Database Design
CS 6360.003(Spring 2020)
SQL Programming Project 2: Project Report
Divya Birla
2021514344

1.OBJECTIVE:

The goal of this project is to implement a (very) rudimentary database engine that is loosely based on a hybrid between MySQL and SQLite and is called DavisBase.

2.FUNCTIONAL CAPABILITIES OF APPLICATION:

Upon launch the prompt **davisql>** is presented wherein we can key in the required DDL, DML or VDL commands we want to execute.

2.1 Extra Command

- HELP:
  This command allows us to see the guide of all the commands that the database is currently capable of processing.

```
Anshumans-MacBook-Pro:DavisBaseMaster riku$ javac DavisBase.java
Anshumans-MacBook-Pro:DavisBaseMaster riku$ java DavisBase
--------------------------------------------------------------------------------
Welcome to DavisBase
--------------------------------------------------------------------------------
davisql> help;

        SHOW TABLES;                                                    Display all the tables in the database.
        CREATE TABLE table_name (<column_name datatype>);               Create a new table in the database.
        INSERT INTO table_name VALUES (value1,value2,..);               Insert a new record into the table.
        SELECT * FROM table_name;                                       Display all records in the table.
        SELECT * FROM table_name WHERE column_name operator value;      Display records in the table where the given condition is satisfied.
        DROP TABLE table_name;                                          Remove table data and its schema.
        HELP;                                                           Show this help information.
        EXIT;                                                           Exit DavisBase.

davisql>
```

2.2 DDL Commands

- SHOW TABLES:
  This command allows us to see all the tables currently in the database.

```
Anshumans-MacBook-Pro:DavisBaseMaster riku$ javac DavisBase.java
Anshumans-MacBook-Pro:DavisBaseMaster riku$ java DavisBase
--------------------------------------------------------------------------------
Welcome to DavisBase
--------------------------------------------------------------------------------
davisql> help;

        SHOW TABLES;                                                    Display all the tables in the database.
        CREATE TABLE table_name (<column_name datatype>);               Create a new table in the database.
        INSERT INTO table_name VALUES (value1,value2,..);               Insert a new record into the table.
        SELECT * FROM table_name;                                       Display all records in the table.
        SELECT * FROM table_name WHERE column_name operator value;      Display records in the table where the given condition is satisfied.
        DROP TABLE table_name;                                          Remove table data and its schema.
        HELP;                                                           Show this help information.
        EXIT;                                                           Exit DavisBase.

davisql> show tables;
--------------------
table_name          |
--------------------
davisbase_tables    |
davisbase_columns   |
univeristy          |
complex             |
univ                |
student             |
companies           |

davisql>
```

- CREATE TABLE:
  This command allows us to create new tables in the database.

```
Anshumans-MacBook-Pro:davisbasemaster riku$ java DavisBase
--------------------------------------------------------------------------------
Welcome to DavisBase
--------------------------------------------------------------------------------
davisql> show tables;
--------------------
table_name          |
--------------------
davisbase_tables    |
davisbase_columns   |
univeristy          |
complex             |
univ                |
companies           |
library             |

davisql> create table countries (Cid INT, Cname TEXT);
davisql> insert into countries values (1,India);
davisql> select * from countries;
--------------
cid   |cname   |
--------------
1     |india   |
davisql>
```

- DROP TABLE:
  This command allows us to drop existing tables from the database.

```
Anshumans-MacBook-Pro:DavisBaseMaster riku$ java DavisBase
--------------------------------------------------------------------------------
Welcome to DavisBase
--------------------------------------------------------------------------------
davisql> show tables;
--------------------
table_name          |
--------------------
davisbase_tables    |
davisbase_columns   |
univeristy          |
complex             |
univ                |
student             |
companies           |
library             |

davisql> select * from student;
---------------------------------------
sid   |sname                |scourse   |
---------------------------------------
1     |divya birla          |cs6360    |
2     |william shakespeare  |cs6375    |
3     |enid blyton          |cs5343    |
davisql> drop table student;
davisql> show tables;
--------------------
table_name          |
--------------------
davisbase_tables    |
davisbase_columns   |
univeristy          |
complex             |
univ                |
companies           |
library             |

davisql>
```

2.3 DML Commands

- INSERT INTO TABLE:
  This command allows us to insert into existing tables in the database.

```
Anshumans-MacBook-Pro:davisbasemaster riku$ java DavisBase
-------------------------------------------------------------------------------
Welcome to DavisBase
-------------------------------------------------------------------------------
davisql> show tables;
--------------------
table_name          |
--------------------
davisbase_tables    |
davisbase_columns   |
univeristy          |
complex             |
univ                |
companies           |
library             |

davisql> create table countries (Cid INT, Cname TEXT);
davisql> insert into countries values (1,India);
davisql> select * from countries;
--------------
cid   |cname   |
--------------
1     |india   |
davisql>
```

2.4 VDL Commands

- SELECT-FROM:

This command allows us to query existing tables from the database to display all records present in a table.
Combined example of this command and the same added with a where clause is presented in the section below.

- SELECT-FROM-WHERE:
  This command allows us to query existing tables from the database to display all records present in a table that satisfy a certain condition of the WHERE clause.
  The condition can use a variety of logical operators such as : =(equal to), >(greater than), <(less than), >=(greater than or equal to) and <=(less than or equal to).

```
davisql> select * from student;
-------------------------------------
sid   |sname                  |scourse   |
-------------------------------------
1     |divya birla            |cs6360    |
2     |william shakespeare    |cs6375    |
davisql> select * from student where sid=1;
-------------------------------
sid   |sname          |scourse   |
-------------------------------
1     |divya birla    |cs6360    |
davisql> select * from student where scourse=cs6375;
-------------------------------------
sid   |sname                  |scourse   |
-------------------------------------
2     |william shakespeare    |cs6375    |
davisql> select * from student where sid>1;
-------------------------------------
sid   |sname                  |scourse   |
-------------------------------------
2     |william shakespeare    |cs6375    |
davisql> select * from student where sname=divya birla;
-------------------------------
sid   |sname          |scourse   |
-------------------------------
1     |divya birla    |cs6360    |
davisql> insert into student values (3,Enid Blyton, cs5343);
davisql> select * from student;
-------------------------------------
sid   |sname                  |scourse   |
-------------------------------------
1     |divya birla            |cs6360    |
2     |william shakespeare    |cs6375    |
3     |enid blyton            |cs5343    |
davisql> insert into student values (1,Dan Brown, cs6320);
Key constraint violation
```

The last example in the above screenshot demonstrates Entity constraint violation.

```
Anshumans-MacBook-Pro:davisbasemaster riku$ javac DavisBase.java
Anshumans-MacBook-Pro:davisbasemaster riku$ java DavisBase
--------------------------------------------------------------------------------
Welcome to DavisBase
--------------------------------------------------------------------------------
davisql> help;

        SHOW TABLES;                                            Display all the tables in the database.
        CREATE TABLE table_name (<column_name datatype>);       Create a new table in the database.
        INSERT INTO table_name VALUES (value1,value2,..);       Insert a new record into the table.
        SELECT * FROM table_name;                               Display all records in the table.
        SELECT * FROM table_name WHERE column_name operator value;   Display records in the table where the given condition is satisfied.
        DROP TABLE table_name;                                  Remove table data and its schema.
        HELP;                                                   Show this help information.
        EXIT;                                                   Exit DavisBase.

davisql> show tables;
--------------------
table_name          |
--------------------
davisbase_tables    |
davisbase_columns   |
univeristy          |
complex             |
univ                |
companies           |
library             |
countries           |

davisql> create table players (Pid INT NOT NULL, Pname TEXT, PTeam TEXT);
davisql> insert into players values (NULL, Lionel Messi, Argentina);
Entity constraint violation

davisql> insert into players values (1,Lionel Messi, Argentina);
davisql> select * from players;
--------------------------------
pid   |pname          |pteam      |
--------------------------------
1     |lionel messi   |argentina  |
davisql>
```

The above screenshot demonstrates Entity constraint violation.

- EXIT:
  This command allows the user to exit the davisql prompt .

```
Anshumans-MacBook-Pro:DavisBaseMaster riku$ java DavisBase
--------------------------------------------------------------------------------
Welcome to DavisBase
--------------------------------------------------------------------------------
davisql> show tables;
--------------------
table_name          |
--------------------
davisbase_tables    |
davisbase_columns   |
univeristy          |
complex             |
univ                |
student             |
companies           |
library             |

davisql> select * from student;
---------------------------------------
sid   |sname                 |scourse   |
---------------------------------------
1     |divya birla           |cs6360    |
2     |william shakespeare   |cs6375    |
3     |enid blyton           |cs5343    |
davisql> drop table student;
davisql> show tables;
--------------------
table_name          |
--------------------
davisbase_tables    |
davisbase_columns   |
univeristy          |
complex             |
univ                |
companies           |
library             |

davisql> exit;

Anshumans-MacBook-Pro:DavisBaseMaster riku$
```

3.DOCUMENT VERSION:

VERSION NUMBER	CREATED BY	DATE
V1.1	Divya Birla	2020-05-05