API Documentation

OctopusDB

Prepared by:

- Ahmed Atef
- Aya Hamed

APIs

Index

---

createDB.sh

Description

Creates a new database in your system

Example

```
bash createDB.sh dbName
```

Arguments

- **$1** Database Name: Should not contain any special characters or contain only numbers

Exit Codes

- **0** : If successfully created a new database
- **1** : If there is a syntax error in arguments or database name is not valid or database exists

---

dropBD.sh

Description

Removes a given database from the system

Example

```
bash dropDB.sh dbName
```

Arguments

- **$1** Database Name

Exit Codes

- **0** : If successfully dropped the database
- **1** : If there is a syntax error in arguments or database doesn't exists

---

listBD.sh

Description

Lists the databases in the system

Example

```
bash listDB.sh
```

Arguments

- no_args

Exit Codes

- **0** : If successfully listed databases
- **1** : If system doesn't contain any database

---

openBD.sh

Description

Opens a specific database for the user to start managing tables inside it

Example

```
bash openDB.sh dbName
```

Arguments

- **$1** Database Name

Exit Codes

- **0** : If successfully opened the database
- **1** : If database doesn't exist

---

printBD.sh

Description

Prints a file that contains records in a structured table

Example

```
bash printTable.sh tableData.d tableMetadata.md
```

Arguments

- **$1** : The file that contains the records
- **$2** : The file that contains column names

Exit Codes

- **0** : If successfully printed the table
- **1** : If there is a syntax error in arguments or files don't exist

---

insertColumns.sh

Description

Contains test functions and insert function for columns into a table, these functions are used in create table script

Functions

```
function columnDataCheck ()
# @description Checks if the datatypes of columns are valid or not
# @args $1 column names and datatypes in the following syntax: col1Name datatype constrain,col2Name datatype constrain ..etc
# @returncode 0 If valid
# @returncode 1 If not valid

function PKCheck ()
# @description The function checks if primary key data type is repeated
# @arg $1 column names and datatypes in the following syntax: col1Name datatype constrain,col2Name datatype constrain ..etc
# @returncode 0 If not repeated
# @returncode 1 If repeated

function textPKCheck ()
# @description The function checks if primary key is assigned to text data type
# @arg $1 column names and datatypes in the following syntax: col1Name datatype constrain,col2Name datatype constrain ..etc
# @returncode 0 If not assigned
# @returncode 1 If assigned
```

```
function insert ()
# @description The function inserts column names and
datatypes into table metadata file
# @arg $1 column names and datatypes in the following
syntax: col1Name datatype constrain,col2Name datatype
constrain ..etc
```

Arguments

- no_args

Exit Codes

- Not specified

---

createTable.sh

Description

Creates a new table in a database

Example

```
bash createTable.sh tableName
```

Arguments

- **$1** Table Name: Should not contain any special characters or contain only numbers

Exit Codes

- **0** : If successfully created a new table
- **1** : If there is a syntax error in arguments or table name is not valid or table exists or column names are repeated or database is not selected

---

dropTable.sh

Description

Removes a given table from a database

Example

> `bash dropTable.sh tableName`

Arguments

- **$1** Table Name

Exit Codes

- **0** : If successfully dropped the table
- **1** : If there is a syntax error in arguments or table doesn't exist or database is not selected

---

listTables.sh

Description

Lists tables inside a database

Example

> `bash listTables.sh`

Arguments

- no_args

Exit Codes

- **0** : If successfully listed tables
- **1** : If database doesn't contain any tables or database is not selected

---

showTable.sh

Description

Shows all records from a given table

Example

> `bash showTable.sh tableName`

Arguments

- **$1** Table name

Exit Codes

- **0** : If successfully displayed table records
- **1** : If there is a syntax error in arguments or table name doesn't exist or database is not selected

---

test.sh

Description

Contains all test functions for validation

Functions

```
function argsCheck ()
# @description Checks if number of args passed to a
script corresponds to valid args number
# @arg $1 number of arguments passed to the script
# @arg $2 valid number of arguments for the script
# @returncode 0 If valid
# @returncode 1 If not valid

function validName ()
# @description Checks if a given name is valid; doesn't
contain any characters or contains only numbers
# @arg $1 name
# @returncode 0 If valid
# @returncode 1 If not valid
```

```
function dbExist ()
# @description Checks if a database exists or not
# @arg $1 database name
# @returncode 0 If exists
# @returncode 1 If doesn't exist

function dbSetEmpty ()
# @description CThe fuhecks if the databases set in the
system is empty
# @noargs
# @returncode 0 If set is empty
# @returncode 1 If set is not empty

function dbUsed ()
# @description Checks if user is currently using a
database
# @noargs
# @returncode 0 If user is using a database
# @returncode 1 If no database is being used

function intCheck ()
# @description Checks if a value is integer
# @arg1 $1 Value
# @returncode 0 If integer
# @returncode 1 If not
```

Arguments

- no_args

Exit Codes

- Not specified

---

insertRecord.sh

Description

Adds records into table as a full record

Example

```
bash insertRecord.sh tableName VALUE1 VALUE2 ...
```

Arguments

- **$1** Table name
- **$n** Columns values

Exit Codes

- **0** : If successfully inserted the record
- **1** : If there is a syntax error in arguments or no database is currently used or table doesn't exist or primary key is repeated or a data type is invalid for the column or if number of input arguments to add to table greater than or less than number of columns

---

deleteRecord.sh

Description

Deletes records from a table

Example

```
bash deleteRecord.sh tableName record
```

Arguments

- **$1** Table name
- **$2** The record given in the following syntax where you can specify multiple columns: COLUMN=VALUE COLUMN=VALUE

Exit Codes

- **0** : If successfully deleted the record
- **1** : If there is a syntax error in arguments or table doesn't exist or no database is currently used or column name doesn't exist or the record doesn't exist

---

updateRecord.sh

Description

Updates a column in specific records

Example

```
bash updateRecord.sh tableName record columnName
```

Arguments

- **$1** Table name
- **$2** The record given in the following syntax where you can specify multiple columns: COLUMN=VALUE COLUMN=VALUE
- **$last** The updated value given in the following syntax COLUMN=VALUE

Exit Codes

- **0** : If successfully updated the record
- **1** : If there is a syntax error in arguments or table doesn't exist or no database is currently used or column name doesn't exist or the record doesn't exist

---

selectRecord.sh

Description

Displays specific records in a table

Example

```
bash selectRecord.sh tableName record
```

Arguments

- **$1** Table name
- **$2** The record given in the following syntax where you can specify multiple columns: COLUMN=VALUE COLUMN=VALUE

Exit Codes

- **0** : If successfully displayed the record
- **1** : If there is a syntax error in arguments or table doesn't exist or no database is currently used or column name doesn't exist or table doesn't exist or the record doesn't exist

---

help.sh

Description

Prints a User manual for the available options

Arguments

- no_args

Exit Codes

- Not specified