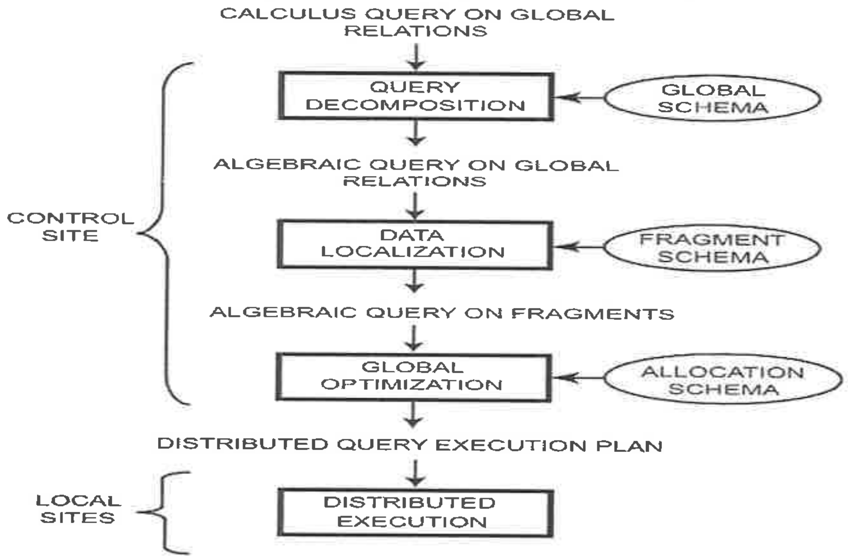
71594

(3 Hours) Total Marks:80

Answer Model

**1. (a)** Explain Query Processing in distributed databases. **2**

Give diagram for query processing phases with explanation




**(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 queries that an application makes to its database. It generally allows an attacker to view data that they are not normally able to retrieve. This might include data belonging to other users, or any other data that the application itself is able to access. In many cases, an attacker can modify or delete this data, causing persistent changes to the application's content or behavior.

In some situations, an attacker can escalate an SQL injection attack to compromise the underlying server or other back-end infrastructure, or perform a denial-of-service attack.

A successful SQL injection attack can result in unauthorized access to sensitive data, such as passwords, credit card details, or personal user information. Many high-profile data breaches in recent years have been the result of SQL injection attacks, leading to reputational damage and regulatory fines. In some cases, an attacker can obtain a persistent backdoor into an organization's systems, leading to a long-term compromise that can go unnoticed for an extended period.

SQL injection examples

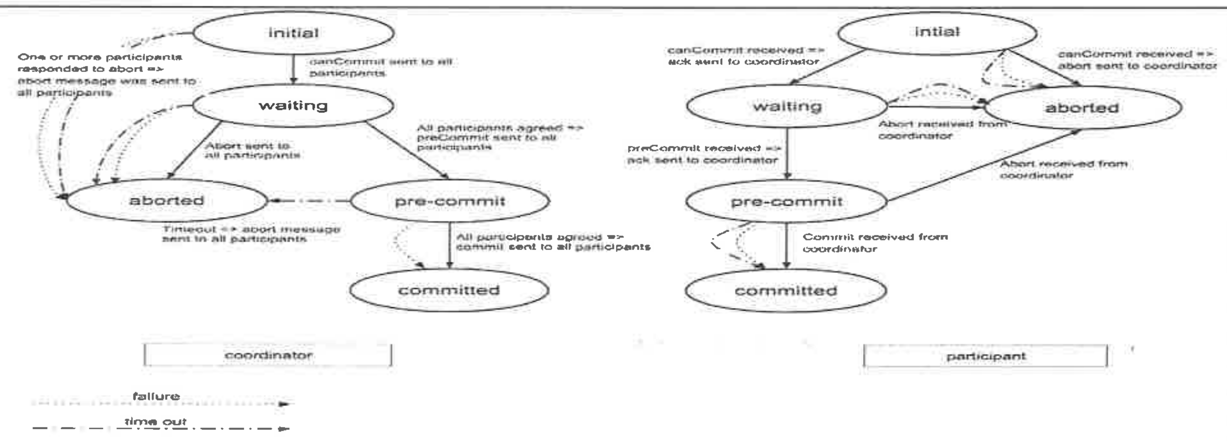
There are a wide variety of SQL injection vulnerabilities, attacks, and techniques, which arise in different situations. Some common SQL injection examples include:

- 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. (a) What is External sorting in Query processing 1(

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 data-base files. The typical external sorting algorithm uses a 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 ($n_R$) are dictated by the number of file blocks (b) and the available buffer space ($n_B$). For example, if the number of available main memory buffers $n_B$ = 5 disk blocks and the size of the file b = 1024 disk blocks, then $n_R$= ($b/n_B$) 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.<br><br>In the merging phase, the sorted runs are merged during one or more mergepasses. Each merge pass can have one or more merge steps. The degree of merging ($d_M$) 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, $d_M$ is the smaller of ($n_B - 1$) and $n_R$, and the number of merge passes is ($\log_{d_M}(n_R)$) . In our example where $n_B$ = 5, $d_M$ = 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:<br><br>Horizontal fragmentation<br><br>Select * from Department where location='Dadar';<br><br>Completeness, Reconstruction<br><br>vertical fragmenattion<br><br>Select Eid , Ename  from Employee where Dno=10;<br><br>Completeness,Disjointness, Reconstruction | 1( |
| | | | |
| 3. | (a) | Explain  Three Phase Commit Protocol in detail | 1( |




(b) What is XML? also explain xml schema document with example. 10

:XML definition and use

```
<xsd:schemaxmlns:xsd=http://www.w3.org/2001/XMLSchema>

<xsd:element name="bank" type="BankType"/>

<xsd:element name="account">

<xsd:complexType>

<xsd:sequence>

<xsd:element name="account-number" type="xsd:string"/>

<xsd:element name="branch-name"      type="xsd:string"/>

<xsd:element name="balance"             type="xsd:decimal"/>

</xsd:squence></xsd:complexType>

</xsd:element>

….. definitions of customer and depositor ….

<xsd:complexType name="BankType"><xsd:squence>

        <xsd:element ref="account"   minOccurs="0" maxOccurs="unbounded"/>

        <xsd:element ref="customer" minOccurs="0" maxOccurs="unbounded"/>

        <xsd:element ref="depositor" minOccurs="0" maxOccurs="unbounded"/>

    </xsd:sequence>

</xsd:complexType>
</xsd:schema>
```

| | | | |
|---|---|---|---|
| 4. | (a) | Draw and explain architecture for distributed database system<br><br>Diagram and explanation | 1 |
| | (b) | Explain different types of Spatial Data models<br>Data can be defined as verifiable facts.<br><br>□Informationis data organized to reveal patterns, and to facilitate search.<br><br>□Spatial informationis difficult to extract from spatial data, unless the data are organized<br><br>primarily by spatial attributes.<br><br>□Spatial objectsare characterized by attributes that are both spatial and nonspatial, and the digital description of objects and their attributes comprise spatial datasets.<br><br>ZERO DIMENSIONS POINTS<br><br>ONE DIMENSION LINES<br><br>TWO DIMENSIONS AREAS<br><br>THREE DIMENSIONS VOLUMES<br><br>NATURAL SPATIAL OBJECTS, IMPOSED SPATIAL OBJECTS<br><br>RASTER MODEL<br><br>Spatial data of different types can be overlaid without the need for the complex geometric calculations.<br><br>□Each layer of grid cells in a raster model records a separate attribute.<br><br>□The cells (pixels, picture elements) are constant in size and are generally square.<br><br>□It is unnecessary to store the coordinates of each cell as the cells are arranged in a regular pattern<br><br>□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<br><br>georeferencing<br><br>Vector modeling | 1 |

| | | | |
|---|---|---|---|
| | | Vector maps are point, segment, or polygon maps. Most of the maps are obtained by digitizing or by importing them. Vector maps require less disk storage space and are suitable for creating high quality outputs. They are less suitable for a number of GIS operations, especially those dealing with map overlaying. You can rasterize the vector maps into raster maps | |
| | | | |
| 5. | (a) | Write a short note on Temporal Data Model<br>A temporal database stores data relating to time instances. It offers temporal data types and stores information relating to past, present and future time. Temporal databases could be uni-temporal, bi-temporal or tri-temporal.<br><br>More specifically the temporal aspects usually include valid time, transaction time or decision time<br><br>• Valid time is the time period during which a fact is true in the real world.<br>• Transaction time is the time period during which a fact stored in the database was known.<br>• Decision time is the time period during which a fact stored in the database was decided to be valid.<br><br>Uni-Temporal<br><br>A uni-temporal database has one axis of time, either the validity range or the system time range.<br><br>Bi-Temporal<br><br>A bi-temporal database has two axes of time.<br><br>• valid time.<br>• transaction time or decision time.<br><br>Tri-Temporal<br><br>A tri-temporal database has three axes of time.<br><br>• valid time.<br>• transaction time<br>• decision time.<br><br>This approach introduces additional complexities.<br><br>Temporal databases are in contrast to current databases (not to be confused with currently | 1( |

| | | | |
|---|---|---|---|
| | | available databases), which store only facts which are believed to be true at the current time.<br><br>Features<br><br>Temporal databases support managing and accessing temporal data by providing one or more of the following features:[1][2]<br><br>• A time period datatype, including the ability to represent time periods with no end (infinity or forever)<br>• The ability to define valid and transaction time period attributes and bitemporal relations<br>• System-maintained transaction time<br>• Temporal primary keys, including non-overlapping period constraints<br>• Temporal constraints, including non-overlapping uniqueness and referential integrity<br>• Update and deletion of temporal records with automatic splitting and coalescing of time periods<br>• Temporal queries at current time, time points in the past or future, or over durations<br>Predicates for querying time periods, often based on Allen's interval relations | |
| | (b) | Explain Discretionary access control in detail . | 1( |
| | | The typical method of enforcing discretionary access control in a database system is based on the granting and revoking privileges.<br><br>The account level: At this level, the DBA specifies the particular privileges that each account holds independently of the relations in the database.<br><br>The relation (or table level):At this level, the DBA can control the privilege to access each individual relation or view in the database.<br>he 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 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.<br><br>The second level of privileges applies to the relation level,whetherthey are base relations or virtual (view) relations.<br><br>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 | |

| | | | |
|---|---|---|---|
| | | represents the types of privileges (read, write, update) that subject i holds on object j. | |
| 6. | | Write a short note on (Any 2 ) | 2( |
| | (a) | Single-Level Ordered Indexes<br><br>There are several types of ordered indexes. A primary index is specified on the ordering key field of anordered fileof records. Recall from Section 17.7 that anordering 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 a clustering index, can be used. The data file is called a clustered file in this latter case. Notice that a file can have at most one physicalordering field, so it can have at most one primary index or one clustering index, butnot both. A third type of index, called asecondary index, can be specified on any nonorderingfield of a file. A data file can have several secondary indexes in additionto its primary access method.<br>Also gives explanation of indexes. | |
| | (b) | Replication and Allocation Techniques for Distributed Database Design.<br><br>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 remarkablybecause the system can continue to operate as long as at least one site is up. It also improves performance of retrieval for 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,<br>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 dis-joint, except for the repetition of primary keys among vertical (or mixed) fragments. This is also called nonredundant allocation.<br>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 copies of each fragment can range from one up to the total number of sites in the distributed system. A special case of partial replication is occurring heavily in applications where mobile workers—such as sales forces, financial plan-ners, and claims adjustors—carry partially replicated databases with them on laptops and PDAs and synchronize them periodically with the server database.[7] A descrip-tion of the replication of fragments is sometimes called a replication schema. | |

**(c)** Mandatory Access Control  Access Control for Multilevel Security

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 mandatoryaccess control (MAC),  It 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.Typicalsecurity 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 class(S) and to the classification of an object O as 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 class(S) ≥class(O). This is known as the simple security property.
2. A subject S is not allowed to write an object O unless class(S) ≤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.