


National University of Computer and Emerging Sciences, Lahore Campus

	Course Name:	Data Warehousing & Data Mining	Course Code:	CS409
	Program:	BS(CS)	Semester:	Fall 2018
	Duration:	3 Hours	Total Marks:	50
	Paper Date:	Thu 27-Dec-2018	Weight	40%
	Section:	CS	Page(s):	6
	Exam Type:	Final		

Student : Name: _____ **Roll No.** _____

Section: CS

- Instructions/Notes:**
1. Scratch sheet 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.
 2. You will not get any credit if you do not show proper working, reasoning and steps as asked in question statements.
 3. Calculators are ALLOWED.

Q1. (2+2+2+3+3+3= 15 points) Give the appropriate answers of the following questions very briefly:

- a.** List any two categories of NOSQL systems with at least one name of NOSQL system in each category.

Ans:

Document-based NOSQL systems: MongoDB, CouchDB

NOSQL key-value stores: DynamoDB (Amazon)

Column-based NOSQL systems: BigTable

Graph-based NOSQL systems: Neo4J, GraphBase

Hybrid NOSQL systems: Cassandra (Facebook)

Object databases

XML database

- b.** What is the CAP theorem? Which of the three properties (consistency, availability, partition tolerance) are most important in NOSQL systems?

Ans: The CAP theorem states that it *is not possible to guarantee all three* of the desirable properties—consistency, availability, and partition tolerance—at the same time in a distributed system with data replication. If this is the case, then designer would have to choose two properties out of the three to guarantee.

Weaker consistency level is often acceptable in NOSQL distributed data store, **guaranteeing availability and partition tolerance more important**, eventual consistency often adopted.

- c.** Which type of OLAP can handle large amounts of data (i.e. no data limitation)? Why?

Ans: ROLAP; ROLAP storage structure based on relational database, which are designed to support large data sets.

d. Explain the difference between initial data load and full data refresh. When do you use these loading strategies?

Ans: Initial Load: populating all the data warehouse tables for the very first time.

Full data refresh: completely erasing the contents of one or more tables and reloading with fresh data (initial load is a refresh of all the tables).

e. Name any five types of activities that are part of the ETL process. Which of these are time consuming?

Ans:

Determine all the target data needed in the data warehouse.

Determine all the data sources, both internal and external.

Prepare data mapping for target data elements from sources.

Establish comprehensive data extraction rules.

Determine data transformation and cleansing rules.

Plan for aggregate tables.

Organize data staging area and test tools.

Write procedures for all data loads.

ETL for dimension tables.

ETL for fact tables

- Activities in **bold** font are time consuming.

f. Name four major flavors of materialized views. Which of these two are apply only to shared nothing RDBMS implementations?

Ans: Join indexing, aggregate join indexing, reversal tables, replicated tables

Reversal tables and replicated tables are apply only to shared-nothing RDBMS implementations.

Q2. (5 points)

A database has ten transactions.

TID	Items-Purchased
101	milk, bread, eggs
102	milk, juice
103	juice, butter
104	milk, bread, eggs
105	coffee, eggs
106	coffee
107	coffee, juice
108	milk, bread, cookies, eggs
109	cookies, butter
110	milk, bread

The set of items is {milk, bread, cookies, eggs, butter, coffee, juice}. Find all frequent itemsets using Apriori algorithm with $\text{min_sup}=3$, i.e., any itemset occurring in less than 3 transactions is considered to be infrequent. Also list all of the strong association rules with $\text{min_sup}=3$ and $\text{min_conf}=80\%$.

Ans:

L1: milk(5), bread(4), cookies(2), eggs(4), butter(2), coffee(3), juice(3);

L2: {milk,bread}(4), {milk,eggs}(3), {milk,coffee}(0), {milk,juice}(1), {bread,eggs}(3), {bread,coffee}(0), {bread,juice}(0), {eggs,coffee}(1), {eggs,juice}(0), {coffee,juice}(1);

L3: {milk, bread, eggs}(3);

**F= { milk→bread (80%, 4/5)
bread→milk (100%, 4/4)
{milk,eggs}→bread (100%, 3/3)
{bread,eggs}→milk (100%, 3/3)
}**

Q3. (10 points)

For a retail company, design a star (i.e. multi-fact star) schema that includes a base fact table to track the purchase quantity with four dimension tables i.e. time, store, product, and supplier. Also build a 1-way aggregate fact table and a 2-way aggregate fact table. List the possible attributes for each of the dimension tables. Your supplier dimension table must support the history of changes to the supplier. Show the primary keys, foreign keys and all the relationships between the dimension and fact tables. Note: Draw only one diagram that includes base fact table as well as aggregate fact tables.

Ans: Text Book Ch10.

Consider the following description for next two Questions i.e. Q4 and Q5:

Consider the following tables and statistics which are part of a student registration system:
Student (RollNo, Name, gpa, DeptID, BatchID, DegreeID,); Attendance (RollNo, CourseCode, Semester, AttFlag,);

Assume student and attendance tables containing 128,000 and 1,280,000 rows respectively (Student:Attendance ratio is 1:10). Each table row and each index entry takes 128 bytes and 8 bytes space respectively. Data block size is 8KB and available memory size is 100 blocks. Suppose *DeptID*= ('CS' or 'EE') has a selectivity of (40% + 20%) and *BatchID*= ('2015' or '2014') has a selectivity of (5% + 2%) and. Assume secondary (traditional B-tree based) single indexes exist on *student.DeptID*, *student.BatchID*, *student.RollNo*, and *attendance.RollNo* columns separately. A composite secondary (traditional B-tree based) index is also exist on (*DeptID*, *BatchID*) columns of student table.

Q4. (10 points)

How many data blocks need to be accessed to answer the following query?

```
SELECT DeptID, BatchID, COUNT(*) FROM student
WHERE (DeptID='CS' OR DeptID='EE') AND (BatchID='2015' OR BatchID='2014')
GROUP BY DeptID, BatchID;
```

Examine and use the best possible access path. Justify your selection and show all steps clearly.

Ans:

$R=128$; $R_i=8$; $r_{std}=128,000$; $r_{attn}=1,280,000$; $B=8K$; $K=100$; $bfr=64$; $bfr_i=1024$; $b_{(std.)}=2000$; $b_{(attn.)}=20,000$; $bi_{(std.)}=125$; $bi_{(attn.)}=1250$

Combine selectivity of student is 60% of (7% of (128,000))) = 5376 rows.

1- Best Path: Composite index cost = 3+1+2+1= 7 (base table access not required)

[I/O cost for combination:

c1: 40% of 5% of 128,000 = 2560/1024 = 3

c2: 40% of 2% of 128,000 = 1024/1024 = 1

c3: 20% of 5% of 128,000 = 1280/1024 = 2

c4: 20% of 2% of 128,000 = 512/1024 = 1]

2- Combining Multiple Indexes cost = 75 + 9 = 84 (base table access not required)

[deptID (CS or EE) 60%= 76,800/1024=75, batchID (2015 or 2014) 7%= 8960/1024=9; base table access is not required.]

3- FTS cost = 2000

4- Single Index cost (index + base table)= 9 + 2000

[using best selectivity column batchID (2015 or 2014) 7%= 8960/1024=9; base table access is also required].

5- Static bitmap index cost: (which is not given in question here)

One bitmap access cost = 128,000/(1024*8*8)= 2 block

Total cost (to access 4 bitmaps only) = 8 blocks

Q5. (10 points)

How many data blocks need to be accessed to answer the following query?

```
SELECT * FROM student JOIN attendance ON student.rollno=attendance.rollno
WHERE (DeptID='CS' OR DeptID='EE') AND (BatchID='2015' OR BatchID='2014');
```

You are supposed to filter the condition first and then join. Examine and use the best possible joining technique. Justify your selection and show all steps clearly.

Ans: $R=128$; $R_i=8$; $r_{std}=128,000$; $r_{attn}=1,280,000$; $B=8K$; $K=50$; $bfr=64$; $bfr_i=1024$; $b_{(std.)}=2000$; $b_{(attn.)}=20,000$; $bi_{(std.)}=125$; $bi_{(attn.)}=1250$

Combine selectivity of student is 60% of (7% of (128,000))) = 5376 rows.

Student table blocks after considering qualifying rows = $5376/62 = 87$

- Best option is

1- HJ (It's best case of hash join, as build input table size is less than available memory.):

Filtering Cost of student + Hashing Cost

$2000 + (87 + 20,000) = 22,087$

2- SMJ (poor join as both tables need to be sorted):

Filtering Cost + SORT student-filtered table + SORT attendance table + Merge Cost

$= 2000 + 87 + (20,000 * \log(20,000/100)) + (87 + 20,000)$

$= 2000 + 87 + (20,000 * 8) + (87 + 20,000) = 182,174$

3- NLJ (poor join as the selectivity is not good):

= student's filter cost + qualifying rows * (attendance index access + base table access)

$= 2000 + 5376 * (\log(1280,000/1024)=11 + 10) = 2000 + 112,896 = 114,896$