National University of Computer and Emerging Sciences, Lahore Campus

Course Name:	Data Warehousing and Data Mining	Course Code:	CS4058
Degree Program:	BS (Computer Science)	Semester:	Fall 2023
Exam Duration:	60 Minutes	Total Marks:	30
Paper Date:	Fri 10-Nov-2023	Weight	15%
Section:	BCS-7A	Page(s):	6
Exam Type:	Midterm-2 - SOLUTION	Total Questions:	3

Name: ________________________ **Roll No:** _______________

Instructions: Scratch sheets can be used for rough work however, all the questions and steps are to be shown on question paper. ***No extra/rough sheets should be submitted with question paper***.
You will not get any credit if you do not show proper working, reasoning and steps as asked in question statements. *You may use a calculator*.

Q1. *(6 points)* Give the appropriate answers to the following questions:
You manage a data warehousing system for a financial institution that stores transaction data from multiple branches. The data warehouse is updated daily with transaction data from various branches. However, recently, there have been performance issues during data extraction/loading due to the increasing volume of data. Design an efficient extraction/loading strategy for this scenario.

a. Which extraction and loading technique do you prefer in the need to update the data warehouse with new transactions from each branch daily and why?

b. When ETL process be performed to minimize the impact on the source systems and the extraction time? Specify appropriate time window.

c. Branch A uses numeric account numbers, while Branch B uses alphanumeric customer codes as identifiers. To ensure consistency in the data warehouse, you need to restructure these keys. Explain with an example.

Ans: See your textbook.

Consider the following description for next the Questions:

Consider the following tables and statistics which are part of a Library system:

Book *(BookID, Title, Publisher, PublishYear, Author, …);* ***BookLoans*** *(BookID, BranchID, CardNo, DateOut, DueDate, …);*

Assume Book and BookLoans tables containing one *million and twenty million* rows respectively. Each table row and each index entry take *150 bytes* and 15 *bytes of* space respectively. Data block size is 16*KB* and available memory size is *100 blocks*. Suppose selectivity of publisher 'Pearson'= 5%, publisher 'Wiley'= 3%, publishYear '2018'= 3%, publishYear '2020'= 2%, and publishYear '2022'= 1%.

Q2. *(12 points)* Calculate the total I/O cost for the Query using the following indexing techniques. Show all steps clearly.

Query: *SELECT * FROM book WHERE publisher IN ('Pearson', 'Wiley') AND publishYear IN ('2018', '2020', '2022');*

a. Single index access *(Assume Single indexes exist on Publisher and PublishYear columns separately)*
b. Combining multiple indexes *(Assume Single indexes exist on Publisher and PublishYear columns separately)*
c. Composite Index access *(Assume a composite index exist on Publisher and PublishYear columns, with index entry size=15 bytes)*

Answer: ***Combine selectivity of account is 8% of (6% off 1,000,000) = 4800 rows.***

K=100; **B**=16k (i.e. 16,384); **R**=150; **R_i**=15; **bfr**=109 (i.e. $\lfloor B/R \rfloor$=16k/150); **bfr_i=1092** (i.e. $\lfloor B/R_i \rfloor$=16k/15);

r_B=1m; **r_L**=20m;

b_B=9175 (i.e. $\lceil r_B/bfr \rceil$= 1m/109); **b_L=183,487** (i.e. $\lceil r_L/bfr \rceil$= 20m/109);

b_{Bi}=916 (i.e. $\lceil r_B/bfr_i \rceil$= 1m/1092); **b_{Li}=18,316** (i.e. $\lceil r_L/bfr_i \rceil$= 20m/1092);

a) Single index access:

Index cost = Choose highest selectivity column (i.e. publishYear - 6% of rows): 6% of 1m/bfr_i = 60,000/1092 = **55**
As qualifying rows are greater than total no of blocks of book table (i.e. 60,000 > 9175), so we have to read all the blocks of book table.
Total cost = Index access cost + base table access cost
= 55 + 9175 = **9230** I/Os

b) Combining multiple indexes:

Publisher Index cost = 8% of 1m/bfr_i = 80,000/1092 = 74
PublishYear Index cost = 6% of 1m/bfr_i = 60,000/1092 = 55

Total cost = Indexes access cost + base table access cost
= (74+55) + 4800 = **4929** I/O

c) Composite index access:

Combination#1 (Pearson, 2018): 5% of (3% of 1m)= 1500 RIDs and 2 I/Os (i.e. RIDs/bfr_i = 1500/1092)
Combination#2 (Pearson, 2020): 5% of (2% of 1m)= 1000 RIDs and 1 I/O (i.e. RIDs/bfr_i = 1000/1092)
Combination#3 (Pearson, 2022): 5% of (1% of 1m)= 500 RIDs and 1 I/O (i.e. RIDs/bfr_i = 500/1092)
Combination#4 (Wiley, 2018): 3% of (3% of 1m)= 900 RIDs and 1 I/O (i.e. RIDs/bfr_i = 900/1092)
Combination#5 (Wiley, 2020): 3% of (2% of 1m)= 600 RIDs and 1 I/O (i.e. RIDs/bfr_i = 600/1092)
Combination#6 (Wiley, 2022): 3% of (1% of 1m)= 300 RIDs and 1 I/O (i.e. RIDs/bfr_i = 300/1092)
Total Index access cost = 2+1+1+1+1+1 = **7**
Total I/Os (Index access cost + base table access cost) = 7 + 4800 blocks

Roll No: _______________

Q3. *(12 points)* Calculate the total I/O cost for the Query using the following joining techniques. Show all steps clearly. *Assume there is a **clustered index on BookID** column of BookLoans table, but there is no index on BookID column of Book table.*

Query: *SELECT * FROM Book JOIN BookLoans ON Book.BookID = BookLoans.BookID*
WHERE publisher IN ('Pearson', 'Wiley') AND publishYear IN ('2018', '2020', '2022');

a. Hash Join
b. Sort Merge Join
c. Nested Loop Join *(Identify the most efficient variant of NLJ in this scenario, then compute the I/O cost of that variant only.)*

Answer:

a) Hash Join

Book's filter cost + hashing cost
= 9175 + (45 + 183,487) [As the memory is sufficient to store build-input table, so recursive partition is not required.]
= **192,707** I/Os.

b) Sort Merge Join

Book's filter cost + (sort Book) + (merge cost)
= 9175 + (45) + (45 + 183,487) [No sorting required for BookLoans due to clustered index on joining column.]
= **192,752** I/Os.

c) Nested Loop Join

Best NLJ is **Clustered NLJ**, as there is a clustered index exist on joining column.
Book's filter + (qualifying blocks of outer table * (clustered index access cost + inner base table access cost))
= 9175 + (4800 * (1+1)) = 9175 + 9600= **18,775** I/Os.