Final exam

Due Jul 29 at 11:59pm **Points** 300 **Questions** 22 **Available** Jul 29 at 6am - Jul 29 at 11:59pm about 18 hours
Time Limit 135 Minutes

Instructions

This is the final exam. If you don't have any special accommodation, you will have 2 hours 15 minutes to finish this exam.

This quiz was locked Jul 29 at 11:59pm.

Attempt History

	Attempt	Time	Score
LATEST	Attempt 1	134 minutes	274 out of 300

Score for this quiz: **274** out of 300
Submitted Jul 29 at 2:05pm
This attempt took 134 minutes.

Question 1

4 / 4 pts

Which one of the following statements about normal forms is FALSE?

- BCNF is stricter than 3NF
- Lossless, dependency-preserving decomposition into 3NF is always possible
- Correct! [✓ Loss less, dependency–preserving decomposition into BCNF is always possible]
- Any relation with two attributes is BCNF

Question 2

4 / 4 pts

Suppose relation *R*(*A*,*B*,*C*,*D*,*E*) has the following functional dependencies:

A -> B

B -> C

BC -> A

A -> D

E ->A

D -> E

Which of the following is ***not*** a candidate key?

- A
- E
- B
- D

Correct!

[✓ None of the above]

Question 3

4 / 4 pts

Which of the following is ***not*** true about a subquery?

- If a subquery is guaranteed to produce one tuple, then the subquery can be used as a value
- A parenthesized SELECT-FROM-WHERE statement (subquery) can be used as a value(s) in a number of places, including FROM and WHERE clauses
- Correct! [✓ The result of the main query is returned to the subquery]

Question 4

4 / 4 pts

A disk with sector size being 1024 bytes, 2000 tracks per surface, 50 sectors per track, 5 double-sided platters and average seek time: 10msec. Which of the followings is NOT a valid block/page size?

- 2048 bytes
- 4096 bytes
- 3072 bytes
- Correct! [✓ 512 bytes]

Question 5

4 / 4 pts

The minimum space utilization for a B+ tree index (except root node) is

- 25%
- Correct! [✓ 50%]
- 75%
- unknown

Question 6

4 / 4 pts

A *run* in multi-way merge sort algorithm refers to:

- A collection of merges that goes through entire data
- A merge process

Correct!

- [✓ A sorted sublist]
- An entire array

Question 7

4 / 4 pts

What is the information stored in a catalog of database (in which metadata is stored) ?

- Number of tuples and attributes
- Number of pages
- Number of distinct key values
- Correct! [✓ All of the above]
- None of the above

Question 8

4 / 4 pts

Benefits of extendable/extendible hashing are: (i)__________________________ and (ii)________________________

- (i) Having good hash function, (ii) more efficient than static hashing techniques
- Correct! [✓ (i) Hash performance does not degrade with growth of file, (ii) minimal space overhead]
- (i) Changing the hash function on the fly, (ii) saving spaces.

Question 9

4 / 4 pts

Consistency property was introduced in the 70s but is now considered as a consequence of:

- Isolation and Durability
- Durability and Atomity
- Correct! [✓ Atomity and Isolation]
- Durability and Consistency

Question 10

4 / 4 pts

The first step in optimizing a query is

- Correct! [✓ Generate a subset of all possible physical query plans]

○ Estimating the cost of each physical query plan

○ Find out which join algorithms to use

○ Find out the access path

Question 11

8 / 8 pts

Given two relations *R*1 and *R*2, where *R*1 contains N1 tuples, *R*2 contains N2 tuples, and N2 > N1 > 0, give the minimum and maximum possible sizes (in tuples) for the resulting relation produced by each of the following relational algebra expressions. In each case, describe the schema of the resulting relation:

Operators	Describe result schema	Min	Max
$R_1 \cup R_2$			
$R_1 \cap R_2$			
$R_1 - R_2$			
$R_1 \times R_2$			

Your Answer:

Union	R1 and R2 should have exactly same schema. Result table has same schema as R1 and R2 Result table contains tuples of R1 and R2	N2	N1 + N2
intersection	R1 and R2 should have exactly same schema. Result table has same schema Result table contains tuples that have same tuples existing in R1 and R2	0	N2

Difference	R1 and R2 should have exactly same schema. Result table have the same schema Result table contains all tuples of R1 that do not exist in R2	0	N1
Cartesian product	R1 and R2 can have different schema. Results will have all attributes of R1 and R2	N1*N2	N1 * N2

Question 12

4 / 4 pts

What is the difference between logical operators and physical operators? Please give an example of a logical operator and a physical operator which could be associated with that logical operator.

Your Answer:

Logical operators describe the relational operations used to process the query statement

Example: Selection, Projection, Join, Union, Grouping, etc

Physical operators implements the operations described by the logical operators

Example: Nested loop join, Sort-merge join, hash-join, etc

Question 13

4 / 4 pts

What is wrong with the following query?

```
SELECT subject_code, count(name)
FROM students;
```

Your Answer:

There is no statement group by so the count(*) will not work.

There should be a group by (subject_code) to return the number of students in each subject_code.

Question 14

3 / 4 pts

Please explain the difference between Hash indexes and B+tree indexes. In particular, discuss how equality (e.g age = 20) and range searches (e.g age >= 20) work

Your Answer:

Hash indexes are used only when all conjuncts in the predicate must be equality and set of attributes in the key should be a subset of attributes appearing in conjuncts. So this index will not work for range searches.

B+ Tree indexes are used when the prefix subset of search key is a subset of attributes appearing in conjuncts. This can work for both equality and range searches.

Need more argument when hash doesn't support range searches.

Question 15

15 / 15 pts

Consider the following relations:

Flights(*flno*: integer, *from*: string, *to*: string, *distance*: integer, *departs*: time, *arrives*: time)

Aircraft(*aid*: integer, *aname*: string, *cruising_range:* integer)

Certified(*eid*: integer, *aid*: integer)

Employees(*eid*: integer, *ename:* string, *salary*: integer)

Note that the Employees relation describes pilots and other kinds of employees as well; every pilot is certified for some aircraft (otherwise, he or she would not qualify as a pilot), and only pilots are certified to fly.

Write the following queries in **SQL** and **relational algebra:**

Find the names of pilots certified for some Boeing aircraft. (Boeing aircraft is aircraft name or *aname*)

For relational algebra, use the following notation:

Selection: σ (Click to symbol $\sqrt{x}$ -> Greek tab) OR write it out as SELECTION

Projection: π (Click to symbol $\sqrt{x}$ -> Greek tab) OR write it out as PROJECTION

Natural join: $\bowtie$ (Click to $\sqrt{x}$ symbol -> Relationship tab) OR write it out as NATURAL JOIN

Product: $\times$ (Click to $\sqrt{x}$ symbol -> Basic tab) OR write it out as PRODUCT

Your Answer:

SELECT E.ename
FROM Employees E, Aircraft A, Certified C

WHERE E.eid = C.eid

AND C.aid = A.aid

AND A.aname = 'Boeing';

PROJECTION $_{\text{ename}}$ (SELECTION $_{\text{Employees.eid = Certified.eid AND Certfied.aid = Aircraft.aid AND Aircraft.aname = 'Boeing'}}$ (Employees NATURAL JOIN Certified NATURAL JOIN Aircraft))

Question 16

15 / 15 pts

Consider the following relations:

Flights(*flno*: integer, *from*: string, *to*: string, *distance*: integer, *departs*: time, *arrives*: time)

Aircraft(*aid*: integer, *aname*: string, *cruising_range:* integer)

Certified(*eid*: integer, *aid*: integer)

Employees(*eid*: integer, *ename:* string, *salary*: integer)

Note that the Employees relation describes pilots and other kinds of employees as well; every pilot is certified for some aircraft (otherwise, he or she would not qualify as a pilot), and only pilots are certified to fly.

Write the following queries in **SQL** and **relational algebra**

Find the names of pilots who make more than *$100,000.*

For relational algebra, use the following notation:

Selection: σ (Click to symbol $\sqrt{x}$ -> Greek tab) OR write it out as SELECTION

Projection: π (Click to symbol $\sqrt{x}$ -> Greek tab) OR write it out as PROJECTION

Natural join: $\bowtie$ (Click to $\sqrt{x}$ symbol -> Relationship tab) OR write it out as NATURAL JOIN

Product: $\times$ (Click to $\sqrt{x}$ symbol -> Basic tab) OR write it out as PRODUCT

Your Answer:

Find the names of pilots who make more than *$100,000.*

SELECT DISTINCT E.ename

FROM Employees E, Certified C

WHERE E.eid = C.eid

AND E.salary > 100,000;

PROJECTION $_{\text{Employees.ename}}$ (SELECTION $_{\text{Employees.eid = Certified.eid AND Employees.salary > 100,000}}$(Employees NATURAL JOIN Certified))

Question 17

30 / 30 pts

Consider a relation R with five attributes ABCDE. You are given the following dependencies:

A → B, BC → E, and ED → A.

1. List all keys for R.

2. Is R in 3NF? Please justify.

3. Is R in BCNF? Please justify.

Your Answer:

1. CED, ACD, BCD

2. R is in 3NF

Condition for 3NF

1. left hand side of FD is super key, or

2. Right hand side of FD is prime attribute

For all FDs, B, E, and A are prime attributes.

3. R is not in BCNF

Condition for BCNF

1. Left hand side of each FD is super key

Candidate key: CED, ACD, BCD

All left hand side (A, BC, ED) of each FD is not a candidate key.

Question 18

30 / 30 pts

Assuming that you have a three-frame buffer pool. Please show the buffer content given the workload listed in the table below using LRU and MRU strategies. We start from T1 and moves forward by one on each page reference. For your solution, underline the letter in the frame in the buffer pool each time that a memory access caused a “miss” in the buffer pool (i.e., a page is read that is not currently in the buffer pool), when the page is put in the buffer pool. When the buffer pool has unused slots (such as at the beginning, when all slots are empty), it will put newly read data in the first unused slot. The pages to be read from disk are labeled A through D. For each access, the page is pinned, and then immediately unpinned. Assume one page on disk fits perfectly with one frame on the buffer pool.

Time	Page Read
T1	A
T2	B
T3	C
T4	D
T5	B
T6	C
T7	D
T8	C
T9	D
T10	D

1. In your answer, please create two buffer pools, one for LRU strategy and one for MRU strategy. You could create a similar buffer pool by click to table icon ⊞▾

Buffer Pool

LRU strategy

Time	Frame 1	Frame 2	Frame 3
T1	A (example)		
T2			

T3			
T4			
T5			
T6			
T7			
T8			
T9			
T10			

MRU strategy

Time	Frame 1	Frame 2	Frame 3
T1	A (example)		
T2			
T3			
T4			
T5			
T6			
T7			
T8			
T9			
T10			

2. Which strategy is working better in this scenario? Please justify your answer.

Your Answer:

LRU strategy

Time	Page read	Frame 1	Frame 2	Frame 3
T1	A	A		
T2	B	A	B	
T3	C	A	B	C
T4	D	D	B	C
T5	B	D	B	C
T6	C	D	B	C

T7	D	D	B	C
T8	C	D	B	C
T9	D	D	B	C
T10	D	D	B	C

Misses: 4

MRU strategy

Time	Page read	Frame 1	Frame 2	Frame 3
T1	A	A		
T2	B	A	B	
T3	C	A	B	C
T4	D	A	B	D
T5	B	A	B	D
T6	C	A	C	D
T7	D	A	C	D
T8	C	A	C	D
T9	D	A	C	D
T10	D	A	C	D

5 misses

2. Which strategy is working better in this scenario? Please justify your answer.

LRU strategy is working better in this scenario because it has less misses than MRU strategy .

LRU had 4 misses while MRU had 5 misses.

Question 19

30 / 30 pts

Given two files A and B and each file includes 4 pages (each page contains 3 entries only) as shown as the following:

A: [[2,-5, 6], [10, 4, 7], [11,3,8],[5, 0, 20]

B: [[9,5,2], [25,-15, 0], [33, 12, 15], [19, 16, 25]]

Assume you have 3 buffer pages in main memory.

1. Show the **2-way merge sort** (sort each page separately in the first pass) procedure for the given example.

2. Compute the I/O cost for this procedure.

Your Answer:

1. Instead of using all three buffers in the first phase (sort), I sort each page separately and write the sorted page out

Input:

[[2,-5, 6], [10, 4, 7], [11,3,8],[5, 0, 20]

[[9,5,2], [25,-15, 0], [33, 12, 15], [19, 16, 25]]

Output:

[2, 5, 6], [4, 7, 10],

[3, 8 11], [0,5,20]

[2, 5, 9], [-15, 0, 25],

[12, 15, 33], [16, 19, 25]

Cost: 2*8 = 16 I/O

Input:

[2, 5, 6], [4, 7, 10],

[3, 8 11], [0,5,20]

[2, 5, 9], [-15, 0, 25],

[12, 15, 33], [16, 19, 25]

Output:

[2, 4, 5], [6, 7, 10]

[0, 3, 5], [8, 11, 20]

[-15, 0, 2], [5, 9, 25]

[12, 15, 16], [19, 25, 33]

Cost: 2*8 = 16 I/O

Input

[2, 4, 5], [6, 7, 10]

[0, 3, 5], [8, 11, 20]

[-15, 0, 2], [5, 9, 25]

[12, 15, 16], [19, 25, 33]

Output

[0, 2, 3], [4, 5, 5], [6, 7, 8], [10, 11,20]

[-15, 0, 2], [5, 9, 12], [15, 16, 19], [25, 25, 33]

Cost: 2* 8 =16 I/O

Input

[0, 2, 3], [4, 5, 5], [6, 7, 8], [10, 11,20]

[-15, 0, 2], [5, 9, 12], [15, 16, 19], [25, 25, 33]

Output

[-15, 0, 0], [2, 2, 3], [4, 5, 5], [5, 6, 7], [8, 9, 10], [11, 12 15], [16, 19, 20], [25, 25, 33]

Cost: 2 * 8 = 16 I/O

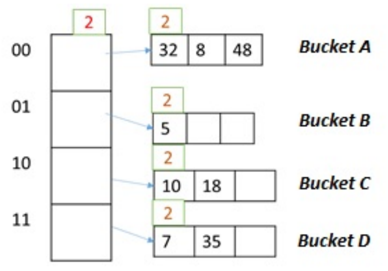
2.

There are 4 passes with 16 I/O

So the total I/O cost is = 64 I/O

Question 20

32 / 40 pts



Given the above hash table.

1. What is the current hash function?

2. What happens if we insert 26? Which bucket should we change?

3. What happens if we insert 40? What is the new hash function?

4. Explain why local depth and global depth are needed.

Your Answer:

1. H(x) = X mod 4

2.

26 mod 4 = 2

26 is inserted into Bucket C.

Bucket C has free space so 26 can be inserted into the bucket

3.

40 mod 4 = 0

40 is inserted into Bucket A

Bucket A has full capacity.

We checked Global depth = Local depth so we double the table size.

Update the hash function by H(x) = X mod 8

Increment the local depth of Bucket A to 3.

Increment global depth by 1. So the new global depth = 3

Add new bucket pointed from index 100 of the table. The local depth of the new table is 3.

Redistribute the data in Bucket A by using the new Hash function.

Create a pointer from Index 101 pointing to Bucket B

Create a pointer from Index 110 pointing to Bucket C

Create a pointer from Index 111 pointing to Bucket D

4.

When there is no free space for new entry to be inserted in the bucket, we compare global depth and local depth for solving overflow problem.

If the global depth > Local depth, we can create a new bucket and redistribute the entries stored in the bucket that is full by using the hash function.

If the global depth = local depth, we decide to double the size of table and redistribute the entries stored in the bucket that is full by using new hash function.

This helps to minimize the space used and not to degrade with the growth of a file.

3. Need to double once more. x mod 16

Question 21

38 / 50 pts

Recalling our university database, consider the following relations:

Student(**snum:string**, sname:string, major:string, level:string, age:int);

Enrolled(**snum: string, cname: string**)

And consider the following query

```
    SELECT s.sname, s.major
    FROM     Student s, Enrolled e
    WHERE  s.snum = e.snum and e.cname ='Database Management Systems' and s.level ='JR';
```

You are given the following information:

T(Students) = 3000 records

T(Enrolled) = 30000 records

B(Students) = 300 pages

B(Enrolled) = 400 pages

V(Student, level) = 4

V(Enrolled, cname) = 600

You always used the **block-nested loop join**

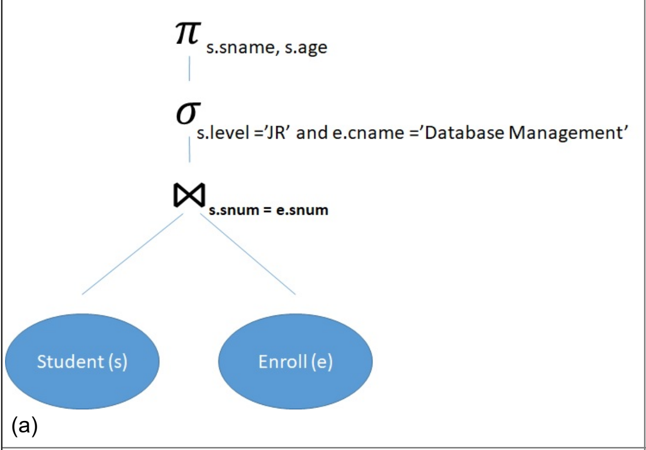
M = 102 (pages) (please use 100 pages in calculation of join cost)

First question: (15 points)

Here are four different logical query plans. Identify **two plans by their labels (a,b,c or d)** that do the following:

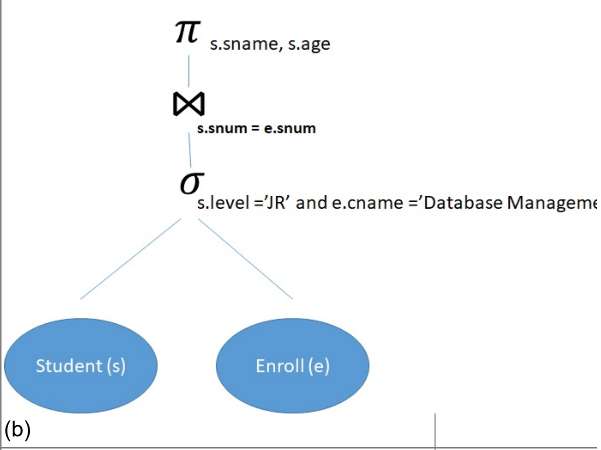
(i) one query plan would join two tables first, then perform selection on level and course name, then display the name and age of students;

(ii) the other query plan would modify the query plan mentioned in (i) by applying push selection rule from equivalence rules



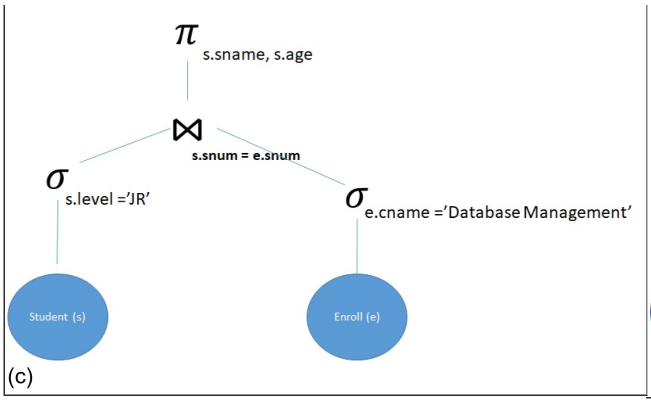
π s.sname, s.age
σ s.level ='JR' and e.cname ='Database Management'
$\bowtie$ s.snum = e.snum
Student (s)
Enroll (e)

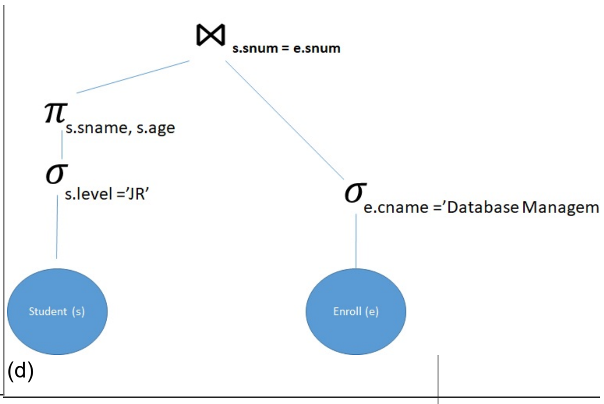
(a)



π s.sname, s.age
$\bowtie$ s.snum = e.snum
σ s.level ='JR' and e.cname ='Database Manageme
Student (s)
Enroll (e)

(b)





Second question (35 points)

Compute the I/O costs for the query plans in (i) and (ii). Which plan is better and why

Your Answer:

First question

i = A

A joins two tables first, then performs selection on level and course name, then display the name and age of students

ii = C

C modifies the query plan mentioned in A by applying push selection rule from equivalence rules

Second question

i)

Relation Students has 300 pages < Relation Enrolled has 400 pages.

So Relation Students is used as outer relation in BNLJ.

Cost of BNLJ = # pages of outer relation + # pages of inner relation * #outer blocks

# Outer block = Ceiling of (# pages of outer relation / B -2).

# Outer block = Ceiling of 300/100 = 3.

Cost of BNLJ = 300 + 400*3 = 1500 I/O.

ii)

# pages of students + # pages of students / selection cost = 300 + 300/4 = 375

# pages of Enroll + # pages of Enroll / selection cost = 400 + 400/600 =401

Cost of BNLJ = 375 + 401(375/100) = 375 + 401*4 = 1979 I/O

Query plan i has lower I/O cost than query plan ii. 1500 <1979.

Therefore, query plan i is better.

Cost for c is incorect.

Question 22

25 / 30 pts

You are hired to help the Lead Technical Analyst of the FBI Behavioral Unit in joining various tables and doing cross-references to help the FBI field agents to fight crimes. You are working up against an incredibly tight deadline because the suspects (the "unsub") may strike again in 24 hours. The tables that you are working with are big, but not sorted or indexed. You need to implement a decent join algorithm. Will you choose tuple-based nested-loop join, block-nested-loop join, index-nested-loop join, sort-merge join, or hash join? Please justify your answer, i.e., clearly explain why you think your choice is the best among the 5 alternatives. Please feel free to make any assumptions in your answer.

Your Answer:

I would recommend Hash Join for the join algorithm.

I assume the predicates that are used in cross-references are equalities because the data stored in the table is not sorted or indexed. Therefore, the predicates used in the references to tables must be equality predicates. The condition to use Hash join is met.

Hash Join does not require sorted or indexed data so I can use Hash Join in the case.

Block based nested loop cannot be used because the tables are not sorted. To use this algorithm, the tables must be sorted first, which will cost more I/O than Hash Join.

Tuple based Nested loop join can work as a join algorithm but it will cost more I/Os because it has to read every tuple of tables that are to be joined.

Index-nested loop join cannot be used because the tables are not indexed. To use the join, I have to create indexing for tables to be joined. This will cost more I/O.

Sort Merge join can work but it will cost more I/O because it needs to sort and merge tables to be joined before the join. This will cost more I/O than Hash Join.

Hash join IO Code needed

Quiz Score: **274** out of 300