**Final Report on CS Project**

***RUN SQL.CPP TO TEST THE PROGRAM***

**A Project on SQl C++ Connectivity**

Team Members : -

* Pradeep Yadav
* Parth Loya
* Rahul Singhal
* Sandeep Subramanian
* Preetham
* Payal
* Prateek

**Contents :-**

* Introduction
* Functionality of SQL used
* User manual
* Working
* Header Files used
* Libraries used
* Assumptions used
* Future of the project
* Evaluation of the members
* Bibliography
* Special thanks

**Introduction : -**

SQL - Sql is a database management system in which user can have an access over the given database.SQL (often referred to as Structured Query Language) is a programming language designed for managing data in relational database management systems (RDBMS).

Originally based upon relational algebra and tuple relational calculus, its scope includes data insert, query, update and delete, schema creation and modification, and data access control.SQL was one of the first commercial languages for Edgar F. Codd's relational model, as described in his influential 1970 paper, "A Relational Model of Data for Large Shared Data Banks". Despite not adhering to the relational model as described by Codd, it became the most widely used database language.

Though often described as, and to a great extent is a declarative language, SQL also includes procedural elements. SQL became a standard of the American National Standards Institute (ANSI) in 1986, and of the International Organization for Standards (ISO) in 1987. Since then the standard has been enhanced several times with added features. However, issues of SQL code portability between major RDBMS products still exist due to lack of full compliance with, or different interpretations of the standard. Among the reasons mentioned are the large size, and incomplete specification of the standard, as well as vendor lock-in.

Functionalities of sql used –

 Data types:-  
char  
int  
float  
  
Functional Capabilities:-  
not   
select  
where   
from  
and  
or  
not  
average  
min  
max  
count  
distinct  
\*  
arithmatic operations  
natural join  
pattern matching  
set operations  
order by  
group by  
with as  
nested queries  
 having

User manual :-

**THE TABLES AVALAIBLE FOR QUERY ARE INFO , INFO1,INFO2,INFO3,INFO4**

**General syntax given by user**:

select \*/distinct "column name"/ all "column name"/("column name", "column name".....) from "table name"; /order by/ where "conditions" and/or/not "conditions" ; / group by "column name" ; / having aggregate function("column name") "inequality"(< ,>, =) "value"

**Examples**:

1.select name, salary from table1 order by id;

2.select distinct name from table1 where name>'ram' and name<'sita' having id=45;

3.select name , sum(salary) from table1 where age>34 or age=23 group by name having sum(salary)>12000;

4.select distinct name from table2 where name>'parth' and not name='rahul' order by name desc;

**Working :-**

The steps of working of the program : -

* The program will be compiled and executed
* User will type the requisite command as per the correct SQL syntax in the command line .
* The syntax will be first checked for the validity of the syntax
* The data will be read from the table which is the file name in our project in a 3D char array.
* Then the command will be executed in similar fashion as sql.
* The result will be displayed on the terminal

**Coding Logics : -**

* Validity check :-
* The individual components of SQL command is accessed by command like “strtok” and the sequence of strings are logically intertwined as per the SQL intelligence.
* Rea**ding the database :-**

Works on 3D array principle - the text has been disintegrated in the form 3D arrays through which the individual elements of the database can be accessed.

* Implementation of the query :-

1. The sql Query are disintegrated into various substrings

2. The substrings are then read and detected as per the sequence.

3. The necessary function is called corresponding to the string.

4. The logic is repeated over all the substrings

**Header Files used : -**

* **general.h**
* **comfun.h**
* **essenfun.h**
* **sql.h**
* **tempfun.h**
* **wherefun.h**
* **stdio.h**

**Libraries Used :-**

* **string**
* **iostream**
* **cstring**
* **cstdlib**

Assumptions : -

* we are using "only one table at a time" sort of sql and not multiple tables at a time.
* the name of databases should not have .txt extension.
* the name of the columns as well as those of database files must be in capital.
* the database assumed is of the following format -

1. the first row contains the column name along with datatype of the column separated by ","
2. the two columns are separated by tabs even the last column is followed by tabs
3. after each row including the last "enter" is pressed

* we have currently not specified "between" and "like" - if user wants the results of "between " in his syntax he can do so by using inequality operators . For ex - if user wants name of the employee having salary between 10,000 and 20,000 he can do it so by -

select name from table1 where salary > 10000 and salary < 20000 ;

* as we are using function atof(char\*p) which converts a string directly into a float value, this function returns a value zero hence not using zero as an element of anycolumn of table
* operating on only float ( all int converted into float)and that too with 4 places after decimal point.

**Future planning :-**

* the code can be extended to multiple data files at a time
* the code can be enhanced by handling syntaxes like between , like , etc
* the intelligence of code can be improved further by differentiating the int datatype from float one.
* all the functions and capabilities of sql like updation , deletion and insertion can be incorporated inthe program

**Devotion of the members (Time input in hr:min ) :-**

Name D1 D2 P T D M

payal 5:30 3 8 8 - -

pradeep 5 4 18 4 2 -

sandeep 10 12 8 30 - 3

rahul 10 22 12 20 - 5

parth 8:15 5 6:30 1:20 5 6:20

preetham 10 6 10 4 - 5

prateek 4:30 - - - - 1

Note -

D1 - Discussion D2 - Design D3- Documentation

P - Programming T - Testing M - Miscellaneous

**Evaluation of various members :-**

Name Roll no Peer reviewed marks(max=10)

parth loya 110040068 6.75

rahul singhal 110050023 10

sandeep 110260028 10

preetham 110050073 7.5

pradeep 110010033 7

payal 115090008 6.75

prateek 110040118 1

**Bibliography :-**

Various sites used : -

* [www.w3schools.com](http://www.w3schools.com/)
* [www.cplusplus.com](http://www.cplusplus.com/)
* [www.mysql.com](http://www.mysql.com/)

**Special Thanks :-**

We are quite grateful to **Prof. D.B Phatak** for their constant motivation and guidance throughout the project . We thank him for extending this platform to us. Also we extend a warm gratitude towards our **T.A Rahul Kumar** who directed us throughout the project and cleared our queries….

**Last Word :-**

The project we took was quite enormous and had infinite expansional possibilities. The work can be extended to the other spheres of SQL management including the updation and deleting part of the RDBMS….