Basic GIS Training – Attribute Table Query

Query ကိုအသုံးပြုပြီး Attribute များကို Filter လုပ်ခြင်း

Query ဆိုတာက GIS ဒေတာထဲမှာ ပါရှိတဲ့ information တွေ အကုန်လုံးထဲက ကိုယ်လိုချင်တဲ့ အချက်လေးကိုပဲ (selective information) လှမ်းပြီးတော့ ထုတ်နုတ် အသုံးပြုတာကို ဆိုလိုပါတယ်။ ဘယ်လိုထုတ်သလဲဆိုတော့ database rule တစ်ခုနဲ့ သတ်မှတ်ပြီးတော့ ထုတ်ပါတယ်။ အခြေခံထားတဲ့နည်းကတော့ SQL database ကို SQL query command ခေါ် expression တွေနဲ့ အခြေခံ ထားပါတယ်။

SQL command တွေ အကုန်လုံးကို အခြေခံကနေ အသေးစိတ် စတင် လေ့လာချင်သူ အတွက် online resource ဖြစ်သော http://www.w3schools.com/sql/ တွင် ဆက်လက် လေ့လာ နိုင်ပါသည်။

GIS Database ထဲတွင် ဒေတာပေါင်း မြောက်များစွာ တည်ရှိနေနိုင်ပါသည်။ ဥပမာ- မြန်မာနိုင်ငံ၏ ကျေးရွာများကို ထည့်သွင်းထားသော Village GIS database ကဲ့သို့ဖြစ်ပါမည်။ ၎င်း database ထဲက (database ကို မပြောင်းလဲ၊ မထိခိုက်စေပဲ) ကိုယ်အသုံးပြုမည့် မြို့နယ်တစ်ခုထဲတွင် ရှိနေသော ကျေးရွာများကိုသာ Retrieve လုပ်ကာ ခွဲထုတ်ပြီး၊ Analysis ပြုလုပ်ခြင်း၊ မြေပုံထုတ်လုပ်ခြင်း စသည်တို့ ပြုလုပ်နိုင်ပါသည်။ ၎င်းကဲ့သို့ ပြုလုပ်နိုင်ရန် အတွက် SQL command တွေထဲက SELECT ဆိုတဲ့ command တစ်ခုကို မသိမဖြစ် သိထားရပါမည်။ ဘာလို့လဲဆိုတော့ ကျွန်တော်တို့က GIS Attribute ဖိုင်ထဲကနေ ကိုယ်လိုချင်တဲ့ အပိုင်းလေးကိုသာ SELECT လုပ်ပြီးတော့ သုံးရမှာ ဖြစ်ပါတယ်။ SELECT ဆိုတဲ့ command အတွက် WHERE ဆိုတဲ့ condition နဲ့ တွဲပြီး သုံးရပါတယ်။ ဥပမာ- ပင်းတယ မြို့နယ် အတွက်

```
SELECT * FROM VillageFile     WHERE Township = 'Pindaya'
```

GIS software မှာက Shapefile အမျိုးအစားကို သုံးထားရင် Layer ဖိုင်အမည်ပေါ်မှာ Select လုပ်ထားပြီး ၎င်းရဲ့ Attribute table ကို Query လုပ်ရမှာ ဖြစ်ပါတယ်။ Database ဖိုင်အမျိုးအစား ဆိုရင်တော့ import လုပ်ကတည်းက တစ်ခါတည်း Select ပြီး Query ပြုလုပ်ပေးနိုင်ပါတယ်။ Select ဆိုတာက လွယ်ပါတယ်။ Layer ပေါ်ကို mouse တစ်ချက် Click ပေးရုံပါပဲ။ ဒီမှာ where ဆိုတဲ့ အခြေအနေကို မှန်မှန်ကန်ကန် ဖြစ်အောင် ထည့်ပေးတတ်ဖို့ အရေးကြီးတယ်။ GIS ဖိုင်များ၏ Attribute table ကို Query လုပ်ဖို့ရန်အတွက် SELECT နဲ့ WHERE command အသုံးပြုပုံကို အောက်ပါ နမူနာမှာ လေ့လာနိုင်ပါတယ်။

```
SELECT column_name
FROM Attribute_table_name
WHERE column_name operator value;

SELECT column_name(s)
FROM Attribute_table_name
WHERE column_name LIKE pattern;
```

Query Builder မှာ ဘာရေးရမှာလဲ

တကယ်တမ်းတော့ QGIS ရဲ့ Query Builder ထဲမှာ SELECT တစ်ခုလုံးအတွက် အပြည့်အစုံ ရေးရတာ မဟုတ်ဘဲ WHERE ဆိုတဲ့ clause အပိုင်းလေး အတွက်ကိုပဲ Expression ထည့်ရေးပေးရတာပါ။

WHERE clause မှာ အသုံးပြုနိုင်တဲ့ Operator တွေကတော့

Operator	Description
=	Equal
<>	Not equal. Note: In some versions of SQL this operator may be written as !=
>	Greater than
<	Less than
>=	Greater than or equal

<=	Less than or equal
BETWEEN	Between an inclusive range
LIKE	Search for a pattern
IN	To specify multiple possible values for a column

= Operator ကို အသုံးပြုပုံမှာ

SELECT * FROM Customers

WHERE ***Country='Germany'***

= က ကွက်တိ တူတဲ့ အရာကို ရှာဖို့ ဖြစ်ပါတယ်။ ကွက်တိ တူချင်တဲ့အခါမှာ သုံးပါတယ်။ Number ရော Text ပါ အသုံးပြုလို့ရပါတယ်။

!= သို့မဟုတ် **<>** Operator ကို အသုံးပြုပုံမှာ

SELECT * FROM Customers

WHERE ***Country='Germany'***

!= , <> က ကွက်တိ မတူတဲ့ အရာကို ရှာဖို့ ဖြစ်ပါတယ်။ ၎င်းမှလွဲ၍ အခြားသော အရာကို ရှာချင်တဲ့အခါမှာ သုံးပါတယ်။ Number ရော Text ပါ အသုံးပြုလို့ရပါတယ်။

AND Operator ကို အသုံးပြုပုံမှာ

SELECT * FROM Customers

WHERE ***"SOV0NAME" = 'Germany' AND "NAME" = 'Berlin'***

AND က ၂ ခုစလုံးနဲ့ ကိုက်ညီတဲ့ အရာကို ရှာဖို့ ဖြစ်ပါတယ်။

OR Operator ကို အသုံးပြုပုံမှာ

SELECT * FROM Customers

WHERE ***"NAME" = 'Berlin' OR "NAME" = 'Munchon'***

OR က ၂ ခုထဲက တခုခုနဲ့တူတဲ့၊ ကိုက်ညီတဲ့ အရာကို ရှာဖို့ ဖြစ်ပါတယ်။ ၂ ခုစလုံးနဲ့လည်း တူနိုင်ပါတယ်။

AND နှင့် OR ကို တွဲပြီး အသုံးပြုပုံမှာ

SELECT * FROM Customers

WHERE ***"SOV0NAME" = 'Germany' AND "NAME" = 'Berlin' OR "NAME" = 'Munchon'***

Number နှင့် Text

ဒီမှာ တစ်ခု သိထားရမှာက Query လုပ်မယ့် Attribute Column သည် အက္ခရာစာလုံး အမျိုးအစား ဖြစ်လျှင် single quote ‘ ’ ထဲတွင် ထည့်ပေးရပြီး၊ ဂဏန်း အမျိုးအစား ဖြစ်လျှင် SQL expression ၌ တိုက်ရိုက် ထည့်ရေးလို့ ရသည်။

LIKE ကို အသုံးပြုပုံ မှာ

SELECT column_name(s)

FROM table_name

WHERE ***column_name LIKE pattern***

LIKE က ဆင်တူတဲ့ အရာကို ရှာဖို့ဖြစ်ပါတယ်။ အတိအကျ မသိတဲ့အခါမှာ သုံးပါတယ်။ LIKE အတွက် % နဲ့ တွဲသုံးရပါတယ်။ ဒီနေရာမှာ % က ဘာမဆို ဖြစ်နိုင်တဲ့ Text အတွက် variable အဖြစ် အသုံးပြုထားပါတယ်။

LIKE နှင့်% Wildcard ကို Text field တွင် အသုံးပြုပုံမှာ

SELECT * FROM Customers

WHERE ***"NAME" LIKE 'ber%'***

ber နှင့် စထားသော စကားလုံး၊ အမည်များ အားလုံးကို စစ်ထုတ်ဖို့အတွက် ဖြစ်ပါတယ်။ ဆင်တူတွေကို ရှာချင်တဲ့အခါမှာ သုံးပါတယ်။

SELECT * FROM Customers

WHERE ***"NAME" LIKE '%es%'***

စကားလုံးများ၏ ကြားတွင် es ပါသော အမည်များ အားလုံးကို ရှာဖို့ဖြစ်ပါတယ်။ ဆင်တူတွေကို ရှာချင်တဲ့အခါမှာ သုံးပါတယ်။

SELECT * FROM Customers

WHERE ***"NAME" LIKE '_erlin'***

ရှေ့စာလုံး ၁ လုံး ပါရှိပြီး (၁ လုံးမသိ) _erlin နှင့် အဆုံးသတ်ထားသော စကားလုံး၊ အမည်များ အားလုံးကို ရှာဖို့ဖြစ်ပါတယ်။ စကားလုံး အရေအတွက်တူပြီး ဆင်တူတဲ့ အမည်တွေကို ရှာချင်တဲ့အခါမှာ သုံးပါတယ်။

SELECT * FROM Customers

WHERE ***NAME" LIKE 'L_n_on'***

L_n_on ပါရှိပြီး၊ ဒုတိယစာလုံးမသိ (၁ လုံးမသိ) ၊ စတုတ္ထ စာလုံးမသိ သော စကားလုံး၊ အမည်များ အားလုံးကို ရှာဖို့ဖြစ်ပါတယ်။ စကားလုံး အရေအတွက်တူပြီး ဆင်တူတဲ့ အမည်တွေကို ရှာချင်တဲ့အခါမှာ သုံးပါတယ်။

' % ' ခေါ် Character Wildcard

Wildcard ဆိုတာက မည်သည့် Character စာလုံးမဆို၊ စာလုံးအရေအတွက် ဘယ်နှစ်လုံးပဲရှိရှိ ကိုယ်စားပြုဖို့ရန် အတွက် Expression ထဲမှာ သင်္ကေတ အနေနဲ့ သုံးပြီး ' % ' ဆိုပြီး ထည့်ရေးပေး ရတယ်။

IS နှင့်တွဲပြီး NULL တန်ဖိုးကို အသုံးပြုပုံ မှာ

SELECT LastName,FirstName,Address FROM Persons

WHERE Address IS NULL

IS NULL က Blank ဖြစ်နေတဲ့ တန်ဖိုး တွေကို ရှာဖို့ ဖြစ်ပါတယ်။

IS NOT နှင့်တွဲပြီး NULL ကို အသုံးပြုပုံမှာ

SELECT LastName,FirstName,Address FROM Persons

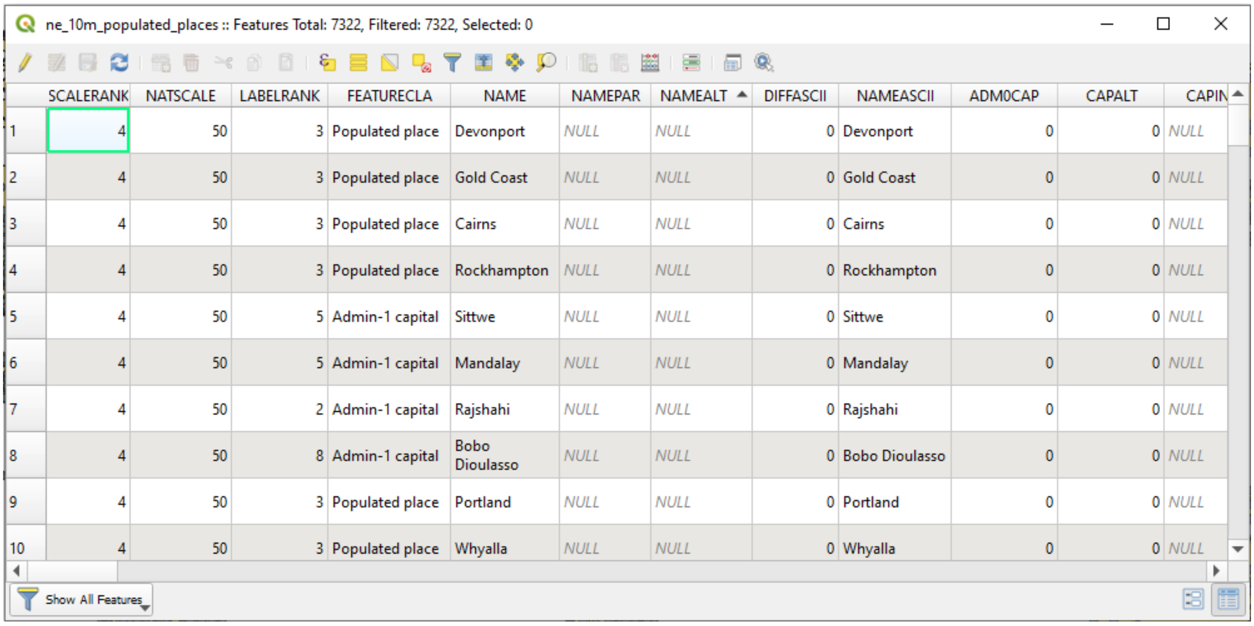
WHERE Address IS NOT NULL

IS NOT NULL က Blank မဟုတ်တဲ့ တန်ဖိုး တွေကို ရှာဖို့ ဖြစ်ပါတယ်။

ဒီလိုနည်းနဲ့ Filter လုပ်ပြီး ရလာတဲ့ ဒေတာကို မြေပုံပေါ်မှာ ဖော်ပြတာပဲဖြစ်ဖြစ်၊ analysis ဆက်လုပ် တာပဲဖြစ်ဖြစ် အသုံးပြုလို့ ရပါတယ်။ အခု Visualization အခန်းမှာတော့ မြေပုံပေါ်မှာ ပြဖို့ သက်သက်ပဲ နမူနာ လုပ်မယ်။ GIS ဖိုင်ရဲ့ Attribute Database ထဲကနေ ကိုယ်လိုချင်တဲ့ အချက်အလက်ကို ကျွမ်းကျင်စွာနဲ့ Query ဆွဲတတ်အောင် ဆက်လေ့ကျင့်သွားပါ။

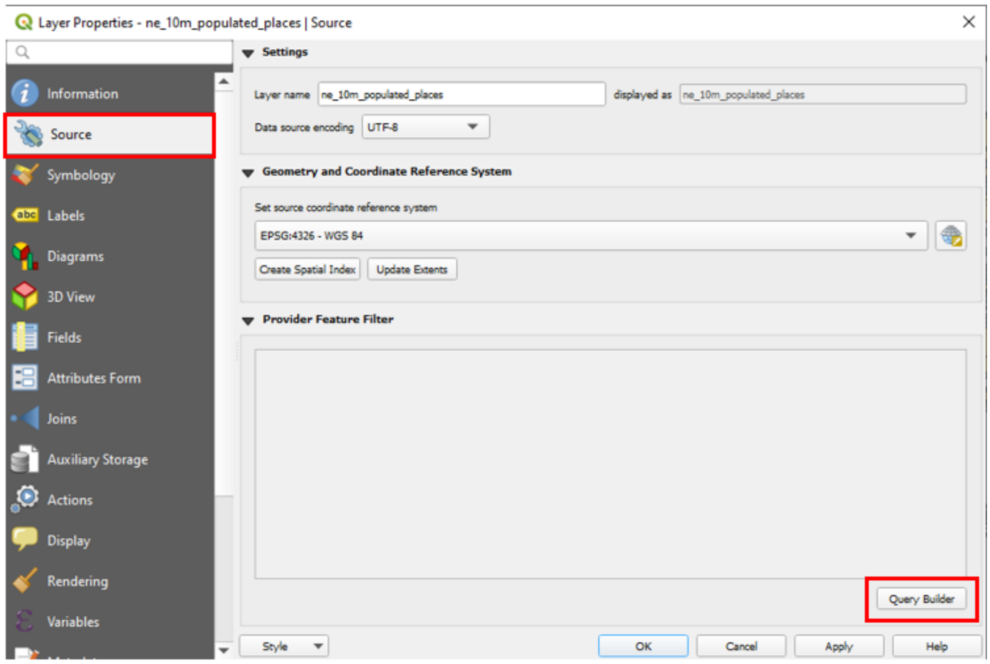
လက်တွေ့လေ့ကျင့်ရန်အတွက် Training data/ World folder ထဲမှ ne_10m_populated_places.shp ဟူသော ကမ္ဘာ့နိုင်ငံများ၏ မြို့များ ပါရှိသော ဖိုင်ကို ဖွင့်လိုက်ပါ။

ရှေ့ဆက် မသွားခင်လေးမှာ ဒီ Layer ရဲ့ Attribute table ကို တစ်ချက် ဖွင့်ပြီး၊ ဘာ Field / Column တွေပါ သလဲဟု စူးစမ်း လေ့လာကြည့်လိုက်ပါ။ အဲဒီ table က ကျွန်တော်တို့ Query လုပ်မည့် database ဖိုင်ပါပဲ။ ADM0CAP ဆိုတဲ့ field ပါသလား။ ကြည့်လိုက်ပါ။

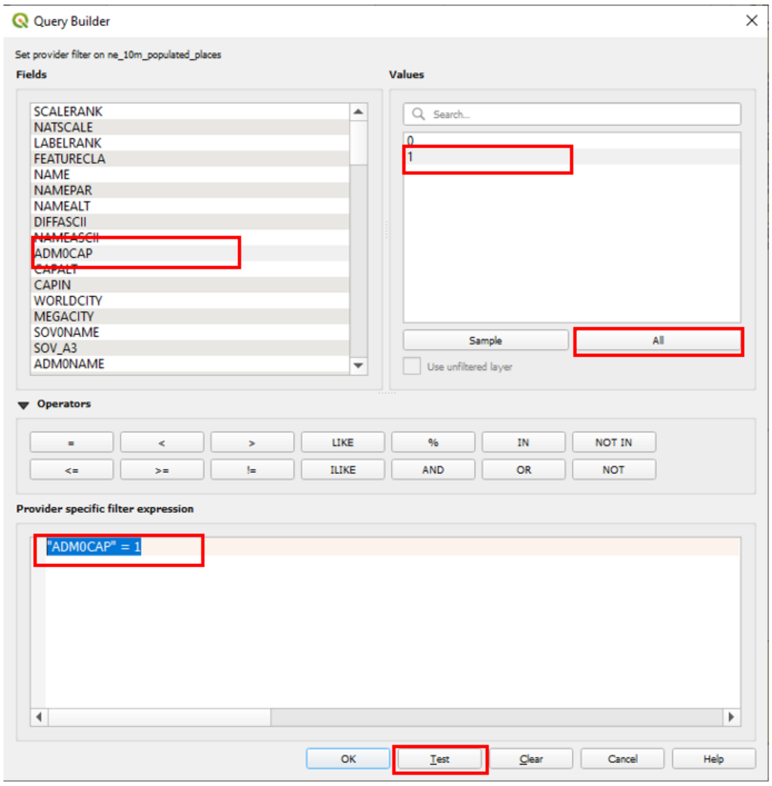


QGIS ရဲ့ Layers pane ထဲ၌ne_10m_populated_places နာမည်ပေါ်သို့ Right-click တင်ပြီး ပေါ်လာသော Command များထဲမှ Properties ဟူသော စာသားကို နှိပ်ပါ။ ထို Layer အတွက် properties window တစ်ခု ပေါ်လာပါမည်။

properties window ရဲ့ ဘယ်ဘက် pane ထဲမှာ၊ အပေါ်နားက Source ဟူသော tab ကို နှိပ်လိုက်ပါ။ သူနဲ့ သက်ဆိုင်တဲ့ page ပေါ်လာမည်။



ပေါ်လာတဲ့ page ရဲ့ ညာဘက် အောက်နားမှာ Query Builder ဆိုတဲ့ button လေးကို ဆက်နှိပ်လိုက်ပါ။ Query Builder window တစ်ခု ပေါ်လာ လိမ့်မည်။

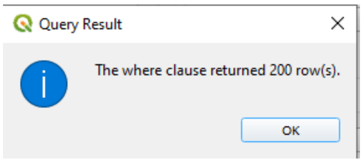


Fields ဆိုတဲ့ List ထဲက **ADM0CAP** ကို Double-click နှိပ်ပြီး expression သို့မဟုတ် formula အကွက် ထဲမှာ **"ADM0CAP"** ပေါ်လာ လိမ့်မည်။

ထို့နောက် Operators ဆိုတဲ့ စာရင်းထဲက ညီမျှခြင်း ဖြစ်တဲ့ = ကို တစ်ချက်နှိပ်လိုက်ပါ။ အခုလောက်ဆို Expression အကွက်ထဲမှာ **"ADM0CAP" =** ဆိုပြီး ပေါ်နေသင့်ပါသည်။

ဒီ Field ထဲက Value တွေ ကို အကုန် (အကုန်လုံး) populate လုပ်ဖို့ရန် အတွက် ညာဘက် အပေါ်နားက All ဆိုတဲ့ button ကို နှိပ်လိုက်ပါ။ ဒီ Field ထဲမှာ ပါဝင်တဲ့ **0** နဲ့ **1** တွေ ပေါ်လာလိမ့်မယ်။

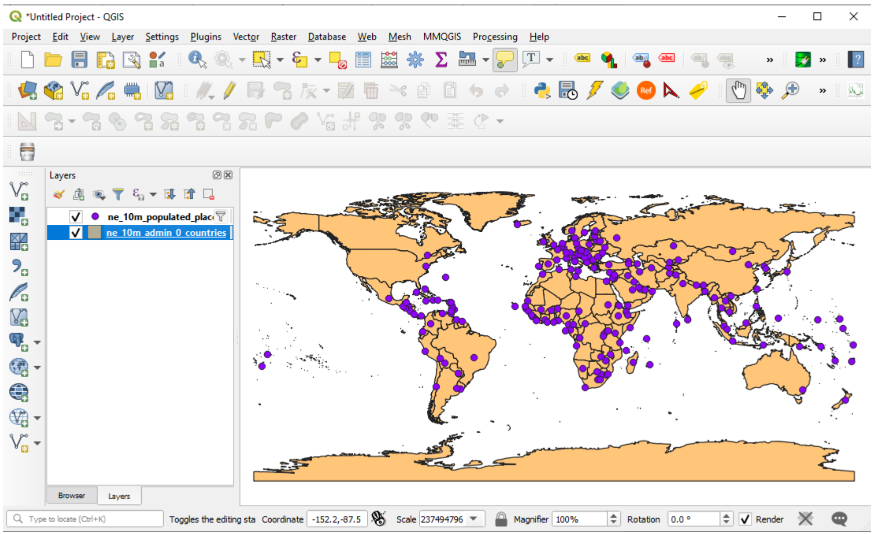
အဲဒီ အထဲက **1** ဆိုတာကို Double-click နှိပ်ပြီး ရွေးပေးလိုက်ပါ။ အခုလောက်ဆို Expression အကွက်ထဲမှာ **"ADM0CAP" = 1** ဆိုပြီး ပေါ်နေသင့်ပါသည်။



အဲဒီနောက် အောက်နားက Test ဆိုတဲ့ button ကို နှိပ်ပြီး မိမိ၏ Expression မှန်၊ မမှန် စမ်းကြည့် နိုင်ပါသည်။ မှန်ပါက အောက်ပါ ကဲ့သို့ Query Resultmessage မျိုး ပြနေပါလိမ့်မည်။

OK များ ဆက်နှိပ်ပြီး မြေပုံကို ပြန်ကြည့်လိုက်ပါ။ နိုင်ငံများ အသီးသီး၏ မြို့တော်များကိုသာ ဖော်ပြနေပါလိမ့်မည်။

Layer pane ထဲက ဖိုင်နာမည်ပေါ် Right-click > Filter နှိပ်သော် ဘာပေါ်လာသနည်း။



အထက်ပါ ကျွန်တော်တို့ လုပ်ခဲ့သော Query သည် populate places GIS ဖိုင်ထဲရှိ မြို့တော်များထဲမှ နိုင်ငံ၏ မြို့တော်ကိုသာ စစ်ထုတ်ယူပြီး မြေပုံပေါ်တွင် လာရောက် ပြသခြင်း ဖြစ်ပါသည်။ Query ပြုလုပ်ခြင်း သည် original database ကို ပြုပြင်ပြောင်းလဲခြင်း မဟုတ်ပေ၊ မထိခိုက်ပေ။

ဤကဲ့သို့ Query ပြုလုပ်နိုင်ရန် အတွက် Attribute database ထဲတွင် မြို့တော် ဟုတ်သည် မဟုတ်သည် စသည့် အချက်အလက် ကို ADM0CAP ထဲ၌ 1 နှင့် 0 ဟု code များဖြင့် စနစ်တကျ ထည့်သွင်းထားပါသည်။

Copyright © MIMU 2020