

Team Magnesium - DavisBase Project

Team Members

- Sowmya Sivaramakrishnan (sxs230043)
- Nivedha Shankar (nxs230138)
- Korey Pickering (kap200004)
- Ishva Patel (irp190001)
- Aiswarya Madhu (axm230384)

Navigating the Prompt

- Upon loading the MagnesiumSQL DBMS you will be greeted with the opening prompt. You may begin by using the `help;` command:

```
magnesiumsql> help;
*****
SUPPORTED COMMANDS

All commands below are case insensitive

SHOW TABLES;
    Display the names of all tables.

CREATE TABLE <table_name> (<column_name> <data_type> <not_null> <unique>);
    Creates a table with the given columns.

DROP TABLE <table_name>;
    Remove table data (i.e. all records) and its schema.

UPDATE TABLE <table_name> SET <column_name> = <value> [WHERE <condition>];
    Modify records data whose optional <condition>
    is <column_name> = <value>.

INSERT INTO <table_name> (<column_list>) VALUES (<values_list>);
    Inserts a new record into the table with the given values for the given columns.

SELECT <column_list> FROM <table_name> [WHERE <condition>];
    Display table records whose optional <condition>
    is <column_name> = <value>.

VERSION;
    Display the program version.

HELP;
    Display this help information.

EXIT;
    Exit the program.
```

```
*****
```

Commands

All commands are listed under the help display. All commands must be terminated by a semicolon: ;.

Show Tables

DDL Commands

Create Table

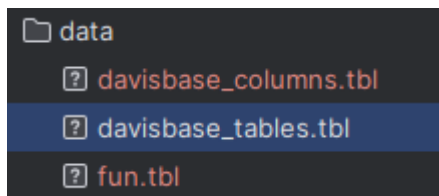
The `CREATE TABLE` command is used to insert a new table and follows the following format syntax:

```
CREATE TABLE table_name ( column_name1 data_type1 [PRIMARY KEY][NOT NULL]
[UNIQUE], column_name2 data_type2 [NOT NULL][UNIQUE], ... );
```

Example:

```
magnesiumsql> CREATE TABLE fun(id int primary key, name text);
* Table created
```

```
magnesiumsql> Show Tables;
table_name          record_count  avg_length  root_page
-----
davisbase_tables    3             0           0
davisbase_columns   13            0           2
fun                  0             0           0
```



Dropping Tables

The `DROP TABLE` command is used for deleting a table and follows the following format syntax:

```
DROP TABLE table_name;
```

Example:

```
magnesiumsql> DROP TABLE fun;
STUB: This is the dropTable method.
      Parsing the string:"drop table fun"

1 record(s) deleted!

2 record(s) deleted!
table deleted
```

```
magnesiumsql> SHOW TABLES;
table_name          record_count    avg_length    root_page
-----
davisbase_tables      2              0             0
davisbase_columns    11             0             2
```

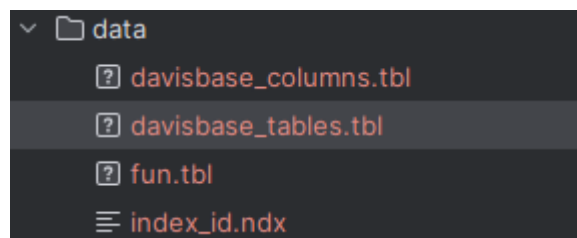
Create Index

The `CREATE INDEX` command is used to insert a new table and uses the following format syntax:

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

Example:

```
magnesiumsql> CREATE INDEX index_id ON fun (id);
* Index "index_id" created on the column: id
```



Dropping an Index

The `DROP INDEX` command is used to insert a new table and uses the following format syntax:

```
DROP INDEX index_name ON table_name;
```

Example:

```
magnesiumsql> DROP INDEX index_id ON fun;  
* Index file index_id deleted.
```

The index file will then be removed from the data folder

DML Commands

Insert Command

The `INSERT INTO` command is used to insert a new value into a single table using the following format syntax:

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

Example:

```
magnesiumsql> INSERT INTO TABLE fun VALUES (3, "uno");  
* Record Inserted
```

```
magnesiumsql> select * from fun;  
id          name  
-----  
1           hopscotch  
2           monopoly  
3           uno
```

Deleting Rows

The `DELETE` command is used to delete a row from a table. The deletion command requires a condition in the where clause. The syntax for deleting a table:

```
DELETE FROM TABLE table_name [WHERE condition];
```

Example:

```
magnesiumsql> DELETE FROM TABLE fun where name = "uno";  
1 record(s) deleted!
```

```
magnesiumsql> select * from fun;
id            name
-----
1            hopscotch
2            monopoly
```

DQL Commands

Selection

The `SELECT` command has three possible clauses, but requires both the `SELECT` and `FROM` clause. Multi-line statements don't terminate until a semicolon is reached. The following is the correct syntax for selection:

```
SELECT * FROM table_name WHERE column_name=value;
```

```
SELECT * FROM table_name WHERE column_name1>value1 AND column_name2>=value2;
```

```
SELECT * FROM table_name WHERE NOT column_name=value;
```

```
SELECT * FROM table_name WHERE column_name1 > value1 OR NOT column_name2 >=
value2;
```

Examples:

Type "help;" to show the list of available commands.

```
magnesiumsql> select * from fun;
```

id	name
----	------

1	hopscotch
---	-----------

2	monopoly
---	----------

```
magnesiumsql> select * from fun where id > 1;
```

id	name
----	------

2	monopoly
---	----------

```
magnesiumsql> select * from fun where name = "hopscotch";
```

id	name
----	------

1	hopscotch
---	-----------

```
magnesiumsql> select *
```

```
from fun
```

```
where id > 1;
```

id	name
----	------

2	monopoly
---	----------

Displaying the rowid requires the first selection option to be rowid followed by a comma using the following syntax:

```
SELECT rowid, * FROM table_name;
```

Examples:

Type "help;" to show the list of available commands.

magnesiumsql> *select * from fun;*

id	name
----	------

1	hopscotch
---	-----------

2	monopoly
---	----------

magnesiumsql> *select rowid, * from fun;*

rowid	id	name
-------	----	------

1	1	hopscotch
---	---	-----------

2	2	monopoly
---	---	----------

magnesiumsql> *select * from fun;*

id	name
----	------

1	hopscotch
---	-----------

2	monopoly
---	----------