UNIT VI – QUERY PROCESSING & SECURITY

Notes By: Raju Poudel (Mechi Multiple Campus)

Overview of Query Processing

- ✓ Query Processing is a translation of high-level queries into low-level expression.
- ✓ It is a step wise process that can be used at the physical level of the file system, query optimization and actual execution of the query to get the result.
- ✓ It requires the basic concepts of relational algebra and file structure.
- ✓ It refers to the range of activities that are involved in extracting data from the database.
- ✓ It includes translation of queries in high-level database languages into expressions that can be implemented at the physical level of the file system.
- ✓ In query processing, we will actually understand how these queries are processed and how they are optimized.

Overview of Query Processing

In the above diagram,

- The first step is to transform the query into a standard form.
- A query is translated into SQL and into a relational algebraic expression. During this process, Parser checks the syntax and verifies the relations and the attributes which are used in the query.
- The second step is Query Optimizer. In this, it transforms the query into equivalent expressions that are more efficient to execute.
- The third step is Query evaluation. It executes the above query execution plan and returns the result.

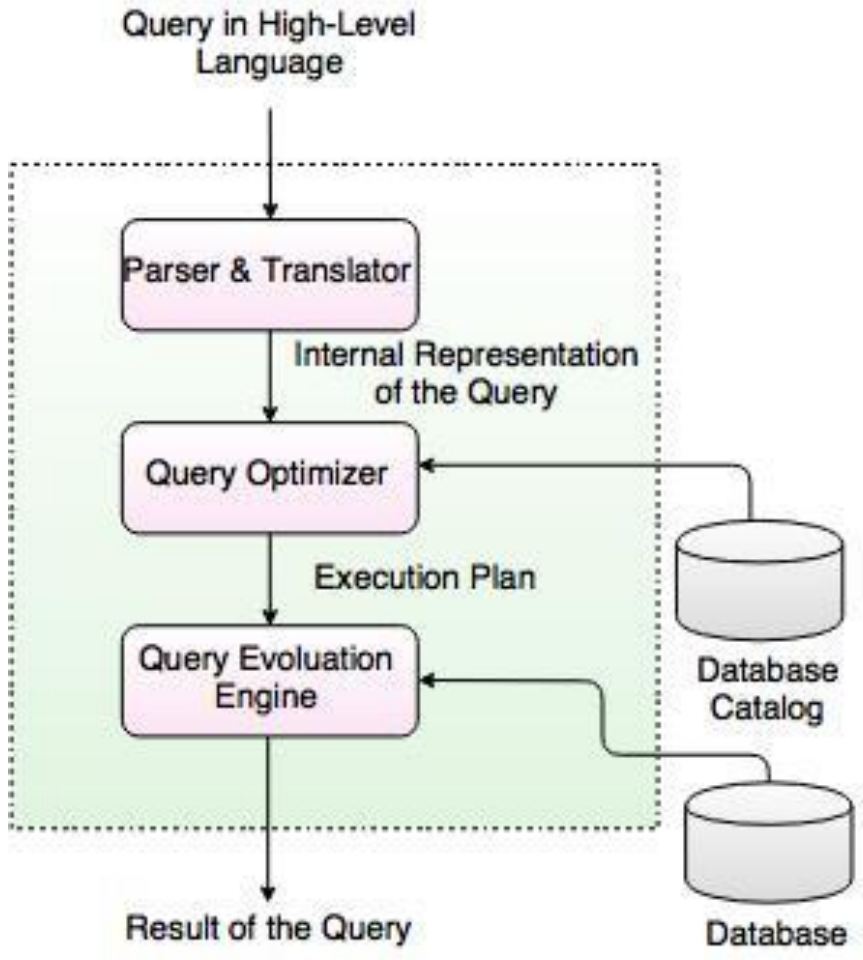


Fig. Query Processing

Measuring of Query Cost

- ✓ Cost is generally measured as total elapsed time for answering query.
- ✓ Many factors contribute to time cost - disk accesses, CPU, or even network communication.
- ✓ Typically disk access is the predominant cost, and is also relatively easy to estimate.
- ✓ Measured by taking into account –
  - Number of seeks
  - Number of blocks read
  - Number of blocks written

- ✓ Cost to write a block is greater than cost to read a block – data is read back after being written to ensure that the write was successful.

Measuring of Query Cost

- ✓ For simplicity we just use number of block transfers from disk as the cost measure.
- ✓ We ignore the difference in cost between sequential and random I/O for simplicity.
- ✓ We also ignore CPU costs for simplicity. Costs depends on the size of the buffer in main memory. Having more memory reduces need for disk access.
- ✓ Amount of real memory available to buffer depends on other concurrent OS processes, and hard to determine ahead of actual execution.

- ✓ We often use worst case estimates, assuming only the minimum amount of memory needed for the operation is available.
- ✓ Real systems take CPU cost into account, differentiate between sequential and random I/O, and take buffer size into account. We do not include cost to writing output to disk in our cost formulae.

Selection Operation

- ✓ **File scan –** search algorithms that locate and retrieve records that fulfill a selection condition.

- ✓ **Algorithm A1 (linear search).** Scan each file block and test all records to see whether they satisfy the selection condition.
  - – Cost estimate (number of disk blocks scanned) = br
    - - br denotes number of blocks containing records from relation r
  - – If selection is on a key attribute, cost = (br/ 2) (stop on finding record)

- ✓ Linear search can be applied regardless of
  - ∗ selection condition, or
  - ∗ ordering of records in the file, or
  - ∗ availability of indices

Selection Operation

- **A2** (*binary search*). Applicable if selection is an equality comparison on the attribute on which file is ordered.
  - Assume that the blocks of a relation are stored contiguously
  - Cost estimate (number of disk blocks to be scanned):

$$= \lceil \log_2(b_r) \rceil + \left\lceil \frac{SC(A, r)}{f_r} \right\rceil - 1$$

    * $\lceil \log_2(b_r) \rceil$ — cost of locating the first tuple by a binary search on the blocks
    * $SC(A, r)$ — number of records that will satisfy the selection
    * $\lceil SC(A, r)/f_r \rceil$ — number of blocks that these records will occupy
  - Equality condition on a key attribute: $SC(A, r) = 1$; estimate reduces to $\lceil \log_2(b_r) \rceil$

Sorting Operation

- ✓ We may build an index on the relation, and then use the index to read the relation in sorted order. May lead to one disk block access for each tuple.

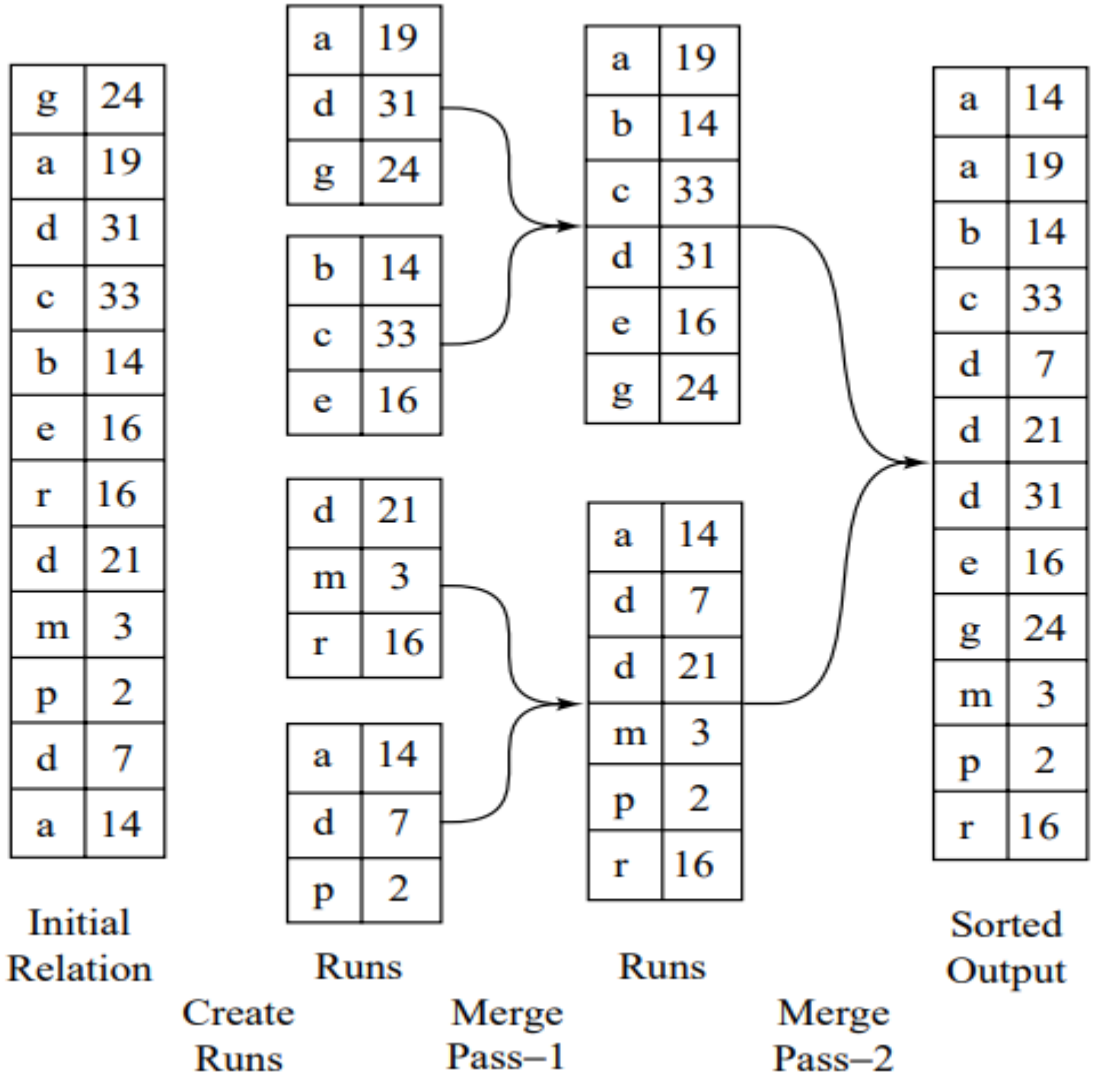
- ✓ For relations that **fit in memory**, techniques like **quicksort** can be used. For relations that **don’t fit in memory**, **external sort-merge** is a good choice.

External Sort–Merge

Let M denote memory size (in pages).

1. Create sorted **runs** as follows. Let i be 0 initially. Repeatedly do the following till the end of the relation:
   (a) Read M blocks of relation into memory
   (b) Sort the in-memory blocks
   (c) Write sorted data to run R_i; increment i.
2. Merge the runs; suppose for now that $i < M$. In a single merge step, use i blocks of memory to buffer input runs, and 1 block to buffer output. Repeatedly do the following until all input buffer pages are empty:
   (a) Select the first record in sort order from each of the buffers
   (b) Write the record to the output
   (c) Delete the record from the buffer page; if the buffer page is empty, read the next block (if any) of the run into the buffer.

External Sort–Merge Example



Join Operation

- ✓ Several different algorithms to implement joins
  - – Nested-loop join
  - – Block nested-loop join
  - – Indexed nested-loop join
  - – Merge-join
  - – Hash-join

- ✓ Choice based on cost estimate
- ✓ Join size estimates required, particularly for cost estimates for outer-level operations in a relational-algebra expression.

Query Optimization

- ✓ A single query can be executed through different algorithms or re-written in different forms and structures.
- ✓ Hence, the question of **query optimization** comes into the picture – Which of these forms or pathways is the most optimal? The query optimizer attempts to determine the most efficient way to execute a given query by considering the possible query plans.
- ✓ A **query optimizer** is a critical database management system (DBMS) component that analyses Structured Query Language (SQL) queries and determines efficient execution mechanisms.
- ✓ A query optimizer generates one or more query plans for each query, each of which may be a mechanism used to run a query. The most efficient query plan is selected and used to run the query.
- ✓ Database users do not typically interact with a query optimizer, which works in the background.

Importance of Query Optimization

- ✓ The goal of query optimization is to reduce the system resources required to fulfill a query, and ultimately provide the user with the correct result set faster.
  - First, it provides the user with faster results, which makes the application seem faster to the user.
  - Secondly, it allows the system to service more queries in the same amount of time, because each request takes less time than unoptimized queries.
  - Thirdly, query optimization ultimately reduces the amount of wear on the hardware (e.g. disk drives), and allows the server to run more efficiently (e.g. lower power consumption, less memory usage).

Query Optimization Process

There are broadly two ways a query can be optimized:

- ✓ **Analyze and transform equivalent relational expressions**: Try to minimize the tuple and column counts of the intermediate and final query processes.

- ✓ **Using different algorithms for each operation**: These underlying algorithms determine how tuples are accessed from the data structures they are stored in, indexing, hashing, data retrieval and hence influence the number of disk and block accesses.

Database Administrator (DBA)

- ✓ A **database administrator (DBA)** is a specialized computer systems administrator who maintains a successful database environment by directing or performing all related activities to keep the data secure.

- ✓ The top **responsibility** of a **DBA professional** is to maintain **data integrity**. This means the DBA will ensure that data is secure from unauthorized access but is available to users.

- ✓ A database administrator will often have a working knowledge and experience with a wide range of database management products such as Oracle-based software, and SQL, in addition to having obtained a degree in Computer Science and practical field experience and additional, related IT certifications.

Types of DBA

✓ There are different types of DBAs depending on the an organization's requirements:

1. **Administrative DBA –** maintains the servers and databases and keeps them running. Concerned with backups, security, patches, replication. These are activities mostly geared towards maintaining the database and software platform, but not really involved in enhancing or developing it.
2. **Development DBA -** works on building SQL queries, stored procedures, and so on, that meet business needs. This is the equivalent of a programmer, but specializing in database development. Commonly combined the role of Administrative DBA.
3. **Data Architect –** designs schemas, builds tables indexes, data structures and relationships. This role works to build a structure that meets a general business needs in a particular area.
4. **Data Warehouse DBA -** this is a relatively newer role, responsible for merging data from multiple sources into a data warehouse. May have to design the data warehouse as well as cleaning up and standardizing the data before loading using specialist data loading and transformation tools.

Roles & Responsibilities of DBA

1. Database installation, upgrade and patching
2. Install and configure relevant network components
3. Ensure database access, consistency and integrity
4. Resolving issues related to performance bottlenecks
5. Provide reporting on various metrics including availability, usage and performance
6. Performance testing and benchmark activities
7. Work with development staff on architectures, coding standards, and quality assurance policies
8. Create models for new database development or changes to existing ones
9. Respond to and resolve database access and performance issues
10. Monitor database system details
11. Design and implement redundant systems, policies, and procedures for disaster recovery

Roles & Responsibilities of DBA

12. Monitor, optimize and allocate physical data storage for database systems
13. Plan and coordinate data migrations
14. Develop, implement, and maintain change control and testing processes
15. Perform database transaction and security audits
16. Establish end-user database access control levels
17. Implement database encryption and data encryption
18. Plan and ensure compliance with established best practices, related policies and legislation
19. Participate as a member of a team to move the team toward the completion of its goals
20. Capacity planning, installation, configuration, database design, migration, performance monitoring, security, troubleshooting, as well as backup and data recovery.

Database Security

- ✓ Database security refers to the collective measures used to protect and secure a database or database management software from illegitimate use and malicious threats and attacks.

- ✓ Database security covers and enforces security on all aspects and components of databases. This includes:
  - Data stored in database
  - Database server
  - Database management system (DBMS)
  - Other database workflow applications

- ✓ Database security is generally planned, implemented and maintained by a database administrator and or other information security professional.

Database Security

- ✓ Database security refers to the collective measures used to protect and secure a database or database management software from illegitimate use and malicious threats and attacks.

- ✓ Database security covers and enforces security on all aspects and components of databases. This includes:
  - Data stored in database
  - Database server
  - Database management system (DBMS)
  - Other database workflow applications

- ✓ Database security is generally planned, implemented and maintained by a database administrator and or other information security professional.

Database Security Issues

1) No Security Testing Before Deployment
2) Poor Encryption and Data Breaches Come Together
3) Stolen Database Backups
4) Flaws in Features
5) Weak and Complex DB Infrastructure
6) Limitless Administration Access = Poor Data Protection
7) Inadequate Key Management
8) Irregularities in Databases

Types of Database Security

The database security can be broadly classified into physical and logic security.

- ✓ **Physical security -** security of hardware associated with the system and the protection of the site where the computer resides.

- ✓ **Logical security -** security measures residing in the operating system or the DBMS designed to handle threats to data.

Following measures can be used to secure database:

- Access authorization.
- Access controls.
- Backup and recovery of data.
- Data integrity.
- Encryption of data.

Access Protection/Control

- ✓ Access control is a security technique that regulates who or what can view or use resources in a computing environment. It is a fundamental concept in security that minimizes risk to the business or organization.
- ✓ There are two types of access control: physical and logical.
- ✓ **Physical access control** limits access to campuses, buildings, rooms and physical IT assets.
- ✓ **Logical access control** limits connections to computer networks, system files and data.
- ✓ To secure a facility, organizations use electronic access control systems that rely on user credentials, access card readers, auditing and reports to track employee access to restricted business locations and proprietary areas, such as data centres.
- ✓ Access control systems perform identification authentication and authorization of users and entities by evaluating required login credentials that can include passwords, personal identification numbers (PINs), biometric scans, security tokens or other authentication factors. Multifactor authentication, which requires two or more authentication factors, is often an important part of layered defence to protect access control systems.

User Account and Database Audits

- ✓ Database auditing involves observing a database so as to be aware of the actions of database users.
- ✓ Database administrators and consultants often set up auditing for security purposes, for example, to ensure that those without the permission to access information do not access it.

- ✓ Audit is an analysis of an organization's Computer and information systems in order to evaluate the efficiency, correctness & integrity of its database systems as well as to uncover potential Security Cracks.
- ✓ Auditing is done to verify that DBMS operations are properly implemented and executed.

Mandatory access control (MAC)

- ✓ Mandatory Access Control (MAC) is a set of security policies constrained according to system classification, configuration and authentication.
- ✓ MAC policy management and settings are established in one secure network and limited to system administrators.
- ✓ MAC defines and ensures a centralized enforcement of confidential security policy parameters.

Advantages

- MAC provides tighter security because only a system administrator may access or alter controls.
- MAC policies reduce security errors.

Discretionary access control (DAC)

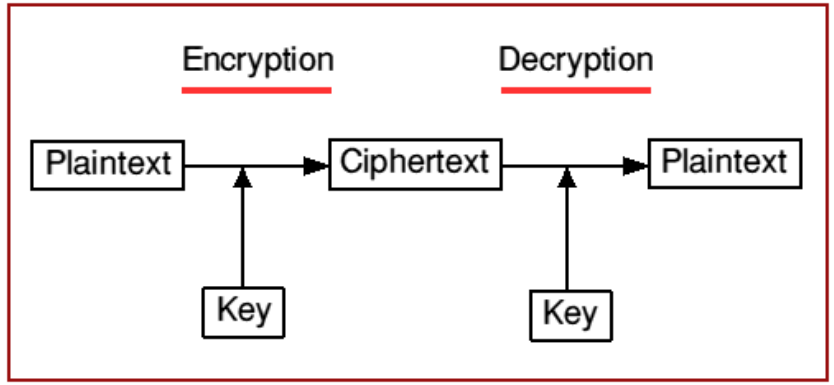
- ✓ Discretionary access control (DAC) is a type of security access control that grants or restricts object access via an access policy determined by an object's owner group and/or subjects.
- ✓ DAC mechanism controls are defined by user identification with supplied credentials during authentication, such as username and password.
- ✓ DACs are discretionary because the subject (owner) can transfer authenticated objects or information access to other users. In other words, the owner determines object access privileges.

Advantages

- ACL maintenance or capability
- Grant and revoke permissions maintenance

Data Encryption and Decryption

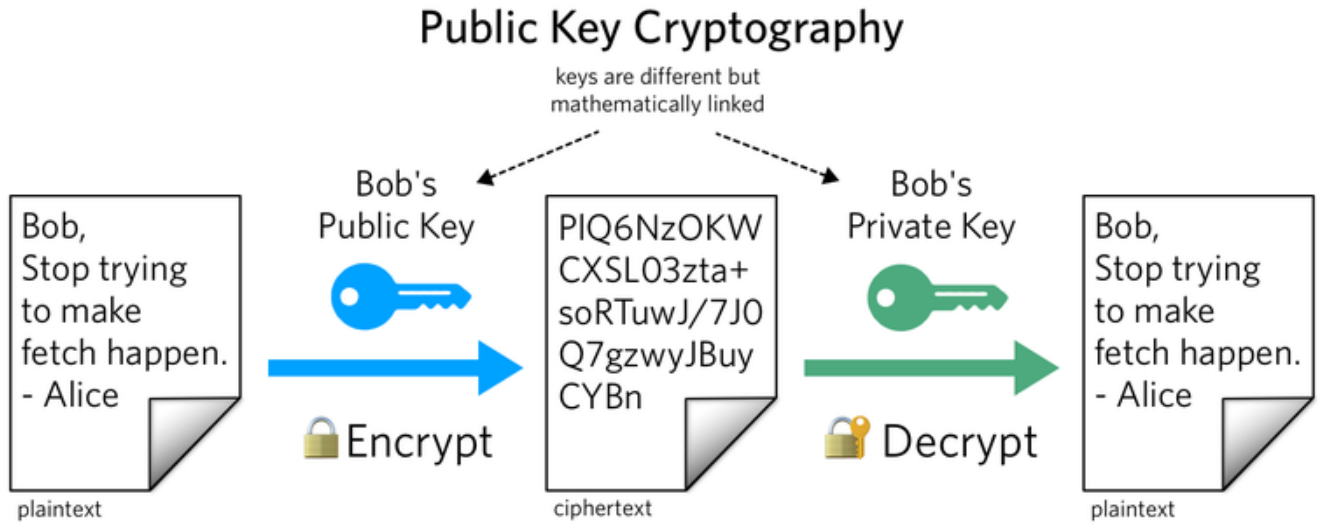
- ✓ **Database encryption** is the process of converting data, within a database, in plain text format into a meaningless cipher text by means of a suitable algorithm.
- ✓ **Database decryption** is converting the meaningless cipher text into the original information using keys generated by the encryption algorithms.



- ✓ Numerous algorithms are used for encryption. These algorithms generate keys related to the encrypted data.
- ✓ These keys set a link between the encryption and decryption procedures. The encrypted data can be decrypted only by using these keys.

Public Key Cryptography

- ✓ Public key cryptography (PKC) is an encryption technique that uses a paired public and private key (or asymmetric key) algorithm for secure data communication.
- ✓ A message sender uses a recipient's public key to encrypt a message.
- ✓ To decrypt the sender's message, only the recipient's private key may be used.



Secret Key Cryptography

- ✓ Here only one key is used for both encryption and decryption. This type of encryption is also referred to as symmetric encryption.