Name

CS451 Database Systems
Spring 2018

Sakire Arslan Ay, PhD

School of Electrical Engineering and Computer Science
Washington State University

Sample Final Exam

Time Limit: 110 minutes

- Print your name and WSU ID below. In addition, initial your name in the upper right corner of every page.

Name: _____ WSU ID:_____

- Including this cover page, this exam booklet contains 10 pages. Check if you have missing pages.
- The exam is closed book and closed notes. You are allowed to use a letter size cheat sheet (you may use both sides).
- No calculators or other electronic devices are permitted. Any form of cheating on the examination will result in a zero grade.
- Please write your solutions in the spaces provided on the exam. You may use the blank areas and backs of the exam pages for scratch work.
- Please make your answers clear and succinct; you will lose credit for verbose or confusing answers. Simplicity does count!
- The exam has 8 questions.
- You should look through the entire exam before getting started, to plan your strategy.

Q1 (12pts)	Q2 (8pts)	Q3 (16pts)	Q4 (16pts)	Q5 (20pts)	Q6 (12pts)	Q7 (6pts)	Q8 (10pts)	Total

Name

Question 1 (12pts) True/False Questions

For each of the following statements indicate whether it is TRUE or FALSE by circling your choice. **(Each question is worth 1 pts)**

(a) (1.5pts) Every relationship in an E-R diagram must translate to an individual relation in the relational model.

TRUE / FALSE

Answer:

(b) (1.5pts) Given an SQL query, there are often multiple ways of writing it in relational algebra.

TRUE / FALSE

Answer:

(c) (1.5pts) For any relation, it is possible to construct two separate unclustered indexes on different keys.

TRUE / FALSE

Answer:

(d) (1.5pts) Given a relation that is not in BCNF there is always a **unique** decomposition into relations that are in BCNF.

TRUE / FALSE

Answer:

(e) (1.5pts) B+ trees can include duplicate keys.

TRUE / FALSE

Answer:

(f) (1.5pts) Consider the below relation:

```
Student(sid, name, major, age)
```

A clustered hash index on <major> for Student will help to run the following query efficiently.

```
SELECT sid FROM Student WHERE  major like '%EE%';
```

TRUE / FALSE

Answer:

(g) (1.5pts) If A → BC and B→ D holds on relation R(A,B,C,D) then A → CD also holds on R.

TRUE / FALSE

Answer:

(h) (1.5pts) Consider the below relation:

movies (name, year, director, rank)

```
Q1:  SELECT *
     FROM movies AS m1 LEFT OUTER JOIN movies AS m2
         ON m1.name=m2.name
     WHERE m1.year=2000 AND m2.rank<7.3

Q2:  SELECT *
     FROM movies WHERE year = 2000 AND rank < 7.3
```

The above queries return the same result.

TRUE / FALSE

Answer:

Question 2 (8pts) Multiple Choice Questions

(4pts) 2.a) Which of the following statements correctly describe the difference between the HAVING and the WHERE clauses in SQL?

a) The WHERE clause tests conditions on aggregated rows resulting from the evaluation of the FROM clause; the HAVING clause tests conditions on individual rows resulting from the execution of sub-queries;
b) The WHERE clause tests conditions on individual and aggregated rows resulting from the evaluation of the FROM clause; the HAVING clause tests conditions only on aggregations calculated after the application of the GROUP BY clause;
c) The WHERE clause tests conditions on individual rows resulting from the evaluation of the FROM clause; the HAVING clause tests conditions on aggregations calculated after the application of the GROUP BY clause;
d) The WHERE clause is part of the SQL standard, and can be used in any relational DBMS systems; the HAVING clause is not part of the SQL standard, so it is not supported in all systems.

Answer:

(4pts) 2.b) Consider a relation R(a,b,c) with 10,000 records, and 100 pages (100 tuples on each page). R.a is the primary key and is a non-negative integer. How many pages will be read from disk to answer the selection query $\sigma_{a<2500}(R)$ for the following scenario?

"Relation R is stored in a heap file. There also exists an unclustered B+ tree index with search key **a**. The height of the B+tree is 3. Assume that 100 records match the selection predicate."

a) 13
b) 1000
c) 1003
d) 103
e) None of the above

Answer:

Name

Question 3 (16pts) SQL and Relational Algebra

The following questions refer to the database schema below:

Book(bookid, title, author, publisher, year, price)
Bookstore(name, address, city)
Sold(name, bookid, quantity)

bookid is the primary key for relation Book; name is the primary key for relation Bookstore; (name,bookid) is the primary key for relation Sold.
Sold.bookid is a FK referencing Book.bookid and Sold.name is a FK referencing Bookstore.name.

Only the first author for each book is recorded in the book table. The quantity attribute in Sold is at least 1 (i.e. quantity >=1).

a) Write the SQL SELECT query for the following:

"Find the names of the authors whose books were sold more than 10,000 copies in the city of Seattle. "

b) Write the SQL SELECT query for the following:

"Find the names of the bookstores which have sold more books than any other bookstore in the same city."

Name

Book(<u>bookid</u>, title, author, publisher, year, price)
Bookstore(<u>name</u>, address, city)
Sold(<u>name</u>, <u>bookid</u>, quantity)

c) Write the equivalent SQL SELECT query for the following relational algebra expression

$$\Pi_{\text{name,numBooks}} (\gamma_{\text{name,sum(quantity)->numBooks}} (\sigma_{\text{publisher='McGrawHill' AND year='2015'}} (\text{Bookstore} \bowtie \text{Sold} \bowtie \text{Book}))$$

Name

Question 4 (16pts) Relational Design Theory

Consider a relation R(A,B,C,D,E), with FDs
AB → C,
C → A,
C → BD,
D → E

(6pts) (a) List all the keys of R. Do not list superkeys which are not (minimal) keys.

(10pts) (b) Is this relation in BCNF? If you answer is yes, explain why it is. If you answer is no, decompose the relation into BCNF, showing your decomposition steps.

Name

Question 5 (20pts) Indexing

Consider the following relational schema for a portion of a company database:
Project (pno, proj_name, proj_base_dept, proj_mgr, topic, budget)
Manager (mid, mgr_name, mgr_dept, salary, age, sex)

Manager table stores the information about project managers.
Note that:

- pno is the primary key for Project and mid is the primary key for Manager.
- Project.proj_mgr is a foreign key referencing Manager.mid
- each project is based in some department (proj_base_dept),
- each manager is employed in some department (mgr_dept), and
- the manager of a project need not be employed in the same department (in which the project is based).

Suppose you know that the following queries are the five most common queries in the workload for this university and all five are roughly equivalent in frequency and importance:

1. List the names, departments, and ages of the managers who earn more than $90K. (i.e., salary>$90K)
2. List the names, ages, and salaries of managers of a user-specified sex (male or female) working in a given department. You can assume that, while there are many departments, each department contains very few managers.
3. List the names, topics and budgets of all projects with managers whose ages are in a user-specified range (e.g., younger than 30).
4. List the departments such that a manager in this department manages a project based in the same department.
5. List the project names and their departments with budget more than $900K.
6. List the name of the project with the lowest budget.
7. List the mid's of all managers in the same department as a given project *pno*.

These queries occur much more frequently than updates, so you should build whatever indexes you need to speed up these queries. However, you should not build any unnecessary indexes (or include any unnecessary attributes in an index), as updates will occur (and would be slowed down by unnecessary indexes). Assume index only query plans are possible.

Given this information, suggest indexes that will give good performance for each of the queries above . In particular decide,

(i) which attributes should be indexed (both single- and multiple-attribute index search keys are permitted),
(ii) whether each index should be a clustered index or an unclustered index, and
(iii) whether it should be a B+ tree or a hashed index.

Name

Question 6 (12pts)

Consider the relation R(a,b,c,d,e) with the following properties/statistics:

- The primary key is ab (neither a nor b values are unique in R).
- There are a total of **15,000** tuples in R stored in **1,000** pages (when stored in a heap file).
- The number of distinct values of attribute a in R is **300.**
- The "a" values are uniformly distributed in relation R.

Suppose we have a B+tree index available for attribute "a" of R. The height of the B+ tree is 3. Assume each node of the B+ tree index occupies one page. Assume the cost is measured by disk I/Os for accessing the index and the records.

We run the following query on relation R:

```
SELECT *
FROM R
WHERE a = 1000
```

a) (**6pts**) Estimate the average cost of executing the above query when the B+tree index is clustered, i.e., the records with the same a value are stored consecutively in the disk but may spread in different blocks. (Assume the B+tree has 67% occupancy, i.e., the physical data pages are 1.5 times more than original data file.)

b) (**6pts**) Estimate the cost of executing the above query when the index is unclustered. (Assume the size of a data entry in the B+tree is around 20% of the data record (the data entry also includes the RID)).

Name

Question 7 (6pts)

Mention four advantages of using database management systems (DBMS) over file systems.

Question 8 (10pts)

TBA