

Національний технічний університет України
"Київський політехнічний інститут імені Ігоря Сікорського"
Фізико-технічний інститут

Розрахунково-графічна робота
з кредитного модуля «Бази даних та ІКС»
Варіант № 4

Виконала
студентка 4 курсу ФТІ
Колодяжна Олена
Групи ФІ-72
Перевірив:
Коломицев М.В.

Оцінка:

Варіант № 4

Предметна область – Деканат.

Задачі, що вирішує інформаційна система – навчальне навантаження викладачів.

Завдання на обробку даних:

1. Створити впорядковані по кафедрам списки:

- Кафедра - викладач - дисципліна;
- Викладачів, у яких посада не відповідає наукового ступеня (для кандидата - доцент, для доктора - професор).

2. Обчислення:

- Переконатися, що у кожного викладача не більше 3-х занять на день.

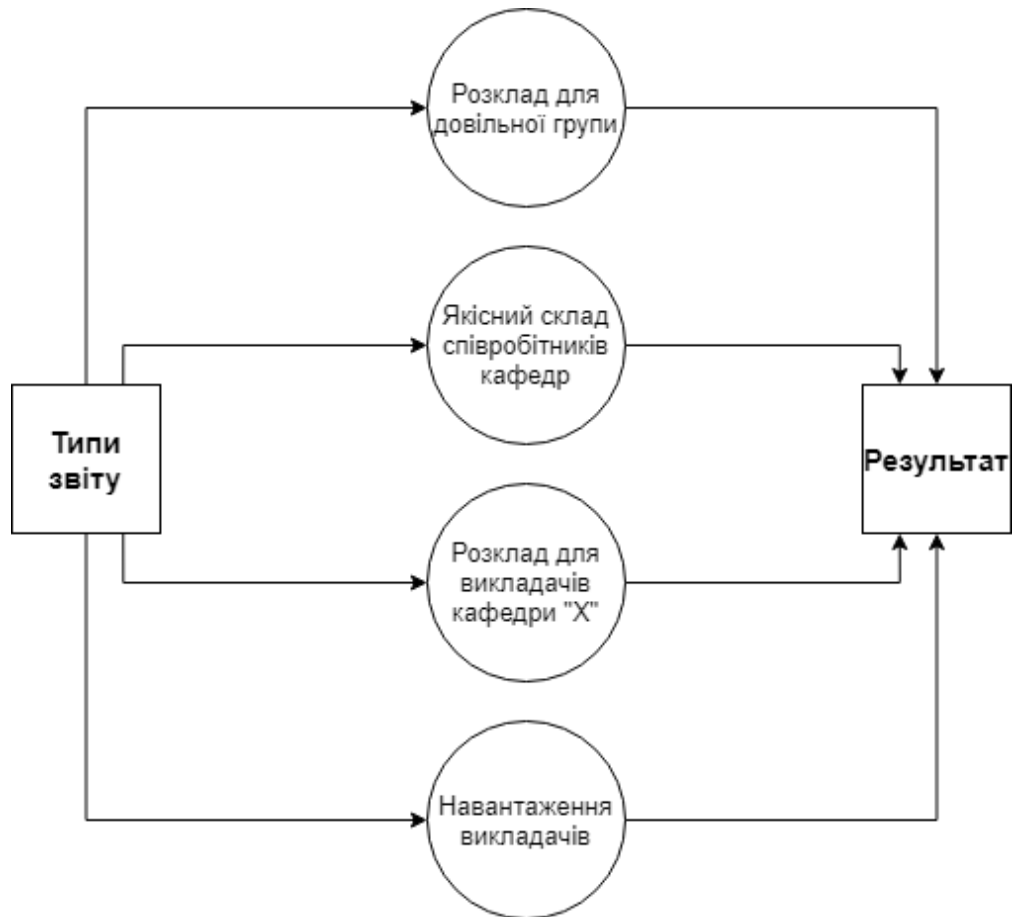
3. Корекція:

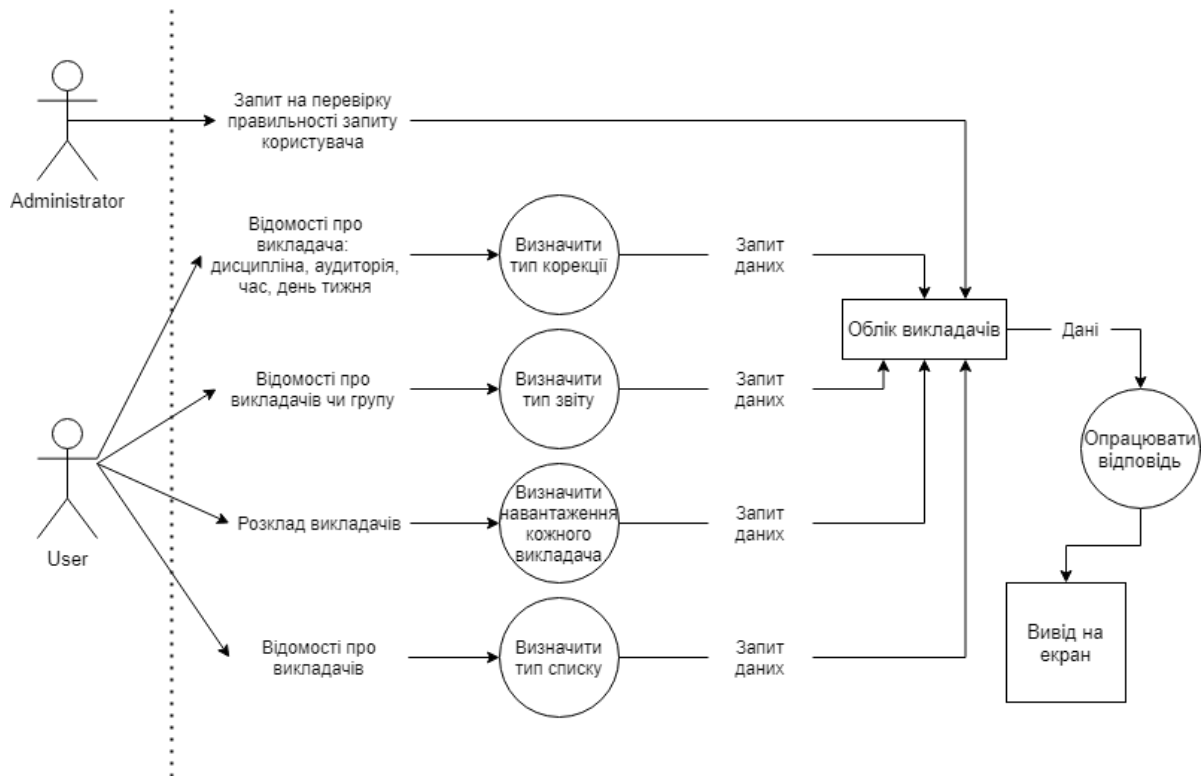
- Зміна розкладу для викладача (аудиторія, день тижня, час).

4. Звіти виду:

- Розклад занять для однієї довільної групи.
- "Якісний склад співробітників кафедр": кафедра - кількість професорів - кількість доцентів - кількість старших викладачів;
- Розклад для викладачів кафедри «Х».
- "Навантаження викладачів": кафедра - викладач - кількість дисципліни, які він веде і кількість годин з цих дисциплін

1. Діаграми потоків даних (DFD)





2. Код sql

```
DROP TABLE Subjects, Department, Position, Degree, Week_day, Teacher, Teacher_subject, Schedule
```

```
CREATE TABLE Subjects
```

```
(
  subject_id INT IDENTITY PRIMARY KEY,
  subject_name VARCHAR(40) NOT NULL,
);
```

```
CREATE TABLE Department
```

```
(
  department_id INT IDENTITY PRIMARY KEY,
  department_name VARCHAR(40) NOT NULL,
);
```

```
CREATE TABLE Position
```

```
(
  position_id INT IDENTITY PRIMARY KEY,
  position_name VARCHAR(40) UNIQUE NOT NULL
);
```

```
CREATE TABLE Degree
```

```
(
  degree_id INT IDENTITY PRIMARY KEY,
  degree_name VARCHAR(40) UNIQUE NOT NULL
);
```

```
CREATE TABLE Week_day
```

```
(
  day_id INT IDENTITY PRIMARY KEY,
  day_name VARCHAR(40) UNIQUE NOT NULL
);
```

```
CREATE TABLE Teacher
```

```
(
  teacher_id INT IDENTITY PRIMARY KEY,
  teacher_name VARCHAR(40) NOT NULL,
  teacher_surname VARCHAR(20) NOT NULL,
  teacher_gender VARCHAR(1) NOT NULL,
);
```

```

CONSTRAINT emp_gender CHECK(teacher_gender LIKE 'M' or teacher_gender LIKE 'F'),
teacher_position INT NOT NULL,
teacher_degree INT NOT NULL,
teacher_department INT NOT NULL,
CONSTRAINT t_position FOREIGN KEY(teacher_position) REFERENCES Position(position_id),
CONSTRAINT t_department FOREIGN KEY(teacher_department) REFERENCES
Department(department_id),
CONSTRAINT t_degree FOREIGN KEY(teacher_degree) REFERENCES Degree(degree_id)
);

CREATE TABLE Teacher_subject
(
    teacher_id INT FOREIGN KEY REFERENCES Teacher(teacher_id),
    subject_id INT FOREIGN KEY REFERENCES Subjects(subject_id)
);

CREATE TABLE Schedule
(
    teacher_id INT FOREIGN KEY REFERENCES Teacher(teacher_id),
    subject_id INT FOREIGN KEY REFERENCES Subjects(subject_id),
    day_id INT FOREIGN KEY REFERENCES Week_day(day_id),
    room INT NOT NULL,
    lesson_time TIME NOT NULL,
    group_name VARCHAR(20) NOT NULL
);

INSERT INTO Week_day(day_name) VALUES ('Monday')
INSERT INTO Week_day(day_name) VALUES ('Tuesday')
INSERT INTO Week_day(day_name) VALUES ('Wednesday')
INSERT INTO Week_day(day_name) VALUES ('Thursday')
INSERT INTO Week_day(day_name) VALUES ('Friday')
INSERT INTO Week_day(day_name) VALUES ('Saturday')

INSERT INTO Subjects(subject_name) VALUES ('Discrete mathematics')
INSERT INTO Subjects(subject_name) VALUES ('Math analysis')
INSERT INTO Subjects(subject_name) VALUES ('Physics')
INSERT INTO Subjects(subject_name) VALUES ('Theory of probability')
INSERT INTO Subjects(subject_name) VALUES ('Algorithms and structures')
INSERT INTO Subjects(subject_name) VALUES ('Symmetric cryptography')
INSERT INTO Subjects(subject_name) VALUES ('Mathematical physics')
INSERT INTO Subjects(subject_name) VALUES ('Differential equations')

INSERT INTO Department(department_name) VALUES('IS');
INSERT INTO Department(department_name) VALUES('MMIS');
INSERT INTO Department(department_name) VALUES('Physics');

INSERT INTO Position(position_name) VALUES('Assistant');
INSERT INTO Position(position_name) VALUES('Docent');
INSERT INTO Position(position_name) VALUES('Professor');

INSERT INTO Degree(degree_name) VALUES('PhD student');
INSERT INTO Degree(degree_name) VALUES('D.Sc. ');
INSERT INTO Degree(degree_name) VALUES('PhD');

INSERT INTO Teacher(teacher_name, teacher_surname, teacher_gender, teacher_position,
teacher_degree, teacher_department) VALUES
('Sergey', 'Yakovlev', 'M', 2, 2, 2);
INSERT INTO Teacher(teacher_name, teacher_surname, teacher_gender, teacher_position,
teacher_degree, teacher_department) VALUES
('Anna', 'Potapova', 'F', 2, 2, 1);
INSERT INTO Teacher(teacher_name, teacher_surname, teacher_gender, teacher_position,
teacher_degree, teacher_department) VALUES
('Stanislav', 'Beh', 'M', 1, 2, 3);
INSERT INTO Teacher(teacher_name, teacher_surname, teacher_gender, teacher_position,
teacher_degree, teacher_department) VALUES
('Andrey', 'Dorogovtsev', 'M', 3, 3, 2);
INSERT INTO Teacher(teacher_name, teacher_surname, teacher_gender, teacher_position,
teacher_degree, teacher_department) VALUES

```

```

('Alla', 'Klymenko', 'F', 1, 1, 1);
INSERT INTO Teacher(teacher_name, teacher_surname, teacher_gender, teacher_position,
teacher_degree, teacher_department) VALUES
('Andrey', 'Melnichenko', 'M', 3, 3, 2);
INSERT INTO Teacher(teacher_name, teacher_surname, teacher_gender, teacher_position,
teacher_degree, teacher_department) VALUES
('Maxym', 'Stepanov', 'M', 1, 1, 3);
INSERT INTO Teacher(teacher_name, teacher_surname, teacher_gender, teacher_position,
teacher_degree, teacher_department) VALUES
('George', 'Ryabov', 'M', 2, 2, 1);
INSERT INTO Teacher(teacher_name, teacher_surname, teacher_gender, teacher_position,
teacher_degree, teacher_department) VALUES
('Nadiya', 'Pavlova', 'F', 1, 1, 1);
INSERT INTO Teacher(teacher_name, teacher_surname, teacher_gender, teacher_position,
teacher_degree, teacher_department) VALUES
('Alex', 'Motor', 'M', 2, 2, 3);

INSERT INTO Teacher_subject(teacher_id, subject_id) VALUES (1, 1);
INSERT INTO Teacher_subject(teacher_id, subject_id) VALUES (1, 6);
INSERT INTO Teacher_subject(teacher_id, subject_id) VALUES (2, 2);
INSERT INTO Teacher_subject(teacher_id, subject_id) VALUES (3, 3);
INSERT INTO Teacher_subject(teacher_id, subject_id) VALUES (4, 4);
INSERT INTO Teacher_subject(teacher_id, subject_id) VALUES (5, 5);
INSERT INTO Teacher_subject(teacher_id, subject_id) VALUES (6, 6);
INSERT INTO Teacher_subject(teacher_id, subject_id) VALUES (7, 7);
INSERT INTO Teacher_subject(teacher_id, subject_id) VALUES (8, 8);
INSERT INTO Teacher_subject(teacher_id, subject_id) VALUES (8, 4);
INSERT INTO Teacher_subject(teacher_id, subject_id) VALUES (9, 2);
INSERT INTO Teacher_subject(teacher_id, subject_id) VALUES (10, 7);
INSERT INTO Teacher_subject(teacher_id, subject_id) VALUES (10, 3);

INSERT INTO Schedule(teacher_id, subject_id, day_id, room, lesson_time, group_name) VALUES
(1, 1, 1, 211, '14:15:00', 'FI-01');
INSERT INTO Schedule(teacher_id, subject_id, day_id, room, lesson_time, group_name) VALUES
(1, 1, 1, 211, '08:30', 'FI-02');
INSERT INTO Schedule(teacher_id, subject_id, day_id, room, lesson_time, group_name) VALUES
(1, 6, 1, 203, '10:25', 'FI-73');
INSERT INTO Schedule(teacher_id, subject_id, day_id, room, lesson_time, group_name) VALUES
(1, 6, 3, 203, '10:25', 'FI-73');
INSERT INTO Schedule(teacher_id, subject_id, day_id, room, lesson_time, group_name) VALUES
(1, 6, 3, 211, '14:15', 'FI-74');
INSERT INTO Schedule(teacher_id, subject_id, day_id, room, lesson_time, group_name) VALUES
(2, 1, 203, '8:30', 'FI-91');
INSERT INTO Schedule(teacher_id, subject_id, day_id, room, lesson_time, group_name) VALUES
(2, 2, 1, 211, '10:25', 'FI-92');
INSERT INTO Schedule(teacher_id, subject_id, day_id, room, lesson_time, group_name) VALUES
(2, 2, 4, 211, '8:30', 'FI-93');
INSERT INTO Schedule(teacher_id, subject_id, day_id, room, lesson_time, group_name) VALUES
(2, 2, 4, 211, '10:25', 'FI-94');
INSERT INTO Schedule(teacher_id, subject_id, day_id, room, lesson_time, group_name) VALUES
(2, 2, 5, 215, '10:25', 'FI-91');
INSERT INTO Schedule(teacher_id, subject_id, day_id, room, lesson_time, group_name) VALUES
(2, 2, 5, 215, '10:25', 'FI-92');
INSERT INTO Schedule(teacher_id, subject_id, day_id, room, lesson_time, group_name) VALUES
(2, 2, 5, 215, '10:25', 'FI-93');
INSERT INTO Schedule(teacher_id, subject_id, day_id, room, lesson_time, group_name) VALUES
(2, 2, 5, 215, '10:25', 'FI-94');
INSERT INTO Schedule(teacher_id, subject_id, day_id, room, lesson_time, group_name) VALUES
(3, 3, 2, 201, '8:30', 'FF-91');
INSERT INTO Schedule(teacher_id, subject_id, day_id, room, lesson_time, group_name) VALUES
(3, 3, 2, 201, '10:25', 'FF-92');
INSERT INTO Schedule(teacher_id, subject_id, day_id, room, lesson_time, group_name) VALUES
(4, 4, 3, 215, '14:15', 'FI-81');
INSERT INTO Schedule(teacher_id, subject_id, day_id, room, lesson_time, group_name) VALUES
(4, 4, 3, 215, '14:15', 'FI-82');
INSERT INTO Schedule(teacher_id, subject_id, day_id, room, lesson_time, group_name) VALUES
(4, 4, 3, 215, '14:15', 'FI-83');

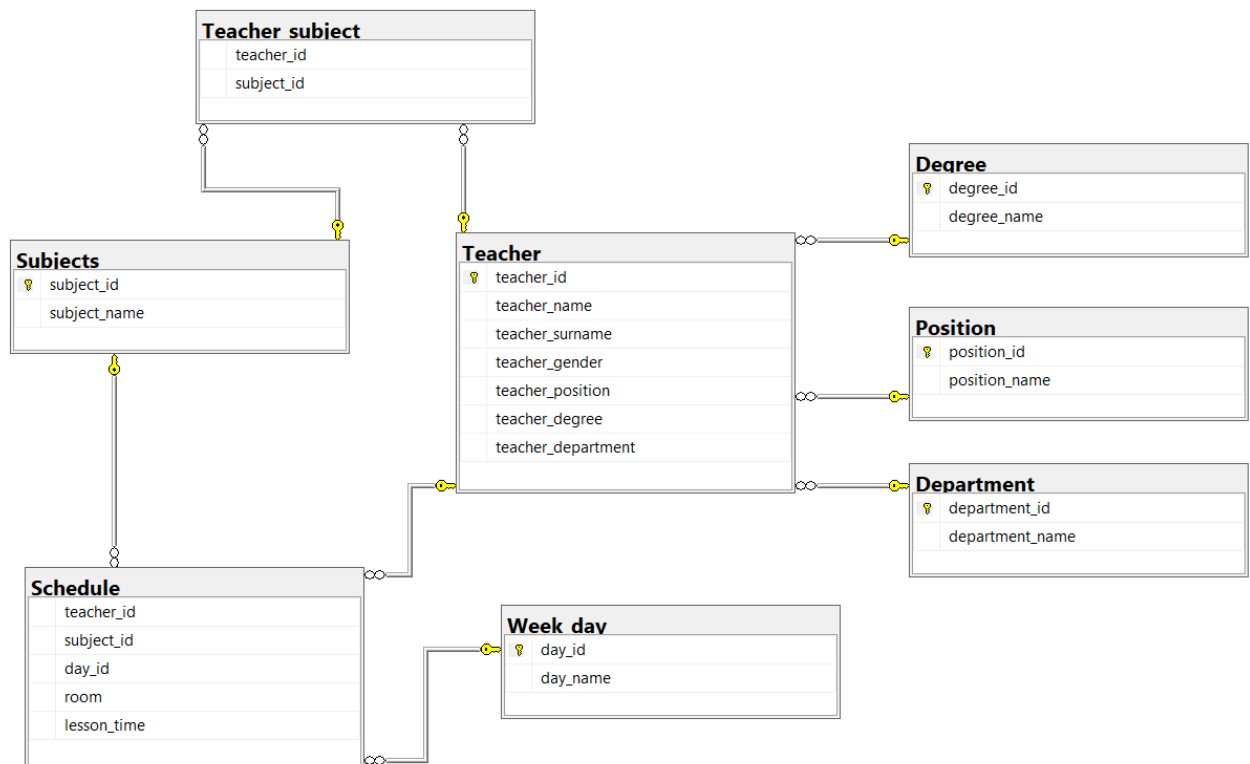
```

```

INSERT INTO Schedule(teacher_id, subject_id, day_id, room, lesson_time, group_name) VALUES
(4, 4, 3, 215, '14:15', 'FI-84');
INSERT INTO Schedule(teacher_id, subject_id, day_id, room, lesson_time, group_name) VALUES
(5, 5, 2, 303, '14:15', 'FF-91');
INSERT INTO Schedule(teacher_id, subject_id, day_id, room, lesson_time, group_name) VALUES
(5, 5, 2, 303, '10:25', 'FE-92');
INSERT INTO Schedule(teacher_id, subject_id, day_id, room, lesson_time, group_name) VALUES
(5, 5, 5, 303, '8:30', 'FF-92');
INSERT INTO Schedule(teacher_id, subject_id, day_id, room, lesson_time, group_name) VALUES
(5, 5, 5, 303, '10:25', 'FE-91');
INSERT INTO Schedule(teacher_id, subject_id, day_id, room, lesson_time, group_name) VALUES
(6, 6, 2, 211, '12:00', 'FI-73');
INSERT INTO Schedule(teacher_id, subject_id, day_id, room, lesson_time, group_name) VALUES
(6, 6, 2, 211, '12:00', 'FI-74');
INSERT INTO Schedule(teacher_id, subject_id, day_id, room, lesson_time, group_name) VALUES
(7, 7, 1, 204, '12:00', 'FF-91');
INSERT INTO Schedule(teacher_id, subject_id, day_id, room, lesson_time, group_name) VALUES
(7, 7, 6, 204, '12:00', 'FF-92');
INSERT INTO Schedule(teacher_id, subject_id, day_id, room, lesson_time, group_name) VALUES
(8, 8, 5, 201, '8:30', 'FF-83');
INSERT INTO Schedule(teacher_id, subject_id, day_id, room, lesson_time, group_name) VALUES
(8, 4, 4, 201, '8:30', 'FI-81');
INSERT INTO Schedule(teacher_id, subject_id, day_id, room, lesson_time, group_name) VALUES
(9, 2, 6, 305, '14:15', 'FI-01');
INSERT INTO Schedule(teacher_id, subject_id, day_id, room, lesson_time, group_name) VALUES
(10, 3, 2, 204, '8:30', 'FF-02');
INSERT INTO Schedule(teacher_id, subject_id, day_id, room, lesson_time, group_name) VALUES
(10, 7, 2, 204, '12:00', 'FF-92');

```

- ER-діаграма:



- Створення впорядкованих списків

```
DROP VIEW IF EXISTS first_list, second_list
GO
```

1. Кафедра - викладач – дисципліна

```
CREATE VIEW first_list
AS
    SELECT t.teacher_name, t.teacher_surname, d.department_name, s.subject_name
    FROM ((teacher t INNER JOIN department d ON t.teacher_department = d.department_id)
          INNER JOIN teacher_subject st ON t.teacher_id = st.teacher_id)
          INNER JOIN subjects s ON st.subject_id = s.subject_id;

GO

--SELECT *
--FROM first_list
```

2. Викладачів, у яких посада не відповідає наукового ступеня (для кандидата - доцент, для доктора - професор).

```
CREATE VIEW second_list
AS
    SELECT t.teacher_name, t.teacher_surname, p.position_name, d.degree_name
    FROM (teacher t INNER JOIN position p ON t.teacher_position = p.position_id)
          INNER JOIN degree d ON t.teacher_degree = d.degree_id
    WHERE (p.position_name = 'Assistant' AND (d.degree_name = 'D.Sc.' OR d.degree_name = 'PhD')) OR
           (p.position_name = 'Docent' AND (d.degree_name = 'PhD' OR d.degree_name = 'PhD student')) OR
           (p.position_name = 'Professor' AND (d.degree_name = 'D.Sc.' OR d.degree_name = 'PhD student'))

GO

--SELECT *
--FROM second_list
```

- Обчислення:

1. Переконалися, що у кожного викладача не більше 3-х занять на день.

Для того, щоб виконати дане обчислення, створимо додатковий view для правильного виведення інформації. Даний view буде містити інформацію про викладачів, кафедр, до яких вони належать, про предмети, які вони ведуть та в які дні.

```
DROP VIEW IF EXISTS help_view
GO
```

```
CREATE VIEW help_view
AS
    SELECT t.teacher_name, t.teacher_surname, d.department_name, s.subject_name,
    sc.day_id
    FROM (((teacher t INNER JOIN department d ON t.teacher_department =
    d.department_id)
           INNER JOIN teacher_subject st ON t.teacher_id = st.teacher_id)
          INNER JOIN subjects s ON st.subject_id = s.subject_id)
          INNER JOIN schedule sc ON (s.subject_id = sc.subject_id and
    st.teacher_id = sc.teacher_id)
    GROUP BY t.teacher_name, t.teacher_surname, d.department_name, sc.day_id,
    sc.lesson_time, s.subject_name

GO
```


Тепер створимо процедуру безпосередньо для виконання обчислення

```
DROP PROCEDURE IF EXISTS check_num_lessons
GO

CREATE PROCEDURE check_num_lessons
(
    @teacher_name varchar(40),
    @teacher_surname varchar(40)
)
AS
BEGIN
    IF EXISTS
    (
        SELECT hv.teacher_name, hv.teacher_surname, w.day_name, COUNT(hv.subject_name)
        FROM help_view hv INNER JOIN week_day w ON hv.day_id = w.day_id
        WHERE hv.teacher_name = @teacher_name AND hv.teacher_surname =
        @teacher_surname
        GROUP BY hv.teacher_name, hv.teacher_surname, w.day_name
        HAVING COUNT(hv.subject_name) > 3
    )
        PRINT ('Teacher ' + @teacher_name + ' ' + @teacher_surname + ' ' + 'has more
        than 3 lessons per day.')
    ELSE
        PRINT('This teacher has 3 or less lessons per day')
    END
GO

--EXEC check_num_lessons 'Sergey', 'Yakovlev'
```

- Корекція:

1. Зміна розкладу для викладача (аудиторія, день тижня, час).

```
DROP PROCEDURE IF EXISTS correction
GO

CREATE PROCEDURE correction
(
    @teacher_id INT,
    @subject_id INT,
    @day_id INT,
    @room INT,
    @lesson_time TIME,
    @group_name VARCHAR(20)
)
AS
BEGIN
    UPDATE schedule
    SET room = @room, day_id = @day_id, lesson_time = @lesson_time
    WHERE teacher_id = @teacher_id AND subject_id = @subject_id
    AND group_name = @group_name
    END
GO

--EXEC correction 10, 3, 2, 315, '8:30', 'FF-02'
```

- Створення звітів

1. Розклад занять для однієї довільної групи.

```
DROP PROCEDURE IF EXISTS group_schedule
GO

CREATE PROCEDURE group_schedule
(
    @group_name VARCHAR(20)
)
AS
BEGIN
    SELECT t.teacher_name, t.teacher_surname, w.day_name, sc.room, sc.lesson_time,
    sc.group_name, s.subject_name
    FROM (((teacher t INNER JOIN teacher_subject st ON t.teacher_id =
    st.teacher_id)
            INNER JOIN subjects s ON st.subject_id = s.subject_id)
            INNER JOIN schedule sc ON (s.subject_id = sc.subject_id
and st.teacher_id = sc.teacher_id))
            INNER JOIN week_day w ON sc.day_id = w.day_id
    WHERE sc.group_name = @group_name
    ORDER BY sc.day_id
END
GO

--EXEC group_schedule 'FI-81'
```

2. "Якісний склад співробітників кафедр": кафедра - кількість професорів - кількість доцентів - кількість старших викладачів;

```
DROP VIEW IF EXISTS staff_of_departments
GO

CREATE VIEW staff_of_departments
AS
    SELECT d.department_name, p.position_name, COUNT(p.position_name) AS amount
    FROM (teacher t inner join department d ON t.teacher_department = d.department_id)
            inner join position p ON t.teacher_position = p.position_id
    GROUP BY d.department_name, p.position_name
GO

--SELECT *
--FROM staff_of_departments
--ORDER BY department_name
```

3. Розклад для викладачів кафедри «Х».

```
DROP PROCEDURE IF EXISTS department_schedule
GO

CREATE PROCEDURE department_schedule
(
    @department_name VARCHAR(20)
)
AS
BEGIN
    SELECT t.teacher_name, t.teacher_surname, w.day_name, sc.room, sc.lesson_time,
    sc.group_name, s.subject_name, d.department_name
    FROM (((teacher t INNER JOIN teacher_subject st ON t.teacher_id =
    st.teacher_id)
            INNER JOIN subjects s ON st.subject_id = s.subject_id)
            INNER JOIN schedule sc ON (s.subject_id = sc.subject_id
and st.teacher_id = sc.teacher_id))
```

```

INNER JOIN week_day w ON sc.day_id = w.day_id)
INNER JOIN department d ON t.teacher_department =
d.department_id
WHERE d.department_name = @department_name
ORDER BY sc.day_id
END
GO

--EXEC department_schedule 'IS'

```

4. "Навантаження викладачів": кафедра - викладач - кількість дисципліни, які він веде і кількість годин з цих дисциплін

```

DROP VIEW IF EXISTS loading_of_teachers
GO

CREATE VIEW loading_of_teachers
AS
SELECT teacher_name, teacher_surname, department_name, COUNT(teacher_name) as
number_of_hours
FROM help_view
GROUP BY teacher_name, teacher_surname, department_name
GO

--SELECT * FROM loading_of_teachers

```

3. Перевірка нормалізації розробленої моделі

Спочатку потрібно перевірити чи належить дана модель до першої нормальної форми. Так, належить, бо не має повторюваних записів і повторюваних груп полів.

Далі перевіримо чи належить до другої нормальної форми. Для цього модель повинна бути в першій нормальній формі, а також не повинно бути неповних функціональних залежностей не ключових атрибутів первинного ключа тобто будь-яке не ключове поле повинно однозначно ідентифікуватися повним набором ключових полів. Для даної моделі виконується, тобто розроблена модель належить до другої нормальної форми.

Залишилось перевірити належність до третьої нормальної форми. Для цього відношення повинно бути в другій нормальній формі та не повинно бути транзитивних залежностей тобто ні одне з ключових полів не повинно ідентифікуватися за допомогою іншого неключового поля. Видно, що у розробленій моделі відсутні такі залежності. Отже, модель належить до третьої нормальної форми.

