Data structure Project

DBMS

OS group no.6 :

Abdelrahman Nabil Ragab Mohamed Elhoshy

Aliaa Abd Elhalim Abd Elhai

Mohamed Ahmed Soliman Baghdad

Sara Elsayed Elsayed Elskely

DBMS

The Programs steps are simply :

-Writing the SQL command:
-where SQL command is written in file called "sqlcom.txt" placed in C:\ drive & saving it
_Opening the Project file.c

-Compiling the file:
-the program reads input command from the text file & palce it in char *command
-SQL command is then analysed & Tokenized according to the SQL command Keyword
-each Keyword runs a function for a different command
-Handling each function depends on the data that comes from analysing the command &

The Implementation design depends on 2 files (meta file, table file) & retrieving data from them

*The meta file contains the table name & info to be stored in the meta file
where:
a struct meta holds a set of(colum order, column header name)
except for the first struct that holds (number of columns in the table, table name)
each struct node is saved in the file line by line in the format "%d %s\n"

```
struct meta
{
int colorder;     //column order except first node
char name[20];};
```

*The table file contains all table in the form of cells where cell values are treated as an array of characters regardless of their datatype where: a struct cell holds a set of (row order,column order,cell data as "value") each struct node is saved in the file line by line in the format "%d %d %s\n"

```
struct cell
{
int colorder;
int roworder;
char value[20];
};
```

*The row & column information are saved as "One-based" indexing

*By reading the last line in the table file gives us the last cell node in table where the information it holds gives us the number of rows & columns in the table

*By getting the number of rows & columns in the table we dynamically allocate a "Zero-based" 2D array of rows & columns that holds values of struct cell node

*Another dynamic allocation for an array of struct meta array[columns+1] that holds the information of table headers

*Each function for a SQL command deals with these data to get the desired output

We have covered the main commands in our DBMS,

like:

DDL -Data Definition language- in Create command.

DML –Data Manipulation language- in Insert , Update and Delete.

DQL –Data Query Language- in Select command.

Here is a list of command syntax used in this program:

(1)Create command:

```
CREATE TABLE table_name(
column1 datatype,
column2 datatype,
column3 datatype,
.....);
```

Ex: create table employee (name char, salary int, id int);

(2)Select command:

```
Select column1,column2,columnN from table_name.

Select * from table_name.
```

Ex: Select name,salary from employee;

Select * from employee

(3)Update command:

Update table_name Set column1='value1',column2='value2'

Where condition='value'.

Ex: Update employee set name='ahmed',salary=5000

Where id=1;

(4)Delete command:

Delete from table_name where condition=value.

Ex: delete from student where sid = 3;

(5)Insert command:

Insert Into table_name values (value1,value2,value3).

Insert into table_name (column1, column2, column3) values (value1, value2, value3).

Ex: insert into employees (id, name, salary) values (2,aly, 4000);

Insert into employees values (2, aly, 4000);

That’s the syntax we should stick to and any other syntax may cause unwanted problems.