IMPORTANT: Michigan State University adheres to the policies on academic honesty as specified in **Student Handbook and resource Guide**. Any student found guilty of academic dishonesty may receive a 0.0 for the course. In all such cases, a letter will be sent to the dean of the college in which the student is enrolled.

You **MUST NOT** discuss this exam with anybody else. If you think that a question is not clearly stated, you must not talk to another fellow student or any one else for clarification. Instead, you make your own assumptions and answer the question based on those assumptions. Write those assumptions clearly in your answer.
1. Answer the following questions for the transaction table given below:

The table lists the transactions for sales of items A, B, C, D. Each row of the table indicates the particular items sold in a transaction. 1 in the entry means the item is present in a transaction and 0 means that the item is not present in the transaction. The count indicates the total number of transactions with those particular items.

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>8</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>25</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>6</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>30</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Total:</td>
</tr>
<tr>
<td>100</td>
</tr>
</tbody>
</table>

Assume Min Support = .1

(a) (5 points) In the Apriori algorithm, in the kth iteration it needs to join a pair of frequent k-item sets only if their prefixes of length (k-1) are identical. This step creates frequent item sets of length k+1. Explain with examples, if we did join other frequent item sets whose prefixes of length k-1 are not identical, will not create frequent item sets.

(b) (5 points) Give the frequent 1-item sets

(c) (5 points) Give the candidate frequent 2-item sets.

(d) (5 points) Using the candidate frequent 2-item sets compute the frequent 2-item sets.

(e) (5 points) Give all the frequent item sets and then derive the association rules from the frequent item sets using minimum confidence requirements of .1. Show the details of your work applying optimization heuristics of Apriori Algorithm.
2. In this problem you will apply Map/Reduce paradigm to solve association rule mining problem. Instead of attempting to design optimized rule mining algorithms such as the Apriori Algorithm, you will use a simple algorithm to compute frequent patterns. Finding frequent patterns is the primary time consuming task in association rule generation. At the same time, the number of frequent patterns is very small compared to the total number of transactions. Actual rule generation can be done from this result set of frequent patterns on a single node.

**PROBLEM:** Following problem relates to application of Map/Reduce to find frequent patterns for the transactions given in the table of the previous problem.

*If transaction id of the ith transaction is denoted by \( t_i \), then the first two rows of the following table will generate the following 5 transactions: \( t1:D, t2:D, t3:D, t4:AC, t5:AC \)*

(a) (5 points) Describe briefly an appropriate map function for the problem. Give input and output of your map function.

(b) (5 points) Describe the shuffling that distributes the output of the map function.

(c) (5 points) Describe briefly an appropriate reduce function. Give input and output of your reduce function.

(d) (5 points) Combiner function can optimize map-reduce by pushing some of the reduce operations to map workers. Note that the users are not aware of these optimization. Can this optimization be achieved in this problem? explain your answer.

Assume these transactions are equally partitioned between 4 Map workers. There are 4 Reduce workers.

3. This problem relates to Google Big Table:

(a) (10 points) Explain with examples how rows, column families and columns are defined, implemented, and stored in Google Big table.

(b) (5 points) How is data searched in Google Big Table?

4. In Hadoop distributed file architecture, replication of blocks is done to increase data availability against node/rack failure, maximize read bandwidth while minimizing the cost of writing replicated data.

(a) (5 points) Describe a strategy that Hadoop uses to achieve this optimization.

(b) (5 points) What is Hadoop Rack Awareness and which node handles Hadoop rack awareness.

5. Following questions are related to NoSQL and semi-structured databases.
(a) (15 points) Give an example of AVRO schema and an instance for this schema, 
JASON schema and an instance for this schema, and XML schema and an instance 
for this schema. 
Examples of XML schema is given in the text book (Elmasry and Navathe). 
However, if you use the syntax of DTD that will suffice for this problem. 

(b) (10 points) Compare and contrast at least two key features between these three 
three approaches. 

(c) (5 points) Which one of these three approaches is most effective for key/value 
stores and why. 

6. Following questions relate to XML documents (instances of XML schema). 

(a) (10 points) What is serialization and why ordering of child nodes in XML-Trees 
is important for serialization. 

(b) (5 points) Why are XML documents effective for transmitting across the internet 
over, say, transmitting relational tables across the internet. 

(c) (5 points) What are the advantages and disadvantages of having XML schema (or 
DTD) over not having XML schema (DTD),i.e., create XML document without 
having XML schema. 

(d) (10 points) For the following XML document, give the XML-tree for it. 

    <stockItems><category type=parts><ItemName id='20'>Nuts</ItemName><weight>10grams</weight> 
                                                   <ItemName id='30'>Bolts</ItemName><weight>10grams</weight> 
                                                   </category><category type=tools><ItemName id='100'>Screw Driver </ItemName> 
                                                   <ItemName id='90'>Chain Saw</ItemName></category> 
    </stockItems> 

(e) (5 points) Give XPath for the following query: 
    Get names of all parts weighing less than 16 grams. 

(f) (5 points) Give XQuery for the following query: 
    Create an XML document of all stock items of all categories with only those items 
having ItemName id < 100. 

7. Following question is related to Object-oriented DataBases (ODB). Use the following 
schematic diagram of the Document-TechinalReport-Article databases: 

(a) (5 points) Give the schema for this using ODL syntax. 

(b) (5 points) Create an instance of a database using at least one object of type 
Document, one object of type TechnicalReport and one object of type Article. 
You may use O2 syntax.
(c) (5 points) Implement the following query using OQL syntax for this database:
Give all documents that are not Articles; i.e., documents that are neither articles nor TechnicalReports or documents that are only technical reports (i.e., not just document or just articles).

(d) (5 points) Give one key feature that is present in ODB and not present in Object-relational databases, and one key feature that is present in Object-Relational databases and not present in ODB.

8. Multi-valued attributes in object-relational databases are handled by nested relations which are many tiny relations, each tiny relation containing the multiple values in an attribute. Following is an example of a nested relation.

Ename: employee name, Project: project employee is working on, Dependent: employee dependent, P: project name, O: project office, N: dependent name, A: dependent age.

EMP: Nested Relation for employees

<table>
<thead>
<tr>
<th>Ename</th>
<th>Project</th>
<th>Dependent</th>
</tr>
</thead>
<tbody>
<tr>
<td>N1</td>
<td>p1 o1</td>
<td>n1 a1</td>
</tr>
<tr>
<td></td>
<td>p2 o2</td>
<td>n2 a2</td>
</tr>
</tbody>
</table>
(a) (5 points) Assume that the database is a relational database and not an object-relational database, store the above as a relation in this database.

(b) (5 points) What is the redundancy problem of storing data in this way.

(c) (5 points) Solve this redundancy problem by decomposition of the relation.

(d) (5 points) Does nested relation solve redundancy problem? Do we need decompositions for a nested table? Explain it using the above table.

(e) (5 points) How does REF attributes along with nested relations help in this regard.

9. Following questions are related to homework problems on object-relational.

(a) (5 points) In the EER diagram and answer schema of homework on object-relational, player scores for a game, e.g., soccer game, is not completely captured because a player can switch teams over time.

Indicate why is this information not captured in the EER diagram and how the diagram can be modified to capture this information.

(b) (5 points) Schema EmployeeType defined in ORACLE is ”NOT INSTANTIABLE NOT FINAL;”

Why EmployeeType is not instantiable not final.

(c) (5 points) In the definition of

```
create table Games (  
gid varchar(10),  
team1 REF TeamType,  
team2 REF TeamType,  
team1Score PlayerScoreTableRefType,  
team2Score PlayerScoreTableRefType  
)  
```

why team1 and team2 have been defined as REF type while team1Score and team2Score have been defined as table of REF type. What are these tables called and where are they stored? Do the applications have access to these tables through SQL?

(d) (5 points) In the function getMaxScore what does COUNT in the for loop and DREF in the Select clause do?
(e) Based on the schema definition of the homework, answer the following queries in SQL.

i. (5 points) Get the names of those players who have not scored highest in any games.

ii. (5 points) Get the names of the highest scoring players of all winning teams in league A.

10. Overview of KNN queries is on page 11 of Lecture Indexing Multimedia databases. Most of the details is in lecture KNN-Queries and the KNN paper posted (Nearest Neighbor Queries by Nick Roussopoulos).

(a) (5 points) Justify briefly, why at a minmax distance from the query point a data point is guaranteed to exist.

(b) (5 points) In KNN-search algorithm, how is the first kth candidate nearest neighbor obtained?

(c) (5 points) There are three basic rules, described in the KNN paper, for pruning the search. Which of these rules can be used to derive the first kth candidate nearest neighbor.

11. (a) (15 points) Write Oracle SQL to retrieve the first 10 nearest neighbors (photo_id and photo) of a query image from the table stockphotos created as follows:

```sql
CREATE TABLE stockphotos (photo_id NUMBER,
    photographer VARCHAR2(64),
    annotation VARCHAR2(255),
    photo ORDSYS.ORDImage,
    photo_sig ORDSYS.ORDImageSignature);
```

Following is an example of computing the weighted sum of distances of the features between the query image and a database image.

```sql
ORDSYS.ORDImageSignature.evaluateScore(DBimage_sig,QRYimage_sig,
    'color=1.0,texture=0,shape=0,location=0');
```

(b) (5 points) Describe briefly the type of performance that can be measured for the above retrieval system using recall and precision.

**Precision:** Number of relevant images retrieved divided by the total number of images retrieved by the query.

**Recall:** Number of relevant images retrieved divided by the total number of relevant images in the database (which should have been retrieved).

Relevant images for a given query image are those images which are predetermined to be the answer images for the query image. Knowledge of relevant images for a query image is necessary for performance evaluation of the retrieval system.
12. BIRCH uses a clustering algorithm which is based on the CF-tree. Most clustering algorithms have \( n^2 \) time complexity while BIRCH has \( n \log n \) time complexity where \( n \) is the number of data points to be clustered. \( n \log n \) time complexity for Birch is due to the logarithmic nature of trees. This tree-structure which is created by splitting leaf nodes and growing the tree upwards from the root. Growing tree upwards in this fashion produces balanced tree. This approach has similarity to disk-based multi-dimensional indexing like the R-tree. After all, indexes also group data in leaf nodes based on closeness, like the clustering does. In fact, BIRCH in-memory CF-tree can be mapped into BIRCH disk-based CF-tree by mapping each node of CF-tree into a node on disk. This allows for clustering big data sets. However, accessing nodes on disk takes more time.

Because every time a feature vector (data point) is inserted into the CF-tree, it starts with the root node and finds its way to the bottom, reaching an appropriate leaf node. This is similar to searching in the R-tree. Then, it updates the centroid of an entry in the leaf node. This updated centroid requires updating centroids of the entries (CFs) in the parent node, recursively going all the way up to root node. For example, if the tree is of height 3, this will require 3 node accesses going down the tree and two more accesses going up tree, assuming no nodes are saved in memory buffer.

Buffered Approach to Reduce the Number of node Accesses:

If we keep a buffer in front of each leaf node and collect the inserted points (feature vectors) in the buffer, we can insert the points into leaf only when the buffer overflows. Now all the centroid updates of the CFs in the leaf node are done collectively for all the points in the buffer. Following questions are for disk-based buffered CF-tree.

(a) (10 points) Indicate how much savings in node accesses will occur in buffered approach over non-buffered approach if the height of the tree is \( h \) and the number of data points is \( N \).

(b) (5 points) Buffered approach will delay node splits. What is the impact of delayed node split on cluster quality.

(c) (10 points) Updating centroids of CFs in BIRCH is bottom up. That is, centroid of a CF in leaf node is updated first as a result of an insert, changes of this CF is then propagated to upper level nodes. On the other hand, if this update is done top-down (i.e., centroid of the CF in the root is done first as the data point follows the path to the leaf) will it create the same CF-tree.

(d) (5 points) Will top-down updates make BIRCH more efficient?