1. (a) Explain Query Processing in distributed databases.

Give diagram for query processing phases with explanation

![Diagram of query processing phases]

(b) What is Need of Document Oriented database

Document databases make it easier for developers to store and query data in a database by using the same document-model format they use in their application code. The flexible, semistructured, and hierarchical nature of documents and document databases allows them to evolve with applications' needs.

Document-oriented databases allow users to store, retrieve, and manage data and documents. Organizations usually use NoSQL document databases to store semi-structured or unstructured data such as user data, messaging data, device data, images and videos.

Eg. NoSQL Database Types. Document databases pair each key with a complex data structure known as a document. Documents can contain many different key-value pairs, or key-array pairs, or even nested documents. Graph stores are used to store information about networks of data, such as social connections.

(c) Explain cost based query optimization.
Explanation with example

(d) What is SQL Injection? Given one example

SQL injection is a web security vulnerability that allows an attacker to interfere with the
<table>
<thead>
<tr>
<th>SQL injection examples</th>
<th>UNION attacks, where you can retrieve data from different database tables</th>
</tr>
</thead>
<tbody>
<tr>
<td>There are a wide variety of SQL injection vulnerabilities, attacks, and techniques, which arise in different situations. Some common SQL injection examples include:</td>
<td></td>
</tr>
<tr>
<td>- Retrieving hidden data, where you can modify an SQL query to return additional results.</td>
<td></td>
</tr>
<tr>
<td>- Subverting application logic, where you can change a query to interfere with the application's logic.</td>
<td></td>
</tr>
</tbody>
</table>

There are many different situations where SQL injection can occur. Some common examples include:

- Retrieving hidden data, where you can modify an SQL query to return additional results.
- Subverting application logic, where you can change a query to interfere with the application's logic.
- UNION attacks, where you can retrieve data from different database tables.

**2. What is External sorting in Query processing?**

External sorting refers to sorting algorithms that are suitable for large files of records stored on disk that do not fit entirely in main memory, such as most database files. The typical external sorting algorithm is the sort-merge strategy, which starts by sorting small subfiles—called runs—of the main file and then merges the sorted runs, creating larger sorted subfiles that are merged in turn. The sort-merge algorithm, like other database algorithms, requires buffer space in main memory, where the actual sorting and merging of the runs is performed. The basic algorithm, outlined in Figure 19.2, consists of two phases: the sorting phase and the merging phase. The buffer space in main memory is part of the DBMS cache—an area in the computer's main memory that is controlled by the DBMS. The buffer space is divided into individual buffers, where each buffer is the same size in bytes as the size of one disk block. Thus, one buffer can hold the contents of exactly one disk block.
In the sorting phase, runs (portions or pieces) of the file that can fit in the available buffer space are read into main memory, sorted using an internal sorting algorithm, and written back to disk as temporary sorted subfiles (or runs). The size of each run and the number of initial runs (nR) are dictated by the number of file blocks (b) and the available buffer space (nB). For example, if the number of available main memory buffers nB = 5 disk blocks and the size of the file b = 1024 disk blocks, then nR = (b/nB) or 205 initial runs each of size 5 blocks (except the last run which will have only 4 blocks). Hence, after the sorting phase, 205 sorted runs (or 205 sorted subfiles of the original file) are stored as temporary subfiles on disk.

In the merging phase, the sorted runs are merged during one or more merge passes. Each merge pass can have one or more merge steps. The degree of merging (dM) is the number of sorted subfiles that can be merged in each merge step. During each merge step, one buffer block is needed to hold one disk block from each of the sorted subfiles being merged, and one additional buffer is needed for containing one disk block of the merge result, which will produce a larger sorted file that is the result of merging several smaller sorted subfiles. Hence, dM is the smaller of (nB - 1) and nR, and the number of merge passes is \( \log_{dM}(nR) \). In our example where nB = 5, dM = 4 (four-way merging), so the 205 initial sorted runs would be merged 4 at a time in each step into 52 larger sorted subfiles at the end of the first merge pass. These 52 sorted files are then merged 4 at a time into 13 sorted files, which are then merged into 4 sorted files, and then finally into 1 fully sorted file, which means that four passes are needed.

(b) ANS:

Horizontal fragmentation

Select * from Department where location='Dadar';

Completeness, Reconstruction

vertical fragmentation

Select Eid, Ename from Employee where Dno=10;

Completeness, Disjointness, Reconstruction

3. (a) Explain Three Phase Commit Protocol in detail
XML definition and use

What is XSLT? Also explain XML schema document with example.
4. (a) Draw and explain architecture for distributed database system

Diagram and explanation

(b) Explain different types of Spatial Data models

Data can be defined as verifiable facts.

- Information is data organized to reveal patterns, and to facilitate search.
- Spatial information is difficult to extract from spatial data, unless the data are organized primarily by spatial attributes.
- Spatial objects are characterized by attributes that are both spatial and nonspatial, and the digital description of objects and their attributes comprise spatial datasets.

**ZERO DIMENSIONS POINTS**

**ONE DIMENSION LINES**

**TWO DIMENSIONS AREAS**

**THREE DIMENSIONS VOLUMES**

**NATURAL SPATIAL OBJECTS, IMPOSED SPATIAL OBJECTS**

**RASTER MODEL**

Spatial data of different types can be overlaid without the need for the complex geometric calculations.

- Each layer of grid cells in a raster model records a separate attribute.
- The cells (pixels, picture elements) are constant in size and are generally square.
- It is unnecessary to store the coordinates of each cell as the cells are arranged in a regular pattern.
- It is enough to determine the cell size and the parameters to transform the X and Y coordinates and the cell locations in the raster map (rows/lines and columns). This process is called georeferencing.

Vector modeling
Temporal databases are in contrast to static databases (not to be confused with currently
This approach introduces additional complexities:

- Decision time
- Transaction time
- Valid time

A T-Temporal database has three axes of time:

- T-Temporal
- Transaction time or decision time
- Valid time

A B-Temporal database has two axes of time:

- B-Temporal
- Range

A Uni-Temporal database has one axis of time, either the validity range or the system time

Uni-Temporal

A decision is the time period during which a fact stored in the database was decided to be valid.

Known

- Decision time is the time period during which a fact stored in the database was decided to be valid.
- Transaction time is the time period during which a fact stored in the database was.
- Valid time is the time period during which a fact is true in the real world.

Temporal databases usually include valid time, transaction time or decision
more specifically the temporal aspect of T-Temporal.
Temporal databases store information relating to past, present and future time. Temporal databases could be uni-
A Uni-Temporal database stores data relating to time instances. It offers temporal data types and

Write a short note on Temporal Data Model

maps

especially useful dealing with map overlays. You can rasterize the vector maps into raster
for creating high quality output. They are less suitable for a number of GIS operations,
for digitizing or importing data. Vector maps require less disk space and are suitable
Vector maps are point, segment or polygon maps. Most of the maps are obtained by
available databases), which store only facts which are believed to be true at the current time.

Features

Temporal databases support managing and accessing temporal data by providing one or more of the following features:[1][2]

- A time period datatype, including the ability to represent time periods with no end (infinity or forever)
- The ability to define valid and transaction time period attributes and bitemporal relations
- System-maintained transaction time
- Temporal primary keys, including non-overlapping period constraints
- Temporal constraints, including non-overlapping uniqueness and referential integrity
- Update and deletion of temporal records with automatic splitting and coalescing of time periods
- Temporal queries at current time, time points in the past or future, or over durations

Predicates for querying time periods, often based on Allen’s interval relations

(b) Explain Discretionary access control in detail.

The typical method of enforcing discretionary access control in a database system is based on the granting and revoking privileges.

The account level: At this level, the DBA specifies the particular privileges that each account holds independently of the relations in the database.

The relation (or table level): At this level, the DBA can control the privilege to access each individual relation or view in the database.

The privileges at the account level apply to the capabilities provided to the account itself and can include the CREATE SCHEMA or CREATE TABLE privilege, to create a schema or base relation; the CREATE VIEW privilege; the ALTER privilege, to apply schema changes such as adding or removing attributes from relations; the DROP privilege, to delete relations or views; the MODIFY privilege, to insert, delete, or update tuples; and the SELECT privilege, to retrieve information from the database by using a SELECT query.

The second level of privileges applies to the relation level, whether they are base relations or virtual (view) relations.

The granting and revoking of privileges generally follow an authorization model for discretionary privileges known as the access matrix model, where the rows of a matrix M represents subjects (users, accounts, programs) and the columns represent objects (relations, records, columns, views, operations). Each position M(i,j) in the matrix
<table>
<thead>
<tr>
<th>6.</th>
<th>Single-Level Ordered Indexes</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Write a short note on (Any 2)</strong></td>
<td></td>
</tr>
</tbody>
</table>

There are several types of ordered indexes. A primary index is specified on the ordering key field of an ordered file of records. Recall from Section 1.7 that an ordering key field is used to physically order the file records on disk, and every record has a unique value for that field. If the ordering field is not a key field—that is, if numerous records in the file can have the same value for the ordering field—another type of index, called an auxiliary index, can be used. The data file is also called a clustered file in this latter case. Notice that a file can have at most one primary index, but not both. A third type of index, called an auxiliary index, can be specified on any non-indexing field of a file. A data file can have several secondary indexes in addition to its primary access method. Also gives explanation of indexes.

| Replication and Allocation Techniques for Distributed Database Design |

Replication is useful in improving the availability of data. The most extreme case is replication of the whole database at every site in the distributed system, thus creating a fully replicated distributed database. This can improve availability remarkably because the system can continue to operate as long as at least one site is up. It also improves performance locally from retrieval of global queries because the results of such queries can be obtained locally from any one site. Hence, a retrieval query can be processed at the local site where it is submitted if that site includes a server module. The disadvantage of full replication is that it can slow down update operations drastically, since a single logical update must be performed on every copy of the database to keep the copies consistent. This is especially true if many copies of the database exist. Full replication makes the concurrency control and recovery techniques more expensive than they would be if there was no replication.

The other extreme from full replication involves having no replication—that is, each fragment is stored at exactly one site. In this case, all fragments must be disjoint, except for the repetition of primary keys among vertical (or mixed) fragments. This is also called nonredundant allocation.

Between these two extremes, we have a wide spectrum of partial replication of the data—that is, some fragments of the database may be replicated whereas others may not. The number of replicated databases with them on laptops and PDAs and synchronize them periodically with the server database. A description of the replication of fragments is sometimes called a replication schema.
The discretionary access control technique of granting and revoking privileges on relations has traditionally been the main security mechanism for relational database systems. This is an all-or-nothing method: A user either has or does not have a certain privilege. In many applications, an additional security policy is needed that classifies data and users based on security classes. This approach, known as mandatory access control (MAC), is important to note that most commercial DBMSs currently provide mechanisms only for discretionary access control. However, the need for multilevel security exists in government, military, and intelligence applications, as well as in many industrial and corporate applications. Some DBMS vendors—for example, Oracle—have released special versions of their RDBMSs that incorporate mandatory access control for government use. Typical security classes are top secret (TS), secret (S), confidential (C), and unclassified (U), where TS is the highest level and U the lowest. Other more complex security classification schemes exist, in which the security classes are organized in a lattice. For simplicity, we will use the system with four security classification levels, where TS ≥ S ≥ C ≥ U, to illustrate our discussion. The commonly used model for multilevel security, known as the Bell-LaPadula model, classifies each subject (user, account, program) and object (relation, tuple, column, view, operation) into one of the security classifications TS, S, C, or U. We will refer to the clearance (classification) of a subject S as $\text{class}(S)$ and to the classification of an object O as $\text{class}(O)$. Two restrictions are enforced on data access based on the subject/object classifications:

1. A subject $S$ is not allowed read access to an object $O$ unless $\text{class}(S) \geq \text{class}(O)$. This is known as the simple security property.

2. A subject $S$ is not allowed to write an object $O$ unless $\text{class}(S) \leq \text{class}(O)$. This is known as the star property (or *-property).

The first restriction is intuitive and enforces the obvious rule that no subject can read an object whose security classification is higher than the subject’s security clearance. The second restriction is less intuitive. It prohibits a subject from writing an object at a lower security classification than the subject’s security clearance. Violation of this rule would allow information to flow from higher to lower classifications, which violates a basic tenet of multilevel security. For example, a user (subject) with TS clearance may make a copy of an object with classification TS and then write it back as a new object with classification U, thus making it visible throughout the system.