MaxwellBase User Guide

Opening the MaxwellBase Engine:

1. Check that Java is installed on your system using this link: https://www.java.com/en/download/verifyjavaversion.jsp
   The version used for this project was Java 18 and above.
2. Open a command line (i.e cmd for windows).
3. Navigate to the MaxwellBase\src folder in the command line.
4. Compile the .java by using the command:

```
~\MaxwellBase\> javac src/MaxwellBase/* src/Constants/* -d out/
```

5. Upon successful compilation, finally execute the java program with:

```
~\MaxwellBase\src\MaxwellBase> java MaxwellBase.java
```

OR

1. Open any IDE and import the MaxwellBase folder.
2. Run the MaxwellBase.java file.

The resulting command line should look like this.

```
--------------------------------------------------------------------------------
Welcome to MaxwellBase
MaxwellBase Version v1.0
©2022 Team Maxwell

Type "help;" to display supported commands.
--------------------------------------------------------------------------------
maxwellsql>
```

From here, commands can be typed to start doing database operations. For help with what commands can be used, type “help;”.

Creating and showing tables:

1. Tables in MaxwellBase can be created using the CREATE TABLE table_name clause and command.

```
CREATE TABLE table_name (
        column_name1 data_type1 [NOT_NULL][UNIQUE],
        column_name2 data_type2 [NOT_NULL][UNIQUE],
        ...
);
```

2. Additionally, after a table has been created, the list of tables can be shown with the SHOW TABLES command:

```
SHOW TABLES;
```

The maxwell_columns and the maxwell_tables are automatically generated by the MaxwellBase engine and keep metadata about the tables.

3. Here is an example of the commands in action:

```
maxwellsql>  create table student ( name text primary_key, id int not_null unique primary_key, gpa int not_null);
Case: CREATE
Stub: parseCreateTable method
Command: create table student ( name text primary_key , id int not_null unique primary_key , gpa int not_null )
maxwellsql> show tables;
Case: SHOW
Command: show tables
----------------------
|  table_name          |
----------------------
|  maxwellbase_tables  |
|  maxwellbase_columns |
|  student             |
----------------------
```

INSERTing Records from a Table:

1. Now that table has been made, it is time to add records to it. Records can be added to a table using the INSERT command. The syntax below:

```
INSERT INTO table_name(column_list) VALUES (value1,value2,value3, ...);
```

2. Here is an example of inserting values:

```
maxwellsql> insert into student values(yash,1,6);
Case: INSERT
Command: insert into student values ( yash , 1 , 6 )
Stub: This is the insertRecord method
maxwellsql> insert into student (name,id,gpa) values(sai,2,4);
Case: INSERT
Command: insert into student ( name , id , gpa ) values ( sai , 2 , 4 )
Stub: This is the insertRecord method
```

Notice the difference between the first and second INSERT command. The first command does not specify the column names, but the second command does. However, both commands insert into the same columns because when columns are not specified for values, the first n columns are inserted into. In other words, the columns follow ORDINAL_ORDER for which values will be inserted when the columns are not mentioned.

SELECTing Records from a Table

1. To look at the data inserted into a table use the SELECT command. The syntax is shown below:

```
SELECT *
FROM table_name
WHERE [NOT] condition;
```

Note the * wildcard symbol will select all columns from table_name (except for rowid). The * can be replaced with a column name if you want just certain columns to be selected and printed to the application.

The where condition syntax for String type columns does not have quotations around value.

EX: WHERE name = sai; (CORRECT) , WHERE name = “sai” (INCORRECT)

2. Here is an example of selecting from the Students table from the previous examples

```
maxwellsql> select * from student;
Case: SELECT
Command: select * from student
------------------
|  name | id | gpa |
------------------
|  yash | 1  | 6   |
|  sai  | 2  | 4   |
------------------
maxwellsql>
```

SELECTing on a Condition with WHERE

1. You may have noticed the WHERE from the syntax earlier. A powerful tool with relational databases is the ability to query the database based on a condition. Currently MaxwellBase supports a single simple WHERE clause condition.
2. Here is an example that shows the WHERE clause in use. It also shows a little more about the metadata tables that upkeep that tables made the user:

```
maxwellsql> select * from maxwellbase_columns where not table_name = student;
Case: SELECT
Command: select * from maxwellbase_columns where not table_name = student
------------------------------------------------------------------------------------------------
|  table_name          | column_name      | data_type | ordinal_position | is_nullable | column_key |
------------------------------------------------------------------------------------------------
|  maxwellbase_tables  | table_name       | TEXT      | 1                | No          | PRI        |
|  maxwellbase_columns | table_name       | TEXT      | 1                | No          | NULL       |
|  maxwellbase_columns | column_name      | TEXT      | 2                | No          | NULL       |
|  maxwellbase_columns | data_type        | TEXT      | 3                | No          | NULL       |
|  maxwellbase_columns | ordinal_position | TINYINT   | 4                | No          | NULL       |
|  maxwellbase_columns | is_nullable      | TEXT      | 5                | No          | NULL       |
|  maxwellbase_columns | column_key       | TEXT      | 6                | No          | NULL       |
------------------------------------------------------------------------------------------------
maxwellsql>
```

The student table is nowhere to be found since the query asked for NOT named student.

3. Here are more examples of using the SELECT clause with WHERE. This example shows what the meta-data tables store in to give some hints what is going on behind the scenes with the MaxwellBase engine.

```
maxwellsql> select * from maxwellbase_columns where table_name = student;
Case: SELECT
Command: select * from maxwellbase_columns where table_name = student
--------------------------------------------------------------------------------
|  table_name | column_name | data_type | ordinal_position | is_nullable | column_key |
--------------------------------------------------------------------------------
|  student    | name        | TEXT      | 1                | NO          | PRI        |
|  student    | id          | INT       | 2                | NO          | PRI        |
|  student    | gpa         | INT       | 3                | NO          | NULL       |
--------------------------------------------------------------------------------
maxwellsql>
```

```
maxwellsql> select * from maxwellbase_columns;
Case: SELECT
Command: select * from maxwellbase_columns
-----------------------------------------------------------------------------------------------
|  table_name          | column_name      | data_type | ordinal_position | is_nullable | column_key |
-----------------------------------------------------------------------------------------------
|  maxwellbase_tables  | table_name       | TEXT      | 1                | No          | PRI        |
|  maxwellbase_columns | table_name       | TEXT      | 1                | No          | null       |
|  maxwellbase_columns | column_name      | TEXT      | 2                | No          | null       |
|  maxwellbase_columns | data_type        | TEXT      | 3                | No          | null       |
|  maxwellbase_columns | ordinal_position | TINYINT   | 4                | No          | null       |
|  maxwellbase_columns | is_nullable      | TEXT      | 5                | No          | null       |
|  maxwellbase_columns | column_key       | TEXT      | 6                | No          | null       |
|  student             | name             | TEXT      | 1                | NO          | PRI        |
|  student             | id               | INT       | 2                | NO          | PRI        |
|  student             | gpa              | INT       | 3                | NO          | NULL       |
-----------------------------------------------------------------------------------------------
maxwellsql>
```

UPDATE

1.To Update data in a table use the UPDATE SET command. The syntax is shown below:

```
UPDATE table_name SET column_name = value
WHERE condition;
```

2. Below is an example of how to use update command

```
maxwellsql> select * from students;
Case: SELECT
Command: select * from students
------------------
| name | id | gpa |
------------------
| yash | 1  | 6   |
| sai  | 2  | 4   |
------------------
maxwellsql> update students set gpa = 3;
Case: UPDATE
Command: update students set gpa = 3
Stub: This is the parseUpdate method
2 rows updated!
maxwellsql> select * from students;
Case: SELECT
Command: select * from students
------------------
| name | id | gpa |
------------------
| yash | 1  | 3   |
| sai  | 2  | 3   |
------------------
maxwellsql>
```

DELETE

1.To delete data in a table use the DELETE command. The syntax is shown below:

```
DELETE FROM table_name WHERE condition;
```

2. Below is an example of how to use delete command

```
maxwellsql> create table students(name text primary_key, id int unique, gpa int not_null);
Case: CREATE
Command: create table students ( name text primary_key , id int unique , gpa int not_null )
maxwellsql>
insert into students values(yash,1,6);
Case: INSERT
Command: insert into students values ( yash , 1 , 6 )
maxwellsql> insert into students (name,id,gpa) values(sai,2,4);
Case: INSERT
Command: insert into students ( name , id , gpa ) values ( sai , 2 , 4 )
maxwellsql> delete from students where name = yash;
Case: DELETE
Command: delete from students where name = yash
1 rows are deleted!
maxwellsql> select * from students;
Case: SELECT
Command: select * from students
------------------
| name | id | gpa |
------------------
| sai  | 2  | 4   |
------------------
```

DROP TABLE

1.To delete a table use the DROP TABLE command. The syntax is shown below:

```
DROP TABLE table_name;
```

2. Below is an example of how to use delete command

```
maxwellsql> show tables;
Case: SHOW
Command: show tables
-----------------------
| table_name          |
-----------------------
| maxwellbase_tables  |
| maxwellbase_columns |
| students            |
-----------------------
maxwellsql> drop table students;
Case: DROP
Command: drop table students
maxwellsql> show tables;
Case: SHOW
Command: show tables
-----------------------
| table_name          |
-----------------------
| maxwellbase_tables  |
| maxwellbase_columns |
-----------------------
maxwellsql>
```

CREATE INDEX FILE

1.To create an index file on a column for a table use the CREATE INDEX command. The syntax is shown below:

```
CREATE INDEX table_name (column_name);
```

2. Below is an example of how to use Create index command

```
maxwellsql> create index students(name);
Case: CREATE
Stub: parseCreateIndex method
Command: create index students ( name )
maxwellsql>
```

3. This will create an index file in the location : data > user_data > table_name.column_name.ndx

EXIT

Type EXIT; to exit the system.

```
maxwellsql> exit;
Exiting...
```