

## Лабораторийн ажил 3. Хүснэгт засварлах, хязгаарлал тодорхойлох, бичлэгтэй ажиллах

### Даалгавар-1

Оюутан бүр Жишээ болгон сонгож

- Лаборатори №2 дээр үүсгэсэн скрипт файлыг хуулж Лаборатори №3 файл үүсгэ
- Шинэ файлын Зарим командыг засаж/хасаж Хүснэгт засварлах үйлдэл ашиглан буцааж нэмэхэд бэлдэнэ.

### Даалгавар-2

Хүснэгт засварлах үйлдэл ашиглан дараах үйлдлүүдийн гүйцэтгэж скрипт файлыг тайлбарлах

- Багана нэмэх, засах, устгах /Багадаа 3ш багана дээр туршилт хийнэ/
  - NULL/NOT NULL – Хоосон байж болох эсэхийг тодорхойлно
  - DEFAULT – Өгөгдмөл утгыг тодорхойлно
- Хязгаарлалт, нэмэх, устгах
  - PRIMARY KEY - Анхдагч түлхүүр тодорхойлно
  - UNIQUE –Давтагдашгүй утга авахыг тодорхойлно
  - /Багадаа 2ш багана дээр туршилт хийнэ/
  - CHECK – Логик илэрхийллийн тусламжтайгаар баганы утгыг шалгах
  - /Багадаа 2ш багана дээр туршилт хийнэ/
  - FOREIGN KEY – Гадаад түлхүүр тодорхойлно
  - /Багадаа 3ш багана дээр туршилт хийнэ/
- Мөр бичлэг нэмэх  
Хүснэгт бүд 10ш мөр жинхэнэ өгөгдөл нэмнэ

### Даалгавар-3

- Хүснэгтийн бүтцэд өөрчлөлт орсон байдлыг **DESC tablename** команд ашиглаж харуулна
- Хязгаарлалтанд өөрчлөлт орсон байдлыг  
**select \* from information\_schema.table\_constraints where Table\_name = 'хүснэгт нэр'**  
ашиглаж харуулна

### Хугацаа

Лабораторийн цаг дээр үзүүлнэ.

### Гүйцэтгэж хамгаалах шаардлага

- Лабораторийн ажил №2 засаж үүсгэсэн скрипт файлаас ӨС-ийн үүсгэх хэсгийг ажиллуулна
- Хүснэгтэд багана нэмэх, засах, устгах хэсгүүдийг ажиллуулж турших (алхам бүрд өөрчлөлтийг харуулна)
- Хүснэгтэд хязгаарлалт нэмэх, устгах хэсгүүдийг ажиллуулж турших (алхам бүрд өөрчлөлтийг харуулна)
- Хүснэгт бүрд өгөгдөл нэмэх (insert команд ашиглаж хүснэгт бүрд бичлэг багадаа 10ш жинхэнэ өгөгдөл нэмнэ. Хэрэв хүйс гм хүснэгт байвал эр/эм гэж 2ш бичлэг команд байна)

## Нэмэлт материал

**Жишээнүүдэд дараах Student хүснэгтийг үүсгэж ашиглав.**

```
CREATE TABLE Student
(
  //B170920001 бүтэцтэй утга авах тул
  studCode varchar(10) NOT NULL PRIMARY KEY,
  //оюутны нэр нь кирилл үсгээр бичигдэх тул өгөгдлийн төрөл нь
  nvarchar(урт) байна
  firstName varchar(50) NOT NULL,
  //оюутны овог нь кирилл үсгээр бичигдэх тул өгөгдлийн төрөл нь
  nvarchar(урт) байна
  lastName varchar(50) NULL,
  //сургууль төгссөн он нь 2016 гэх мэт утга авах тул 4 урттай тэмдэгтэн
  төрлөөр зарлав
  schoolGradYear varchar(4) NULL,
  //төрсөн газрын лавлахтай холбогдох код тул бүхэл тоон төрлөөр зарлав
  birthPlace tinyint NULL,
  //УЦ20311032 бүтэцтэй кирилл үсэг бүхий утга авах тул nvarchar(10)
  regNumber varchar(10) NOT NULL,
  //утасны дугаар нь тоон утга авах ч бодолт хийгдэхгүй тул char төрлөөр
  зарлах нь тохиромжтой байдаг
  mobilePhone char(8) NULL
)
```

## Хүснэгт засварлах

**Багана нэмэх: Student хүснэгтэнд birthday гэсэн огноо төрөлтэй багана нэмэх**

```
ALTER TABLE Student
  ADD birthday date;
```

**Багана устгах: Student хүснэгтээс birthday баганыг устгах**

```
ALTER TABLE Student
  DROP COLUMN birthday;
```

**Багана засварлах:**

<b>CHANGE</b>	Баганы нэр болон төрлийг хамт шинэчилнэ
<b>MODIFY</b>	Зөвхөн төрийг шинэчилнэ
<b>RENAME COLUMN</b>	Зөвхөн баганы нэрийг шинэчилнэ
<b>ALTER</b>	Зөвхөн default утга тодорхойлно

**Student хүснэгтийн mobilePhone баганыг mobilePhone, төрлийг varchar(8) болгож өөрчлөх**

```
ALTER TABLE Student
  CHANGE mobilePhone mPhone varchar(8);
```

**Багана засварлах/MySQL/: Student хүснэгтийн mobilePhone баганы төрлийн varchar(8) болгож өөрчлөх**

```
ALTER TABLE Student
  MODIFY mobilePhone varchar(8);
```

**Багана засварлах/MSSQL/: Student хүснэгтийн mobilePhone баганы төрлийн varchar(8) болгож өөрчлөх**

```
ALTER TABLE Student  
    ALTER COLUMN mobilePhone varchar(8);
```

**Student хүснэгтийн mobilePhone баганыг mobilePhone мPhone нэртэй болгож өөрчлөх**

```
ALTER TABLE Student  
    RENAME COLUMN mobilePhone TO мPhone;
```

**Student хүснэгтийн mobilePhone баганыг mobilePhone-д default утга тодорхойлох**

```
ALTER TABLE Student  
    ALTER mobilePhone SET DEFAULT '99999999';
```

**Багана засварлах командаар default утга тодорхойлох:**

```
ALTER TABLE Student  
    add constraint df DEFAULT '1' for SchoolYear
```

**Багана засварлах командаар default хязгаарлалыг устгах/MySQL/:**

```
ALTER TABLE Student ALTER SchoolYear DROP DEFAULT;
```

**Багана засварлах командаар default хязгаарлалыг устгах/MSSQL/:**

```
ALTER TABLE Student ALTER COLUMN SchoolYear DROP DEFAULT;
```

**Хүснэгт устгах**

```
DROP TABLE [IF EXISTS]  
tbl_name [, tbl_name] ...
```

**Жишээ:**

```
DROP TABLE product;
```

## Хязгаарлал тодорхойлох жишээнүүд

**Жишээ 1: (Default): Student хүснэгтийн schoolGradYear талбарт өгөгдмөл утга тодорхойлох.** Энэ нь ихэнхдээ 2016 гэсэн утга авдаг гэвэл:

```
ALTER TABLE Student  
    ALTER schoolGradYear SET DEFAULT 2016
```

**Жишээ 2: (Check) Төрсөн газар баганад 1-9 хооронд утга авах хязгаарлал нэмэх**

```
ALTER TABLE Student  
ADD CONSTRAINT ch_place CHECK (birthPlace >= 1 and irthPlace<10);  
GO;
```

**Жишээ 3: (Check) Оюутны код баганы утгыг шалгах хязгаарлал нэмэх**

MySQL: LIKE Condition

Wildcard	Explanation
%	Дурын урттай дурын тэмдэгт
_	Нэг дурын тэмдэгт

```
MySQL /BXXXXXXXXX бүртэцтэй өгөгдөл зөвхөн зөвшөөрнө/:  
ALTER TABLE Student  
ADD CONSTRAINT CK_Stud_id CHECK (StudCode LIKE 'B_____');
```

MSSQL/B999999999 бүртэцтэй өгөгдөл зөвхөн зөвшөөрнө/::

```
ALTER TABLE Student  
ADD CONSTRAINT CK_Stud_id  
CHECK (StudCode LIKE  
'[B][1-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]');
```

### Хэв тааруулалт –REGEXP, RLIKE

Where нөхцөл болон check constraint – хязгаарлалт хэсэгт **хэв тааруулалт** ашиглаж зөв бүтэцтэй эсэхийг шалгаж болно.

Хэв тааруулалт- Regular Expressions нь оролт хэсэгт өгөгдсөн мэдээллийг шалгах бүтцийг тусгай тэмдгээр тодорхойж зөв эсэхийг шалгана.

#### MYSQL: expr REGEXP pattern

```
ALTER TABLE Student ADD CONSTRAINT CK_Stud_id  
CHECK (StudCode REGEXP '[B][1-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]');
```

#### MSSQL\Oracle: expr LIKE pattern

```
ALTER TABLE Student ADD CONSTRAINT CK_Stud_id  
CHECK (StudCode LIKE '[B][1-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]');
```

#### Pattern –Тусгай тэмдэг

^	Эхлэл
\$	Төгсгөл
[a-z]	a-z жижг тэмдэгт
[A-Z]	A-Z том тэмдэгт
[0-9]	0-9 цифр
[^0-9]	0-9 цифр агуулаагүй
?	Хоосон эсвэл нэг
*	Хоосон эсвэл олон
+	Нэг болон түүнээс олон
{2}	2 удаа
{2,}	Хоёр түүнээс их
{2,4}	2 -- 4 удаа
.	Дурын нэг тэмдэгт
(a b)	a эсвэл b
\s	Хоосон зай (known as whitespace)

**Жишээ 4: (Unique -баганы утга нь NULL утга авах ба давтагдаж болно бусад утга нь давтагдашгүй байна ) Оюутан хүснэгтийн регистрийн дугаар багана давтагдашгүй утга авахыг тодорхойлох**

```
ALTER TABLE Student  
ADD CONSTRAINT u_reg_no  
UNIQUE (regNumber_no);
```

**Жишээ 5: (Primary Key) Member хүснэгтийн member\_no баганыг анхдагч түлхүүр болгох**

```
ALTER TABLE member
    ADD CONSTRAINT PK_member_no
        PRIMARY KEY (member_no);
```

**Жишээ 6: (Foreign Key) Student хүснэгтийн birthPlace баганыг Place хүснэгтийн place\_id баганатай гадаад түлхүүрээр холбох**

```
ALTER TABLE Student
    ADD CONSTRAINT fk_bPlace
        FOREIGN KEY(birthPlace) REFERENCES Place(place_id);
```

**Жишээ7: (Foreign Key) Олон баганаас бүрдэх гадаад түлхүүр тодорхойлох.**

Доорх жишээнд Sales хүснэгтийн stor\_id, ord\_num, title\_id гэсэн гурван баганыг Sales\_Achieve хүснэгтийн stor\_id, ord\_num, title\_id гурван баганатай гадаад түлхүүрээр холбож байна.

```
ALTER TABLE Sales
    CONSTRAINT FK_order
        FOREIGN KEY (stor_id, ord_num, title_id)
        REFERENCES Sales_Achieve (stor_id, ord_num, title_id);
```

**Хязгаарлалыг түр зогсоох боломжтой.**

**Жишээ/MySQL/:** Student хүснэгтийн ch\_Place хязгаарлалын үйлчилгээг зогсоох

```
USE StudentRegistration
ALTER TABLE Student
    ALTER CONSTRAINT ch_Place NOT ENFORCED;
```

**Үйлчилгээг сэргээхдээ:**

```
USE StudentRegistration
ALTER TABLE Student
    ALTER CONSTRAINT ch_Place ENFORCED;
```

**Жишээ/MSSQL/:** Student хүснэгтийн ch\_Place хязгаарлалын үйлчилгээг зогсоох

```
USE StudentRegistration
ALTER TABLE Student
    NOCHECK CONSTRAINT ch_Place;
```

**Үйлчилгээг сэргээхдээ:**

```
USE StudentRegistration
ALTER TABLE Student
    CHECK CONSTRAINT ch_Place;
```

**Хязгаарлалыг устгах**

**Жишээ:** Student хүснэгтийн анхдагч түлхүүрийг устгая

```
ALTER TABLE Student
    DROP CONSTRAINT PK_StudentID;
```

**Жишээ:** Student хүснэгтийн schoolGradYear багана дээр тодорхойлсон default хязгаарлалыг устгая

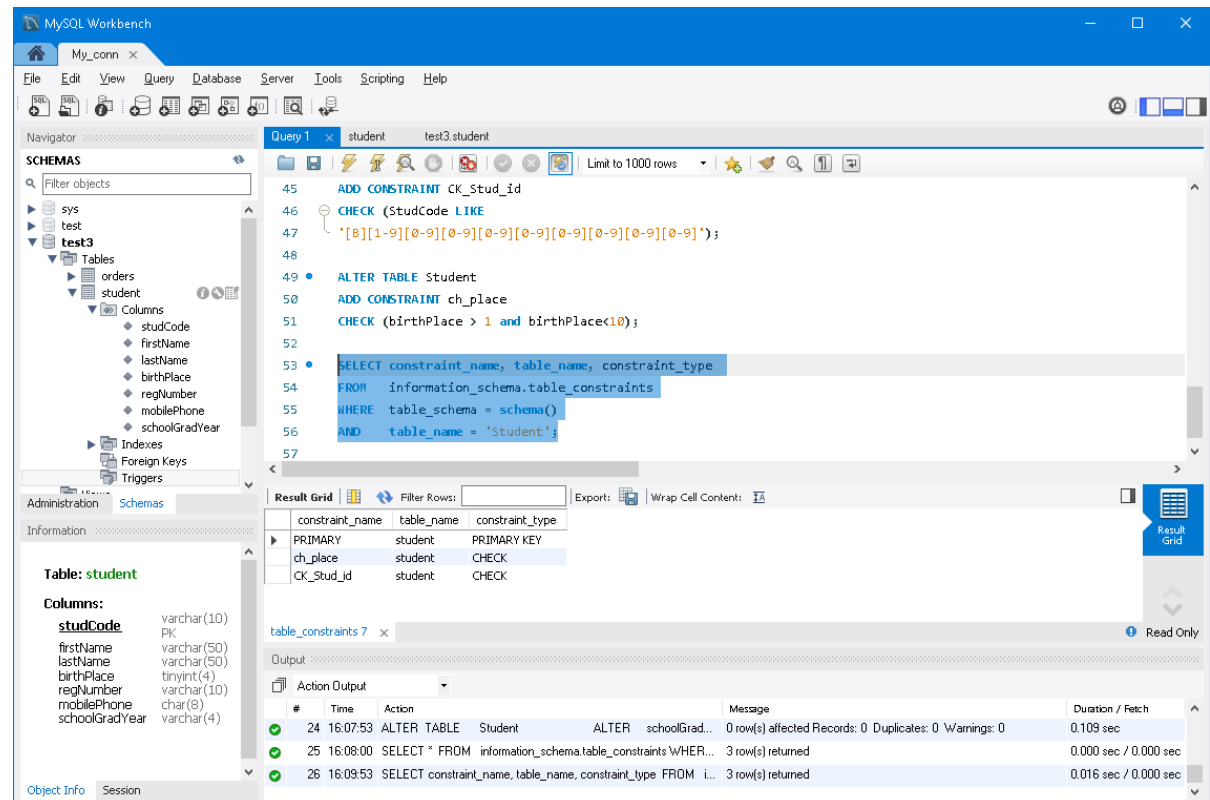
```
ALTER TABLE Student
    ALTER COLUMN schoolGradYear DROP DEFAULT;
```

### Хязгаарлалын нэрийг харах:

```
SELECT constraint_name, table_name, constraint_type FROM  
information_schema.table_constraints  
WHERE table_schema = schema() AND table_name = 'Student';
```

### Эсвэл

```
select * from information_schema.table_constraints  
where Table_name = 'хүснэгт нэр'
```

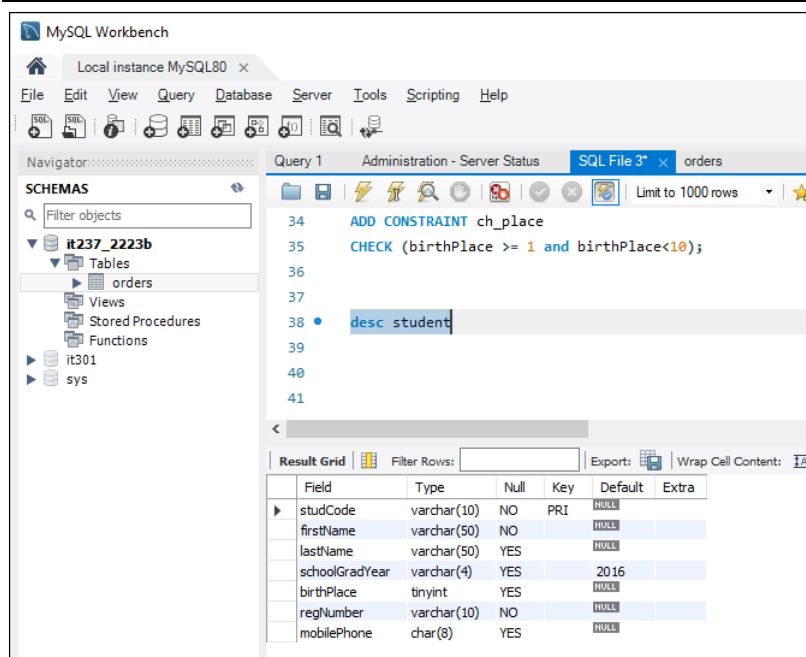


Зураг 1. Хязгаарлалын нэрийг харах

### Хүснэгтийн тодохойлолтыг харуулах:

```
DESC Student;
```

```
DESCRIBE student;
```



Зураг 2. Хүснэгтийн тодорхойлолтыг харгах

## Хүснэгтэнд бичлэг нэмэх /оруулах

### Бичигдэх хэлбэр:

```
INSERT INTO table_name (Column1,Column2...)
VALUES (ColumnValue1, ColumnValue2...)
```

### Жишээ:

```
CREATE TABLE person (
per_id INT IDENTITY(1,1) NOT NULL PRIMARY KEY,
per_first_name VARCHAR(255) NOT NULL ,
per_last_name VARCHAR(255) NOT NULL ,
per_dob DATE NULL ,
);
```

### Үүсгэсэн хүснэгтэндээ 2 бичлэг оруулья:

```
INSERT INTO person (per_first_name, per_last_name)
VALUES ('first1', 'last1');
INSERT INTO person (per_first_name, per_last_name, per_dob)
VALUES ('first2', 'last2', '2000.01.01');
```

per\_dob атрибутыг null гэж заасан учраас утга оруулахгүй байж болно.

Мөн per\_id атрибутыг auto increment гэж тодорхойлсон тул энэ атрибутын утгыг оруулахгүй, ӨС өөрөө автоматаар нэмэгдэх утгыг оруулж байна. Мөн нэг командаар хэд хэдэн бичлэг оруулж болно:

```
INSERT INTO person
( per_first_name,
per_last_name,
```

```
per_dob )  
VALUES  
( 'first1', 'last1', null ),  
( 'first2', 'last2', '2000.01.01' );  
•Атрибутын жагсаалт нь тодорхой байх ёстой.
```

Бичлэг нэмэх үйлдлийн үед хэрэв бүх атрибутад утга оруулах бол атрибутын жагсаалтыг бичих шаардлагагүй. Жишээ нь: дараах 2 INSERT үйлдэл нь ижил утгатай:

```
INSERT INTO person  
(per_first_name, per_last_name, per_dob)  
VALUES ( 'first2', 'last2', '2000.01.01' );  
INSERT INTO person  
VALUES ( NULL, 'first2', 'last2', '2000.01.01' );
```

Хэрэв баганы жагсаалтаас ялгаатай тооны утгын жагсаалт оруулсан байвал SQL алдааны мэдээлэл өгнө:

```
INSERT INTO person VALUES  
( 'first2', 'last2', '2000.01.01' );
```

## Тэмдэгтийн олонлог ба collation

**Тэмдэгтийн олонлог тодорхойлох**

```
CREATE DATABASE StudentReg  
COLLATE Latin1_General_100_CS_AS_SC;  
GO;
```

**Collation** –ий мэдээлэл харах:

- SELECT CONVERT (varchar, SERVERPROPERTY('collation'));
- EXECUTE sp\_helpsort;
- SELECT name, description FROM sys.fn\_helpcollations();
- SELECT name, collation\_name FROM sys.databases;
- SELECT CONVERT (varchar, DATABASEPROPERTYEX('database\_name', 'collation'));

**Баганы collation-г харах**

```
SELECT name, collation_name FROM sys.columns  
WHERE name = '<insert character data type column name>';
```