



Name/id: abdelkareem yousef mamdoh soubar/19110022

Assignment Title: Database Programming

Course Tutor: Dr. Ahmad Barghash

Submission Date: January 31st, 2022

Problem

i have been requested to create a database management system for a phone shop, the system should be able to take the employees data in, and it should contain the data of the available phones in shop, this system should be capable of count and alert the owners when the stock is low or an plus it can do some delete, backup and searching for and all enquirers are update able changeable plus editable which means the owner can add/ remove tables and edit them for future update.

The mapping

employee(id, fname ,lname ,pos ,gender ,salary ,year ,month ,day ,char ,address)

phones (id, name, brand, color, price, count)

customer (id, name, gender, phone_number, year, month, day, email)

business use case

HR need to add employees to the employees table.

Employees need to the phone/ products to the phone table.

customer need to be inserted by the employees in to the system.

part1

1-

Storage: have the data stored and I used it by creating a Doubly Linked List

Storage manager: it has control over storage and it can control the way of storing the data.

Index Manager: it provides help to find index location so you can do an action in that index such as add,delete,edit,.

Query Compiler: its the place were you can write SQL query in in the project it was the base when the functions were written .

Execution Engine: its the translator between us and the machine so the machine can execute in the project its the interface.

Buffer manager: its were temporary data is saved and its a step during execution its used while adding a new table .

Logging and recovery: it place more protection by that in case of anything happens to the data base you can always check the recovery and logs why it happened in the project it was placed as an optional thing when your done .

DDL Compiler:it has the information such as storage details and name of the file plus data items so (metadata) in the project it was presented as FILE to write and read files plus mapping across.

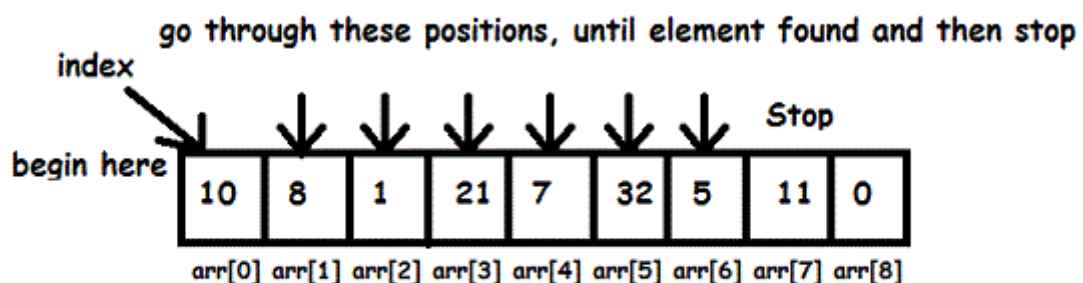
Transaction Manager: It has the ability to coordinate transactions across one or more resources.

Concurrency Control: it makes sure that the database is performing accurately it's transactions without having it to break the rules of data integrity of the database it was used on look of a sequence.

2-

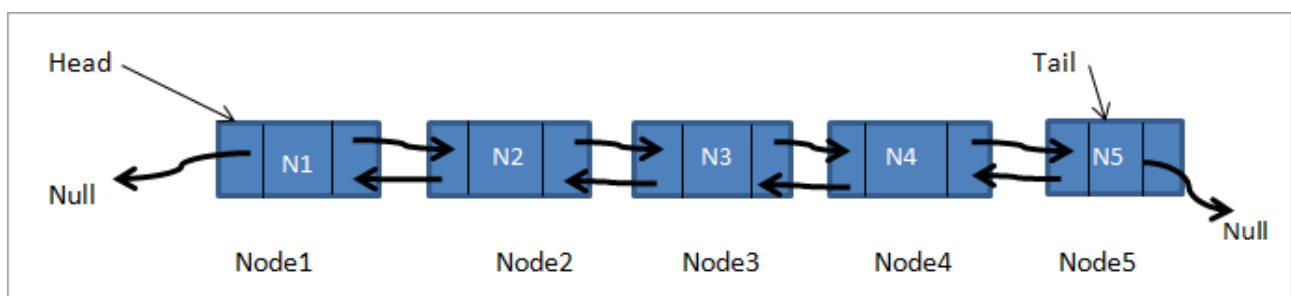
the phone shop requested me build them a database management system and the system should contain basic function create, delete, edit and view and it must contain tables for employees, phone and customers all basic function must be able to deal with the tables so it would be able to tell the count of how many phones are in the shop with details and who worked in the shop with positions and salary and be able to know who the customer are with their detail's and all of that need to run of a computer with main OS on it Linux arch/Debian based with updated gcc compiler to compile c language this device's requires at minimum a keyboard a and have the internal hardware of i3 12th gen 8 GB of ram ups for when the power goes down 256 GB of storage and w mother board that comes with 1Gb of Ethernet connection so stay updated

3-in this program data transfer between components is based data structure and algorithms but the way they got linked in the programs is a bit hard to get now as a start for searching in general, I used a Linear search algorithm which is based on the going through each record in the data base until the element is found



Element to search : 5

as for the tables, it was dealt with them as Doubly Linked List but not in a direct way by that I mean that I don't have a function called Doubly Linked List but across the entire program its being used for transfer and talk to the tables



threading was used as final step in the program for backing up and exit .

4-

the program design is fully functional with of add, delete, update, search, quit and backup and exit with the interface and it contains all of the tables and what the phone needs from knowing who works in the shop and the HR hiring on the system even for the cutromers accounts, as for the interface it was build simple so the user can read and make choice of what the user needs.

part2

2

all of the examples are going to be on the customer table had been made and build in the c program but if we were using SQL we would write those commands.

Add in the c program

```
INSERT INTO Customer (id, fname, lname, position, gender, salary, year, month, day, Address)
VALUES ('1','kareem','sober','management','male','8900','2022','2','6','Norway');
```

update in the c program

```
UPDATE Customer SET name='hamziah' WHERE CustomerID=3;
```

delete in the c program

```
DELETE FROM Customer WHERE id='3';
```

view in the c program

```
SELECT * FROM Customer;
```

3

a Doubly Linked List was used in the project to interact with the tables through that it was global on the program which means the adder used the list and some buffers to add to the list the data and in the search the data was called multiple time to check and compare because it was based off Linear search which means it had to though each element as for view the interaction with list was the same but here it prints the data each time it gets it/call it unlike the delete, the delete call the list for comparing but when it finds the target row it delete it's it as for the back up it starts a thread that takes the three tables and places them on a separate file's and save them then exits the program.

4

in the project the way that I used to map was call and compare the data in every case but the add. In the add I inserted data so their was no comparing and what made the call easy is that I used Doubly Linked List for controlling the tables and it was global which means that I had accesses across the program over the list (tables).

prt3

1

DBMS concurrency control techniques

Concurrency management in an exceedingly management System could be a methodology of dominant several actions at constant time while not their meddlesome. It guarantees that info transactions ar completed in an exceedingly timely and correct manner to deliver correct results while not jeopardizing the knowledge base's data integrity.

Lock-Based Protocols.

The Two section lockup Protocol, typically referred to as the 2PL protocol, could be a technique of concurrency management in package that maintains serializability by securing dealing knowledge with a lock that stops future transactions from accessing identical knowledge at identical time. The Two-Phase lockup mechanism aids within the elimination of package concurrency problems.

Two Phase Locking Protocol.

Strict-Two section protection is sort of a dead ringer for 2PL. Strict-2PL differs solely therein it ne'er releases a lock once it's been used. It keeps all the locks till the commit purpose so releases all of them at constant time once the procedure is finished.

Timestamp-Based Protocols.

In a management system, a timestamp-based protocol could be a technique that serializes the execution of synchronous transactions by mistreatment the System Time or Logical Counter as a timestamp. The timestamp-based protocol assures that each one conflicting scan and write operations ar allotted within the sequence such that by the timestamp.

Validation-Based Protocols.

Validation-based Protocol in direction Systems (DBMS), conjointly called Optimistic Concurrency management Technique, may be a technique for avoiding dealing concurrency. Rather of change the info itself, this protocol updates native copies of the dealing information, leading to less interference throughout dealing execution.

part4

2

as a test plan I would say that you need to minimum create 2 items in one of the tables when your done go check if the view works then go the search and search for on of their id's when that s done go and and delete on of them afterword go to backup and exit then recreate the same possess for all tables.

3

I am going to start by saying that the system is based off 4 main functions add, delete, update and view, and more for later but I have to say before I start explaining that for all this to work I had to put some ground rolls and by that I mean that I made the choices between making variables global to intact with functions and transmit data on more large scale than return or I can make the table global and keep on editing it from any where in the code so I made the second choice which is to make the table why you would say ?

Making the tables global would mean I can edit them easily from any place in the code by calling the right pointers.

The tables them self's are made using struct each table and each struct contained the details of the table its meant to be then the stuct was pointed at it so I multi locations in address memory so I can set a limit by using array in pointers and each one of the arrays is editable on its own for each array

to get the right memory size located for it I used malloc before adding details in the table the problem with malloc that when it locates it can add data with rubbish/random values to because it booked the space in the memory.

I started everything by creating the add function, and it based off taking each table looking at its components and checking each one of them to fill them with data and the user inputs through the interface the function takes pointer count from main and which table number(name) to add to.

As for the search, it only looks for Id because id is unique for and all tables have Id, so the user enter the table number that its being looked for and afterwards it goes to ask the user to input the id , now the search process is really simple it loop goes through the limit in each on in the limit it looks in the table for the id when it gets found, it would print all the details about the id which means the id indicates which row it is and then the entire row gets printed.

As for the delete, its was build on the search but search prints the row delete remove the row and it s the same Idea the user enters the table number and id then the loop goes through the id's in a specific table but here it removes it instead of print it.

Update is a bit more different than others , update starts by asking the user about the table to change into it, now a new loop starts and it asks the user about the which number columns do want to change when entered the programs through a switch case to check which which columns and then takes input again for that specified spot on the table.

View is really simple its based on a loop after the user input's the desired table to be viewed and the loop prints every item in the table to the interface.

Backup and exit and its a function, basically it open up a thread the thread takes all of the tables and writes them in a text file each table gets a text file after the writing is complete the program closes

interface was designed to be simple to use all of the operation's are on the interface and if the program requires anything from the user, it would take the input from the same interface.

