Q1. DATA MODELING

Does the relational model adequately model today’s environment in terms of unstructured data, multimedia data, streaming data and particularly the data on the web? Discuss pros and cons of using the relational model and related DBMSs to do so. In particular, point out the shortcomings of the relational model and the difficulties of dealing with these data types in the relational model. What approaches are being proposed to deal with modeling of unstructured data for storage and management in traditional (including relational and object-oriented) database environments? What is the state of the art of query languages and of indexing schemes to deal with such data? Answer the question with some illustrative examples.
Q2. DATABASE DESIGN

A. Assume that a sizeable application exists using a legacy data model such as Hierarchical or Network database management system in an organization which took over 20 man-years of effort to design, implement, test and deploy and has been running for the last 20 years. There is a committee assigned to evaluate whether this application should be moved over to relational environment and you are asked to prepare an “issues to consider” document for it. List a series of issues that you will consider to come up with a go vs. no-go decision on migrating the application. (Be sure to consider user, data, programming, testing and deployment related issues.)

B. Assuming that the decision is made to migrate the existing data and application from the legacy to relational environment, describe the series of steps that the team will go through to convert to relational environment, describe each step briefly and comment on the difficulties involved. Do you feel the technology today provides enough tool support to do these steps? - name some tools and comment on their pluses and minuses.

C. “The relational model loses semantics of the application in mapping from a conceptual schema to relational”— to illustrate this point, do the following. Start with a relational design of any application of your choice and reverse engineer to an EER schema and illustrate how the EER schema explains the application much easier compared to the relational model.
Q3. QUERY PROCESSING

Consider the situation where we can define materialized views over our relational database. Since we store the result of the view physically, we can use this view relation when processing queries rather than using the base relations of our database. Using the materialized view may result in a more efficient query execution plan. In this question, I want you to discuss some of the issues involved with being able to rewrite an SQL query in terms of a materialized view, e.g. when is it correct to do so. In your discussion, you should consider the following cases:

- An SQL query that does not contain any aggregate function and a materialized view that does contain an aggregate function(s) and grouping
- An SQL query that contains aggregate function(s) and grouping, and a materialized view that also contains aggregate function(s) and grouping.

In your discussion include some examples considering the single relation,

```
CustomerSales(CustomerID, ProductID, Month, Sales)
```

where the primary key is underlined and the view definition,

```
Create View V AS
    Select ProductID, Month, Sum(Sales), Count(*) AS NumEntries
    From CustomerSales
    Where ProductID > 500
    Group BY ProductID, Month;
```
Q4. PHYSICAL ORGANIZATION

Consider a relational database management system that needs to provide optimized access for applications that primarily do one of the following:

(a) Read access (e.g., Data warehouses, where some of the columns or aggregate values are returned)

(b) Write access (e.g., Online transaction processing, where new records are inserted)

Discuss different physical storage organizations for a relation that may be better suited for (a), and those better suited for (b). Give some justification to support your storage organization for a relation in each of the two cases.
Q5. INFORMATION RETRIEVAL AND DATA MINING

Suppose that we enumerate all words that appear in the Web, w1, w2, w3, … in the order of their frequency. We shall represent a document D by a vector <x1x2x3…x100000> where xj = 1 if word wj+500 appears anywhere in D and 0 otherwise. Our goal is to cluster documents so that documents on the same topic are likely to appear in the same cluster.

(a) Our scheme for representing documents by vectors appears to ignore the 500 most frequent words. Why might it make sense to do so?

(b) One of the major distinctions among clustering algorithms is whether the distance measure is based on a Euclidean space or is a non-Euclidean distance measure. Explain the difference.

(c) Give an example of a Euclidean space and distance measure that could be applied to documents as described above.

(d) Define a non-Euclidean distance measure on these documents. Suggest one such measure and explain why it might yield a good clustering.

(e) Would you generally prefer the Euclidean or non-Euclidean distance measure you have defined to cluster Web documents? Elaborate your answer.
XML has been considered by many as the lingua franca of the data today in both business and governments, though the large amount of business transactions are still stored and managed using the relational data model and RDBMS such as Oracle, DB2 or SQL server.

(a) Comment on the pros and cons of XML representation of data in terms of data modeling, data storage, data processing, and data transportation.

(b) Show two examples of XML applications that demonstrate the advantages of XML over relational data model.

(c) Show two examples of relational applications that demonstrate the disadvantage of using XML as the representation and processing data format.

(d) Design an XML based DBMS system and explain three unique challenges of developing an XML DBMS compared with relational DBMS.

(e) Describe your ideas or solutions to address the above three challenges.
Q7. DATABASE TECHNOLOGIES AND APPLICATIONS

Many technologies related to the database area are affecting the overall design and implementation of database applications. In particular, answer the following:

1) Why should one use an ETL (Extract/Transform/Load) Tool during a Data Warehousing Project? What is their typical functionality? What new research is needed to deal with extracting data from non-conventional sources?

2) What approaches are available for interoperating multiple database applications – in terms of sharing of storage, processing and mediating between DBMSs themselves? What is the state of the art of the “middleware” technology when it comes to database applications?

3) Comment on any favorite area of application that interests you besides the ones alluded to in 1) and 2). Point out the challenging research issues and the solution approaches being pursued.
Two-Phase Locking (2PL) is the commonly used concurrency control method in production relational database management systems. Transactions using the general version of 2PL acquire all the locks before they release any of the locks. The main variant of 2PL adopted by production systems is the Strict 2PL, which releases the locks only after commit/abort decision. Explain the advantages and disadvantages of Strict 2PL, as compared to the general 2PL, in terms of:

1. Total amount of concurrency allowed during transaction execution.
2. Complexity of recovery algorithms, particularly when aborts happen.
3. Using Strict 2PL as a distributed concurrency control mechanism in a distributed environment. [Hint: Herlihy’s global/local properties help explain the problem with general 2PL.]
Q9. CONCURRENCY CONTROL AND RECOVERY (2)

The Two-Phase Commit (2PC) protocol guarantees atomicity property in distributed databases, but 2PC has been considered too restrictive for many practical situations.

1. One reason 2PC is considered too restrictive is the time it forces the participants to maintain locks. This is particularly the case when the participants use Strict 2PL (see question above). Use quantitative arguments (e.g., the number of messages used by 2PC and its duration) to explain this problem for short (sub-second) atomic transactions.

2. Another reason 2PC is considered too restrictive is the atomicity property, which cannot be easily sub-divided. Explain the problem caused by atomicity using two illustrative distributed applications that contain long duration activities (hours, days or months).

3. Compare the atomicity property guaranteed by 2PC (e.g., as specified by the WS-AtomicTransaction protocol) with an extensible model (e.g., as specified by the WS-Coordination protocol). Your comparison should be concrete (e.g., in terms of data structures, protocol steps, and applications they serve) and limited to two major differences.