Consider the following relational schema:

```
Employee (ID, Lname, Fname, Salary, Dnumber, City)
```

where ID is the primary key. The number of rows in the table is 100,000 and the number of pages used to store the table is 10,000. We have the following two indexes for the table.

- DINDEX: a sparse clustered index on Dnumber with 100 leaf nodes
- SINDEX: a nonclustered index on Salary with 200 leaf nodes

DINDEX consists of 2 levels from the root to a leaf (inclusive) where SINDEX consists of 3 levels. A leaf contains pointers to data pages. Random access to a single node in the index or a single data page requires 6ms. Sequential access to 50 contiguously stored pages (or index nodes) requires 10ms. We can assume that all data pages are stored contiguously and that all leaf nodes in an index are stored contiguously.

Consider the following query:

```
SELECT    ID
FROM       EMPLOYEE
WHERE   (Salary = 25000 OR Salary = 350000) AND Dnumber = 5;
```

Assume the following:
- 1% of the employees work in department number 5,
- Salary values are 25,000 or 50,000 or 100,000 or 150,000 or 200,000 or 250,000 or 300,000 or 350,000 and are uniformly distributed over all employees

Describe the execution plan and its cost (I/O time) that uses

(a) only the DINDEX index

(b) only the SINDEX index

(c) both the DINDEX and SINDEX indexes

Also, be sure to specify and justify any additional assumptions you make. (Please provide answers for (a), (b) and (c).)
Question 2.

Please answer (a) and (b).

Consider the following simple computer system to support our database server.

We can assume that the disk controller, bus and main memory can handle the data transfer rate of the three disks. We are given several tables to store in our database and a set of queries that access one or more tables along with a frequency of occurrence for each query.

(a) Describe two scenarios that would result in a different placement or partitioning of the tables on the available disks.

(b) Are file organizations like B+ trees and extendible hash files still useful? Support your answer.

Question 3.

Assume that you are developing an information system for the Department of Homeland Security to extract data from the databases of several other agencies. Explain - in details - which role XML could play in this system. You should assume that the current existing systems in each agency will not be replaced, but will need to be integrated in the new system DHS is building.

Explain - step by step - what would need to be done to design and implement such a system. Discuss any problems you foresee in this process.

Provide a drawing and explanation of the architecture and functions of such a system.
Question 4.

Dependency theory has played a major role in the development of the theory of relational design. Point out the different types of dependencies that have been defined for the relational data model. For each of them show with an example each how that dependency is used to define a particular aspect/property of the design. (example as a hint: think of 2NF, 3NF, BCNF as an offshoot of a certain type of dependency.). Can you define certain dependencies for the object model and for the XML as a data model – why or why not? Take a firm position and defend or oppose the following statement: “The notion of normalization does not make sense for the OO and XML models.”

Question 5.

Please answer (a), (b) and (c):

(a) Data Warehousing is typically associated with ETL (extract/transform/load) and OLAP tools. Comment on the typical capabilities of these tools and the current shortcomings of the ETL tools.

(b) OLAP uses multidimensional data model. Compare and contrast the multidimensional data model with the standard relational data model. Consider operations like roll-up and drill-down applied to data in a warehouse. What other operations can you think of as the basis of an algebra for cubes? Why or why not?

(c) Consider the basis relational algebra operations of join and union. Propose analogous operations on data cubes in a precise manner.
**Question 6.**

Please answer either (a) or (b):

(a) Web Information Retrieval and Database are two different research communities with some problems of common interest. Examples include efficient retrieval in the context of the WWW and indexing of large collection of documents. Please comment on (1) unique techniques that distinguish WebIR from database systems and (2) unique techniques that distinguish DB research and DB systems from Web IR research. You are encouraged to use applications you have experienced with to elaborate on your answers.

(b) Assume that you have access to several document databases organized with documents in plain text, HTML, XML, or formatted data like relations, etc. You have been asked to develop a retrieval system that will process general queries against this collection of databases. You are asked to first classify the collection by topics and then to develop a query formulation framework to be able to retrieve documents of interest as defined by the query. Describe the issues and the approaches related to the following:
   (1) integration of the documents,
   (2) classification of the documents into topics,
   (3) designing a language to query the documents, and
   (4) efficient execution of the queries.

**Question 7.**

Please answer (a), (b) and (c):

Snapshot Isolation is a widely used non-serializable protocol (e.g., supported by commercial databases such as Oracle).

(a) Succinctly and informally define Snapshot Isolation. Show an example of schedule (sequence of R/W operations) that satisfy Snapshot Isolation but is not serializable.

(b) Compare Snapshot Isolation with the four levels of isolation originally defined for 2-phase locking.

(c) A schedule that satisfies Snapshot Isolation can be serializable (the TPC-C benchmark has been shown to be serializable when running on Oracle). Show a concrete sequence that satisfies both the Snapshot Isolation and serializability.
Question 8.

Please answer (a), (b) and (c).

Consider a shared-nothing parallel database system running on a cluster of machines. The machines are co-located in the same facility, and are connected by a high-speed network. The database system is intended to support an OLTP workload, and we are concerned with both correctness and throughput.

(a) What concurrency control mechanisms would you recommend for this parallel database system? What are the advantages and disadvantages of this approach?

Now imagine that we are running the same application, but the database nodes are geographically dispersed (e.g., in different offices of a multinational corporation.) The connection between the databases is the Internet.

(b) Would you use the same concurrency control scheme as in part (a)? If so, justify why that scheme is still a good idea. If not, why not, and what would you change?

Now imagine that the application is running on a peer-to-peer system, with geographically dispersed nodes, not all of which are owned by the same entity and not all of which are available at all times.

(c) Would your scheme from part (b) work? If so, justify why that scheme is still a good idea. If not, why not, and what would you change? Would we need to re-examine our requirements of correctness and throughput?
**Question 9.**

Please answer both (a) and (b).

Traditional database systems assume data is stored on disk, and queried by reading data from disk into memory. However, many recent projects have explored systems that never actually store the base data, although they may store intermediate or final query results. Let us refer to such projects as “data stream systems.” Examples range from systems for querying sensor networks, to monitoring status streams, to querying event-based systems, and so on.

(a) What are the key architectural differences between “data stream systems” and traditional stored-data database systems? (You do not need to list all differences, focus on the key differences.) What about data streams motivates the need for these differences?

(b) What are the key architectural similarities between “data stream systems” and traditional stored-data database systems? In other words, which components of a traditional system could you reuse (possibly with minor modification) in an implementation of a data stream system?

In your answer, please give examples of specific systems from the literature to illustrate the architectural similarities and differences.