CITY UNIVERSITY OF HONG KONG

Course code & title	:	CS5481 Data Engineering
Session	:	Semester A 2020/21
Time allowed	:	Two hours

This paper has 8 pages (including this cover page).

1. This paper consists of 4 questions.
2. Each question carries equal marks.
3. Answer ALL questions.
4. Specify the Question number clearly for EACH answer in the answer script.
5. Submit ONE pdf file to Canvas.
6. Use your Student ID to name the pdf file.

This is an ***open-book*** *examination.*

NO *access to the Internet, except for the operation of the examination.*

Candidates are allowed to use an approved calculator.

CS Departmental Hotline (phone, whatsapp, wechat): +852 6375 3293

Copy-and-paste the following academic honesty pledge on the first page of the answer script.

"I pledge that the answers in this examination/quiz are my own and that I will not seek or obtain an unfair advantage in producing these answers. Specifically,

- ***I will not plagiarize (copy without citation) from any source;***
- ***I will not communicate or attempt to communicate with any other person during the examination/quiz; neither will I give or attempt to give assistance to another student taking the examination/quiz; and***
- ***I will use only approved devices (e.g., calculators) and/or approved device models.***
- ***I understand that any act of academic dishonesty can lead to disciplinary action."***

Write the following together with your student ID and name to reaffirm the academic honesty pledge onto the first page of the answer script.

"I pledge to follow the Rules on Academic Honesty and understand that violations may lead to severe penalties."

Student ID: ____________________________________

Student Name: ____________________________________

Q1. [25 marks]

I. Relational Algebra [10 marks]

Consider the following relations containing airline information:

Aircraft (*aid*, *aname*, *cruisingrange*)
Certified (*eid*, *aid*)
Employees (*eid*, *ename*, *salary*)

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).

Write the following queries in relational algebra.

a) Find the names of pilots certified for some Boeing aircraft (Boeing is the name of aircrafts).

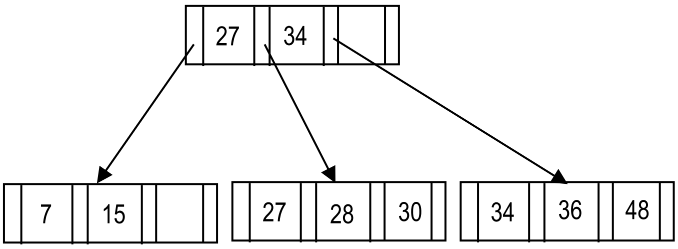
b) List the names and salaries of pilots who are certified for more than two aircrafts.

c) Find the eids of employees who make the highest salary.

II. B+-tree [15 marks]

a) Consider the following B^+-tree with n=4. What is the ***minimum*** number of search-key values you must insert to increase the height of the tree ***by one level***? Show the sequence of insertions and draw a diagram for ***each*** insertion.

[8 marks]



b) Consider a B^+-tree dense index on a file containing 20,000 records. The search-key for this B^+-tree index is a 70-byte string, and it is a candidate key. Pointers are 10-byte values. The size of one disk block is 2,048 bytes. The index was built in a bottom-up fashion for bulk-loading, and the nodes at each level were filled up as much as possible. Find the number of levels in the tree and the number of nodes at each level.

[7 marks]

Q2. Query processing and optimization [25 marks]

Consider the following relations, where the keys are underlined:

ENGINEER (ID, Name, Profile)
PROJECT (PID, ICEngID, Budget)

The ICEngID attribute in PROJECT is the ID of the engineer who is in charge of the project and PID is the ID of the project. Suppose all engineers must be in charge of projects.

Consider the following query.

```
SELECT      *
FROM        ENGINEER E, PROJECT P
WHERE       E.ID=P.ICEngID
```

Given the following statistics and indices:

- number of tuples in ENGINEER: 10,000
- number of tuples in PROJECT: 20,000
- tuples do not span across blocks
- size of attributes ID, ICEngID: 5 bytes
- size of attribute Name: 10 bytes
- size of attribute Profile: 20 bytes
- size of attribute PID: 7 bytes
- size of attribute Budget: 25 bytes
- *min* (Budget, PROJECT) = 10
- *max* (Budget, PROJECT) = 60
- disk block size: 1,024 bytes
- 3-level B^+-tree primary index on ID for ENGINEER
- 4-level B^+-tree primary index on PID for PROJECT
- 3-level B^+-tree secondary index on ICEngID for PROJECT

a) Estimate the number of output tuples for the query. Explain.

[2 marks]

b) Suppose both relations are stored at the same site. What is the lowest ***worst-case cost*** in ***number of disk block transfers*** if the query is processed with the ***indexed nested-loop join*** algorithm? Show the steps clearly.

[5 marks]

c) If the condition P.Budget > 20 is added to the WHERE clause in the query, would you apply the "perform selection early" heuristic to optimize the evaluation plan in part b)? Explain your choice by finding the ***worst-case cost*** in ***number of disk block transfers*** in both cases.

[8 marks]

d) Suppose ENGINEER is stored at site A and PROJECT is stored at site B. The query is submitted at site B and the result is also needed at site B. **Suppose now only a certain number of engineers are qualified to be in charge of a project.** Determine the ***maximum*** number of qualified engineers to justify the use of the ***semi-join*** strategy. Describe the steps clearly.

[10 marks]

Q3. [25 marks]

I. Transactions [7 marks]

Consider the following two transactions and some possible schedules.

T_1: $w_1(x)$; $w_1(y)$; $r_1(z)$;
T_2: $w_2(x)$; $r_2(y)$; $w_2(z)$;

S_1: $w_1(x)$ $w_1(y)$ $w_2(x)$ $r_1(z)$ $r_2(y)$ $w_2(z)$ c_2 c_1
S_2: $w_2(x)$ $w_1(x)$ $w_1(y)$ $r_1(z)$ $r_2(y)$ c_1 $w_2(z)$ c_2
S_3: $w_2(x)$ $w_1(x)$ $w_1(y)$ $r_1(z)$ c_1 $r_2(y)$ $w_2(z)$ c_2
S_4: $w_2(x)$ $w_1(x)$ $r_2(y)$ $w_2(z)$ c_2 $w_1(y)$ $r_1(z)$ c_1

For each of the above schedules, complete the following table to indicate whether the schedule is ***recoverable***, ***cascadeless*** and ***conflict serializable***. If the schedule is conflict serializable, show its ***serializability order***.

Schedule	recoverable (Y/N)	cascadeless (Y/N)	conflict serializable (Y/N)	serializability order
S_1				
S_2				
S_3				
S_4				

II. Concurrency Control [8 marks]

Consider the three transactions T_1, T_2, and T_3, and the schedule S given below.

T_1: $r_1(x)$; $r_1(z)$; $w_1(x)$
T_2: $r_2(z)$; $r_2(y)$; $w_2(z)$; $w_2(y)$
T_3: $r_3(y)$; $r_3(x)$; $w_3(y)$
S: $r_1(x)$; $r_2(z)$; $r_3(y)$; $r_1(z)$; $r_3(x)$; $w_1(x)$; $w_3(y)$; $r_2(y)$; $w_2(z)$; $w_2(y)$

a) Explain whether the schedule S is ***conflict serializable*** with the help of a ***precedence graph***.

b) Consider using the basic ***two-phase locking*** protocol (i.e., no lock conversion), insert lock (*sl* for lock-s or *xl* for lock-x) and unlock (*ul*) instructions to the schedule S and describe its execution result with the help of a ***wait-for graph***, assuming that a waiting transaction does not block the following non-conflicting instructions of other transactions.

III. Database Recovery [10 marks]

Consider the following sequence of log records.

<T_0 start>
<T_0, B, 2000, 2050>
<T_1 start>
<T_1, A, 100, 200>
<T_0 commit>
<checkpoint L>
<T_2 start>
<T_2, D, 50, 70>
<T_3 start>
<T_3, E, 300, 500>
<T_1, C, 700, 600>
<T_2, D, 50>
<T_3 commit>

Suppose a crash occurs *after* the last log record is written out.

a) What are the possible values of all the data items **on disk** *after* the crash but *before* recovery?

b) What are the log records added during recovery? Show the log records in output order.

c) What are the final values of all the data items *after* recovery?

Q4. [25 marks]

I. Parallel Databases [17 marks]

a) Consider a relation consisting of 1,000 tuples to be divided into 5 partitions by ***range partitioning*** and the values of the ***partitioning attribute*** are integers ranging from 1 to 600.
[10 marks]

(i) Construct a ***range-partitioning vector*** on the partitioning attribute and state any assumption(s) you need to make in order to construct the partitioning vector.

(ii) Describe the potential problem that would result in a loss of performance when the relation is partitioned by the range-partitioning vector you constructed in Part (i).

(iii) If the following histogram of the partitioning attribute is available in the database catalog, construct a ***balanced range-partitioning vector*** to alleviate the problem in Part (ii).

Range	frequency
1–60	150
61–120	100
121-180	150
181-240	50
241-300	100
301-360	50
361-420	100
421-480	200
481-540	50
541-600	50

b) Explain how ***partitioned join*** can be used for $r \bowtie_{r.A<s.A \wedge r.B=s.B} s$. In particular, how is it possible for each node to compute the join of its own partitions locally in a parallel database?
[7 marks]

II. Distributed Transactions [8 marks]

Consider the following scenario in a distributed database system in which it is guaranteed that at most one global transaction is active at any time and every local site ensures local serializability. Suppose there are two sites and four transactions. T_1 and T_2 are local transactions, running at site 1 and site 2 respectively. T_{G1} and T_{G2} are global transactions running at both sites. x_1, y_1 are data items at site 1, and x_2, y_2 are at site 2.

T_1: $w(y_1)$; $r(x_1)$;
T_2: $r(x_2)$; $w(y_2)$;
T_{G1}: $r(y_1)$; $w(x_2)$;
T_{G2}: $r(y_2)$; $w(x_1)$;

Show a possible non-serializable global schedule in such a system. Show all the serializability orders and explain.