# Хязгаарлалт болон Триггерийн

# хэрэглээ

# (Лаборатори №2)

У. Төрболд

ХШУИС, Мэдээллийн технологийн IV-р түвшний оюутан, turbold1125@gmail.com

# ОРШИЛ

Энэхүү лабораторын ажил нь SQL хэлийн хоёр чухал шинж чанар болох хязгаарлалт болон триггерийг судлах болно. Эдгээр функцууд нь өгөгдлийн бүрэн бүтэн байдлын дүрмийг дагаж мөрдөх, мэдээллийн сан дахь тодорхой үйл явдлууд дээр үндэслэн үйлдлүүдийг автоматжуулах боломжийг олгодог ба эдгээрийг ашиглан практик дээр туршина.

# 2. ЗОРИЛГО

Энэхүү лабораторын зорилго нь өгөгдлийн сан дээрх хязгаарлалт буюу constraint, триггер буюу автоматжуулалтыг хэрэгжүүлэх юм.

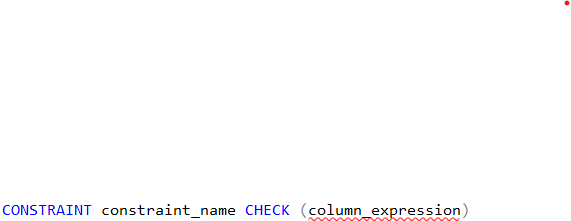
# 3. ОНОЛЫН СУДАЛГАА

## 3.1 Constraint

Тодорхойлолт:

Хүснэгтэнд орж болох өгөгдлийн төрлийг хязгаарлахад Constraint – ыг ашигладаг. Энэ нь хүснэгтэнд байгаа өгөгдлийн тодоорхой хязгаарлалт болон үнэн зөв, найдвартай байдлыг баталгаажуулдаг. Constraint болон өгөгдлийн үйлдлийн хооронд ямар нэгэн зөрчил байвал үйлдлийг зогсооно. Constraint нь баганын түвшин эсвэл хүснэгтийн түвшин байж болно. Баганын түвшний хязгаарлалт нь баганад, хүснэгтийн түвшний хязгаарлалт нь бүх хүснэгтэд хамаардаг.

Syntax:



- constraint\_name: Шалгах хязгарлалтын хэрэглэгчийн тодорхойлсон нэр

- column\_expression: Хүчинтэй утгыг Үнэн, буруу утгыг Худал гэж үнэлдэг логик илэрхийлэл

Давуу тал:

Өгөгдлийн баталгаажуулалтын үндсэн дүрмийг хэрэгжүүлэх энгийн бөгөөд үр дүнтэй арга.

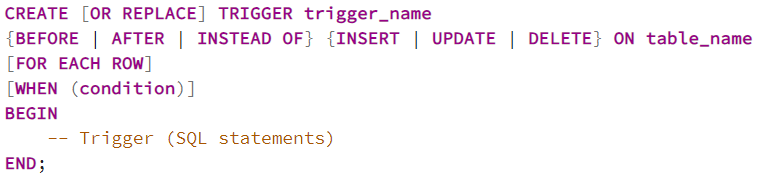
Буруу оруулах, шинэчлэхээс сэргийлж мэдээллийн чанарыг сайжуулна.

## 3.2 Trigger

Тодорхойлолт:

SQL – д триггер нь датабааз дахь тодорхой хүснхэгт дээрх үйл явдлыг автоматаар ажилладаг хадгалагдсан процедур юм.

Syntax:



CREATE TRIGGER нь шинэ триггер үүсгэх эсвэл байгаа нэгийг солиход ашиглана.

BEFORE, AFTER нь хэзээ асах ёстойх заана.

INSERT, UPDATE, DELETE нь триггерийн үйлдлийн төрлийг заана.

WHEN нь нөхцөл

# 4. ХЭРЭГЖҮҮЛЭЛТ

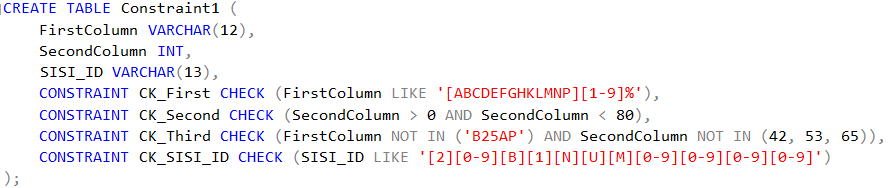
1. Хүснэгт үүсгээд талбарт тус бүрт дараах хязгаарлалтыг хийж өг

1.1 Эхний орон ABCDEFGHKLMNP үсгүүдийн аль нэг нь байх, 2 дахь орон нь 1-9 хооронд тоо байх, сүүлийн орон нь ZYXLMNO үсгүүдийг агуулаагүй байна. 3 дахь оронгоос эхлээд сүүлийн оронгийн хооронд дурын урттай тоо, үсэг байж болох баганад зориулж CK бич

1.2. баганын утга нь 0 ээс их 80 аас бага байх шаардлагатай CK бич

1.3. эхний багана B25AP -аас өөр утгатай байх бөгөөд 2 дахь багана нь 42,53,65 утгыг агуулаагүй байх хүснэгтийн хувьд CK бич

1.4. SISI id -г зөв оруулах CK бич



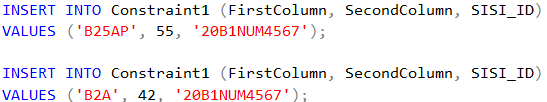
FirstColumn багана нь A-P хүртэлх утгыг аваад, 2 дахь орон нь 1-9 хүртэлх тоо байх ба энэхүү функцийг LIKE функцээр дамжуулан хэрэгжүүлсэн.

SecondColumn багана нь 0 – 80 хүртэл тоог жиших үйлдлээр хэрэгжүүлсэн.

CK\_Third constraint нь FirstColumn B25AP утгыг агуулаагүй байх ба SecondColumn нь 42,53,65 гэх утгыг NOT IN функцийн тусламжтайгаар хэрэгжүүлсэн.

SISI\_ID шалгахдаа эхний орон 2 – оор эхлэх ба дараагийн орон нь 0 – 9 хүртэлх цифр авах ба B гэсэн үсэг авч араас нь 1 гэсэн утгыг авна. 20B1NUM гэдэгтэй адилаар NUM гэх утгыг авч сүүлийн 4 орон нь цифр байх боломжтой.

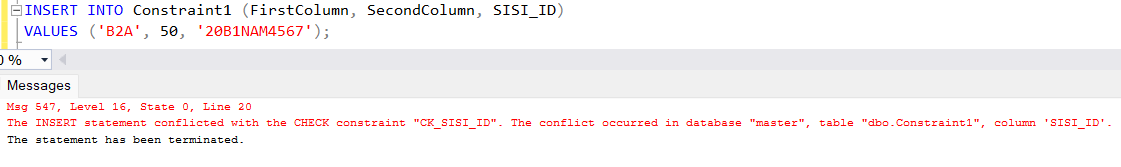
Үр дүнг шалгах:



Дээрх функцийг хэрэгжүүлэхэд алдаа гарах ба 42, B25AP гэсэн утгуудыг оруулж өгсөн нь CK\_Third constraint – ийг зөрчиж байгаа тул доорх алдаа гарна.

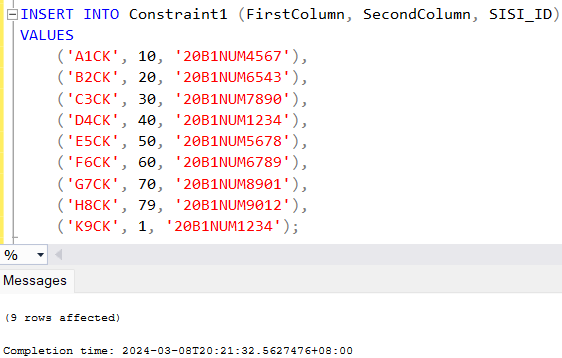
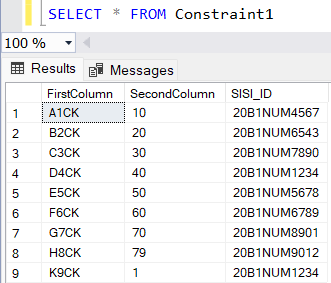


Доорх функцийг хэрэгжүүлэхэд SISI\_ID утга оруулахад NAM хэмээн оруулсан тул CK\_SISI\_ID constraint – ийг зөрчиж байгаа тул доорх алдаа гарна.



Үр дүн:

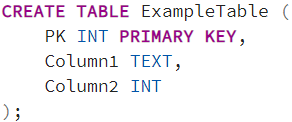
Үүний дараа шалгаж дуусаад зөв утгуудыг оруулахад амжилттай болсон гэх мэдэгдлийг өгнө.

1. Өгөгдлийн санд өгөгдлийг програмын аргаар оруулах

2.1 Янз бүрийн өгөгдлийн төрөлтэй багануудтай хүснэгт үүсгэ. Int төрөлтэй 1 баганыг

PK болгоно



Энэ нь Text, Int төрөлтэй 3 багана байх ба PK баганыг Primary Key болгосон.

2.2 Хүснэгтэд мөр бичлэгийг нэмэхдээ PK баганын утгыг нэгээр нэмэгдүүлж бусад

баганын мэдээллийг 10, 20, 30 яваад өөрчилөгдөхөөр програмчлана.

Жишээ нь:

             1 texta 20

             2 texta 20

               ...

            10 textb 20

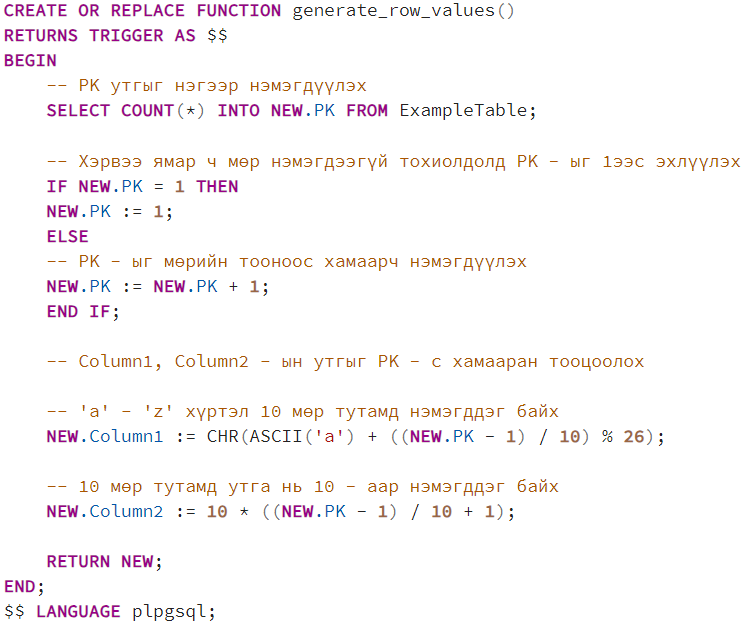
            11 textb 20

               ...

            20 textb 30

            21 textb 30

Үүнийг програмчлахын тулд бидэнд нэг функц болон триггер ашиглан хийх боломжтой.



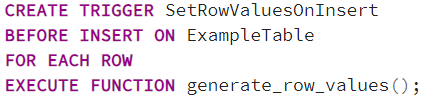
Энэхүү функц нь ExampleTable хүснэгтэд шинэ мөр оруулахаас өмнө гүйцэтгэх зориулалттай.

COUNT(\*) INTO NEW.PK нь тухайн хүснэгтийн мөрийг тоолж шинээр оруулсан мөрийн PK – д тоолох утгыг онооно.

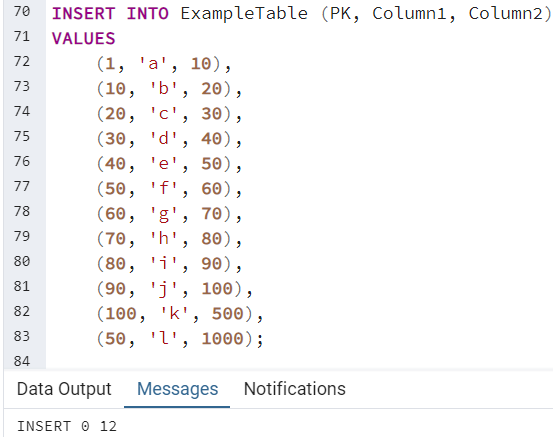
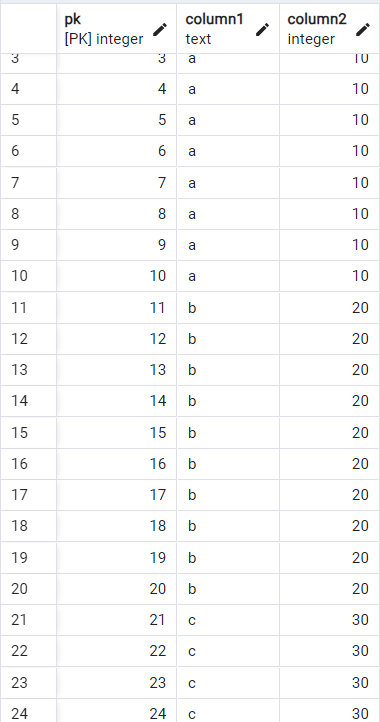
IF NEW.PK = 1 THEN … END IF нь NEW.PK утгыг 1 – тэй тэнцүү эсэхийг шалгадаг. Хэрэв байгаа бол хүснэгтэд одоо мөр байхгүй гэсэн үг. NEW.PK утгыг 1 гэж оноосон ба мөр байгаа тохиолдолд NEW.PK утгыг 1 ээр нэмэгдүүлнэ.

1, 2 баганын утгыг тооцоолохдоо % ашиглан 10 мөр тутамд a – z, 10 – аар нэмэгддэг.

Return NEW нь өөрчилсөн шинэ мөрийн утгыг буцаана. Триггер функцийг ажиллуулсны дараа шинэчлэгдсэн утгууд бүхий мөрийн ExampleTable хүснэгтэнд оруулах болно.



Энэхүү триггер нь ExampleTable – д утга оруулахын өмнө ажиллах ёстой ба хүснэгтэд оруулсан мөр бүрт триггер нэг удаа ажиллахыг тодорхойлсон ба дээрх функцийг гүйцэтгэнэ.

 Үр дүн:

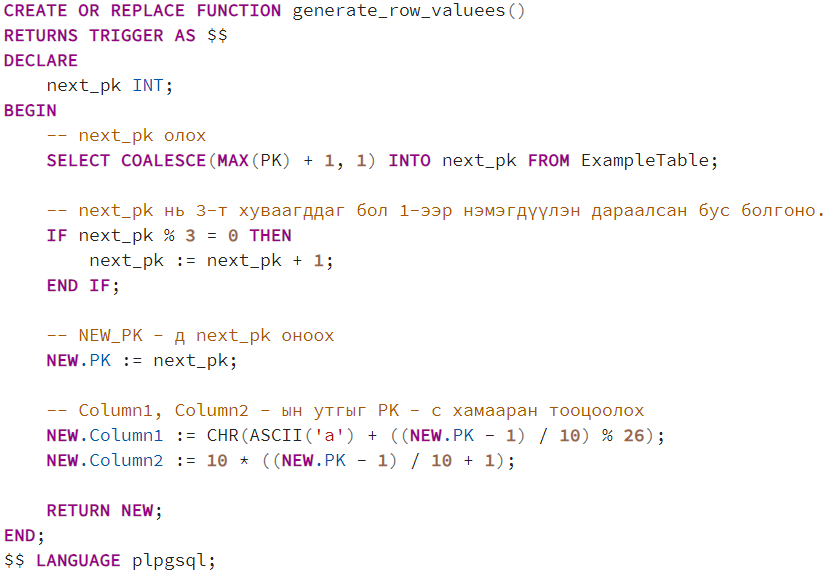
1.3 PK болсон багана дахь кодыг дараалсан биш байдлаар үүсгэдэг болгох

          Жишээ нь

      1

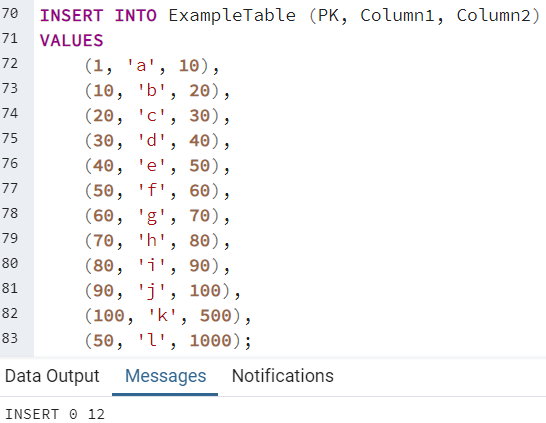
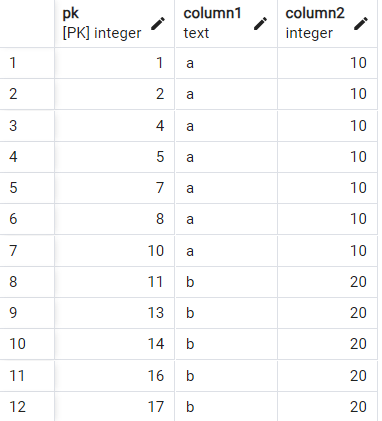
      2

      4



Pk баганаас хамгийн их утгыг сонгоод 1 ээр нэмэгдүүлэх замаар дараагийн боломжтой үндсэн түлхүүрийн утгыг олно. Хэрэв мөр байхгүй тохиолдолд 1ээс эхэлнэ. Next\_pk нь 3-т хуваагддаг бол үүнийг дараалалгүй болгохын тулд 1 ээр нэмэгдүүлнэ. New.pk – ийн утгыг next\_pk болгож өөрчилнө. Үүний дараа Trigger ийг Before Insert төлөвт ажиллуулдаг болгосноор доорх үр дүн гарна.

Үр дүн:

# 5. ДҮГНЭЛТ

Энэхүү лабораторын ажлаар SQL өгөгдлийн сан дахь хязгаарлалт, триггерүүдийн практик хэрэгжилтийг судалсан болно. Өгөгдөл оруулах дүрмийг тодорхойлох замаар өгөгдлийн бүрэн бүтэн байдал, үнэн зөв байдлыг хангахын тулд хязгаарлалтуудыг ашигласан бол өгөгдлийн сан дахь тодорхой үйл явдлуудад тулгуурлан үйлдлүүдийг автоматжуулахын тулд триггерийг ашигласан.