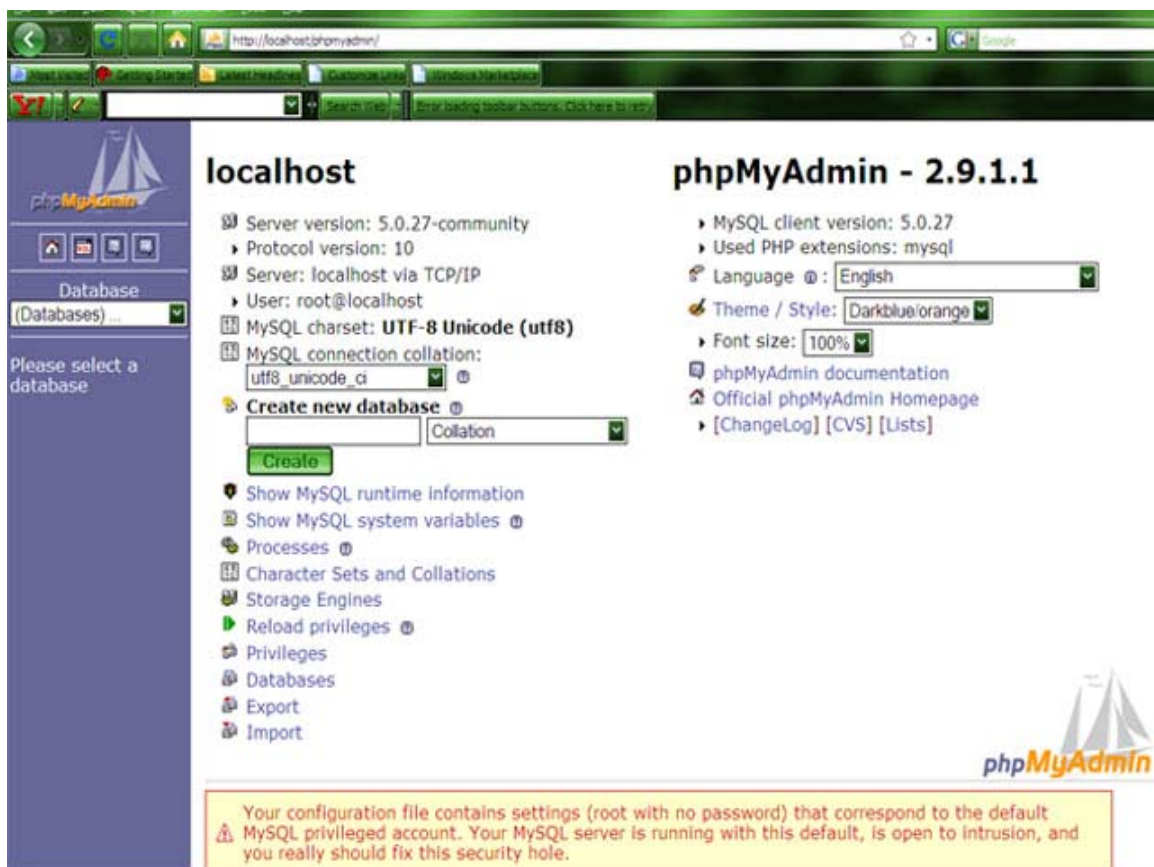


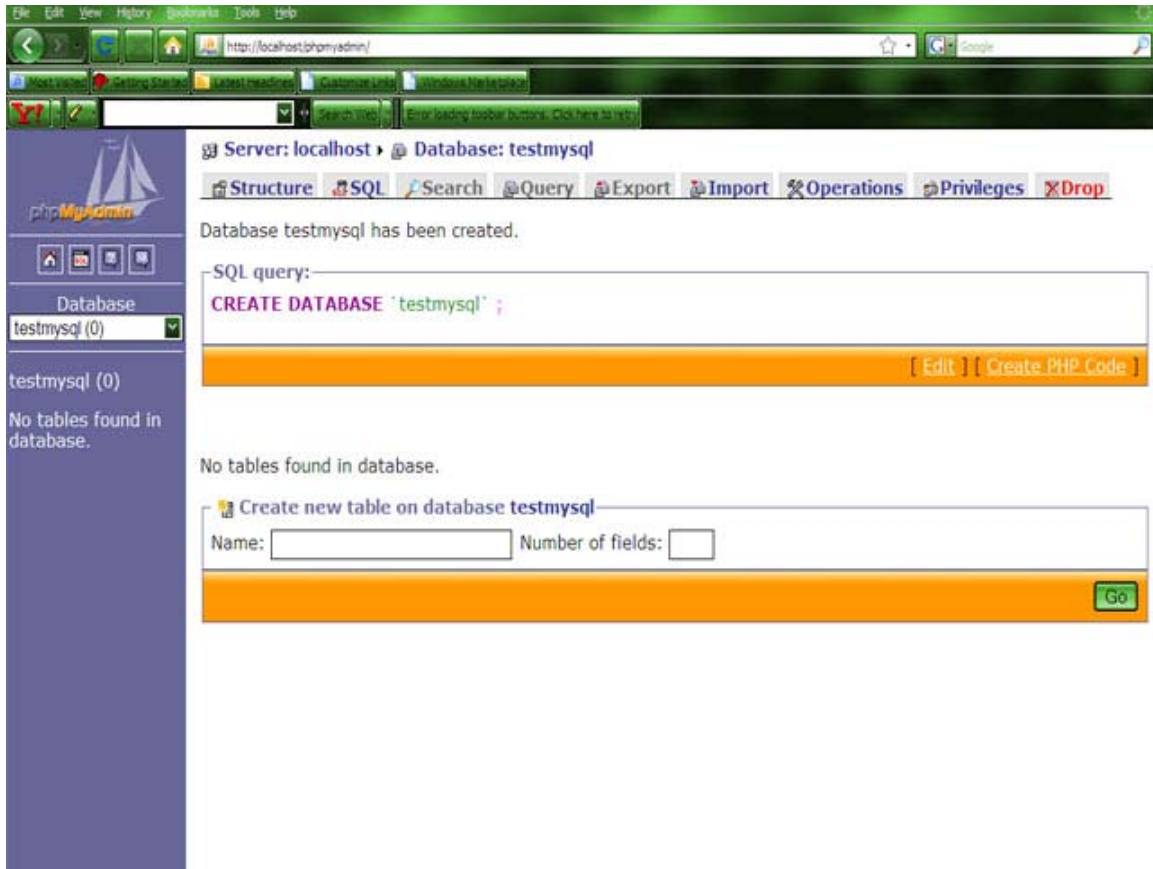
ဒီ MySQL ကိုကျွန်တော်နားလည်သလို သိရှိသလိုရေးသားထားခြင်းဖြစ်ပါသည်။
များသောအားဖြင့် အင်တာနက် ထဲမှကိုးကားပြီးရေးထားခြင်းဖြစ်ပါသည်။
ကျွန်တော်သိသလောက်ရေးထားခြင်းဖြစ်သဖြင့် လိုအပ်သည်များရှိပါက
မှားယွင်းသည်များရှိပါက နားလည်ပေးစေလိုပါသည်။

MySQL

ပထမဆုံး MySQL Run နည်းအရင်ပြောပါမည်။Xamp_Start ကိုအရင်ဖွင့်လိုက်ပါ။(Xamp_Start လုပ်နည်းကို ကျွန်တော့်၏ PHP မြန်မာလိုမှာရှင်းပြထားပြီးသားပါ)ဟုတ်ကဲ့ ပြီးရင် Browser ဖွင့်ပြီး localhost/phpmyadmin ဆိုပြီးရိုက်လိုက်ပါ။



အခုလိုပေါ်လာပါလိမ့်မည်။ပြီးရင်တော့ Creat new database ဆိုတဲ့အောက်က Textbox လေးမှာ Database နာမည်ရေးပြီး Create ကိုနှိပ်လိုက်ပါ။ database name ကို အခုတော့ testmysql ဆိုပြီးပေးလိုက်ရအောင်။



အခုလိုဖြစ်သွားပါမည်။ပြီးရင်ကျွန်တော်တို့ **Table** တစ်ခုတည်ဆောက်လိုက်ရအောင် **Name** မှာကိုယ်လုပ်ချင်တဲ့ နာမည်ဖြည့်ပြီး **Number of fields** မှာတော့ **Table** မှာပါဝင်ချင်တဲ့ **field** အရေအတွက်ကိုဖြည့်လိုက်ပါ။အားလုံး ပြီးရင်တော့ **Go** ဆိုတာလေးကိုနှိပ်လိုက်ပါ။ဒါဆိုရင်တော့ **Table** တစ်ခုတည်ဆောက်ပြီးသားဖြစ်သွားပါပြီ။ ကျွန်တော်ကတော့ **Table Name** ကို သင်တန်းတွေမှာပေးခိုင်းနေကျထုံးစံအတိုင်း **Student** ဆိုပြီး ပေးလိုက်ပါတယ်။**Field** ကတော့ ၃ ခုပေးလိုက်ပါတယ်။မှာချင်တာက ကိုယ်ပေးလိုက်တဲ့ **Database Name, Table Name, Field Name** တွေကိုစာလုံးပေါင်း အကြီးအသေး သေချာမှတ်ထားစေချင်ပါတယ်။သို့မှသာ **Data** တွေဖြည့်ရာမှာအမှားအယွင်းနည်းမှာဖြစ်ပါတယ်။ အောက်ကပုံအတိုင်းပေါ်ပါလိမ့်မည်။အခု **Field name** တွေဖြည့်လိုက်ရအောင်။

1. ID
2. Name
3. Subject

ဖြည့်လိုက်ပါ။ပြီးရင် **ID** ကို **Primary Key** ပေးလိုက်ရအောင်။ပုံ (၂) ကိုကြည့်ပါ။

Server: localhost Database: testmysql Table: Student

Field	Type	Length/Values ¹	Collation	Attributes	Null
	VARCHAR				not null
	VARCHAR				not null
	VARCHAR				not null

Table comments: Storage Engine: MyISAM Collation:

Save Or Add 1 field(s) Go

¹ If field type is "enum" or "set", please enter the values using this format: 'a','b','c'...
If you ever need to put a backslash ("\") or a single quote (") amongst those values, precede it with a backslash (for example '\\xyz' or 'a\\b').

² For default values, please enter just a single value, without backslash escaping or quotes, using this format: a

³ Please enter the values for transformation options using this format: 'a','b','c'...
If you ever need to put a backslash ("\") or a single quote (") amongst those values, precede it with a backslash (for example '\\xyz' or 'a\\b').

For a list of available transformation options and their MIME type transformations, click on [transformation descriptions](#)

0-3

Extra						Comments	MIME type	Browser transformation

0-J

Extra ဘေးမှာ radio button ပုံလေးတွေတွေ့ကြမှာပါ ပထမဆုံးက Primary ပါ အဲ့ Key အပေါ်မှာ mouse pointer ထောက်ကြည့်ရင်ဘာဆိုတာပေါ်လာပါလိမ့်မည်။ဟုတ်ကဲ့အခု Primary key လုပ်လိုက်ရအောင်။ Radio button လေးကို နှိပ်လိုက်ပါ။ကိုယ်ပေးချင်တဲ့ field နေရာမှာနော်။အခုကျွန်တော် Type ကိုရှင်းပြပါမယ် Field ဆိုတဲ့ခေါင်းစဉ်လေးဘေးနားမှာပါ။

MySQL Data Type

MYSQL FIELD TYPE	DESCRIPTION
bigint(length)	Numeric field that stores integers from -9223372036854775808 to 9223372036854775807. (Adding the unsigned parameter allows storage of 0 to 18446744073709551615.) The parameter length limits the number of characters to be displayed. Please refer to the manual at http://www.mysql.com before performing mathematical calculations involving the bigint type.
bit	Equal to tinyint(1) field. A value of zero represents false; a non-zero value represents true.
blob	Equal to a text field, except it is case-sensitive when sorting and comparing. Stores up to 65535 characters.
bool	Equal to tinyint(1) field. A value of zero represents false; a non-zero value represents true.
boolean	Equal to tinyint(1) field. A value of zero represents false; a non-zero value represents true.
char(length)	Any characters can be in this field, but the field will have a fixed length. The length parameter can be between 0 and 255. If the length parameter is not defined, the default value is 1. Adding the BINARY attribute will make comparisons and sorting results case-sensitive.
date	Stores a date as yyyy-mm-dd. Allows values from 1000-01-01 to 9999-12-31.
datetime	Stores date and time as yyyy-mm-dd hh:mm:ss. Allows values from 1000-01-01 00:00:00 to 9999-12-31 23:59:59.

MYSQL FIELD TYPE	DESCRIPTION
dec(length,dec)	Equal to decimal field.
decimal(length,dec)	Numeric field that can store decimals. The length parameter limits the number of digits that will be displayed, and the dec parameter limits the number of decimal places that can be stored. The length parameter does not count decimal points and “-” for negative values. A price field that would store prices up to 999.99, for example, would be defined as decimal(5,2). If the length parameter is not specified, the default is 10. Adding the unsigned attribute allows only non-negative values.
double(length,dec)	A medium-sized floating point number that stores values from -1.7976931348623157E+308 to -2.2250738585072014E-308, 0, and 2.2250738585072014E-308 to 1.7976931348623157E+308. The length parameter determines how many digits will be displayed; the dec parameter determines how many decimal places are displayed. (Adding the unsigned parameter allows only positive numbers to be stored.) Using double without parameters specified represents a double-precision floating point number.
double precision(length, dec)	Equal to double
enum(“option1”, “option2”, ...)	Allows only certain values to be stored in this field, such as true and false , or a list of states. 65,535 different options can be allowed
fixed(length,dec)	Equal to decimal field.
float(length,dec)	A small floating point number that stores values from -3.402823466E+38 to -1.175494351E-38, 0, and 1.175494351E-38 to 3.402823466E+38. The length parameter determines how many digits will be displayed; the dec parameter determines how many decimal places are displayed. (Adding the unsigned parameter allows only positive numbers to be stored.)
float(precision)	Equal to float(length,dec) except the length and dec parameters are undefined. To be used with a true floating point number. (Adding the unsigned parameter allows only positive numbers to be stored.) The precision parameter can be from 0 to 24 for single-precision floating point numbers, and from 25-35 for double-precision floating point numbers.
int(length)	Numeric field that stores integers from -2147483648 to +2147483647, but can be limited with the length parameter. length limits the number of digits that can be shown, not the value. Mathematical functions can be performed on data in this field. Signifying the unsigned parameter permits positive integers (and zero) up to 4294967295.
integer(length)	Equal to int.
longblob	Equal to longtext except it is case-sensitive when sorting and comparing.

MYSQL FIELD TYPE	DESCRIPTION
longtext	Allows storage of up to 4294967295 characters
mediumblob	Equal to mediumtext field except it is case-sensitive when sorting and comparing.
mediumint(length)	Numeric field that stores integers from -8388608 to 8388607. (Adding the unsigned parameter allows storage of 0 to 16777215.) length limits the number of digits to be displayed
mediumtext	Allows storage of up to 16777215 characters
numeric(length,dec)	Equal to decimal field.
real(length,dec)	Equal to double field.
set("option1", "option2", ...)	Similar to enum field, but with set there can be none or more than one of the available options. Set allows up to 64 options.
smallint(length)	Numeric field that stores integers from -32768 to 32767. (Adding the unsigned parameter allows storage of 0 to 65535.) The length parameter limits the number of characters to be displayed.
text	Any character can be in this field, and the maximum size of the data is 64K (65535 characters).
time	Stores time as hh:mm:ss.
timestamp	Stores date and time as yyyy-mm-dd hh:mm:ss. Useful for automatically capturing current date and time.
tinyblob	Equal to tinytext field, except it is case-sensitive when sorting and comparing.
tinyint(length)	Numeric field that stores integers from -128 to 127. (Adding the unsigned parameter allows storage of 0 to 255.) The length parameter limits the number of characters to be shown.
tinytext	Allows storage of up to 255 characters.
varchar(length)	Any character can be in this field, and the data can vary from 0 to 255 characters. Maximum length of field is denoted with the length parameter. Adding the BINARY attribute causes comparisons and sorting results to be case-sensitive
year(length)	Stores a year in 4-character format (by default). In this format, values from 1901 to 2155, and 0000 are acceptable. It is possible to specify a 2-year format by signifying so with the length parameter. In this format, the values from 70 to 69 are acceptable (1970-2069).

Data Type နဲ့ Description ကိုမြန်မာလိုမပြန်တော့ဘူး။မူရင်းအတိုင်းပဲတင်ပေးလိုက်ပါတယ်။ကျွန်တော်လည်း အကုန်မသိပါဘူး။ကဲ..အခု Type မှာ ID ကို Tinyint အနောက်က Length မှာတော့ 4 လို့ဖြည့်လိုက်ပါတယ်။ Length ဆိုတာကဒီ Field ရဲ့ကန့်သတ်ချက်ကိုပြောတာပါ အခု ၄ ဆိုတော့ ၈၈၈၈ ၄ လုံးပဲထားတာပါ။ Name နဲ့ Subject ကို Varchar ပဲထားပြီး Length ကို 20 စီထားလိုက်ပါ။ပြီးရင်တော့ Save Button နှိပ်လိုက်ပါ။ ဒါဆို Table တစ်ခုဖန်တီးပြီးသွားပြီဖြစ်ပါသည်။

Table Student has been created.

SQL query:

```
CREATE TABLE `Student` (
  `ID` TINYINT(4) NOT NULL,
  `Name` VARCHAR(20) NOT NULL,
  `Subject` VARCHAR(20) NOT NULL,
  PRIMARY KEY (`ID`)
) ENGINE = MYISAM;
```

Field	Type	Collation	Attributes	Null	Default	Extra	Action
<input type="checkbox"/> ID	tinyint(4)			No			
<input type="checkbox"/> Name	varchar(20)	latin1_general_ci		No			
<input type="checkbox"/> Subject	varchar(20)	latin1_general_ci		No			

Check All / Uncheck All With selected:

Print view Relation view Propose table structure

Add 1 field(s) At End of Table At Beginning of Table After ID Go

Indexes:					Space usage		Row Statistics	
Keyname	Type	Cardinality	Action	Field	Type	Usage	Statements	Value
PRIMARY	PRIMARY	0		ID	Data	0 B		
Create an index on 1 columns Go					Index	0 B		
					Total	0 B		

အခုလိုရသွားပါလိမ့်မည်။ပြင်ချင် ဖျက်ချင်ရင် ဒီမှာပဲပြင်ဖျက်လို့ရပါတယ်။Field ဖျက်ချင်ရင်ကိုယ်ဖျက်ချင်တဲ့ Field ရှေ့က Check Box လေးကို Click ပြီးအောက်နားက အမှားပုံလေးကိုနှိပ်လိုက်ရင်ရပါတယ်။field အသစ် ထပ်ထည့်ချင်ရင်တော့ Add ဆိုတဲ့ဘေးနားက Text box လေးမှာထပ်ထည့်ချင်တဲ့အရေအတွက်ကိုဖြည့်ပါ။ At end of table , At Beginning of table ဒါမှမဟုတ် ဘယ် field ပြီးရင်ထည့်ချင်လည်းဆိုတာကို After ဘေးနားက Box မှာရှိပြီးသား field တွေအကုန်လုံးရှိပါတယ်။ကိုယ်ထားချင်တဲ့ field ကိုရွေးလိုက်ပါ။ အားလုံးပြီးရင်တော့ Go နှိပ်လိုက်ပါ။ဒါဆို Field Name ပေးတဲ့နေရာပြန်ရောက်သွားမှာပါ။လုပ်ထားပြီးသား Field တွေကိုပြန်ပြင်ချင်တယ်ဆိုရင်တော့ Check Box ကို on ပြီးခဲတံပုံလေးတွေကြမှာပါ အဲဒါကိုနှိပ်ပြီး ပြန်ပြင်ဆင်နိုင်ပါတယ်။ဒါကတော့ Database နဲ့ Table တည်ဆောက်ပုံပါ။ အားလုံးပဲ ပြန်စမ်းကြည့် ကြပါ။ဆက်ပြီးတော့ Data ဖြည့်တဲ့အပိုင်းကိုပြောပြမှာပါ။

MYSQL WITH PHP

အရင်ဆုံး `server` ကို `connect` လုပ်ရပါမယ်။ဒီလိုရေးပါတယ်။

```
<?
$servername="localhost";
$username="root";
$password="";
$con=mysql_connect($servername,$username,$password);
if(!$con)
{
    die("Could not connect:".mysql_error());
}
print "connection established";

?>
```

ပြီးရင်တော့ `connect.php` ဆိုပြီး `save` လုပ်လိုက်ပါ။`save` ကိုတော့ထုံးစံအတိုင်း `htdocs` အောက်မှာ လုပ်ရပါမယ် `Folder` တစ်ခုလုပ်လိုက်ပါမယ် နာမည်ကိုတော့ `testmysql` ဆိုပြီးပေးလိုက်ပါတယ်။ဒီ `xampp server` မှာတော့ `localhost,root` တို့ကပုံသေပါပဲ။တကယ် `hosting` မှာတော့ `username,password` ကကိုယ် `server` လုပ်တဲ့ `username,password` ပါပဲ `servername` ကတော့ကိုယ့် `hosting` မှာတစ်ခါတည်းပေးပြီးသား ရှိပါတယ်။အဲ့ဒါကိုပြောင်းပေးလိုက်ရုံပါပဲ။`connect.php` ဆိုပြီးသပ်သပ်မထုတ်ဘဲ `insert` လုပ်ရာမှာတစ်ခါတည်း ပေါင်းရေးရင်လည်းရပါတယ်။အခု `code` ကိုရှင်းပြပါမယ်။`Variable` တစ်ခုစီကြော်ငြာပြီး `localhost` တို့ကိုမထည့်ဘဲရေးလို့လည်းရပါတယ်။ဒီလိုရေးပါတယ်။

```
<?
$con=mysql_connect("localhost","root","");
if(!$con)
{
    die("Could not connect:".mysql_error());
}
print "connection established";

?>
```

အခုလိုဆိုလည်းရပါတယ်။`mysql_connect` ဆိုတာက `connection` လုပ်တဲ့ `function` ပါ။ပြီးရင် `Run` ကြည့် လိုက်ပါ မှန်တယ်ဆို `Connection Established` ဆိုပြီးပေါ်လာပါလိမ့်မည်။ဆက်ပြီး `insert` လုပ်လိုက်ရအောင်။


```
<?
include 'connect.php';
if(isset($_POST['reg']))
{
    $id=$_POST['id'];
    $name=$_POST['sname'];
    $s=$_POST['sub'];

    $database="testmysql";
    mysql_select_db($database);

    mysql_query("INSERT INTO student(ID,Name,Subject)
    VALUES('$id','$name','$s')");

}

?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Untitled Document</title>
</head>

<body>
<form id="form1" name="form1" method="post" action="">
<table width="367" border="0">
<tr>
<td colspan="2">Student Form </td>
</tr>
<tr>
<td width="119">ID</td>
<td width="238"><label>
<input name="id" type="text" id="id" />
```

```

        </label> </td>
    </tr>
    <tr>
        <td>Name</td>
        <td><label>
            <input name="sname" type="text" id="sname" />
        </label> </td>
    </tr>
    <tr>
        <td>Subject</td>
        <td><label>
            <input name="sub" type="text" id="sub" />
        </label> </td>
    </tr>
    <tr>
        <td colspan="2"><label>
            <input name="reg" type="submit" id="reg" value="Register" />
        </label> </td>
    </tr>
</table>
</form>
</body>
</html>

```

insert.php ဆိုပြီး save လုပ်လိုက်ပါ။ပြီးရင် Field တွေမှာဖြည့်လိုက်ပါ။

e.g

ID=1

Name=AungAung

Subject=Web

```
$database="testmysql";
```

```
mysql_select_db($database);
```

ထိုသို့ မရေးဘဲ အခုလိုရေးလို့လည်းရပါတယ်။

```
mysql_select_db("testmysql");
```

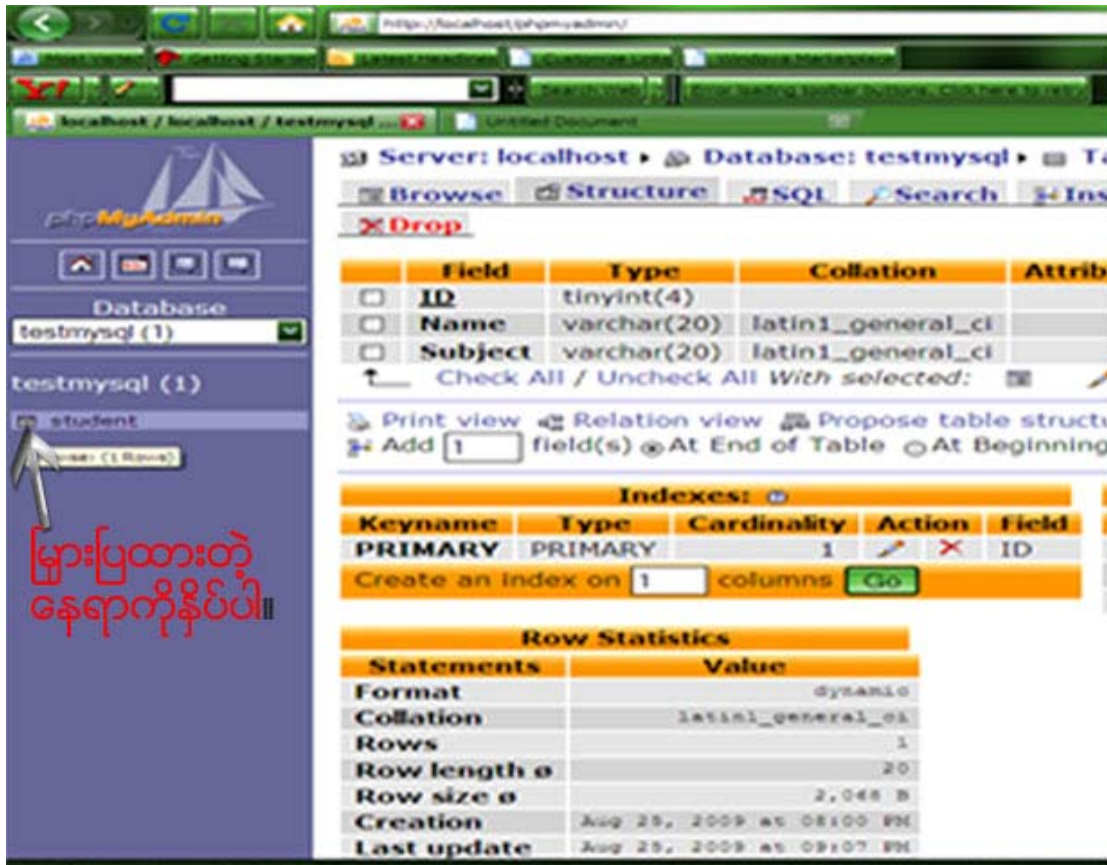
ဒါက insert လုပ်မယ်ဆိုတိုင်းဒီလိုရေးရပါတယ်။

```
mysql_query("INSERT INTO table_name(table_fields_name)
VALUES(Value)");
```

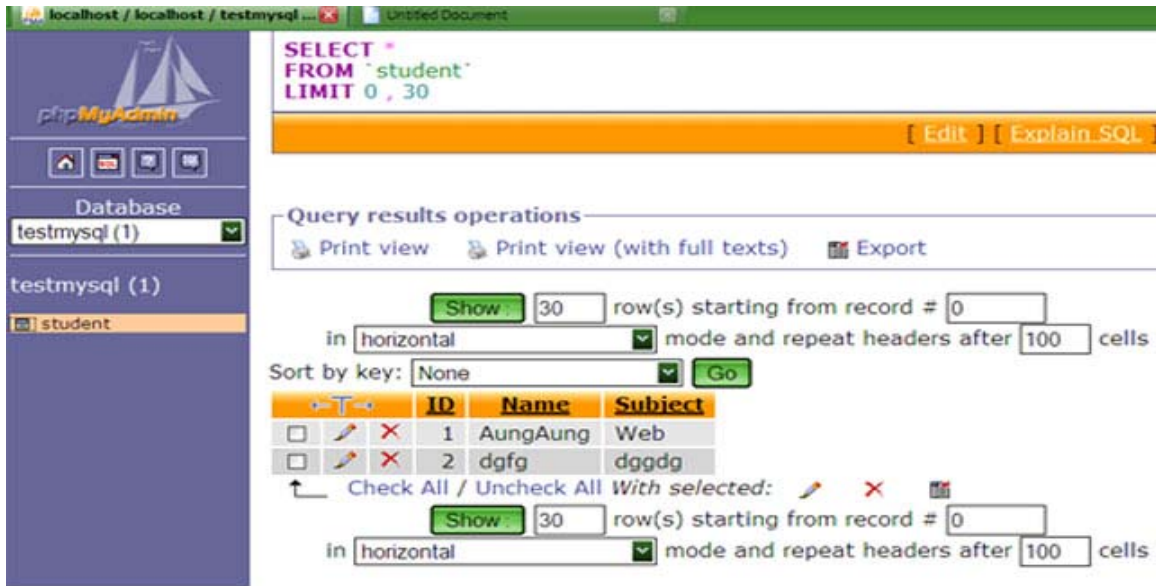
```
mysql_query("INSERT INTO student(ID,Name,Subject)
```

VALUES('\$id','\$name','\$s'));

အထူးသတိထားရမှာက နာမည်ပေးတာတွေမှန်ကန်ရေးပါပဲ။နည်းနည်းလေးအကြီး အသေးလွဲနေရင်လည်း မရပါဘူး။ကျွန်တော်စ သင်တန်းကဆိုဖြစ်တယ်။code ကမှန်တယ်။ဒါပေမယ့် data ကမဝင်ဘူးဘာဖြစ်တာလဲ ဆိုပြီး error ရှာတာမျိုးကိုပြန်ရော။နောက်တော့ နာမည် အကြီး၊အသေးတွေလွဲနေတာလေ။ဒါကြောင့်ဂရုစိုက် ပြီးရေးသင့်ပါတယ်။နောက်တစ်ခုက insert လုပ်တာရေးရာမှာ (` `) လွဲတာပါပဲ။ဒါလည်းသတိထားပါ။ Data ဝင်မဝင်ကြည့်မယ်ဆိုရင်ဒီပုံလေးကိုကြည့်ပါ။



ပုံထဲမှာပြထားတဲ့နေရာမှာနှိပ်ပါ။student ရှေ့က Table ပုံလေးမှာပေါ့။အဲ့ဒါက data ဝင်မဝင်ကြည့်တာပါ။ ဝင်တယ်ဆိုဒီလိုပြပါလိမ့်မယ်။



အဲဒါဆိုရင်တော့ **data** ဝင်သွားပြီဖြစ်ပါတယ်။မဟုတ်ရင်တော့ လွဲနေပြီဖြစ်ပါတယ်။သေချာပြန်စစ်ကြည့်ပါ။
ဟုတ်ပါပြီ။အခုကျွန်တော်တို့ရှိပြီးသား **data** တွေပြန်ကြည့်တဲ့ **code** ဆက်ရေးလိုက်ရအောင်။
ဒီလိုရေးပါမယ်။

```
<?php
$con = mysql_connect("localhost","root","");
if (!$con)
{
    die('Could not connect: ' . mysql_error());
}
mysql_select_db("testmysql", $con);
$result = mysql_query("SELECT * FROM Student");
while($row = mysql_fetch_array($result))
{
    echo $row['ID'] . " " . $row['Name'] . " " . $row['Subject'];
    echo "<br />";
}
mysql_close($con);
?>
```

ပြီးရင် **select.php** ဆိုပြီး **save** မှတ်လိုက်ပါ။**Run** ကြည့်ပါ။**mysql_query** ဆိုတဲ့ **function** သုံးပြီး

Select * From Table_Name နဲ့ ဆွဲထုတ်လိုက်ပါတယ်။ပြီးရင် while loop နဲ့ mysql_fetch_array သုံးလိုက်ပါတယ်။ while loop သုံးရခြင်းမှာ ရှိရှိသမျှကြည့်မယ်ဆိုတဲ့သဘောဖြစ်ပါတယ်။mysql_fetch_array က PHP မှာပါတဲ့ Array သဘောလိုပဲ။ဟုတ်ပါပြီဒီတစ်ခါ where ကိုသုံးပြီး select လုပ်ပြပါမယ်။data တွေ ၃ ၄ ခုလောက်ထပ်ဖြည့်လိုက်ပါ။ပြီးရင်တော့ ဒီ code ကိုရေးပါ။

```
<?php
$con = mysql_connect("localhost","root","");
if (!$con)
{
    die('Could not connect: ' . mysql_error());
}
mysql_select_db("testmysql", $con);
$query="SELECT * FROM Student WHERE Name='AungAung'";
$result=mysql_query($query);
$row=mysql_fetch_array($result);
echo
    "{$row['Name']} <br>" ;

mysql_close($con);

?>
```

သူက Where ဆိုပြီး ကန့်သတ်လိုက်တာပါ။Name field ထဲကမှ AungAung ဆိုတာပဲပြပါဆိုပြီးရေးထားတာပါ ကျန်တာတော့အတူတူပဲ။ Where လိုမျိုးအခြားသုံးလို့ရတာတွေအနည်းငယ်ဖော်ပြပေးလိုက်ပါတယ်။

NAME	DESCRIPTION
BETWEEN...AND....	Check whether a value is within a range of values
COALESCE()	Return the first non-NULL argument
<=>	NULL-safe equal to operator
=	Equal operator
>=	Greater than or equal operator
>	Greater than operator
GREATEST()	Return the largest argument
IN()	Check whether a value is within a set of values
INTERVAL()	Return the index of the argument that is less than the first argument
IS NOT NULL	NOT NULL value test
IS NOT	Test a value against a boolean
IS NULL	NULL value test
IS	Test a value against a boolean
IS NULL()	Test whether the argument is NULL
LEAST()	Return the smallest argument
<=	Less than or equal operator

<	Less than operator
LIKE	Simple pattern matching
NOT BETWEEN...AND...	Check whether a value is not within a range of values
!= or <>	Not equal operator
NOT IN()	Check whether a value is not within a set of values
NOT LIKE	Negation of simple pattern matching
STRCMP	Compare two strings

Example တွေကိုတော့မူရင်းအတိုင်းဖော်ပြပေးလိုက်ပါတယ်။တစ်ခြားဟာတွေလည်းအများကြီးရှိပါသေးတယ်။
ကိုယ့်ဘာကို [Google](http://www.google.com) နဲ့ရှာကြည့်ဖို့တိုက်တွန်းပါရေစေ။www.hsscripts.com မှာလည်းရှိပါတယ်။

Show all data in a table.

```
mysql> SELECT * FROM [table name];
```

Returns the columns and column information pertaining to the designated table.

```
mysql> show columns from [table name];
```

Show certain selected rows with the value "whatever".

```
mysql> SELECT * FROM [table name] WHERE [field name] = "whatever";
```

Show all records containing the name "Bob" AND the phone number '3444444'.

```
mysql> SELECT * FROM [table name] WHERE name = "Bob" AND phone_number = '3444444';
```

Show all records not containing the name "Bob" AND the phone number '3444444' order by the phone_number field.

```
mysql> SELECT * FROM [table name] WHERE name != "Bob" AND phone_number = '3444444'  
order by phone_number;
```

Show all records starting with the letters 'bob' AND the phone number '3444444'.

```
mysql> SELECT * FROM [table name] WHERE name like "Bob%" AND phone_number =  
'3444444';
```

Show all records starting with the letters 'bob' AND the phone number '3444444' limit to records 1 through 5.


```
mysql> SELECT * FROM [table name] WHERE name like "Bob%" AND phone_number = '3444444' limit 1,5;
```

Use a regular expression to find records. Use "REGEXP BINARY" to force case-sensitivity. This finds any record beginning with a.

```
mysql> SELECT * FROM [table name] WHERE rec RLIKE "^a";
```

Show unique records.

```
mysql> SELECT DISTINCT [column name] FROM [table name];
```

Show selected records sorted in an ascending (asc) or descending (desc).

```
mysql> SELECT [col1],[col2] FROM [table name] ORDER BY [col2] DESC;
```

Return number of rows.

```
mysql> SELECT COUNT(*) FROM [table name];
```

Sum column.

```
mysql> SELECT SUM(*) FROM [table name];
```

Database Create လုပ်ချင်ရင်ဒီဟာကိုသုံးပါတယ်။

Syntax: create[db_name];

e.g

create User;

SQL Server အတွင်းရှိတဲ့ **database** တွေကိုကြည့်ချင်တယ်ဆိုရင်တော့

Syntax: show database;

Database တွေကို **Switch** လုပ်ရာမှာဒီလိုရေးပါတယ်။

Syntax: use[db_name];

ဒါကတော့ **database** တွေကိုဖျက်တာပါ။

Syntax: drop[db_name];

Table တွေကိုဖျက်ရာမှာလည်း **drop** ကိုပဲသုံးပါတယ်။

drop table[table_name];

UPDATE <table_name>

SET <column_name> = 'new_value'

WHERE (<column_name> = 'some_value');

ဒီကောင်ကိုဘယ်လိုမျိုးမှာသုံးလဲဆိုတော့ **Vote** လုပ်တဲ့နေရာမျိုးမှာသုံးပါတယ်။ရိုပြီးသား **data** တွေကို ပြင်တာပါ။ပိုရှင်းသွားအောင် **e-book** ထဲမှာရေးထားတဲ့အတိုင်းဖော်ပြပေးလိုက်ပါတယ်။

COMMAND	DESCRIPTION
CREATE databasename	Creates the database.
CREATE tablename (field1, field2, field3, and so on PRIMARY KEY(field))	Creates a new table.
ALTER TABLE tablename WHERE condition	Modifies a table in the database.
RENAME TABLE oldtablename TO newtablename	Renames a table in the database.
INSERT INTO tablename (field1 field2, . . .) VALUES ("value1", "value2" . . .)	Inserts information into the table.
UPDATE tablename SET field1=value1, field2=value2 . . . WHERE condition the table.	Changes information already stored in
DELETE FROM tablename WHERE condition	Deletes a record from the specified table.
DROP tablename	Deletes the table.
DROP database	Deletes the database.
LOAD DATA INFILE "filename" INTO TABLE tablename	Loads a large quantity of data into the database.

ဒါက **Vote code** နမူနာပါ..ကျွန်တော် **database** နာမည်တွေ **table** နာမည်တွေမပြောတော့ပါဘူး။

ကြည့်ပြီးလုပ်ကြည့်ပါအဆင်ပြေမယ်လို့ထင်ပါတယ်။

```
<?
```

```
if(isset($_POST['vote']))
```

```
{
```

```
$servername="localhost";
```

```
$username="root";
```

```
$password="";
```

```
$con=mysqli_connect($servername,$username,$password);
```

```
if(!$con)
{
    die('could not connect');
}
```

```
mysql_select_db("vote",$con);
$query="SELECT * FROM vote";
$result=mysql_query($query);
```

```
$row=mysql_fetch_array($result);
$MrA=$row['MrA'];
$MrB=$row['MrB'];
$MrC=$row['MrC'];
```

```
$people=$_POST['vote'];
if($people=="mra")
{
    $MrA=$MrA+1;
    $query="UPDATE vote
    SET MrA='$MrA'";
    mysql_query($query);
}
```

```
elseif($people=="mrb")
{
    $MrB=$MrB+1;
    $query="UPDATE vote
    SET MrB='$MrB'";
    mysql_query($query);
}
```

```
elseif($people=="mrc")
{
    $MrC=$MrC+1;
    $query="UPDATE vote
    SET MrC='$MrC'";
    mysql_query($query);
}
```

```
}
```

```
else
```

```
{
```

```
echo "U NeeD to choose Vote PersoN";
```

```
}
```

```
mysql_close($con);
```

```
}
```

```
?>
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
```

```
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

```
<html xmlns="http://www.w3.org/1999/xhtml">
```

```
<head>
```

```
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
```

```
<title>Untitled Document</title>
```

```
</head>
```

```
<body>
```

```
<form id="form1" name="form1" method="post" action="">
```

```
<table width="315" border="1">
```

```
<tr>
```

```
<td>&nbsp;</td>
```

```
<td>&nbsp;</td>
```

```
<td>TotalRecord</td>
```

```
</tr>
```

```
<tr>
```

```
<td width="70">MrA</td>
```

```
<td width="40"><label>
```

```
<input name="vote" type="radio" value="mra" />
```

```
</label></td>
```

```
<td width="191">MrA
```

```
<?
```

```
echo $MrA;
```

```
?>
```

```

        </td>
    </tr>
    <tr>
        <td>MrB</td>
        <td><label>
            <input name="vote" type="radio" value="mrb" />
        </label></td>
        <td>MrB
            <?
                echo $MrB;
            ?>
        </td>
    </tr>
    <tr>
        <td>MrC    </td>
        <td><label>
            <input name="vote" type="radio" value="mrc" />
        </label></td>
        <td>MrC
            <?
                echo $MrC;
            ?></td>
    </tr>
    <tr>
        <td colspan="3"><label>
            <input type="submit" name="Submit" value="Vote" />
        </label></td>
    </tr>
</table>
</form>

</body>
</html>

```

ပြောလိုက်ပါဦးမယ်လေ။ အဟီး database ရော table နာမည်က vote, field name တွေက MrA,MrB,MrC, Type က INT ပဲထားလိုက်ပါ။

ဒါတွေကတော့ MySQL အခြေခံတွေပါပဲ။ဒီလောက်ဆိုရင်တော့ရှေ့ဆက်လေ့လာဖို့ အခြေခံရလောက်ပါပြီ။
သတိထားရမှာက ရှေ့မှာပြောခဲ့သလိုပါပဲ database နာမည် table နာမည်. နောက် Insert လုပ်ရာမှာ table
ထဲက field name တွေနဲ့ code ထဲက နာမည်တွေတူဖို့ အရေးကြီးပါတယ်။အကြီး အသေးလေးလွဲတာနဲ့
တော်တော် ဒုက္ခရောက်ပါတယ်။ဆက်ပြီးလေ့လာစရာတွေ အများကြီးရှိပါတယ်။SELECT လုပ်ရာမှာရောပေါ့
ရှာကြည့်ပါ။data row တွေအများကြီးထဲကမှ အပေါ်ဆုံး 5 row ပဲကြည့်ချင်တယ်ဆို ဘယ်လိုရေးရမလဲ။နောက်
Row 10 ကနေမှ ရှိသမျှအကုန်ကြည့်ချင်တယ်ဆိုဘယ်လိုလုပ်ရမလဲ စသဖြင့်ပေါ့။ရှာကြည့်ပါ။
နောက်ဆက်ပြီးတော့လည်းလေ့လာကြပါ။အားလုံးပဲA ဆင်ပြေကြပါစေ။

ကျေးဇူးတင်လွှာ

ကျွန်တော်၏မိဘနှစ်ပါးအား၎င်း၊ကျွန်တော့်ကိုသင်ကြားပေးသောဆရာဇော်လင်းဦး(emo)ကို၎င်း၊
ကျွန်တော်ရေးသမျှနားလည် မလည်လားတော့မသိ၊ဖတ်ပေးကြသော MZ မှ Member အားလုံးကို၎င်း
စာရေးပျင်းသောကျွန်တော့်ကို အတင်းတိုက်တွန်းမရမကရေးခိုင်းသောသူကိုအား၎င်း
အထူးကျေးဇူးတင်ရှိပါသည်။

ပထမတစ်အုပ်လိုစာလုံးပေါင်းမလွဲရအောင်တော့သတိထားလုပ်ထားပါသည်။မှားခဲ့သည်များရှိပါကလည်း
နားလည်ပေးကြပါဗျာ။