



НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ”
ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ

Інформаційно-комунікаційні системи: Бази даних

Лабораторна робота №1

Виконав:
Поночевний Назар ФІ-92
Перевірив:
Кіфорчук К. О.

ЛАБОРАТОРНА РОБОТА №1

СТВОРЕННЯ БАЗИ ДАНИХ

Ціль заняття: Надбання навичок настройки і роботи в середовищі SSMS. Визначення синтаксису SQL: CREATE TABLE.

Результати заняття: Створення бази даних у відповідності з варіантом завдання.

Завдання для лабораторної роботи

1. Вибрати у відповідності до списку групи свій варіант завдання. Проаналізувати опис предметної області. Створити необхідні таблиці.
2. За допомогою SSMS побудувати ER-діаграму створеної бази даних.

Варіант № 12

Предметна область: **Приймальна комісія (абітурієнти)**. Основні предметно-значущі сутності: Абітурієнти, Спеціальності, Предмети. Основні предметно-значущі атрибути сутностей:

- Абітурієнти - прізвище, ім'я, по батькові, стать, дата народження, спеціальність;
- Спеціальності - назва спеціальності;
- Предмети - назва предмета, вид контролю.

Основні **вимоги** до функцій системи:

- Вибрати всіх абітурієнтів по спеціальностях або певної спеціальності;
- Вибрати всіх абітурієнтів, що здали вступні іспити, і їх рейтинг (сума балів за всіма зданих предметів) за спеціальностями або певної спеціальності;
- Підрахувати середній бал з дисциплін і спеціальностей.

Тригери:

1. На додавання запису про результати здачі іспиту. Якщо для зазначеного абітурієнта є оцінка з дисципліни, заборонити додавання запису.
2. Створити **представлення** «Відмінники» з полями «ПІБ абітурієнта», «Дисципліна», «Оцінка», що містить результати вище N. Оновлювати представлення «Відмінники».

Процедура:

Процедура повинна повернути середній бал зазначеного абітурієнта.

1) Код з коментарями:

```
-- Створити таблиці

DROP TABLE IF EXISTS grades, subjects, enrollees, specialties;
GO

CREATE TABLE "enrollees"(
    "enrolleeid" INT IDENTITY,
    "name" NVARCHAR(255) NOT NULL,
    "surname" NVARCHAR(255) NOT NULL,
    "fathername" NVARCHAR(255) NULL,
    "sex" BIT NULL,
    "birthday" DATE NOT NULL,
    "specialtyid" INT NOT NULL
);
ALTER TABLE
    "enrollees" ADD CONSTRAINT "enrollees_enrolleeid_primary" PRIMARY
KEY("enrolleeid");
CREATE TABLE "specialties"(
    "specialtyid" INT IDENTITY,
    "name" NVARCHAR(255) NOT NULL UNIQUE
);
ALTER TABLE
    "specialties" ADD CONSTRAINT "specialties_specialtyid_primary" PRIMARY
KEY("specialtyid");
CREATE TABLE "subjects"(
    "subjectid" INT IDENTITY,
    "name" NVARCHAR(255) NOT NULL,
    "checktype" NVARCHAR(255) NOT NULL
);
ALTER TABLE
    "subjects" ADD CONSTRAINT "subjects_subjectid_primary" PRIMARY
KEY("subjectid");
CREATE TABLE "grades"(
    "enrolleeid" INT NOT NULL,
    "subjectid" INT NOT NULL,
    "grade" FLOAT NOT NULL CHECK("grade" BETWEEN 0 AND 100)
);
ALTER TABLE
    "enrollees" ADD CONSTRAINT "enrollees_specialtyid_foreign" FOREIGN
KEY("specialtyid") REFERENCES "specialties"("specialtyid") ON UPDATE
CASCADE;
ALTER TABLE
    "grades" ADD CONSTRAINT "grades_enrolleeid_foreign" FOREIGN
KEY("enrolleeid") REFERENCES "enrollees"("enrolleeid") ON UPDATE CASCADE;
ALTER TABLE
```

```
"grades" ADD CONSTRAINT "grades_subjectid_foreign" FOREIGN  
KEY("subjectid") REFERENCES "subjects"("subjectid") ON UPDATE CASCADE;  
GO
```

```
-- Вставити дані в них
```

```
INSERT INTO specialties  
  (name)
```

```
VALUES
```

```
  ('Applied Mathematics'),  
  ('Computer Science'),  
  ('Communication'),  
  ('Business'),  
  ('Biological and Biomedical Sciences'),  
  ('Agriculture'),  
  ('Aviation'),  
  ('Education'),  
  ('Finance'),  
  ('Applied Physics');
```

```
INSERT INTO enrollees
```

```
  (name, surname, fathurname, sex, birthday, specialityid)
```

```
VALUES
```

```
  ('Oleksiy', 'Chorny', 'Maksimovich', 1, '2000-10-31', 2),  
  ('Maria', 'Shyshko', 'Yurivna', 0, '2001-12-23', 2),  
  ('Pavlo', 'Kopylov', 'Oleksandrovich', 1, '2000-04-11', 1),  
  ('Katya', 'Bondar', 'Ostapivna', 0, '2000-02-14', 3),  
  ('Olya', 'Simonova', 'Olegivna', 0, '1999-06-03', 2),  
  ('Olesya', 'Folina', 'Yurivna', 0, '1999-03-29', 4),  
  ('Oleg', 'Dydkov', 'Yosipovich', 1, '2000-02-11', 5),  
  ('Gleb', 'Gigo', 'Oleksandrovich', 1, '2001-10-14', 5),  
  ('Matviy', 'Kopylov', 'Timoviyovich', 1, '2000-11-02', 6),  
  ('Misha', 'Martynov', 'Volodymyrovich', 1, '2000-12-07', 7),  
  ('Vova', 'Elter', 'Yosipovich', 1, '2002-11-24', 1);
```

```
INSERT INTO subjects
```

```
  (name, checktype)
```

```
VALUES
```

```
  ('Ukrainian language and literature', 'ZNO'),  
  ('Mathematics', 'ZNO'),  
  ('English', 'ZNO'),  
  ('Physics', 'ZNO'),  
  ('Motivational letter', 'Letter'),  
  ('Creative competition', 'Competition'),  
  ('History', 'ZNO');
```

```
INSERT INTO grades
```

```
        (enrolleeid, subjectid, grade)
```

```
VALUES
```

```
    (1, 1, 56),  
    (1, 2, 81.2),  
    (1, 6, 80),  
    (2, 1, 91.6),  
    (2, 2, 94),  
    (2, 3, 89),  
    (3, 1, 77),  
    (3, 2, 86.1),  
    (4, 1, 65),  
    (4, 4, 93.2),  
    (4, 2, 75),  
    (4, 5, 100),  
    (4, 3, 79),  
    (5, 1, 38),  
    (5, 2, 43),  
    (6, 1, 68),  
    (6, 2, 84.6),  
    (6, 3, 100),  
    (6, 4, 51),  
    (6, 5, 96),  
    (6, 6, 70);
```

```
GO
```

```
-- Вибрати всіх абітурієнтів по спеціальностям або певної спеціальності
```

```
SELECT en.name AS firstname, en.surname AS surname, sp.name AS  
specialityname  
FROM enrollees AS en  
JOIN specialties AS sp ON en.specialityid = sp.specialityid  
WHERE sp.name IN ('Applied Mathematics', 'Business')  
ORDER BY sp.name ASC;
```

```
GO
```

```
-- Вибрати всіх абітурієнтів, що здали вступні іспити, і їх рейтинг (сума  
балів за всіма зданих предметів) за спеціальностями або певної  
спеціальності
```

```
SELECT  
    en.name AS firstname, en.surname AS surname, en.fathername AS  
fathername,  
    sp.name AS specialityname, gr.testsamount, gr.rating  
FROM (  
    SELECT  
        enrolleeid, SUM(grade) AS rating, COUNT(grade) AS  
testsamount
```

```

        FROM grades GROUP BY enrolleeid
    ) AS gr
    JOIN enrollees AS en ON gr.enrolleeid = en.enrolleeid
    JOIN specialties AS sp ON en.specialityid = sp.specialityid
WHERE sp.name IN (
    'Applied Mathematics',
    'Computer Science',
    'Biological and Biomedical Sciences',
    'Business')
ORDER BY gr.rating DESC;
GO

```

-- Підрахувати середній бал з дисциплін і спеціальностей

```

SELECT
    su.name AS subjectname, gr.testsamount, gr.avggrade
FROM (
    SELECT
        subjectid, AVG(grade) AS avggrade, COUNT(grade) AS
testsamount
        FROM grades GROUP BY subjectid
    ) AS gr
    JOIN subjects AS su ON gr.subjectid = su.subjectid
WHERE su.checktype = 'ZNO'
ORDER BY gr.avggrade DESC;
GO

```

```

SELECT
    sp.name AS specialityname, gr.testsamount, gr.avggrade
FROM (
    SELECT
        en.specialityid, AVG(grade) AS avggrade, COUNT(grade) AS
testsamount
        FROM grades AS gr
            JOIN enrollees AS en ON gr.enrolleeid = en.enrolleeid
        GROUP BY en.specialityid
    ) AS gr
    JOIN specialties AS sp ON gr.specialityid = sp.specialityid
ORDER BY gr.avggrade DESC;
GO

```

-- На додавання запису про результати здачі іспиту. Якщо для зазначеного абітурієнта є оцінка з дисципліни, заборонити додавання запису

```

DROP TRIGGER IF EXISTS notCopyTestGrade;
GO

```

```

CREATE TRIGGER notCopyTestGrade ON grades AFTER INSERT AS
BEGIN
    DECLARE @ienrolleeid INT
    DECLARE @isubjectid INT
    SELECT
        @ienrolleeid = enrolleeid, @isubjectid = subjectid
        FROM INSERTED
    DECLARE @icheck INT
    SELECT @icheck = COUNT(1) FROM grades
        WHERE enrolleeid = @ienrolleeid AND subjectid = @isubjectid
    IF @icheck > 1
    BEGIN
        ROLLBACK TRANSACTION;
        RAISERROR('Test grade for this student already exist.', 16, 1)
    END
END;
GO

```

```

INSERT INTO grades (enrolleeid, subjectid, grade) VALUES (1, 3, 87);
INSERT INTO grades (enrolleeid, subjectid, grade) VALUES (1, 3, 17);
GO

```

```

SELECT * FROM grades ORDER BY enrolleeid;
GO

```

-- Створити представлення «Відмінники» з полями «ПІБ абітурієнта», «Дисципліна», «Оцінка», що містить результати вище N. Оновлювати представлення «Відмінники»

```

CREATE OR ALTER VIEW excellences AS
    SELECT
        en.name AS firstname, en.surname AS surname, en.fathername AS
fathername,
        su.name AS subjectname, gr.grade AS grade
    FROM grades AS gr
        JOIN enrollees AS en ON gr.enrolleeid = en.enrolleeid
        JOIN subjects AS su ON gr.subjectid = su.subjectid
    WHERE gr.grade > 80;
GO

```

```

SELECT * FROM excellences ORDER BY grade DESC;
GO

```

-- Процедура повинна повернути середній бал зазначеного абітурієнта

```

DROP PROCEDURE IF EXISTS getAvgGrade;
GO

```

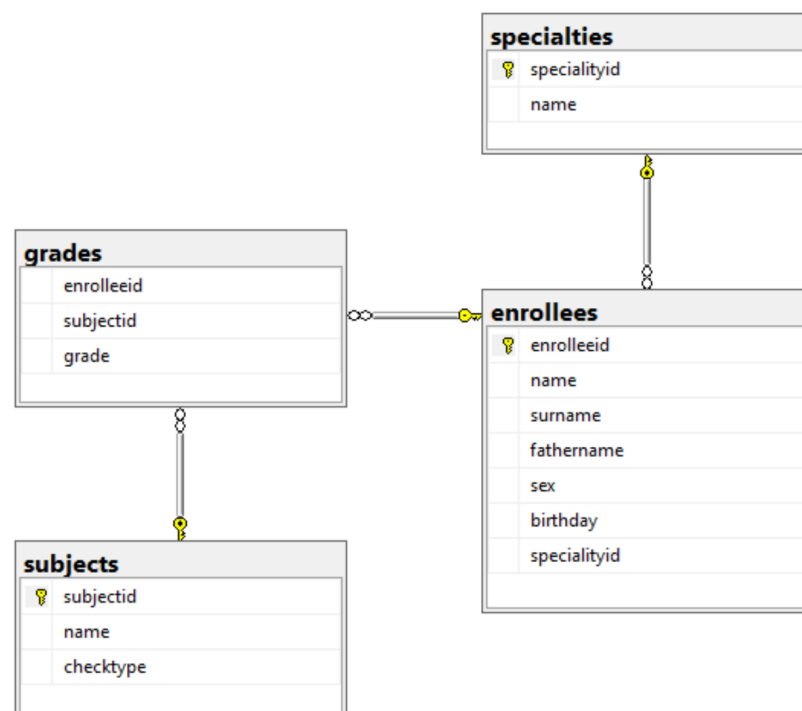
```

CREATE PROCEDURE getAvgGrade
    @firstname nvarchar(255), @surname nvarchar(255), @fathername
nvarchar(255)
AS
    SELECT
        gr.avggrade
    FROM (
        SELECT
            enrolleeid, AVG(grade) AS avggrade, COUNT(grade) AS
testsamount
        FROM grades GROUP BY enrolleeid
    ) AS gr
    JOIN enrollees AS en ON gr.enrolleeid = en.enrolleeid
    WHERE en.name = @firstname AND en.surname = @surname AND
en.fathername = @fathername;
GO

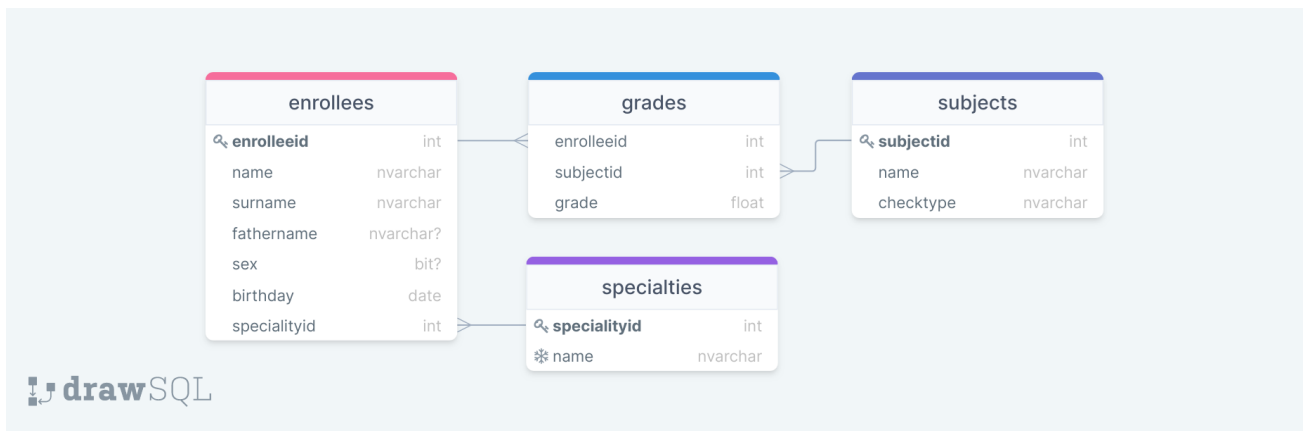
EXEC getAvgGrade @firstname = 'Pavlo', @surname = 'Kopylov', @fathername =
'Oleksandrovych';
GO

```

2) За допомогою SSMS побудувати ER-діаграму.



Але мені більше подобається діаграма у DrawSQL:



3) Вибрати всіх абітурієнтів по спеціальностям або певної спеціальності

	firstname	sumame	specialityname
1	Pavlo	Kopylov	Applied Mathematics
2	Vova	Elter	Applied Mathematics
3	Olesya	Folina	Business

4) Вибрати всіх абітурієнтів, що здали вступні іспити, і їх рейтинг (сума балів за всіма зданих предметів) за спеціальностями або певної спеціальності

	firstname	sumame	fathurname	specialityname	testsamount	rating
1	Olesya	Folina	Yurivna	Business	6	469.6
2	Maria	Shyshko	Yurivna	Computer Science	3	274.6
3	Oleksiy	Chomyi	Maksimovych	Computer Science	3	217.2
4	Pavlo	Kopylov	Oleksandrovych	Applied Mathematics	2	163.1
5	Olya	Simon...	Olegivna	Computer Science	2	81

5) Підрахувати середній бал з дисциплін і спеціальностей

	subjectname	testsamount	avggrade
1	English	3	89.33333333333333
2	Mathematics	6	77.31666666666667
3	Physics	2	72.1
4	Ukrainian I...	6	65.93333333333333

	specialityname	testsamount	avggrade
1	Communication	5	82.44
2	Applied Mathematics	2	81.55
3	Business	6	78.266...
4	Computer Science	8	71.6

6) Тригер: На додавання запису про результати здачі іспиту. Якщо для зазначеного абітурієнта є оцінка з дисципліни, заборонити додавання запису

```
Msg 50000, Level 16, State 1, Procedure notCopyTestGrade, Line 15 [Batch Start Line 188]
Test grade for this student already exist.
Msg 3609, Level 16, State 1, Line 191
The transaction ended in the trigger. The batch has been aborted.
```

	enrolleeid	subjectid	grade
1	1	1	56
2	1	2	81.2
3	1	6	80
4	1	3	87
5	2	1	91.6
6	2	2	94
7	2	3	89
8	3	1	77

7) Створити представлення «Відмінники» з полями «ПІБ абітурієнта», «Дисципліна», «Оцінка», що містить результати вище N. Оновлювати представлення «Відмінники»

	firstname	sumame	fathename	subjectname	grade
1	Katya	Bondar	Ostapivna	Motivational letter	100
2	Olesya	Folina	Yurivna	English	100
3	Olesya	Folina	Yurivna	Motivational letter	96
4	Maria	Shyshko	Yurivna	Mathematics	94
5	Katya	Bondar	Ostapivna	Physics	93.2
6	Maria	Shyshko	Yurivna	Ukrainian langu...	91.6
7	Maria	Shyshko	Yurivna	English	89
8	Oleksiy	Chomyi	Maksimo...	English	87

8) Процедура повинна повернути середній бал зазначеного абітурієнта

	avggrade
1	81.55