Project Documentation

OurDB DataBase

Zeel sheladiya (Rollno : 97)

Mihir Surati (Rollno : 99)

Parth Dabheliya (Rollno : 13)

Pranav patel (Rollno : 74)

Introduction

Scope Of The Project

Database Management systems are widely used by companies and organizations to maintain and manage their knowledge and information resources. After completing a database management course, students might work in any number of industries including,

1. Automotive
2. Banking
3. Education
4. Legal
5. Insurance
6. Government
7. Pharmaceutical
8. Retail

Overview Of The Existing Systems And Technologies

Types of Database Management Systems

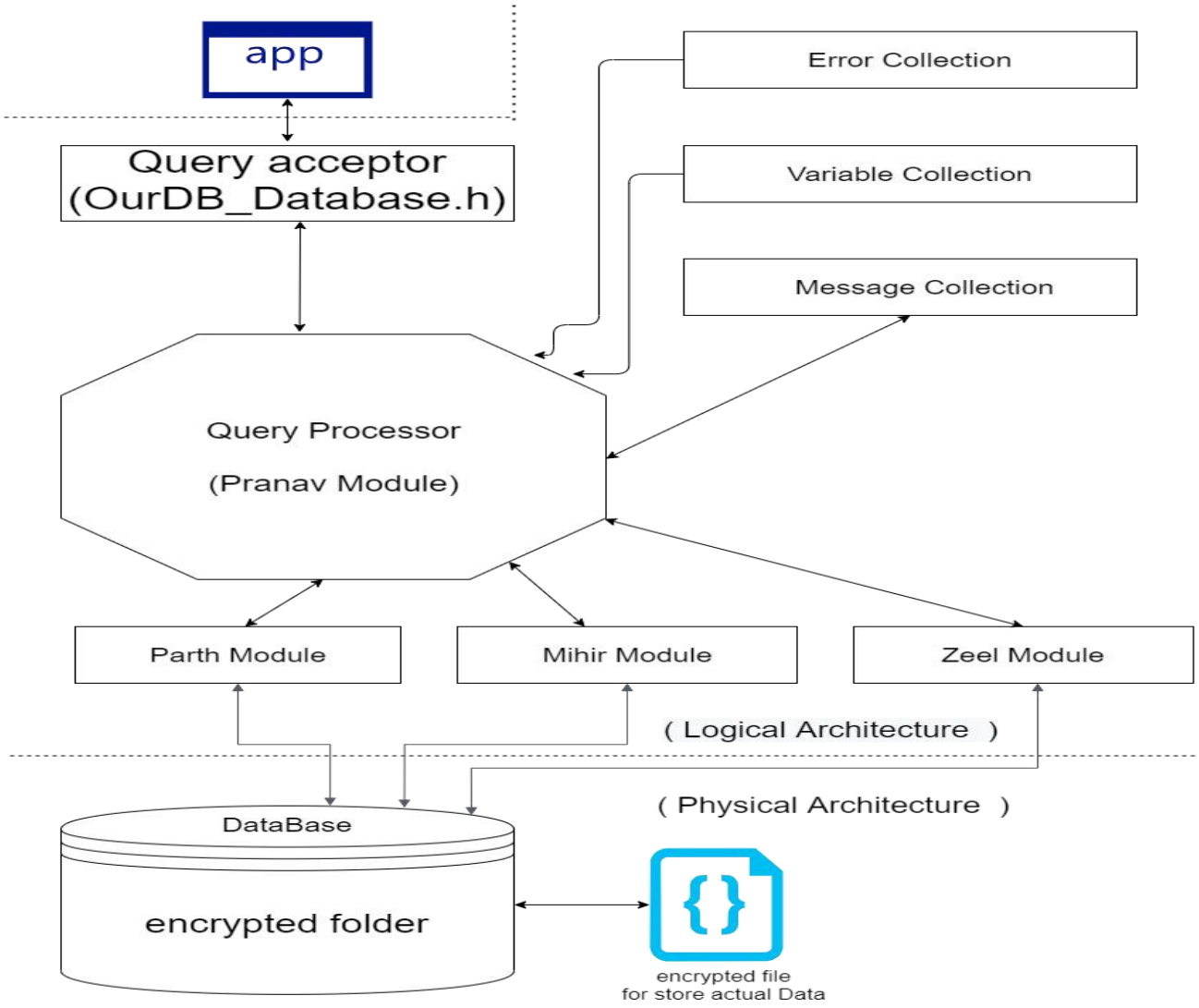
There are several types of database management systems. Here is a list of seven common database management systems:

1. Hierarchical databases
2. Network databases
3. Relational databases
4. Object-oriented databases
5. Graph databases
6. ER model databases
7. Document databases
8. NoSQL databases

List Of The Top Database Management Software

SolarWinds Database Performance Analyzer , DbVisualizer , ManageEngine Applications Manager , Altibase , Oracle RDBMS , IBM DB2 , Microsoft SQL Server , SAP Sybase ASE , Teradata , ADABAS , MySQL , FileMaker , Microsoft Access , Informix , SQLite , PostgresSQL , AmazonRDS , MongoDB , Redis , CouchDB , Neo4j , OrientDB , Couchbase , phpMyAdmin , SQL Developer , MariaDB and many more.

Overview Of The Project



Architecture Of OurDB

Process Flow Of OurDB

In this Database management system , after connecting the database with your application. Whenever you use **run_query() ,** through this function your query will be gone for process via OurDB_Database.h to query_process.h . In the **query processor** your query gets some operation on it and through these operations, it will be selected by **three main modules**. After the module operation if there is data for insertion or updation, it will write in encrypted form into the **physical storage area**. However , if there is data for viewing purpose then the module will perform operation on actual data and then present you in the form of string.

Objectives Of The Project

- **You don't need to install this database to use. It's make this database super portable.**
- **You can make your own query syntax very easily . it's make your query code super protected.**
- **It has all data files in encrypted format.**
- **You can connect this database with your application with just one line of code.**
- **This database works in both application console and GUI.**
- **Database import and export is super easy.**
- **You can make a shared database without any connection string or any extra connection.**
- **You can add your own module in the database for the query process.**
- **It's very lite in terms of size.**
- **To run this database you don't need any extra specification requirement in your system. So it can work smoothly in low specs machines.**

Feasibility Study

1. Financial Feasibility :

It is too light weight and it doesn't need any high end requirements to run. So it can be too cheap or very affordable for all kinds of developers.

2. Technical feasibility :

OurDB is a completely data operation based application. The main technologies and tools that are associated with OurDB are ,

- C++ Compiler

This technology is freely available and technical skills required are manageable. From These it’s clear that the project OurDB is technically feasible.

3. Resource and Time Feasibility :

Resource that are required for the OurDB project includes,

1. C++ compiler
2. Space in physical disk to record the data
3. C and C++ supported environment

Consideration

1. Performance
2. Security
3. Usability and ease of use
4. Capacity and Scalability
5. Availability
6. Maintainability

GETTING STARTED

This will tell you about the OurDB database. And how to use it.Its so simple to use this database system you don't need to install whole database system like any other. But just add some database files not more than(under 1.5 MB) and need to install c++17 compiler(minimum requirement).

- We need to follow some step to get started with database..

    1. You have to include Ourdb_Database.h header file to your application so that you can use database.

    2. Then you have to create a database.

    3. After creating database, you can create table and do all table related operation.

    4. Then you can insert into the the table.

    5. Even you can perform update and delete operation with & and | operators.

- Additional functionality:

    a) We have added functionality for the developer to convert data to json object.

    b) Convert the data to stringTable(represent data in table format in console application)

    c) Convert the data to map

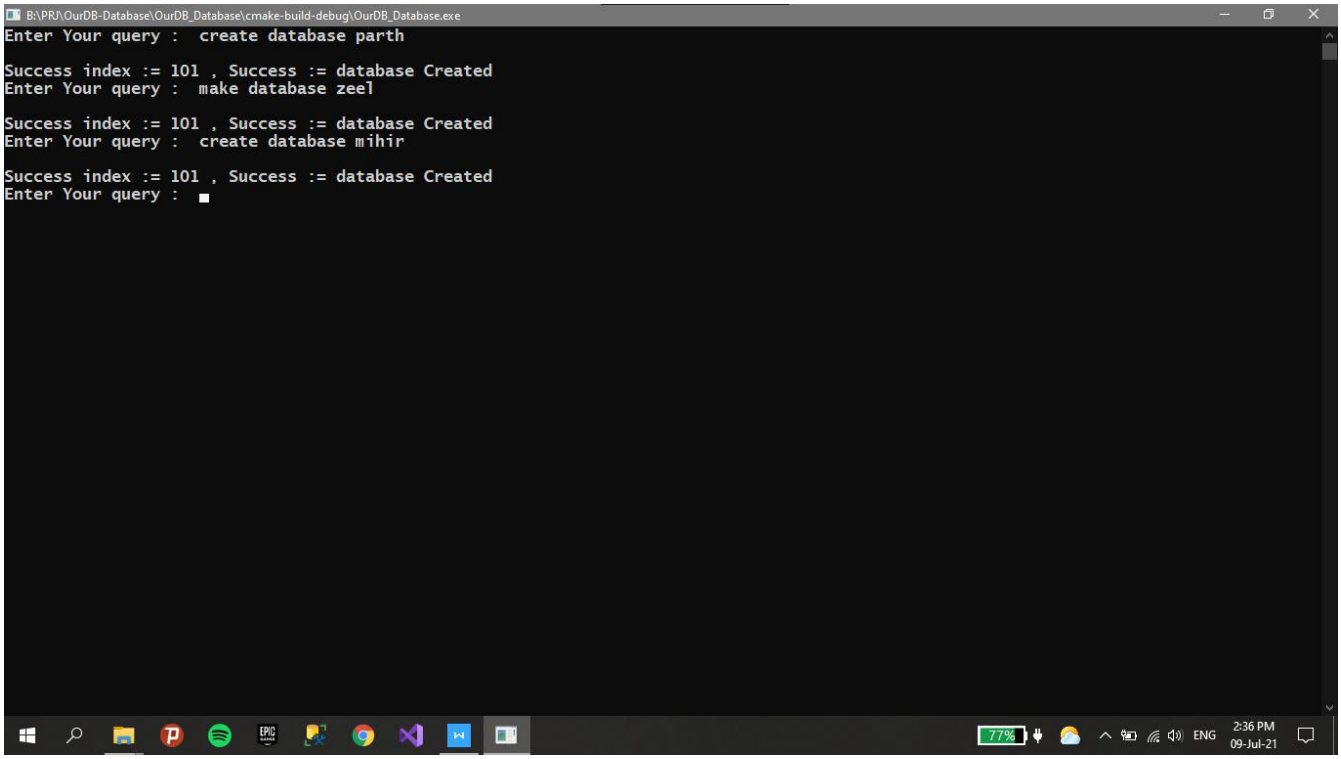
- Developer can easily change the filter to convert to the above three format, by default it is string.

Syntax and uses

- **Create database:**

  First step to enter in the database system is to create database. A physical storage area where all table resides.

Syntax : **create/make database <database_name>**



Above example shows that it is very simple to create a database..

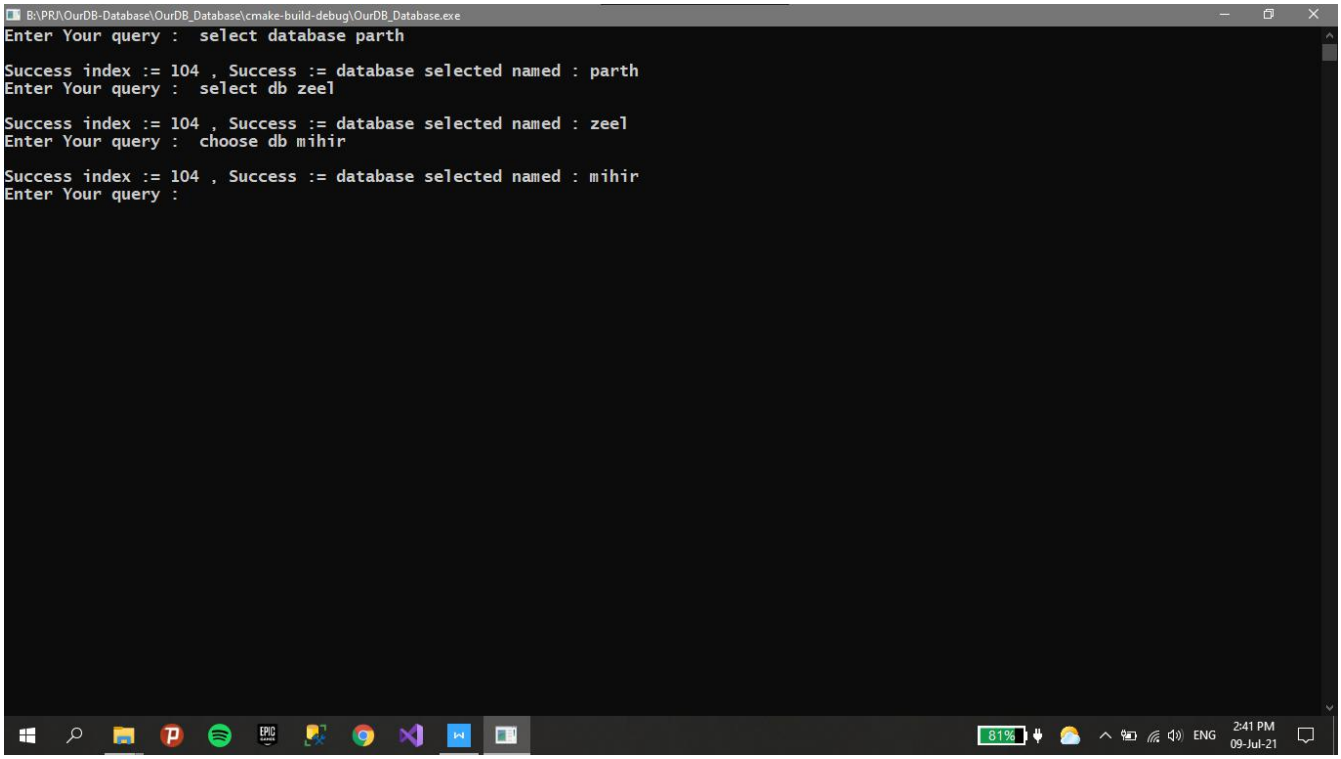
Even you can change the (create) keyword in the code(instruction for developer). its dynamic add or remove keyword etc.

- **Select database :**

select the database among the existing databases else returns the error.

Syntax : select/choose database/db <database_name>

After creating database you have to compulsory select the database in order to create table and perform some action/operations.



Note : database is so flexible that developer can change any keyword..

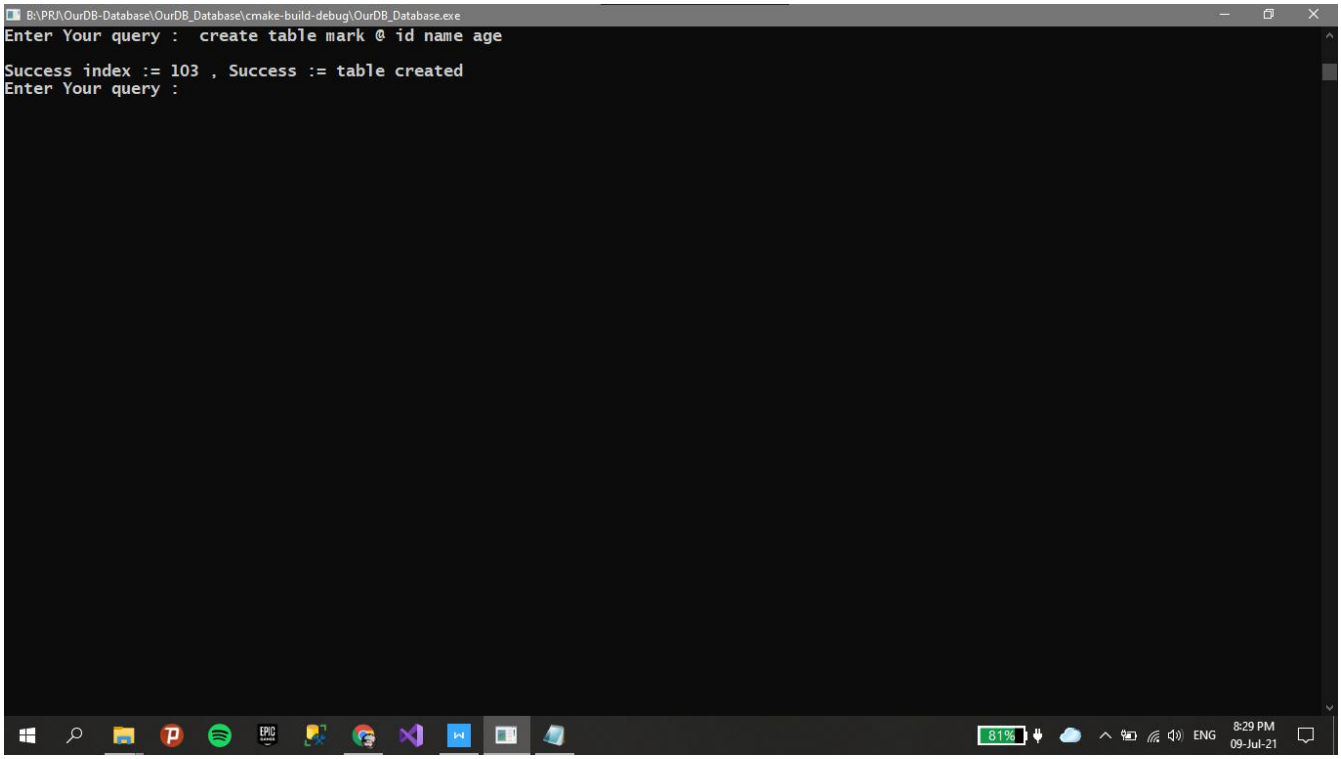
- **Create table :**

  Creates table in the database along with the column in it.

Keyword : create or make

Syntax : **create/make table <table_name> @ <column1_name> <column2_name>...**

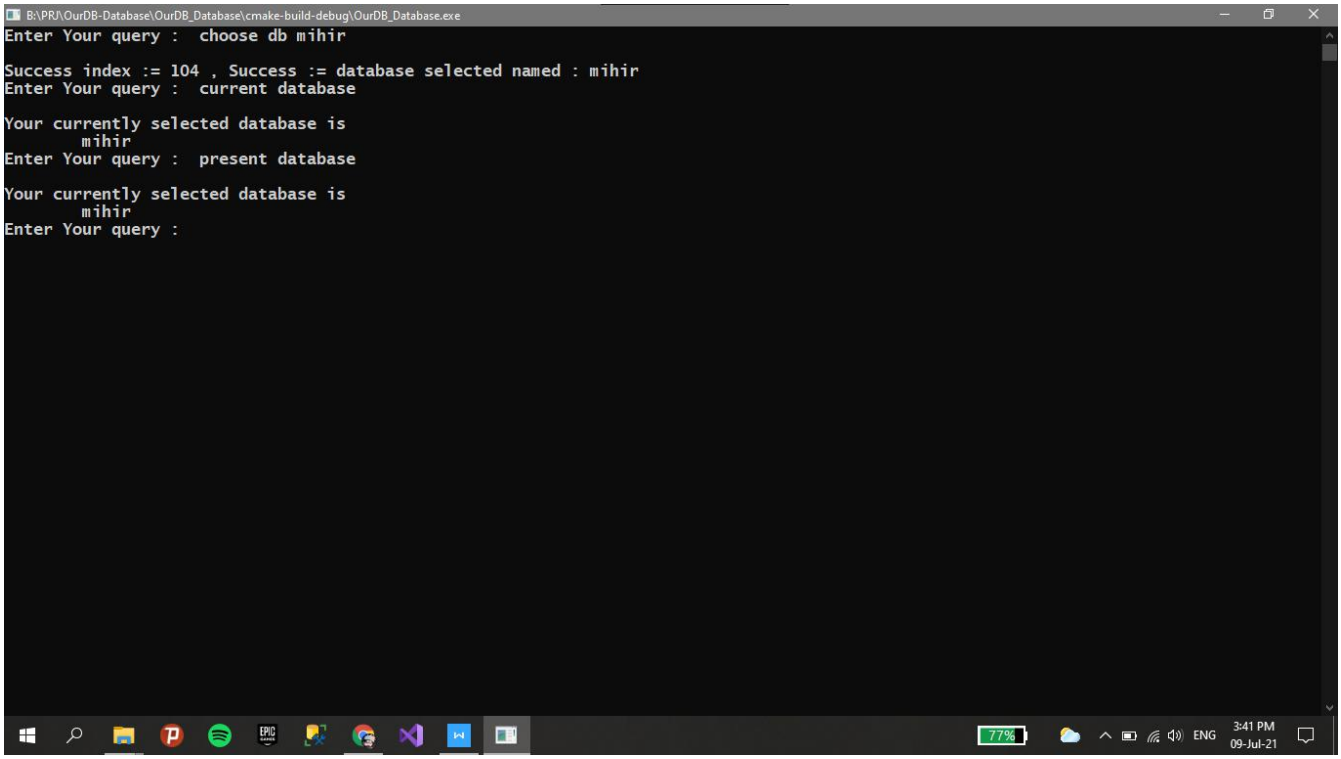
NOTE : After creating database you have to compulsory select the database in order to create table and perform some action/operations.



NOTE : with this simple syntax we can easily create the table in database.

- **Current database :**
  it will show the selected database/the database you are in right now.

  Syntax : current/present database/db



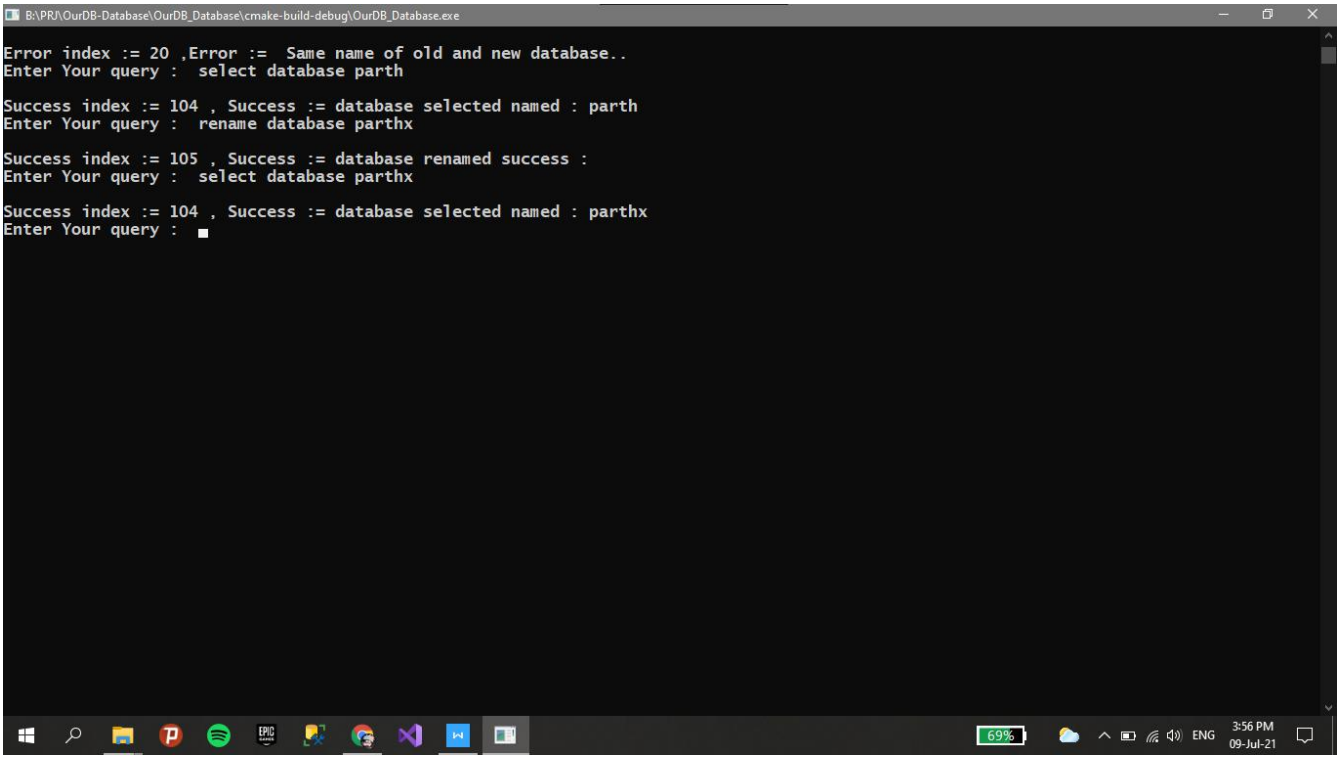
Note: this query used when you don’t know what database you are in right now.

- **Rename database :**

  It will rename the database in the system..

NOTE : first you have to select the database then and then you can perform rename query.and after renaming you have to again select the database to perform the operation.

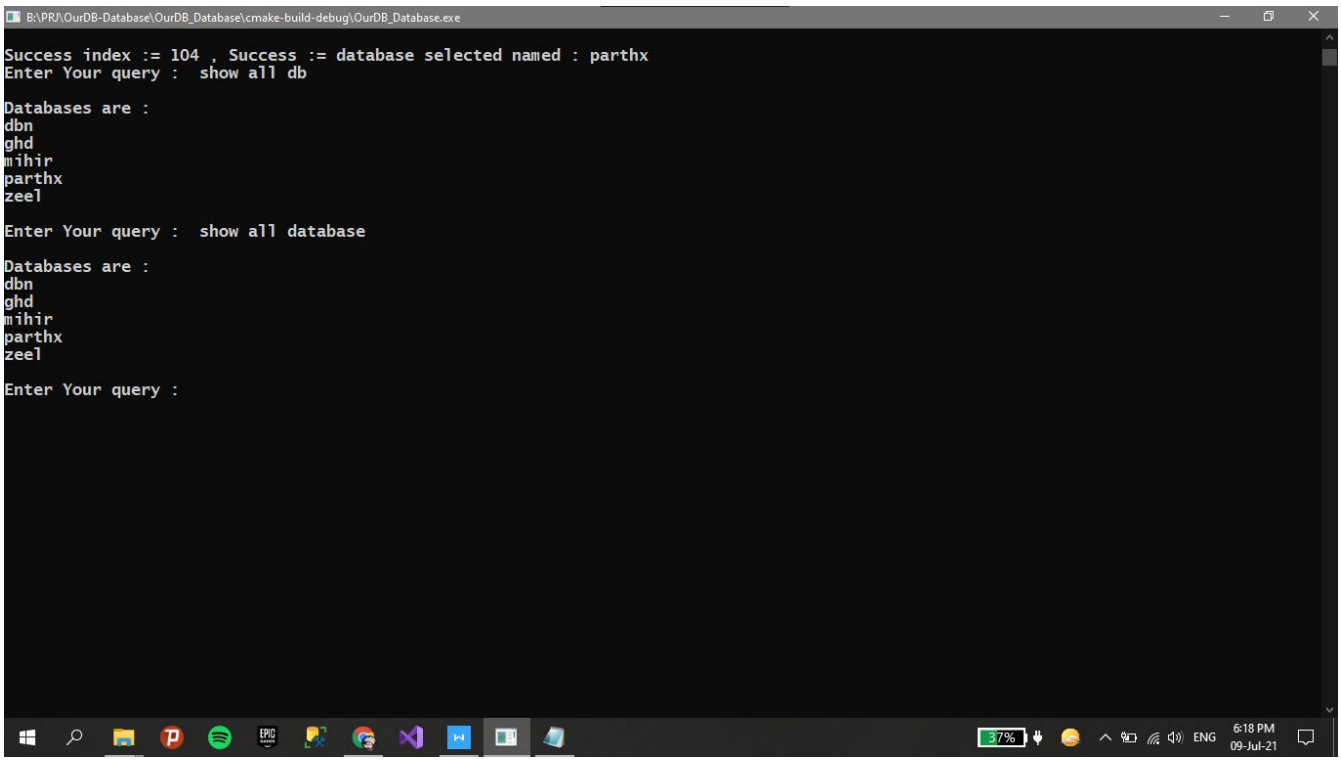
Syntax : rename database <database_name>



Note: we have given this facility who wants to change the name of the database but it is not good practice to do so..

- **Show databases :**
  It will show all the database resides in the system.

Syntax : show/display/view all db/database

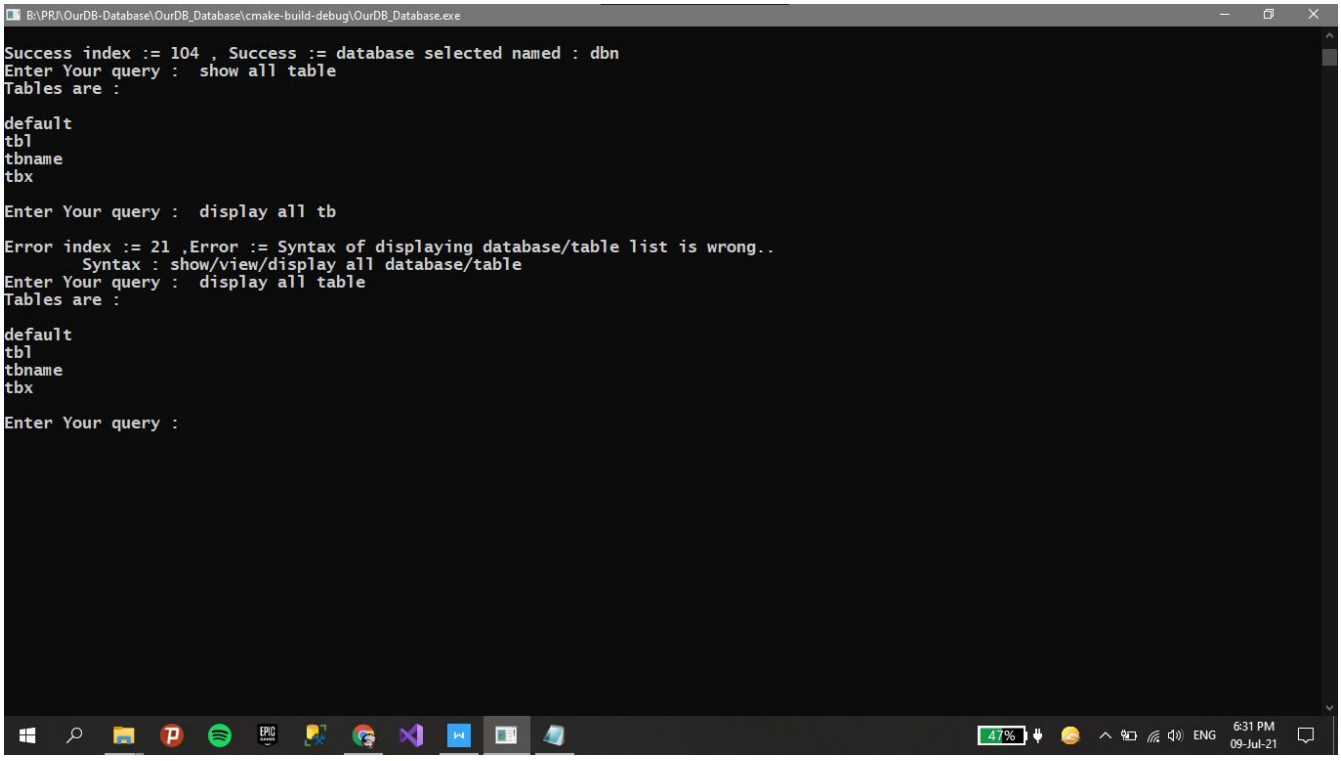


NOTE : it is advisable to run query prior to any other query so that you can see all the database exists in the system

- **Show tables :**
  It will show all the tables resides in the selected database.

Note : first we have to select the database so that we can see the table resides in it.

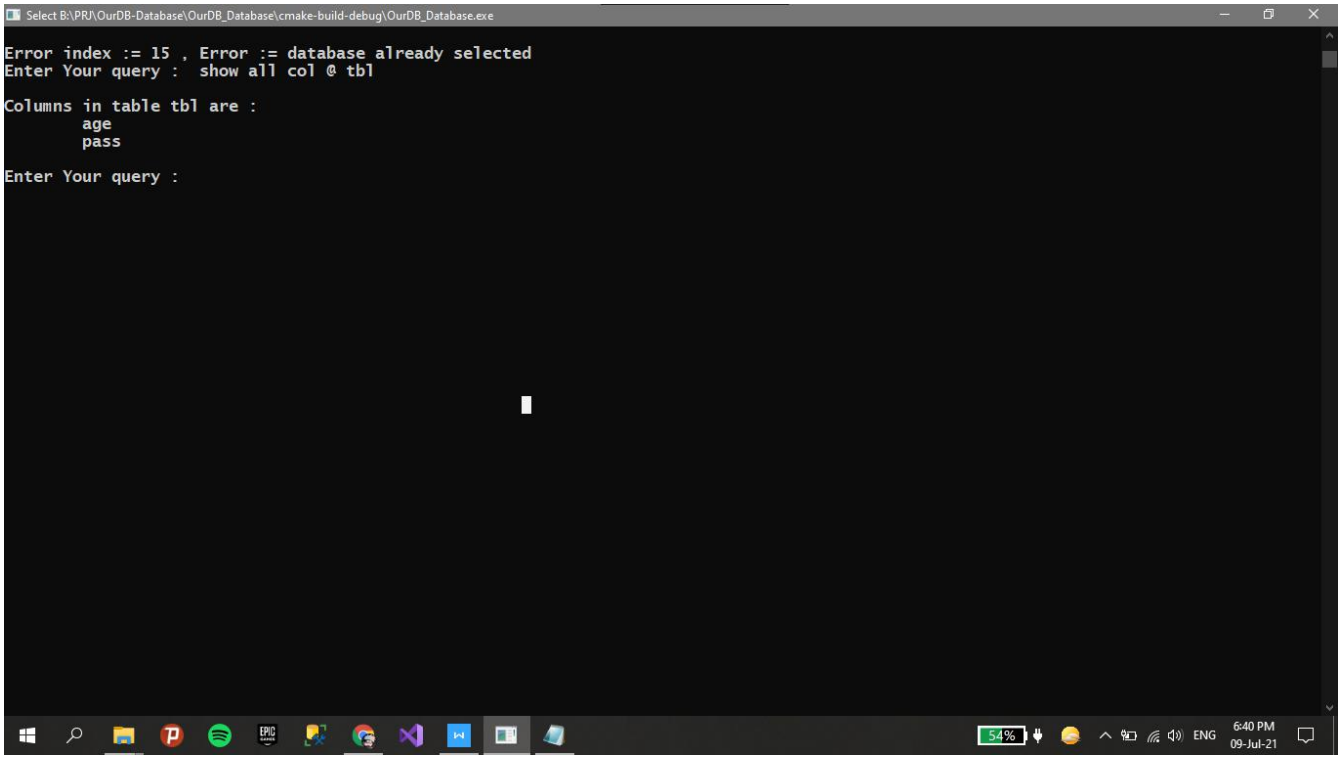
Syntax : show/display/view all table



- **Show column in table :**
  It will list out all the column which is present in the table

Note : After selecting database , create table and then you will be able to use this feature.

Syntax : **show all col @ <table_name>**

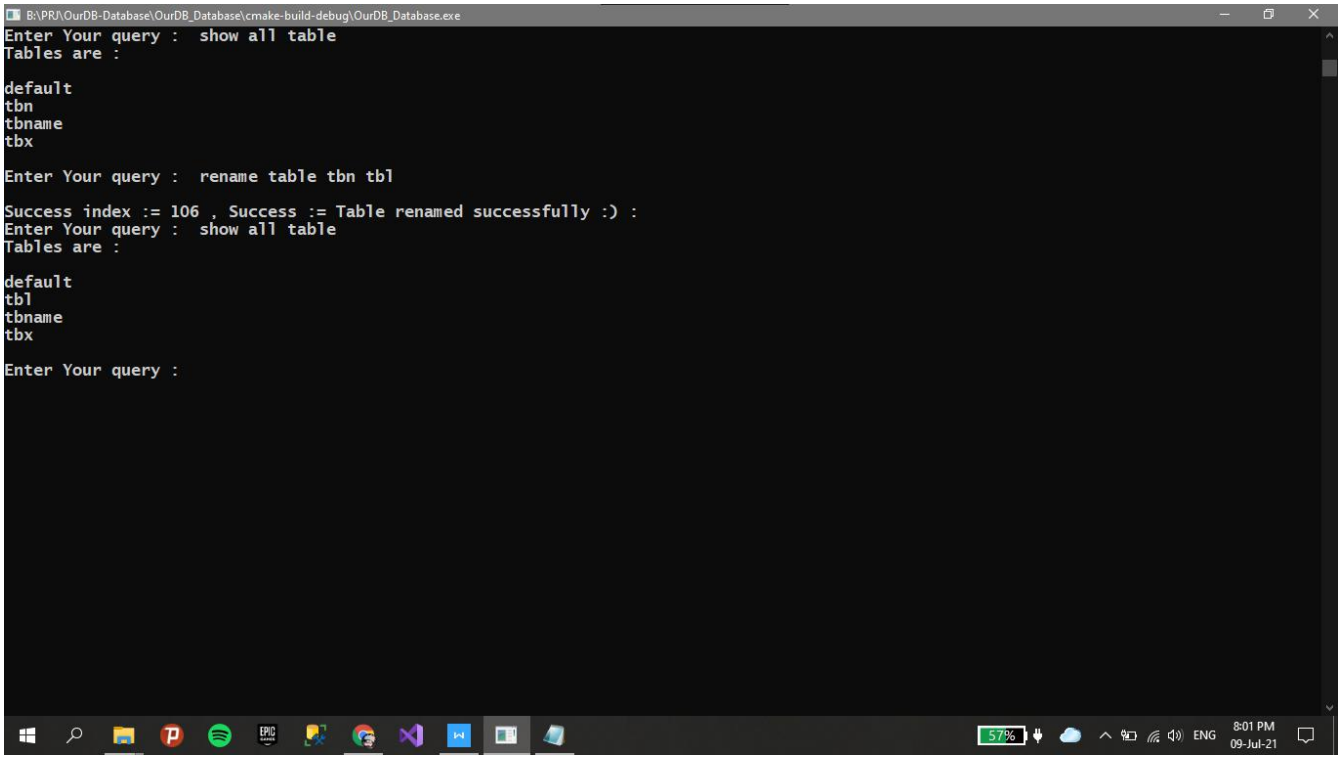


Note: show column in table never return null value because column in the table decide at the creation of table.

- **Rename table in database :**
  This command will rename the table exist in the particular database.

Note : make sure that table must exist before renaming the table.

Syntax : **rename table <old_table_name> <new_table_name>**

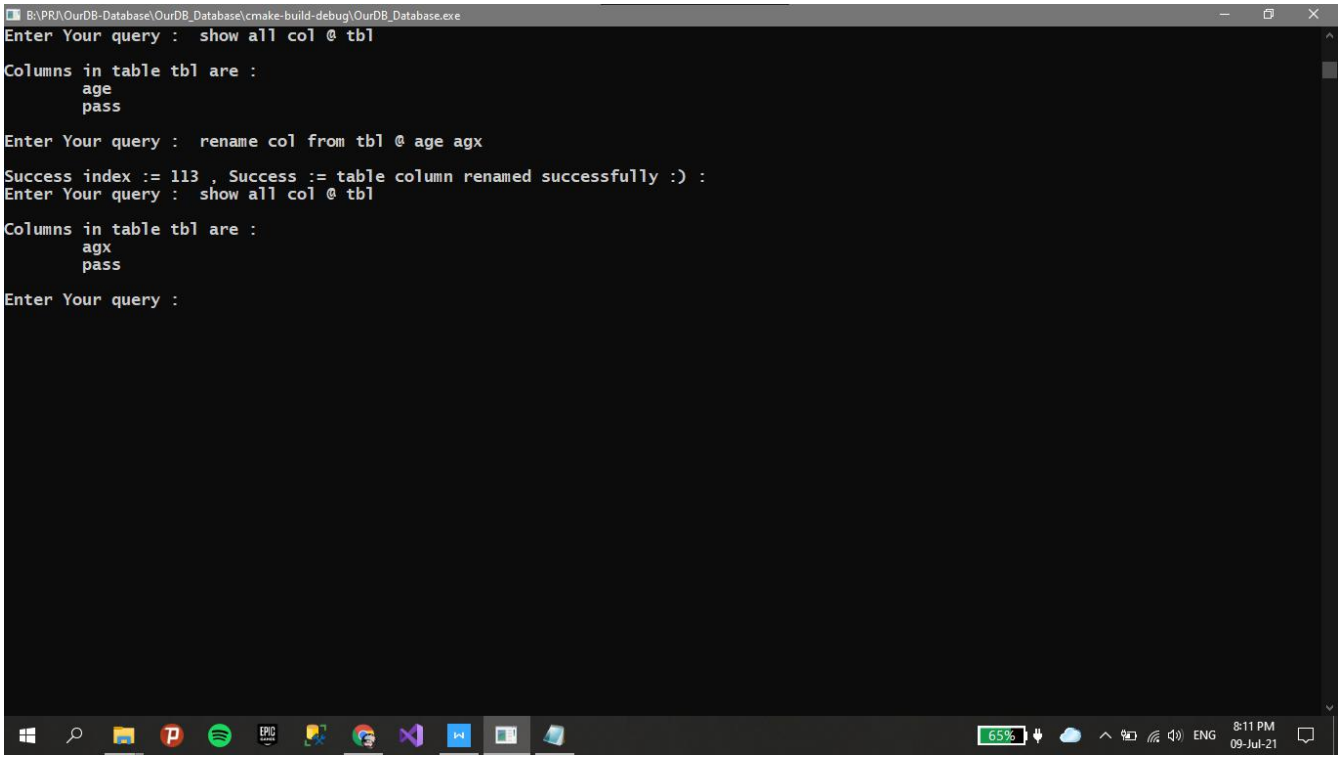


- **Rename column in table :**
  This command will rename the column in the particular table

Note : make sure to check that column exist in the table by show all column @ table.

Keyword : **column/col** both will be accepted but not simultaneously.

Syntax :rename column/col from <table_name> @ <old_col_name> <new_col_name>



Note : this feature looks so simple but most important while working with the database .

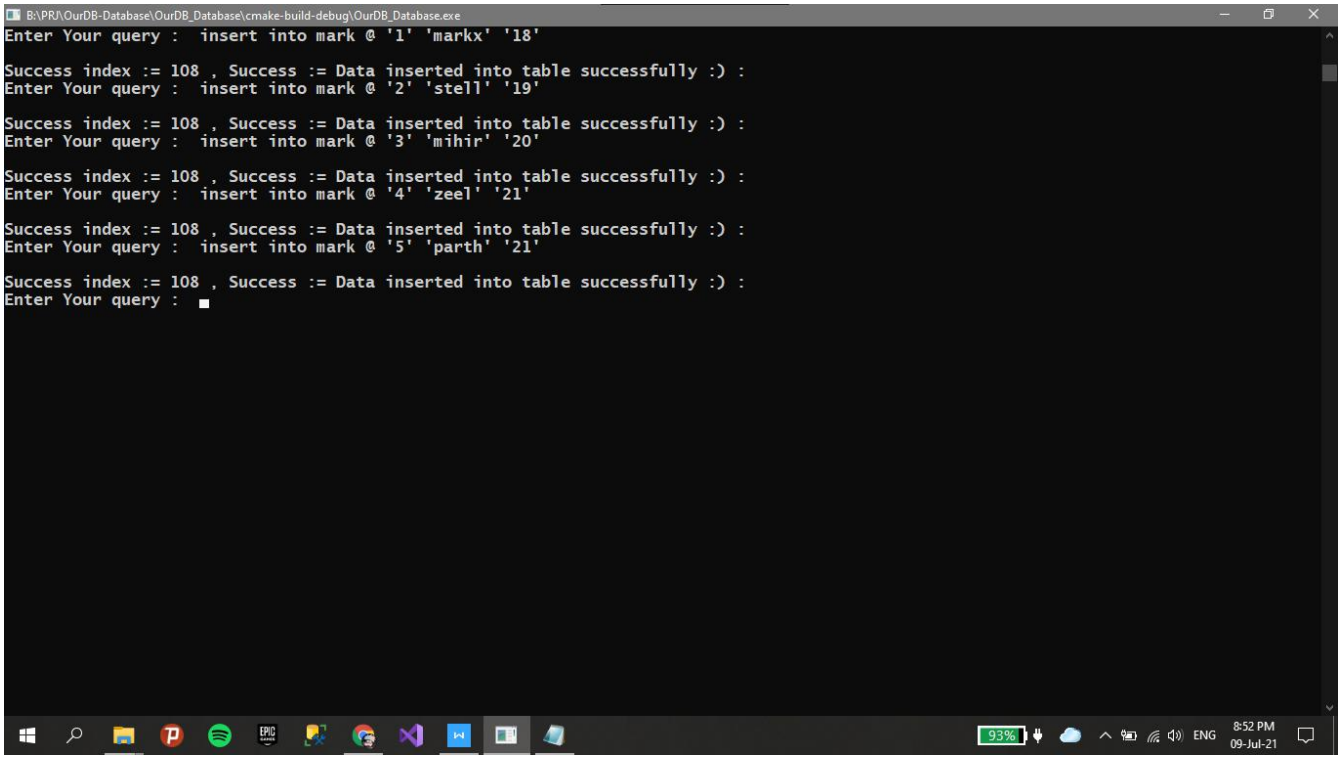
- **Insert data in to the table :**

Insert query is used to enter the data in the empty column which was created with the table during creation of table.

Note : make sure in which table you are going to insert the data ,that must be created before insertion of data.

Keyword :**insert or add** both will be accepted but not simultaneously.

Syntax : insert into <table_name> @ '<col1_value>' '<col2_value>' ...



Note : you can see the data in the table by select command we will look for it later on.

Single quote is necessary for inserting the value in to the table.

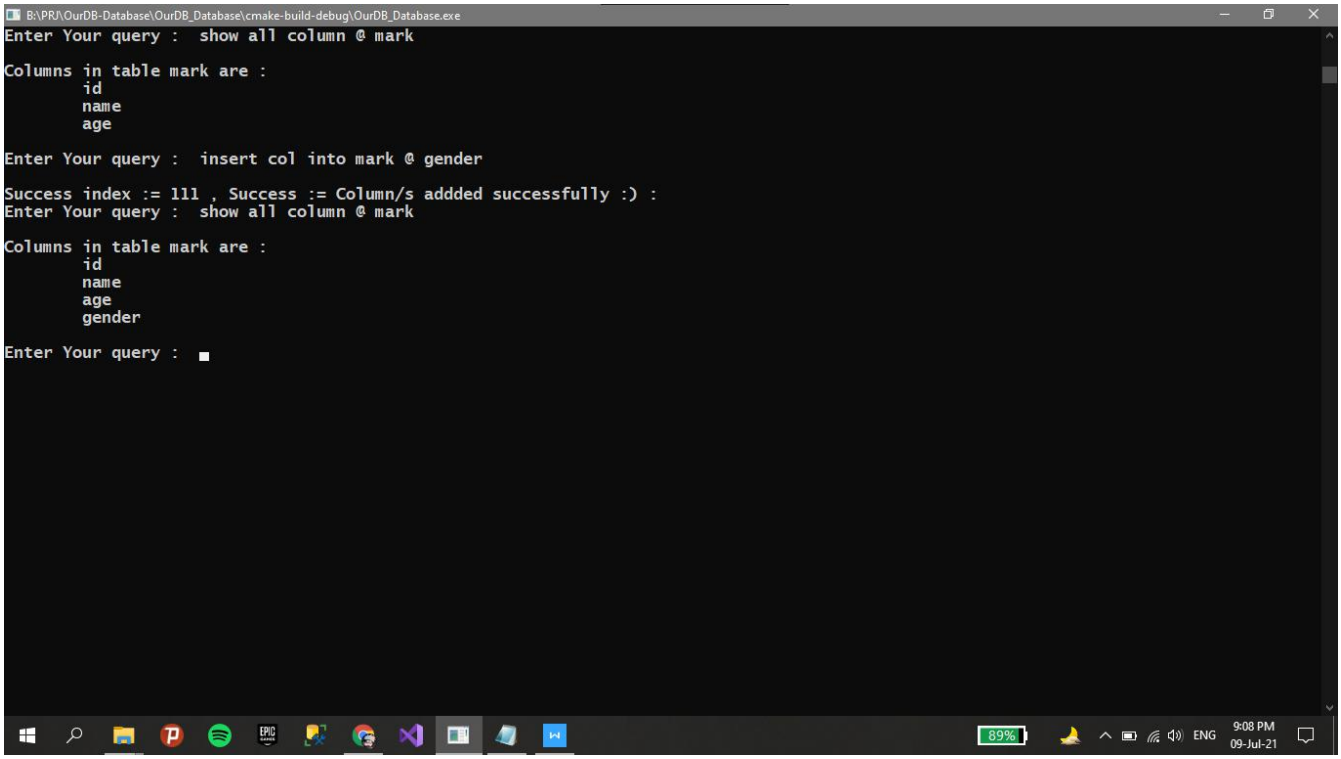
- **Adding new column in the table :**

This command will add a new column in the **existing** table.

Note : make sure in which table you are going to insert the column ,that must be created before insertion of column.

Keyword :**insert or add** both will be accepted but not simultaneously.

Syntax : insert/add col/column into <table_name> @ <col_name>



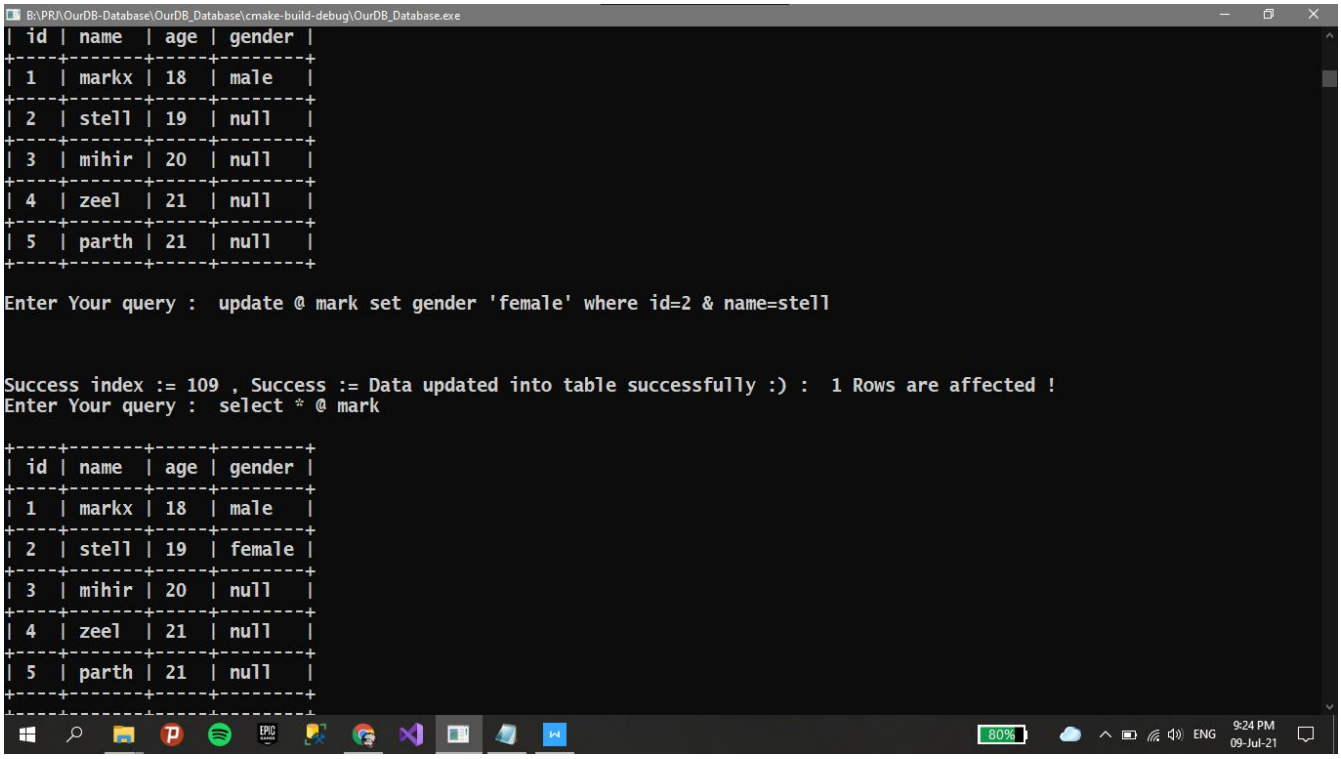
- **Updating a data in to the table :**
  It used to update a data in to the table ..

Here at particular row/rows you can update a data.

Here after where table field and value connected with the = operator.
Or !(not equal) operator.

And for more than one condition you can concatenation condition with either &(and) or |(or)

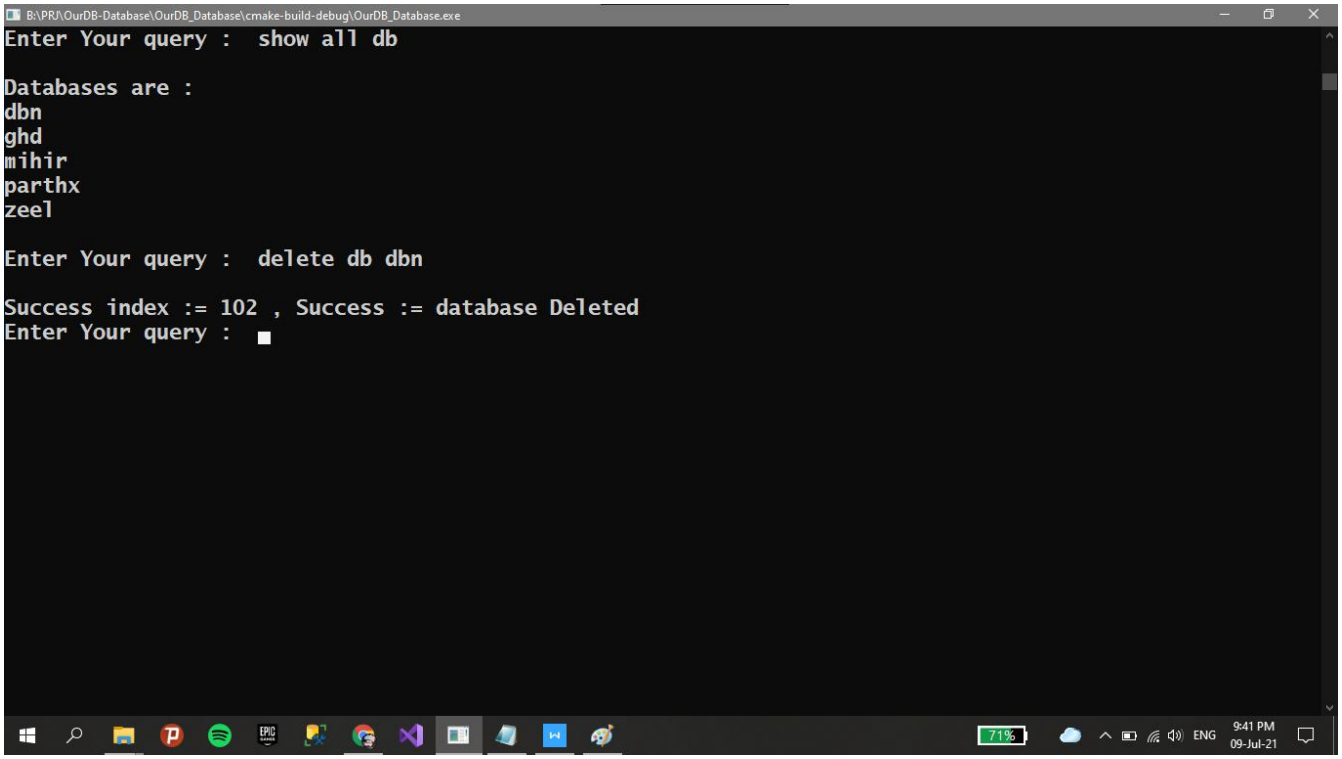
Syntax : update @ <table_name> set/put <table_field> '<field_value>' where <table_field1>(=|!)<field_value1> & <table_field2>(=|!)<field_value2>



- **Delete the database :**
  It will delete the database from the system.

Keyword : delete or destroy both are accepted but not simultaneously

Syntax : delete/destroy database <database_name> // parth module

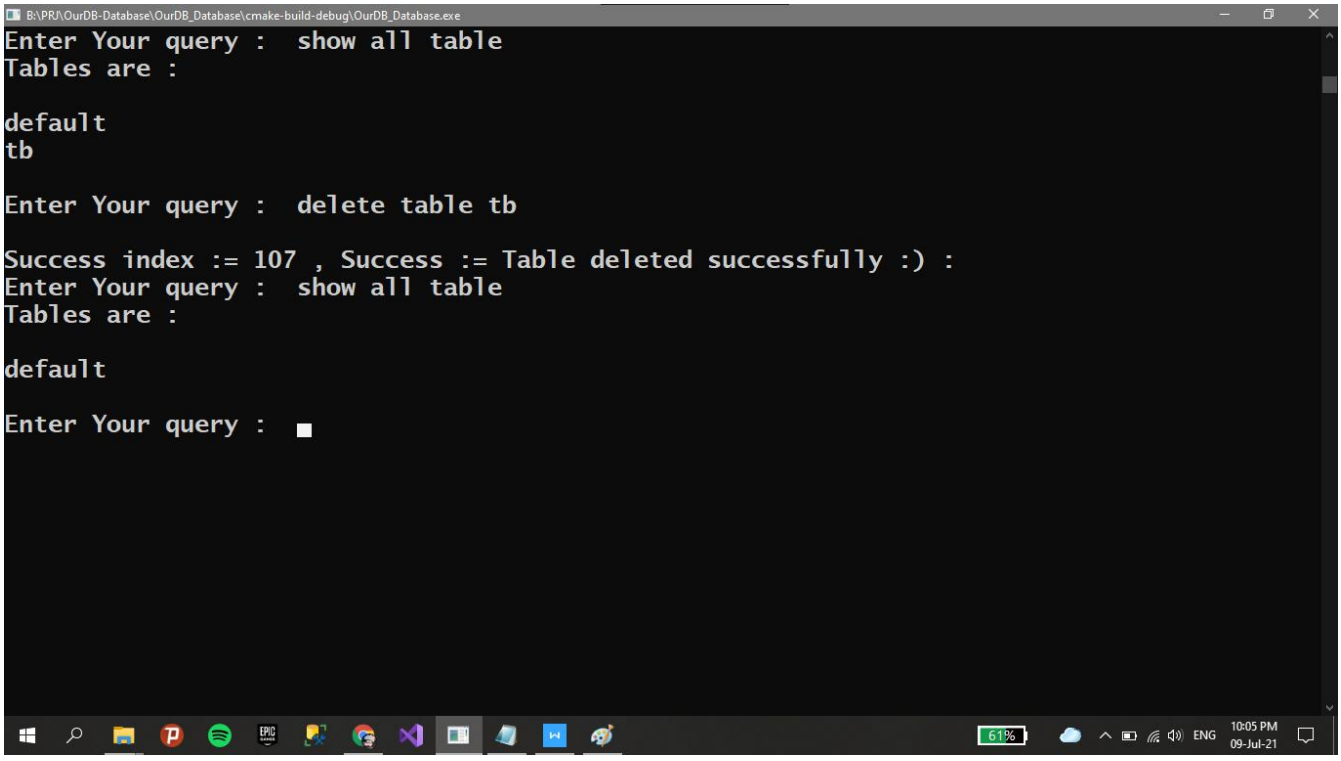


NOTE : Before deleting the database always make sure that your all the tables are deleted or you may loss important data.

- **Delete the table :**
  Delete the table from the database.

Note: make sure to select the database before deleting table.

Syntax : delete table <table_name>



- **Delete the particular table data :**
  Delete the data from the table at particular row/rows .

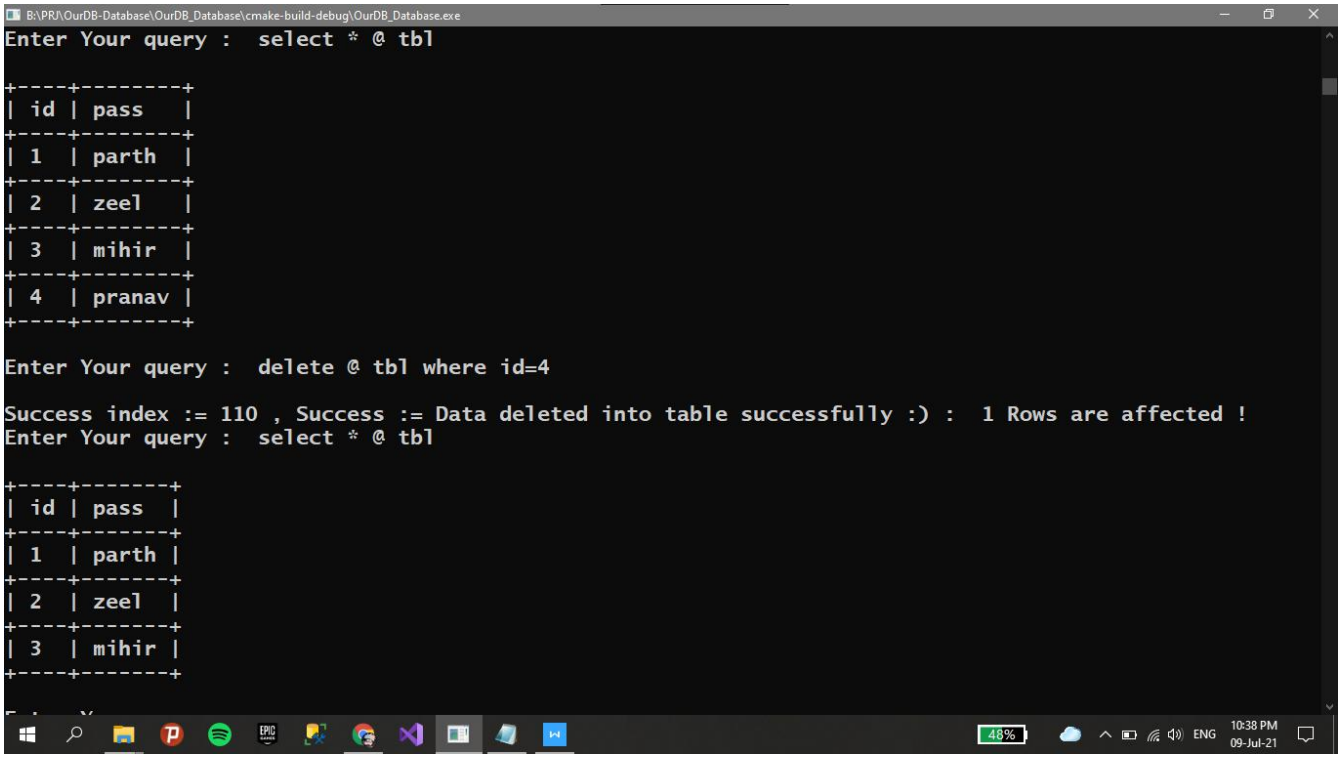
Note: make sure to select the database before deleting table.

Here at particular row/rows you can delete a data.

Here after where table field and value connected with the = operator.
Or !(not equal) operator.

And for more than one condition you can concatenation condition with either &(and) or |(or)

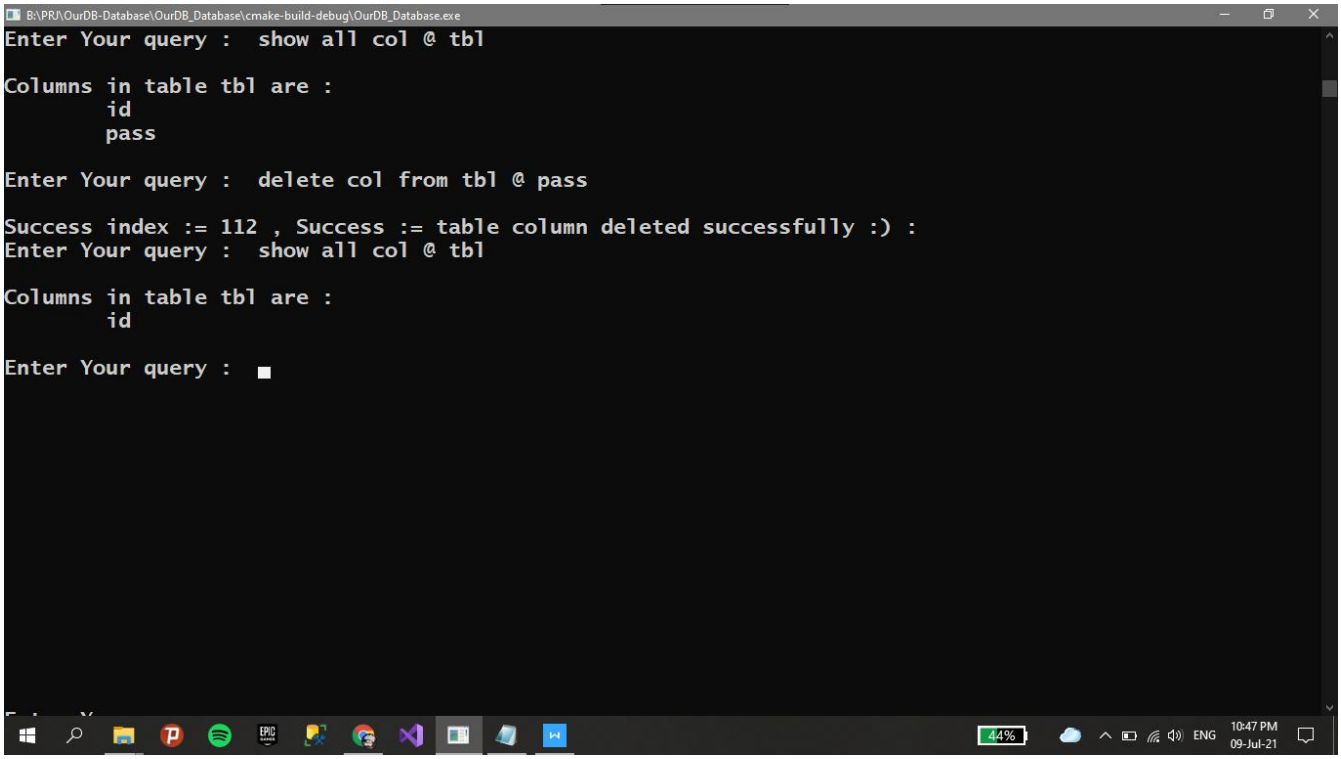
Syntax : delete/destroy @ <table_name> where <column_name> (=/!) <column_value> (&/|) <column_name1> (=/!) <column_value1>



- **Delete the column :**
  This will delete the column from the table .

Note : make sure in which table you are going to insert the column ,that must be created before insertion of column.

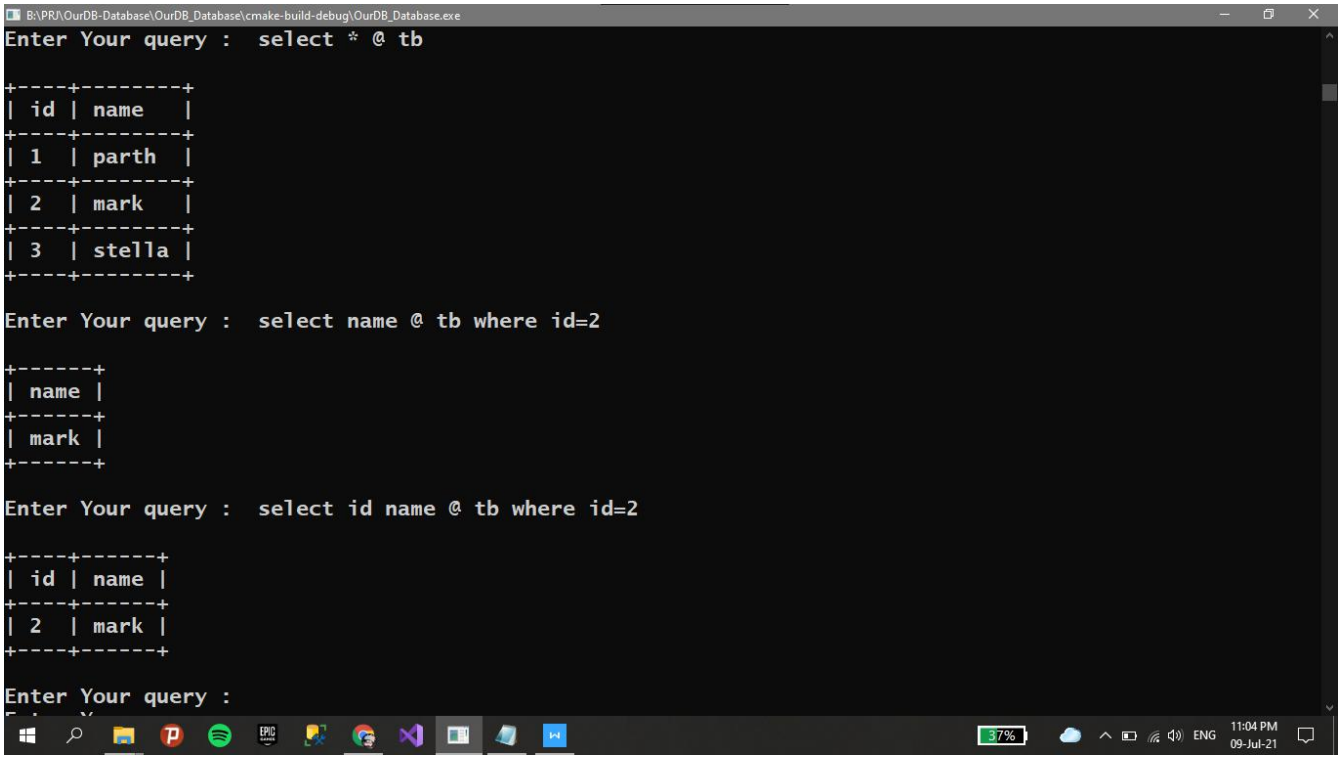
Syntax : delete/destroy column/col from <table_name> @ <col_name>



- **Select data from table:**
  It will select and display the the data into the table format.

Syntax : select/choose * @ <table_name>

Syntax : select/choose col1 col2 @ <table_name> where <column_name> (=/!) <column_value> (&/|) <column_name1> (=/!) <column_value1>



Encryption And Decryption:

The data stored in the database should be secure and can't be read by the user or any other ,While they are stored in the database.

So that we created our own encryption and decryption method to encrypt and decrypt data respectively.

The encryption and decryption use in this we gave name as **69Cryption**.

This **69Cryption** method encrypt the data before stored in the json format And decrypt the data after user ask to retrieve it. So that user can see the encrypted data.

How actually data stored in the table :

The Encrypted data are stored in to the <file_name>.Ourdb (here Ourdb extension is especially for our database system).

While user run query "create table" , the <file_name>.Ourdb is stored in specified database. Which was previously selected by the user.

While user uses "insert into" query, the data stored in the file as Json format.

Future Enhancement :

We have code in our github directory.we will put directory as public so every one will be able to take advantage of database and also then can add more feature . we have created in such a way that , changes can be easily made.

So that we want to create a big community which can enhance the code.