

ПРАКТИКАЛЫҚ ЖҰМЫС №4. SQL ТІЛІНЕ КІРІСПЕ. SQL ТІЛІНДЕ КЕСТЕЛЕРДІ ЖӘНЕ ШЕКТЕУЛЕРДІ ҚҰРУ.

4.1. Практикалық сабақтың мақсаты

Transact – SQL құрылымдық сұранымдар тілін үйрену. Индекстер және кестелермен жұмыс жасайтын SQL-операторларды үйрену. Кестені құру, өзгерту және жоюға арналған sql командаларды, шектеулерді қолдану.

4.2. Кіріс мәліметтер

Кіріс мәліметтер болып жеке тапсырмалары бойынша орындалған жұмыстар мен практикалық жұмыс нәтижелері табылады.

4.3. Қолданылатын программалар

"SQL Server Management Studio" программасы және Microsoft SQL Server сервері.

4.4. Теориялық мәлімет

4.4.1. Transact – SQL программалау негіздері

SQL тілінің атқаратын қызметтері

- *Деректерді ұйымдастыру.*
- *Деректерді оқу.*
- *Деректерді оңдеу.*
- *Мүмкіндіктерді басқару.*
- *Деректерді бірігіп қолдану.*
- *Деректер бүтіндігі*

Комментарийлердің 2 түрі бар: жолдық 2 «-» символы және блоктық, /* ... */.

4.4.2. Кестені құруға, өзгертуге және жоюға арналған SQL командалар
Data Definition Language (DDL) метамәліметтерді басқаруға арналған SQL
тілінің бөлігі.

Кестелер Transact-SQL тілінің келесі командаларымен басқарылады:

CREATE TABLE – құру

ALTER TABLE - өзгерту

DROP TABLE - жою

Transact-SQL қолданылатын типтер:

Деректер типінің атауы	Класс	Өлшемі (байт)	Деректер қасиеттері
Bit	Integer	1	0 немесе 1 мәндерін сақтау үшін
Bigint	Integer	8	$-2^{62} \dots 2^{62}-1$ (шамамен плюс және минус 92 квадриллион)
Int	Integer	4	-2147483648 .. 2147483647 аралығындағы бүтін сандар ($-2^{31} \dots 2^{31}-1$)
SmallInt	Integer	2	-32768 .. 32767 аралығындағы бүтін сандар ($-2^{15} \dots 2^{15}-1$)
TinyInt	Integer	1	0 .. 255 аралығындағы бүтін сандар
Decimal, Numeric	Decimal/ Numeric	әртүрлі	жылжамалы үтірлі нақты сан. Екі атау синоним болып табылады.
Money	Money	8	акша бірлігінің типі
SmallMoney	Money	4	-214748,3648 .. +214748,3647 аралығындағы акша бірлігінің типі
Float	Float	8	-1,79E+308 .. 1,79E+308 аралығындағы нақты сан
Real	Real	4	нақты сан
DateTime	Data/Time	8	1753 жылдың 1- қаңтарынан 9999 жылдың 31- желтоқсанына дейінгі мерзім және уақыт типі
SmallDate- Time	Data/Time	4	1900 жылдың 1- қаңтарынан 2079 жылдың 6- маусымына дейінгі мерзім және уақыт типі
Char	Character		1- ден 8000 символға дейінгі нақты ұзындықтағы мәтіндік тип
VarChar	Character	әртүрлі	1- ден 8000 символға дейінгі айнымалы ұзындықтағы мәтіндік тип
Text	Character	әртүрлі	өлшемі 2 Гб- ка дейінгі үлкен көлемді символдық деректер типі
Ntext	Unicode	әртүрлі	максимум 1073741823 символға дейінгі Unicode-дерек символдары
Binary	Binary	әртүрлі	максимал ұзындығы 8000 байтқа дейінгі нақты ұзындықты екілік деректер
VarBinary	Binary	әртүрлі	максимал ұзындығы 8000 байтқа дейінгі айнымалы ұзындықты екілік деректер
Image	Binary	әртүрлі	максимал ұзындығы 2147483647 байтқа дейінгі айнымалы ұзындықты екілік деректер бұл өрісте бейне және кез- келген екілік және сонымен қатар мәтіндік ақпарат сақталады

- CREATE TABLE (Transact-SQL) -кестені құру операторы

Жазылуы:

```
CREATE TABLE название_таблицы
(өріс атауы1 мәлімет типі өріс атрибуты1,
өріс атауы2 мәлімет типі өріс атрибуты2,
.....
өріс атауыN мәлімет типі өріс атрибутыN,
кесте атрибуттары
```

)

Мысалы:

```
CREATE TABLE Customers
```

(

```
Id INT,
```

```
Age INT,
```

```
FirstName NVARCHAR(20),
```

```
LastName NVARCHAR(20),
```

```
Email VARCHAR(30),
```

```
Phone VARCHAR(20)
```

)

- PRIMARY KEY - бастапқы кілт немесе уникальды индекс

```
1. CREATE TABLE Customers
```

(

```
Id INT PRIMARY KEY,
```

```
Age INT,
```

```
FirstName NVARCHAR(20),
```

```
LastName NVARCHAR(20),
```

```
Email VARCHAR(30),
```

```
Phone VARCHAR(20)
```

)

```
2. CREATE TABLE Customers
```

(

```
Id INT,
```

```
Age INT,
```

```
FirstName NVARCHAR(20),
```

```
LastName NVARCHAR(20),
```

```
Email VARCHAR(30),
```

```
Phone VARCHAR(20),
```

```
PRIMARY KEY(Id)
```

)

- DEFAULT - келісім бойынша алынған мән

```
CREATE TABLE Customers
```

(

```
Id INT PRIMARY KEY IDENTITY,
```

```
Age INT DEFAULT 18,
```

```
FirstName NVARCHAR(20) NOT NULL,
```

```
LastName NVARCHAR(20) NOT NULL,
```

```
Email VARCHAR(30) UNIQUE,
```

```
Phone VARCHAR(20) UNIQUE
```

);

- **IDENTITY** - мәні берілген қадамға артып отыратын өріс. **IDENTITY** қасиеті **tinyint, smallint, int, bigint, decimal(p,0)** немесе **numeric(p,0)** типтеріне қатысты.

```
CREATE TABLE Customers
(
  Id INT PRIMARY KEY IDENTITY,
  Age INT,
  FirstName NVARCHAR(20),
  LastName NVARCHAR(20),
  Email VARCHAR(30),
  Phone VARCHAR(20)
)
```

- **UNIQUE**-уникальды кілт
- **NULL | NOT NULL** -нөлдік мән болу/болмауын көрсету.

```
CREATE TABLE Customers
(
  Id INT PRIMARY KEY IDENTITY,
  Age INT,
  FirstName NVARCHAR(20) NOT NULL,
  LastName NVARCHAR(20) NOT NULL,
  Email VARCHAR(30) UNIQUE,
  Phone VARCHAR(20) UNIQUE
)
```

- **CONSTRAINT** -шектеудің аты көрсетіледі. **PRIMARY KEY, NOT NULL, UNIQUE, FOREIGN KEY** немесе **CHECK** мәндерінде.

FOREIGN KEY REFERENCES-Бағынышты кілт

```
CREATE TABLE child_table
(
  column1 datatype [ NULL | NOT NULL ],
  column2 datatype [ NULL | NOT NULL ],
  ...
  CONSTRAINT fk_name
  FOREIGN KEY (child_col1, child_col2, ... child_col_n)
  REFERENCES parent_table (parent_col1, parent_col2, ... parent_col_n)
  [ ON DELETE { NO ACTION | CASCADE | SET NULL | SET DEFAULT } ]
  [ ON UPDATE { NO ACTION | CASCADE | SET NULL | SET DEFAULT } ]
);

CREATE TABLE Orders
```

```
(
  Id INT PRIMARY KEY IDENTITY,
  CustomerId INT,
  CreatedAt Date,
  CONSTRAINT FK_Orders_To_Customers FOREIGN KEY (CustomerId) REFERENCES Customers(Id)
);
```

ON DELETE және ON UPDATE

ON DELETE және **ON UPDATE** көмегімен мәліметтерді өзгерту немесе жою кезінде негізгі кестедегі жазбаларға орындалатын әрекетті анықтауға болады:

- **CASCADE**: негізгі кестеде жазба өзгертілсе немесе жойылса, бағынышты кестеде автоматты түрде өзгереді немесе жойылады.
- **NO ACTION**: негізгі кестеде жазба өзгертілсе немесе жойылса да, бағынышты кестеде ешқандай әрекет жасалмайды.
- **SET NULL**: негізгі кестеде жазба өзгертілсе немесе жойылса, бағынышты кестеде сыртқы кілт мәніне NULL мәні қойылады.
- **SET DEFAULT**: негізгі кестеде жазба өзгертілсе немесе жойылса, бағынышты кестеде сыртқы кілт мәніне DEFAULT мәнімен көрсетілген келісім бойынша алынған мән қойылады. Егер берілмесе, NULL мәні қойылады.

CHECK - өріс мәндері үшін шектеулер қояды.

Мысалы:

```
CREATE TABLE Customers
```

```
(
  Id INT PRIMARY KEY IDENTITY,
  Age INT DEFAULT 18,
  FirstName NVARCHAR(20) NOT NULL,
  LastName NVARCHAR(20) NOT NULL,
  Email VARCHAR(30) UNIQUE,
  Phone VARCHAR(20) UNIQUE,
  CHECK((Age > 0 AND Age < 100) AND (Email != "") AND (Phone != ""))
) – 3 өріске шектеу қойылып тұр.
```

- Кесте құрылымын өзгерту

```
ALTER TABLE table_name
ADD column_1 column-definition,
column_2 column-definition,
...
column_n column_definition;
```

Кестеге жаңа өрістер қосу:

```
ALTER TABLE employees  
ADD last_name VARCHAR(50),  
first_name VARCHAR(40);
```

Кестедегі өрісті өзгерту:

```
ALTER TABLE employees  
ALTER COLUMN last_name VARCHAR(75) NOT NULL;
```

Кестедегі өрісті жою:

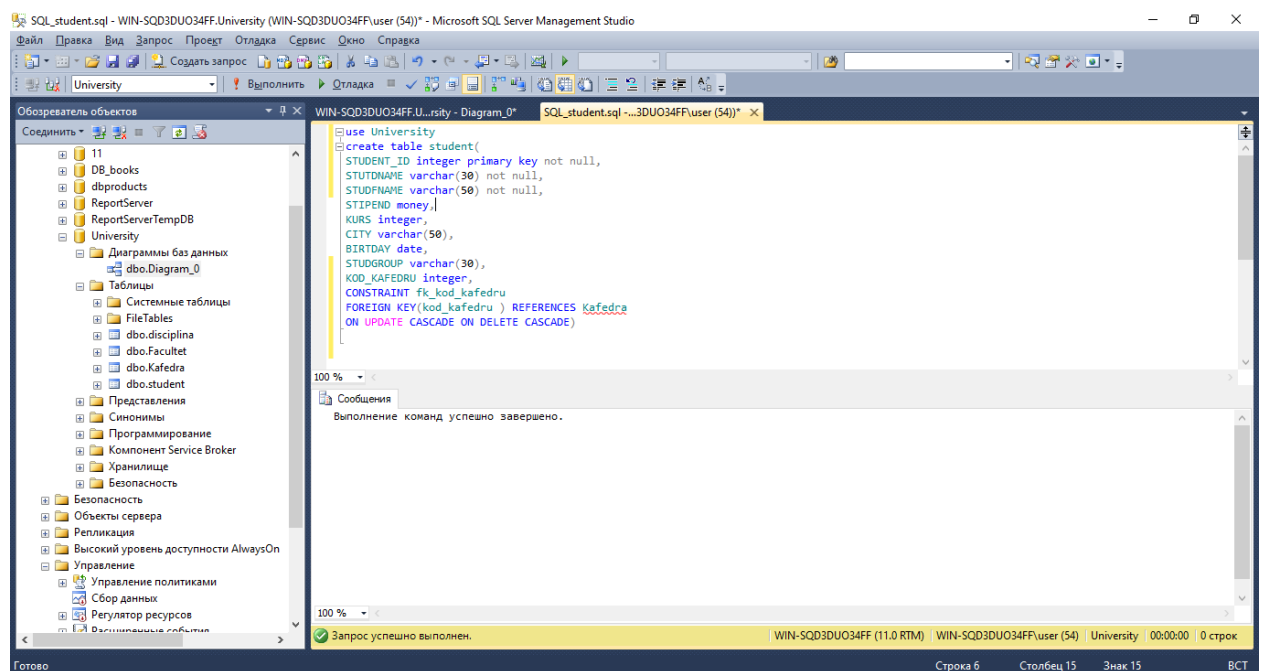
```
ALTER TABLE employees  
DROP COLUMN last_name;
```

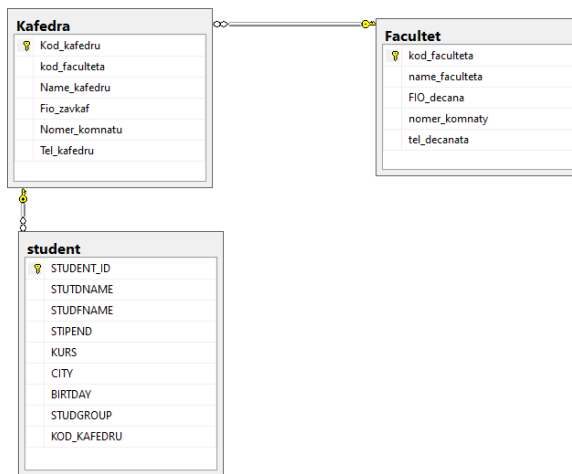
- Кестені жою операторы:

```
DROP TABLE table1 [, table2, ...]
```

```
DROP TABLE Customers
```

Құрыңыздар:





SQLQuery3.sql - WIN-SQD3DUO34FF.master (WIN-SQD3DUO34FF\user (52)) - Microsoft SQL Server Management Studio

Файл Правка Вид Запрос Проект Отладка Сервис Окно Справка

master Выполнить Отладка

Обозреватель объектов

Соединить

11 DB_books dbproducts ReportServer ReportServerTempDB University

Диаграммы баз данных

dbo.Diagram_0

Таблицы

Системные таблицы

FileTables

dbo.disciplina

dbo.Facultet

dbo.Kafedra

dbo.student

Представления

Синонимы

Программирование

Компонент Service Broker

Хранилище

Безопасность

Безопасность

Объекты сервера

Репликация

Высокий уровень доступности AlwaysOn

Управление

Управление политиками

Сбор данных

Регулятор ресурсов

Регистрируемые события

SQLQuery3.sql - WIN-SQD3DUO34FF\user (52)

```

create table teacher(
    KOD_TEACHER integer not null primary key,
    KOD_KAFEDRU integer,
    NAME_TEACHER varchar(150),
    INDEF_KOD integer UNIQUE,
    DOLGNOST varchar(70) check(DOLGNOST in ('профессор','доцент','ара окытуш','окытуш')) default 'окытуш',
    ZVANIE varchar(50) check(ZVANIE in ('доктор','кандидат','phd','магистр')) default 'жок',
    SALARY decimal default 100000,
    RISE decimal check(RISE>0) default 0,
    DATA_HIRE date default getdate(),
    BIRTHDAY date,
    POL varchar(6) check(pol in('male','female'))],
    TEL_TEACHER CHAR(10) NOT NULL CHECK (TEL_TEACHER LIKE '[1-9][0-9]-[0-9][0-9]-[0-9][0-9]');
  
```

100 %

Сообщения

Выполнение команд успешно завершено.

100 %

Запрос успешно выполнен. WIN-SQD3DUO34FF (11.0 RTM) WIN-SQD3DUO34FF\user (52) master 00:00:00 0 строк

Готово Строка 12 Столбец 47 Знак 47 BCT

SQLQuery4.sql - WIN-SQD3DUO34FF.University (WIN-SQD3DUO34FF\user (55)) - Microsoft SQL Server Management Studio

Файл Правка Вид Запрос Проект Отладка Сервис Окно Справка

University Выполнить Отладка

Обозреватель объектов

Соединить

11 DB_books dbproducts ReportServer ReportServerTempDB University

Диаграммы баз данных

dbo.Diagram_0

Таблицы

Системные таблицы

FileTables

dbo.disciplina

dbo.Facultet

dbo.Kafedra

dbo.student

dbo.teacher

Представления

Синонимы

Программирование

Компонент Service Broker

Хранилище

Безопасность

Безопасность

Объекты сервера

Репликация

Высокий уровень доступности AlwaysOn

Управление

Управление политиками

Сбор данных

Регулятор ресурсов

Регистрируемые события

SQLQuery4.sql - WIN-SQD3DUO34FF\user (55) SQL_teacher.sql - WIN-SQD3DUO34FF\user (52)

```

use University
ALTER TABLE teacher ADD CONSTRAINT fk_kod_kafedru1
FOREIGN KEY (kod_kafedru ) REFERENCES kafedra
ON UPDATE CASCADE ON DELETE CASCADE
  
```

100 %

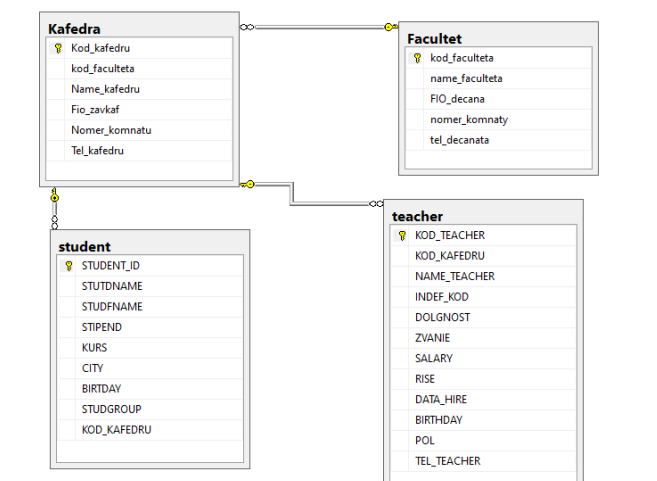
Сообщения

Выполнение команд успешно завершено.

100 %

Запрос успешно выполнен. WIN-SQD3DUO34FF (11.0 RTM) WIN-SQD3DUO34FF\user (55) University 00:00:00 0 строк

Готово Строка 2 Столбец 51 Знак 51 BCT



Кестеге мәлімет енгізу, өзгерту және жою (DML командалары)

Енгізу командасы:

```
INSERT <Имя таблицы> [(<Список полей>)]
```

```
VALUES (<Значения полей>)
```

Мұндағы <Кесте аты> – мәліметтер енгізілетін кесте аты, (<Өрістер тізімі>) – мәліметтер енгізілетін өрістер тізімі, егер көрсетілмесе барлық өріс алынады, (<Өрістер тізімі>) – үтір арқылы бөлінген өріске енгізілетін мәндер тізімі.

Мән ретінде Default тұрақтысы көрсетілсе, келісім бойынша алынған мән енгізіледі.

Мысал: Студенттер кестесіне жазбаны енгізу ФАӨ = Иванов, Адрес= Актобе, Код специальности = 5

```
INSERT Студенттер (ФАӨ, Адрес, [Мамандық коды])
```

```
VALUES ('Иванов А.А.', 'Актобе', 5)
```

```
INSERT INTO Products (ProductName, Price, Manufacturer)
```

```
VALUES ('iPhone 6S', 41000, 'Apple')
```

Кестеден жазбаларды жою командасы:

```
DELETE <Кесте аты>
```

```
[WHERE <Шарт>]
```

<Шарт> - шартын қанағаттандыратын жазбалар жойылады, егер шарт көрсетілмесе кестенің барлық жазбасы жойылады.

Мысалы: Студенттер кестесінен Адрес = Актобе болатын жазбаларды жою

```
DELETE Студенты
```

```
WHERE Адрес = 'Актобе'
```

Өзгерту командасы:

```
UPDATE <Кесте аты>
```

```
SET
```

```
<Өріс аты1> = <Өрнек1>,
```

```
[<Өріс аты 2> = <Өрнек 2>],
```

```
...
```

```
[WHERE <Шарт>]
```

```
UPDATE Products
```



```
SET Manufacturer = 'Samsung Inc.'  
WHERE Manufacturer = 'Samsung'
```

Енгізу командасының көмегімен мәліметті енгізіңіз:

Kafedra 10 жазбамен,

Teacher 20 жазбамен,

Student 40 жазбамен толтырыңыз.

Тапсырма:

- 1. Жеке тақырыптарыңыз бойынша кесте құрыңыздар. Программалық жолмен бастапқы және бағынышты кілтті анықтаңыздар. Құрылған кестелерді 5 жазбадан толтырыңыздар.**
2. Кестені құру операторының толық жазылуын конспектілеп келіңіздер.