Unit -8

Query Processing and Security

Query Processing

- Query Processing is the activity performed in extracting data from the database.
- In query processing, it takes various steps for fetching the data from the database.
- The steps involved are:
  - Parsing and translation
  - Optimization
  - Evaluation

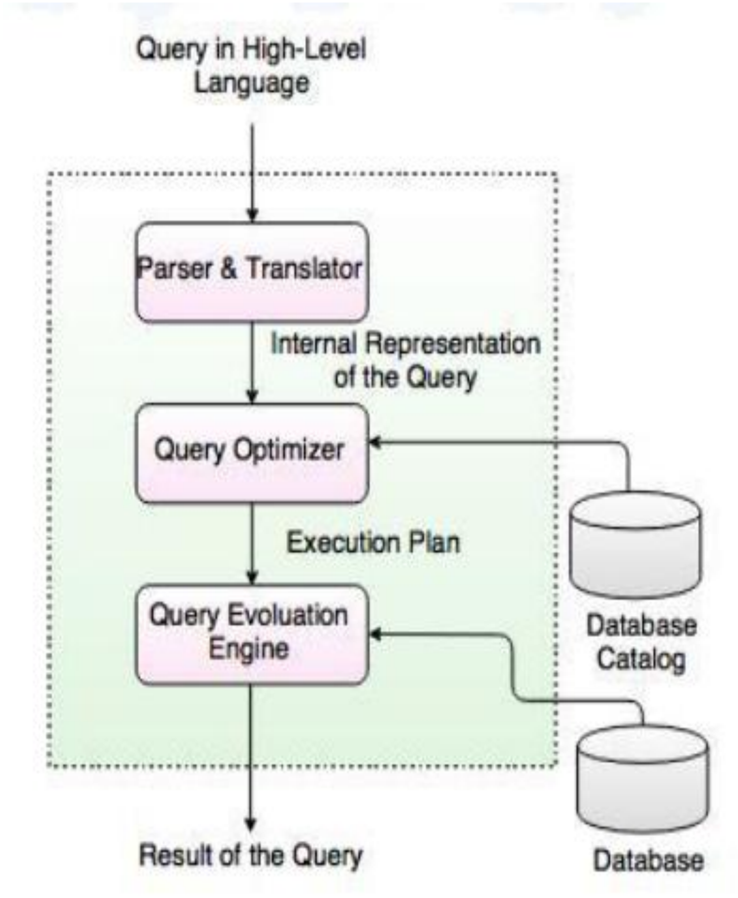


Fig. Query Processing

Overview of Query Processing

Parsing and Translation

- During query processing , Parser checks the syntax and verifies the relations and the attributes which are used in the query.
- Then the translator translates it into the form of relational algebra. During translation, a single query can be translated into two or more relational algebra expressions.
- For eg: we have a query

  select salary from Employee where salary>1000;

  It can be translated into the following relational algebra expressions:

  - $\boldsymbol{\sigma_{salary>1000} (\pi_{salary} (Employee))}$
  - $\boldsymbol{\pi_{salary} (\sigma_{salary>1000} (Employee))}$

Query Optimization

- A **query optimizer** is a critical database management system component that analyses Structured Query Language queries and determines efficient execution mechanisms.
- It selects the best query from the two or more relational algebra expressions created by the translator and that best query will be placed on the execution plan.

Evaluation

- The third step is Query evaluation. The query evaluation engine take the execution plan, executes it and returns the result.

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

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 (component that analyses Structured Query Language (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 un-optimized 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.

Techniques of Query Optimization

- Do not use Select *. Instead specify the column names.
- Use indexing.
- Remove the use of select distinct if not necessary.
- Limit the number of rows fetched using 'LIMIT'
- Try to use inner join instead of outer join.

- Note: In oracle LIMIT is implemented as:
- **for oracle 12 and later versions**
- select * from employees fetch first 10 rows only;

- **for older oracle versions**
- select * from employees where rownum<=10;

- **to get the last 10 rows**
- select * from(select * from employees order by employee_id desc)where rownum<=10;

Evaluation of Query Expression or Relational Algebra Expression

- In query processing system, two methods are used for evaluating an expression carrying multiple operations:
  - Materialization (**Materialized evaluation)**
  - Pipelining (**Pipelined evaluation**)

Materialized Evaluation

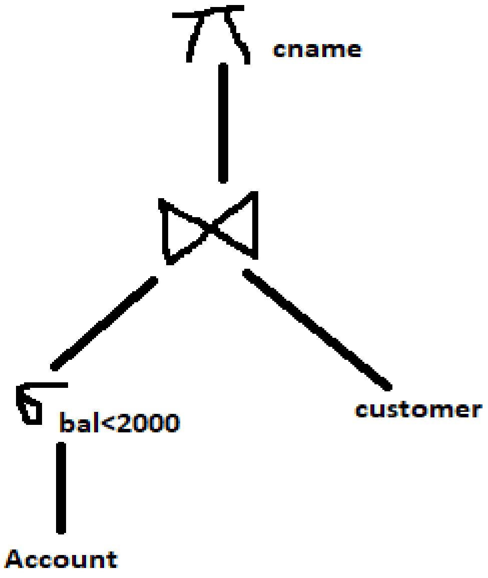
- It is the process of storing the output of an operation in a temporary relation for processing by the next operation.
- It starts from the lowest level operation in the expression which are at the bottom of query tree.

Consider the below relational algebra query as

$\Pi_{cname}\ (\sigma_{bal<2000}(account) \bowtie customer))$

The pictorial representation of the above query as shown below.

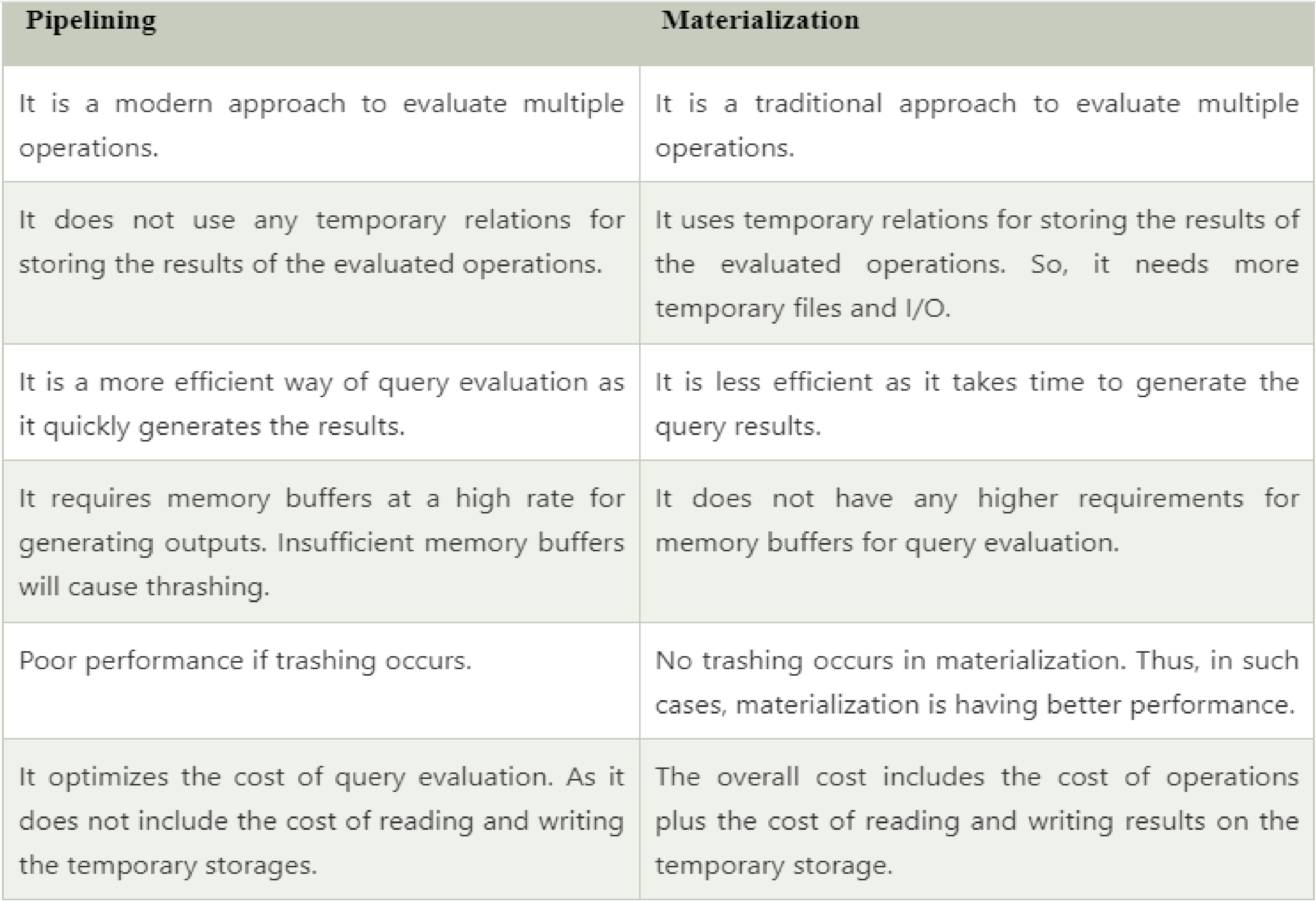
Main problem with this approach is the need to construct temporary relation which must be written to disk.



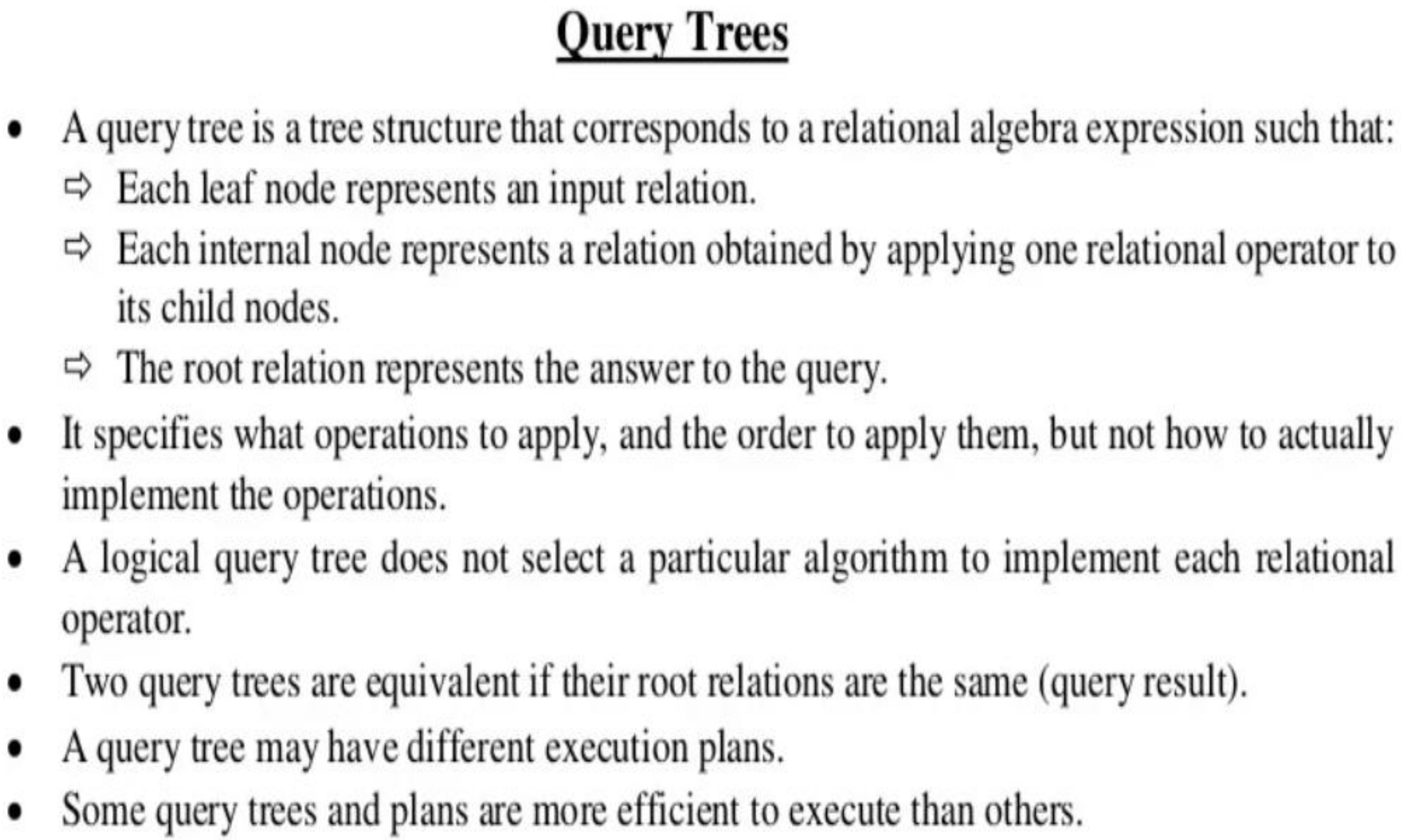
Pipelined Evaluation

- Pipelining is an alternate method or approach to the materialization method.
- This method evaluates several operations simultaneously. Result of one operation is passed to the next operation.
- This method evaluates the expression in a bottom-up manner and doesn't store intermediate results to temporary files.
- It is used to improve the performance of queries by eliminating the cost of reading and writing to temporary relations.
- A buffer is created for each pair of adjacent operation to hold tuples being passed from 1st operation to second one.

Pipelining Vs Materialization



Pipelining	Materialization
It is a modern approach to evaluate multiple operations.	It is a traditional approach to evaluate multiple operations.
It does not use any temporary relations for storing the results of the evaluated operations.	It uses temporary relations for storing the results of the evaluated operations. So, it needs more temporary files and I/O.
It is a more efficient way of query evaluation as it quickly generates the results.	It is less efficient as it takes time to generate the query results.
It requires memory buffers at a high rate for generating outputs. Insufficient memory buffers will cause thrashing.	It does not have any higher requirements for memory buffers for query evaluation.
Poor performance if trashing occurs.	No trashing occurs in materialization. Thus, in such cases, materialization is having better performance.
It optimizes the cost of query evaluation. As it does not include the cost of reading and writing the temporary storages.	The overall cost includes the cost of operations plus the cost of reading and writing results on the temporary storage.



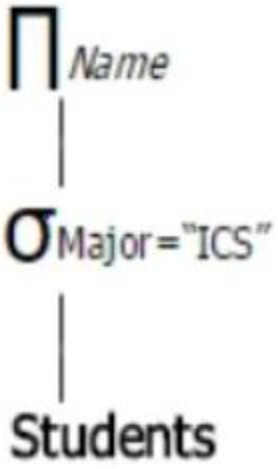
Query Trees

- A query tree is a tree structure that corresponds to a relational algebra expression such that:
  - ⇨ Each leaf node represents an input relation.
  - ⇨ Each internal node represents a relation obtained by applying one relational operator to its child nodes.
  - ⇨ The root relation represents the answer to the query.
- It specifies what operations to apply, and the order to apply them, but not how to actually implement the operations.
- A logical query tree does not select a particular algorithm to implement each relational operator.
- Two query trees are equivalent if their root relations are the same (query result).
- A query tree may have different execution plans.
- Some query trees and plans are more efficient to execute than others.

Constructing a query tree

- Example 1:

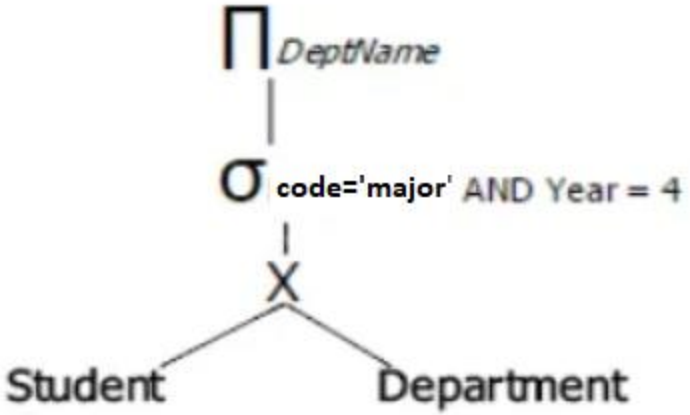
  select name from students where major='ICS';



Constructing a query tree

- Example 2

  select deptname from student,department where code= 'major' and year=4;



DBA (Database Administrator)

- The most technical class of database users.
- Have the knowledge of how to design and manage the database.
- Responsible for proper working of database and RDBMS, has the responsibility of making database backup and recovery plan.

Roles and Responsibilities of DBA

- **Installing and upgrading the DBMS Servers: –** DBA is responsible for installing a new DBMS server for the new projects. He is also responsible for upgrading these servers as new versions comes in the market or requirement. If there is any failure in upgradation of the existing servers, he should be able revert the new changes back to the older version, thus maintaining the DBMS working.
- **Design and implementation: –** Designing the database and implementing is also DBA's responsibility. He should be able to decide proper memory management, file organizations, error handling, log maintenance etc for the database.
- **Performance tuning: –** Since database is huge and it will have lots of tables, data, constraints and indices, there will be variations in the performance from time to time. Also, because of some designing issues or data growth, the database will not work as expected. It is responsibility of the DBA to tune the database performance. He is responsible to make sure all the queries and programs works in fraction of seconds

Roles and Responsibilities of DBA

- **Migrate database servers: –** Sometimes, users using oracle would like to shift to SQL server It is the responsibility of DBA to make sure that migration happens without any failure, and there is no data loss.

- **Backup and Recovery: –** Proper backup and recovery programs needs to be developed by DBA. Data should be backed up regularly so that if there is any crash, it should be recovered without much effort and data loss.

- **Security: –** DBA is responsible for creating various database users and roles, and giving them different levels of access rights.

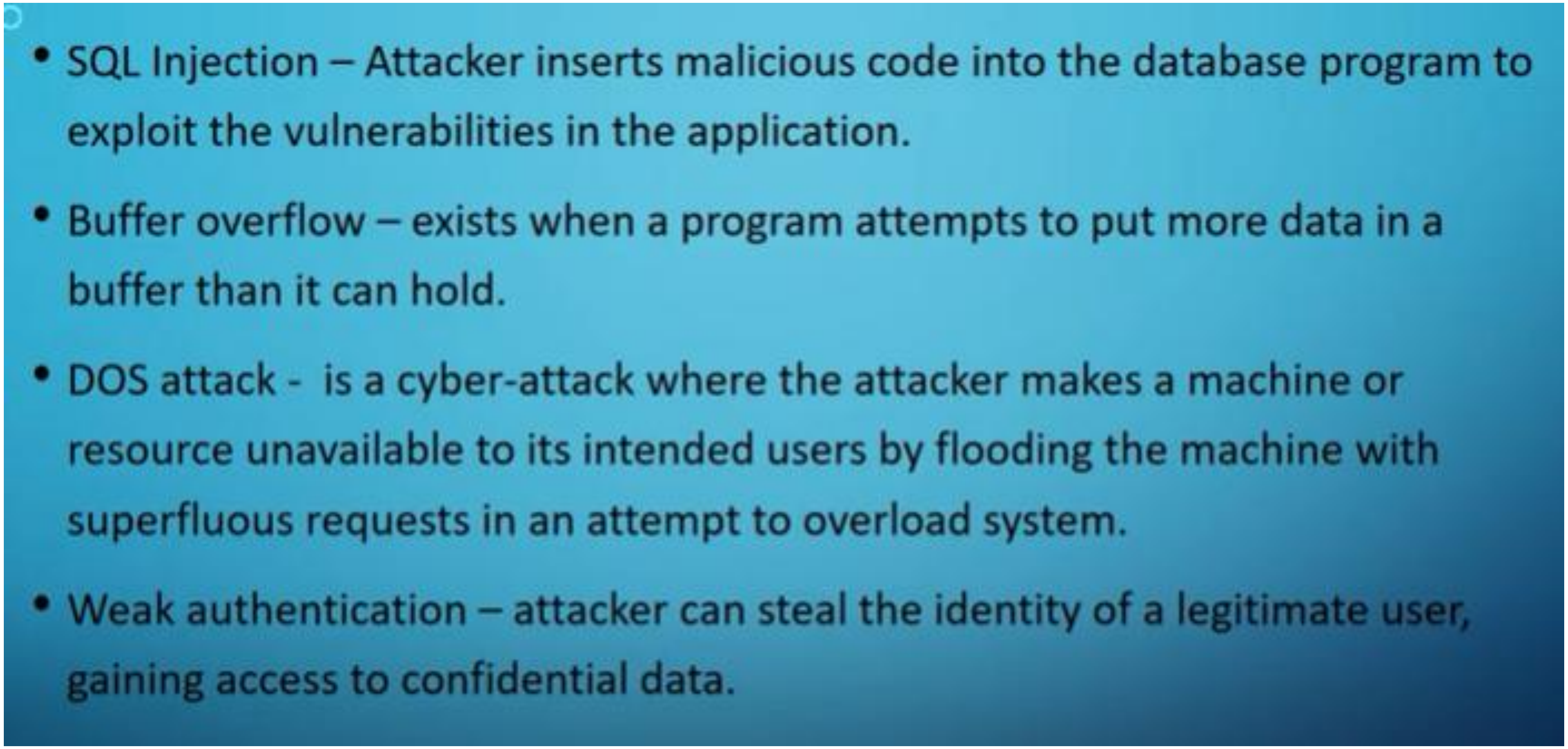
- **Documentation: –** DBA should be properly documenting all his activities so that if he quits or any new DBA comes in, he should be able to understand the database without any effort.

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 protects the confidentiality, integrity and availability (CIA) of an organization's database.
- Database security is generally planned, implemented and maintained by a database administrator.

Security Threats(Most Common Attacks)

- SQL Injection
- Buffer Overflow Vulnerabilities
- DOS (Denial Of Service) Attack
- Weak Authentication



- SQL Injection – Attacker inserts malicious code into the database program to exploit the vulnerabilities in the application.
- Buffer overflow – exists when a program attempts to put more data in a buffer than it can hold.
- DOS attack - is a cyber-attack where the attacker makes a machine or resource unavailable to its intended users by flooding the machine with superfluous requests in an attempt to overload system.
- Weak authentication – attacker can steal the identity of a legitimate user, gaining access to confidential data.

Methods To secure Database

- Access Control
- Auditing
- Authentication
- Encryption
- Backup

Methods To secure Database

- **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.
  - **1. Discretionary Access Control (DAC):** The owner of data determines who can access specific resource.
  - **2. Mandatory Access Control (MAC):** Access rights are regulated by a central authority based on multiple levels of security.
  - **3. Role Based Access Control (RBAC):** Access based on job title (role). For example, there should not be permission for HR (Human Resource) specialist to create network account.

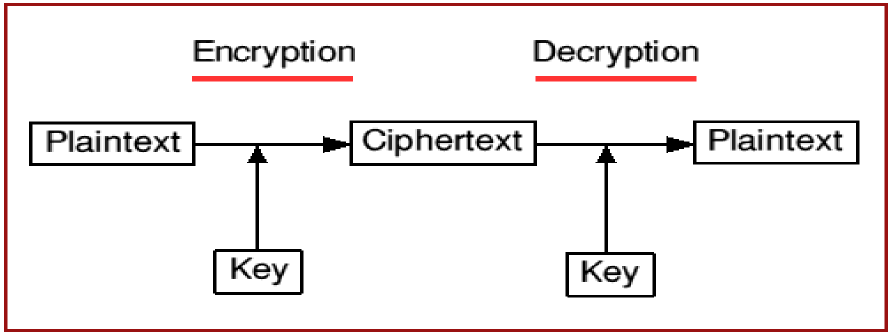
- **Auditing:** Database auditing involves observing a database so as to be aware of the actions of database users.
- Database administrators and consultants often set up auditing for security purposes, for example, to ensure that those without the permission to access information do not access it
- Auditing is done to verify that DBMS operations are properly implemented and executed

Methods To secure Database

- **Authentication** means verifying the identity of someone who wants to access data, resources or applications. Validating that identity establishes a trust relationship for further interactions.
- **Encryption:** It is the process of encoding a message or information in such a way that only authorized parties can access it.
- **Backup:** Database backup is the process of backing up the operational state, architecture and stored data of database software. It enables the creation of a duplicate instance or copy of a database in case the primary database crashes, is corrupted or lost.

Data Encryption and Decryption

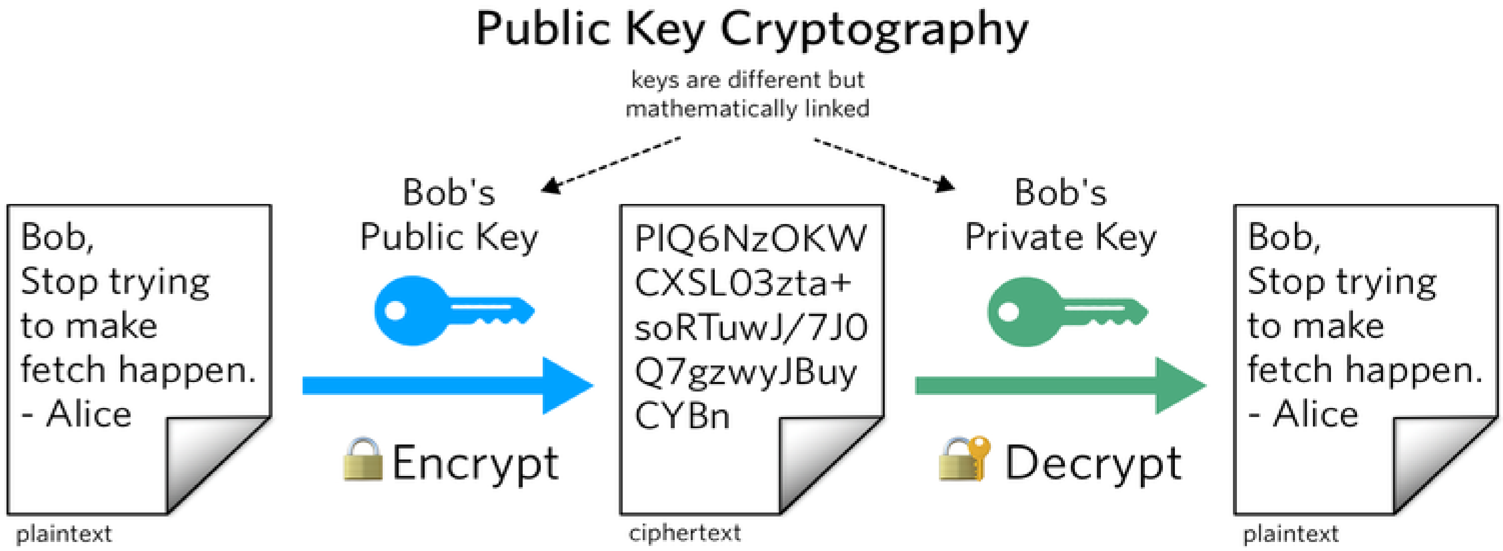
- **Database encryption** is the process of converting data, within a database, in plain text format into a meaningless cipher text by means of a suitable algorithm
- **Database decryption** is converting the meaningless cipher text into the original information using keys generated by the encryption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

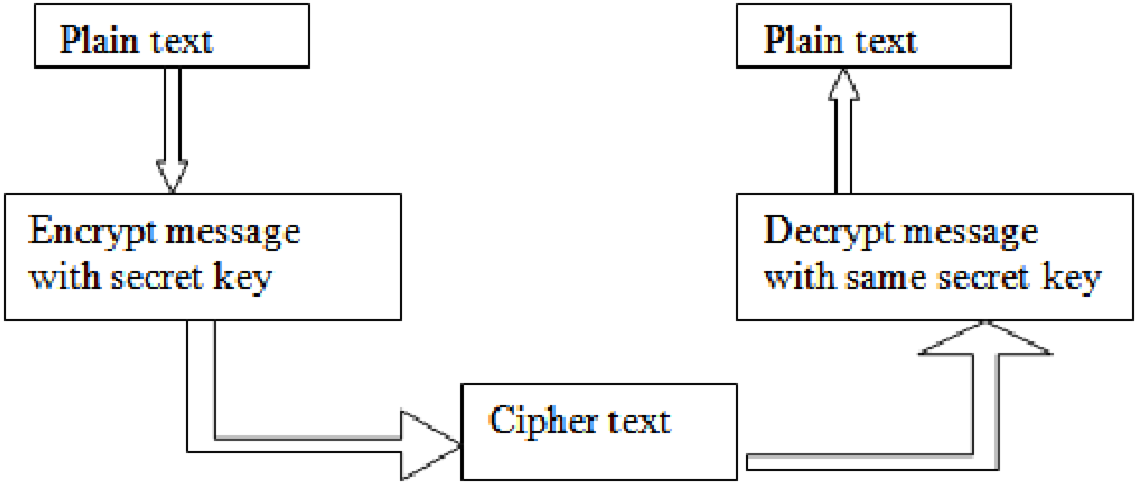
Public Key Cryptography (Asymmetric Key Cryptography)

- Public key cryptography 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.


Public Key Cryptography
keys are different but mathematically linked
Bob's Public Key
Bob's Private Key
Bob,
Stop trying to make fetch happen.
- Alice
plaintext
Encrypt
PIQ6NzOKW CXSL03zta+ soRTuwJ/7J0 Q7gzwyJBuy CYBn
ciphertext
Decrypt
Bob,
Stop trying to make fetch happen.
- Alice
plaintext

Secret Key Cryptography (Symmetric Key Cryptography)

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


Plain text
Encrypt message with secret key
Cipher text
Decrypt message with same secret key
Plain text