

အာကာဖြိုး(နည်းပညာ)



SQL လေ့လာခြင်းနှင့် လက်တွေ့အသုံးချ သင်ခန်းစာများ

www.akarphyoe.com whereheart8@gmail.com



ဒီစာအုပ်လေးမှာ ပါသော အကြောင်းအရာ သင်ခန်းစာများသည် SQL ကိုလေ့လာနေသောသူတစ်ယောက်အတွက် အခြေခံကျကျ ရှင်းပြထားသောကြောင့်

နားလည်လွယ်ပြီး အထောက်အကူဖြစ်စေမည်ဟု ယူဆပါသည်

စာအုပ်ပါသင်ခန်းစာများသည် ပြုစုဖန်တီးသူ ကိုယ်တိုင်ရေးသားထားခြင်းမဟုတ်ပဲ

ကိုကျော်ကို ရေးထားသောပိုဒ်များကို စုစည်းတင်ပြထားခြင်းသာ ဖြစ်သည်

ပြုစုသူကိုယ်တိုင်ကား SQL သင်ခန်းစာများကို အများနားလည်နိုင်စေရန် ရှင်းပြလောက်သည် အထိ တတ်ကျွမ်းနားလည်မှု မရှိသည်ကိုနစ်နပ်ပါသည်

ရှင်းလင်းစာင်ပြထားသော ကိုကျော်ကိုအားကျေးဇူးတင်ပါကြောင်း ဒီစာအုပ်ဖြင့်

သက်သေပြုလိုက်ပါသည်

ကျေးဇူးတင်လျှက်

အာကာဖြိုး(နည်းပညာ)

what is SQL?

SQL (Structure Query Language) သည် data တွေကို ကိုင်သွယ် အသုံးပြုနိုင်ရန်နှင့် data တွေကို ချိတ်ဆက်လုပ်ဆောင်နိုင်သော programming language တစ်ခုဖြစ်ပါတယ်။ SQL သည် ANSI (American National Standards Institute) standard language တစ်ခု ဖြစ်ပါတယ်။ ဒါပေမဲ့ သူရဲ့ version တွေအလိုက် အနည်းနဲ့ အများကွားခြားမှုတွေရှိပါတယ် ၊ အခြေခံ ကတော့ အတူတူပဲဖြစ်ပါတယ် ၊ SQL statement တွေသည် database ထဲမှ data တွေကို ပြောင်းလဲနိုင်တယ် ၊ ပြန်လဲရယူနိုင်ပါတယ်။ web page ပေါ်မှာဆိုရင် server-side scripting language ဖြစ်ကြတဲ့ PHP or ASP တွေနှင့် အသုံးပြုနိုင်ပါတယ်။ visual Basic program မှာလဲ database တွေကို ချိတ်ဆက်လုပ်ဆောင်နိုင်ပါတယ်။

RDBMS (relational database management system) သည် SQL language ၏ အခြေခံဖြစ်သလို modern database system တွေဖြစ်ကြတဲ့ MS

SQL Server, IBM DB2, Oracle, MySQL, and Microsoft Access တို့တွေအားလုံး၏ အခြေခံလဲဖြစ်ပါတယ်။ RDBMS ၏ အတိုကောက်က DBMS ပေါ့ဗျာ။ DBMS ကို အလွယ်နားလည်မယ်ဆိုရင် data တွေကို table ထဲမှာ သိမ်ဆည်းပါတယ် ၊ အဲဒီ table ထဲမှာ columns and rows တွေပါဝင်ပါတယ် ။ column မှာ column name ၊ data type တွေ နှင့် column အတွက် အခြား ကန့်သတ်ချက်တွေ ပါဝင်ပါတယ်။ row တွေမှာတော့ column တွေအတွက် data တွေနှင့် records တွေ ပါဝင်ပါတယ်။ အဲဒီ table ထဲမှာရှိသော data တွေကို ပြောင်းလဲပြင်ဆင်နိုင် တယ် ၊ ဖော်ပြနိုင်တယ် ၊ အဲဒီလို လုပ်နိုင်ရန်အတွက် database system တွေကိုနားလည်နိုင်တဲ့ server တစ်ခုလိုပါမယ် ၊ ကျွန်တော်ကတော့ Microsoft SQL Server Express ကိုပဲသုံးပါတယ် ၊ Microsoft SQL Server Express သုံးပုံလေးကို အရင် ပုံလေးတွေနှင့်ဖော်ပြပေးသွားမှာပါ ၊ ပြီးတော့ မှ SQL syntax တွေအကြောင်းကို လေ့လာမှာဖြစ် ပါတယ်။

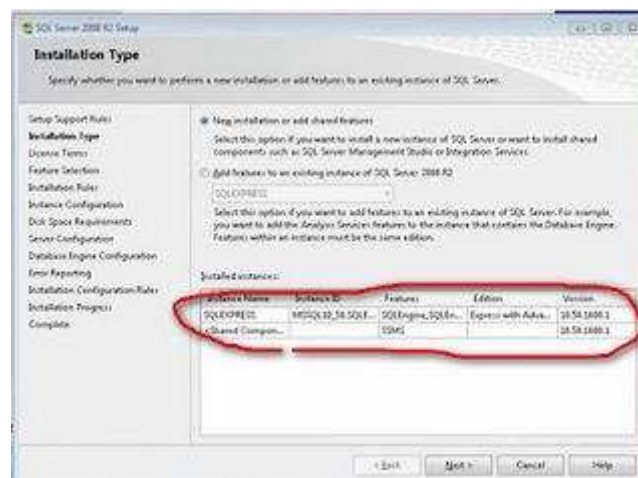
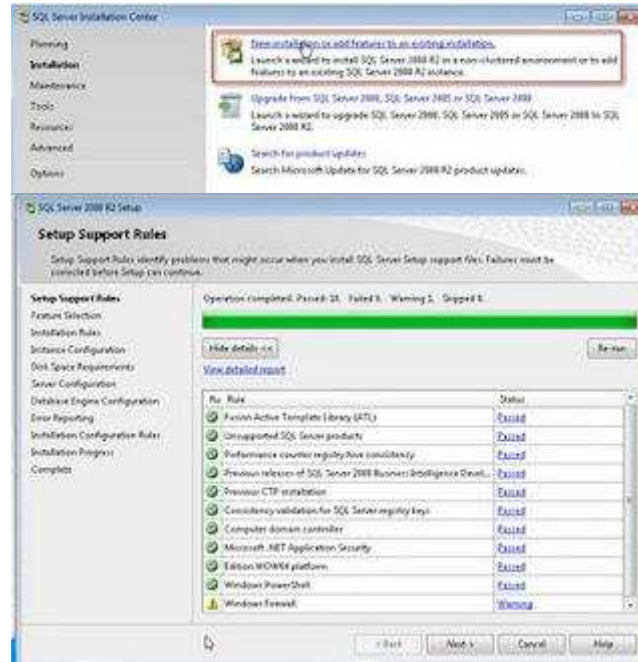
Server installation

အောက်ကပေးထားသော [SQL Server](#) မူရင်းဆိုက်မှာ download လုပ်ပါ။
run ပုံအဆင့်ဆင့် -->



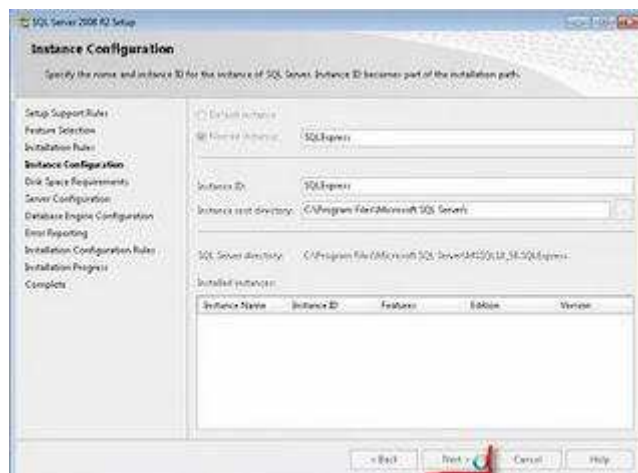
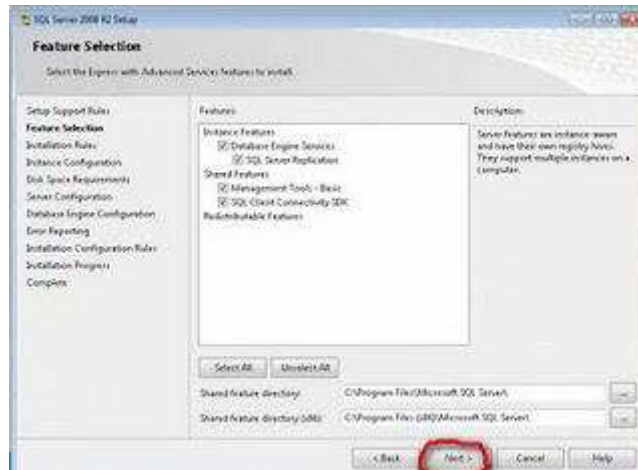
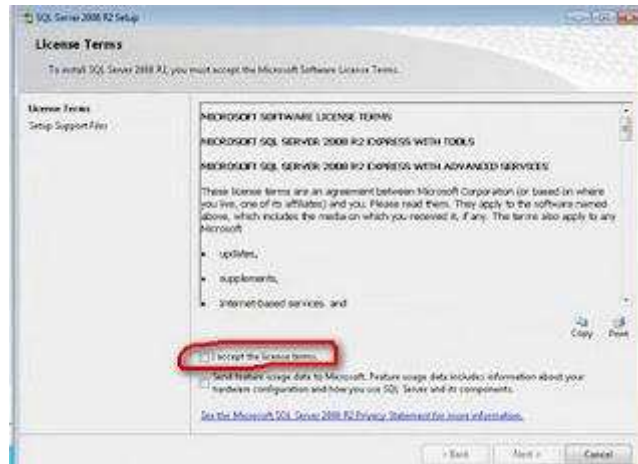
SQL လေ့လာခြင်း နှင့် လက်တွေ့ အသုံးချ သင်ခန်းစာများ

www.akarphyoe.com



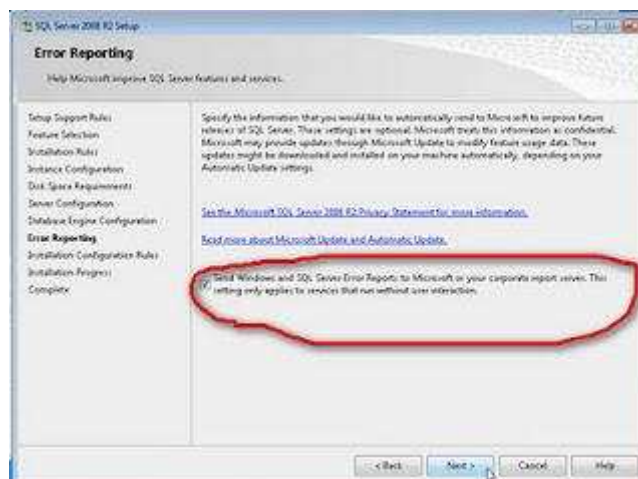
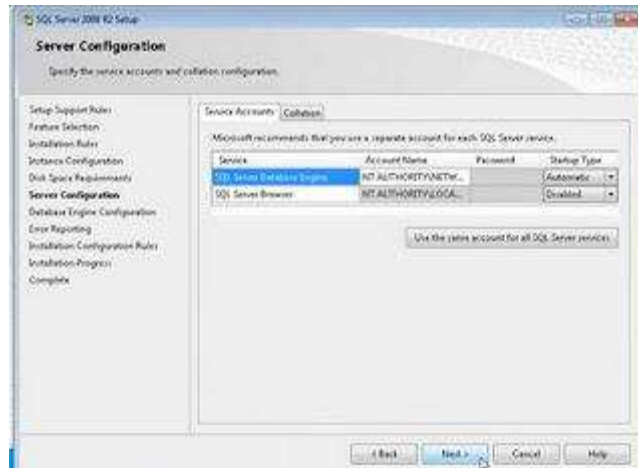
SQL လေ့လာခြင်း နှင့် လက်တွေ့ အသုံးချ သင်ခန်းစာများ

www.akarphyoe.com



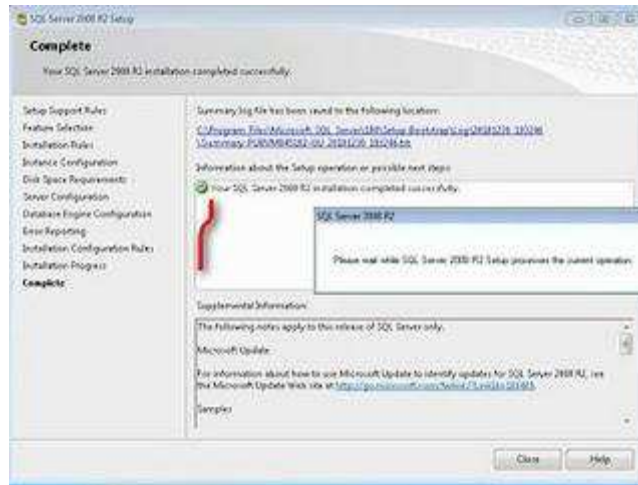
SQL လေ့လာခြင်း နှင့် လက်တွေ့ အသုံးချ သင်ခန်းစာများ

www.akarphyoe.com



SQL လေ့လာခြင်း နှင့် လက်တွေ့ အသုံးချ သင်ခန်းစာများ

www.akarphyoe.com

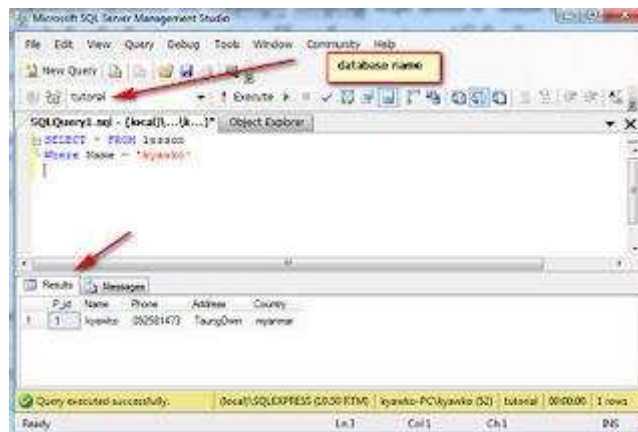
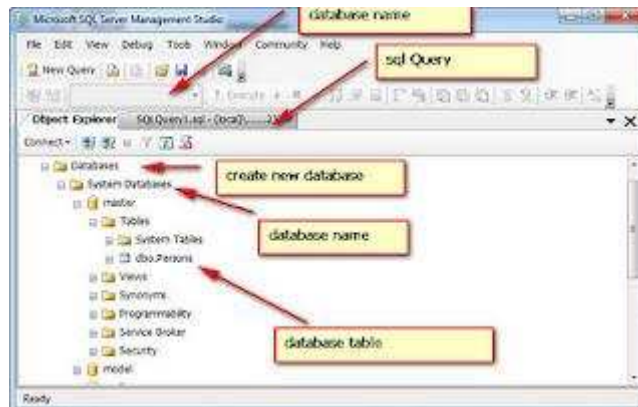


installation ပြီးသွားဆိုရင် server ကို connect လုပ်ပါမယ် ၊ sql server ကို ဖွင့်လိုက်ပါ။



SQL Server အသုံးပြုပုံ

Server ထဲမှာ database and table တည်ဆောက်ပုံ တွေ ၊ sql query နှင့် data တွေ အသုံးချပုံတွေ ကို ဖော်ပြထားပါတယ် ၊ ပုံနှင့်ဖော်ပြထားသောကြောင့် နားမလည်စရာမရှိဘူးလို့ ထင်ပါတယ်။



database အသစ်ဆောက်ခြင်တယ်ဆိုရင် database ကို right click နှိပ်ပြီး ဆောက်နိုင်ပါတယ် ၊ အောက်ကပုံကိုထပ်ကြည့်ပါ _

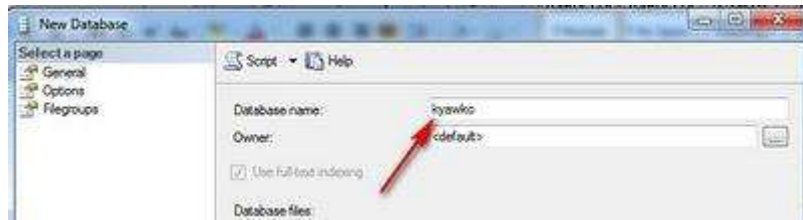


Table Basic

Relational database system မှာ table တစ်ခု (သို့) တစ်ခုထက်ပိုသော table object တွေ ဝင် ပါတယ်။ database တွေအတွက် data တွေ information တွေ ကို table ထဲမှာ သိမ်ဆည်းပါတယ်။ အဲဒီ table ထဲမှာ column and row တွေ ပါဝင်ပါတယ် ၊ column မှာ column name ၊ data type တွေ နှင့် column အတွက် အခြား ကန့်သတ်ချက်တွေ ပါဝင်ပါတယ်။ row တွေမှာတော့ column တွေအတွက် data တွေနှင့် records တွေ ပါဝင်ပါတယ်။

Database ဆောက်ပြီးပြီးဆိုရင် kyawko ဆိုပြီး object explorer မှာပေါ်နေပါလိမ့်မယ် ၊ အဲဒီ database ထဲတွင် table ဆောက်ခြင်းတယ်ဆိုရင် kyawko database ထဲမှာ table ကို right click လုပ်ပြီး new table ဆောက်နိုင်ပါတယ် ၊ table ဆောက်တဲ့အခါမှာ data type တွေကို ရွေးပေးပြီး ဆောက်နိုင်ပါတယ် ၊ table ထဲမှာ သတ်မှတ်သော data type တွေအကြောင်းကို နောက်ပိုင်းမှာလေ့လာရမှာဖြစ်ပါတယ်။ table design လုပ်တယ်ဆိုတာ column name နှင့် data type တွေကို ကြိုတင်သတ်မှတ်ခြင်းပါ။ table design လုပ်ပုံ -->

SQL လေ့လာခြင်း နှင့် လက်တွေ့ အသုံးချ သင်ခန်းစာများ

www.akarphyoe.com

Column Name	Data Type	Allow Nulls
P_id	int	<input checked="" type="checkbox"/>
Name	varchar(50)	<input checked="" type="checkbox"/>
Phone	nchar(10)	<input checked="" type="checkbox"/>
Address	varchar(50)	<input type="checkbox"/>
Country	varchar(MAX)	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

Table design လုပ်ပြီးပြီးဆိုရင် table ထဲသို့ data တွေကို ထည့်သွင်းရန် table ကို right click လုပ်ပြီး Edit Top 200 Rows မှာ ထည့်သွင်းနိုင်ပါတယ်။ data ထည့်သွင်းပြီး ပြီးဆိုရင် execute sql ပြန်လုပ်ပေးပါ။

	P_id	Name	phone	Address	Country
1	1	Kyauko	09258741	taungDwin	myanmar
2	2	warwar	093692581	pago	myanmar
3	3	koko	054258147	yangon	myanmar
4	4	eiei	063258471	mgway	myanmar
*	NULL	NULL	NULL	NULL	NULL

အပေါ်က table ကိုလေ့လာကြည့်မယ်ဆိုရင် columns 5 ခုနဲ့ row 4 ခု ပါဝင်ပါတယ်။ database မှာတော့ row တွေကို record လို့ခေါ်ပါတယ် ၊အပေါ်မှာဆောက်ခဲ့တဲ့ database ကိုခြုံကြည့်မယ်ဆိုရင်

Database name = Tutorial

Database table = Lesson

ကျွန်တော်အပေါ်မှာပြောခဲ့တဲ့ database ဆောက်တဲ့ပုံစံသည် sql server မှာပါသော tool တွေကို အသုံးချပြီး ဆောက်ပြထားတာ ဖြစ်ပါတယ်။ sql code တွေနဲ့ဆောက်တဲ့ ပုံစံကို နောက်ပိုင်းမှာ ဖော်ပြထား ပါတယ် ။ ကျွန်တော်တို့ table လဲဆောက်ပြီးပြီးဆိုတော့ sql language ကိုနောက်ခေါင်းစဉ်မှာ စပြီးလေ့လာကြပါမယ်

SQL - Queries

Sql ထဲမှာ အများဆုံးအသုံးပြုတာကတော့ query ပဲဖြစ်ပါတယ်။ query ဆိုတာ လွယ်လွယ် ပြော ရမယ်ဆိုရင်တော့ question တစ်ခုပဲဖြစ်ပါတယ်။ database နှင့် ပြန်အလှန် အသုံးပြုသော language ကို query language လို့ခေါ်ပါတယ်။ query သည် database ထဲမှာ သတ်မှတ်ထားသော အချက်အလက်တွေကို ရယူသော နည်းလမ်းပဲဖြစ်ပါတယ်။ database query မှာ select query နှင့် action query တို့ကို အသုံးပြုနိုင်ပါတယ်။ select query သည် တစ်ခု(သို့) တစ်ခုထက်ပိုသော table တွေမှ data တွေကို ရယူနိုင်ပါတယ်။ action query သည် သတ်မှတ်ထားသော data တွေကို insert ,updating and deleting လုပ်နိုင်ပါတယ်။ SQL Statement တွေကို စာလုံး အသေး (သို့) အကြီး နှစ်မျိုးလုံးဖြစ် အသုံးပြုနိုင်ပါတယ်။ statement တစ်ကြောင်းပြီးတိုင် semicolon (;) ကော်လံ နဲ့ အဆုံးသတ်ပေးရပါတယ်။ semicolon သည် database system ထဲမှာ statement တစ်ခုစီကို ပိုင်းခြားပေးပါတယ်။ MS SQL Server မှာ (;) မသုံးလဲရပါတယ်။

SQL DML and DDL

Sql သည် database ထဲမှာ data တွေကို လုပ်ဆောင်တဲ့အခါမှာ အစိတ်ပိုင်း နှစ်ခုခွဲထားပါတယ်။ Data Manipulation Language (DML) and Data Definition Language (DDL) တို့ဖြစ်ပါတယ်။

Data Manipulation Language (DML)

DML သည် database ထဲမှာ သတ်မှတ်ထားသော data တွေကို ပြန်လဲရယူခြင်း သိမ်းဆည်းခြင်း၊ ပြောင်းလဲခြင်း၊ ဖျက်ခြင်း နှင့် ထည့်သွင်းခြင်း စသည်တို့ကို လုပ်ဆောင်နိုင်ပါတယ်။ Query နှင့် update comment ပုံစံတွေသည် DML ၏ အစိတ်ပိုင်းထဲ မှာ ပါဝင်ပါတယ်။

SELECT - database ထဲမှာ data တွေကို ဆွဲယူအသုံးချရန်

UPDATE - database ထဲမှာ data တွေကို ပြောင်းလဲရန်

DELETE - database ထဲမှာ data တွေကို ဖျက်ရန်

INSERT INTO - database ထဲတွင် data အသစ်တွေပေါင်းထည့်ရန်။

Data Definition Language (DDL)

Sql ၏ DDL အစိတ်ပိုင်း သည် database table ကို create and delete လုပ်ရန်ခွင့်ပြုပေးပါတယ်။ ထိုအပြင် အောက်မှာဖော်ပြထားသော DDL statement တွေကိုလဲ လုပ်ဆောင်ပါတယ်။

CREATE DATABASE - database အသစ်ဖန်တီးရန်

ALTER DATABASE - database ကို ပြုပြင်ရန်

CREATE TABLE - database ထဲမှာ table အသစ်ဆောက်ရန်

ALTER TABLE - table ပြုပြင်ရန်

DROP TABLE - table ဖျက်ရန်

CREATE INDEX - table ထဲမှာ မာတိကများသတ်မှတ်သောပုံစံ(search key)

DROP INDEX - table ထဲမှာ မာတိက များကို ဖျက်ပစ်ရန်

SQL SELECT Statement

Select statement သည် database ထဲမှာ data တွေကို ရွေးချယ်ပြီး လုပ်ဆောင်ခြင်းတဲ့အခါမှာ၊ အသုံးပြုပါတယ်။ Select statement သည် SQL language မှာ အဓိကသုံးသော statement ဖြစ်ပါတယ်။ select statement ၏ သတ်မှတ်သောပုံစံကို အရင် ကြည့်ကြည့်ပါ -

SELECT column_name(s)
FROM table_name

SELECT သည် table ထဲမှ မည်သည့် column ကို ရွေးချယ်ဖော်ပြမယ်(သို့) table တစ်ခုလုံးကို ဖော်ပြပေးပါတယ်။ FROM သည် database ထဲမှာ ရှိသော table တွေအများကြီးထဲကမှ tableတစ်ခု ကို ရွေးချယ်ပါတယ်။ မှတ်ချက် - တစ်ခုထဲရှိရင်တော့ ရွေးချယ်စရာမလိုဘူးပေါ့ဗျာ။

ဥပမာ ~ ကျွန်တော်တို့က အောက်က table ထဲမှ Name column ကို ဖော်ပြချင်တယ် ဆိုရင်

	P_id	Name	phone	Address	Country
1	1	kyawko	09258741	laungDwin	myanmar
2	2	warwar	093692581	pago	myanmar
3	3	loko	054258147	yangon	myanmar
4	4	eni	063258471	ngway	myanmar
*	NULL	NULL	NULL	NULL	NULL

Select Name

from lesson

အဲလိုရေးလိုက်ရင် Name column မှာရှိသော data
တွေအားလုံးပေါ်ပြပေးမှာဖြစ်ပါတယ်။

	Name
1	eiei
2	Khine
3	koko
4	kyawko
5	warwar

(*) သည် sql language ၏ shortcat တစ်ခုဖြစ်ပါတယ်။(*) shortcat
သည် table တစ်ခုလုံးကို ဖော်ပြချင်တဲ့အခါမှာ အသုံးပြုပါတယ် ၊ Select * from
lesson လို့ရေးလိုက်ရင် table တစ်ခုလုံးကိုဖော်ပြပေးမှာဖြစ်ပါတယ်။

DISTINCT Statement and Where clause

Column တစ်ခုတွင် တူညီသော data တွေနှင့် မတူညီသော data တွေ ပေါင်းစပ်ပါဝင်နေပါတယ်။ အဲ့ဒီ column မှ မတူညီတဲ့ data တွေကိုပဲ လိုချင်တယ်ဆိုရင် DISTINCT Keyword ကို သုံးပါတယ်။ DISTINCT Keyword သတ်မှတ်သောပုံစံ

```
SELECT DISTINCT column_name(s)
FROM table_name
```

အောက်က table ထဲမှာ country column ကို DISTINCT လုပ်ပြထားပါတယ် ၊

P_id	Name	Phone	Address	Country
1	kyawko	092581473	TaungOwin	myanmar
2	warwar	093692581	ngway	myanmar
3	koko	89258369	moscow	russia
4	eel	06258147	Shanghai	China

```
SELECT DISTINCT Country FROM lesson
```

Country column ထဲမှာ တူနေသော data တွေကို ချန်ခဲ့ပြီး မတူညီသော data တွေကို ဖော်ပြပေးပါတယ်။ ပေါ်လာသော column -

	Country
1	China
2	myanmar
3	russia

Where Clause

Clause ထဲမှာ statement နှင့် queries အစိတ်ပိုင်းတွေ ပေါင်းစပ်ပါဝင်ပါတယ်။ ဒါကြောင့် where clause သည် conditional statement တွေနဲ့ တွဲ၍ အသုံးပြုနိုင်သလို query တွေနဲ့လဲ အသုံးပြုနိုင်ပါတယ်။ SQL သည် တစ်ချို့မှာ row တစ်ခုကိုပဲ အလုပ်လုပ်ပါတယ်။ conditional statement တွေသည် အချိန်ချိန်တိုင်း true value ကို return ပြန်ပေးနေတဲ့ အချို့မှာ row တွေက လိုက်ပြီး return ပြန်ပေးနေမှာဖြစ်ပါတယ်။ where clause တွင် operator တွေကို သတ်မှတ်ပေးတဲ့အခါမှာ စာသားတွေနှင့် ကိန်းတန်ဖိုးတွေပါလာနိုင်ပါတယ်။ အဲဒီ တန်ဖိုးတွေကို သတ်မှတ်ရန် single quotes or double quotes ကို အသုံးပြုရပါမည်။

SQL လေ့လာခြင်း နှင့် လက်တွေ့ အသုံးချ သင်ခန်းစာများ

www.akarphyoe.com

မှတ်ချက် - numeric values တန်ဖိုးတွေကိုတော့ quotes အသုံးပြုလိုမရပါဘူး ။
where clause သတ်မှတ်သောပုံစံ -->

```
SELECT                                     column_name(s)
FROM                                       table_name
WHERE column_name operator value
```

အောက်က table ထဲမှ Name column ကို where clause အသုံးပြုပြီး ဖော်ပြထားပါတယ်။ where clause နှင့် တွဲဖက်ပြီးသုံးသော operator အကြောင်းကို အောက်မှာဆွေးနွေးထားပါတယ်။ database name=tutorial ,tablename =lesson ဖြစ်ပါတယ်။

P_id	Name	Phone	Address	Country
1	kyawko	092581473	TaungOwin	myanmar
2	warwar	093692581	ngway	myanmar
3	koko	09258369	moscow	russia
4	eel	06258147	Shanghai	China

```
SELECT * FROM lesson
```

SQL လေ့လာခြင်း နှင့် လက်တွေ့ အသုံးချ သင်ခန်းစာများ

www.akarphyoe.com

Where Name = 'kyawko'

Kyawko ပါသော Name column မှာ row ကို ဖော်ပြပေးမှာဖြစ်ပါတယ်။

P_id	Name	Phone	Address	Country
1	kyawko	092581473	TaungDwin	myanmar

WHERE Clause ထဲမှာ အသုံးပြုနိုင်သော Operators

= Equal

<> Not equal

> Greater than

< Less than

>= Greater than or equal

<= Less than or equal

BETWEEN တန်ဖိုးနှစ်ခုရဲ့ကြား ကဏတစ်ခုကို ဖော်ပြပေးပါတယ်။

LIKE Column ထဲမှာသတ်မှတ်ထားသော data တွေကို ရှာရန် ၊

IN အတိကျ သိသော data တွေ select လုပ်ရန်။

SQL AND & OR Operators

And operators သည် ပထမ အခြေအနေ နဲ့ ဒုတိယအခြေအနေ နှစ်ခုလုံးမှန် မှသာလျှင် သတ်မှတ်ထားသော record ကိုဖော်ပြပေးများဖြစ်ပါတယ်။ operator တွေကိုတော့ where clause နှင့် တွဲ၍ အသုံးပြုပါတယ်။

P_id	Name	Phone	Address	Country
1	kyawko	092581473	TaungOwin	myanmar
2	warwar	093692581	ngway	myanmar
3	koko	09258369	moscow	russia
4	eiei	06258147	Shanghai	China

`SELECT * FROM lesson`

`Where Name = 'kyawko' AND Country = 'myanmar'`

အပေါ်က code သည် Name column နှင့် Country column တို့၏ တန်ဖိုးတွေမှန် မှသာ record ကိုဖော်ပြပေးမယ်လို့ ဆိုလိုပါတယ်။ အခြေအနေ နှစ်ခုလုံးမှန်သောကြောင့် record (kyawko) ကို ဖော်ပြပေးပါတယ်။

SQL လေ့လာခြင်း နှင့် လက်တွေ့ အသုံးချ သင်ခန်းစာများ

www.akarphyoe.com

P_id	Name	Phone	Address	Country
1	kyawko	092581473	TaungDwin	myanmar

OR operator သည် ပထမ အခြေအနေ နဲ့ ဒုတိယအခြေအနေ နှစ်ခုထဲမှ တစ်ခုမှန် (သို့) နှစ်ခုလုံးမှန်မယ် ဆိုရင်လဲ record ကို ဖော်ပြပေးမှာဖြစ်ပါတယ်။ table ကတော့ အပေါ်က table ကိုပဲ အသုံးပြုထားတာဖြစ်ပါတယ်။

```
SELECT * FROM lesson
```

```
Where Name ='warwar' OR Country ='Thai'
```

P_id	Name	Phone	Address	Country
2	warwar	093692581	mgway	myanmar

And နှင့် OR ပေါင်းစပ်အသုံးပြုသော ပုံစံ

SELECT * FROM lesson

Where Name = 'warwar' AND (Country = 'myanmar' OR Country = 'Thai')

P_id	Name	Phone	Address	Country
2	warwar	093692581	mgway	myanmar

ORDER BY Keyword

Order by keyword သည် sql query မှ ရရှိလာသော result ကို ပြန်လည်၍ ငယ်စဉ်ကြီးလိုက် စီပေးခြင်းဖြစ်ပါတယ်။ record တွေကို ကြီးစဉ်ငယ်လိုက် ပြန်စီချင်တယ်ဆိုရင်တော့ DECS keyword ကို အသုံးပြုရပါမည် ၊ သတ်မှတ်သောပုံစံ -->

```
SELECT column_name(s)
FROM table_name
ORDER BY column_name(s) ASC|DESC
```

အပေါ်မှာ ဖော်ပြခဲ့သော lesson table ကို အသုံးပြုထားပါတယ်။ database name=tutorial ,tablename =lesson ဖြစ်ပါတယ်။

SELECT * FROM lesson

SQL လေ့လာခြင်း နှင့် လက်တွေ့ အသုံးချ သင်ခန်းစာများ

www.akarphyoe.com

ORDER BY Name

	P_id	Name	Phone	Address	Country
1	4	eiei	06258147	Shanghai	China
2	3	koko	89258369	moscow	russia
3	1	kyawko	092581473	TaungDwin	myanmar
4	2	warwar	093692581	mgway	myanmar

ORDER BY DESC

record တွေကို ကြီးစဉ်ငယ်လိုက် ပြန်စီချင်တယ်ဆိုရင်တော့ DECS keyword ကို အသုံးပြုရပါမည်။

SELECT * FROM lesson

SQL လေ့လာခြင်း နှင့် လက်တွေ့ အသုံးချ သင်ခန်းစာများ

www.akarphyoe.com

ORDER BY Name DESC

P_id	Name	Phone	Address	Country
2	warwar	093692581	mgway	myanmar
1	kyawko	092581473	TaungDwin	myanmar
3	koko	89258369	moscow	russia
4	eiei	06258147	Shanghai	China

SQL AND & OR Operators

And operators သည် ပထမ အခြေအနေ နဲ့ ဒုတိယအခြေအနေ နှစ်ခုလုံးမှန် မှသာလျှင် သတ်မှတ်ထားသော record ကိုဖော်ပြပေးများဖြစ်ပါတယ်။ operator တွေကိုတော့ where clause နှင့် တွဲ၍ အသုံးပြုပါတယ်။

P_id	Name	Phone	Address	Country
1	kyawko	092581473	TaungOwin	myanmar
2	warwar	093692581	ngway	myanmar
3	koko	09258369	moscow	russia
4	eiei	06258147	Shanghai	China

SELECT * FROM lesson

Where Name = 'kyawko' AND Country = 'myanmar'

အပေါ်က code သည် Name column နှင့် Country column တို့၏ တန်ဖိုးတွေမှန် မှသာ record ကိုဖော်ပြပေးမယ်လို့ ဆိုလိုပါတယ်။ အခြေအနေ နှစ်ခုလုံးမှန်သောကြောင့် record (kyawko) ကို ဖော်ပြပေးပါတယ်။

P_id	Name	Phone	Address	Country
1	kyawko	092581473	TaungDwin	myanmar

OR operator သည် ပထမ အခြေအနေ နဲ့ ဒုတိယအခြေအနေ နှစ်ခုထဲမှ တစ်ခုမှန် (သို့) နှစ်ခုလုံးမှန်မယ် ဆိုရင်လဲ record ကို ဖော်ပြပေးမှာဖြစ်ပါတယ်။ table ကတော့ အပေါ်က table ကိုပဲ အသုံးပြုထားတာဖြစ်ပါတယ်။

SELECT * FROM lesson

Where Name ='warwar' OR Country ='Thai'

P_id	Name	Phone	Address	Country
2	warwar	093692581	mgway	myanmar

And နှင့် OR ပေါင်းစပ်အသုံးပြုသော ပုံစံ

SELECT * FROM lesson

Where Name = 'warwar' AND (Country = 'myanmar' OR Country = 'Thai')

P_id	Name	Phone	Address	Country
2	warwar	093692581	mgway	myanmar

ORDER BY Keyword

Order by keyword သည် sql query မှ ရရှိလာသော result ကို ပြန်လည်၍ ငယ်စဉ်ကြီးလိုက် စီပေးခြင်းဖြစ်ပါတယ်။ record တွေကို ကြီးစဉ်ငယ်လိုက် ပြန်စီချင်တယ်ဆိုရင်တော့ DECS keyword ကို အသုံးပြုရပါမည် ၊ သတ်မှတ်သောပုံစံ -->

```
SELECT column_name(s)
FROM table_name
ORDER BY column_name(s) ASC|DESC
```

အပေါ်မှာ ဖော်ပြခဲ့သော lesson table ကို အသုံးပြုထားပါတယ်။ database name=tutorial ,tablename =lesson ဖြစ်ပါတယ်။

SELECT * FROM lesson

SQL လေ့လာခြင်း နှင့် လက်တွေ့ အသုံးချ သင်ခန်းစာများ

www.akarphyoe.com

ORDER BY Name

	P_id	Name	Phone	Address	Country
1	4	eiei	06258147	Shanghai	China
2	3	koko	89258369	moscow	russia
3	1	kyawko	092581473	TaungDwin	myanmar
4	2	warwar	093692581	mgway	myanmar

ORDER BY DESC

record တွေကို ကြီးစဉ်ငယ်လိုက် ပြန်စီချင်တယ်ဆိုရင်တော့ DECS keyword ကို အသုံးပြုရပါမည်။

SELECT * FROM lesson

ORDER BY Name DESC

P_id	Name	Phone	Address	Country
2	warwar	093692581	mgway	myanmar
1	kyawko	092581473	TaungDwin	myanmar
3	koko	89258369	moscow	russia
4	eiei	06258147	Shanghai	China

Insert Into Statement and UPDATE Statement

Insert Into Statement သည် table ထဲကို data အသစ်တွေ ထည့်ပေးခြင်းတဲ့အခါမှာ အသုံးပြုပါ တယ်။ data တွေကို insert လုပ်တဲ့အခါမှာ ပုံစံနှစ်မျိုးနှင့် ထည့်သွင်းနိုင်ပါတယ်။

Data ထည့်သွင်းမည့် column name မသတ်မှတ်ဘဲနဲ့ ထည့်သွင်းခြင်း ။

Data ထည့်သွင်းမည့် column name သတ်မှတ်ပြီး ထည့်သွင်းခြင်း ။

ပထမ နည်းလမ်း

အောက်မှာဖော်ပြထားတဲ့ table ထဲကို new data(row) တစ်ခုကိုထပ်ပြီးထည့်သွင်းပါမယ် ၊ အဲ့ဒီလို ထည့်သွင်းတဲ့ data တွေသည် numeric တန်ဖိုး တွေ (သို့) စာသားတွေ ပါဝင်နိုင်ပါတယ်။ ကိန်းဂဏန်း

SQL လေ့လာခြင်း နှင့် လက်တွေ့ အသုံးချ သင်ခန်းစာများ

www.akarphyoe.com

တွေအတွက်ပြဿနာမရှိပေးမဲ ၊ စာသားတွေ ကို ထည့်သွင်းပေးတဲ့အခါမှာတော့ quotes (' ') ကို အသုံးပြုပေးရပါမယ်။

P_id	Name	Phone	Address	Country
1	kyawko	092581473	TaungDwin	myanmar
2	warwar	093692581	ngway	myanmar
3	loko	09258369	moscow	ruasia
4	eiei	06258147	Shanghai	China

ပုံ(1)

INSERT INTO lesson

VALUES (5,'Khnie',0936258,'yangon','myanmar')

P_id	Name	Phone	Address	Country
1	kyawko	092581473	TaungDwin	myanmar
2	warwar	093692581	ngway	myanmar
3	loko	09258369	moscow	ruasia
4	eiei	06258147	Shanghai	China
5	Khnie	0936258	yangon	myanmar

ပုံ(2)

ဒုတိယ နည်းလမ်း

တစ်ခါတစ်လေကျရင် row တစ်ခုလုံး မဟုတ်ဘဲနဲ့ column တစ်ခု၏ တစ်နေရာရာမှာ data တွေ ထည့်သွင်းဖို့ လိုလာပြီဆိုရင် - ကိုယ်လိုချင်သော column ကို သတ်မှတ်ပေးရပါတယ်။ အောက်မှာ သတ်မှတ် သောပုံစံကို ဖော်ပြထားပါတယ်။ သတ်မှတ်သောပုံစံ -->

```
INSERT INTO table_name (column1, column2, column3,...)
VALUES (value1, value2, value3,...)
```

ပုံ(2) မှ table ထဲသို့ data တွေ ထည့်သွင်းပါမယ်။

```
INSERT INTO lesson (P_id ,Name ,Address )
VALUES(6,'Zin Mar',092581472)
```

P_id	Name	Phone	Address	Country
1	Kyawko	092581473	TaungDwin	myanmar
2	warwar	093692581	ngalay	myanmar
3	koko	89258369	moscow	russia
4	ein	06258147	Shanghai	China
5	Khine	0936258	yangon	myanmar
6	Zin Mar	0936587		

ပုံ(3)

UPDATE Statement

SQL UPDATE သည် ရှိနေသော table rows တွေကို တန်ဖိုးအသစ်တွေ ပေါင်းထည့်ပေးချင်တဲ့အခါမှာ အသုံးပြုတဲ့ command တစ်ခုဖြစ်ပါတယ်။ UPDATE syntax ထဲက where clause ကို သတ်မှတ် တဲ့အခါမှာ သတိထားရပါမယ်။ where clause သည် record တစ်ခု (သို့) records တွေအားလုံးကို update လုပ်နိုင်ပါတယ်။ တကယ်လို့ ကျွန်တော်တို့က where clause မပါဘဲနဲ့ SET ကိုသာ သုံးလိုက်မယ်ဆိုရင် records အားလုံးသည် update လုပ်လိုက်သော data တွေဖြစ်ကုန်ပါလိမ့်မယ်။ where clause မှ column name တွေကို conditional statement တွေနဲ့လဲ တွဲ၍ အသုံးပြုနိုင်ပါတယ်။

သတ်မှတ်သောပုံစံ -->

```
UPDATE table_name
SET column_name=value, column_name=value2,...
WHERE column_name
```

အောက်တွင်ဖော်ပြထားသော table ထဲမှ နောက်ဆုံး record ကို update လုပ်မှာဖြစ်ပါတယ်။

P_id	Name	Phone	Address	Country
1	kyawko	092581473	TaungDwin	myanmar
2	warwar	093692581	nigway	myanmar
3	koko	89258369	moscow	russia
4	eiei	06258147	Shanghai	China
5	Kline	0936258	yangon	myanmar
6	Zin Mar	0936587		
7	nwe nwe			

UPDATE lesson

SET phone=09258978, Country = 'malaysia' ,Address = 'kuala lumpur'

WHERE Name = 'thuthu'

SQL လေ့လာခြင်း နှင့် လက်တွေ့ အသုံးချ သင်ခန်းစာများ

www.akarphyoe.com

p_id	Name	Phone	Address	Country
1	Kyauka	092581473	TaungOwin	myanmar
2	warwar	093692581	mgway	myanmar
3	koko	89258369	moscow	russe
4	eel	06258147	Shanghai	China
5	Khine	0936258	yangon	myanmar
6	Zin Mar	0936587		
7	nwe nwe	9258978	kuala lumpur	malaysia

SQL LIKE Operator and DELETE Statement

Like operator သည် column တွေထဲမှာ သတ်မှတ်ထားသော data တွေကို ရှာဖွေရန်အသုံးပြု ပါတယ်။ Like operator ကို SQL wildcards တွေနှင့် တွဲ၍ အသုံးပြုနိုင်ပါတယ်။ ကျွန်တော် like operator နှင့် wildcards ကို တစ်ခါတည်းပေါင်းရှင်းမှာ ဖြစ်ပါတယ်။ သတ်မှတ်သောပုံစံ ~

```
SELECT column_name(s)
FROM table_name
WHERE column_name LIKE pattern
```

Wildcards % အသုံးပြုပုံ

ကျွန်တော်တို့က data ၏ နာမည်ကို အစပဲ သိတယ်ဆိုရင် wildcards % ကိုအသုံးပြုပြီး data ကို ရှာဖွေနိုင်ပါတယ်။ ဒါပေမဲ့ သူက where clause နှင့်တွဲ၍ အသုံးပြုသောကြောင့် ရှာချင်တဲ့ data ၏ column ကိုတော့ သတ်မှတ်ပေးရပါမယ်။ မှတ်ချက်- အောက်မှာဖော်ပြထားသော table ကို ဥပမာ အားလုံးအတွက် သုံးမှာဖြစ်သောကြောင့် table ကို ထပ်ခါထပ်ခါ မဖော်ပြတော့ပါဘူး။

P_id	Name	Phone	Address	Country
1	Iyanko	092581473	TaungDwin	myanmar
2	warwar	093692581	ngway	myanmar
3	Iziko	09258369	moscow	ruissia
4	eel	06258147	Shanghai	China
5	Khine	0936258	yangon	mayanmar

country column မှာရှိသော m နှင့်စတဲ့ data တွေကို ဖော်ပြချင်တာဖြစ်ပါတယ်။

```
SELECT * FROM lesson
```

```
WHERE Country LIKE 'm%'
```


SQL လေ့လာခြင်း နှင့် လက်တွေ့ အသုံးချ သင်ခန်းစာများ

www.akarphyoe.com

P_id	Name	Phone	Address	Country
1	kyawko	092581473	TaungDwin	myanmar
2	warwar	093692581	mgway	myanmar
5	Khine	0936258	yangon	m yanmar

အပေါ်ပြောခဲ့သလို ကိုယ်ရှာချင်တဲ့ data ၏ အစကို မသိဘူး character တစ်ချို့ကိုပဲ မှတ်မိတယ် ဆိုရင် wildcard **% char %** ကိုသုံးပြီး ရှာနိုင်ပါတယ်။ table ကိုတော့ အပေါ်မှာဖော်ပြထားတဲ့ table ကိုပဲသုံးထားပါတယ်။

```
SELECT * FROM lesson
```

```
WHERE Name LIKE '%hin%'
```

SQL လေ့လာခြင်း နှင့် လက်တွေ့ အသုံးချ သင်ခန်းစာများ

www.akarphyoe.com

P_id	Name	Phone	Address	Country
5	Khine	0936258	yangon	mayanmar

Wildcards _ အသုံးပြုပုံ

(_) ၎င်း သင်္ကေတ သည် data ၏ အစစလုံးတစ်လုံးကိုသာ ကိုယ်စားပြု ရှာဖွေပေးပါတယ်။ table ထဲမှ khine ကို select ပြထားပါတယ်။

```
SELECT * FROM lesson
```

```
WHERE Name LIKE '_hine'
```

P_id	Name	Phone	Address	Country
5	Khine	0936258	yangon	mayanmar

Wildcard [charlist] အသုံးပြုပုံ

[] wildcard သည် data တွေ၏ အစာလုံးတစ်ချို့ကို မှတ်မိပြီး မသဲကွဲ မသေချာ ဖြစ်နေတဲ့ အခါမှာ အသုံးပြုနိုင်ပါတယ်။ ဥပမာဗျာ - ကျွန်တော်က table ထဲမှ Name column ၏ eiei ကို select လုပ်ချင်တာ ဒါပေမဲ့ စိတ်ထဲမှာ အစာလုံးက e or m or q လား အဲလို မသေချာတဲ့ အချိန်ကျရင် [] wildcard ကို အသုံးပြုနိုင်ပါတယ်။

```
SELECT * FROM lesson
```

```
WHERE Name LIKE '[mqe]%'
```

P_id	Name	Phone	Address	Country
4	eiei	06258147	Shanghai	China

[[^] charlist] သည် အပေါ်မှာ ဖော်ပြခဲ့တဲ့ [] wildcard နဲ့ ဆန့်ကျင်ဘက်ဖြစ်ပါတယ်။
ဒီနေရာမှာ server version တွေအလိုက် ကွဲပြားမှုလေးတွေရှိပါတယ်။
ကျွန်တော်ခုသုံးတဲ့ Microsoft SQL Server Express ကတော့ [[!] charlist] နဲ့ မရပါဘူး ၊
[[^]charlist] နဲ့ သုံးမှ select လုပ်ပေးပါတယ်။

SELECT * FROM lesson

WHERE Country LIKE '[[^]msc]%'

P_id	Name	Phone	Address	Country
3	koko	89258369	moscow	russia

DELETE Statement

SQL လေ့လာခြင်း နှင့် လက်တွေ့ အသုံးချ သင်ခန်းစာများ

www.akarphyoe.com

DELETE Statement သည် table ထဲမှာ မလိုချင်သော row တွေကို ဖျက်ချင်တဲ့အခါမှာ အသုံးပြုပါတယ်။ where clause မှာ conditional statement တွေနဲ့လဲတွဲ၍ အသုံးပြုနိုင်ပါတယ်။ သတိထားရန် □ where clause မပါဘဲနဲ့ ကြည့်လိုက်ရင် table ထဲမှာရှိသော rows အားလုံးကို ဖျက်ဆီးပါလိမ့်မယ်။ သတ်မှတ်သောပုံစံ -->

DELETE FROM table_name
WHERE column_name

အောက်မှာဖော်ပြထားသော table ထဲမှ နောက်ဆုံး row ကို ဖျက်မှာ ဖြစ်ပါတယ်။

P_id	Name	Phone	Address	Country
1	kyawko	092581473	TaungOwin	myanmar
2	warwar	093692581	mgway	myanmar
3	loko	89258369	moscow	rusia
4	eiei	06258147	Shanghai	China

DELETE FROM lesson

WHERE Name ='eiei'

SQL Execute ပြန်လုပ်ပြီး ကြည့်လိုက်ရင် row(eiei) မရှိတော့ တာကို တွေ့ရမှာပါ။

BETWEEN Operator

Between operator သည် သတ်မှတ်ထားသော data တန်းဖိုးနှစ်ခု၏ ကြား data တွေကို select လုပ်ပေးပါတယ် ၊ ဒါပေမဲ့ database ဆောက်ခဲ့တဲ့ ပေါ်မှာ မူတည်ပြီးတော့ ကွဲပြားခြားနားမှုတွေရှိပါတယ် ၊ between operator သတ်မှတ်သော ပုံစံသည် -->

```
SELECT column_name(s)
FROM table_name
WHERE column_name
BETWEEN value1 AND value2
```

ကျွန်တော် အောက်က table ကို နမူနာ ပြ၍ database တွေအလိုက် ကွဲပြားမှုတွေကိုရှင်းပြမှာ ဖြစ်ပါတယ်၊ အဲဒီ လို့ ရှင်းတဲ့အခါမှာ table ထဲမှ name တွေကို သုံးပြီး ရေးမှာ ဖြစ်ပါတယ် ၊ အရင်ဆုံး between operator နှင့် select လုပ်ပုံကို အရင် လေ့လာကြည့်လိုက်ပါ --

P_id	Name	Phone	Address	Country
1	kyawko	092581473	TaungDwin	myanmar
2	warwar	093692581	ngway	myanmar
3	koko	89258369	moscow	russia

SELECT * FROM lesson
WHERE Name
BETWEEN 'kyawko' AND 'koko'

အပေါ်ကတိုင်းကြေညာလိုက်ရင် sql server က kyawko နှင့် koko ကြားမှာ ရှိသော warwar ကို select လုပ်ပြမှာ ဖြစ်ပါတယ်။

P_id	Name	Phone	Address	Country
2	warwar	093692581	mgway	myanmar

1. တစ်ချို့ database တွေမှာ တော့ kyawko နှင့် koko data တွေသည် Name မှာ ပါဝင်တဲ့ data list မဟုတ်ကြပါဘူး ၊ ဘာကြောင့်လဲဆိုတော့ between operator သည် AND operator နှင့် စစ်လိုရ တဲ့ အဖြေမှန်ကို ဖယ်ထုတ်ပြီး ကြားထဲမှာရှိသော data field ကို သာ select လုပ်ပေးပါတယ်။
2. တစ်ချို့ database တွေမှာ တော့ kyawko နှင့် koko data တွေသည် Name မှာ ပါဝင်တဲ့ data list တွေဖြစ်ကြပါတယ် ၊ ဘာကြောင့်လဲဆိုတော့ between operator သည် AND operator နှင့် စစ်လိုရတဲ့ အဖြေမှန်တွေ select လုပ်ပေးပါတယ်။
3. တစ်ချို့ database တွေမှာ တော့ kyawko နှင့် koko data တွေသည် Name မှာ ပါဝင်တဲ့ data list ဖြစ်ကြပါတယ် ၊ ဘာကြောင့်လဲဆိုတော့ between operator

သည် AND operator နှင့် သတ်မှတ် ထားသော တန်ဖိုးနှစ်ခုမှ ရှေ့က တန်ဖိုးကိုသာ select လုပ်ပေးပြီ ၊ နောက်ကတန်ဖိုးကို select လုပ်မပေးပါဘူး။

ကျွန်တော် အပေါ်မှာ ဖော်ပြခဲ့တဲ့ အချက်တွေနဲ့ မိမိဆောက်ထားတဲ့ database ကို စစ်ဆေးကြည့်ပါ ကျွန်တော် ဆောက်ထားတဲ့ database မှာ တော့ ဒုတိယ အချက်နဲ့ ညီပါတယ်၊

SQL Joins

Joins keyword သည် table နှစ်ခု (သို့) နှစ်ခုထက်ပိုသော table တွေကို ချိတ်ဆက်အသုံးပြု ချင်တဲ့ အခါမှာ သုံးပါတယ်၊ အဲဒီလို table တစ်ခုနဲ့ တစ်ခု ချိတ်ဆက်အသုံးပြုတဲ့ အခါမှာ table တွေကို ကိုယ်စားပြု key column တွေနဲ့ ချိတ်ဆက်လုပ်ဆောင်ကြပါတယ် ၊ key column တွေသည် row တစ်ခုချင်း စီကို လဲ ကိုယ်စားပြုပါတယ်၊ အဲဒီ key column တွေသည် မည်သည့် data နှင့်မျှ မထပ်သော ဂုဏ်သတ္တိတွေ ရှိကြပါတယ်၊ key column တွေများလားတဲ့ အခါမှာ ဘယ် column က primary ၊ ဘယ် column ကတော့ secondary ဖြစ်သင့်တယ်ဆိုတာကို သတ်မှတ်ပေးရပါမယ်၊ primary column ဆိုတာ row တစ်ခုစီကို ကိုယ် စား ပြုတဲ့ data တွေ မထပ်သော column ကို ခေါ်ဆိုပါသည်၊ ဒီလောက်ဆိုရင် primary သတ်မှတ်ခြင်းသည် table တွေကို ချိတ်ဆက်တဲ့အခါမှာ အရေးပါတယ်ဆိုတာကို သိလောက်ပြီ လို့ ထင်ပါတယ်။

ကျွန်တော် table တွေကို primary သတ်မှတ်တဲ့အခါမှာ server ထဲမှာပါသော tool ကို သုံးပြီးသတ်မှတ်သွားမှာ ဖြစ်ပါတယ် ၊ နောက်ပိုင်းမှာတော့ SQL Query တွေနဲ့ create database ၊ table ၊ data type တွေ primary column သတ်မှတ်ပုံတွေကို ရေးမှာဖြစ်ပါတယ်၊ server ထဲမှာ primary column

သတ်မှတ်ခြင်းရင် □ table -> design > သတ်မှတ်ခြင်းသော column ကို right click -> primary ပေါ့ ။

ကျွန်တော် အောက်မှာ table နှစ်ခုဆောက်ထားပါတယ်၊ အဲဒီ table တွေနဲ့ပဲ inner join ,left join , right join and full join စသော ဥပမာ တွေကို ဖော်ပြ သွားမှာဖြစ်ပါတယ် ၊ အောက်က table မှာ P_id column သည် lesson table ၏ primary column ဖြစ်ပါတယ် ၊ ဒုတိယ table မှာတော့ O_id သည် list table ၏ primary column ဖြစ်ပါတယ်။ list table ထဲမှာ ပါသော P_id column သည် lesson table ကို ကိုယ်စားပြုပါတယ်။

Lesson table

P_id	Name	Phone	Address	Country
1	Iryawko	092581473	TaungDwin	myanmar
2	wanmar	093692581	ngway	myanmar
3	loko	89258369	moscow	russia
4	esi	06258147	Shanghai	China
5	Khine	0936258	yangon	myanmar

Darabase =tutorial ,Table name= lesson ,primary column = P_id

List table

O_id	NRIC	P_id
1	TTK147528	3
2	PK125486	2
3	Y1254785	4
4	mt123654	1
5	lo25836	5

Database =tutorial , tablename = list , primary column = O_id

အပေါ်မှာ ဖော်ပြခဲ့ table နှစ်ခုစီ၏ data တွေ ဆောက်ပြီးသွားပါပြီ ၊ table နှစ်ခုကို ချိတ်ဆက်မဲ့ နည်းလမ်းတွေ အောက်က ခေါင်းစဉ်တွေမှာ ဆွေးနွေးထားပါတယ်။

INNER JOIN Keyword

INNER JOIN Keyword သည် table နှစ်ခုမှ ယှဉ်တွဲလိုရသော row တွေကို ဖော်ပြပေးပါတယ် ၊ ယှဉ်တွဲလိုရသော row တွေကို primary key တွေနဲ့ သတ်မှတ်ထားပါတယ်၊ INNER JOIN သတ်မှတ်သောပုံစံ -->

```
SELECT column_name(s)
FROM table_name1
INNER JOIN table_name2
ON table_name1.column_name= table_name2.column_name
```

select နောက်တွင် table နှစ်ခုလုံးမှာရှိသော column တွေကို ကြေညာပေးလိုရပါတယ် ၊ ကျွန်တော် color တွေနဲ့လဲ ခွဲပေးထားပါတယ်၊

```
SELECT lesson.Name,lesson.Country,list.NRIC
```

```
FROM lesson
```

```
INNER JOIN list
```

```
ON lesson .p_id= list.p_id
```

အောက်က table သည် table နှစ်ခုကို ချိတ်ဆက်ပြီး ရလာသော result table ဖြစ်ပါတယ်။

name	Country	NRIC
kyawko	myanmar	mt123654
warwar	myanmar	PK125486
koko	russia	TTK147528
eiei	China	Y1254785
Khine	mayan...	lo25836

LEFT JOIN Keyword

တစ်ခါတစ်ရံ table နှစ်ခုမှာ ရှိသော row တွေက တူချင်မှ တူပါလိမ့်ပါတယ် ။ အဲဒီလို row တွေမတူတဲ့ table တွေကို ချိတ်ဆက်တဲ့အခါ

စဉ်းစားရမှာဖြစ်ပါတယ်။ inner join က ပြဿနာမရှိပေမဲ့ left join and right join မှာတော့ ပြဿနာရှိလာပါတယ်။ primary column မှာ ရှိတဲ့ row တွေက ဒုတိယ table မှာ မရှိတော့ပေမဲ့ left join နဲ့ ချိတ်ဆက်ထား သောကြောင့် ပထမ table မှာ ရှိသော row တွေကို အကုန်ဖော်ပြမှာဖြစ်ပါတယ်။ အလွယ်မှတ်မယ်ဆိုရင်တော့ ပထမ table မှာ ရှိသော row အားလုံးကိုဖော်ပြချင်တဲ့အခါသုံးပါတယ် ၊ ပထမ table ကတော့ အပေါ်မှာဖော်ပြခဲ့တဲ့ table ပဲဖြစ်ပါတယ်။ left join keyword သတ်မှတ်သောပုံစံ -->

```
SELECT column_name(s)
FROM table_name1
LEFT JOIN table_name2
ON table_name1.column_name=table_name2.column_name
```

ဒုတိယ table name □ test2

O_id	NRIC	P_id
1	1232465	2
2	9876541	4
3	154679	1

```
SELECT lesson .Name ,lesson .Country ,test2.NRIC
```

FROM lesson

LEFT JOIN test2

ON lesson .P_id =test2 .P_id

ဒုတိယ table မှာ မသတ်မှတ်ထားသော NRIC တန်ဖိုးတွေနေရာမှာ NULL လို့ ဖော်ပြပေးထားပါတယ် ၊ ပထမ table မှာ ရှိသော row တွေကို အကုန်လုံးဖော်ပြထားပါတယ်၊အောက်မှာကြည့်လိုက်ပါ --

	Name	Country	NRIC
1	kyawko	myanmar	154679
2	warwar	myanmar	1232465
3	koko	ruusia	NULL
4	eiei	China	9876541
5	Khine	mayanmar	NULL

RIGHT JOIN KEYWORD

ဒုတိယ table မှာ ရှိသော row အားလုံးကိုဖော်ပြချင်တဲ့အခါသုံးပါတယ် ၊ ပထမ table ကတော့ အပေါ်မှာဖော်ပြခဲ့တဲ့ table ပဲဖြစ်ပါတယ်၊ right join keyword သတ်မှတ်သောပုံစံ သည် left နေရာမှာ right ဖြစ်ပါတယ်၊ အပေါ်မှာ သတ်မှတ်ခဲ့တဲ့ table တွေကိုပဲ အသုံးပြုထားပါတယ်။

```
SELECT lesson .Name ,lesson .Country ,test2.NRIC
FROM lesson
RIght JOIN test2
ON lesson .P_id =test2 .P_id
```

Name	Country	NRIC
eiei	China	9876541
kyawko	myanmar	154679
Khine	mayanmar	4567811
koko	russia	785649
NULL *	NULL *	251478
NULL *	NULL *	25974

FULL JOIN Keyword

Full join keyword သည် table နှစ်ခုလုံးမှာ ရှိသော row တွေအားလုံးကို result table တစ်ခုတည်းမှာ စုစည်းပြီး အားလုံးဖော်ပြပေးပါတယ်။ဒါပေမဲ့ primary key သတ်မှတ်ထားသော row တွေအားလုံးကိုပဲဖော်ပြပေးတာဖြစ်ပါတယ်။

```
SELECT lesson .Name ,lesson .Country ,lesson.Phone ,lesson.Address ,test2.NRIC
FROM lesson
full JOIN test2
```

SQL လေ့လာခြင်း နှင့် လက်တွေ့ အသုံးချ သင်ခန်းစာများ

www.akarphyoe.com

ON lesson .P_id =test2 .P_id

Name	Country	Phone	Address	NRIC
warwar	myanmar	093692581	mgway	1232465
koko	russia	89258369	moscow	785649
eiei	China	06258147	Shanghai	9876541
Khine	mayanmar	0936258	yangon	4567811
NULL	NULL	NULL	NULL	251478
NULL	NULL	NULL	NULL	25974

ပထမ table တွင် row (4) ခုရှိပါတယ် ၊ ဒုတိယ table မှာ row (6) ခု ရှိပါတယ် ၊ full join နှင့် ဆက်လိုက်တဲ့ အခါမှာ တော့ row အားလုံးကိုဖော်ပြပေးပါတယ်။ တန်းဖိုး မရှိသော နေရာတွေကို NULL တန်ဖိုးဖြင့် သတ်မှတ်ပေးပါတယ်။

CEARTE DATABASE and CREATE TABLE

Server ထဲပါသော tool ကို အသုံးမပြုပဲနဲ့ sql statement နှင့် database တွေ table တွေ ၊ primary column တွေ data type တွေကို သတ်မှတ်လို့ရပါတယ်၊ သတ်မှတ်သောပုံစံ -->

CREATE DATABASE database_name

CREATE TABLE table_name

Table မှာ column တွေ data type တွေသတ်မှတ်လို့ရပါတယ် ၊ column တွေ data type တွေက ကွင်းစ ကွင်းပိတ်ထဲမှာ ရှိရပါမယ်။ column တစ်ခုသတ်မှတ်ပြီးတိုင် ကော်မာ (,) ခံပေးရပါတယ် ကျွန်တော် ဒီခေါင်းစဉ်အောက်မှာတော့ data type တွေ အကြောင်း ကို ချန်ခဲ့မှာ ဖြစ်ပါတယ်။ table တစ်ခု ဖန်တီးကြည့်ရအောင် -->

```
CREATE TABLE test3
(
  K_No int,
  Name varchar(255),
  Phone varchar(255),
  Address varchar(255),
  Country varchar(255)
)
```

Table မှာ ပါဝင်သော column တွေ ဖန်တီးပြီးသွားပါပြီ ၊ column တွေထဲမှာ သတ်မှတ်သော data တွေကို INSERT INTO , UPDATE စသော statement တွေနဲ့ ထည့်သွင်းပေးလို့ရပါတယ်။ နောက်ထပ်ခေါင်း စဉ်တွေမှာ SQL constraints တွေအကြောင်းကို ဖော်ပြထားပါတယ်။

NULL

Table ၏ column တွေမှာ null တန်ဖိုးတွေ ပါဝင်နေပါတယ် ၊ အဲဒီ NULL တန်ဖိုးတွေသည် မရှိသေးတဲ့ data တွေကို ကိုယ်စားပြုပါတယ် ။ ဆိုလိုတာက -

ကျွန်တော်တို့တွေက column တစ်ခုမှာ ထည့်သွင်းရမည့် data ကို မေ့နေတယ် ၊ အဲဒီ data ကို နောက်မှ ထည့်မည်ဆိုပြီး ထားခဲ့လိုက်တယ်၊ ဒါပေမဲ့ SQL server ကတော့ ဘာမှ ထည့်မထားတဲ့ record ကို NULL တန်ဖိုး အဖြစ် ထည့်ပေးမှာ ဖြစ်ပါတယ်၊ အရင် ခေါင်းစဉ်တွေက table မှာလဲ တွေ့ခဲ့ပြီးပြီးဆိုတော့ မဖော်ပြတော့ပါဘူး။ table ထဲမှ NULL တန်ဖိုးတွေ သတ်မှတ်ထားလား (သို့) သတ်မှတ်မထားဘူးလား ဆိုတာကို စစ်ဆေးမည့် တန်ဖိုးတွေအကြောင်းကို လေ့လာမှာ ဖြစ်ပါတယ်။

IS NULL

IS NULL သည် table ထဲမှ NULL တန်ဖိုးပါဝင်သော column တွေကို Select လုပ်မှာဖြစ်ပါတယ်။ အောက်က table ကိုနမူထား ၍ ရှင်းပြမှာ ဖြစ်ပါတယ်။

k_no	Name	Address	City
1	kyawko	NULL	Taung Dwin
2	warwar	Shwe O	Taung Dwin
3	koko	NULL	Taung Dwin
4	Min Min	Shwe kyar	Taung Dwin

အပေါ်က table မှ NULL တန်ဖိုးကို SELECT လုပ်မှာ ဖြစ်ပါတယ်။

```
SELECT k_no ,Name,Address FROM koko
WHERE Address IS NULL
```

k_no	Name	Address
1	kyawko	NULL
3	koko	NULL

IS NOT NULL

NULL တန်ဖိုး မဟုတ်သော data တွေကို select လုပ်မှာ ဖြစ်ပါတယ်။

```
SELECT k_no ,Name,Address FROM koko  
WHERE Address IS NOT NULL
```

k_no	Name	Address
2	warwar	Shwe O
4	Min Min	Shwe kyar

SQL constraints

Constraints လုပ်တယ်ဆိုတာ table ထဲသို့ ပေါင်းထည့်မဲ့ data တွေ၏ data type တွေကို ကန့်သတ်ခြင်းဖြစ်ပါတယ်။ table ထဲမှာ ကန့်သတ်လို့ရမဲ့ ပုံစံတွေကို အောက်မှာ ဖော်ပြထားပါတယ်။

NOT NULL

UNIQUE

PRIMARY KEY

FOREIGN KEY

CHECK

DEFAULT

NULL constraint

Table ၏ column တွေမှာ null တန်ဖိုးတွေ ပါဝင်နေပါတယ် ၊ အဲဒီလို ပါဝင်နေမှုကို မလိုချင်ဘူး ဆိုရင်တော့ not null constraint ကို အသုံးပြုရပါတယ် ၊ not null သည် null တန်ဖိုးတွေ ကို column တွေမှာ လက်မခံရန် ကန့်သတ်ထားပါတယ်၊ not null သတ်မှတ်ထားသော column သည် row တွေ ထပ်ပေါင်းထည့်ခြင်းကို ခွင့်မပြုပါဘူး ၊ ပို၍ နားလည်အောင် အောက်က ဥပမာ ကိုလေ့လာကြည့် ပါ ။

```
CREATE TABLE kyawko  
(K_No int NOT NULL,  
Last_Name varchar (30) NOT NULL,  
First_Name varchar(30));
```

အပေါ်က table ၏ column မှ lastname , K_No ကို Not Null အဖြစ်သတ်မှတ်ထားပါ ၊ ကျွန်တော် တို့က data တွေ row တွေ ထည့်ခြင်းတို့ အခါမှာ null တန်ဖိုးသတ်မှတ်ထားရင် ပြဿနာမရှိပေမဲ့ not null ဆိုရင်တော့ အသစ်ထည့်သော row တွေ data တွေကို လက်ခံမှာ မဟုတ်ပါဘူး ၊ ဘာကြောင့်လဲဆိုတော့ စစ်ချင်း column မှာ not null ဖြစ်သောကြောင့် နောက်ထပ် column တွေကိုလဲ not null အဖြစ်သတ်မှတ် လိုက် ပါ တယ် ၊ ပြောရမယ်ဆိုရတော့ အဲဒီ table မှ စစ်ချင်း column က ဦးဆောင်နေပါတယ် ။

```
INSERT INTO kyawko (Last_Name, First_Name) values ('ma','Khine');
```

INSERT INTO နဲ့ values တွေ ထည့်တဲ့ အခါမှ sql server က လက်မခံသော message ကို ရရှိမှာ ဖြစ်ပါတယ် ။

SQL Constraints

Unique constraint သည် database table ထဲမှ record တစ်ခုစီ အတွက် တူမူထူးခြားသော (သို့) မတူညီသော တန်ဖိုးတွေ ကို သတ်မှတ်ခြင်းဖြစ်ပါတယ်။ unique constraint နှင့် primary key constraint နှစ်ခုလုံးသည် unique တန်ဖိုးတွေနှင့် သတ်မှတ်ကြပါတယ် ၊ primary key constraint သည် unique constraint တွေ အလိုလျှောက် ရှိကြပါတယ် ၊ unique တန်ဖိုးတွေ သတ်မှတ်တိုင်း primary key မဖြစ်ပါဘူး ။

```
CREATE TABLE student1  
  
(  
  
P_Id int NOT NULL unique,  
  
Name varchar(255) NOT NULL,  
  
Address varchar(255),  
  
City varchar(255)  
  
);
```

အပေါ်က ဥပမာ တွင် P_id ကို unique အဖြစ် သတ်မှတ်ထားသောကြောင့် P_id column မှာ data တွေ ထပ်နေလျှင် error message ပြပါလိမ့်မယ် ။

SQL PRIMARY KEY Constraint

Table တွေ မှာ column တွေ ရှိကြပါတယ် ၊ အဲဒီ column တွေမှာ တန်ဖိုးတွေ ပါဝင်ပါတယ် ၊ အဲဒီတန်ဖိုးတွေပါဝင်သော row တစ်ခုခြင်းစီမှာလဲ မတူညီသော တန်ဖိုးတွေပါဝင်ကြပါတယ်၊ အဲဒီလို မတူညီသော တန်ဖိုးတွေ ပါဝင်သော columns တွေကို primary key ဟုခေါ်ပါတယ်၊ ကျွန်တော်တို့ဟာ table တစ်ခုကို ဖန်တီးချိန်မှာ primary key constraint ကို သတ်မှတ်လိုက်ခြင်းဖြင့် primary key ဖန်တီးလိုက်ခြင်းပဲဖြစ်ပါတယ်၊ table တစ်ခုမှာ primary key တစ်ခုသာ သတ်မှတ်လို့ရပါတယ်၊ primary key constraint သတ်မှတ်ထားသော column သည် NULL တန်ဖိုးတွေကို လက်ခံလို့ မရပါဘူး ၊ ဘာကြောင့်လဲဆိုတော့ primary key constraint သည် မတူညီသော(unique data) တန်ဖိုးတွေကိုသာ သတ်မှတ်ခွင့်ပြုပါတယ်၊ table အတွက် primary key constraint သတ်မှတ်လိုက်တဲ့ အချိန်မှာ Database Engine သည် primary key column အတွက် unique index ဖန်တီး လိုက်ခြင်းဖြင့် တူညီသော data တွေမှာ ကန့်သတ်ချက်တွေ ရှိပြီး ဖြစ်နေပါလိမ့်မယ်၊ primary key ကို queries ထဲမှာ အသုံးပြုလိုက်တဲ့အချိန် တွင် unique index တွေသည် data တွေကို လျင်မြန်စွာနဲ့ ရှာဖွေနိုင်မှာဖြစ်ပါတယ်၊ primary key constraint သတ်မှတ်သော ပုံစံသည် server တွေအလိုက် ကွာပြားခြားနားမှုတွေရှိပါတယ်၊ sql server မှာ သတ်မှတ်သောပုံစံ -->

```
CREATE TABLE Student
(
P_Id int NOT NULL PRIMARY KEY,
Name varchar(255) NOT NULL,
FirstNama varchar(255),
Address varchar(255),
Country varchar(255)
)
```

SQL FOREIGN KEY Constraint

SQL လေ့လာခြင်း နှင့် လက်တွေ့ အသုံးချ သင်ခန်းစာများ

www.akarphyoe.com

Foreign key သည် column တစ်ခု (သို့) column တွေကို ပေါင်းစပ်ထားခြင်းဖြစ်ပါတယ်။ အဲဒီ column တွေသည် table နှစ်ခုတွင် ရှိသော data တွေ၏ ကြားမှာ ချိတ်ဆက်ပေးသော link တွေဖြစ်ပါတယ်။ table တစ်ခုမှာ foreign key constraints သတ်မှတ်လိုက်ခြင်းဖြင့် foreign key သတ်မှတ်ပြီးသား ဖြစ်နေပါလိမ့်မယ်။ table တစ်ခုအတွက် primary key သတ်မှတ်ထားသော column သည် အခြား table တစ်ခု၏ column ကို ချိတ်ဆက်နေတဲ့ အချိန်တွင် အဲဒီ table နှစ်ခုကြားမှာ link ဖြစ်ပေါ်နေပါတယ်။ အဲဒီ link ဖြစ်နေသော column တွေသည် အခြား table တစ်ခု၏ foreign key ဖြစ်နေပါတယ်။ sql server သတ်မှတ်သောပုံစံ -->

```
CREATE TABLE list
(
O_Id int NOT NULL PRIMARY KEY,
NRIC int NOT NULL,
P_Id int FOREIGN KEY REFERENCES lesson(P_Id)
)
```

ကျွန်တော် join table ခေါင်းစဉ်မှာလဲ ရှင်းပြခဲ့ပါတယ် ၊ အောက် ကtable နှစ်ခုကို ကြည့်လိုက်ပါ ။

lesson

P_id	Name	Phone	Address	Country
1	Kyawko	092581471	TaungDwin	myanmar
2	wannar	093692581	ngway	myanmar
3	loko	89258369	moscow	russia
4	eel	06258147	Shanghai	China
5	Khine	0936258	yangon	myanmar

List table

O_id	NRIC	P_id
1	TTK147528	3
2	PK125486	2
3	Y1254785	4
4	mt123654	1
5	lo25836	5

P-id သည် lesson table ၏ primary key ဖြစ်ပါတယ် ၊ O_id သည် list table ၏ foreign key ဖြစ်ပါတယ် ၊ list table မှ P_id သည် lesson table မှ P_id ကို ချိတ်ဆက်ထားပါတယ်။

SQL CHECK Constraint

Check constraint သည် column တွေထဲမှာ ရောက်လာသော တန်ဖိုးတွေ ကို အစီစဉ်တကျနဲ့ ကန့်သတ်ပေးပါတယ် ၊ Check constraint သည် column ထဲမှာ သတ်မှတ်မဲ့ တန်ဖိုးတွေက valid ဖြစ်မဖြစ်ကို စစ်ဆေး ပါတယ် ၊ valid ဖြစ်မဖြစ်ကို logical expression (true or false)တန်ဖိုးတွေ နဲ့ စစ်ပါတယ်၊ logical expressin တန်ဖိုးက false ဖြစ်မယ်ဆိုရင် column ထဲသို့ data insert or updatae

လုပ်လို့ရမှာမဟုတ်ပါဘူး။ ALTER table statement တွေနဲ့ တွဲ၍ အသုံးပြုနိုင်ပါတယ်။
သတ်မှတ်သောပုံစံ-->

Column_Name Datatype (size) CHECK (Logical_Expression)

အောက်က ဥပမာတွင် p_id column ကို check လုပ်ထားပါတယ်။ p_id column ထဲကို သတ်မှတ်မည့် data တွေသည် သုည (integer) ထက်ကြီးသော တန်ဖိုးတွေပဲ ဖြစ်ရမည်။ အဲဒီလို မဟုတ်ဘဲနဲ့ သုည ထက်ငယ်နေတယ်ဆိုရင်တော့ လက်ခံမှာ မဟုတ်ပါဘူး။

```
CREATE TABLE Persons
(
P_Id int NOT NULL CHECK (P_Id>0),
Name varchar(255) NOT NULL,
phone varchar(255),
Address varchar(255),
Country varchar(255)
)
```

SQL DEFAULT Constraint

Default constraint သည် column တစ်ခုထဲသို့ default တန်ဖိုးတွေ ထည့်သွင်းရန် အသုံးပြုပါတယ်။ column တစ်ခုမှာ တန်ဖိုးတွေသတ်မှတ်မထားဘူးဆိုရင် default တန်ဖိုးသတ်မှတ်ပေးခြင်းဖြင့် new record အားလုံးကို အဲဒီ default တန်ဖိုးတွေ ပေါင်းထည့်ပေးပါလိမ့်မယ်။

ဆိုလိုတာက -- column တစ်ခုအတွက် တူညီသောတန်ဖိုးတစ်ခုထည်း သတ်မှတ်ပေးခြင်းဖြစ်ပါတယ်။

ဥပမာ - table ၏ column တစ်ခုမှာ ပါဝင်သော data တွေက အားလုံးတူညီနေမယ်ဆိုရင် default value သတ်မှတ်ပေးခြင်းဖြင့် အချိန်ကုန်သက်သာစေပါတယ်။ default value သတ်မှတ်သော ပုံစံ -->

```
CREATE TABLE student
(
P_Id int NOT NULL,
Name varchar(255) NOT NULL,
Address varchar(255),
City varchar(255) DEFAULT 'TaungDwingyi'
)
```

အပေါ်က ဥပမာတွင် City ကို default value သတ်မှတ်ပေးထားသောကြောင့် column City ကို edit လုပ်တဲ့ အခါ NULL တန်ဖိုးတွေ နေရာယူမှာ ဖြစ်ပါတယ်။ sql query

လုပ်မှ default values တွေ ပေါင်းထည့်ပေးမှာဖြစ်ပါတယ်။ ရလာသော table ကို လေ့လာ ကြည့်ပါ။

P_Id	Name	Address	City
1	kyawko	ng min road	TaungDwingyi
2	warwar	khine tazin road	TaungDwingyi
3	winthat	shweO2	TaungDwingyi
4	Su Pyi Moe	Taung Pyin road	TaungDwingyi

SQL CREATE INDEX Statement

Insex statement သည် table ထဲမှ အညွှန်းခေါင်းစဉ်တွေကို ဖန်တီးရန် အသုံးပြုပါတယ်။ ဖန်တီးထားသော index တွေကို user တွေက မြင်လို့ရမှာ မဟုတ်ပါဘူး။ ဘာကြောင့်လဲဆိုတော့ index statement တွေသည် data တွေကို ရှာရန်နှင့် query တွေကို မြန်မြန်လုပ်ဆောင် နိုင်ရန်အတွက်ဖြစ်ပါတယ်။ သတ်မှတ်သောပုံစံ-->

CREATE INDEX index_name
ON table_name (column_name)

databasename= tutorial , tablename= lesson

CREATE INDEX namelist

ON lesson (Name)

Column တွေအများကြီးကို index လုပ်ချင်တယ်ဆိုရင် column name တွေကို ကော်မာ(,)ခြားပြီး လုပ်နိုင်ပါတယ်။

DROP Statement

Drop statement သည် ဖန်တီးပြီးသား database , table ,column and index တွေကို လွယ်ကူစွာနဲ့ ဖျက်နိုင်ပါတယ် ။

Table ထဲမှ index တွေကို ဖျက်ရန် --> DROP INDEX
table_name.index_name

Table ကိုဖျက်ရန် --> DROP TABLE table_name

Database ကို ဖျက်ရန်--> DROP DATABASE database_name

Column ကို ဖျက်ရန်--> DROP COLUMN column_Name (alter statement နှင့်တွဲသုံးရန်)

ALTER Staement

Alter statement သည် ရှိပြီးသား table ထဲမှ column တွေကို ပြုပြင်နိုင်တယ် ၊ပေါင်းထည့်နိုင်တယ် (သို့) ဖျက်နိုင်ပါတယ်။

Columns ပေါင်းထည့်ရန် သတ်မှတ်သောပုံစံ -->

ALTER TABLE table_name

ADD column_name data type

အောက်က ဥပမာသည် student table ထဲသို့ column အသစ် ပေါင်းထည့်ထားသော ပုံစံဖြစ်ပါတယ်။ အဲဒီလို ပေါင်းထည့်တဲ့ အခါမှာ data type ကို သတ်မှတ်ပေးရမှာဖြစ်ပါတယ်။ အောက်မှာ ဖော်ပြထားသော table ထဲသို့ Name column ပေါင်းထည့်မှာ ဖြစ်ပါတယ်။

P_Id	Address	City
1	ng min road	TaungDwingyi
2	khine tazin road	TaungDwingyi
3	shweO2	TaungDwingyi
4	Taung Pyin road	TaungDwingyi

ALTER TABLE student

ADD Name varchar

ပေါ်လာသော table ၏ name column ကို ကြည့်ပါ။

P_Id	Address	City	Name
1	ng min road	TaungDwingyi	NULL
2	khine tazin road	TaungDwingyi	NULL
3	shweO2	TaungDwingyi	NULL
4	Taung Pyin road	TaungDwingyi	NULL
NULL	NULL	NULL	NULL

Column မှ datatype ပြောင်းလဲခြင်း

တစ်ခါတစ်လေကျရင် data တွေထည့်သွင်းအခါ မှာ အရင် သတ်မှတ်ထားသော data type တွေကြောင့် ထည့်သွင်းလို့ မရပါဘူး ၊အဲဒီလိုမျိုးဖြစ်လာပြီဆိုရင် data type တွေပြင် ဖို့လို့လာပါပြီ ၊ data type တွေပြင်တော့မယ်ဆိုရင် ALTER statement ကို အသုံးပြုပါရပါမည်။ သတ်မှတ်သောပုံစံ-->

ALTER TABLE table_name

ALTER COLUMN column_name datatype

အောက်က ဥပမာသည် student1 table မှ Name column ၏ data type ကို ပြောင်းလဲထားပါတယ် ၊ column မှာ အရင် data တွေရှိနေတယ် ဆိုရင် တော့ ပြဿနာ ရှိပါတယ် ၊ဘာဖြစ်လို့လဲဆိုတော့ အရင် ရှိနေတဲ့ data type နှင့် နောက်သတ်မှတ်သော data type တွေက မတူသောကြောင့်ဖြစ်ပါတယ်။

ALTER TABLE student1

ALTER COLUMN Name Text

DROP COLUMN

SQL လေ့လာခြင်း နှင့် လက်တွေ့ အသုံးပြု သင်ခန်းစာများ

www.akarphyoe.com

table ထဲမှာ column တွေကို ဖျက်ချင်တဲ့ အခါမှာ DROP ကို အသုံးပြုပါတယ်။ DROP ခေါင်းစဉ်မှာလဲ မြင်ဖူးမှာပါ။ သတ်မှတ်သောပုံစံ -->

ALTER TABLE **table_name**

DROP COLUMN **column_name**

databasename=tutorial ,tablename=stdnt1 ဖြစ်ပါတယ် ။

ALTER TABLE student1

DROP COLUMN Name

SQL CREATE INDEX Statement

Insex statement သည် table ထဲမှ အညွှန်းခေါင်းစဉ်တွေကို ဖန်တီးရန် အသုံးပြုပါတယ်။ ဖန်တီးထားသော index တွေကို user တွေက မြင်လို့ရမှာ မဟုတ်ပါဘူး။ ဘာကြောင့်လဲဆိုတော့ index statement တွေသည် data တွေကို ရှာရန်နှင့် query တွေကို မြန်မြန်လုပ်ဆောင် နိုင်ရန်အတွက်ဖြစ်ပါတယ်။ သတ်မှတ်သောပုံစံ-->

CREATE INDEX **index_name**
ON **table_name** (column_name)

databasename= tutorial , tablename= lesson

CREATE INDEX **namelist**
ON **lesson** (Name)

Column တွေအများကြီးကို indexed လုပ်ချင်တယ်ဆိုရင် column name တွေကို ကော်မာ(,)ခြားပြီး လုပ်နိုင်ပါတယ်။

DROP Statement

Drop statement သည် ဖန်တီးပြီးသား database , table ,column and index တွေကို လွယ်ကူစွာနဲ့ ဖျက်နိုင်ပါတယ် ။

Table ထဲမှ index တွေကို ဖျက်ရန် --> DROP INDEX
table_name.index_name

Table ကိုဖျက်ရန် --> DROP TABLE **table_name**

Database ကို ဖျက်ရန်--> DROP DATABASE **database_name**

Column ကို ဖျက်ရန်--> DROP COLUMN column_Name (alter statement နှင့်တွဲသုံးရန်)

ALTER Statement

Alter statement သည် ရှိပြီးသား table ထဲမှ column တွေကို ပြုပြင်နိုင်တယ် ၊ပေါင်းထည့်နိုင်တယ် (သို့) ဖျက်နိုင်ပါတယ်။

Columns ပေါင်းထည့်ရန် သတ်မှတ်သောပုံစံ -->

ALTER TABLE **table_name**

ADD **column_name data type**

အောက်က ဥပမာသည် student table ထဲသို့ column အသစ် ပေါင်းထည့်ထားသော ပုံစံဖြစ်ပါတယ်။ အဲဒီလို ပေါင်းထည့်တဲ့ အခါမှာ data type ကို သတ်မှတ်ပေးရမှာဖြစ်ပါတယ်။ အောက်မှာ ဖော်ပြထားသော table ထဲသို့ Name column ပေါင်းထည့်မှာ ဖြစ်ပါတယ်။

P_Id	Address	City
1	ng min road	TaungDwingyi
2	khine tazin road	TaungDwingyi
3	shweO2	TaungDwingyi
4	Taung Pyin road	TaungDwingyi

ALTER TABLE student

ADD Name varchar

ပေါ်လာသော table ၏ name column ကို ကြည့်ပါ။

P_Id	Address	City	Name
1	ng min road	TaungDwingyi	NULL
2	khine tazin road	TaungDwingyi	NULL
3	shweO2	TaungDwingyi	NULL
4	Taung Pyin road	TaungDwingyi	NULL
NULL	NULL	NULL	NULL

Column မှ datatype ပြောင်းလဲခြင်း

တစ်ခါတစ်လေကျရင် data တွေထည့်သွင်းအခါ မှာ အရင် သတ်မှတ်ထားသော data type တွေကြောင့် ထည့်သွင်းလို့ မရပါဘူး ၊အဲဒီလိုမျိုးဖြစ်လာပြီဆိုရင် data type တွေပြင် ဖို့လို့လာပါပြီ ၊ data type တွေပြင်တော့မယ်ဆိုရင် ALTER statement ကို အသုံးပြုပါရပါမည်။ သတ်မှတ်သောပုံစံ-->

ALTER TABLE **table_name**

ALTER COLUMN **column_name** **datatype**

အောက်က ဥပမာသည် student1 table မှ Name column ၏ data type ကို ပြောင်းလဲထားပါတယ် ၊ column မှာ အရင် data တွေရှိနေတယ် ဆိုရင် တော့ ပြဿနာ ရှိပါတယ် ၊ဘာဖြစ်လို့လဲဆိုတော့ အရင် ရှိနေတဲ့ data type နှင့် နောက်သတ်မှတ်သော data type တွေက မတူသောကြောင့်ဖြစ်ပါတယ်။

ALTER TABLE student1

ALTER COLUMN Name Text

DROP COLUMN

table ထဲမှာ column တွေကို ဖျက်ချင်တဲ့ အခါမှာ DROP ကို အသုံးပြုပါတယ်။ DROP ခေါင်းစဉ်မှာလဲ မြင်ဖူးမှာပါ။ သတ်မှတ်သောပုံစံ -->

ALTER TABLE **table_name**

DROP COLUMN **column_name**

databasename=tutorial ,tablename=stdnt1 ဖြစ်ပါတယ် ။

ALTER TABLE student1

DROP COLUMN Name

SQL Date

Date တွေကို table column တွေထဲမှာ timestamp ပုံစံတွေအနေနဲ့ သိမ်းဆည်းပါတယ်။ SQL timestamp တွေမှာ လ၊ ရက်၊ နှစ်၊ နာရီ၊ မိနစ်၊ စက္ကန့်၊ စသော ပုံစံတွေနဲ့ အစဉ်လိုက် ဖွဲ့စည်းထားပါတယ်။ SQL timestamp သတ်မှတ်သောပုံစံသည် standard date သတ်မှတ်သောပုံစံတွေနှင့် အတူတူပဲဖြစ်ပါတယ်။ database table ထဲမှာ data and time တန်ဖိုးတွေကို သတ်မှတ်တော့မယ်ဆိုရင် အောက်ကပုံစံတွေအတိုင်း သတ်မှတ်နိုင်ပါတယ် ။

DATE - format YYYY-MM-DD

DATETIME - format: YYYY-MM-DD HH:MM:SS

SMALLDATETIME - format: YYYY-MM-DD HH:MM:SS

TIMESTAMP - format: a unique number

Sql server မှာပါသော built_in function တွေအကြောင်းကို အရင်လေ့လာမှာဖြစ်ပါတယ် ၊ ပြီးမှ SQL Query နှင့် အသုံးပြုပုံကို

လေ့လာမှာဖြစ်ပါတယ်။ ကျွန်တော် table တစ်ခုတည်းနဲ့ built_in function တွေအကြောင်း ဖော်ပြသွားမှာဖြစ်ပါတယ် ၊ ထပ်ခါထပ်ခါ ဖော်ပြနေရမှာဆိုလို့ ပါ။ built_in function တွေကို အသုံးပြုခြင်းဖြင့် row အ သစ်တွေထပ်တိုးလာတိုင်း အလိုလျောက် date တွေကို ဖြည့်ပေးသွားမှာပါ။ date function တွေသည် ကွန်ပျူတာမှာ ပါသော timestamp ပုံစံအတိုင်းဖော်ပြပေးမှာဖြစ်ပါတယ်။

GETDATE() function

GETDATE() function သည် sql server မှ date and time ကို ဖော်ပြပေးပါတယ် ၊ ဖော်ပြပေးသော date ပုံစံမှာ à 2012-12-05 02:26:45.443 ဖြစ်ပါတယ် ။ ကျွန်တောတို့ ဥပမာတွေနဲ့ ကြည့်လိုက်ရအောင် -- အောက်က table မှ class datetime column ကို getdate() function ကြေညာထားပါတယ် ၊ table ထဲသို့ row တွေ ပေါင်းထည့်သော အချိန်ကို အလိုလျောက်သတ်မှတ်ပေးမှာဖြစ်ပါတယ်။ datetime သည် date တွေအတွက် datatype ဖြစ်ပါတယ်။

```
CREATE TABLE student
(
No int NOT NULL PRIMARY KEY,
Name varchar(50) NOT NULL,
class datetime NOT NULL DEFAULT GETDATE()
)
```

Table ဆောက်ပြီးပြီးဆိုတော့ data ထည့်သွင်းကြည့်ရအောင် ။

```
INSERT INTO student4(st_no, Name) VALUES (3,'War War')
```

st_no	Name	class
1	kyawko	01-05 03:19:41.533
2	Khing	2012-01-05 03:...
3	War War	2012-01-05 03:...

Datepart() function

Datepart() function သည် table ထဲမှာ သတ်မှတ်ထားသော date တွေရဲ့ အစိတ်ပိုင်း(နာရီ၊မိနစ်၊စက္ကန့်.)

စသည်တို့ကို အသေးစိတ်ပြန်လည်ဖော်ပြချင်တဲ့အခါမှာ အသုံးပြုပါတယ် ၊

datepart ()function မှာ arguments (နှစ်ခု)ပါဝင်ရပါတယ် ။

သတ်မှတ်သောပုံစံ --> DATEPART(datepart,date)

1. Datapart = date expression
2. Date = argument (column name)

အောက်မှာဖော်ပြထားသော date expression တွေကို datepart() function မှာ အသုံးပြုနိုင်ပါတယ်။

datepart	Abbreviation
year	yy,yyyy,year
quarter	qq,q

month	mm,m
datofyear	dy,y
day	day,
week	wk,ww
Weekday	dw,w
Weekday	dw,w
hour	hh
minutes	mi,n
second	ss,s
millisecond	ms
microsecond	mcs

အပေါ်မှာ ဖော်ပြခဲ့တဲ့ datepart သတ်မှတ်သောပုံစံနဲ့ select လုပ်ပုံကို အရင်လေ့လာကြည့်ရအောင်။

SELECT

DATEPART(YEAR,'2012-5-12') AS 'OrderYear'

အပေါ်က ('2012-5-12') မှာ Year ကို select လုပ်ပြထားပါတယ်။

	OrderYear
1	2012

table ထဲမှာ သတ်မှတ်ထားသော date တွေကို datepart() function သုံးပြီး select လုပ်ပါမယ်။ table ကိုတော့ student4 table ကို သုံးမှာဖြစ်ပါတယ်။

```
SELECT DATEPART(YEAR ,class) AS Year,  
DATEPART(MINUTE ,class) AS Month,  
DATEPART(DAY ,class) AS Day,  
Name FROM student4  
WHERE st_no=1
```

ပေါ်လာသော table --

	Year	Month	Day	Name
1	2012	19	5	kyawko

DateADD() function

DateADD() function သည် table ထဲမှာ သတ်မှတ်ထားသော date တွေကို ပြောင်းလဲပြင်ဆင်နိုင်ပါတယ်။ DateADD() function မှာ သော argument 3 ခုရှိပါတယ် ၊ သတ်မှတ်သောပုံစံ --> DATEADD(datepart,number,date)

1. Number = ထပ်ပေါင်း (သို့) နုတ်ခြင်သော အရေအတွက်ဖြစ်ပါတယ် ၊ နုတ်ခြင်တယ်ဆိုရင်တော့ (-) ထည့်ပေးလိုက်ပေါ့ဗျာ။ အောက်က ဥပမာသည် 2012 ကို 2014 ဖြစ်အောင် ပေါင်းထည့်ထားတာဖြစ်ပါတယ်။

```
SELECT DATEADD(year, 2, 2012-5-12) AS AddoneYear;
```

	AddoneYear
1	1907-06-19 00:00:00.000

table ထဲမှာ သတ်မှတ်ထားသော date တွေကို dateadd() function သုံးပြီး select လုပ်ပါမယ်။ table ကိုတော့ student4 table ကို သုံးမှာဖြစ်ပါတယ်။ အောက်က code တွေသည် student4 table ထဲမှာ သတ်မှတ်ထားသော date 2012 ကို 2016 ဖြစ်အောင် ပြောင်းလဲသတ်မှတ်မှာဖြစ်ပါတယ်။

```
SELECT st_no,DATEADD(YEAR,4,class) AS OrderPayDate
```

```
FROM student4
```

st_no	OrderPayDate
1	2016-01-05 03:19:41.533
2	2016-01-05 03:19:51.237
3	2016-01-05 03:21:16.317

DateDIFF()function

dateDIFF()function သည် သတ်မှတ်ထားသော date နှစ်ခုရဲ့ကြား time ကို ဖော်ပြပေးပါတယ်။သတ်မှတ်သောပုံစံ-->
DATEDIFF(datepart,startdate,enddate)

```
SELECT DATEDIFF(yy,'2012-05-12','2011-08-05') AS betweenDate
```

	betweenDate
1	-1

SQL Function

SQL server တွင် data တွေ၏လုပ်ဆောင်ချက်တွေ return ပြန်ပေးခြင်း (သို့) တွက်ချက်နိုင်သော built-in function တွေအများကြီး ပါဝင်ပါတယ်။ လေးမျိုးထဲကမှ Aggregate Functions နှင့် Scalar Functions အကြောင်းကို ရေးသားမှာ ဖြစ်ပါတယ်။ sql function type လေးမျိုးရှိပါတယ် ၊

1. Rowset Functions
2. Aggregate Functions
3. Ranking Functions
4. Scalar Functions

Aggregate Functions

Aggregate Functions သည် တန်ဖိုးတွေကို တွက်ချက်ပေးပါတယ် ၊ ဒါပေမဲ့ ရလာသောတန်ဖိုးတွေရဲ့ တစ်ခုကိုသာ ဖော်ပြပေးပါတယ်။ Aggregate Functions တွင်ပါဝင်သော function များ --

- 1.AVG() - Returns the average value
- 2.COUNT() - Returns the number of rows
- 3.FIRST() - Returns the first value
- 4.LAST() - Returns the last value
- 5.MAX() - Returns the largest value
- 6.MIN() - Returns the smallest value
- 7.SUM() - Returns the sum

AVG() function

AVG() function သည် ကိန်းဂဏန်းတွေသတ်မှတ်ထားသော column ၏ ပျမ်းမျှတန်ဖိုးကို တွက်ချက်ဖော်ပြပေး ပါတယ်။ ဥပမာ အားလုံးကို book table တစ်ခုတည်းနဲ့ ဖော်ပြသွားမှာဖြစ်ပါတယ်။ AVG() သတ်မှတ်သောပုံစံ □

SELECT AVG(column_name) FROM table_name

အောက်မှာဖော်ပြထားသော Book table မှ စာအုပ်တွေ၏ ရောင်းရငွေကို ပျမ်းမျှရှာပေးသောပုံစံဖြစ်ပါတယ်၊

b_no	Name	book	Price
1	kyawko	homebook	500
2	warwar	love bird	1500
3	Khine	Love Story	528
4	Ei Zar	loneliness	750

SELECT **AVG**(Price) **AS** bookAverage **FROM** Book

	bookAverage
1	819.5

နောက်ထပ်ဥပမာတစ်ခုကတော့ - ပျမ်းမျှတန်ဖိုးထက် များသော customer name ကို ရှာမှာဖြစ်ပါတယ်။ ဒါဆိုရင် တန်ဖိုးနှစ်ခုကို နှိုင်းယှဉ်ရတော့မှာ ဖြစ်ပါတယ် ၊ အဲဒီအတွက် operator တွေ အသုံးပြုရတော့မှာပါ။ ကိုယ်လိုချင်သော column အားလုံးကိုနှိုင်းယှဉ်မှာ ဖြစ်တဲ့ အတွက် where clause ကို အသုံးပြုရပါမယ်။ book table မှာ နှိုင်းယှဉ်မှာ ဖြစ်ပါတယ်။

SELECT * **FROM** Book
WHERE Price>(SELECT **AVG**(Price)**FROM** Book)

b_no	Name	book	Price
2	warwar	love bird	1500

Count () Function

Count () Function သည် table ထဲမှာ သတ်မှတ်ထားသော Row အရေအတွက် ကို ဖော်ပြပေးပါတယ်။ ဒါပေမဲ့ Count () Function NULL တန်ဖိုးတွေကို ရေတွက်ပေးမှာ မဟုတ်ပါဘူး။ သတ်မှတ်သောပုံစံ

SELECT COUNT(column_name) FROM table_name

Table ထဲမှာ ရှိသော row တွေအားလုံးကို ရေတွက်ချင်ရင်တော့ (*) ကိုသုံးပြီးကြေညာရပါတယ်။

SELECT COUNT(*) FROM table_name

အောက်က ဥပမာသည် price column မှ 500 တန် စာအုပ်ဘယ်နှစ်အုပ် ပါသလဲဆိုတာကို ရေတွက်ပေးထားတာ ပုံစံဖြစ်ပါတယ်။

**SELECT COUNT(price) AS CustomerNilsen FROM book
WHERE Price='500'**

CustomerNilsen	
1	2

SQL COUNT(DISTINCT column_name) သည် column ထဲမှ မတူညီသော တန်ဖိုးတွေကို ရေတွက်ပေးတဲ့ အခါမှာ အသုံးပြုပါတယ်။ အောက်ဥပမာသည် Name column မှ မတူညီသော အမည်တွေ ၏ အရေတွက်ကိုဖော်ပြပေးထားသော ပုံစံဖြစ်ပါတယ်။

```
SELECT COUNT(DISTINCT Name) AS NumberofBookcustomer FROM book
```

	NumberofBookcustomer
1	5

FIRST () function သည် column ၏ ပထမဆုံးတန်ဖိုးကိုဖော်ပြပေးပါတယ်။
သတ်မှတ်သောပုံစံ ->

```
SELECT FIRST(column_name) FROM table_name
```

LAST () function သည် column ၏ နောက်ဆုံးတန်ဖိုးကိုဖော်ပြပေးပါတယ်။
သတ်မှတ်သောပုံစံ -->

```
SELECT LAST(column_name) FROM table_name
```

Max() function

Max() function သည် ရွေးချယ်ထားသော column ၏ အနည်းဆုံးတန်ဖိုးကို ဖော်ပြပေးပါတယ်။ char ဆိုရင်လဲ အက္ခရာအစဉ်လိုက် ရေတွက်ပေးပါတယ်။
အောက်က ဥပမာသည် Price column မှ အနည်းဆုံး တန်ဖိုးကို ရေတွက် ပြထားသောပုံစံဖြစ်ပါတယ်။

```
SELECT MAX(Price) AS LargestOrderPrice FROM book
```

	LargestOrderPrice
1	1500

MIN() function

MIN() function သည် ရွေးချယ်ထားသော column ၏ အများဆုံးတန်ဖိုးကို ဖော်ပြပေးပါတယ်။ char ဆိုရင်လဲ အက္ခရာအစဉ်လိုက် ရေတွက်ပေးပါတယ်။ အောက်က ဥပမာသည် Price column မှ အများဆုံး တန်ဖိုးကို ရေတွက် ပြထားသောပုံစံဖြစ်ပါတယ်။

```
SELECT MIN(Price) AS smallestOrderPrice FROM book
```

	smallestOrderPrice
1	500

SUM() Function

SUM() Function သည် ကိန်းဂဏန်းတွေ သတ်မှတ်ထား သော column ၏ စုစုပေါင်းကို ရေတွက်ပေးပါတယ်။

သတ်မှတ်သောပုံစံ □ □ `SELECT SUM(column_name) FROM table_name`
အောက်က ဥပမာသည် price column မှ စုစုပေါင်းကို ရေတွက်ပြထားသော ပုံစံဖြစ်ပါတယ်။

`SELECT SUM(Price) As Pricestotla FROM book`

	Pricestotla
1	3778

□

Scalar Functions

Scalar Functions သည် ထည့်သွင်းလိုက်သော data တွေပေါ်မှာ အခြေခံ၍ တန်ဖိုးတစ်ခုကိုပြန်လည်ဖော်ပြပေးပါတယ်။ Scalar Functions တစ်ချို့ကို အောက်က ခေါင်းစဉ်တွေမှာ တစ်ခုချင်းစီ ရေးပေးထားပါတယ်။

UCASE()Function

UCASE()Function သည် သတ်မှတ်ထားသော data တွေကို Uppercase သို့ပြောင်းပေးပါတယ်။ သတ်မှတ်သောပုံစံ □ SQL server တွင် UPPER() ပုံစံဖြင့် အသုံးပြုရပါတယ်။

`SELECT UPPER(column_name) FROM table_name`

အောက်မှာဖော်ပြထားသော table မှ Name column ကို စာလုံးမကြီးသို့ ပြောင်းမှာဖြစ်ပါတယ်။

SQL လေ့လာခြင်း နှင့် လက်တွေ့ အသုံးချ သင်ခန်းစာများ

www.akarphyoe.com

P_Id	Name	Address	City
5	Kyaw ko	ng min road	Taungdwingyi
7	warawr	ng min road	Taungdwingyi
9	su pyi moe	shweo road	Taungdwingyi

`SELECT UPPER(Name) as Name FROM student3`

	Name
1	KYAW KO
2	WARAWR
3	SU PYI MOE

LCASE()Function

LCASE()Function သည် သတ်မှတ်ထားသော data တွေကို lowercase သို့ပြောင်းပေးပါတယ်။ သတ်မှတ်သောပုံစံ ၊ SQL server တွင် LOWER() ပုံစံဖြင့် အသုံးပြုရပါတယ်။ UPPER နေရာမှာ Lower ပြောင်းလဲသတ်မှတ်ပေးလိုက်ပါ။

P_Id	Name	Address	City
5	Kyaw ko	ng min road	TAUNGWIN
7	warawr	ng min road	TAUNGWIN
9	su pyi moe	shweo road	TAUNGWIN


```
SELECT lower(City) as myNativetown FROM student3
```

	myNativetown
1	taungdwin
2	taungdwin
3	taungdwin

MID() Function

MID() Function သည် သတ်မှတ်ထားသော characters တွေကို ဆွဲယူ ဖော်ပြနိုင်ရန် အသုံးပြုပါတယ်။ မှတ်ချက် -- တစ်ချို့ server တွေမှာ MID() function မရပါ။

သတ်မှတ်သောပုံစံ ~

```
SELECT MID(column_name,start[,length]) FROM table_name
```

Column_name = ဆွဲထုတ်ပြီး ဖော်ပြချင်သော characters ၏ column ကို သတ်မှတ်ပေးရန်၊

Start = characters တွေဆွဲထုတ်မဲ့ နေရာကို သတ်မှတ်ပေးသည် ၊ (1)က စပါတယ်။

Length= ဆွဲထုတ်ခြင်သော characters တွေ၏ အရေတွက်ကို သတ်မှတ်ပေးရန်။

student3 table ၏ Name column မှ ပထမဆုံး characters လေးလုံးကို ဆွဲထုတ်ဖော်ပြမှာ ဖြစ်ပါတယ်။

SELECT MID(Name,1,4) **as** myfriendsname **FROM** student3

	Name
1	Kyaw
2	wara
3	su pyl

LEN() Fucntion

LEN() Fucntion သည် သတ်မှတ်ထားသော data တွေ၏ အရေတွက်ကို ဖော်ပြပေးပါတယ်။ သတ်မှတ်သောပုံစံ **SELECT LEN(column_name)**
FROM table_name

အောက်က ဥပမာသည် stdent3 table ၏ Address column မှ ပါဝင်သော စာလုံးအရေအတွက်ကို ဖော်ပြထားပါတယ်။

SELECT LEN(address) **FROM** student3

	charlength
1	11
2	11
3	10

ROUND()Function

ROUND()Function သည် ဒသမ ကိန်း တွေ့နဲ့ ဖော်ပြထားသော တန်ဖိုးတွေကို အနီးစပ်ဆုံးကိန်း ပုံစံတွေ အဖြစ် ပြောင်းလဲပေးပါတယ်။
သတ်မှတ်သောပုံစံ -- >

SELECT ROUND(column_name,decimals) FROM table_name

1. Decimals= ဒသမကိန်း ဖြတ်ရန် (ဒသမ နှစ်နေရာ)။

အောက်မှာဖော်ပြထား book table ၏ price column တွင် ပါသော တန်ဖိုးတွေ ကို ကိန်းပြည့်တွေဖြစ် အောင် ဖော်ပြပေးထားပါတယ်။

b_no	Name	book	Price
1	kyawko	homebook	500.36
2	warwar	love bird	1500.25
3	Khine	Love Story	528.0012
4	El Zar	loneliness	750.36
5	Su Pyi Moe	HTML Intro	500.02

`SELECT Price, ROUND(Price,3) as UnitPrice FROM book`

	Price	UnitPrice
1	500.36	500.36
2	1500.25	1500.25
3	528.0012	528.001
4	750.36	750.36
5	500.02	500.02

SQL Data type

Microsoft SQL server မှာ data type အမျိုးအစားတွေအများကြီးရှိပါတယ်။ database table ၏ column တစ်ခုစီတွင် data type တွေရှိကြပါတယ်။ အဲဒီ column တွေမှာ data type တွေကို သတ်မှတ်ပေး ခြင်းဖြင့် data တွေ မှားယွင်းစွာ ထည့်ခြင်းကို ကာကွယ်ပေးနိုင်ပါတယ်။ data type တွေကို သူနေရာအလိုက် သတ်မှတ်ခြင်းဖြင့် memory နေရာယူမှုကို လဲ အထောက်အကူပြုစေပါတယ်။ဥပမာ - int သတ်မှတ်ပေးထားသော column ကို char တွေ ထည့်လိုက်ရင် အဲဒီ column

က လက်ခံမှာ မဟုတ်ပါဘူး။ အောက်မှာ data type အမျိုးအစား တစ်ချို့ကို ဖော်ပြထားပါတယ်။ ကျွန်တော် အကျဉ်း ပဲရေးသားထားသောကြောင့်မပြည့်စုံပါဘူး၊ ပြီးတော့ data type တွေ၏ သဘောတရား ကို ယေဘုယ ခြုံရေးထားတာ ဖြစ်ပါတယ်။

Character data type

Character data type တွင် အမည်၊ လိပ်စာ၊ Zip code တွေနှင့် ဖုန်းနံပါတ် စသည် တို့ကို သိမ်းဆည်းနိုင်ပါတယ်။ character တွေ သတ်မှတ်တဲ့ အခါမှာ fixed-length character string နှင့် variable length character string ဆိုပြီး နှစ်မျိုးရှိပါတယ်။ fixed-length character string သည် user ကနေ character ဘယ်နှလုံး သတ်မှတ်ပေးမည်ကို ဆုံးဖြတ်ပေးရပါတယ်။ ဥပမာ - char(10) လုံး သတ်မှတ်ထားမယ်ဆိုရင် 10 လုံးပိုပြီးလက်ခံလို့ မရပါဘူး။ char(10) လုံးသတ်မှတ်ထားတာကို char ၅လုံး ပဲ သတ်မှတ်လိုက်မယ်ဆိုရင် ကျန်သော ၅လုံး နေရာကို ပါ memory နေရာယူမှာ ဖြစ်ပါတယ်။ variable length character string ကတော့ character ကြိုက်သလိုက်သတ်မှတ်လို့ရပါတယ်။ ဒါပေမဲ့ သူက သတ်မှတ်ထားသော char အရေတွက်ကိုပဲ memory နေရာယူပါတယ်။ အောက်က table မှာ character သတ်မှတ်နိုင်သော data type တွေကို ဖော်ပြထားပါတယ်။ fixed-length နှင့် variable length တို့ကို လဲ ခွဲပေးပြီးသား ဖြစ်သောကြောင့် ထပ်မရှင်းတော့ပါဘူး။

Character String

Data type	Description	Storage
char(n)	Fixed-length character string ဖြစ်ပါတယ်။ characters အများဆုံး 8,000 သတ်မှတ်နိုင်သည်	n
varchar(n)	Variable-length character string ဖြစ်ပါတယ်။ characters	

	အများဆုံး 8,000 သတ်မှတ်နိုင်သည်။	
varchar(max)	Variable-length character string ဖြစ်ပါတယ်. characters အများဆုံး 1,073,741,824 သတ်မှတ်နိုင်သည်။	
text	Variable-length character string ဖြစ်ပါတယ်. Maximum 2GB of text data	

Unicode String

Data type	Description	Storage
nchar(n)	Fixed-length Unicode data type ဖြစ်ပါတယ် ။ characters အများဆုံး 4,000 သတ်မှတ်နိုင်သည်	
nvarchar(n)	Variable-length Unicode data type ဖြစ်ပါတယ်. characters အများဆုံး 4,000 သတ်မှတ်နိုင်သည်	
nvarchar(max)	Variable-length Unicode data. characters အများဆုံး 536,870,912 သတ်မှတ်နိုင်သည်	
ntext	Variable-length Unicode data type ဖြစ်ပါတယ် . Maximum 2GB of text data	

Integer data type

Integer data type တွေသည် numeric data တွေကို သိမ်းဆည်းပါတယ်။ int data type တွေသည် ဂဏန်းတွေ တွက်ချက်ခြင်း မှာ အသုံးပြုပါတယ်။ int တစ်ချို့ကို အောက်မှာဖော်ပြထားပါတယ်။

Data type	Description	Storage
tinyint	Allows whole numbers from 0 to 255	1 byte
smallint	-32,768 and 32,767	2 byte
int	-2,147,483,648 and 2,147,483,647	4 byte
float	-1.79E + 308 to 1.79E + 308.	4 or 8 bytes

Decimal(p,s)	-10 ³⁸ +1 to 10 ³⁸ -1.	5-17 byte
--------------	--	-----------

Date type

Date type တွေသည် combination date တွေနှင့် time value တွေကို သိမ်းဆည်းပေးပါတယ်။ အောက်က table မှ မည့်သည့် ခုနစ် ရက် ကနေ မည်သည့် ခုနစ်ရက် ကြား က time တွေကို သတ်မှတ်နိုင်တယ် ဆိုတာကိုဖော်ပြထားပါတယ်။

Data type	Description	Storage
datetime	January 1, 1753 မှ December 31, 9999 ကြား ရှိ time ကို သတ်မှတ်ရန် ။ 3.33 milliseconds ထိ ဖော်ပြနိုင်ပါတယ်။	8 bytes
datetime2	January 1, 0001 မှ December 31, 9999 ကြား ရှိ time ကို သတ်မှတ်ရန် ။ nanoseconds ထိ ဖော်ပြနိုင်ပါတယ်။	6-8 bytes
smalldatetime	January 1, 1900 မှ June 6, 2079 ကြား ရှိ time ကို သတ်မှတ်ရန် ။ မီးနှစ် ထိသာ ဖော်ပြနိုင်ပါတယ်။	4 bytes
date	January 1, 0001 မှ December 31, 9999 ကြားထိသာ သတ်မှတ်နိုင်ပါတယ်။	3 bytes
time	နာရီကိုသာ သတ်မှတ်နိုင်ပါတယ်။ nanoseconds ထိ ဖော်ပြနိုင်ပါတယ်။	3-5 bytes
datetimeoffset	The same as datetime2 with the addition of a time zone offset	8-10 bytes
timestamp	timestamp သည် binary number တွေကိုသာ ဖော်ပြပေးပါတယ် ။ table တစ်ခုစီ မှာ timestamp variable တစ်ခုသာလျှင် ရှိနိုင်ပါတယ်။	

Binary data type

Binary data type သည် True/False, Yes/No, and On/Off values တို့ကိုသိမ်းဆည်းရန် အသုံးပြုပါတယ်။ Binary data type တွေသည် 0,1 and NULL တန်ဖိုးသုံးခုကိုသာ လက်ခံပါတယ်။

Data type	Description	Storage
bit	Allows 0, 1, or NULL	
binary(n)	Fixed-length binary data. Maximum 8,000 bytes	
varbinary(n)	Variable-length binary data. Maximum 8,000 bytes	
varbinary(max)	Variable-length binary data. Maximum 2GB	
image	Variable-length binary data. Maximum 2GB	

Group By statement

Group By statement သည် aggregate function တွေနဲ့ တွဲ၍ အသုံးပြုပါတယ်။ query လုပ်၍ ရလာသော result ကို အုပ်စုဖွဲ့၍ ပြန်လည်ဖော်ပြပေးပါတယ်။ ဥပမာ- ဆိုင်တစ်ဆိုင်မှာ တစ်နေ့တာ စာရင်းချုပ်တော့မယ်ဆိုရင် ဒီနေ့ လာဝယ်သောသူတွေ၏ အရေတွက် ဘယ်လောက်၊ ထပ်ခါတစ်လဲလဲ ဝယ်သူတွေ၏ စုစု ပေါင်းတန်းဖိုးတွေ က ဘယ်လောက် နှင့် ဝယ်ယူမှု အားလုံး၏ စုစုပေါင်းတွေက ဘယ်လောက်ဆိုတာ ကို ဖော်ပြ နိုင်ပါတယ်။ သတ်မှတ်သော ပုံစံ -- >

```
SELECT column_name, aggregate_function(column_name)
FROM table_name
WHERE column_name operator value
GROUP BY column_name
```

အောက်ကဥပမာ သည် customer တစ်ယောက်ဝယ်ခဲ့သော တန်ဖိုး စုစုပေါင်းကို group ဖြင့်ဖော်ပြထားပါတယ်။ sum() function ကို အသုံးပြုထားပါတယ်။

B_Id	Name	orderDate	BookPrice
1	kyawko	2012-01-20	1500
2	WarWar	2011-05-05	750
3	Su Pyi	2009-02-14	500
4	kyawko	2008-05-25	1000
5	Su Pyi	2012-01-25	1500

```
SELECT Name,SUM(BookPrice) FROM book1  
GROUP By Name
```

	Name	(No column name)
1	kyawko	2500
2	Su Pyi	2000
3	WarWar	750

Book1 table မှ column အားလုံးကို ပေါ်စေချင်တယ် ဆိုရင် အောက်ကပုံစံအတိုင်းကြေညာပါတယ်။

```
SELECT Name,orderDate,SUM(bookPrice) As price FROM book1  
Group By Name,orderDate
```

	Name	orderDate	price
1	kyawko	2008-05-25	1000
2	Su Pyi	2009-02-14	500
3	WarWar	2011-05-05	750
4	kyawko	2012-01-20	1500
5	Su Pyi	2012-01-25	1500

အောက်က ဥပမာသည် ဈေးနှုန်း 500 ထက်ကျော်သော name column ဘယ်နှစ်ခု ရှိလဲဆိုတာ ဖော်ပြထားပါတယ်။ Count() function နဲ့ အသုံးပြုပုံ ☐

SQL လေ့လာခြင်း နှင့် လက်တွေ့ အသုံးချ သင်ခန်းစာများ

www.akarphyoe.com

```
SELECT name, COUNT(*) as "Number of book"  
FROM book1  
WHERE BookPrice > 500  
GROUP BY Name
```

	name	Number of book
1	kyawko	2
2	Su Pyi	1
3	WarWar	1

