Roll No. _____Name _____ Section _CS_

National University of Computer and Emerging Sciences, Lahore Campus

Course:	**Data Warehousing & Data Mining**	**Course Code:**	**CS409**
Program:	**BS(Computer Science)**	**Semester:**	**Fall 2017**
Duration:	**3 Hours**	**Total Marks:**	**50**
Paper Date:	**12-Dec-17**	**Weight**	**40%**
Section:	**CS**	**Page(s):**	**7**
Exam:	**Final Exam**		

Instruction/Notes: 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.
You will not get any credit if you do not show proper working, reasoning and steps as asked in question statements. CALCULATORS are ALLOWED.

Q1. *(2+2+3+3= 10 points)*
Give the appropriate answers of the following questions very briefly:

a. What are the different types of OLAP? What are the operations of OLAP?

- MOLAP, ROLAP, HOLAP, DOLAP - Slice & Dice, Drill down, Roll up, Pivot

b. What is the difference between ELT and ETL?

*Self

c. When are materialized views useful? What is the use of query rewrite in materialized view?

When there are mutliple large tables to be joined frequently and/or complex aggregation is required on large table(s).
Overview of Query Rewrite. ... One of the major benefits of creating and maintaining materialized views is the ability to take advantage of query rewrite, which transforms a SQL statement expressed in terms of tables or views into a statement accessing one or more materialized views that are defined on the detail tables.

d. What kind of situations are there where you might want to use degenerated dimensions? Give an example of degenerated dimension.

A degenerate dimension (DD) acts as a dimension key in the fact table, however does not join to a corresponding dimension table because all its interesting attributes have already been placed in other analytic dimensions. Sometimes people want to refer to degenerate dimensions as textual facts, however they're not facts since the fact table's primary key often consists of the DD combined with one or more additional dimension foreign keys.
Degenerate dimensions commonly occur when the fact table's grain is a single transaction (or transaction line). Transaction control header numbers assigned by the operational business process are typically degenerate dimensions, such as order, ticket, credit card transaction, or check numbers. These degenerate dimensions are natural keys of the "parents" of the line items.

Q2. *(3+3+4= 10 points)*
a. Give an example of each of the following data mining functionalities, using a real life database that you are familiar with: classification, clustering, and association.

Classification Examples:
-classify students based on final result.
-classify countries based on climate, or
-classify cars based on gas mileage
- Find all credit applicants who are poor credit risks
Clustering Examples:
- Identify customers with similar buying habits.
Association Examples:
- Find all items which are frequently purchased with milk.
-What products were often purchased together in your supermart?

b. Suppose you have market basket data consisting of *1000* transactions and *30* items. If the support for *item a* is *70%*, the support for *item b* is *40%* and the support for *itemset {a, b}* is *30%*. Let the support and confidence thresholds be *25%* and *50%*, respectively. Compute the confidence of the association rule *{a}→{b}*. Is the rule interesting according to the confidence measure?

Confidence= support of {a,b}/support of {a} = 30%/70% = 43%.
Rule is NOT interesting because confidence is less than 50%.

c. A database has four transactions.

TID	Items-Bought
10	{A, C, D}
20	{B, C, E}
30	{A, B, C, E}
40	{B, E}

Find all frequent itemsets using Aprori algorithm with min_sup=2, i.e., any itemset occurring in less than 2 transactions is considered to be infrequent. Also list all of the strong association rules with min_sup=2 and min_conf=100%.

First scan (1-itemsets)

ItemSet	Sup Count
A	2
B	3
C	3
~~D~~	~~1~~
E	3

L1

ItemSet	Sup Count
A	2
B	3
C	3
E	3

L2 (second scan)

ItemSet	Sup Count
~~{A,B}~~	~~1~~
{A,C}	2
~~{A,E}~~	~~1~~
{B,C}	2
{B,E}	3
{C,E}	2

L3 (third scan)

ItemSet	Sup Count
{B,C,E}	2

F= {A→C B→E, E→B, BC→E, CE→B}

Roll No. _____Name _____ Section _CS_

Consider the following description for next Questions# 3 and Question# 4:

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 64,000 and 640,000 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 10 blocks. Suppose degree= 'MS' has a selectivity of 3%, batch= ('2015' or '2014') has a selectivity of (5% + 2%), and dept= ('CS or 'EE') has a selectivity of (40% + 20%).

Q3. *(10 points)*
How many blocks of data need to be accessed to answer the query:

SELECT AVG(gpa) FROM student JOIN attendance ON student.rollno=attendance.rollno
WHERE DegreeID='MS' AND (BatchID='2015' OR BatchID='2014') AND (DeptID='CS' OR DeptID='EE');

Assume cluster indexes exist on RollNo column of student table and also on RollNo column of attendance table. 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}= 64,000; r_{attn}= 640,000; B=8K; K=10; bfr= 64; bfr_i= 1024; $b_{(std.)}$ = 1000; $b_{(attn.)}$ = 10,000; $bi_{(attn..)}$ = 625
As the combine selectivity of student is 3% of (7% of (60% of (64,000))) = 81 rows, so

- Best option is NLJ (*i.e. selectivity is very high*): cost
= student's filter cost + qualifying rows * (attendance index access cost only; base table access not required)
= 1000 + 81 * (1) = 1081 (if hash index only)
= 1000 + 81 * (log(640000/1024)= **10) = 1810 (if traditional index only)**

= 1000 + 81 * (1+1) = 1162 blocks (if hash index with base table which is NOT required)
= 1000 + 81 * (10+1) = 1891 blocks (for traditional index with base table which is NOT required)

- Poor options: HJ/SMJ; it is the best case of both, so same cost for both.
Hash Join because hash table may fit in memory which requires only 81/64= 2 blocks.
HJ cost = student's filter cost + hashing cost (by reading attendance index only) = 1000 + (**2**+ 625) = **1627**
HJ cost = student's filter cost + hashing cost (by reading attendance table) = 1000 + (**2**+ 10,000) = **11,002**

Sort Merge Join because both joining tables are pre-sorted due to clustered index on joining column, which requires only merge cost.

Roll No. _____Name _____ Section _CS_

Q4. *(10 points)*

How many blocks of data need to be accessed to answer the query:

SELECT COUNT(*) FROM student
WHERE DegreeID='MS' AND (BatchID='2015' OR BatchID='2014') AND (DeptID='CS' OR DeptID='EE');

Suppose three secondary indexes are created on student's attributes *deptID*, *BatchID*, and *DegreeID*. Examine and use the best possible access path. Justify your selection and show all steps clearly.

Ans:

Best path: Using combining multiple indexes path (Base table access is not required here):

Combine selectivity is 3% of (7% of (60% of (64,000))) = 81 rows

Index cost: **dept**(CS or EE) 60%= 38,400/1024=**38**, **batch**(2015 or 20114) 7%= 4480/1024=**5**, **degree**(MS) 3%= 1920/1024=**2**,

Total cost (index access cost only) = 38+5+2 = **45 blocks**

OR- Best path: Static bitmap index: (If allowed; but not given in question here; full credit may be given)

One bitmap access cost = 64000/(1024*8*8)= 1 block

Total cost (to access 5 bitmaps only)= 5 blocks

2- 2nd possible path: Using dynamic bitmap indexes path (base table access is required due to false positives)

Total cost (index access cost + base table cost) = 45+81**= 126 blocks**

3- 3rd possible path: FTS = **1000 blocks**

4- 4th possible path: Using single index access with best selectivity (i.e. degree=3%) =

Total cost (index access cost + base table cost) = (1920/1024= 2) + 1000**= 1002 blocks**

Roll No. _____Name _____ Section _CS_

Q5. *(6+2+2= 10 points)*
Consider the following three dimensions and a fact table:
Customer: customer-ID, Name, gender, city, country, …
Account: account-ID, account-Number, open-Date, account-Type-Code, …
Month: month-End-date-ID, month-Name, calendar-Month, …
Monthly_Account: month-End-date-ID, account-ID, customer-ID, previous-Balance, total-Deposits, total-Withdrawal, available-Balance.

a. Draw the appropriate star schema that includes a base fact table, a 1-way aggregate fact table (along customer dimension), and a 2-way aggregate fact table (along customer and account dimensions). Show the primary keys, foreign keys and all the relationships between the dimensions and fact tables.
b. Identify the full-additive, semi-additive, and non-additive facts, if any, in the above base fact table.
c. Refer to the customer dimension of above star schema. Show the revised customer dimension schema that also preserves the history of changes to the customer.

Ans:

a. Self

b. **Full-Additive** (total-deposits & total-withdrawals) and **Semi-Additive** (previous-balance & available-balance)
c. **Customer**: **customer-Key**, customer-ID, Name, gender, city, country, **start-date**, **end-date**, …

Roll No. _____Name _____ Section _CS_