|  |  |
| --- | --- |
| A picture of a winding road and trees  Datebase programming  11/5/2024 | Abstract  Tech Solutions suffers from the ability to identify the projects that each employee in the company has done, which leads to difficulty in tracking and managing projects, hinders productivity, and hinders the contributions of each employee, and this is what the current system cannot do.  Hasan alhwietat |

Contents

[Part1: 2](#_Toc167867714)

[1. Provide a well-defined problem statement with a clear set of user and system requirements. Then design a relational database system. 2](#_Toc167867715)

[1.1. Problem statement: 2](#_Toc167867716)

[1.2. User Requirements: 2](#_Toc167867717)

[1.3. System Requirement: 3](#_Toc167867718)

[1.4. Design Relation database: 4](#_Toc167867719)

[2. Describe in detail the architecture of the DBMS and the function of the sub-components along with their programming techniques. The targeted programming techniques correspond to what is programmed in this component such as different types of data structures. You need to mention which components you have decided to use in your DBMS implementation. (Report) 5](#_Toc167867720)

[3. Evaluate the ability to transfer data between different DBMS components. In addition, a data flow diagram to show data movement through the system components and flowcharts to describe how the system works. (Report) 8](#_Toc167867721)

[4. Evaluate the effectiveness of the database and DBMS design in terms of user and system requirements. (Report) 8](#_Toc167867722)

[Part2: 9](#_Toc167867723)

[1. Write the SQL queries based on the system requirement and implement the corresponding CRUD statements in the language of development. Each SQL statement needs to be rewritten in C language and later incorporated into the project. Note: the CRUD statements should act like the original SQL statement. For example, the insert statement will not add the new record to a table if the primary key is duplicated. (Report and Project) 9](#_Toc167867724)

[2. Criticize the process of mapping SQL queries to the developed CRUD statements and suggest improvements to optimize the developed CRUD queries. (Report) 10](#_Toc167867725)

[Part3: 10](#_Toc167867726)

[1. Describe in detail the standard DBMS concurrency control techniques. (Report) 10](#_Toc167867727)

[Part4: 10](#_Toc167867728)

[1. Suggest and apply a testing plan to ensure compatibility and functionality of developed DBMS components. (Report and Project) 10](#_Toc167867729)

[2. Evaluate the DBMS in terms of Improvements needed to ensure the continued effectiveness of any database (Report) 10](#_Toc167867730)

Scenario: -

At the present time, most old companies that have systems and huge data stored in them that they cannot usually access are suffering, and they need data management to simplify that huge data. Therefore, Tech Solutions is a leading technology company in the field of information technology, especially in developing programs and databases, facing great challenges in database development. And its maintenance due to the presence of old database systems that hinder productivity, difficulty in analyzing data, cause major security problems, and complete difficulty in continuing the company’s successes by taking advantage of all fields such as computing and artificial intelligence, and hinder the provision of innovative services to customers, which hinders competitiveness, so it needs to develop the database and rely on The principle of a DBMS that meets most of the needs of companies and to preserve user data with the possibility of development, maintainability, expansion, and obtaining better performance and high security due to data privacy regulations and due to the most complex cybersecurity threats.

# Part1:

## Problem statement:

Tech Solutions suffers from the ability to identify the projects that each employee in the company has done, which leads to difficulty in tracking and managing projects, hinders productivity, and hinders the contributions of each employee, and this is what the current system cannot do.

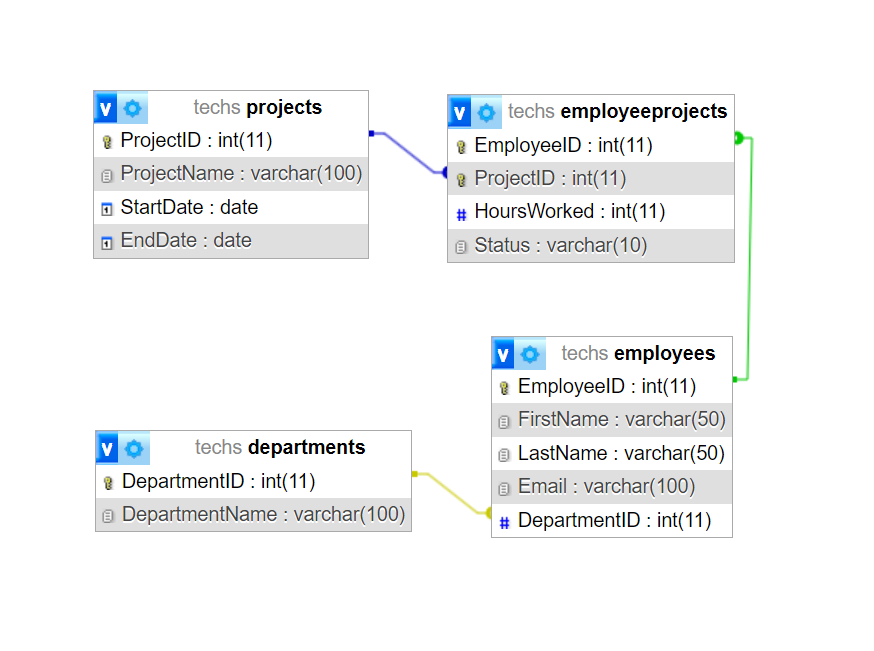
### User Requirements:

* Employee: The system allows employees to fill out their own data in addition to knowing the projects they participate in by entering the employee ID so that each employee can know the projects he worked on and adding your own information such as your identification number, first name, last name, email, and departmentID.
* Managers: The system allows the project administrator to know the projects in the system, the employees working on them, the types of departments the number of working hours, the start and end dates and the status of each employee in the project.

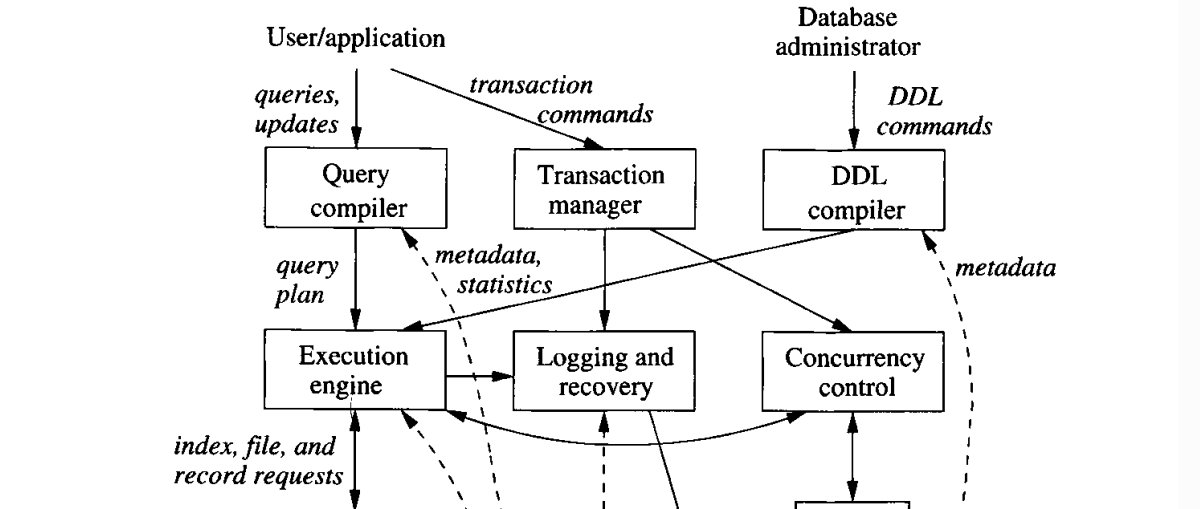
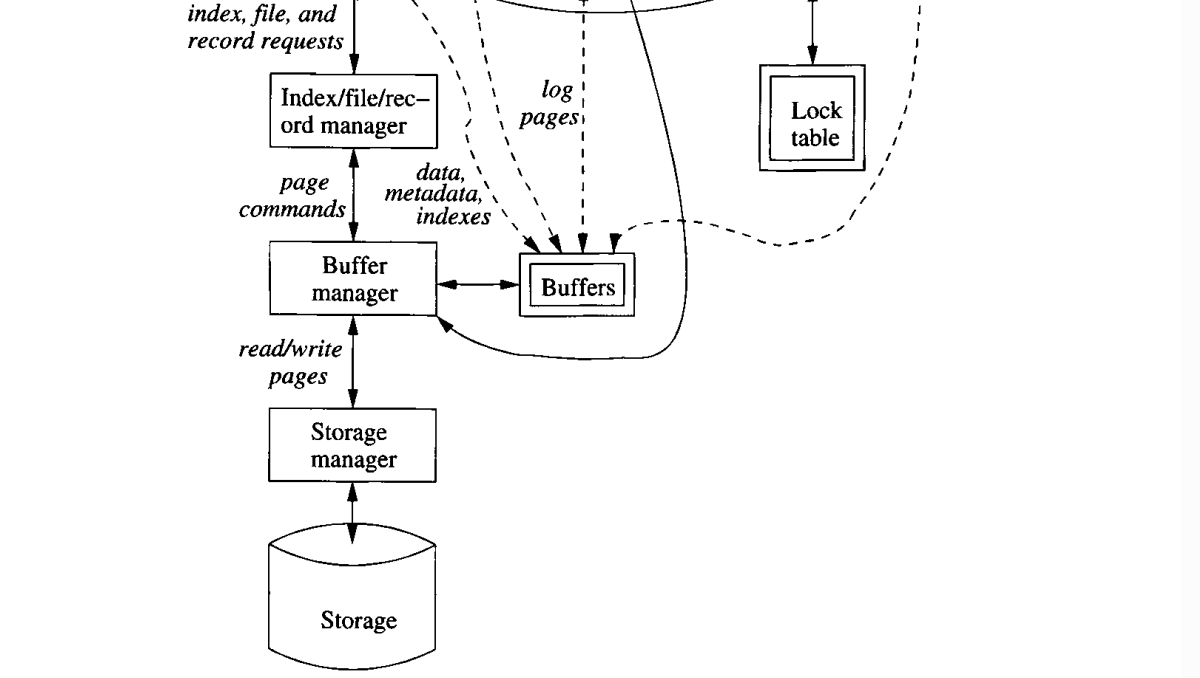
### System Requirement:

1. Dynamic memory allocation: To ensure that the system is working to improve performance by allocating storage memory and adjusting buffer sizes to manage it efficiently.
2. Threading: Implementing it within the system allows simultaneous execution of database operations, ensuring data consistency and integrity, reducing the burden and accelerating operations.
3. Indexing: It improves database queries so that it helps identify the appropriate columns for quick access.
4. Interface: Ensure that there is an interface that interacts with the database system for ease of use and user experience.
5. Transaction manager: Its implementation within the system ensures the ACID properties (Atomicity, Consistency, Isolation, Durability) of transactions.
6. Logging and Recovery: create a file for each table in the project.

### Design Relation database:



## Describe in detail the architecture of the DBMS.



First, above, there are two distinct sources of DBMS commands:

1. Conventional users and application programs that request or modify data.

2. Database Administrator: The person or persons responsible for the structure or schema of the database.

1. Query Compiler: It is a component that translates high-level SQL queries into low-level instructions for execution. It takes queries from the user and sends them to the execution engine.

Technique:

* + 1. Query parser: It builds a brown structure in the form of a tree of textual implementation.
    2. Query preprocessor: It conducts tests on all the relationships mentioned in the query to make sure of their existence already.
    3. Query optimizer: It converts the initial media plan into operations chains on actual data. The best chains and faster to access data are chosen.

1. Transaction Manager: It is a component that accepts transactions orders from the application and is informed when completing the transactions and providing information about the application expectations that implement three Concurrency control, logging and deadlock resolution. Ensures the properties of ACID (Atomicity, Consistency, Isolation, Durability). Interacts with the logging and recovery.

Techniques: ACID (Atomicity, Consistency, Isolation, Durability)

* Atomicity: For each execution process, all or none of its actions are executed. No partial actions will be performed on the data.
* Consistency: Every transaction execution returns the database to the correct state. For example, a change in one table needs changes in the corresponding tables as well otherwise the data will be inconsistent.
* Isolation: Each execution of a transaction is independent of any other executions of concurrent transactions. In other words, controlling access to data from multiple users/transactions simultaneously, without allowing unexpected interactions between users/transactions.
* Durability: If a transaction is successfully executed, its traces are permanently recorded in the database meaning that the work of a completed transaction is never lost.

1. DDL Compiler: It is a component responsible for translating and processing Data Definition Language (DDL) statements that is used to define and manage database structures or schema objects.

Techniques: CREATE, ALTER, DROP

1. Execution Engine: It is a component responsible for executing the query plans created by the query compiler. It interacts with most of the system components through buffers or directly. He requests data from the temporary store manager to process it. It needs to interact with the scheduled to avoid access to closed data. He interacts with the record manager to ensure that all the database changes are recorded correctly. It takes queries from the query compiler and sends data requests to the Index/File/Record Manager.

Techniques: SQL query: CRUD operations.

1. Loging and Recovery: It is a component that records each change in the database separately on disk. It allows the log manager to track the change to find out problems that may occur to the system and allows the manager to retrieve from the change log and restore data stability in the event of a system failure. Its main purpose is to recover data in the event of a failure, so that it receives log updates from the transaction manager and sends log pages to Buffers.

Techniques: File (Backup: Linked List, Tree Serialization, Tree Deserialization).

1. Concurrency Control: It is a component that manages simultaneous access to shared data. It ensures that transactions are executed in isolation, which allows some systems to execute many transactions at once. It must ensure that concurrent transactions are executed correctly to ensure data consistency. It receives DDL commands from the DBA and interacts with the Lock Table.

Techniques: Threading.

Lock table: It is a sub component that prevents or allows access to a part of the data, so it stores information about the locks. These locks are granted by the Concurrency Control so that you can enter a situation that no one can track. The transaction manager bears responsibility for entering and canceling transactions.

Techniques: Lock Modes (shared, exclusive).

1. Index/File/Record Manager: It is a component that manages data storage so that it organizes and coordinates quick access to data by organizing it as files or records for faster access. It receives data requests from the execution engine and sends page commands to the Buffer Manager.

Techniques: Indexing: Single-Level (Primary, Clustering, Secondary), Multilevel: (B-Trees, B+ Trees), Hashing.

1. Buffer Manager: The person responsible for managing Buffer fetches data from secondary storage, the place where data is permanently stored, into Buffer. He communicates with the storage manager to obtain data from the disk. He divides the main memory into Buffer so that the blocks passed from the storage manager are transferred to them. It manages the temporary storage of data pages in memory for quick access. When it receives commands from the index/file/record manager, it sends the read/write pages to the storage manager.

Techniques: Static Memory Allocation, Dynamic Memory Allocation (malloc, realloc, calloc)

Buffers: It is the primary component because all components of the database management system that need information from the disk will interact with Buffers either directly or through the execution engine.

Techniques: Buffer.

1. Storge Manager: It is the component that manages data storage. It controls the placement of data on the disk and its movement between the disk and main memory so that it receives all data requests and takes data from the disk by tracking files and returning them as blocks and passes them to the temporary stores manager and includes operating system commands.

Techniques: Pointer (Liked List), Binary Search Trees (BST), Heaps.

Storage: It is the place where database data is stored, including tables, indexes, and records. It is called the disk, and the database management system is controlled directly on the disk. Receives read/write operations from the storage manager.

Techniques: Store data on tablets and file format. HDD, SSD

* The kinds of information that various components may need include:

1. Data: the contents of the database itself.
2. Metadata: Database scheme that describes the database structure and restrictions on it.
3. Log Records: Information about modern changes in the database. This supports the durability of the database.
4. Statistics: information gathered and stored by the DBMS about data properties such as the sizes of, and values in, various relations or other components of the database.
5. Indexes: data structures that support efficient access to the data.

## Evaluate the ability to transfer data between different DBMS components. In addition, a data flow diagram to show data movement through the system components and flowcharts to describe how the system works. (Report)

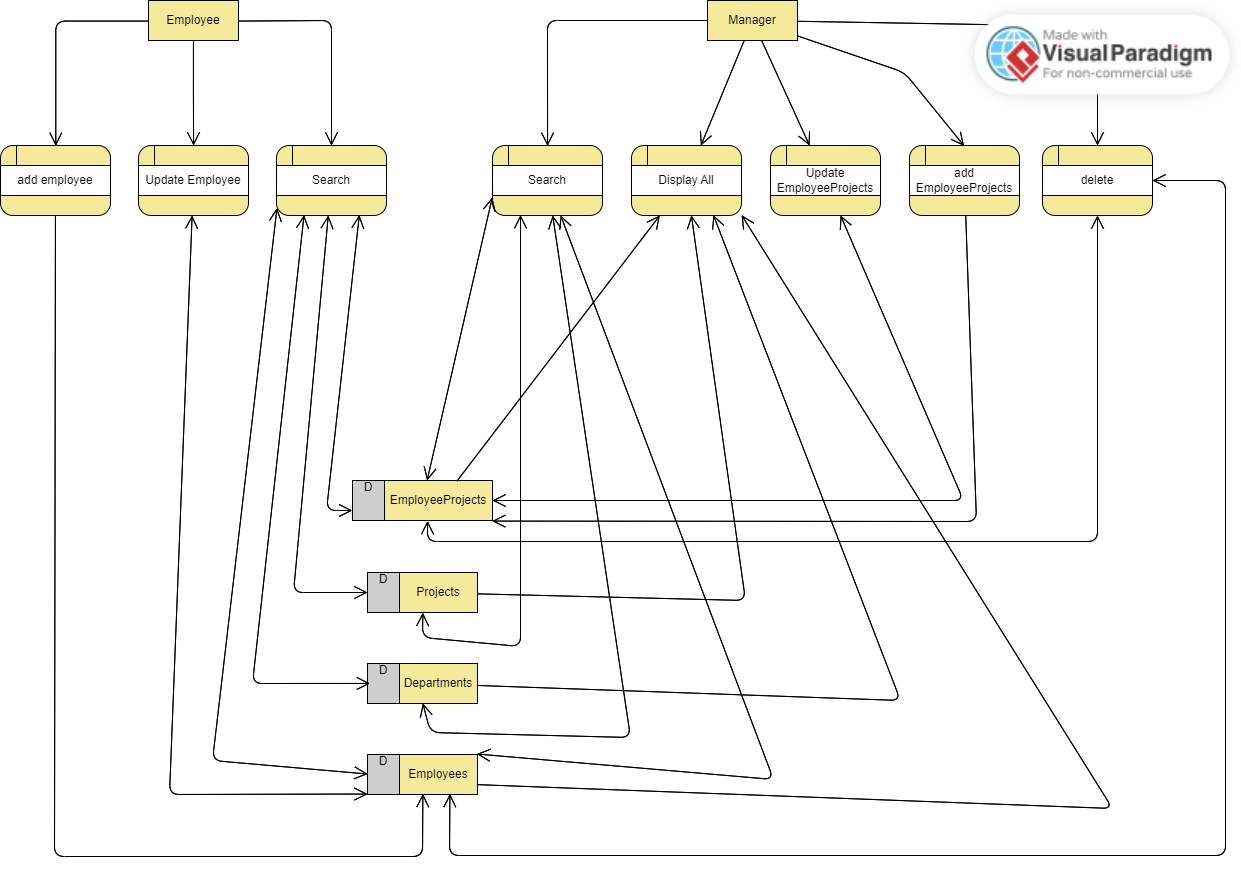
1. Query Compiler: It translates, improves, and assembles high-level queries received from the user into low-level instructions, so that it takes the user’s requests, processes them, and sends them to the execution engine, or DML, which processes the low-level instructions so that they can be executed. It also uses metadata and statistics to determine the sequence of operations to be faster. The query is translated into an internal model called the query plan. The Query Compiler communicates with the execution engine and affects the overall performance of the system and contributes to reducing execution time through internal models that can affect the efficiency of SQL execution to ensure data consistency and integrity.
2. Transaction Manager: Transaction processing queries and DML procedures are grouped into transactions, which are units that must be executed automatically and separated from each other. It is responsible and controls the simultaneous access of transaction commands and is responsible for accepting them from the application. It is notified when transactions are completed and provides information about the application’s expectations that are being executed. Implement three concurrency controls, logging and deadlock resolution. Ensures the properties of ACID (Atomicity, Consistency, Isolation, Durability). Interacts with the logging and recovery, which ensures that the database remains in a consistent state before and after the transaction is completed. Operations are executed in a scheduled manner and provides high durability so that no completed transaction is ever lost. It informs the manager when the transaction started and when it ended, as well as expectations.
3. DDL Compiler: He is responsible for translating and processing the DDL statements that are used to manage database structures. These statements are entered by the database administrator. When they are entered into the DDL, they are analyzed for commands that could change the schema, and then they are passed to the execution engine and then passed to Index, File and Log Manager To change metadata this means changing the database schema. It optimizes storage structures, efficiently manages metadata, and ensures schema integrity. It directly affects the speed and efficiency of accessing and transferring data within the database system.
4. Execution Engine: It is responsible for executing each step of the chosen query plan. It interacts with most components of the DBMS directly or through buffers. It must get data from the database into buffers to be processed. It interacts with the scheduler to avoid accessing locked data and with the log manager to record all changes to the database correctly, which means that it executes the query plan and returns the answers by interpreting the SQL and accessing and returning the data. Ensures efficient data retrieval and reduces unnecessary data transfer.
5. Loging and Recovery: In order to ensure the integrity and durability of the system, all changes that occur to the database are recorded separately from the disk to avoid any crash or failure problems and to keep the data in a consistent state, so that it deals with the buffer to write to them for recording and then deals with the buffer manager by negotiating with it to write to the store. Timer on disk to obtain safety data to be able to restore stability in the event of a system crash. Receives challenge commands from the log manager and sends log pages to the buffers. Reduces downtime and maintains data integrity during failures.
6. Concurrency Control: Responsible for ensuring that transactions are executed separately, but there are many transactions that are executed at the same time, so in the scheduler, he ensures the individual actions of multiple transactions and executes them in order so that they are executed at once. The scheduler also maintains the lock and prevents the access of any transaction that is not allowed to access. The locks are stored in the main memory lock table greatly affects the execution of queries and database operations by preventing the execution engine from accessing the locked parts. It ensures that transactions are executed to ensure data consistency and integrity. It always seeks to increase productivity and system efficiency when many transactions are executed simultaneously and executed in a manner Synchronous: No delay occurs due to a specific operation, but the rest of the operations are executed separately from each other due to synchronization. Improving data transfer efficiency using micro-locks ensures smooth and synchronous access to data, reducing waiting times.

Lock table: It prevents or allows access to a part of the data, so it stores information about the locks. These locks are granted by the Concurrency Control so that you can enter a situation that no one can track. The transaction manager bears responsibility for entering and cancelling transactions.

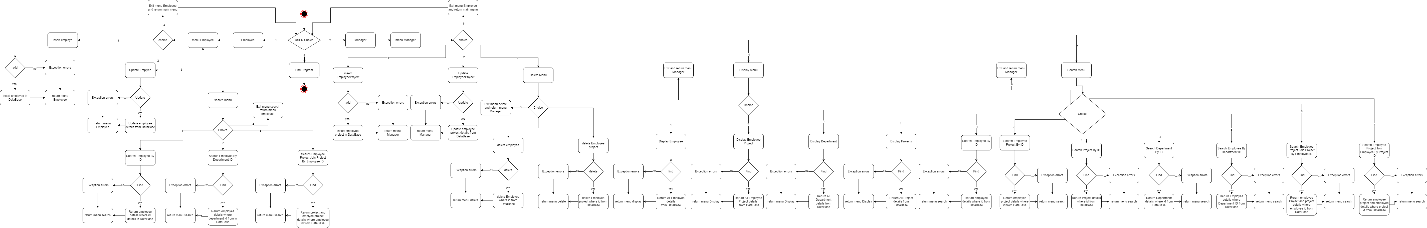
1. Index/File/Record Manager: It is a component that manages data storage so that it organizes and coordinates quick access to data by organizing it as files or records for faster access. It receives data requests from the execution engine and sends page commands to the Buffer Manager, and returns the data to the execution engine. It improves the efficiency of data transfer by retrieving data quickly, by organizing the data on the disk as files, records, or an index that is easily accessible, which speeds up searches for specific data.
2. Buffer Manager: It is responsible for managing the Buffer and dividing the main memory into Buffer, which are small areas to which disk blocks are transferred. Components that need data will interact with the Buffer Manager, and Buffer directly or through the execution engine includes many types of information that the components need, such as data, metadata, log records, statistics, and indexes. The main goal of Buffer is to reduce the number of accesses to the hard disk and put them in the Buffer. Therefore, the Buffer manager is in contact with the storage manager to obtain data and store it in the Buffer. If the index, file, and record administrator requests, the data is returned from Buffer, which facilitates access to the data and makes it easier to access. Faster, while ensuring data integrity and non-loss, and increases system efficiency. Improve data transfer efficiency through buffer management and reduced disk I/O operations
3. Storge Manager: The storage manager is responsible for placing data on the disk and its movement between the disk and main memory. It is responsible for the Buffer Manager's requests. When it requests blocks of data stored on the disk, it tracks the location of the files and returns the data to the Buffer Manager. For high efficiency, the DBMS's controls storage on the disk. In some circumstances it seeks to maintain data consistency and integrity by applying restrictions and rules to preserve the data.

Storage: It is the place where database data is stored, including tables, indexes, and records. It is called the disk, and the database management system is controlled directly on the disk. Receives read/write operations from the storage manager.

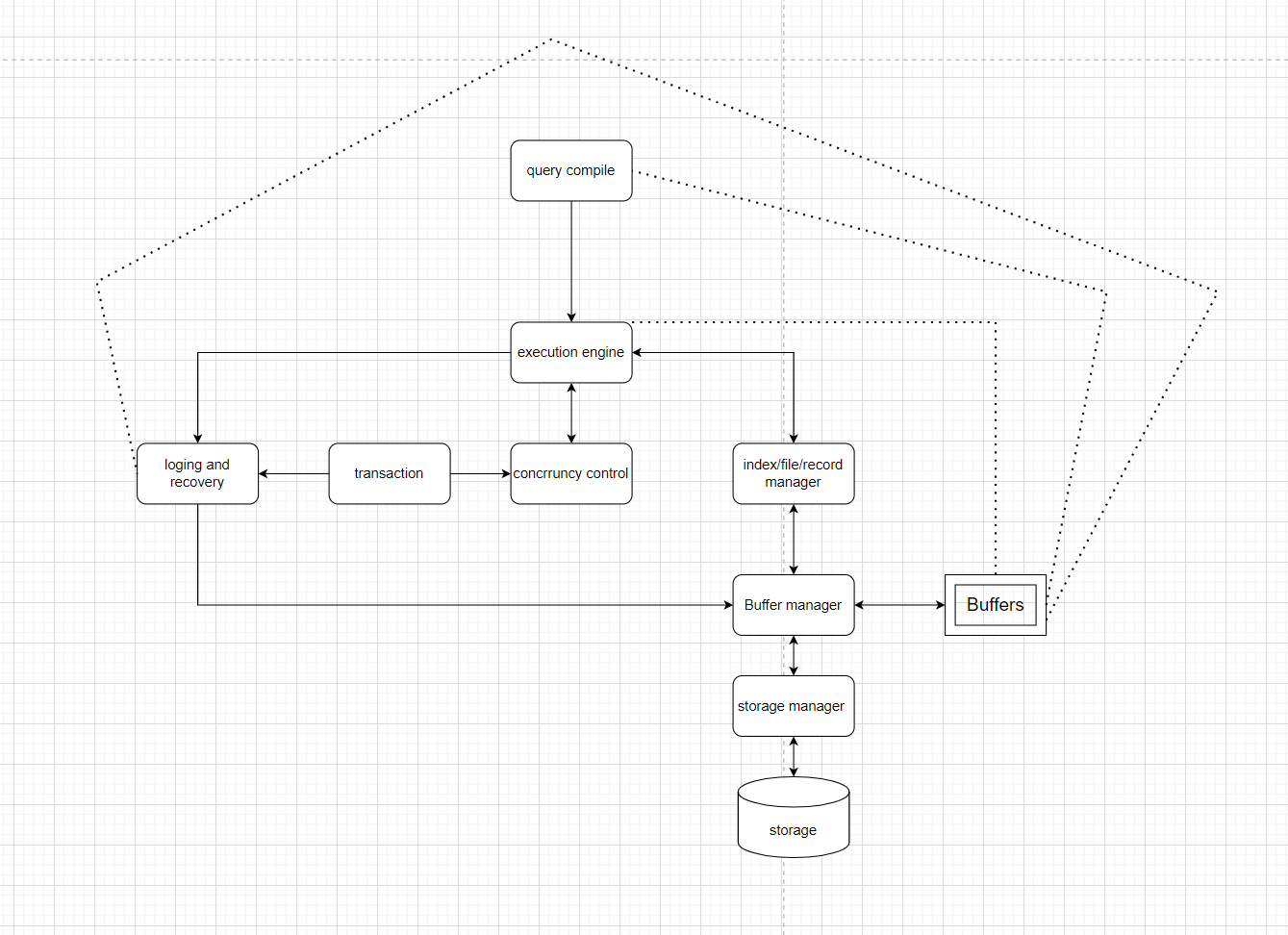
(geeksforgeek, 21 Apr, 2023)



This image shows the data flow diagram of the system by knowing how the users in the system work and the processes that each user performs. For example, the employee is able to add his own data if he has not been registered before, and he can also update his information in addition to searching for the project in which he works and the department. His personal information and his private information about him only. As for the manager user, he is responsible for creating projects and distributing employees to the projects, in addition to knowing comprehensive details about the employees in the company and how many hours they work on each project, in addition to deleting projects that are not useful or have no relation to the company and deleting any employee in addition to conduct a database search to find out more details about employees and projects.



This image fully shows how the system works.



This Flow Chart image of my system shows how the components interact with each other. For example, the query compiler writes and processes the SQL statements, and then the execution engine component executes the sentences and deals with the synchronous controller and recording to save data from loss by recording all changes through the transaction. The execution engages in requesting data from the file, records, and index manager for quick access to the data, and then sending it to the buffer manager, who sends a request to the storage manager for the data he wants, then places it as blocks in the buffer, and then the data is returned to the file, records, and index manager, and then returned to the execution engine.

## Evaluate the effectiveness of the database and DBMS design in terms of user and system requirements. (Report)

User Requirements:

* Employee: By designing the system’s DBMS, it allowed employees to know the number of working hours on each project they worked on. For example, designing the database helped the relationship between the various tables in the system so that the system could achieve the employee’s requirements and what data it needed, such as knowing their personal data and updating it in addition. To know the details of the projects they are working on and which department each employee belongs to. As for the design of the data flow diagram, it shows the operations that the employee performs in addition to sending the data to the Data Store. For example, when an employee is created, the Data Store will confirm that the employee does not exist and who the data is sent to. Data store, as well as operations such as searching and updating, will communicate with the data store to update and return the data. As for the flow chart design, which shows how the components work with each other when creating a new employee, in addition to returning data from the buffer.
* Managers: By designing the system’s DBMS, it allowed managers to know the number of working hours for each employee in each project he worked on, as well as the creation of projects, etc. For example, the design of the database helped the relationship between the various tables in the system so that the system could achieve the managers’ requirements and what data it needed. Such as knowing the details of existing projects in addition to knowing the details of the departments and knowing each department’s number of employees and the number of employees working on a project, etc. As for designing a data flow diagram, it shows the operations carried out by the manager in addition to sending the data to the data store. For example, when a project employee is created, it will be ensured that Data store: The project exists in addition to the employee existing, and then the data is sent to the data store, and operations such as searching, updating, and deleting will also be communicated with the data store so that the data can be updated, returned, and deleted. As for the Flow chart design, which shows how the components work with each other when creating a new employee project in addition to returning data from the buffer.

System Requirements:

1. Dynamic memory allocation: Implementing storage memory using malloc helped manage the buffer by preserving the data and its integrity instead of direct access to the disk and due to repeated use of data and to improve the system’s performance and efficiency in sending and storing data.
2. Threading: Thread execution makes the system faster due to synchronization in operations and through the concurrency control component responsible for accessing data and its speed, which ensures data integrity and consistency and affects system productivity and response time.
3. Indexing: Implementing indexing helps the Records, Indexing, and File Manager component when a data request is sent from the execution engine. It helps in quick access to the data. For example, when sending a request for information about a specific employee, instead of searching for it in the stored files, it is searched for in the index, which is easy to access. To reduce response time and speed up the process
4. Interface: An easy-to-use interface has been provided that interacts with databases appropriately and implements all the components that the system needs for seamless integration with SQL statements and is compatible with the design of the data flow diagram and flow chart of the system in order to ensure that the system works correctly and properly.
5. Transaction manager: The system ensures the implementation of the ACID properties (atomicity, consistency, isolation, and durability) and the integrity of the transactions that are sent through its management and processing so that it ensures the consistency of the data and its safety from damage or failure of the system and records all changes that occur within the system to restore database data. In the event of a system failure to ensure business progress
6. Logging and Recovery: It ensures that a file is executed for each table in the system for data integrity and consistency. In the event of data loss, the files that have been lost are returned and retrieved. This is to preserve the data and its integrity and to prevent a system crash.

My system meets the requirements of the users and the system by designing the database in terms of tables and relationships between them, as well as designing the data flow diagram, which explains the operations performed by each user in addition to the transfer of data from the data store to the operations, and also designing the flow chart, which explains how the components work in The system, how data is transmitted between them, who requests the data, who sends and processes it, who receives it, how the data is stored on the disk, and the data that the system uses repeatedly in the buffer.

# Part2:

## Write the SQL queries based on the system

|  |  |  |
| --- | --- | --- |
| SQL Query | Function | User |
| INSERT INTO employees (employeeId, firstName, lastName, email, departmentId) VALUES (5,'Hasan', 'Ahmad', 'Hasan@gmail.com', 1); | insertEmployee() | Employee |
| INSERT INTO employeeProjects (employeeId, projectId, hoursWorked, status) VALUES (5, 1, 44, 'Finished'); | insertEmployeeProject() | Manager |
| UPDATE employees SET firstName = 'HASAN', lastName = 'ALHWIETAT', email = 'Alhwietat@gmail.com', departmentId = 1 WHERE employeeId = 5; | updateEmployeeByID(int empID) | Employee |
| UPDATE employeeProjects SET hoursWorked = 45, status = 'finshed' WHERE employeeId =5 AND projectId = 1; | updateEmployeeProjectByID(int empID, int projID) | Manager |
| DELETE FROM employeeProjects WHERE employeeId = 5; DELETE FROM employees WHERE employeeId = 5; | deleteEmployeeByID(int empID) | Manager |
| DELETE FROM employeeProjects WHERE employeeId = 5 AND projectId = 1; | deleteEmployeeProjectByID(int empID, int projID) | Manager |
| SELECT \* FROM employees; | displayEmployees() | Manager |
| SELECT \* FROM employeeProjects; | displayEmployeeProjects() | Manager |
| SELECT \* FROM departments; | displayDepartments() | Manager |
| SELECT \* FROM projects; | displayProjects() | Manager |
| SELECT \* FROM employees WHERE employeeId = 1; | searchEmployeeByID(int empID) | Employee, Manager |
| SELECT \* FROM employeeProjects WHERE employeeId = 1; | searchEmployeeProjectByID(int empID, int projID) | Manager |
| SELECT \* FROM departments WHERE departmentId = 1; | searchDepartmentByID(int depID) | Manager |
| SELECT \* FROM projects WHERE projectId = 1; | searchProjectByID(int projID) | Manager |
| SELECT \* FROM employees WHERE departmentId = 1; | searchEmployeeByDepartmentID(int deptID) | Employee, Manager |
| SELECT p.projectName, ep.hoursWorked, ep.status, p.startDate, p.endDate FROM employeeProjects ep JOIN projects p ON ep.projectId = p.projectId WHERE ep.employeeId = 1; | searchEmployeeProjectsJoinProjectsByEmpId(int empID) | Employee, Manager |
| SELECT e.firstName, e.lastName, p.projectName, ep.hoursWorked  FROM employees e JOIN employeeprojects ep ON e.employeeId = ep.employeeId JOIN projects p ON ep.projectId = p.projectId WHERE ep.projectId = 1; | searchEmployeeProjectsJoinEmployeesByProjId(int projID) | Manager |

## Criticize the process of mapping SQL queries to the developed CRUD statements and suggest improvements to optimize the developed CRUD queries. (Report)

updateEmployeeByID (int empID): You do not specify a specific field such as name or email to update, but rather you must update all fields to update the information of the specific employee. This increases the entry of data that you do not want to update and may affect performance. Therefore, when entering the employee ID, you must specify the data that you want. Update it and then enter the new data.

updateEmployeeByID (int empID): When employee data is updated, such as the ID, it is not updated in the EmployeeProject table. This causes many problems, as the employee cannot know the project he is working on when his information is updated, so it must be updated on both tables if the employee ID is changed.

insertEmployee (): When an employee is created, she may be more vulnerable to SQL injection attacks, so the string and not null are checked, but injection attacks are not checked, so security rules must be established to maintain the integrity of the data and not expose the databases to damage.

deleteEmployee (int empID): When a specific employee is deleted, he is permanently deleted from the system. There must be an option to delete, either hard or soft delete. This has a significant impact. For example, when calculating the number of hours worked on a specific project, when an employee is deleted, the total number of hours is not known. For the project, there must be an option to delete the data.

searchEmployeeProjectsJoinEmployeesByProjId (int projID): When a project is searched based on more than one table, it takes a lot of time. It must go to the first table, which is the project, and return the name of the project, then go to the Employees table and return the first and last names, then go to the EmployeeProject table and return the number of working hours on the project for each employee. This takes a lot of time in the research process, so the job must be threaded so that it works with the program simultaneously without slowing down the system and reducing time.

# Part3:

## Describe concurrency control techniques.

Concurrency Control: It is a component that manages simultaneous access to shared data. It ensures that transactions are executed in isolation, which allows some systems to execute many transactions at once. It must ensure that concurrent transactions are executed correctly to ensure data consistency. Which leads to increased overall productivity and system efficiency.

Techniques to control concurrency in DBMS: - (GeeksForGeeks, 05 Feb, 2024)

1. Lock-Based Protocols: Every transaction needs a lock in order to allow data to be accessed or modified, and there are two types of locks:
2. Shared Lock (s-mode): A read lock that allows multiple transactions to read data simultaneously so that it has read-only validity without modifying the data is called a read lock.
3. Exclusive Lock (x-mode): It is called a write lock. It allows a transaction to access data for reading and writing, so that only one transaction can hold the lock on a data element at one time.

Lock requests are submitted to the concurrent control manager through the programmer, and the transaction cannot be continued until the sent request is approved.

1. Timestamp based Protocol: Every transaction has a time attached to it. This time is the time at which the transaction enters the system. It helps arrange the transactions so that the transactions enter in the correct order.

# Part4:

## Testing Plan:

|  |  |  |  |
| --- | --- | --- | --- |
| Test case | Scenario | Expected result | Result |
| Input validation | Ensure that the values ​​entered into the database, such as name and ID, are not empty. In addition, the ID is an integer and not a string. | If the name is empty, the rest of the creation process will not be completed and it will return, and it must not be empty, and the id must also be integer. If it is not, the creation process stops and returns, and the id must be integer. | Pass |
| Id primary | Make sure that the ID is not repeated. | If the ID is duplicate, the creation stops and returns that the ID already exists and is stored. However, if it does not exist, the creation process will continue. | Pass |
| Email validates | Make sure the user entered the email correctly. | If an email is entered that does not include '@' and '.' The creation process will stop and return that the email does not contain '@' and '.' If it is included, we will continue the success process. | Pass |
| Deleted employee | Make sure that the employee data deletion is done correctly by deleting from the table that has the foreign key | When deleting employee data from the file by specifying the number of the employee you want to delete, in case of success, it returns that it has been deleted from the employee file and from the employee’s project file, and in case of failure, it returns that it has failed in the deletion process. | Pass |
| Insert in File | Ensure that the process of creating data within the file is done correctly. | When adding data to the file by entering employee information, it will return in the event of success that it was created, but in the event of failure it will return that the creation process was unsuccessful. | Pass |
| Update in file | Ensure that the process of updating data within the file is done correctly. | When you update the data in the file by entering the employee number, and then entering the new information, it will then return that the update process succeeded or that the process failed. | Pass |
| delete in file | Ensure that the process of deleting data within the file is done correctly. | When you delete the data in the file by entering the employee’s ID, the employee will be deleted, and it will return that the deletion process was completed successfully. However, if the employee you want to delete is not found, it will return that he does not exist. | Pass |
| Display in file | Ensure that all data within the file is displayed correctly. | When all data from the file is displayed, in case of success, it returns all employees in the file, and in case of failure, it returns that there are no employees | Pass |
| Search in file | Ensure that the process of searching for specific data within the file is done correctly. | When the data in the file is searched by entering the employee’s ID, the employee will then be searched for and will return the employee’s data if it exists, and if it does not exist, it will return that it does not exist. | Pass |
| Dynamic Memory allocation | Ensure that the process of reserving a storage location using malloc is done correctly. | If the malloc creation fails, it will be returned that the malloc creation process failed, and in the event of success, the operations will continue and we will confirm by storing and displaying the data. | Pass |
| Menu choice | Make sure that the process of entering numbers in the Menu works well. | If you enter numbers that are not present in the Menu, the number will be returned as invalid, but if you enter the number as correct, it will enter the specified process. | Pass |
| Hash (indexing) | Ensure that any employee entry is stored within the hash table | When you create an employee and enter all the data for creation, he will calculate the hash table and store it. In case of success, it will continue, and in case of failure, it will return that the operation failed. | Pass |
| Thread | Ensure that the thread is created and functioning by displaying all employee data. | When the thread is created, we will check if the creation process fails, and if it succeeds, the process will continue and then all employee data will be returned when called. | Pass |

## Evaluate the DBMS in terms of Improvements needed to ensure the continued effectiveness of any database (Report)

In the future, the system that was built needs additional improvements to improve performance, increase efficiency and productivity, and ensure data integrity and protection. For example, the lock table component responsible for protecting data can be implemented by locking it from unauthorized access through the use of Exclusive Lock and Shared Lock. These techniques can be used to protect data. Timestamp in order to associate with each transaction within the system a time associated with it so that the transactions are arranged to obtain data, which increases the security of the system that was built and knowing the details in the event of problems. It is also possible to place servers responsible for storing data and sending it to buffers. This helps to speed up the speed. Responding and receiving more requests. The indexing principle can also be used better to improve the performance of queries and facilitate access to them in the event of greater data enlargement, in addition to using the principle of multiple threads instead of using the principle of a single thread, so that it helps to balance the system and distribute the loads.

It is also possible to add a new row to the department table containing the name of the person responsible for each department in the company, in addition to writing a new SQL Query for the search that returns the names of the employees, the department, the department manager, and the number of working hours for each employee in this department. This helps the company know which employees are the most accomplished in the company. To put in the effort and reward them, there are many improvements that can be made in order to improve the system in the future, improve the user experience, and maintain the database system and its efficiency.