

CS 564 Final Exam Fall 2018

Answers

A: RELATIONAL ALGEBRA, SQL & NORMALIZATION [24pts]

I. [8pts] Consider a relation $R(A, B, C, D)$ with the following instance.

A	B	C	D
2	2	3	4
2	2	2	4
2	1	3	4
3	4	5	4

For the questions below, clearly **circle** the correct option.

1. The functional dependency $A \rightarrow D$ holds for R .

2. $\{C, D\}$ is not a key for R .

II. [8pts] Consider a relational schema with one relations $R(A, B)$ and the following query in Relational Algebra:

$$q = \pi_B(\sigma_{A=1}(R) \bowtie_{B=B'} \rho_{A \rightarrow A', B \rightarrow B'}(R))$$

Which of the following queries are equivalent to q ? Clearly **circle** all the correct options.

- (a) $\pi_B(\sigma_{A=1}(R))$
- (b) $\sigma_{A=1}(\pi_B(R)) \bowtie_{B=B'} \rho_{A \rightarrow A', B \rightarrow B'}(R)$
- (c) $\pi_B(\sigma_{A'=1}(R \bowtie_{B=B'} \rho_{A \rightarrow A', B \rightarrow B'}(R)))$
- (d) $\pi_B(R) - \pi_B(\sigma_{A \neq 1}(R))$

ANSWER:

III. [8pts] Consider the following relation that describes a labelled directed graph:

Edges (start, end, label)

In this question, we are interested in counting *patterns* in the graph. A *triangle* is a pattern of three edges $(a,b), (b,c), (c,a)$. An *open triangle* is a pattern of two edges $(a,b), (b,c)$ such that (c,a) is **not** an edge in the graph. Write a SQL query that computes the *ratio* of triangles to open triangles.

ANSWER:

B: STORAGE AND INDEXING [28pts]

I. [9pts] Consider the following SQL query:

```
(SELECT  *
FROM    R
WHERE   R.A = 1 AND R.B > 10 )
UNION
(SELECT  *
FROM    R
WHERE   NOT (R.A < 1) AND R.C = 5 )
```

In the following matrix, check the boxes that correspond to combinations of indexes (hash or B+ tree indexes) that can speed up the above query:

	hash(A,B)	B+(A)	B+(C,A)
B+(B)			
hash(A)			
hash(A,B,C)			

--

II. [10pts] Consider a B+ tree index with order $d = 4$ and fill factor $F = 1$ and height $h = 3$. Assume that each leaf node of the B+ tree can hold up to 100 data entries.

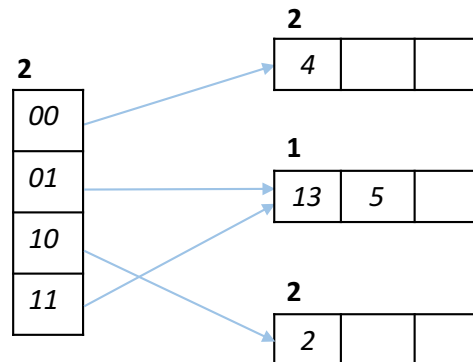
1. [5pts] What is the total number of pages in the above B+ tree?

ANSWER:

2. [5pts] What is the number of records that the above B+ tree can index? Explain your answer clearly.

ANSWER:

III. [9pts] Consider the following extendible hash index. What is the maximum number of entries you can insert in the index before the directory doubles in size? Explain your answer in detail.



ANSWER:

C: QUERY EXECUTION [36pts]

I. [8pts] We are given a relation $R(A, B)$ with 100 pages, and a relation $S(C, D)$ with 200 pages. In relation R , the attribute A is the primary key, and takes values $1, 2, \dots$. Each record in R is 40 bytes long, while each record in S is 10 bytes long. The size of a page is 1,000 bytes.

How many pages do we need to store the output of the following SQL query? Explain your answer in detail.

```

SELECT  *
FROM    R, S
WHERE   R.A = 1 ;
  
```

ANSWER:

II. [8pts] We are given one relation $R(A, B, C, D)$ with N pages, where each attribute has exactly the same size. Suppose we want to perform a **distinct project** on attributes A, B . The buffer pool has size $B = 21$. What is the largest possible N such that sort-based projection needs only 2 passes? Assume that we do not use replacement sort in the first pass. Explain your answer in detail.

ANSWER:

III. [20pts] Consider the following database schema: $R(A, B), S(A, C), T(A, D), U(A, E)$. Relation R has 1,000 pages, S has 500 pages, and T, U have 100 pages each. All four relations are clustered on the attribute A , but there are no other indexes.

Suppose we want to run the following SQL query:

```
SELECT *
FROM   R, S, T, U
WHERE  R.A = S.A AND R.A = T.A AND R.A = U.A ;
```

1. [6pts] Write a *left-deep join* plan for the above SQL query. Draw the plan as a tree.
2. [6pts] How many *left-deep join* plans does the above SQL query have? Explain your answer:
3. [8pts] Suppose that the buffer pool has size $B = 100$ frames. For the left-deep join plan you provided in (1), write the most efficient physical plan and compute its I/O cost.

ANSWER:

D: TRANSACTION MANAGEMENT [12pts]

I. [6pts] For the following questions, **clearly circle** either True or False.

1. Strict two-phase locking (2PL) ensures that transactions never deadlock.
2. The WAL protocol guarantees *atomicity* and *consistency*.

3. If a transaction reads a data item after it is written by an uncommitted transaction, then *isolation* is always violated.

II. [6pts] Consider the following interleaved schedule of transactions T_1, T_2, T_3 :

$R_{T_1}(A), R_{T_2}(B), R_{T_3}(B), W_{T_2}(B), W_{T_1}(A), W_{T_3}(A), R_{T_2}(A), W_{T_2}(C).$

Is this schedule serializable or not? If it is serializable, provide the equivalent serial schedule. Explain your answer in detail.

ANSWER: