**QUERY BUILDER**



A Major Project Report

in partial fulfillment of the degree

**Bachelor of Technology**

in

**Computer Science & Engineering**

**By**

|  |  |
| --- | --- |
| 16K41A0519 | G.Sai Preetham |
| 16K41A0531 | K.Sathwik |
| 16K41A05C4 | A.Prem Sagar |
| 17K45A0507 | M.Mahesh |

**Under the Guidance of**

**Mr. K. Sudheer Kumar**

Assistant Professor

**Submitted to**



**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING S.R.ENGINEERING COLLEGE(A),ANANTHASAGAR, WARANGAL**

**(Affiliated to JNTUH, Accredited by NBA)**

**2019-2020**



**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**CERTIFICATE**

This is to certify that the Majo Project Report entitled “**Query Builder**” is a record of bonafide work carried out by the student(s) G.Sai Preetham, K.Sathwik, A.Prem Sagar, M.Mahesh bearing Roll No(s**)** 16K41A0519, 16k41A0531, 16K41A05C4, 17K45A0507 during the academic year 2019-20 in partial fulfillment of the award of the degree of ***Bachelor of Technology*** in **Computer Science & Engineering** by the Jawaharlal Nehru Technological University, Hyderabad.

**Supervisor** **Head of the Department**

**External Examiner**

**ACKNOWLEDGEMENT**

We express thanks and gratitude to almighty God, parents and other family members and friends without whose unconstrained support; we could not have made this career in Technology.

We express our sincere gratitude to our Management and Principal **D**r.**V. MAHESH, S R Engineering College,** for making available facilities for carrying out our project.

We would like to express our heartfelt gratitude in deep humility to **A. SRINIVAS,** Assistant Professor and Head, Department of Computer Science and Engineering, for his wholehearted support, who has provided all the facilities to conduct our project work with immense cooperation and inspiration.

We would like to express our gratitude and appreciation to our project guide **K. Sudheer Kumar,** Assistant Professor in Computer Science and Engineering for his excellent advice and encouragement. Sir stood as pillar of strength, right through our project, till the preparation of report and helped, by boosting our morale. So we could surmount the difficulties we faced in our project.

Our special thanks to each and every faculty member and programmers in the department for their suggestions and taking care of all software needs.

We would like to express our gratitude towards our parents for their kind cooperation and encouragement which helped us in the completion of this project.

Last but not least, we express our thanks to our colleagues, who gave us their assistance, suggestions and support.

**G.Sai Preetham (16K41A0519)**

**K.Sathwik (16K41A0539)**

**A.Prem Sagar (16K41A05C4)**

**M.Mahesh (17K45A0507)**

**Abstract**

Query Builder is a Graphical User Interface tool with intuitive interface. Query Builder is a fast, easy to use and compact graphical tool for managing your MySQL databases. Query Builder was developed for all who use MySQL. This Query Builder is a tool that allows end-users to create MySQL queries using Select boxes. Whether you enjoy the control of handwritten Query or prefer to work in a visual environment, Query Builder makes it easy for you to get started and provides you with tools to enhance your MySQL experience. The queries are generated dynamically using the query builder that was developed in this project. The query builder takes input parameters, interpret them and then generate a query. It's not a general-purpose tool. The Query Builder assists with the creation of MySQL DML commands.

* The motto of our project yields good results and helps to reduce syntax errors.
* To make it understandable, we are creating a site using HTML, CSS and creating a GUI.

|  |  |  |
| --- | --- | --- |
|  | **TABLE OF CONTENTS** |  |
| **S.NO** | **TOPIC** | **PAGE NO** |
| **1.** | **INTRODUCTION** | **2** |
| **1.1** | EXISTING SYSTEM | 6 |
| **1.2** | PROPOSED SYSTEM | 7 |
| **2.** | **LITERATURE SURVEY** | **8** |
| **2.1** | RELATED WORK | 9 |
| **2.2** | SYSTEM STUDY | 11 |
| **3.** | **DESIGN** | **7** |
| **3.1** | REQUIREMENTS SPECIFICATION | 8 |
| **3.2** | UML DIAGRAMS | 9 |
| **4.** | **IMPLEMENTATION** | **12** |
| **4.1** | MODULES | 13 |
| **4.2** | OVERVIEW TECHNOLOGY | 13 |
| **4.3** | SAMPLE CODE | 21 |
| **5.** | **TESTING** | **39** |
| **5.1** | TEST CASES | 40 |
| **5.2** | TEST RESULTS | 42 |
| **6.** | **RESULTS** | **46** |
| **7.** | **CONCLUSION** | **56** |
| **8.** | **FUTURESCOPE** | **58** |
| **9.** | **BIBLIOGRAPHY** | **60** |
|  |  |  |

|  |  |  |
| --- | --- | --- |
|  | **LIST OF FIGURES** |  |
| **S.NO** | **EXPLANATION** | **PAGE NO** |
| **3.** | **DESIGN** | **7** |
| **3.2.1** | Use Case Diagram | 9 |
| **3.2.2** | Class Diagram | 10 |
| **3.2.3** | Activity Diagram | 11 |
| **4.** | **IMPLEMENTATION** | **12** |
| **4.2.1** | Dynamic web page | 16 |
| **5.** | **TESTING** | **38** |
| **5.2.1** | Select Database | 42 |
| **5.2.2** | Select Table | 42 |
| **5.2.3** | Selecting Database and Table | 43 |
| **5.2.4** | Incomplete Joins | 43 |
| **5.2.5** | Incomplete Custom Input | 44 |
| **5.2.6** | No Logical Operators | 44 |
| **5.2.7** | Result | 45 |
| **6.** | **RESULTS** | **46** |
| **6.1** | Selecting Database Table | 47 |
| **6.2** | Results From Table | 47 |
| **6.3** | Selecting Join | 48 |
| **6.4** | Generate Query | 48 |
| **6.5** | All users information | 49 |
| **6.6** | Selecting Aggregate | 49 |
|  |  |  |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| **6.7** | Selecting Attributes | 50 |
| **6.8** | Generating Query | 50 |
| **6.9** | Run Query Results | 51 |
| **6.10** | Updating Values | 51 |
| **6.11** | Selecting Update Values | 52 |
| **6.12** | Selecting Update Type | 52 |
| **6.13** | Running Update Query | 53 |
| **6.14** | Selecting Delete Commands | 53 |
| **6.15** | Selecting Delete Commands Values | 54 |
| **6.16** | Generating Delete Query | 54 |
| **6.17** | Running Query | 55 |

**CHAPTER 1: INTRODUCTION**

**1. INTRODUCTION**

In this contemporary world, programming is mostly done with IDE, but there are rarely any development environments for accessing database objects. People, generally developers find it hard to remember database commands since they are more complex than usual programming code.

Our solution is to provide a web based application that generates queries automatically without manually entering the whole command. We hope to achieve this by providing drop down menus containing operations and commands. After user selects his preferences, they are validated and the query is generated and sent to the user. Since it’s a self generating system, there are no syntax errors and any user with basic database knowledge can use our system.

These also provide self generated queries but there are restricted to only few commands for which includes all DML commands this separates us from the existing systems.

**1.1 EXISTING SYSTEM:-**

There are few similar existing systems like

* Oracle SQL Developer
* SSMS – only suggestions
* SQL YOG

All these systems have certain short comings, like SQLYog can build only SELECT queries. So, these systems require extra effort on the user’s side. These systems require users to memorize most of the commands. These systems are limited to few commands unlike our system. This makes users frustrating and uncomfortable. This is a user friendly web service.

**PROBLEM DESCRIPTION****:-**

Programming is mostly done with IDE, but there are rarely any development environments for accessing database objects. People, generally developers find it hard to remember database commands since they are more complex than usual programming code.

Our solution is to provide a web based application that generates queries automatically without manually entering the whole command. After user selects his preferences, they are validated and the query is generated and sent to the user. Since it’s a self generating system, there are no syntax errors and any user with basic database knowledge can use our system

**SCOPE AND OBJECTIVES:-**

In the Query builder project, we can build self generated queries using Select menus. Since its automated, it is user flexible. There are no query builders for all DML commands; it can be modified in future to support many other commands.

**OBJECTIVES:-**

* No complex prerequisites.
* The project must be simple and user friendly
* The project must be efficient
* By this software manual effort can be reduced
* Naive user compatibility
* It reduces programming complexity
* Save users time
* To Provide users with a platform of self generating query system

**1.2 PROPOSED SYSTEM: -**

Query Builder is which is concerned with the creation of a website which allows users to manage MySQL databases. The users can also Generate Query with the required commands which user can select to Generate Queries. The Query Builder is the most powerful manager, admin and GUI tool for MySQL, combining the features of MySQL Query Browser, Administrator, phpMyAdmin and other MySQL Front Ends and MySQL GUI tools in a single intuitive interface. Query Builder is a fast, easy to use and compact graphical tool for managing your MySQL databases. Query Builder was developed for all who use MySQL as their preferred RDBMS. Whether you enjoy the control of handwritten SQL or prefer to work in a visual environment, Query Builder makes it easy for you to get started and provides you with tools to enhance your MySQL experience.

**ADVANTAGES OF THE PROPOSED SYSTEM:-**

* User Friendly
* Naïve user compatibility
* Time Saving
* Convenience
* No need to remember all the syntaxes
* Basic knowledge of DBMS is sufficient

**CHAPTER 2: LITERATURE SURVEY**

**2. LITERATURE SURVEY**

**2.1. RELATED WORK**

We have researched various online Query Builder websites to know how what they work and what services they are providing to the users we have observed that every website has limited options and they are not that compatible to build Queries .This makes Query Builder very tiring and the user had only limited option to build Queries. So, users find difficulty in using the Query Builder tools. By going through all these we found that there is a problem regarding Building Query.

We have researched various Query Builders like:

1. SQL Yog
2. Oracle SQL Developer

To know how they function and how they build query we found that to solve this problem everything should be integrated into one platform so that user can easily build and generates query in a single platform with basic knowledge of MySQL.

**2.2. SYSTEM STUDY**

It is always necessary to study and recognize the problems of the existing system, which will help in finding out the requirements for the new system. System study helps in finding different alternatives for better solution. The project study basically deals with different operations and steps involved in it.

In the existing system we have only select statements to build a query that is a limited source to generate all the queries required in real life applications. This Query Builder will have many options in select as well as some more DML commands in order to provide user with more options. Even though in existing system it has select, it does not work with joins and some other operations but the proposed system will be used to generate all types of queries without any syntax errors and most efficient way to generate queries. We provide user with all types of commands in these three types of queries along with sub-queries but the existing system is missing in these minute changes.

**CHAPTER 3: DESIGN**

**3.1 DESIGN**

**3.1. System Requirement Specification:-**

**1. Hardware Requirements:-**

* Processor: 2.0GHz
* Ram Memory: 1GB
* Hard Disk: 10GB

**2. Software Requirements:-**

* Operating system: windows XP or higher/ Linux
* HTML (Hyper Text Markup Language)
* CSS (Cascading Style Sheet)
* JS (Java Script)
* Bootstrap
* XAMPP Server(Cross-platform, Apache, MariaDB (Mysql), PHP and Perl )
* MySQL
* Star UML

The minimum hardware topology for a small dataset scenario is a single-server deployment that contains the following three tiers:

* + - MySQL Server
    - Basic Data Set

The minimum recommended hardware requirements for the servers in this dataset are as follows:

|  |  |
| --- | --- |
| **Component** | **Minimum requirement** |
| Processor | 64-bit, 2.00 GHz minimum per core |
| RAM | 1 GB for developer and evaluation use |
| Hardware | You need not install anything, but you will have to run the application. For production use, you need additional free disk space for day-to-day operations |

* 1. **UML DIAGRAMS**

**Use Case Diagram:**

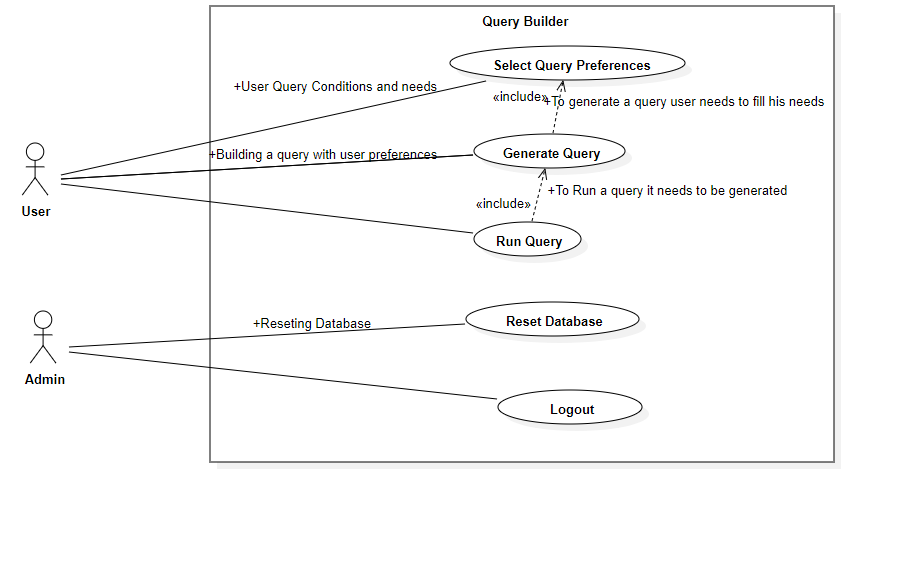


Fig.3.2.1: Use Case Diagram

* A Use Case diagram is used to represent interaction between user and other actors who are involved.
* In this project, the use case diagram represents how the user interacts with the website to do a task.
* After the User entering his specifications for the query then the requirements are sent to Query Builder
* Then the Query Builder Server validates and returns the errors if exists
* If the Requirements passes the validator then the query is generated and returned to the User
* If the user needs to run the Query the Server runs the given query and resulted table or warning is returned to the user.

**Class Diagram:-**

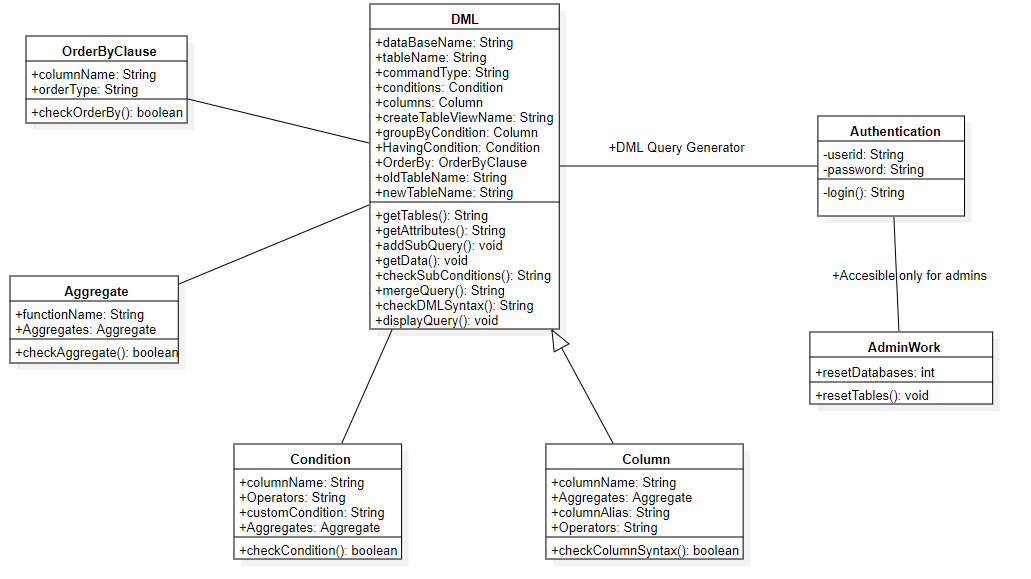
****

Fig.3.2.2: Class Diagram

* A class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.
* In the proposed system we have all the basic commands in MySQL and they are included with the models
* These are provided with some Generic classes and built in classes as well.
* The above are the classes and their attributes along with methods or operations involved in the project.

**Activity Diagram:-**

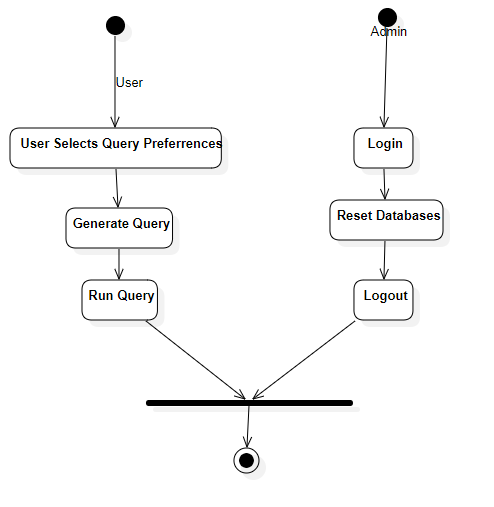


Fig. 3.2.3: Activity Diagram

* A UML diagram used to model the details of complex processes. The documentation about an actor’s title and role in relation to the system and also, relationship between the modules in the system.
* Activity diagram involves both the admin and user where admin alters the database if needed and user accesses the database and controls the operations needed.

**CHAPTER 4: IMPLEMENTATION**

**4. IMPLEMENTATION**

**4.1. MODULES**

**Query Builder:**  
Build complex queries using responsive interface. Visually create MySQL statements without the need of writing commands.

1. Select Module : To use Select command for DML Queries
2. Delete Module : To use Delete command for DML Queries
3. Update Module : To use Update command for DML Queries
4. Insert Module : To use Insert command for DML Queries

**4.2. OVERVIEW TECHNOLOGY**

**Hypertext Markup Language (HTML):-**

Hypertext Markup Language (HTML) is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as cascading style sheets (CSS) and scripting language such as Java script.

Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.

HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects such as interactive forms may be embedded into the rendered page. HTML provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. HTML elements are delineated by *tags*, written using angle brackets. Tags such as <img/> and <input/> directly introduce content into the page. Other tags such as <p> surround and provide information about document text and may include other tags as sub-elements. Browsers do not display the HTML tags, but use them to interpret the content of the page.

HTML can embed programs written in a scripting language such as Java script which affects the behavior and content of web pages. Inclusion of CSS defines the look and layout of content. The World Wide Web Consortium (W3C), former maintainer of the HTML and current maintainer of the CSS standards, has encouraged the use of CSS over explicit presentational HTML since 1997.

**Basic HTML Tags:-**

<! -- --> specifies comments

<A>……….</A> Creates hypertext links

<B>……….</B> Formats text as bold

<BIG>……….</BIG> Formats text in large font.

<BODY>…</BODY> Contains all tags and text in the HTML document

<CENTER>...</CENTER> Creates text

<DD>…</DD> Definition of a term

<DL>...</DL> Creates definition list

<FONT>…</FONT> Formats text with a particular font

<FORM>...</FORM> Encloses a fill-out form

<FRAME>...</FRAME> Defines a particular frame in a set of frames

<H#>…</H#> Creates headings of different levels (1 – 6)

<HEAD>...</HEAD> Contains tags that specify information about a document

<HR>...</HR> Creates a horizontal rule

<HTML>…</HTML> Contains all other HTML tags

<META>...</META> Provides meta-information about a document

<SCRIPT>…</SCRIPT> Contains client-side or server-side script

<TABLE>…</TABLE> Creates a table

<TD>…</TD> Indicates table data in a table

<TR>…</TR> Designates a table row

<TH>…</TH> Creates a heading in a table

**Advantages:-**

A HTML document is small and hence easy to send over the net. It is small because it doesn’t include formatted information.HTML is platform independent and HTML tags are not case-sensitive.

**PHP:-**

Hypertext Preprocessor (or simply PHP) is a general-purpose programming language originally designed for web development. It was originally created by Rasmus Lerdorf in 1994.PHP code may be executed with a command line interface (CLI), embedded into HTML code, or used in combination with various web template systems, web content management systems, and web frameworks. PHP code is usually processed by a PHP interpreter implemented as a module in a web server or as a Common Gateway Interface (CGI) executable. The web server outputs the results of the interpreted and executed PHP code, which may be any type of data, such as generated HTML code or binary image data.PHP can be used for many programming tasks outside of the web context, such as standalone graphic applications and robotic drone control.

**Data Types:-**

The values assigned to a PHP variable may be of different data types including simple string and numeric types to more complex data types like arrays and objects.PHP supports total eight primitive data types: Integer, Floating point number or Float, String, Booleans, Array, Object, resource and NULL.

**Functions:-**

PHP defines a large array of functions in the core language and many are also available in various extensions.

**Objects:-**

An **Object** is an individual instance of the data structure defined by a class. We define a class once and then make many objects that belong to it. Objects are also known as instances.

**Use:-**

PHP is a general-purpose scripting language that is especially suited to server-side web development, in which case PHP generally runs on a web server. Any PHP code in a requested file is executed by the PHP runtime, usually to create dynamic web page content or dynamic images used on websites or elsewhere. It can also be used for command-line scripting and client-side graphical user interface (GUI) applications. PHP can be deployed on most web servers, many operating systems and platforms, and can be used with many relational database management systems (RDBMS). Most web hosting providers support PHP for use by their clients. It is available free of charge, and the PHP Group provides the complete source code for users to build, customize and extend for their own use.

Originally designed to create dynamic web pages, PHP now focuses mainly on server-side scripting, and it is similar to other server-side scripting languages that provide dynamic content from a web server to a client, such as Microsoft's ASP.NET Sun Microsystems Java Server Pages. PHP has also attracted the development of many software frameworks that provide building blocks and a design structure to promote  rapid application development. Some of these frameworks include PRADO, CakePHP, Symphony, CodeIgnitr, Laravel, Yii Framework, Phalcon and Zend Framework offering features similar to other web frameworks.

[](https://en.wikipedia.org/wiki/File:Scheme_dynamic_page_en.svg)

Fig.4.2.1: Dynamic web page

**XAMPP SERVER:-**

XAMPP is a free and open-source cross-platform web server solution stack package developed by Apache Friends, consisting mainly of the Apache HTTP Server, MariaDB database, and interpreters for scripts written in the PHP and Perl programming languages.

**Features:-**

XAMPP is regularly updated to the latest releases of Apache, MariaDB, PHP and Perl. It also includes OpenSSL, phpMyAdmin, MediaWiki, Joomla, WordPress and more. Self-contained, multiple instances of XAMPP can exist on a single computer, and any given instance can be copied from one computer to another. XAMPP is offered in both a full and a standard version (Smaller version)

**Usage:-**

The most obvious characteristic of XAMPP is the ease at which a WAMP web server stack can be deployed and instantiated. Later some common packaged applications that could be easily installed were provided by Bitnami.

Officially, XAMPP's designers intended it for use only as a development tool, to allow website designers and programmers to test their work on their own computers without any access to the Internet. To make this as easy as possible, many important security features are disabled by default.XAMPP has the ability to serve web pages on the World Wide Web. A special tool is provided to password protect the most important parts of the package.

XAMPP also provides support for creating and manipulating databases in MariaDB and SQLite among others.

Once XAMPP is installed, it is possible to treat a localhost like a remote host by connecting using an FTP client. Using a program like Filezilla has many advantages when installing a content management system (CMS) like Joomla or Word press. It is also possible to connect to localhost via FTP with an HTML Editor

**DATABASE:-**

A database management system (DBMS) is computer software designed for the purpose of managing databases, a large set of structured data, and run operations on the data requested by numerous users. Typical examples of DBMSs include Oracle, DB2, Microsoft Access, Microsoft SQL Server, Firebird, PostgreSQL, MySQL, SQLite, FileMaker and Sybase Adaptive Server Enterprise. DBMSs are typically used by Database administrators in the creation of Database systems. Typical examples of DBMS use include accounting, human resources and customer support systems.

Originally found only in large companies with the computer hardware needed to support large data sets, DBMSs have more recently emerged as a fairly standard part of any company back office.

**MYSQL:-**

MySQL is an open-source relational database management system. Its name is combinations of “My”, the name of co-founder Michael Widenius's daughter, and "SQL", the abbreviation for Structured Query Language.

MySQL is a component of the LAMP web application software stack (and others), which is an acronym for Linux, Apache, MySQL, Perl/PHP/Python. MySQL is used by many database-driven web applications, including Drupal, Joomla and WordPress. MySQL is also used by many popular websites, including Face book, Flickr, You Tube MediaWiki and Twitter.

**Features:-**

MySQL is offered fewer than two different editions: the open source MySQL Community Server and the proprietary Enterprise Server. MySQL Enterprise Server is differentiated by a series of proprietary extensions which install as server plugins, but otherwise shares the version numbering system and is built from the same code base.

**Major features as available in MySQL:**

* A broad subset of ANSI SQL 99, as well as extensions
* Cross-platform support
* Stored procedures, using a procedural language that closely adheres to SQL/PSM
* Triggers
* Cursors
* Updatable views
* Online Data Definition Language (DDL) when using the InnoDB Storage Engine.
* Information schema
* Performance Schema that collects and aggregates statistics about server execution and query.
* A set of SQL Mode options to control runtime behavior, including a strict mode to better SQL.
* Transactions with save points when using the default InnoDB Storage Engine.
* ACID compliance when using InnoDB and NDB Cluster Storage Engines[[77]](https://en.wikipedia.org/wiki/MySQL#cite_note-77)
* SSL support
* Query caching
* Sub-SELECTs (i.e. nested SELECTs)
* Built-in replication support
* Asynchronous replication: master-slave from one master to many slaves or many masters to one slave.
* Semi synchronous replication: Master to slave replication where the master waits on replication
* Synchronous replication: Multi-master replication is provided in MySQL Cluster
* Virtual Synchronous: Self managed groups of MySQL servers with multi master support can be done
* using: Galera Cluster  or the built in Group Replication plugin
* Full-text indexing and searching
* Embedded database library
* Unicode support
* Partitioned tables with pruning of partitions in optimizer
* Shared-nothing clustering through MySQL Cluster
* Multiple storage engines, allowing one to choose the one that is most effective for each table.
* Native storage engines InnoDB, MyISAM Merge, Memory (heap), Federated, Archive, CSV.

The developers release minor updates of the MySQL Server approximately every two months. The sources can be obtained from MySQL's website or from MySQL's GitHubrepository, both under the GPL license.

**Functions:-**

1. **ALTER** - Change an existing table, view or index definition (DDL)
2. **AUDIT** - Track the changes made to a table (DDL)
3. **COMMENT** - Add a comment to a table or column in a table (DDL)
4. **COMMIT** - Make all recent changes permanent (DML - transactional)
5. **CREATE** - Create new database objects such as tables or views (DDL)
6. **DELETE**- Delete rows from a database table (DML)
7. **DROP** - Drop a database object such as a table, view or index (DDL)
8. **GRANT** - Allow another user to access database objects such as tables or views (DDL)
9. **INSERT** - Insert new data into a database table (DML)
10. **No AUDIT** - Turn off the auditing function (DDL)
11. **REVOKE** - Disallow a user access to database objects such as tables and views (DDL)
12. **ROLLBACK** - Undo any recent changes to the database (DML - Transactional)
13. **SELECT** - Retrieve data from a database table (DML)
14. **TRUNCATE** - Delete all rows from a database table (cannot be rolled back) (DML)
15. **UPDATE**- Change the values of some data items in a database table (DML)

**STAR UML:-**

StarUML makes a clear conceptual distinction between models, views and diagrams. A Model is an element that contains information for a software model. A View is a visual expression of the information contained in a model, and a Diagram is a collection of view elements that represent the user’s specific design thoughts. StarUML is build as a modular and open tool. It provides frameworks for extending the functionality of the tool. It is designed to allow access to all functions of the model/meta-model and tool through COM Automation, and it provides extension of menu and option items. Also, users can create their own approaches and frameworks according to their methodologies. The tool can also be integrated with any external tools.

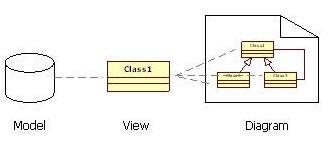


Fig.4.2.2: UML Basic Model

StarUML supports the following diagram types

* Use Case Diagram
* Class Diagram
* Sequence Diagram
* Collaboration Diagram
* Statechart Diagram
* Activity Diagram
* Component Diagram
* Deployment Diagram
* Composite Structure Diagram

**4.3. SAMPLE CODE**

**Index.php:**

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1">

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css" type="text/css">

<link rel="stylesheet" href="https://static.pingendo.com/bootstrap/bootstrap-4.3.1.css">

<script src="Assets/js/Join\_Operations.js"></script>

<script src="Assets/js/Where\_Operations.js"></script>

<script src="Assets/js/OrderBy\_Operations.js"></script>

<script src="Assets/js/GroupBy\_Operations.js"></script>

<script src="Assets/js/Get\_Tables\_Attributes.js"></script>

<script src="Assets/js/Select\_Attribute\_Operations.js"></script>

<script src="Assets/js/Generate\_Query.js"></script>

<script src="Assets/js/Run\_Query.js"></script>

<script src="Assets/js/Having\_Operations.js"></script>

</head>

<script>

function resetEveryThing(){

document.getElementById("number\_of\_joins").value="";

document.getElementById("join\_1").innerHTML="";

try{

for(var i=1;i<=8;i++){

document.getElementById("join\_"+i+"\_table\_1").innerHTML="";

document.getElementById("join\_"+i+"\_table\_2").innerHTML="";

}

}catch(e){}

document.getElementById("attributes\_preview\_textarea").innerHTML="";

document.getElementById("attrbutes\_invisible").innerHTML="";

if(document.getElementById("select\_attribute\_name")!=undefined)

document.getElementById("select\_attribute\_name").innerHTML="";

document.getElementById("attribute\_1\_set\_2").innerHTML="";

document.getElementById("attribute\_1\_set\_3").innerHTML="";

document.getElementById("attribute\_1\_set\_4").innerHTML="";

document.getElementById("where\_attribute").innerHTML="";

document.getElementById("where\_type").value="Select";

if(document.getElementById("where\_attribute\_2")!=undefined){

document.getElementById("where\_attribute\_2").innerHTML="";

}

if(document.getElementById("where\_additional")!=undefined){

document.getElementById("where\_additional").innerHTML="";

}

document.getElementById("where\_preview\_textarea").innerHTML="";

document.getElementById("groupby\_preview\_textarea").innerHTML="";

document.getElementById("groupby\_attribute").innerHTML="";

document.getElementById("having\_preview\_textarea").innerHTML="";

document.getElementById("having\_attribute").innerHTML="";

if(document.getElementById("having\_attribute\_2")!=undefined){

document.getElementById("having\_attribute\_2").innerHTML="";

}

document.getElementById("div\_having\_select\_type").innerHTML="<label class='form-label'>&nbsp;</label><select class='form-control'></select>";

document.getElementById("having\_additional").innerHTML="";

document.getElementById("orderby\_preview\_textarea").innerHTML="";

document.getElementById("orderby\_attributes").innerHTML="";

}

function Load\_All\_Attributes(){

var databasename=document.getElementById("database\_1").value;

var main\_table=document.getElementById("main\_table").value;

var njoins=document.getElementById("number\_of\_joins").value;

if(databasename=="Select"||main\_table=="Select"){

alert("Not Selected Database / Table");

return;

}

a=[];

a.push(document.getElementById("main\_table").value);

for(var i=1;i<=njoins;i++){

var j1=document.getElementById("join\_"+i+"\_table\_1").value;

var j2=document.getElementById("join\_"+i+"\_table\_2").value;

if((j1=="Select"||j1.length==0)||(j2=="Select"||j2.length==0)){

continue;

}

if(!a.includes(document.getElementById("join\_"+i+"\_table\_1").value))

if(document.getElementById("join\_"+i+"\_table\_1").value!="Select")

a.push(document.getElementById("join\_"+i+"\_table\_1").value);

if(!a.includes(document.getElementById("join\_"+i+"\_table\_2").value))

if(document.getElementById("join\_"+i+"\_table\_2").value!="Select")

a.push(document.getElementById("join\_"+i+"\_table\_2").value);

}

console.log(a);

if (window.XMLHttpRequest) {

xmlhttp = new XMLHttpRequest();

} else {

xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");

}

xmlhttp.onreadystatechange = function() {

if (this.readyState == 4 && this.status == 200) {

var s=this.responseText;

document.getElementById("select\_attribute\_name").innerHTML=s;

document.getElementById("where\_attribute").innerHTML=s;

if(document.getElementById("where\_attribute\_2")!=undefined)

document.getElementById("where\_attribute\_2").innerHTML=s;

document.getElementById("groupby\_attribute").innerHTML=s;

document.getElementById("having\_attribute").innerHTML=s;

if(document.getElementById("having\_attribute\_2")!=undefined)

document.getElementById("having\_attribute\_2").innerHTML=s;

document.getElementById("orderby\_attributes").innerHTML=s;

}

};

xmlhttp.open("GET","DML Operations/GetAll\_Where\_Attributes.php?database="+databasename+"&&tables="+a.join(","),true);

xmlhttp.send();

Load\_Attributes\_Datatypes(a);

}

function Load\_Attributes\_Datatypes(a){

if (window.XMLHttpRequest) {

xmlhttp = new XMLHttpRequest();

} else {

xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");

}

xmlhttp.onreadystatechange = function() {

if (this.readyState == 4 && this.status == 200) {

var s=this.responseText;

document.getElementById("hidden\_attributes").value=s;

var s1=s.split("-");

var q="";

for(var i=0;i<s1.length;i++){

q=q+"<tr>";

q=q+"<td>"+s1[i].split(":")[0].trim()+"</td>";

q=q+"<td>"+s1[i].split(":")[1].trim()+"</td>";

q=q+"</tr>";

}

document.getElementById("table\_id\_datatype").innerHTML=q;

}

};

xmlhttp.open("GET","DML Operations/Datatype\_Attributes\_Js.php?database="+document.getElementById("database\_1").value+"&&tables="+a.join(","),true);

xmlhttp.send();

}

</script>

<body>

<nav class="navbar navbar-expand-lg navbar-light">

<div class="container">

<a class="navbar-brand text-primary" href="#"> Query Builder </a>

<button class="navbar-toggler navbar-toggler-right border-0" type="button" data-toggle="collapse" data-target="#navbar5">

<span class="navbar-toggler-icon"></span>

</button>

<div class="collapse navbar-collapse" id="navbar5">

<ul class="navbar-nav ml-auto">

<li class="nav-item"> <a class="nav-link" href="index.php">Select</a> </li>

<li class="nav-item"> <a class="nav-link" href="insert.php">Insert</a> </li>

<li class="nav-item"> <a class="nav-link" href="update.php">Update</a> </li>

<li class="nav-item"> <a class="nav-link" href="delete.php">Delete</a> </li>

</ul>

</div>

</div>

</nav>

<div class="py-2">

<div class="container">

<div class="row">

<div class="col-md-12">

<ul class="nav nav-tabs">

<li class="nav-item"> <a id="main\_tab" href="" class="active nav-link" data-toggle="tab" data-target="#tabone">Main</a> </li>

<li class="nav-item"> <a class="nav-link" href="" data-toggle="tab" data-target="#tabtwo">Join</a> </li>

<li class="nav-item"> <a href="" class="nav-link" data-toggle="tab" data-target="#tabthree">Select</a> </li>

<li class="nav-item"> <a href="" class="nav-link" data-toggle="tab" data-target="#tabfour">Where</a> </li>

<li class="nav-item"> <a href="" class="nav-link" data-toggle="tab" data-target="#tabfive">Group By</a> </li>

<li class="nav-item"> <a href="" class="nav-link" data-toggle="tab" data-target="#tabsix">Having</a> </li>

<li class="nav-item"> <a href="" class="nav-link" data-toggle="tab" data-target="#tabseven">Order By</a> </li>

<style>

.dd{

position: relative;

display: inline-block;

}

.ddc {

display: none;

position: absolute;

background-color: #f9f9f9;

min-width: 160px;

box-shadow: 0px 8px 16px 0px rgba(0,0,0,0.2);

padding: 12px 16px;

z-index: 1;

}

.dd:hover .ddc {

display: block;

}

</style>

<li class="nav-item"> <a href="#" class="nav-link dd"> <span>Datatypes</span>

<div class="ddc">

<table border="1" id="table\_id\_datatype">

</table>

</div>

</a> </li>

</ul>

<br>

<div class="tab-content mt-2">

<div class="tab-pane fade show active" id="tabone" role="tabpanel">

<div class="row">

<div class="col-md-3">

<label class="form-label">Database</label>

<div class="input-group">

<select class="form-control" id="database\_1" onchange="function f(){getAttributes();resetEveryThing();}f();" required="">

<option disabled="" selected="">Select</option>

<option>World</option>

<option>UserDB</option>

</select>

</div>

</div>

<div class="col-md-3">

<label class="form-label">Table</label>

<div class="input-group">

<select class="form-control" id="main\_table" required="" onchange="function f(){resetEveryThing();Load\_All\_Attributes();}f();">

<option></option>

</select>

</div>

</div>

<div class="col-md-3">

<label class="form-label">Display Count</label>

<div class="input-group">

<input type="text" class="form-control" placeholder="Limit 10" id="limit\_count">

</div>

</div>

<div class="col-md-3">

<label class="form-label">Create Table</label>

<div class="input-group">

<input type="text" class="form-control" onkeydown="" pattern="[\_a-zA-Z][\_a-zA-Z0-9]{0,30}" placeholder="Table Name" id="create\_table">

<script>

function identifier\_func(event) {

/\*

var x = event.which || event.keyCode;

if(!((x>=48&&x<=57)||(x>=97&&x<=122)||(x>=65&&x<=90))){

return false;

}

\*/

}

</script>

</div>

</div>

</div>

</div>

<div class="tab-pane fade" id="tabtwo" role="tabpanel"><div class="row" id="div\_preview\_select\_attributes">

</div>

<div class="row">

<div class="col-md-3">

<label class="form-label">&nbsp;</label>

<div class="input-group">

<input type="number" class="form-control" id="number\_of\_joins" placeholder="Number of Joins">

</div>

</div>

<div class="col-md-1">

<label class="form-label"> &nbsp; </label><br>

<button class="btn btn-primary" href="#" onclick="generate\_join\_tables()"> GO </button>

</div>

</div>

<br>

<div class="row" id="join\_1">

</div>

<div class="row" id="join\_2"></div>

<div class="row" id="join\_3"></div>

<div class="row" id="join\_4"></div>

<div class="row" id="join\_5"></div>

<div class="row" id="join\_6"></div>

<div class="row" id="join\_7"></div>

<div class="row" id="join\_8"></div>

</div>

<div class="tab-pane fade" id="tabthree" role="tabpanel">

<div class="row" id="div\_preview\_select\_attributes">

<textarea disabled="" id="attributes\_preview\_textarea" class="form-control" style="margin-left:20px;margin-right:20px;background-color:#FFFFFF;" rows="4" cols="20"></textarea>

</div>

<br>

<button class="btn btn-primary" onclick="Load\_Select\_Attributes()">Load Attributes</button>

&nbsp&nbsp&nbsp

<button class="btn btn-primary" onclick="document.getElementById('attributes\_preview\_textarea').innerHTML='';">Reset</button>

<select id="attrbutes\_invisible" style="visibility:hidden;">

</select>

<br><br>

<div class="row" id="attribute\_1">

<div class="col-md-2">

<label class="form-label">Aggregate</label>

<select class="form-control" id="attribute\_1\_function" onchange="SetUp\_Attributes()">

<option selected="">Select</option>

<option>DISTINCT</option>

<option>LENGTH</option>

<option>SUM</option>

<option>COUNT</option>

<option>UPPER</option>

<option>LOWER</option>

<option>MAX</option>

<option>MIN</option>

<option>AVG</option>

<option>CONCAT</option>

<option>SUBSTRING</option>

</select>

</div>

<div class="col-md-2" id="attribute\_1\_set\_1">

<label class="form-label">Attributes</label>

<select class="form-control" id="select\_attribute\_name">

<option></option>

</select>

</div>

<div class="col-md-2" id="attribute\_1\_set\_2"></div>

<div class="col-md-2" id="attribute\_1\_set\_3"></div>

<div class="col-md-2" id="attribute\_1\_set\_4"></div>

<div class="col-md-2">

<label class="form-label"> &nbsp; </label><br>

<button class="btn btn-primary" onclick="Generate\_Attributes()">GO</button>

</div>

</div>

</div>

<div class="tab-pane fade" id="tabfour" role="tabpanel">

<textarea disabled="" id="where\_preview\_textarea" class="form-control" style="margin-right:20px;background-color:#FFFFFF;" rows="4" cols="20"></textarea>

<br>

<button class="btn btn-primary" onclick="Where\_setAttributes()">Load Attributes</button>

&nbsp&nbsp&nbsp

<button class="btn btn-primary" onclick="document.getElementById('where\_preview\_textarea').innerHTML='';">Reset</button>

<br>

<br>

<div class="row" id="where\_1">

<div class="col-md-1">

<label class="form-label"> &nbsp; </label><br>

<input type="checkbox" id="where\_no">&nbsp; NO </div>

<div class="col-md-2">

<label class="form-label">Attributes</label>

<div class="input-group">

<select class="form-control" id="where\_attribute">

<option></option>

</select>

</div>

</div>

<div class="col-md-2">

<label class="form-label">Arithmetic operators</label>

<div class="input-group">

<select class="form-control" id="where\_arithmetic">

<option>&gt;</option>

<option>&lt;</option>

<option>&gt;=</option>

<option>&lt;=</option>

<option>!=</option>

<option>=</option>

<option>IN</option>

</select>

</div></div>

<div class="col-md-2">

<label class="form-label">Type</label>

<div class="input-group">

<select class="form-control" id="where\_type" onchange="Where\_Type\_Change(this.value)">

<option disabled="" selected="">Select</option>

<option>Custom Input</option>

<option>Sub Query</option>

<option>Attribute</option>

</select>

</div>

</div>

<div class="col-md-2" id="div\_where\_select\_type">

<label class="form-label">&nbsp;</label>

<select class="form-control"></select>

</div>

<div class="col-md-2">

<label class="form-label">Logical operators</label>

<div class="input-group">

<select class="form-control" id="where\_logical">

<option selected>Select</option>

<option>AND</option>

<option>OR</option>

</select>

</div>

</div>

<!--

<div class="col-md-1" id="where\_1\_add\_button">

<label class="form-label"> &nbsp; </label><br>

<button class="btn btn-primary" href="#" onclick="addNewWhere(2)"> +</button>

</div>

--><div class="col-md-1" id="where\_1\_add\_button">

<label class="form-label"> &nbsp; </label><br>

<button class="btn btn-primary" href="#" onclick="Add\_To\_Preview()">GO</button>

</div>

<div class="col-md-12" id="where\_1\_query\_type"></div>

</div>

<br>

<div class="row" id="where\_additional">

</div>

</div>

<div class="tab-pane fade" id="tabfive" role="tabpanel">

<div class="row" id="div\_preview\_groupby">

<textarea disabled="" id="groupby\_preview\_textarea" class="form-control" style="margin-left:20px;margin-right:20px;background-color:#FFFFFF;" rows="4" cols="20"></textarea>

</div>

<br>

<button class="btn btn-primary" onclick="GroupBy\_setAttributes()">Load Attributes</button>

&nbsp;&nbsp;

<button class="btn btn-primary" onclick="document.getElementById('groupby\_preview\_textarea').value='';">Reset</button>

<br><br>

<div class="row" id="groupby\_1">

<div class="col-md-3">

<label class="form-label">Attribute</label>

<select class="form-control" id="groupby\_attribute">

</select>

</div>

<div class="col-md-1">

<label class="form-label">&nbsp; </label><br>

<button class="btn btn-primary" onclick="GroupBy\_Add\_To\_Preview()">GO</button>

</div>

</div>

</div>

<div class="tab-pane fade" id="tabsix" role="tabpanel">

<textarea disabled="" id="having\_preview\_textarea" class="form-control" style="margin-right:20px;background-color:#FFFFFF;" rows="4" cols="20"></textarea>

<br>

<button class="btn btn-primary" onclick="Having\_setAttributes()">Load Attributes</button>

&nbsp&nbsp&nbsp

<button class="btn btn-primary" onclick="document.getElementById('having\_preview\_textarea').innerHTML='';">Reset</button>

<br><br>

<div class="row" id="where\_1">

<div class="col-md-2">

<label class="form-label">Attributes</label>

<div class="input-group">

<select class="form-control" id="having\_attribute">

<option></option>

</select>

</div>

</div>

<div class="col-md-2">

<label class="form-label">Arithmetic operators</label>

<div class="input-group">

<select class="form-control" id="having\_arithmetic">

<option>&gt;</option>

<option>&lt;</option>

<option>&gt;=</option>

<option>&lt;=</option>

<option>!=</option>

<option>=</option>

<option>IN</option>

</select>

</div>

</div>

<div class="col-md-2">

<label class="form-label">Type</label>

<div class="input-group">

<select class="form-control" id="having\_type" onchange="Having\_Type\_Change(this.value)">

<option disabled="" selected="">Select</option>

<option>Custom Input</option>

<option>Sub Query</option>

<option>Attribute</option>

</select>

</div>

</div>

<div class="col-md-2" id="div\_having\_select\_type">

<label class="form-label">&nbsp;</label>

<select class="form-control"></select>

</div>

<div class="col-md-2">

<label class="form-label">Logical operators</label>

<div class="input-group">

<select class="form-control" id="having\_logical">

<option selected>Select</option>

<option>AND</option>

<option>OR</option>

</select>

</div>

</div>

<div class="col-md-1" id="having\_1\_add\_button">

<label class="form-label"> &nbsp; </label><br>

<button class="btn btn-primary" href="#" onclick="Having\_Add\_To\_Preview()">GO</button>

</div>

<div class="col-md-12" id="having\_1\_query\_type"></div>

</div>

<br>

<div class="row" id="having\_additional">

</div>

</div>

<div class="tab-pane fade" id="tabseven" role="tabpanel">

<div class="row" id="div\_preview\_orderby">

<textarea disabled="" id="orderby\_preview\_textarea" class="form-control" style="margin-left:20px;margin-right:20px;background-color:#FFFFFF;" rows="4" cols="20"></textarea>

</div>

<br>

<button class="btn btn-primary" onclick="OrderBy\_setAttributes()">Load Attributes</button>

&nbsp;&nbsp;

<button class="btn btn-primary" onclick="document.getElementById('orderby\_preview\_textarea').value='';">Reset</button>

<br><br>

<div class="row" id="orderby\_1">

<div class="col-md-3">

<label class="form-label">Attributes</label>

<div class="input-group">

<select class="form-control" id="orderby\_attributes">

<option></option>

</select>

</div>

</div>

<div class="col-md-2">

<label class="form-label">Order Type</label>

<div class="input-group">

<select class="form-control" id="orderby\_type">

<option>ASC</option>

<option>DESC</option>

</select>

</div>

</div>

<div class="col-md-1">

<label class="form-label"> &nbsp; </label><br>

<button class="btn btn-primary" href="#" onclick="OrderBy\_Add\_To\_Preview()"> GO</button>

</div>

</div>

</div>

</div>

</div>

</div>

<center>

<div class="row">

<div class="col-md-10">

<textarea disabled="" id="get\_query" class="form-control" style="position: fixed;width: 62%;bottom:10px;margin-left:20px; margin-right:20px; background-color:#FFFFFF;" rows="6" cols="20"></textarea>

</div>

<div class="col-md-1">

<button href="#result" class="btn btn-primary" style="position: fixed; bottom:100px;" onclick="Generate\_Query()">Generate Query</button>

<a class="btn btn-primary" style="position: fixed; bottom:40px;width:135px;" data-toggle="modal" data-target="#myModal" href="#result\_tab" onclick="run\_query();">Run Query</a>

</div>

</div>

</center>

</div>

<div class="modal fade" id="myModal" role="dialog">

<div class="modal-dialog modal-lg">

<div class="modal-content" style="overflow:scroll;">

<div class="modal-header">

<h4 class="modal-title">Result</h4>

<button type="button" class="close" data-dismiss="modal">&times;</button>

</div>

<div class="modal-body">

<p id="loading">Loading..</p>

<table class="table table-striped table-borderless" style="overflow:scroll;">

<thead id="table\_header">

</thead>

<tbody id="table\_body">

</tbody>

</table>

</div>

<div class="modal-footer">

<button type="button" class="btn btn-default" data-dismiss="modal">Close</button>

</div>

</div>

</div>

</div>

<input type="text" id="hidden\_attributes" hidden>

<br><br><br><br><br><br><br><br>

<script src="https://code.jquery.com/jquery-3.3.1.slim.min.js" integrity="sha384-q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo" crossorigin="anonymous"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.6/umd/popper.min.js" integrity="sha384-wHAiFfRlMFy6i5SRaxvfOCifBUQy1xHdJ/yoi7FRNXMRBu5WHdZYu1hA6ZOblgut" crossorigin="anonymous"></script>

<script src="https://code.jquery.com/jquery-3.3.1.slim.min.js" integrity="sha384-q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo" crossorigin="anonymous"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.6/umd/popper.min.js" integrity="sha384-wHAiFfRlMFy6i5SRaxvfOCifBUQy1xHdJ/yoi7FRNXMRBu5WHdZYu1hA6ZOblgut" crossorigin="anonymous"></script>

<select class="form-control" id="orderby\_type">

<option>ASC</option>

<option>DESC</option>

</select>

</div>

</div>

<div class="col-md-1">

<label class="form-label"> &nbsp; </label><br>

<button class="btn btn-primary" href="#" onclick="OrderBy\_Add\_To\_Preview()"> GO</button>

<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js" integrity="sha384-JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf/nJGzIxFDsf4x0xIM+B07jRM" crossorigin="anonymous"></script>

</div>

</body>

</html>

**CHAPTER 5: TESTING**

**5. TESTING**

Testing is the process of finding differences between the expected behavior specified by system models and the observed behavior of the system.

. They are two types of testing. They are:

1. White box testing.

2. Black box testing.

**White box testing:**

White box testing focus on internal structure of the component.

* In the white box testing the designer can derive test cases that guarantee that all independent paths within a module have been exercised at least once
* Exercise all logical decision on their true and false sides.
* Exercise all loops at their boundaries and within their operational bounds
* Exercise internal data structures to ensure their validity.

**Black box testing:**

Black box testing focus on input or output behavior of the component. That is for a swinger the black box testing will give the desired results for what it is meant for, it means that he has to exercise all functional requirements for a program. The black box testing is done after the software is developed.

Black box testing attempts to find errors in the following:

* Incorrect or missing functions.
* Interface errors.
* Errors in data structures or external database access.
* Performance errors.
* Initialization and termination errors.

**5.1. TEST CASES:**

**Index.php:**

|  |  |  |
| --- | --- | --- |
| **Test Case** | **Pre-condition** | **Expected output** |
| TC1 | Without Selecting Database | Select Database |
| TC2 | Without Selecting Table | Select table |
| TC3 | Selecting Database and Table | Displays the generated query |
| TC4 | Without filling all columns in joins | Please fill the joins |
| TC5 | Without giving custom input in Where clause | Please Enter the custom input |
| TC6 | Adding another where clause without using any logical operations previously | Error has occurred ,Please reset |
| TC7 | Without giving custom input in Having clause | Please Enter the custom input |
| TC8 | Adding another Having clause without using any logical operations previously | Error has occurred ,Please reset |
| TC9 | Click run Query | Displays the query result |

**Update.php:**

|  |  |  |
| --- | --- | --- |
| **Test Case** | **Pre-condition** | **Expected output** |
| TC1 | Without Selecting Database | Select Database |
| TC2 | Without Selecting Table | Select table |
| TC3 | Selecting Database and Table | Displays the generated query |
| TC5 | Without giving custom input in Where clause | Please Enter the custom input |
| TC6 | Adding another where clause without using any logical operations previously | Error has occurred ,Please reset |
| TC7 | Click Generate Query | Displays query |
| TC8 | Click Run Query | Displays the query result |

**Delete.php:**

|  |  |  |
| --- | --- | --- |
| **Test Case** | **Pre-condition** | **Expected output** |
| TC1 | Without Selecting Database | Select Database |
| TC2 | Without Selecting Table | Select table |
| TC3 | Selecting Database and Table | Displays the generated query |
| TC5 | Without giving custom input in Where clause | Please Enter the custom input |
| TC6 | Adding another where clause without using any logical operations previously | Error has occurred ,Please reset |
| TC7 | Click Generate Query | Displays query |
| TC8 | Click Run Query | Displays the query result |

**5.2 TEST RESULTS**

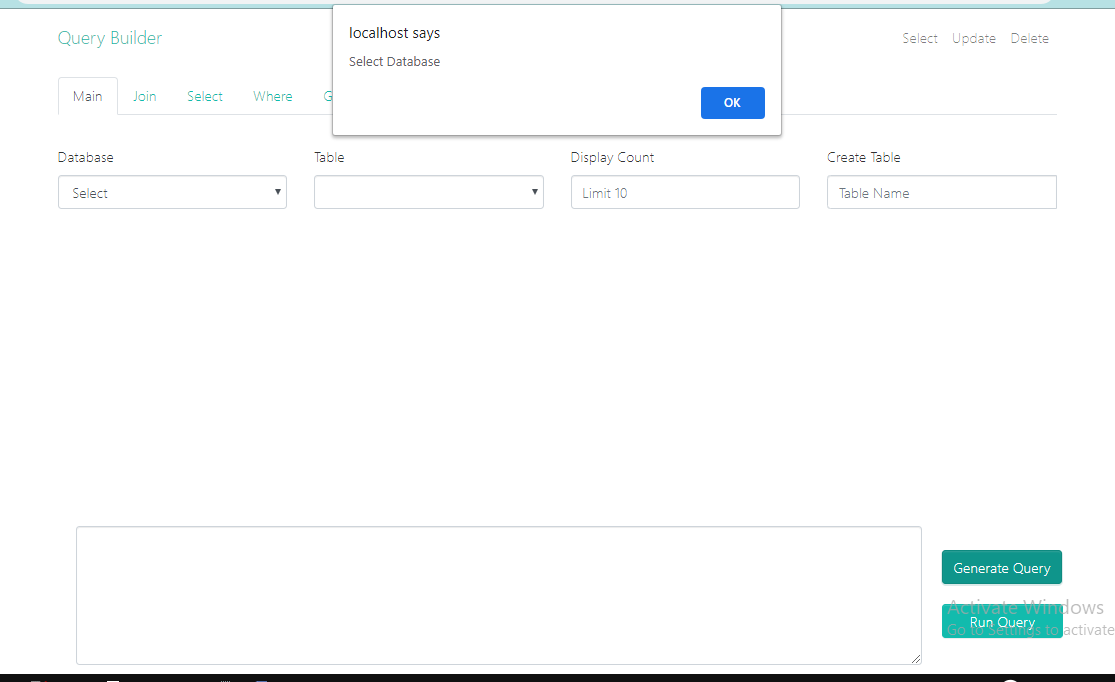
****

Fig.5.2.1: Select Database

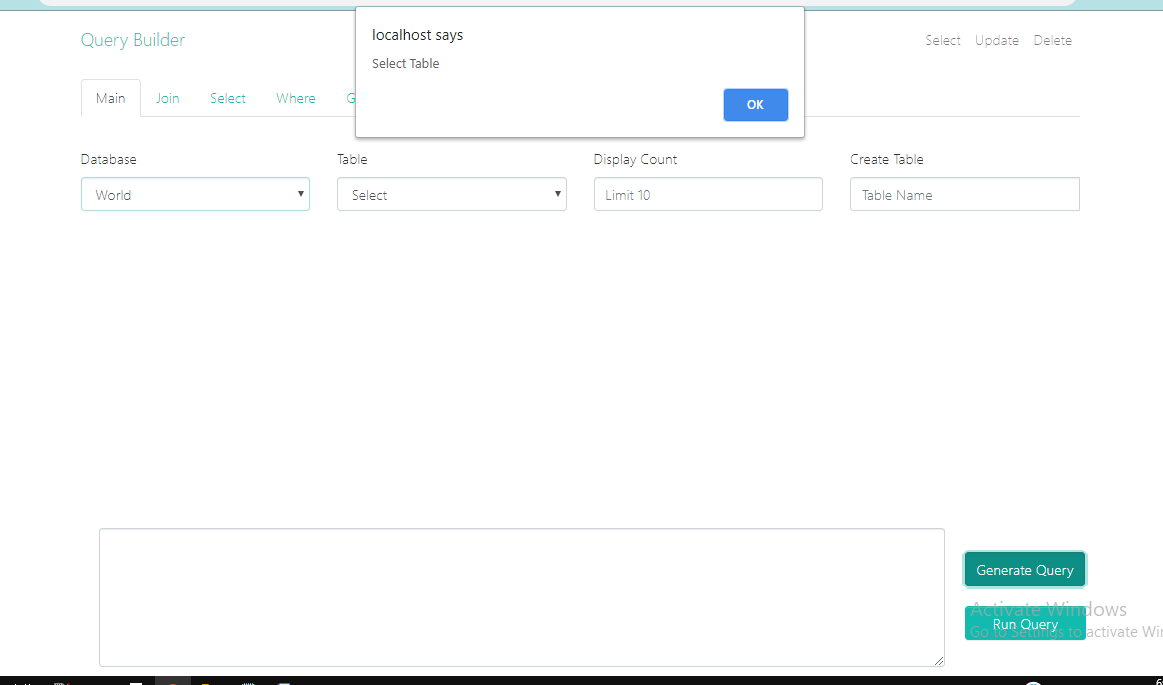
****

Fig.5.2.2: Select Table

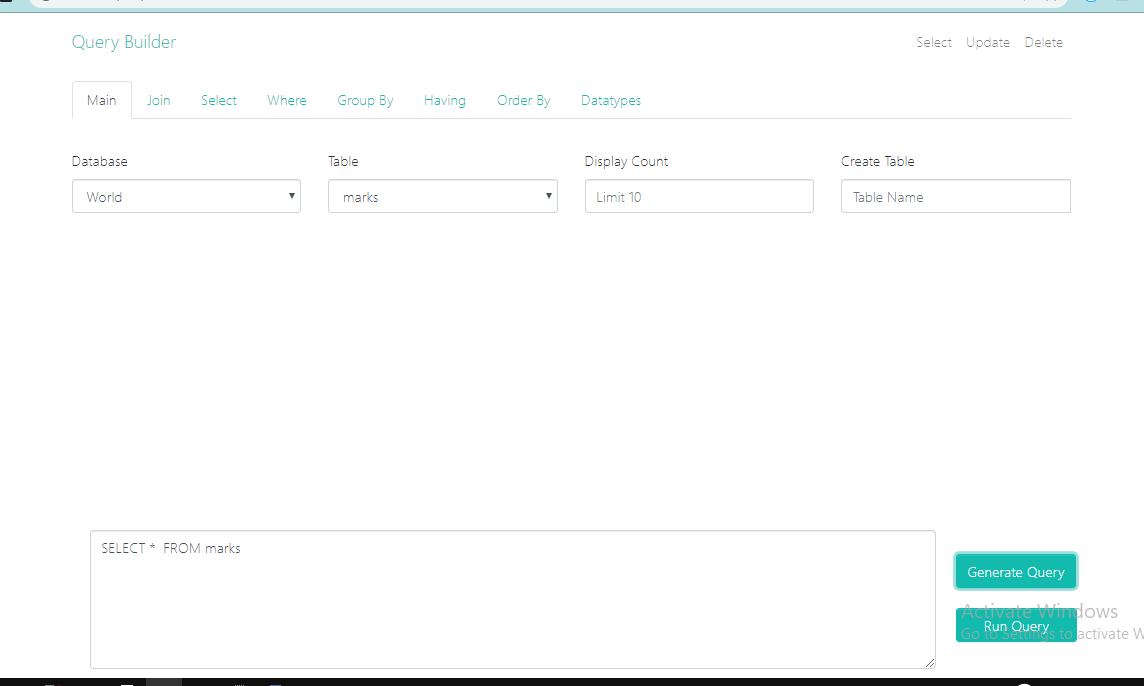
****

Fig.5.2.3: Selecting Database and Table

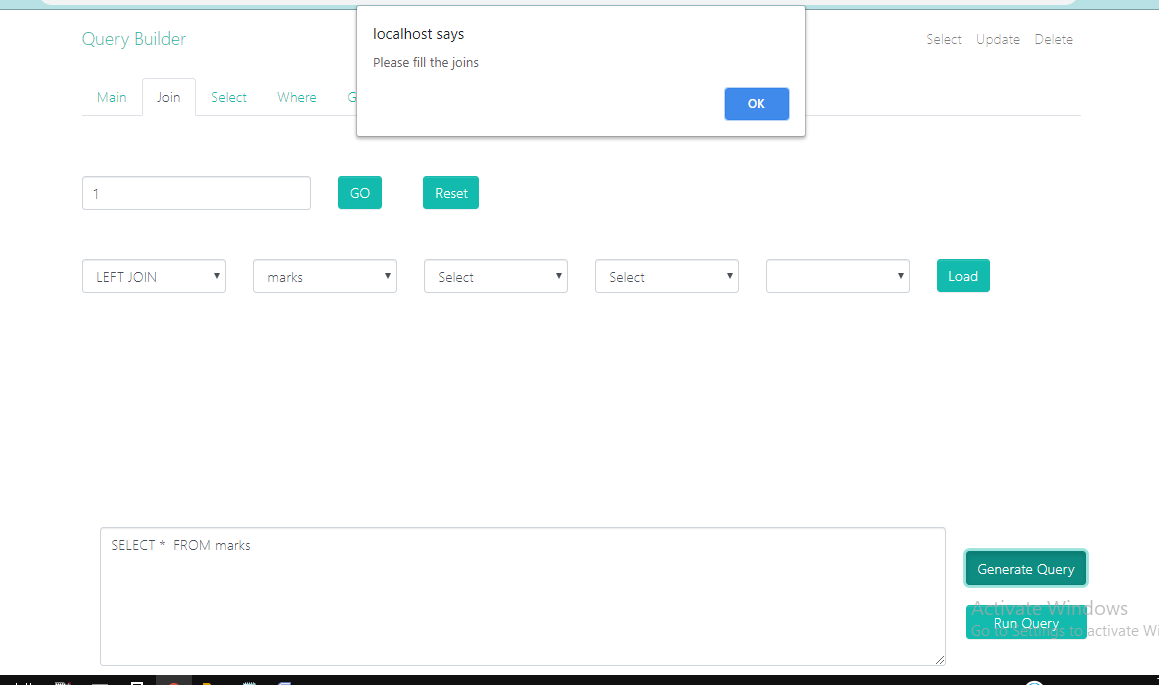


Fig.5.2.4: Incomplete Joins

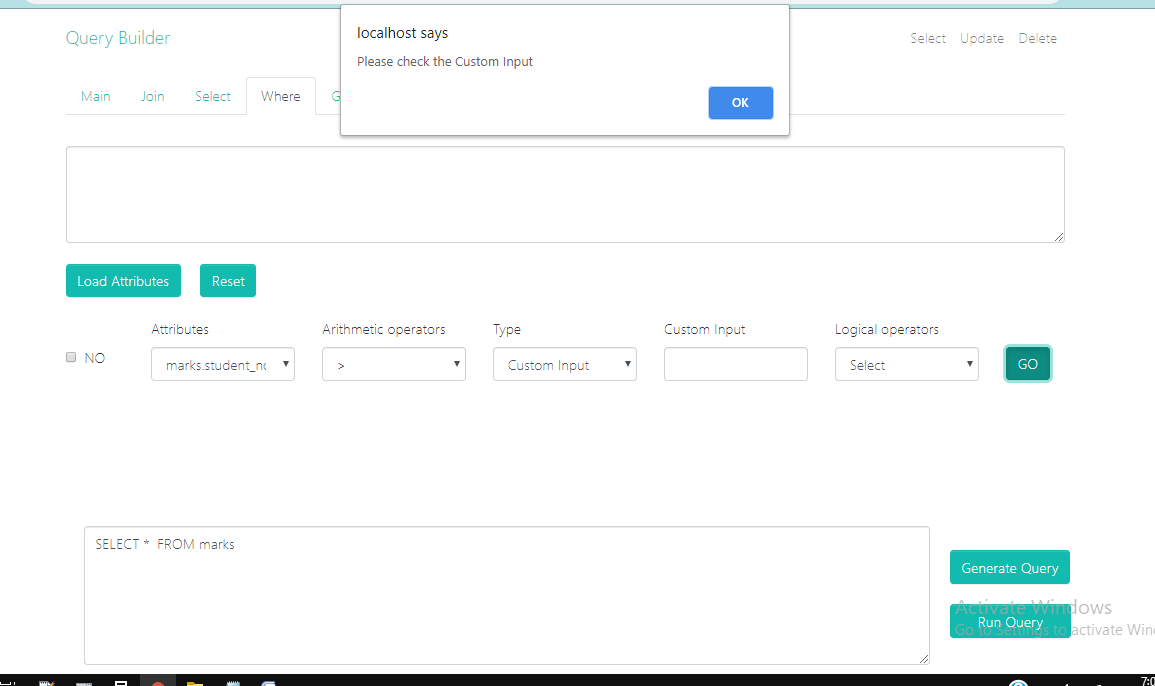
****

Fig.5.2.5 Incomplete Custom Input

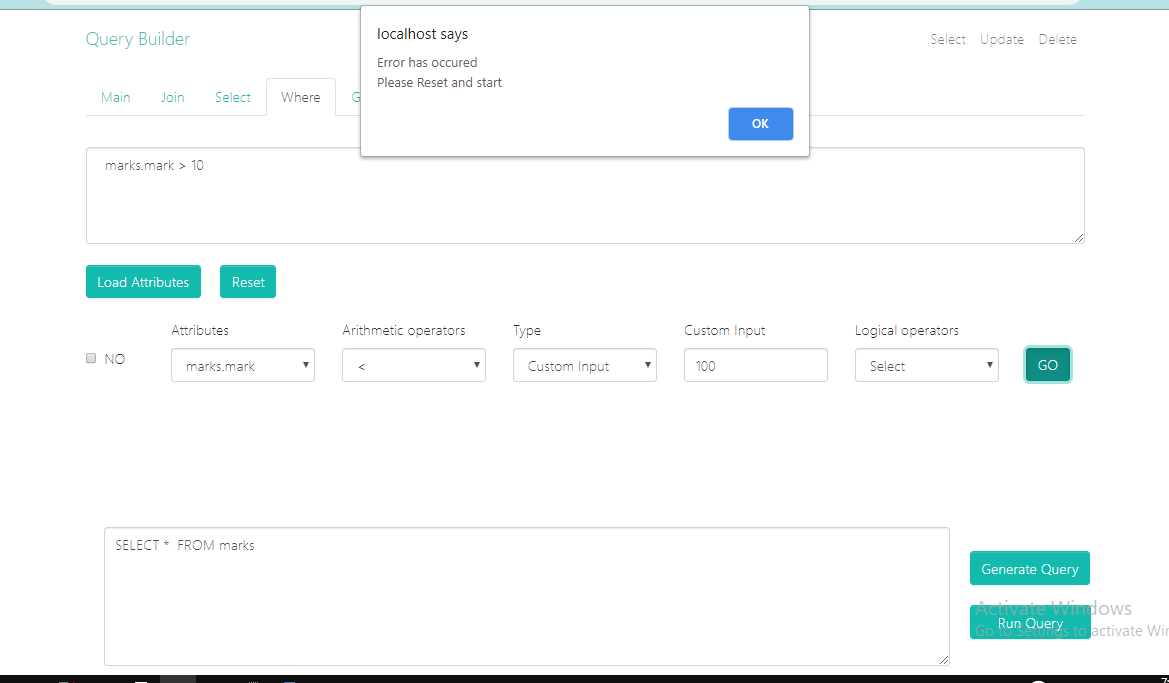
****

Fig.5.2.6 No Logical Operators

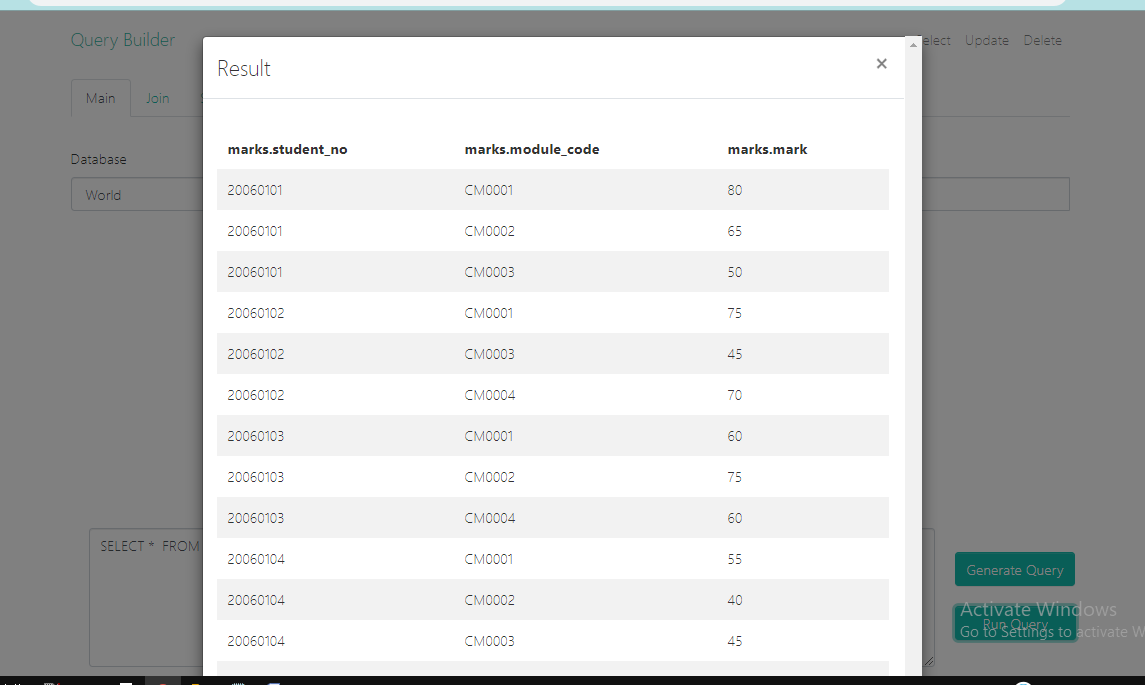
****

Fig.5.2.7 Result

**CHAPTER 6: RESULTS**

**Index page:**

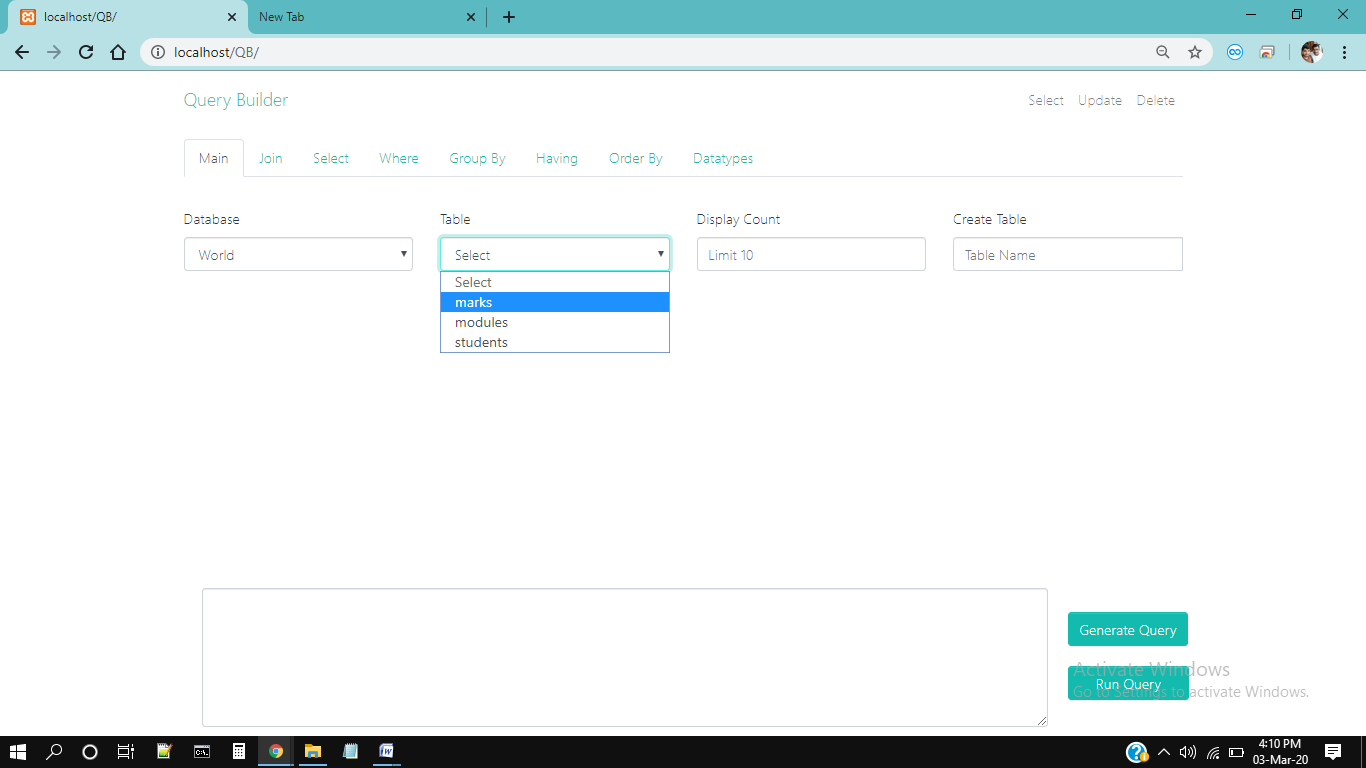
****

Fig.6.1: Selecting Database Table

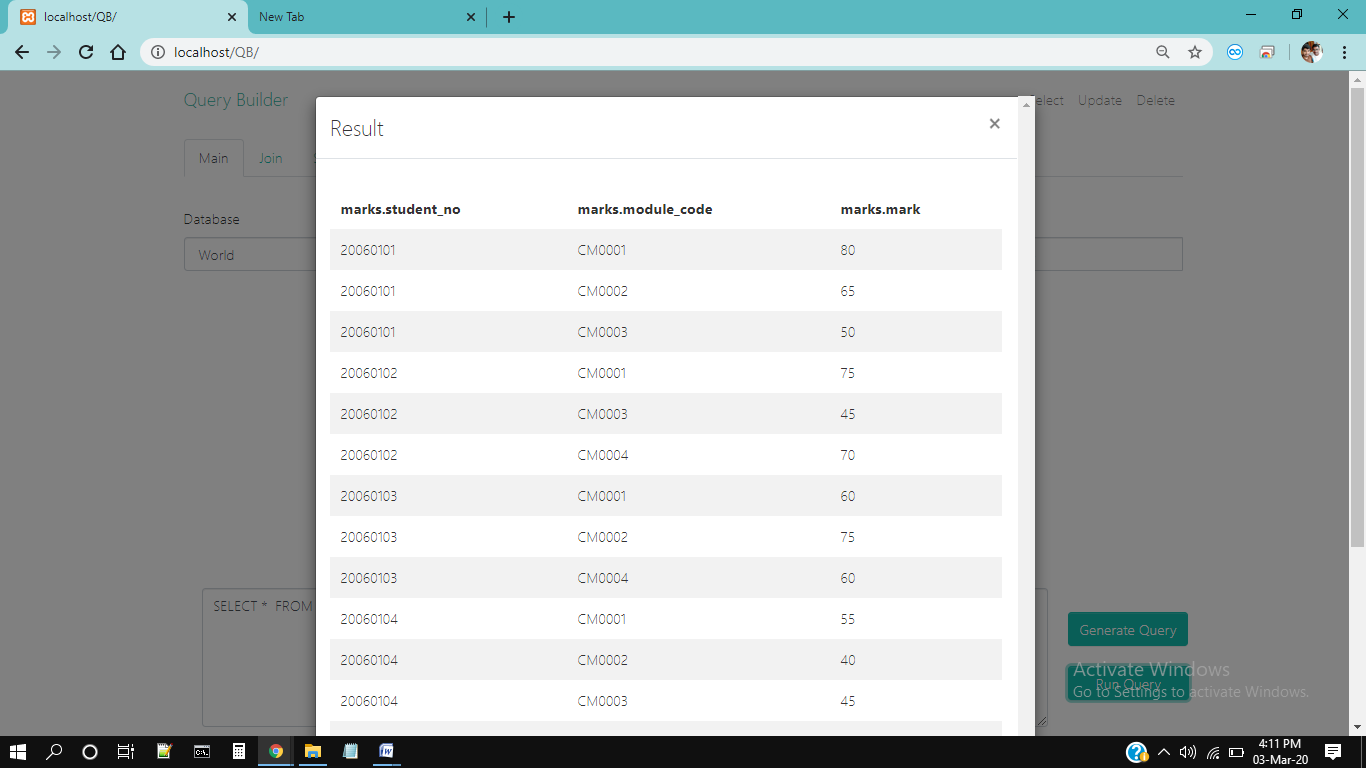
****

Fig.6.2: Results From Table

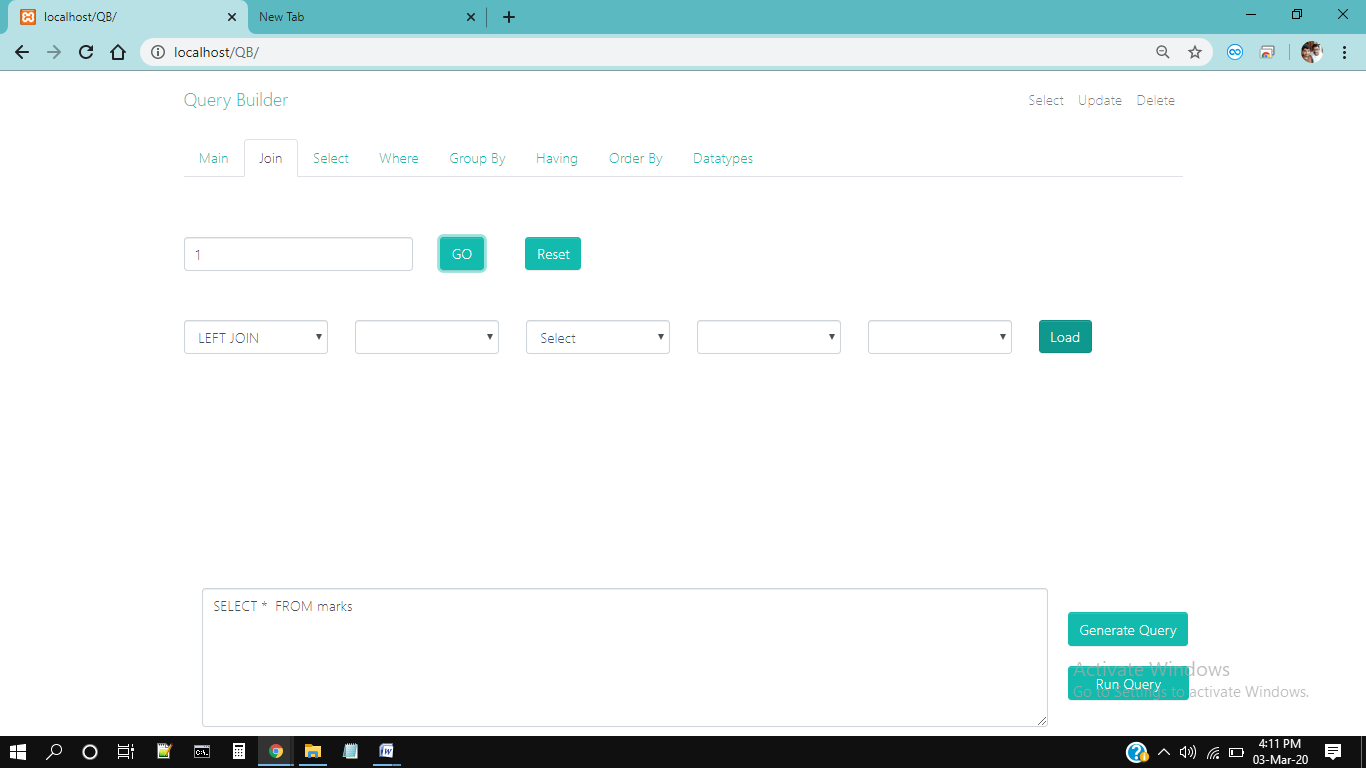
****

Fig.6.3: Selecting Join

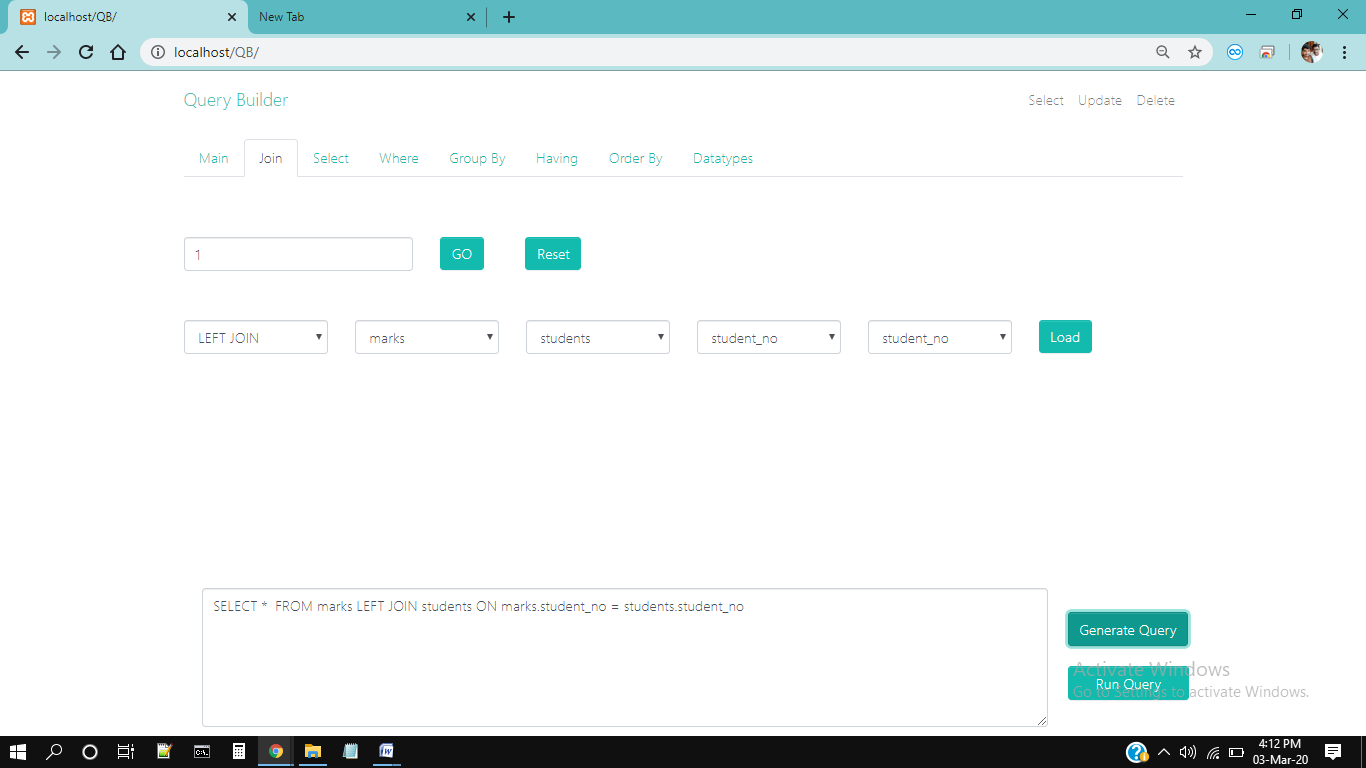
****

Fig.6.4: Generate Query

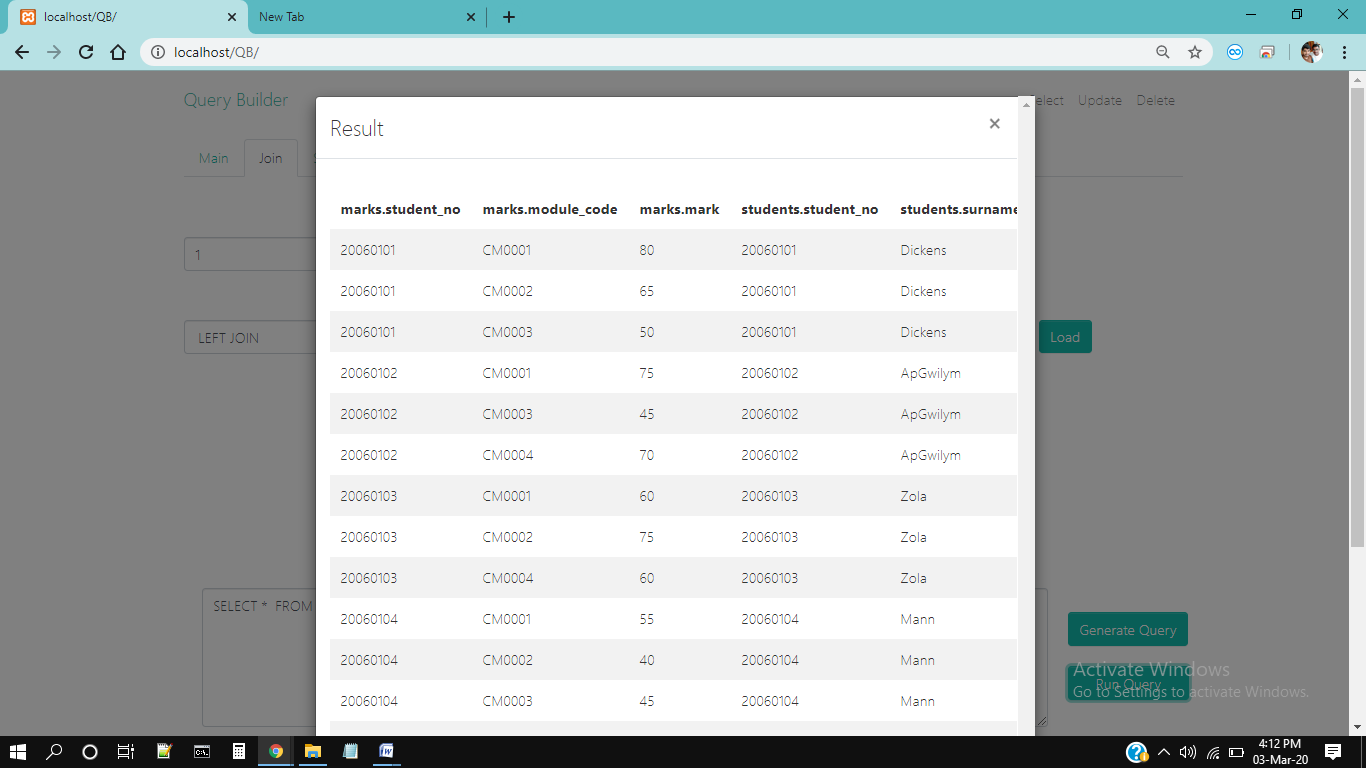
****

Fig.6.5: Run Query

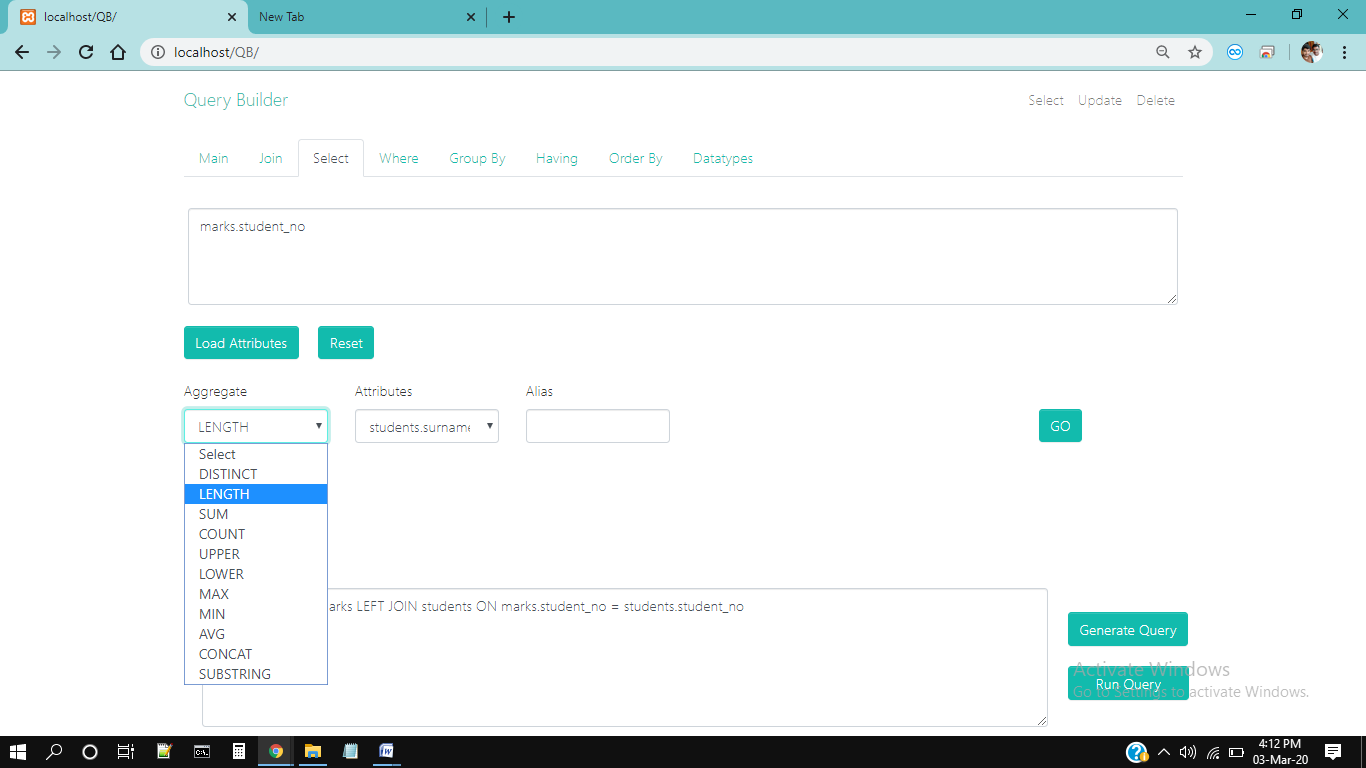
****

Fig.6.6: Selecting Aggregate

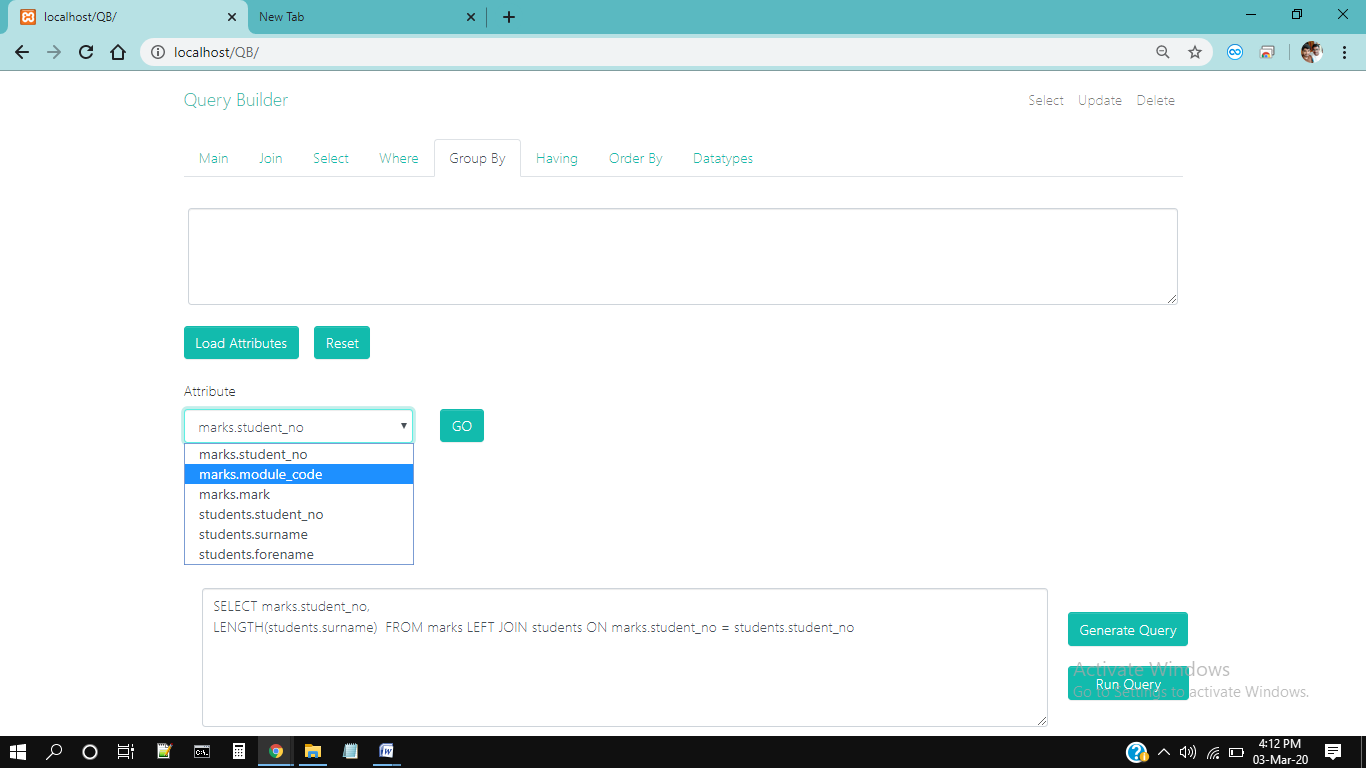
****

Fig.6.7: Selecting Attributes

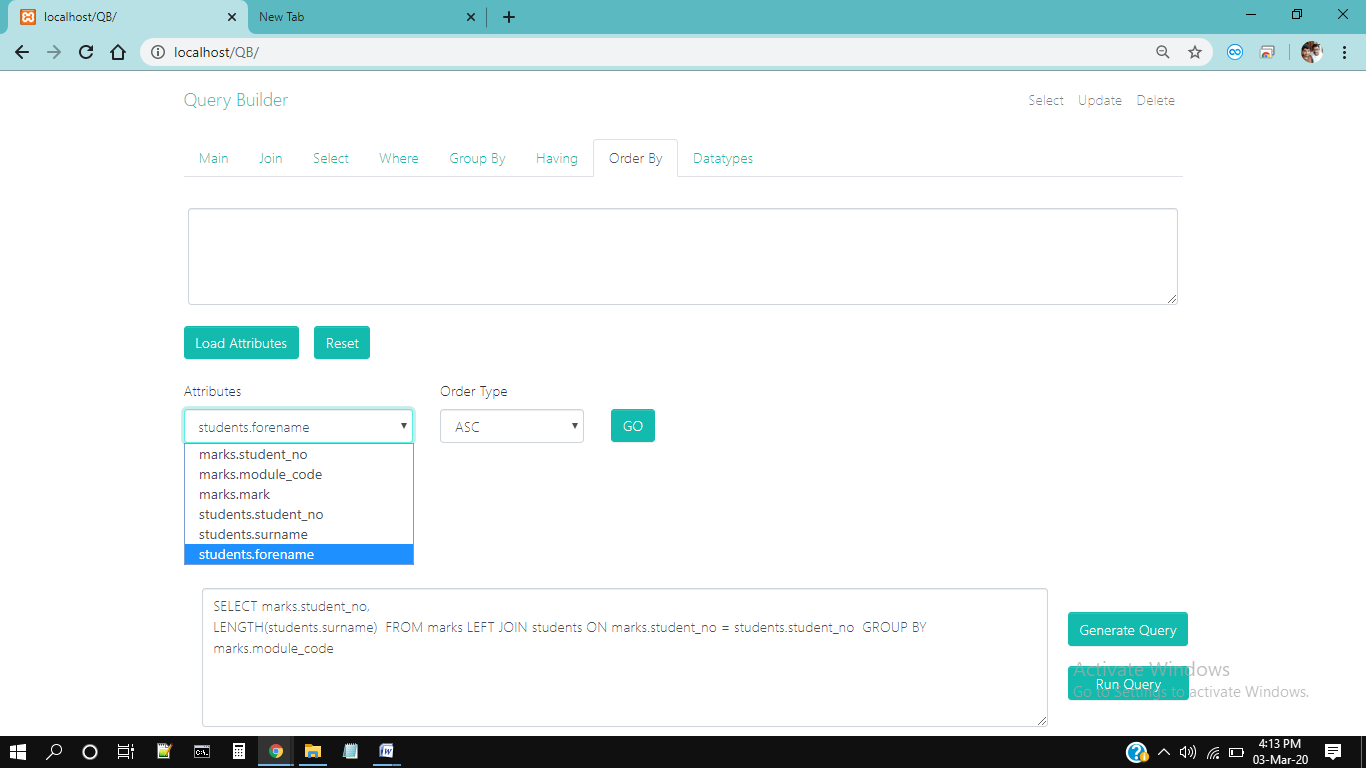
****

Fig.6.8: Generating Query

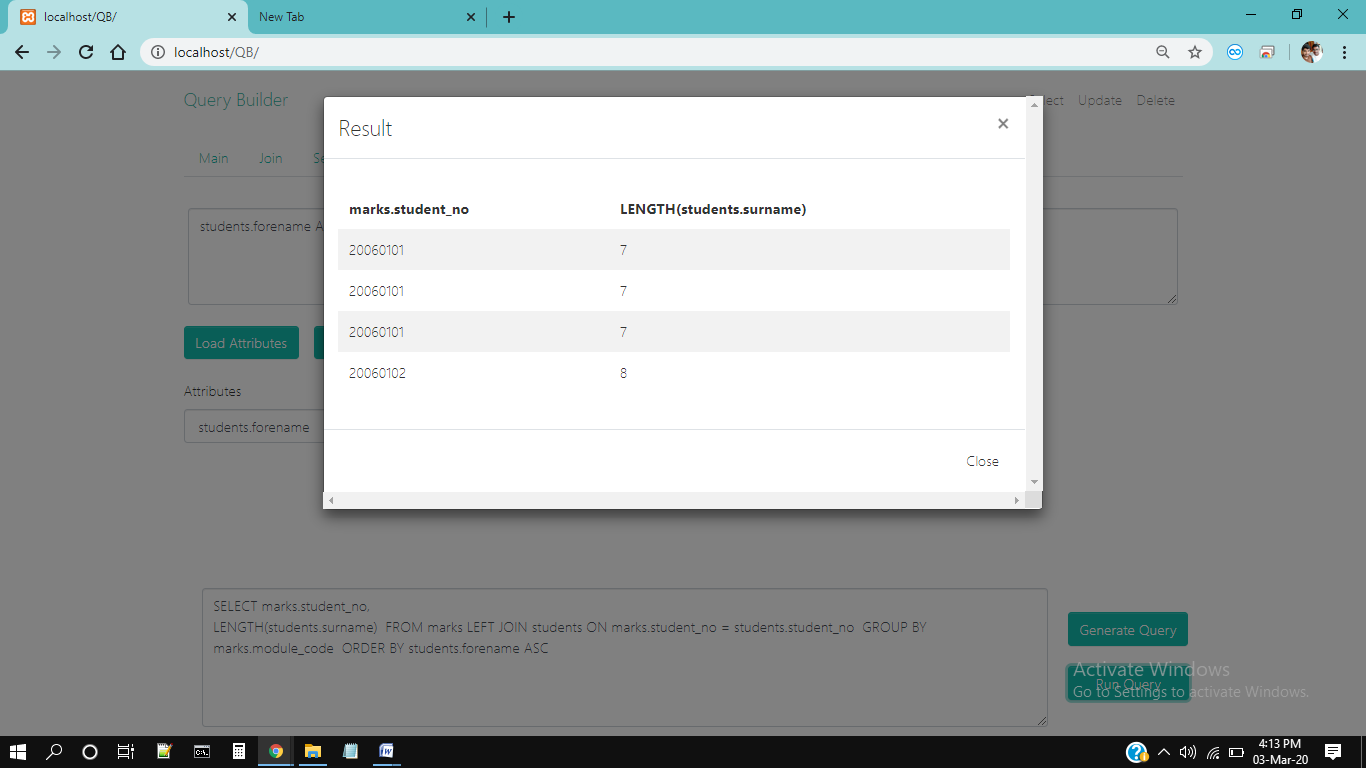
****

Fig.6.9: Run Query Results

**Update Page:**

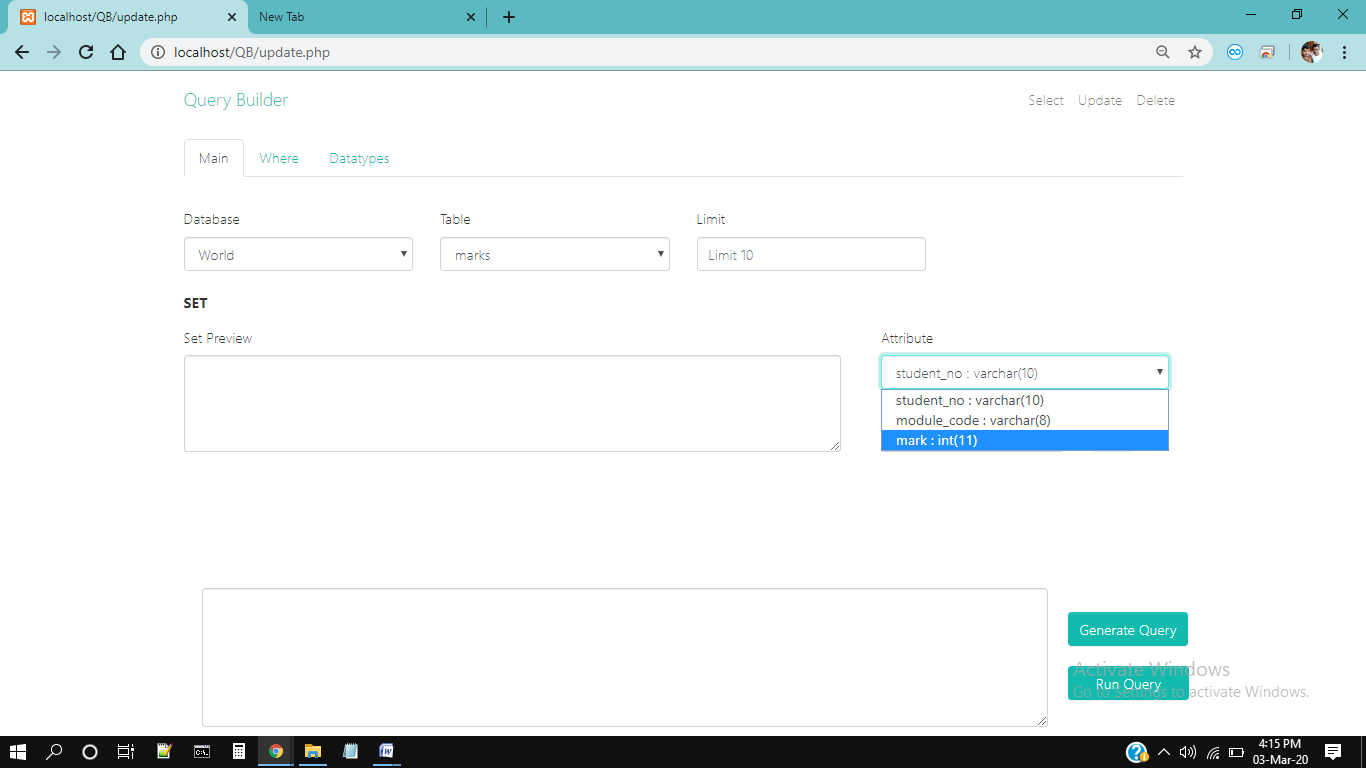
****

Fig.6.10: Updating Values

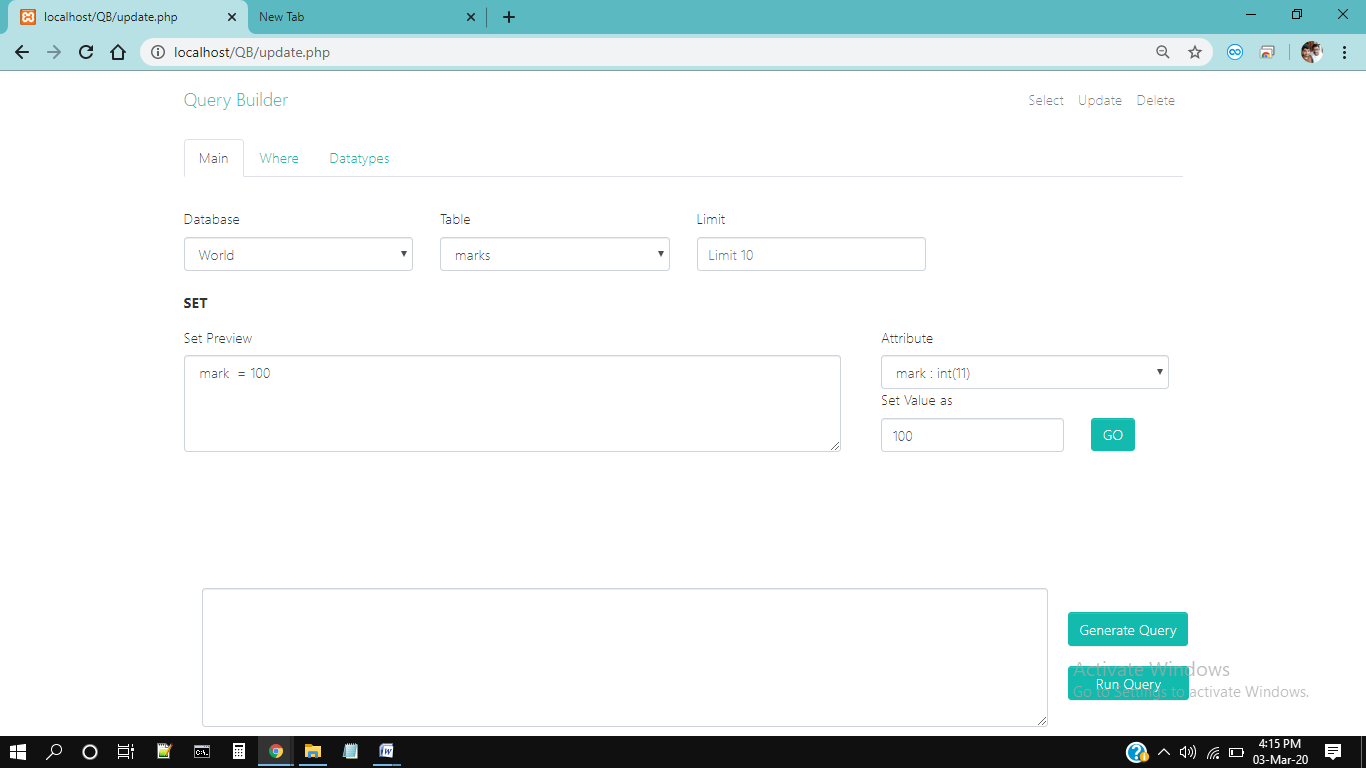
****

Fig.6.11: Selecting Update Values

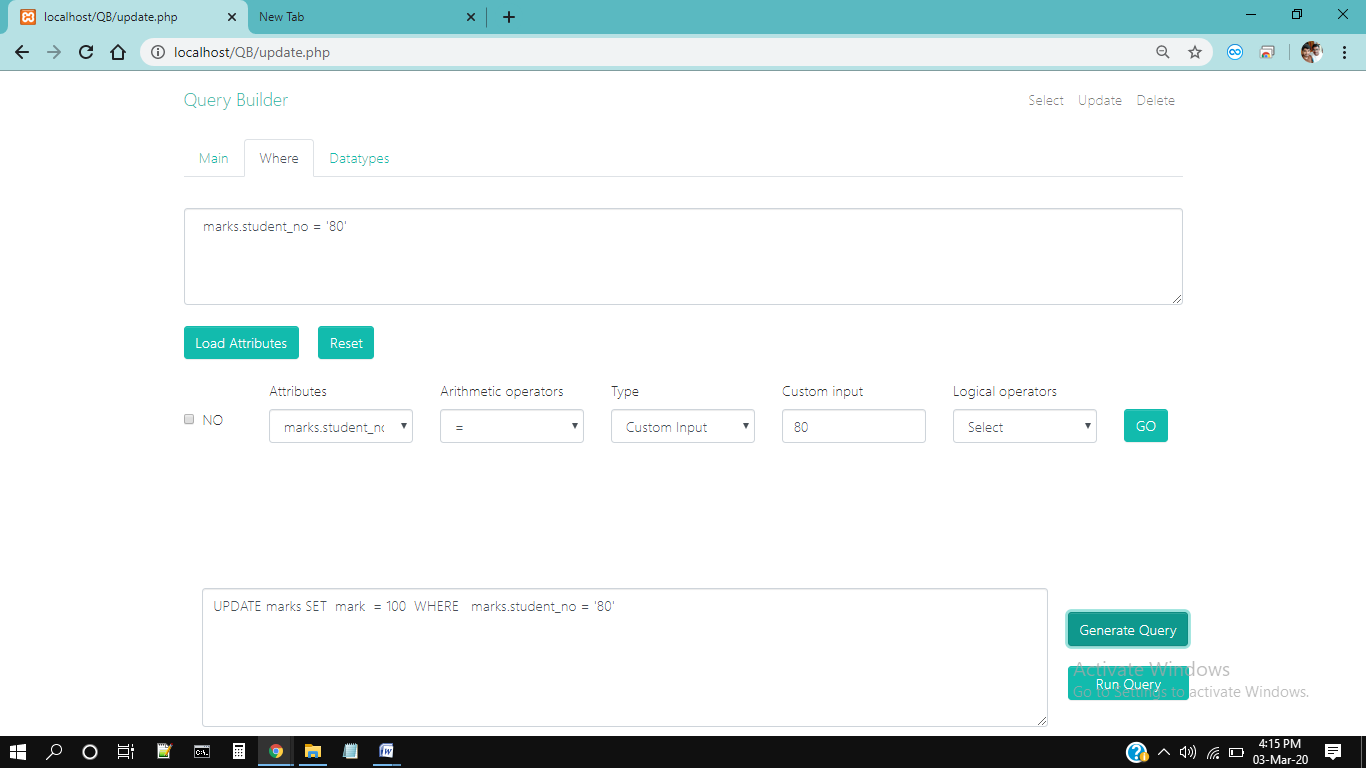
****

Fig.6.12: Selecting Update Type

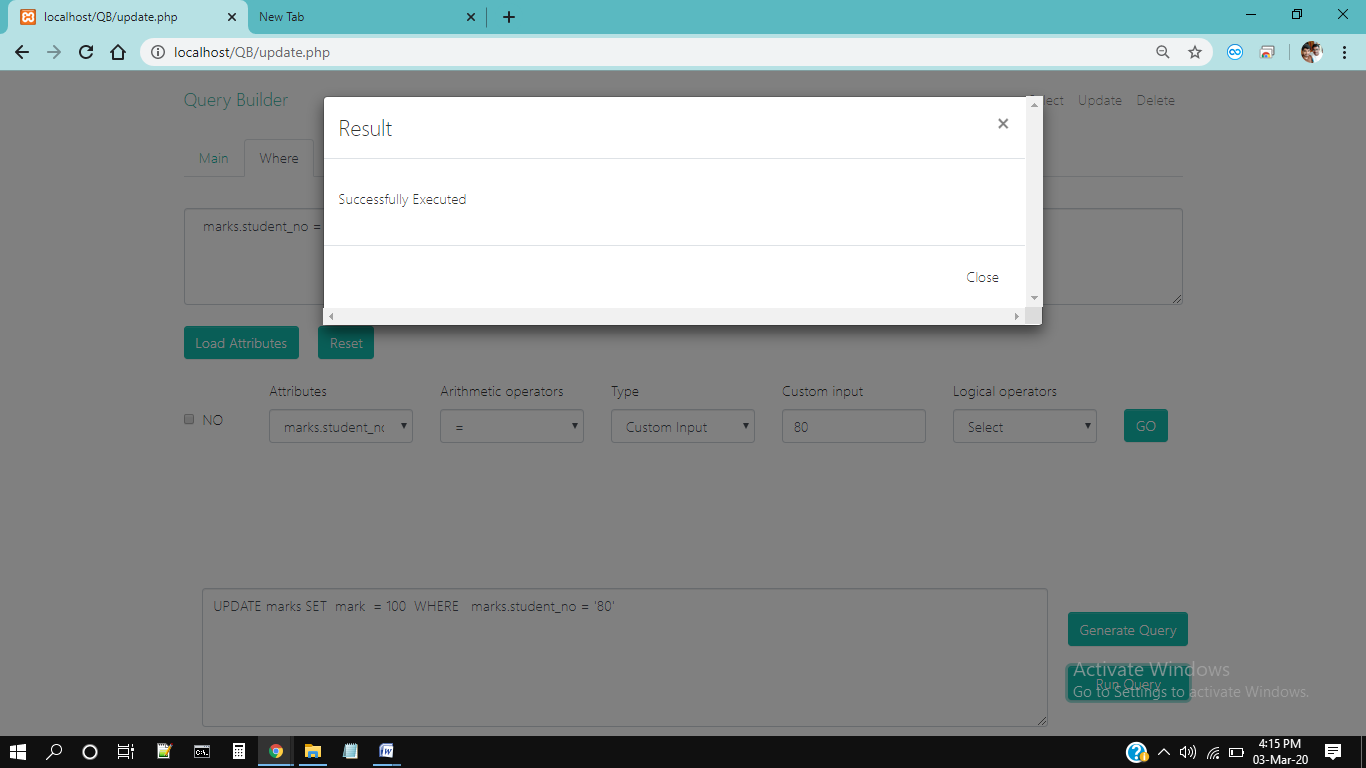
****

Fig.6.13: Running Update Query

**Delete Page:**

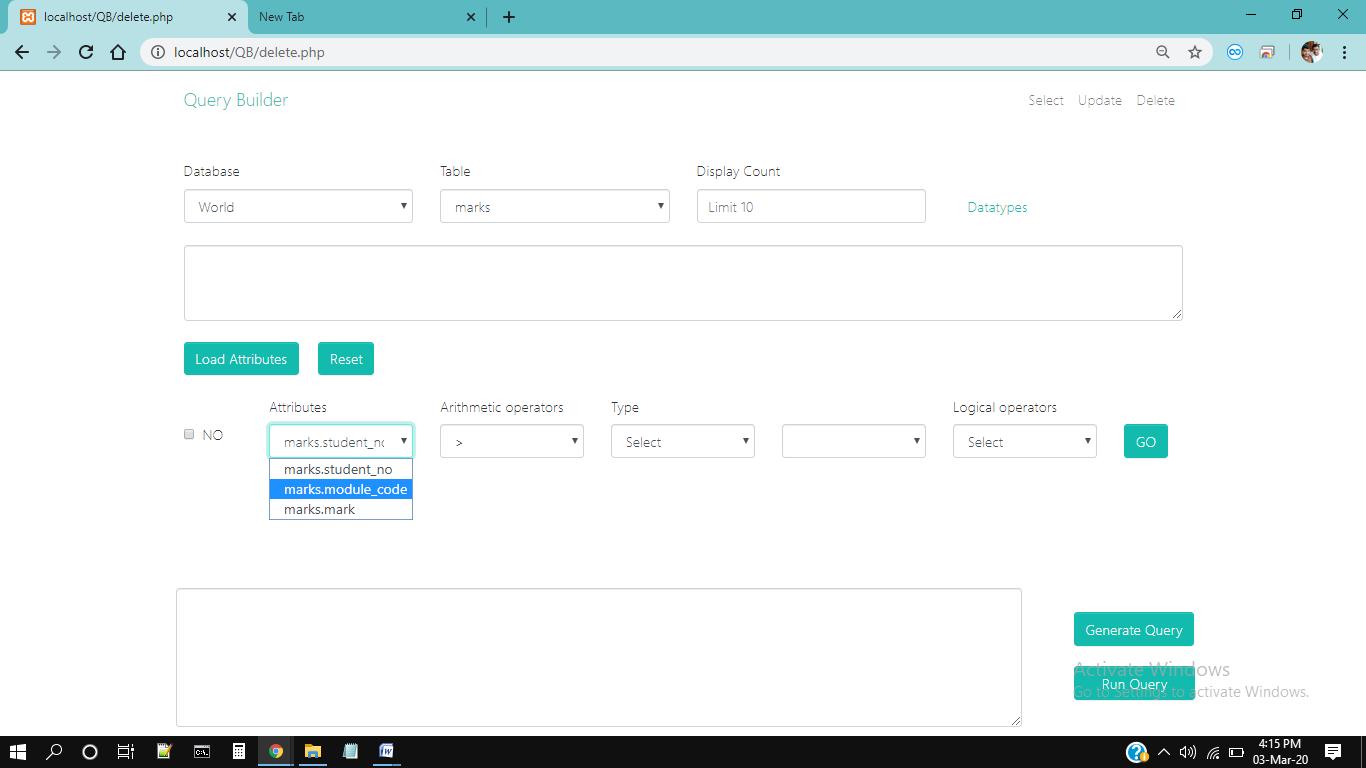
****

Fig.6.14: Selecting Delete Commands

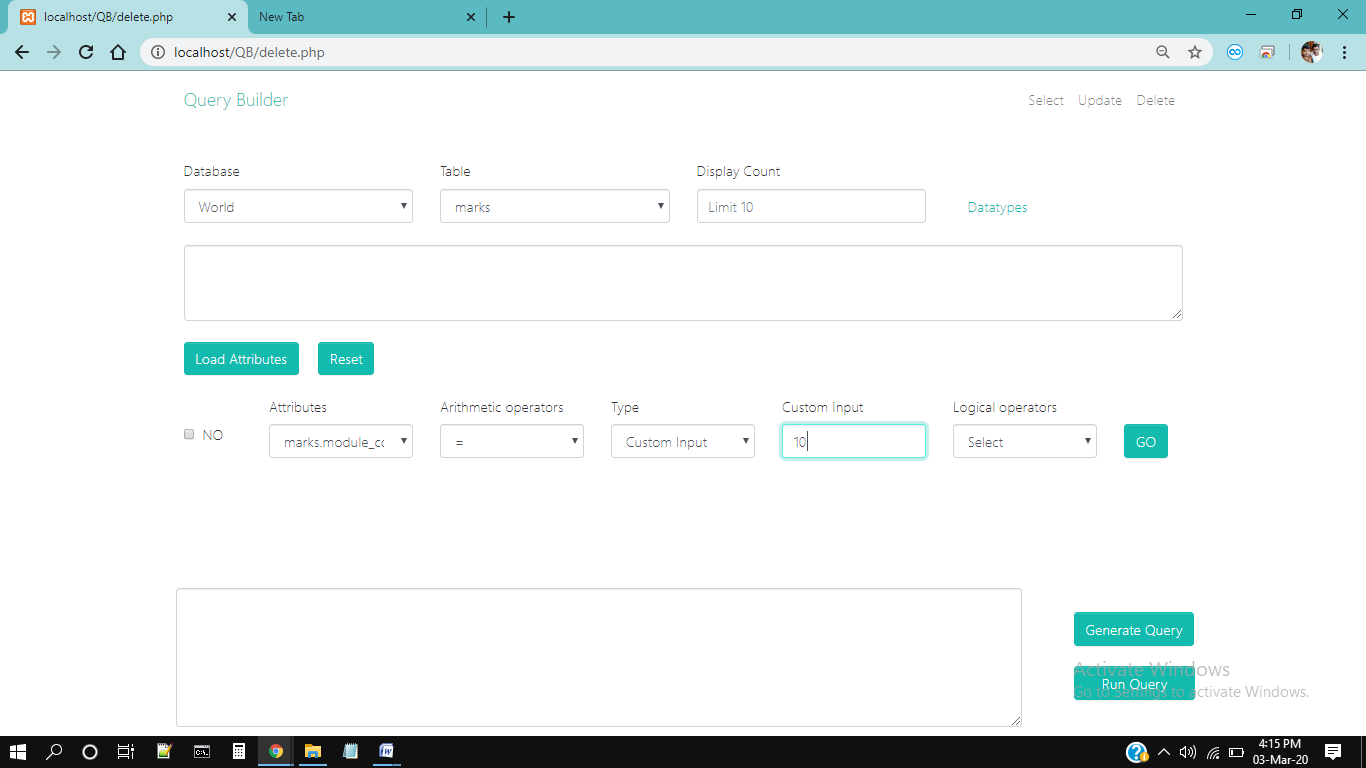
****

Fig.6.15: Selecting Delete Commands Values

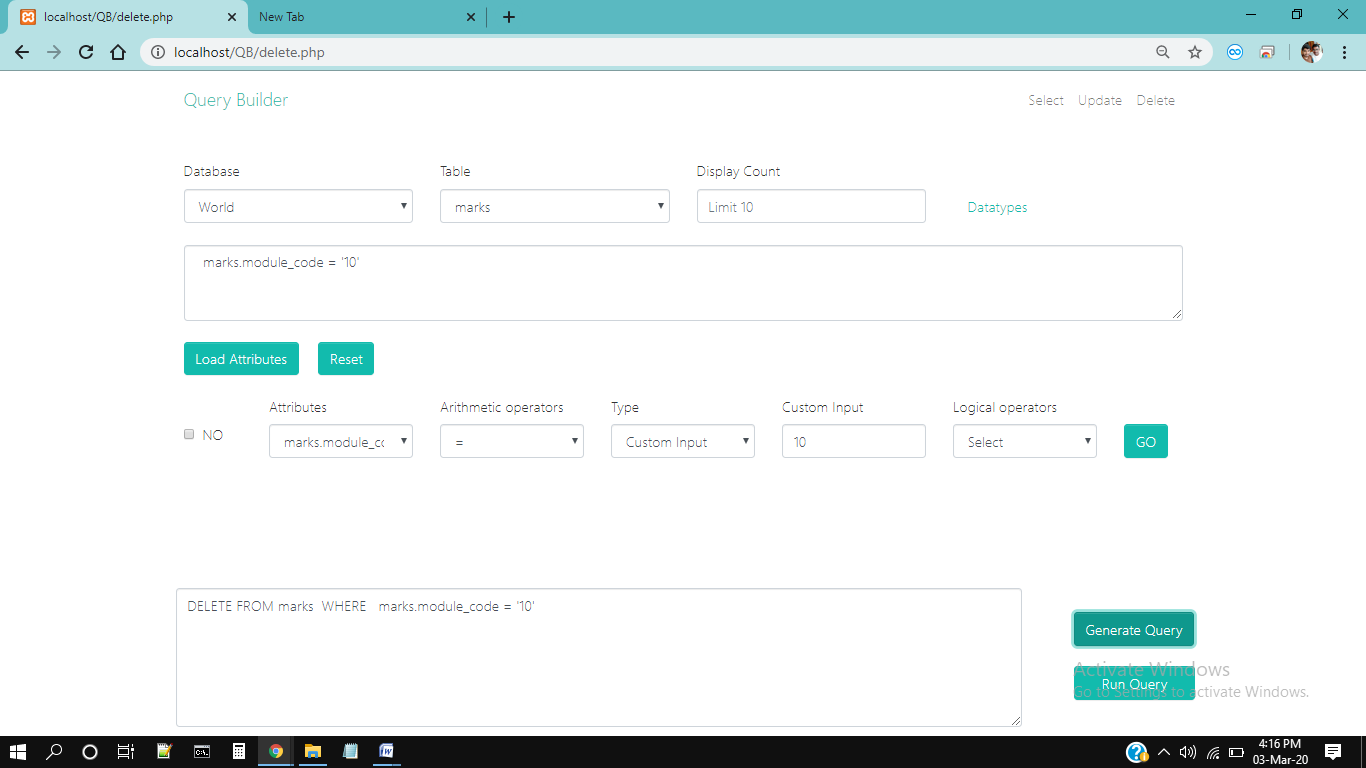
****

Fig.6.16: Generating Delete Query

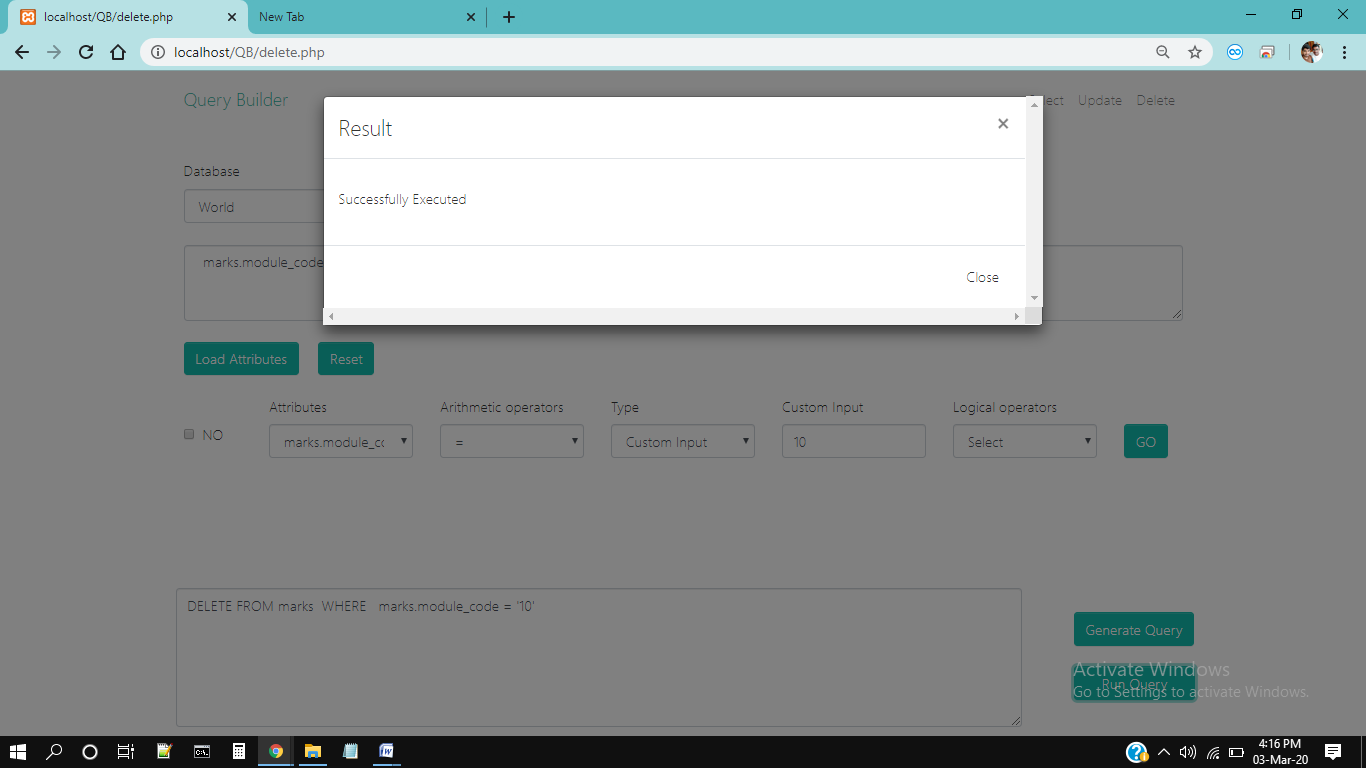
****

Fig.6.17: Running Query

**CHAPTER 7: CONCLUSION**

**7. CONCLUSION**

While developing the system a conscious effort has been made to create and this is developed a package, making use of available tools, techniques and resources – that would generate a proper System. While making the system, any has been kept on making it as user-friendly, as cost-effective and as flexible as possible. As such one may hope that the system will be acceptable to any user and will adequately meet his/her needs. As in case of any system development processes where there are a number of shortcomings, there have been some shortcomings in the development of this system also. The project is still under modification.

**CHAPTER 8: FUTURE SCOPE**

**8. FUTURE SCOPE**

The scope of the project includes that what all future enhancements can be done in this system to make it more feasible to us:-

* Different Type Of SQL Commands Like DDL, DQL, DCL can be added
* Multilingual support can be provided so that it can be understandable by the person of any language.
* More graphics can be added to make it more user-friendly and understandable.
* Manage & backup versions of documents online.

**BIBLIOGRAPHY**

**BIBLIOGRAPHY**

**Reference links:**

1. [**https://www.w3schools.com**](https://www.w3schools.com)
2. [**https://stackoverflow.com/**](https://stackoverflow.com/)
3. [**https://www.javatpoint.com**](https://www.javatpoint.com)
4. [**https://www.geeksforgeeks.org**](https://www.geeksforgeeks.org)

**Reference books:**

1. HTML & CSS: The Complete Reference, Fifth Edition Book By Thomas Powell
2. PHP: The Complete Reference Book By Steven Holster
3. My SQL: The Complete Reference Book By Vikram Vaswani
4. HTML5 Pocket Reference: Quick, Comprehensive Book ByJenniferNeediest
5. Software Testing, 2nd Edition, 2005 By Author: Ron Patton
6. The art of project management Book by Scott Berkun
7. Software Engineering: A Practitioner's Approach Book by Roger S. Pressman