Selection operation in dbms

In RDBMS or relational database systems, the file scan reads a relation only if the whole relation is stored in one file only. When the selection operation is performed on a relation whose tuples are stored in one file, it uses the following algorithms:

A1 (linear search) : Scan each file block and test all records to see whether they satisfy the selection

condition.

A2 (binary search) : Applicable if file is ordered on an attribute and the selection condition is an

equality condition on the attribute.

Index structures provide an access path through which data can be located and accessed

There are two types:-

Primary and Secondary

* **Primary index, equality on a key:** We use the index to retrieve a single record that satisfies the equality condition for making the selection. The equality comparison is performed on the key attribute carrying a primary key.
* **Primary index, equality on nonkey:** The difference between equality on key and nonkey is that in this, we can fetch multiple records. We can fetch multiple records through a primary key when the selection criteria specify the equality comparison on a nonkey.
* **Secondary index, equality on key or nonkey:** The selection that specifies an equality condition can use the secondary index. Using secondary index strategy, we can either retrieve a single record when equality is on key or multiple records when the equality condition is on nonkey. When retrieving a single record, the time cost is equal to the primary index. In the case of multiple records, they may reside on different blocks.
* Selection Operations with Comparisons

For making any selection on the basis of a comparison in a relation, we can proceed it either by using the linear search or via indices in the following ways:

* **Primary index, comparison:** When the selection condition given by the user is a comparison, then we use a primary ordered index, such as the primary B+-tree index. **For example**, when A attribute of a relation R compared with a given value v as A>v, then we use a primary index on A to directly retrieve the tuples. The file scan starts its search from the beginning till the end and outputs all those tuples that satisfy the given selection condition.
* **Secondary index, comparison:** The secondary ordered index is used for satisfying the selection operation that involves <, >, ≤, or ≥ In this, the files scan searches the blocks of the lowest-level index.  
  **(< ≤):** In this case, it scans from the smallest value up to the given value v.  
  **(>, ≥):** In this case, it scans from the given value v up to the maximum value.  
  However, the use of the secondary index should be limited for selecting a few records. It is because such an index provides pointers to point each record, so users can easily fetch the record through the allocated pointers.

SORTING ALGORITHMS IN DBMS

Sorting is required when the query has orderby in it This requires multiple disk access, i.e, disk access for each record to be retrieved Such a method is suitable when the number of records are less. when the data to be sorted is large, i.e, more disk access is required it is better to perform physical sorting.

Physical Sorting means, getting blocks of data in main memory and sorting it.

However, if the blocks of data that you want to sort fit into the main memory, then we perform internal sort eg: quick sort

However, if the number of blocks to be sorted do not fit into the main memory, then we

apply external sorting The most commonly used technique is External Sort-Merge Algorithm

External Sort-Merge Algorithm

Has 2 passes

Sort phase: The records in the file to be sorted are divided into several groups called run. Such

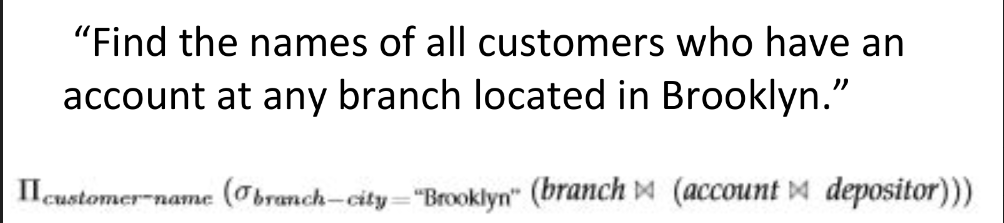
that each run fits into a buffer.

Merge phase: The sorted runs are merged.

(example done in book)

Query optimization is used to access and modify the database in the most efficient way possible. It is the art of obtaining necessary information in a predictable, reliable, and timely manner. Query optimization is formally described as the process of transforming a query into an equivalent form that may be evaluated more efficiently. The goal of query optimization is to find an execution plan that reduces the time required to process a query. We must complete two major tasks to attain this optimization target.

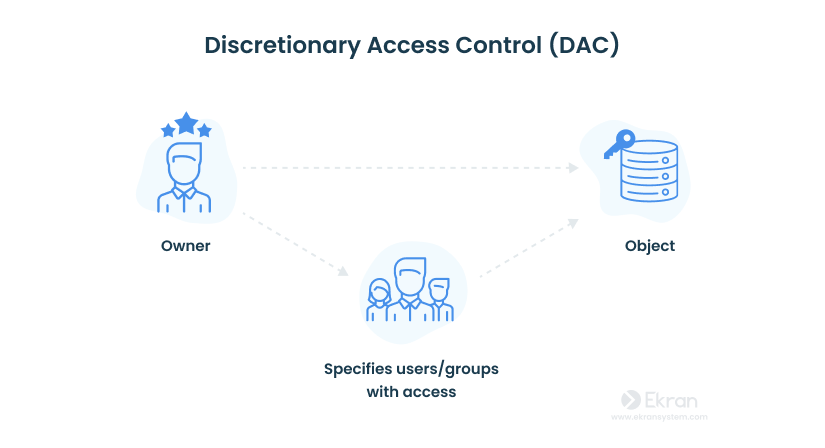
The first is to determine the optimal plan to access the database, and the second is to reduce the time required to execute the query plan.



Chapter 2:-

Discretionary Access Control

A DAC mechanism allows users to grant or revoke access to any of the objects under their control. Discretionary access control is an identity-based access control model that provides users with a certain amount of control over their data. Data owners (document creators or any users authorized to control data) can define access permissions for specific users or groups of users. In other words, whom to give access to and what privileges to grant are decided at the resource owner’s discretion.

Gaining access to a file in the DAC model works like this: User 1 creates a file and becomes its owner, or obtains access rights to an existing file. User 2 requests access to this file. They provide their credentials: username, password, or something else. User 1 grants access at their discretion. However, user 1 can’t grant access rights that exceed their own. For example, if user 1 can only read a document, they can’t allow user 2 to edit it. If there’s no contradiction between the ACL created by an administrator and the decision made by user 1, access is granted.  
  
Pros

User-friendly — Users can manage their data and quickly access data of other users. Flexible — Users can configure data access parameters without administrators.

Cons Low level of data protection — DAC can’t ensure reliable security because users can share their data however they like.  
Overlay of user privileges— A permissions conflict may happen with users of multiple nested workgroups.  
  
Mandatory access control

Mandatory Access Control (MAC) is a security model that enforces access controls based on predefined rules and policies set by a system administrator or security administrator. It's a way to ensure that access to resources (such as files, databases, or network resources) is granted or denied based on the security labels associated with both the resource and the user or process trying to access it. This contrasts with discretionary access control (DAC), where resource owners have more control over access permissions.

The Bell-LaPadula model is a formal security model used in the context of Mandatory Access Control (MAC) to enforce confidentiality. It was developed by David Bell and Leonard LaPadula in the early 1970s and is widely recognized in the field of computer security. The model's primary objective is to prevent unauthorized users from accessing classified information.

The Bell-LaPadula model introduces two main security levels:

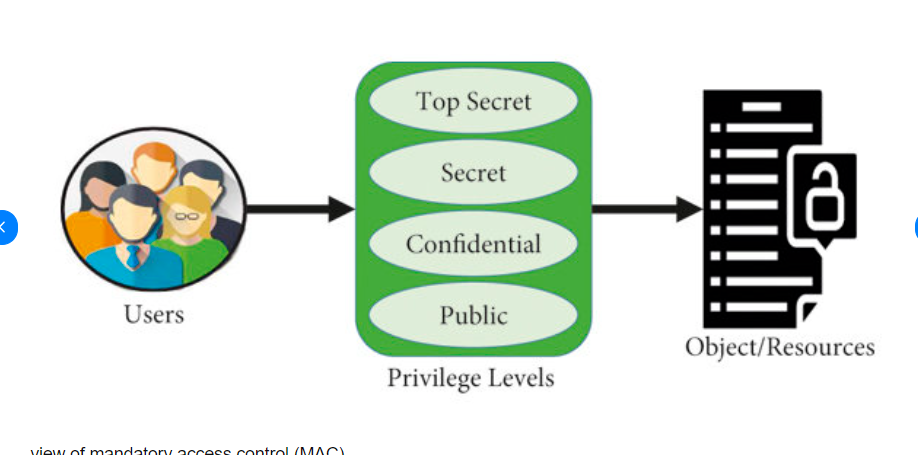
* **Top Secret (TS)**: The highest level of security clearance.
* **Secret (S)**: A lower level of security clearance.

The model also defines two access modes:

* **Read (r)**: The ability to read (access) information.
* **Write (w)**: The ability to modify or create information.

Based on these principles, the Bell-LaPadula model enforces rules such as:

1. A subject with a security clearance at level "S" can read information labeled "S" or lower (i.e., "S" and "Unclassified").
2. A subject with a security clearance at level "TS" can read information labeled "TS," "S," or lower.
3. A subject with a security clearance at level "S" cannot write to information labeled "TS."
4. A subject with a security clearance at level "TS" can write to information labeled "TS," "S," or lower.



Role Based Access Control

Role-Based Access Control (RBAC) can be implemented in a database management system (DBMS) to control and manage access to data stored in the database. RBAC in DBMS works similarly to the general concept of RBAC but is tailored to the specifics of database access and operations. Here's how RBAC can be applied to a DBMS:

1. **Roles in DBMS:** In a DBMS, roles represent a set of privileges or permissions associated with specific user groups. These roles define what actions users with those roles can perform on the database objects (e.g., tables, views, procedures). Common roles might include "Read-Only," "Data Entry," "Administrator," etc.
2. **Privileges and Permissions:** Privileges are specific rights or actions that users can perform on database objects. These can include SELECT, INSERT, UPDATE, DELETE, EXECUTE, and more. Permissions determine what operations a role is allowed to perform on specific database objects.
3. **Users and Role Assignment:** Users are the individuals who interact with the database. Instead of assigning permissions directly to users, you assign roles to users. Users can then inherit the privileges associated with those roles. This separation simplifies user management and access control.

**Database Objects:** Each database object (e.g., table, view, stored procedure) can be associated with specific permissions that dictate what roles are allowed to perform actions on them. For instance, you might grant a "Read-Only" role SELECT permissions on certain tables.

**GRANT and REVOKE Statements:** DBMS systems provide SQL statements such as GRANT and REVOKE to assign and revoke privileges to/from roles. These statements allow administrators to define and modify the access control rules.

A temporal database is a database that needs some aspect of time for the organization of information. In the temporal database, each tuple in relation is associated with time. It stores information about the states of the real world and time. The temporal database does store information about past states it only stores information about current states. Whenever the state of the database changes, the information in the database gets updated. In many fields, it is very necessary to store information about past states. For example, a stock database must store information about past stock prizes for analysis. Historical information can be stored manually in the schema.

here are various terminologies in the temporal database:

* **Valid Time:**The valid time is a time in which the facts are true with respect to the real world.
* **Transaction Time:**The transaction time of the database is the time at which the fact is currently present in the database.
* **Decision Time:**Decision time in the temporal database is the time at which the decision is made about the fact.

**Applications of Temporal Databases**

**Finance: It is used to maintain the stock price histories.**

**It can be used in Factory Monitoring System for storing information about current and past readings of sensors in the factory.**

**Healthcare: The histories of the patient need to be maintained for giving the right treatment.**

**Banking: For maintaining the credit histories of the user.**

A Mobile Database is a type of database that can be accessed by a mobile network and connected to a mobile computing device (or wireless network). Here, there is a wireless connection between the client and the server. In the modern world, **Mobile Cloud Computing** is expanding quickly and has enormous potential for the database industry. It will work with a variety of various devices, including Mobile Databases powered by iOS and Android, among others. Couchbase Lite, Object Box, and other popular databases are examples of databases.

### **Mobile Database Environment has the Following Components:**

* For storing the corporate and providing the corporate applications, a Corporate Database Server and DBMS is used.
* For storing the mobile data and providing the mobile application, a Remote Database and server are used.
* There is always a two-way communication link present between the Mobile DBMS and Corporate DBMS.

### **Features of Mobile Database:**

* There are a lot of features of Mobile Database which are discussed below:
* As more people utilize laptops, smartphones, and PDAs to live on the go.
* To prevent frequent transactions from being missed due to connection failure, a cache is kept.
* Mobile Databases and the main database server are physically independent.

A general-purpose database (often a relational database) which has been improved to contain spatial information that represents objects specified in a geometric space as well as tools for searching and analyzing such data, is known as a **Spatial Database**. The depiction of basic geometric objects like points, lines, and polygons is supported by the majority of Spatial Databases. Some Spatial Databases can handle more complicated structures, including triangulated irregular networks, topological coverages, and 3D objects.

Your spatial dataset can be accessed, stored, and managed with the help of Spatial Databases.

### **There are Several Types of Operations like:**

* **Measurement:**  
  Computes geometry distance, polygon area, line length, etc.
* **Geoprocessing:**  
  Create new features by changing existing ones, for as by surrounding them with a buffer or by intersecting features.
* **GeometryConstructors:**  
  Specifies the vertices (points or nodes) that define the form to create new geometries.
* **ObserverFunctions:**  
  Queries that give detailed answers on a feature, like the location of a circle's center.
* **Predicates:**  
  True/false questions about the spatial relationships between geometries are permissible.

**Example**

**A road map is a visualization of geographic information. A road map is a 2-dimensional object which contains points, lines, and polygons that can represent cities, roads, and political boundaries such as states or provinces.**

**Chapter 3:-**

