join methods for this question.

Question 3 [10 points]: Answering Queries using Views

Let us consider the following schema, inspired from the one used in Halevy 2001[1].

Prof (name, area)
Course (c-number, title)
Teaches (prof, c-number, quarter)
Registered (student, c-number, quarter)
Advices (prof, student).

We assume that professors and students are uniquely identified by their names, and courses are uniquely identified by their numbers. The Registered relation describes the students' registration in classes.

Let Q be the following query:

```
Select Advises.prof, Advises.student, Registered.quarter
From   Registered, Teaches, Advises
Where
       Registered.c-number=Teaches.c-number and
       Advises.prof=Teaches.prof and
       Advises.student=Registered.student and
       Registered.quarter ≥ "winter98"
```

and let us consider the following view V:

```
Create View V as
Select  Registered.student, Teaches.prof, Registered.quarter
From    Registered, Teaches
Where   Registered.c-number=Teaches.c-number and
        Registered.quarter ≥ "winter98"
```

(1) [4 Points] Is V usable to answer Q? Either rewrite Q using V or explain why it is not possible.

(2) [6 Points] We are now looking for professor-student pairs such that the student is advised by the professor and has followed the course 101 in Summer 99. This query is called Q'.

(a) V is not usable for equivalently rewriting this query. Explain why.

(b) Modify Q' (creating Q") by adding a constraint on the professor and the course so that V is usable to answer Q".

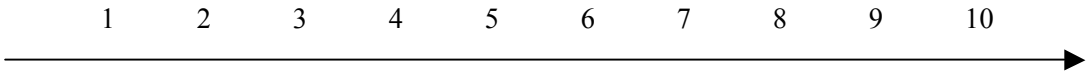
(c) Assume that there are hash indexes on Registered.c-number and (Teaches.c-number,Teaches.quarter). Discuss the usefulness of V to answer Q".

---

[1] A.Y. Halevy: Answering queries using views: A survey. VLDB J. 10(4): 270-294 (2001)

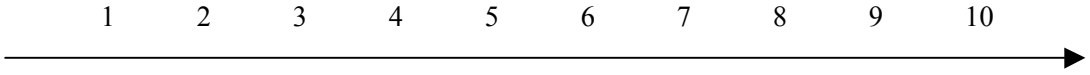
Question 4 [3 parts, 16 points total] Transaction Management

Consider three transactions T1, T2 and T3 and their actions submitted to the lock manager:



T1:	R(A)		W(B)						(Commit)
T2:		R(B)		W(C)					(Commit)
T3:					R(C)	W(A)	R(D)	W(D)	(Commit)

(1) [4 Points] Draw the precedence graph for this schedule. Is this schedule conflict serializable? Why or why not? If it is conflict serializable, give the equivalent serial schedule (just write the order of the transactions in the equivalent serial schedule).



T1:	R(A)		W(B)					(Commit)	
T2:		R(B)		W(C)				(Commit)	
T3:					R(C)	W(A)		(Commit)	

(2) [6 Points] Explain how this sequence of actions will be executed using the **Strict Two Phase Locking** protocol (2PL). The following table is designed for you to specify in time order the activities that take place in the lock manager and in access to the database. The activities include:

- S(O) or X(O): *the lock requested for access to a database object, O*;
- R(O) or W(O): *the read or write access to a database object, O*;
- Ti blocked: *which transaction may be blocked under this protocol*;
- Ti resumed: *until when a blocked transaction can resume, with its lock request granted*; *and*
- Ti commits and releases locks: *when a transaction can commit and release locks*.

The first few actions are already given below. Please specify others using the same format.

Note that the number of rows in the table does not indicate the length of the correct answer. Please feel free to use as many as rows as needed.

Time stamp	**Action:** please choose one action at a time
1	T1: S(A)
2	T1: R(A)
3	T2: S(B)
4	
5	
6	
7	
8	

9	
10	
11	
12	
13	
14	
15	
16	
17	
18	
19	
20	
21	
22	

(3) [6 Points] This question continues from Question (2): For the sequence of actions actually executed under Strict 2PL, let us assume the database system crashed right after the last write action (according to your schedule) before its transaction gets a chance to commit.

When the system reboots and goes through recovery, please shows the entire **history of log records** including those before the crash and those generated during recovery. Please specify the log history using the following format.

LSN	**Xact ID**	**Type of Record** (Update/Commit/Abort/CLR/End)	**Data Object Updated**
1	T2	Update	C
2			
3			
4			
5			
6			
7			
8			
9			
10			
11			
12			
13			
14			
15			

Question 5 [34 points]: OLTP/OLAP and Parallel DBMSs

Consider a social network with 100's of millions of users. The database for the social network needs to store:

(a) user profiles,
(b) friend information,
(c) postings by each user including text, photos, and videos,
(d) messages exchanged among friends, and
(e) page views (clicks) by the users.

The data volume is about 100 terabytes of new data per day, and 100 petabytes in total.

Please address the following questions:

(1) [10 Points] What is your design of the **relational schema** of the database to store all the information described above? Please specify your schema using the CREATE TABLE command. Make sure to list the primary key and foreign key constraints.

(2) [5 Points] The typical workloads in the database include:

- updates to the tables containing items (a)-(d), and reflecting these changes immediately in user views of their friends' activities.
- inserts to the table containing item (e), and using it together with other relevant information, e.g., the user profiles, to determine recommendation and ads for each user.

Please analyze the workloads seen by the database. In particular, are they **OLTP** workloads, **OLAP** workloads, or a mix of both? Please briefly explain your answer.

(3) [6 Points] Which **DBMS** or a mix of DBMS's (among OLTP, OLAP, parallel systems, MapReduce systems) would you choose to implement this database? Furthermore, name **three techniques** among those discussed in class are particularly useful to your database design? Please briefly explain your answer.

(4) [3 Points] Is the SQL query plan produced by the System R-style optimizer optimal for parallel processing? If your answer is no, please give a concrete example to illustrate your point, or explain a missed opportunity in the R-style optimizer that would be helpful in parallel processing.

(5) [10 Points] Suppose that an analytic query has broken the clicks of each user into sessions. The resulting dataset has the format,

```
<user_id, session_id, {list of page_id's in the session}>,
```

where user_id and session_id jointly form the primary key of the dataset. Now we want to perform a task:

> "Find all subsets of page_id's that jointly occur in more than 1 mission user sessions."

Please **design a parallel algorithm** for execution in the cluster. Please use necessary diagrams and text to explain your parallel algorithm. Limit your writing, with diagrams and text, to at most two pages.

Note that this is probably the hardest, open-ended question in the exam. If you find it difficult, please focus on other questions.