Національний технічний університет України

“Київський політехнічний інститут”

БАЗИ ДАНИХ ТА ІНФОРМАЦІЙНІ СИСТЕМИ

Розрахункова робота на тему:

“Створення бази даних результатів сесії”

Виконав:

Студент групи ФБ-74

Варіант 3

Заіграєв Костянтин

Перевірив:

Коломицев М.В.

Київ 2019

Мета роботи: Освоєння методів проектування баз даних і роботи з базами даних у середовищі СУБД MS SQL Server.

Варіант 3

Предметна область – Деканат.
Задачі, що вирішує інформаційна система – результати сесії.
Завдання на обробку даних:

1. Створити впорядковані списки:
   - Студентів груп 'С' другого курсу (за прізвищами);
   - Студентів, що мають максимальний середній бал у своїй групі (за середнім балом та прізвища);
   - Дисциплін, за якими немає іспитів (за алфавітом).
2. обчислення:
   - Обчислити стипендіальний фонд по групі, по курсу, факультету.
3. корекція:
   - Зміна оцінки з дисципліни для обраного студента.
4. Звіти виду:
   - "Студенти": факультет - курс - група - ПІБ - номер заліковки - середня оцінка.
   - "Кількість студентів": факультет - курс - кількість студентів (за денною формою)
   - "Кількість студентів": кількість студентів - форма навчання.
   - "Успішність": група - середній бал по групі-різниця (бал - середній бал). У запиті впорядкувати за збільшення різниці.

1. Завдання Створення діаграми потоків даних (DFD-моделі)

Після аналізу завдання, розробляємо можливу діаграму потоків даних. Для створення DFD-діаграми будемо користуватися сайтом draw.io

Для компактнішого і структурного проектування визначемо декілька підсистем.

Результат роботи видно на рисунках 1,2,3.

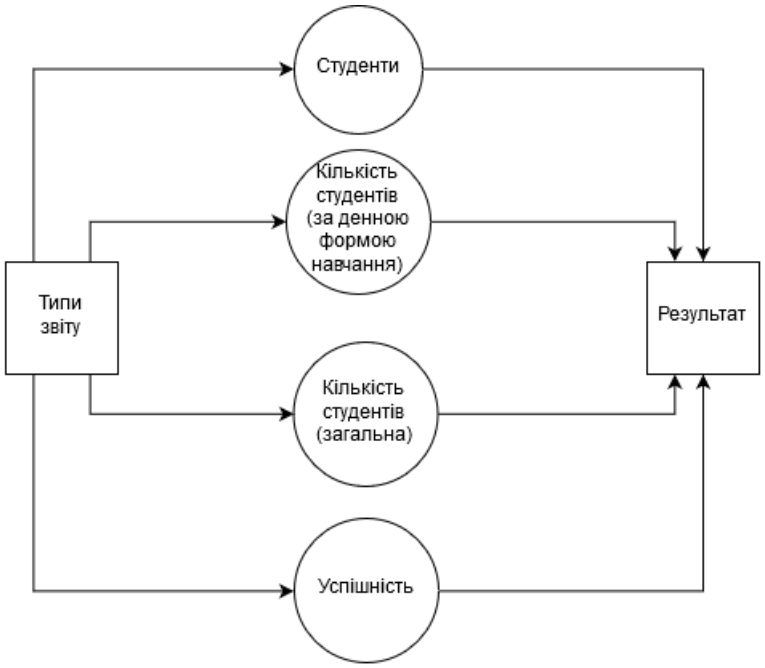


Рисунок 1. Підсистема “Визначення типу звіту”

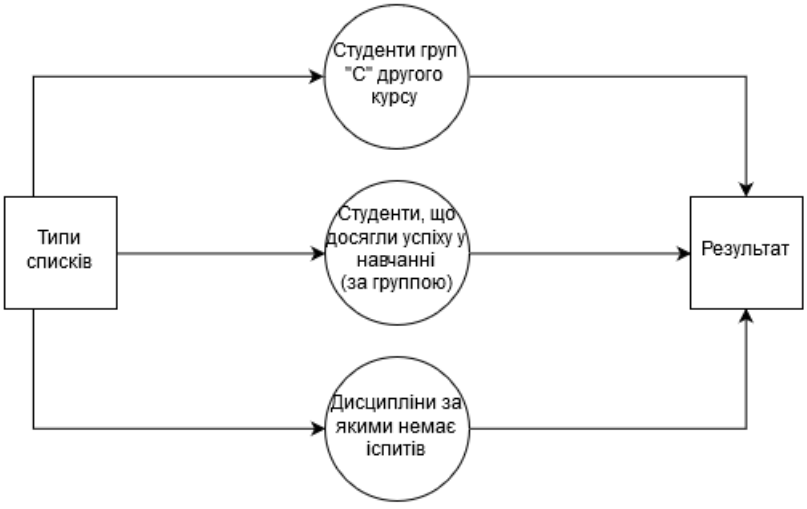


Рисунок 2. Підсистема “Визначення типу списків”

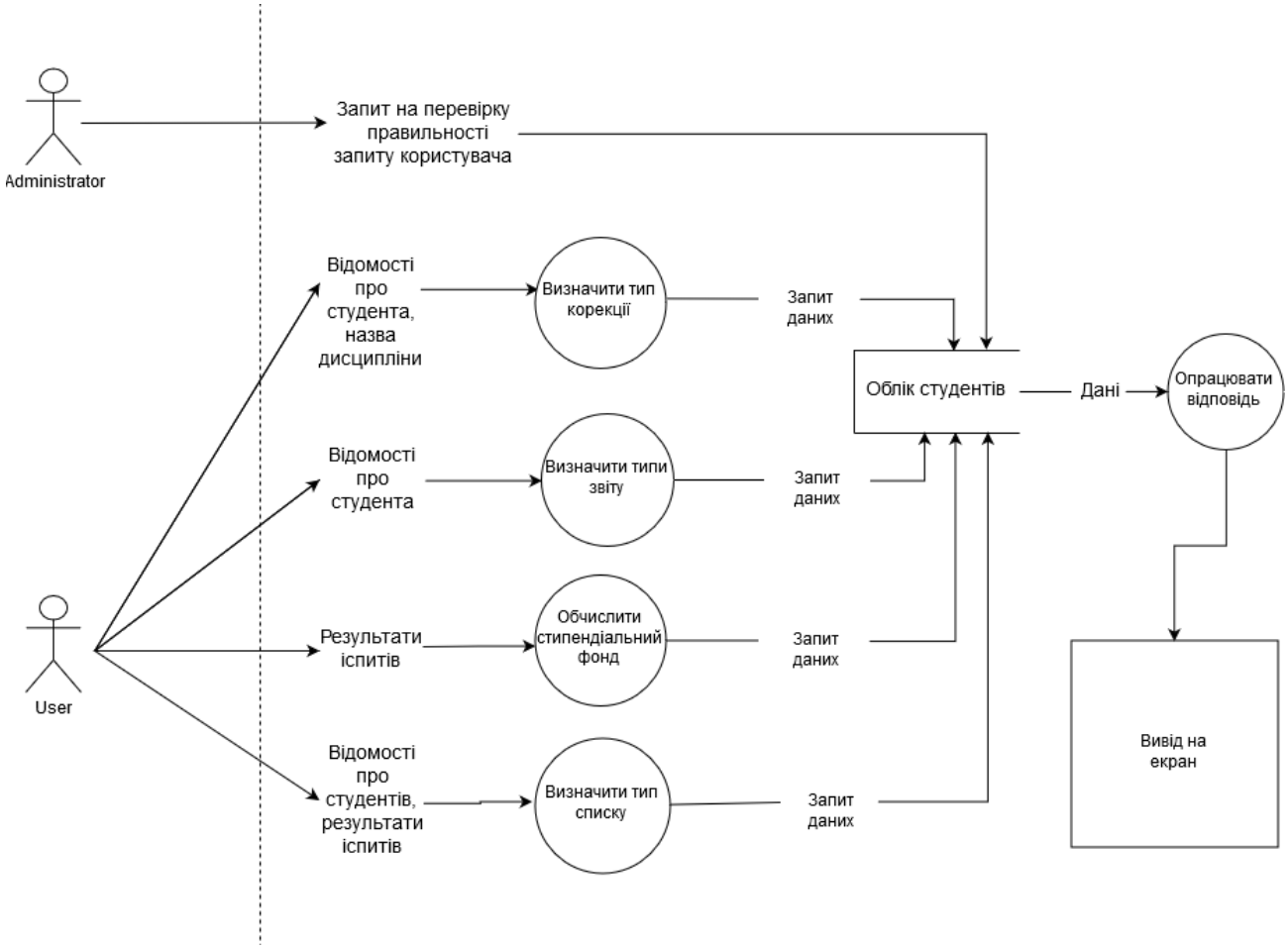


Рисунок 3. Головна діаграма потоків даних

2. Проектування бази данних в MSSQL

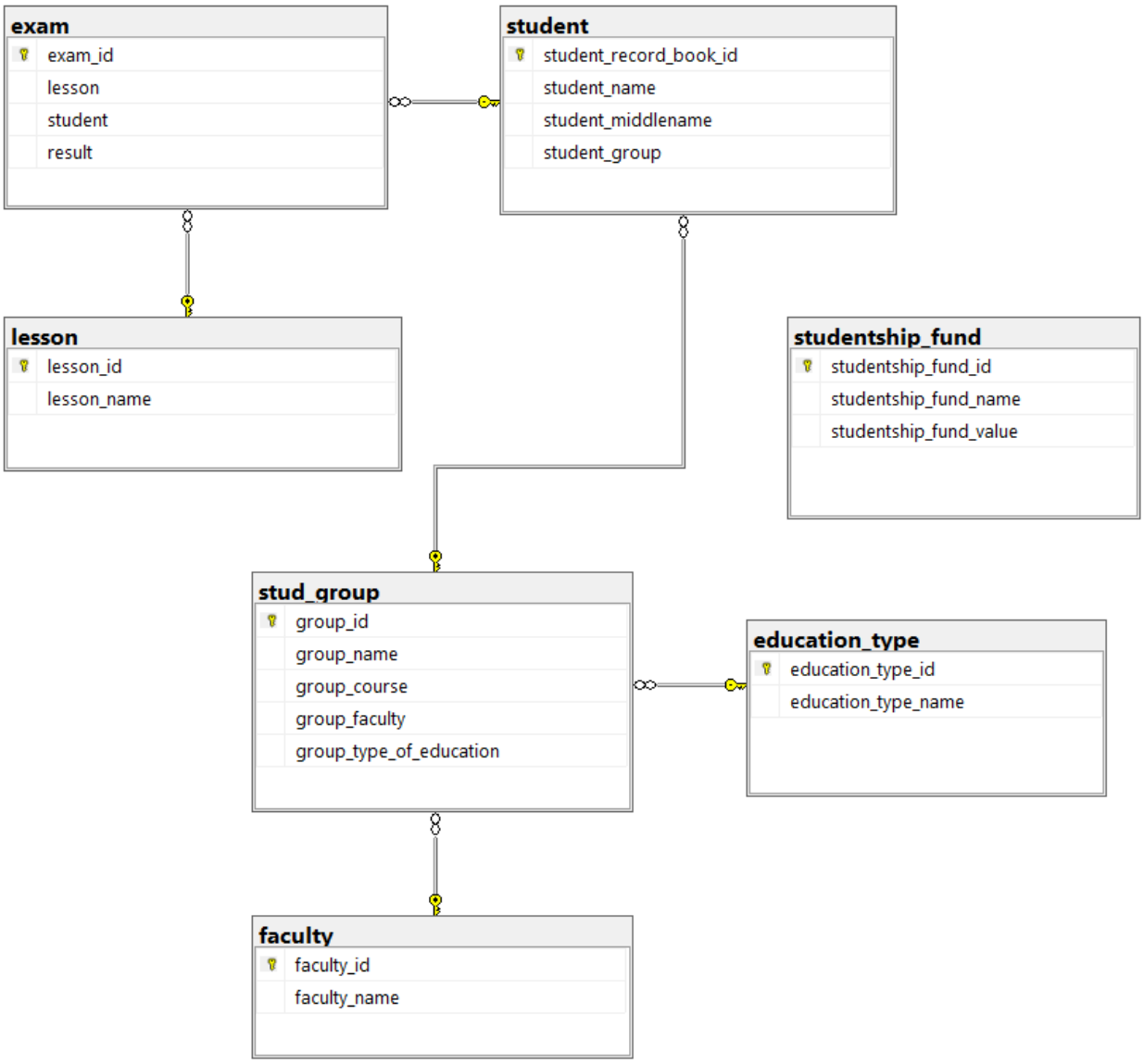
```
use rr_var3
go

drop table exam, student, lesson,  stud_group, education_type, faculty, studentship_fund


create table faculty (
        faculty_id int identity primary key not null,
        faculty_name varchar(40) not null
);

create table education_type(
        education_type_id int identity primary key not null,
        education_type_name varchar(40) not null
);

create table stud_group (
        group_id int identity primary key not null,
        group_name varchar(20) not null,
        group_course int not null,
        group_faculty int foreign key references faculty(faculty_id),
        constraint course_valid check (group_course between 1 and 6),
        group_type_of_education int foreign key references education_type(education_type_id) not null
);

create table lesson (
```

```
        lesson_id int identity primary key not null,
        lesson_name varchar(40) not null
);

create table student (
        student_record_book_id int identity primary key not null,
        student_name varchar(40) not null,
        student_middlename varchar(40) not null,
        student_group int foreign key references stud_group(group_id) not null,
--      student_faculty int foreign key references faculty(faculty_id) not null
);

create table exam (
        exam_id int identity primary key not null,
        lesson int foreign key references lesson(lesson_id),
        student int foreign key references student(student_record_book_id),
        result int not null,
        constraint result_check check (result between 0 and 100)
);

create table studentship_fund (
        studentship_fund_id int identity primary key not null,
        studentship_fund_name varchar(40) not null,
        studentship_fund_value int not null
);

use rr_var3
go

------------------------Studentship fund
insert into     studentship_fund (studentship_fund_name, studentship_fund_value) values ('Повышенная', 100);
insert into     studentship_fund (studentship_fund_name, studentship_fund_value) values ('Академическая',
80);
------------------------Faculty
insert into
        faculty (faculty_name)
        values
        ('IPT');
insert into
        faculty (faculty_name)
        values
        ('IASA');
        ----------------------Education-type
insert into education_type(education_type_name) values ('очна')
insert into education_type(education_type_name) values ('заочна')
------------------------Student Groups
insert into
        stud_group (group_name, group_course, group_faculty, group_type_of_education)
        values          ('fb-91', 1, 1, 1);
        -- first year student of FB on IPT
--insert into
--      stud_group (group_name, group_course, group_faculty)
--      values          ('ff-91', 1, 1);                                                                --
first year student of FF on IPT
insert into
        stud_group (group_name, group_course, group_faculty, group_type_of_education)
        values          ('fb-81', 2, 1, 1);
        -- second year student of FB on IPT
insert into
        stud_group (group_name, group_course, group_faculty, group_type_of_education)
        values          ('ff-81', 2, 1, 2);
        -- second year student of FF on IPT
insert into
        stud_group (group_name, group_course, group_faculty, group_type_of_education)
        values          ('fb-71', 3, 1, 1);
        -- third year student of FB on IPT
--insert into
--      stud_group (group_name, group_course, group_faculty)
--      values          ('ff-71', 3, 1);                                                                --
third year student of FF on IPT
insert into
        stud_group (group_name, group_course, group_faculty, group_type_of_education)
```

```
        values          ('fb-61', 4, 1, 1);
        -- fourth year student of FB on IPT

        --------- IPT over
insert into
        stud_group (group_name, group_course, group_faculty, group_type_of_education)
        values          ('ia-91', 1, 2, 1);
        -- first year student of FM on FMM
--insert into
--      stud_group (group_name, group_course, group_faculty)
--      values          ('is-91', 1, 2);                                                                    --
first year student of FF on FMM
insert into
        stud_group (group_name, group_course, group_faculty, group_type_of_education)
        values          ('ia-81', 2, 2, 1);
        -- second year student of FM on FMM
insert into
        stud_group (group_name, group_course, group_faculty, group_type_of_education)
        values          ('is-81', 2, 2, 2);
        -- second year student of FF on FMM
insert into
        stud_group (group_name, group_course, group_faculty, group_type_of_education)
        values          ('ia-71', 3, 2, 1);
        -- third year student of FM on FMM
--insert into
--      stud_group (group_name, group_course, group_faculty)
--      values          ('is-71', 3, 2);                                                                    --
third year student of FF on FMM
insert into
        stud_group (group_name, group_course, group_faculty, group_type_of_education)
        values          ('ia-61', 4, 2, 1);
        -- fourth year student of FM on FMM

        ------------------------Lessons
insert into lesson (lesson_name) values ('Высшая математика')
insert into lesson (lesson_name) values ('Физика')
insert into lesson (lesson_name) values ('Украинский язык')
insert into lesson (lesson_name) values ('Английский язык')
insert into lesson (lesson_name) values ('Теория вероятности')
insert into lesson (lesson_name) values ('Теория алгоритмов')
insert into lesson (lesson_name) values ('ИКС. Базы данных')
insert into lesson (lesson_name) values ('Специальные разделы программирования')


        ------------------------Students
        -- fb-91
insert into
        student(student_name, student_middlename, student_group)
        values ('Пётр', 'Петров', 1)
insert into
        student(student_name, student_middlename, student_group)
        values ('Анастасия', 'Щукина', 1)
insert into
        student(student_name, student_middlename, student_group)
        values ('Андрей', 'Белов', 6)
insert into
        student(student_name, student_middlename, student_group)
        values ('Злата', 'Молчанова', 6)
        -- ff-91
insert into
        student(student_name, student_middlename, student_group)
        values ('Ульяна', 'Ковальчук', 2)
insert into
        student(student_name, student_middlename, student_group)
        values ('Устин', 'Данилов', 2)
insert into
        student(student_name, student_middlename, student_group)
        values ('Елена', 'Лыткина', 7)
insert into
        student(student_name, student_middlename, student_group)
        values ('Константин', 'Жданов', 7)
        -- fb-81
insert into
```

```
        student(student_name, student_middlename, student_group)
        values ('Прохор', 'Кабанов', 3)
insert into
        student(student_name, student_middlename, student_group)
        values ('Харитон', 'Погомий', 3)
insert into
        student(student_name, student_middlename, student_group)
        values ('Леонард', 'Яковлев', 8)
insert into
        student(student_name, student_middlename, student_group)
        values ('Рафаил', 'Рыбаков', 8)
        -- ff-81
insert into
        student(student_name, student_middlename, student_group)
        values ('Зинаида', 'Дзюба ', 4)
insert into
        student(student_name, student_middlename, student_group)
        values ('Таисия ', 'Шевченко', 4)
insert into
        student(student_name, student_middlename, student_group)
        values ('Богдан', 'Палий', 9)
insert into
        student(student_name, student_middlename, student_group)
        values ('Игнатий', 'Суханов', 9)
        -- fb-71
insert into
        student(student_name, student_middlename, student_group)
        values ('Яромир', 'Романенко', 5)
insert into
        student(student_name, student_middlename, student_group)
        values ('Жигер', 'Шухевич', 5)
insert into
        student(student_name, student_middlename, student_group)
        values ('Андреев', 'Алексей', 10)
insert into
        student(student_name, student_middlename, student_group)
        values ('Алексей', 'Новиков', 10)

                ----------------------Exams
        insert into exam(lesson, student, result) values (1, 1, 65)
        insert into exam(lesson, student, result) values (2, 1, 90)
        insert into exam(lesson, student, result) values (1, 2, 100)
        insert into exam(lesson, student, result) values (2, 2, 95)
        insert into exam(lesson, student, result) values (1, 3, 76)
        insert into exam(lesson, student, result) values (2, 3, 84)
        insert into exam(lesson, student, result) values (1, 4, 60)
        insert into exam(lesson, student, result) values (2, 4, 82)
        insert into exam(lesson, student, result) values (1, 5, 60)
        insert into exam(lesson, student, result) values (5, 5, 60)
        insert into exam(lesson, student, result) values (1, 6, 97)
        insert into exam(lesson, student, result) values (5, 6, 93)
        insert into exam(lesson, student, result) values (5, 7, 70)
        insert into exam(lesson, student, result) values (6, 7, 61)
        insert into exam(lesson, student, result) values (5, 8, 60)
        insert into exam(lesson, student, result) values (6, 8, 91)
        insert into exam(lesson, student, result) values (8, 9, 60)
        insert into exam(lesson, student, result) values (8, 10, 90)
        insert into exam(lesson, student, result) values (7, 11, 89)
        insert into exam(lesson, student, result) values (7, 12, 64)
        insert into exam(lesson, student, result) values (6, 13, 72)
        insert into exam(lesson, student, result) values (6, 14, 70)
        insert into exam(lesson, student, result) values (5, 15, 87)
        insert into exam(lesson, student, result) values (5, 16, 100)
        insert into exam(lesson, student, result) values (6, 17, 89)
        insert into exam(lesson, student, result) values (6, 18, 73)
        insert into exam(lesson, student, result) values (5, 19, 72)
        insert into exam(lesson, student, result) values (5, 20, 88)
```

3. ER-діаграма



4. Створення необхідного функціоналу

4.1. Створення впорядкованих списків

А) Студентів груп 'С' другого курсу (за прізвищами);

```
create view list_first
as
	select
		st.student_name, st.student_middlename, stg.group_name
	from student st
	inner join
		stud_group stg
		on stg.group_id = st.student_group
	where stg.group_course = 2
with check option
```

```
go

select * from list_first
	go
```

В) Студентів, що мають максимальний середній бал у своїй групі (за середнім балом та прізвища);

```
create view list_second
as
	select
		st.student_name, st.student_middlename, stg.group_name as grpn, avg(e.result)
as average_result
	from student st
		inner join
			exam e
			on e.student = st.student_record_book_id
		inner join
			stud_group stg
			on stg.group_id = st.student_group
		inner join
		(select tt.group_id, max(tt.average_result_by_group) as average_for_student
			from
			(select
				st.student_name, stg.group_id ,avg(e.result) as
average_result_by_group
			from exam e
			inner join
				student st
				on st.student_record_book_id = e.student
			inner join
				stud_group stg
				on stg.group_id = st.student_group
			group by stg.group_name, stg.group_id, st.student_name) as tt
			group by tt.group_id) as test
		on test.group_id = stg.group_id
		group by student_name, st.student_middlename, stg.group_name,
test.average_for_student
		having avg(e.result) = max(test.average_for_student)
with check option
go

select * from list_second order by grpn, average_result desc
	go
```

С) Дисциплін, за якими немає іспитів (за алфавітом).

```
create view list_third
as
	select
		l.lesson_name
	from
		lesson l
	where l.lesson_id not in (select e.lesson from exam e)
with check option
go

select * from list_third
	go
```

4.2 Обчислення

А) Обчислити стипендіальний фонд по групі, по курсу, факультету.

```
create proc calculation
(
	@type_of_studship int = 1
)
as
	begin;
	select
		res.styp * stf.studentship_fund_value as final_fund_for_each_group
	from
	(
	select
		stg.group_name, count(st.student_name) as styp
	from
		student st--, studentship_fund stf
		inner join
			stud_group stg
			on stg.group_id = st.student_group
		inner join
			exam e
			on e.student = st.student_record_book_id
		inner join
			(
		select
			stg.group_id, avg(e.result) as average
		from
			exam e
			inner join
			student st on st.student_record_book_id = e.student
			inner join
			stud_group stg on stg.group_id = st.student_group
		group by stg.group_id
			) as tt
		on tt.group_id = stg.group_id
	group by stg.group_name, st.student_name, tt.average
	having avg(e.result) > tt.average
	) as res, studentship_fund stf
	where stf.studentship_fund_id = @type_of_studship
		end;
	go
exec calculation 2
	go
```

4.3 Корекція

А) Зміна оцінки з дисципліни для обраного студента.

```
create proc correction
(
	@lesson int,
	@student int,
	@result int
)
as
	begin;
	update
		exam
	set
		result =@result
	where
		lesson = @lesson and student = @student
```

```
	end;
go

exec correction 1, 1, 100
go
```

4.4 Створення звіту

А) "Студенти": факультет - курс - група - ПІБ - номер заліковки - середня оцінка.

```
select
	f.faculty_name, stg.group_course, stg.group_name, st.student_record_book_id,
st.student_name, st.student_middlename, avg(e.result) as average
from
	student st
	inner join
		exam e
		on e.student = st.student_record_book_id
	inner join
		stud_group stg
		on stg.group_id = st.student_group
	inner join
		faculty f
		on f.faculty_id = stg.group_faculty
	group by
		f.faculty_name, stg.group_course, stg.group_name, st.student_record_book_id,
st.student_name, st.student_middlename
```

В) "Кількість студентів": факультет - курс - кількість студентів (за денною формою)

```
select
	f.faculty_name, stg.group_course, count(st.student_record_book_id) as number
	from
		student st
		inner join
			stud_group stg
			on stg.group_id = st.student_group
		inner join
			faculty f
			on f.faculty_id = stg.group_faculty
		group by
			f.faculty_name, stg.group_course, stg.group_type_of_education
		having stg.group_type_of_education <> 2
```

С) "Кількість студентів": кількість студентів - форма навчання.

```
select
	et.education_type_name, count(st.student_record_book_id) as number
from
	student st
	inner join
		stud_group stg
		on stg.group_id = st.student_group
	inner join
		education_type et
		on et.education_type_id = stg.group_type_of_education
group by
	et.education_type_name
```

D) "Успішність": група - середній бал по групі-різниця (бал - середній бал). У запиті впорядкувати за збільшення різниці.

```
	select
	st.student_name, st.student_middlename, avg(e.result)-tt.average as
distinction_of_result_for_each_student
from
	student st
	inner join
		stud_group stg
		on stg.group_id = st.student_group
	inner join
		exam e
		on e.student = st.student_record_book_id
	inner join
		(
			select
				stg.group_id, avg(e.result) as average
			from
				student st
				inner join
					stud_group stg
					on stg.group_id = st.student_group
				inner join
					exam e
					on e.student = st.student_record_book_id
			group by stg.group_id
		) as tt
		on tt.group_id = stg.group_id
group by
	st.student_name, st.student_middlename, tt.average
```

5. Перевірка нормалізації розробленої моделі

Для цього необхідно перевірити чи належить дана модель до третьої нормальної форми. Для цього модель повинна належати до першої і другої форми, а також не повинна мати транзитивних відношень, тобто залежностей не ключевого атрибуту від іншого не ключового.

Для початку перевіримо належність до першої нормальної форми. Для цього усі атрибути повинні буди атомарні і повинні бути відсутні групи, які повторюються. Виконується.

Далі перевіримо належність до другої нормальної групи. Це буде виконуватись, якщо модель належить до першої нормальної групи (це вже

перевірено), а також будуть відсутні неповні функціональні залежності не ключових атрибутів первинного ключа. Виконується.

Залишилось перевірити відсутність транзитивних відношень, тобто ні одне з ключових полів не має ідентифікуватися за допомогою іншого ключового поля. Значить розроблена модель належить до третьої нормальної форми. Виконується.

Висновок

У ході виконання розрахунково-графічної роботи, я дослідив методи проектування баз даних, навчився проектувати власну базу даних на сайті draw.io, а також створювати функціонал для потреб користувача в MS SQL SERVER.