CSE 414 Final Exam
Autumn 2018
December 13, 2018

- Please read all instructions (including these) carefully.
- **This is a closed-book exam. You are allowed two pages of note sheets that you can write on both sides.**
- Write your name and UW student number below.
- No electronic devices are allowed, including **cell phones** used merely as watches.
- Solutions will be graded on correctness and ***clarity***. Each problem has a relatively simple and straightforward solution. Partial solutions will be graded for partial credit.
- There are 12 pages in this exam, including this one.
- There are 6 questions, each with multiple parts. If you get stuck on a question move on and come back to it later.
- You have 110 minutes to work on the exam.
- Please write your answers in the space provided on the exam, and clearly mark your solutions. You may use the blank pages as scratch paper. **Do not** use any additional scratch paper.

Relational algebra operators:

Union ∪ Difference — Selection σ Projection π Join ⋈
Rename ρ Duplicate elimination δ Grouping and aggregation ɣ Sorting τ

By writing your name below, you certify that you have not received any unpermitted aid for this exam, and that you will not disclose the contents of the exam to anyone in the class who has not taken it.

NAME: _____

STUDENT NUMBER: _____

Problem	Points	Problem	Points
1	20	4	16
2	21	5	18
3	30	6	20
Total		**125**	

Problem 1: Warm Up (20 points total)

Select either True or False for each of the following questions. For each question you get 2 points for answering it correctly. There is no penalty for an incorrect answer.

a) When turning an E/R diagram into a relational schema, every entity and relation in the diagram is represented by a table.

True False

b) In schema design, a key is a superkey with the minimal number of attributes among all superkeys for a given table.

True False

c) An unclustered index can never speed up a query more than a clustered index on the same attributes, assuming only one index on the table.

True False

d) In database concurrency, "isolation" means that no matter the actual schedule executed by the system, the effect should be the same as if the transactions run one after the other.

True False

e) Strict 2PL (assuming row updates but not insert/deletes) guarantees conflict serializable schedules but not durability to crashes.

True False

f) Horizontal partitioning means different columns from the same table are stored on different machines.

True False

g) In query execution, operators that run "on-the-fly" do not look up tuples with disk reads.

True False

h) For a given SQL query, query optimizers consider multiple logical plans and multiple physical plans.

True False

i) The in-memory processing of Spark solves the problem MapReduce had with straggler nodes taking longer to complete processing than other nodes.

True False

j) Under the relational model all tables must be in first normal form.

True False

Problem 2: Transactions (21 points total)

a) Draw the precedence graph of the following schedule. Is this schedule conflict serializable? If so, write YES and the order in which the transactions would have to run in the equivalent serial schedule. Otherwise write NO. (6 points)

$R_1(A);\ R_3(B);\ W_1(A);\ R_2(A);\ R_1(B);\ R_3(A);\ W_1(B);\ W_2(A);\ R_3(A);$

b) Draw the precedence graph of the following schedule. Is this schedule conflict serializable? If so, write YES and the order in which the transactions would have to run in the equivalent serial schedule. Otherwise write NO. (6 points)

$R_1(B);\ W_2(A);\ R_1(A);\ R_2(B);\ W_1(B);\ W_3(B);\ W_1(A);\ W_3(B);\ R_3(A);$

c) Read the schedule below and answer the following questions. A and B are each a field of a tuple in the database. You can assume writes to the database are only updates, not insert/deletes.

Time	Transaction 1	Transaction 2
1		Begin transaction
2	Begin transaction	
3	Lock(A)	
4		Lock(B)
5	Read(A)	
6		Read(B)
7		Unlock(B)
8	Lock(B)	
9	Unlock(A)	
10	Read(B)	
11	Write(B)	
12	Unlock(B)	
13	Commit	
14		Commit

This schedule follows the rules of two-phase locking. (3 points)

True False

This schedule follows the rules of <u>strict</u> two-phase locking. (3 points)

True False

This schedule is serializable. (3 points)

True False

Problem 3: Schema Design (30 points total)

Consider this schema for a database of movies. (There are some extra attributes added since the midterm.) The primary keys are underlined.

ACTOR (pid, fname, lname, agency)
MOVIE (mid, name, year)
DIRECTOR (did, fname, lname, studio)
CASTS (pid, mid, role)
MOVIE_DIRECTORS (did, mid)

A tuple in `Casts` represents the relationship that a person from the `Actor` table with `pid` starred (was cast) in a `Movie` with `mid`. Similarly, `Movie_Directors` represents the relationship that a person from `Director` with `did` directed a `Movie` with `mid`.

a) Draw an E/R diagram for the movie database including entities, relationships, and attributes. **Include a constraint that each movie has exactly one director.** (10 points)

Schema repeated here for your reference:

ACTOR (pid, fname, lname, agency)
MOVIE (mid, name, year)
DIRECTOR (did, fname, lname, studio)
CASTS (pid, mid, role)
MOVIE_DIRECTORS (did, mid)

For the next problems, assume that the `Director` relation is only the following 4 tuples:

did	fname	lname	studio
0	Sofia	Coppala	Paramount
1	Ang	Lee	MGM
2	Steve	McQueen	Warner
3	Francis	Coppala	Paramount

b) Circle whether the following functional dependencies hold on this instance of `Director`. (6 points)

`fname` → `lname` Holds / Does not hold

`lname` → `studio` Holds / Does not hold

`studio` → `fname` Holds / Does not hold

c) Write two new tuples to add to the database such that none of the three functional dependencies listed above will hold on the new instance. (4 points)

did	fname	lname	studio

Schema repeated here for your reference:

ACTOR (pid, fname, lname, agency)
MOVIE (mid, name, year)
DIRECTOR (did, fname, lname, studio)
CASTS (pid, mid, role)
MOVIE_DIRECTORS (did, mid)

b) There is one potential flaw in the database schema as given. The problem is that an actor and director might have the same name, but there is no way to record if they are the same person. To solve this we'll add a new entity to the database called `Person` with attributes (`pid, fname, lname`), and make `Actor` and `Director` subclasses of the Person entity.

Redraw the E/R diagram for the movies database which now includes a `Person` entity with `Actor` and `Director` as subclasses. You may change any attributes names as necessary. Be careful not to have redundancy in the attributes stored in `Actor` or `Director` that are already stored in `Person`. (10 points)

Problem 4: BCNF Decomposition (16 points total)

Assume the following functional dependencies hold on the relation R:

R (A, B, C, D, E, F)

A -> CDE
F -> ABD
D -> E

a) Write the closures of the following attributes: (8 points)

$\{A\}^+$ = _____

$\{B\}^+$ = _____

$\{AB\}^+$ = _____

$\{F\}^+$ = _____

b) Decompose R into Boyce-Codd Normal Form with respect to the above functional dependencies, indicating the new relations and their attributes (for example R1(A, B...), R2(B, C...) (8 points)

Problem 5: Indexing and Query Optimization (18 points total)

Assume we have relations R and S in a database, with the following attributes. The primary keys are underlined.

```
R(a, b)
S(c, d)
```

Say we run the following SQL query on the above tables:

```
SELECT R.a, S.c
FROM   R, S
WHERE R.a = S.d AND
      R.b = '1234'
```

Write a physical query plan for the above query that uses a block nested loop join (assume we use the page-at-a-time refinement.) Your query plan should be a relational algebra tree with annotations for the physical operators. (10 points)

Recall the notation for specifying statistics on the tables in a database:
`T(X)` is the number of tuples in a relation `X`.
`B(X)` is the number of blocks the relation `X` takes up on hard disk.
`V(X, y)` is the number of distinct values for the attribute `y` in the relation `X`.

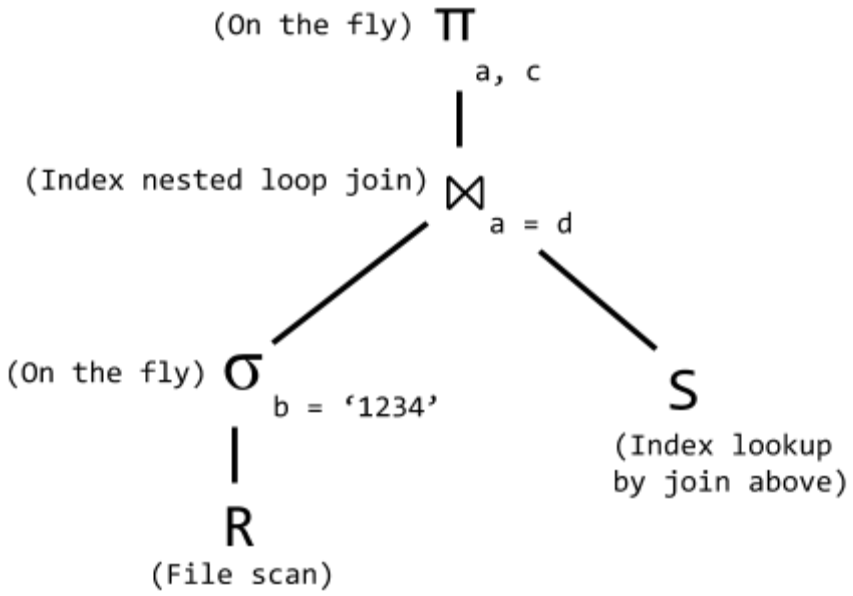
Now assume we have the following statistics on the tables `R` and `S`.

R	S
T(R) = 10,000 B(R) = 500 V(R, a) = 400 V(R, b) = 2,000	T(S) = 10,000 B(S) = 500 V(S, c) = 500 V(S, d) = 1,000 Unclustered B+ tree index on S(d)

b) Write the estimated cost in disk block I/Os to run your query plan with the block nested loop join on the previous page. (4 points)

_____ I/Os

Now consider this physical plan for the same SQL query:



c) Write the estimated cost in disk block I/Os for the above query plan. (4 points).

_____ I/Os

Problem 6: Parallel Databases (20 points total)

Say you are designing a parallel relational database to store purchase data for manufacturing products. The tables are:

```
Manufacturer(mid, name, category, city, state)
Purchase(pid, mid, date, amount) -- mid is a foreign key to Manufacturer
```

Tuples in the purchase table record individual payments in dollar amounts to a manufacturer for purchases of some product. There is a large amount of data in both tables that would have to be spread between multiple machines. As the database designer, you know that the most common query that will be run on the system is:

```
SELECT m.mid, SUM(p.amount) AS total_revenue
FROM   Purchase p, Manufacturer m
WHERE  p.mid = m.mid
GROUP BY m.mid
```

a) Describe in a few sentences how you would partition the data between machines if your goal is to maximize performance of the above query. (5 points)

b) Now consider that instead of maximizing performance of any query, your goal is to minimize skew and store the data evenly across the machines. Describe in a few sentences how you would partition the data in that case. (5 points)

Now you are also considering using a big data processing engine for the database instead of a parallel relational database. Your options are MapReduce (most likely in the form of the free system Hadoop) or Spark.

As the database designer, you are comparing the capabilities of both systems to determine which is the better fit.

For each of the following statements of big data processing system capabilities, circle the system or systems for which the statement holds true. If the statement doesn't hold true for any of the systems, don't circle any of the options.

c) This system can store intermediate results on the hard disk. (2 points)

MapReduce Spark

d) This system uses RDDs to store lineage of operations in a query plan. (2 points)

MapReduce Spark

e) This system has built-in implementations for SQL queries. (2 points)

MapReduce Spark

f) This system supports transactions for simultaneous queries. (2 points)

MapReduce Spark

g) This system runs on a distributed file system. (2 points)

MapReduce Spark