

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ



НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені Ігоря Сікорського»

ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ

Кафедра інформаційної безпеки

Розрахунково-графічна робота

Дисципліна: «Бази даних та ІКС»

Напрямок підготовки 6.040301 «Прикладна математика»

Виконав студент 4 курсу групи ФІ-72

Геніцой Павло Олексійович

Перевірів:

Коломицев Михайло Володимирович

(підпис)

Оцінка

Варіант 2

Предметна область - Деканат.

Задачі, що вирішує інформаційна система - складання розкладу на екзаменаційні сесії.

Завдання на обробку даних:

1. Вивести розклад іспитів:

- Для однієї довільної кафедри;
- Для однієї довільної групи.

2. Обчислення:

- Перевірити, чи проходить у кожної групи між двома іспитами не менше трьох днів.
- Переконатися, що у кожного викладача в день не більше одного іспиту.
- Порахувати кількість іспитів у кожної групи і по курсу в цілому.

3. корекція:

- Зміна дати і аудиторії, в якій проводить іспит зазначений викладач або зазначена група.

4. Звіти виду:

- Розклад іспитів на кафедрі, впорядковане за категоріями викладачів (старший викладач, доцент, професор).
- Інформація виду: "Викладач - дисципліна - консультація (дата, час, аудиторія) - іспит (дата, час, аудиторія)".

Код виконання

Use DataBaseRGR

DROP TABLE IF EXISTS DEANERY

DROP TABLE IF EXISTS LECTURERS

DROP TABLE IF EXISTS GROUPS

DROP TABLE IF EXISTS DISCIPLINES

DROP TABLE IF EXISTS AUDITORIES

DROP TABLE IF EXISTS DEGREE_ALL

DROP TABLE IF EXISTS DEPARTMENTS

DROP TABLE IF EXISTS FORM

DROP VIEW IF EXISTS Main_Deanery

DROP PROC IF EXISTS Schedule_for_Department

DROP PROC IF EXISTS Schedule_for_Group

DROP PROC IF EXISTS Checking_Days_between_2_Exams

DROP PROC IF EXISTS Checking_1_Exam_per_1_Day_for_1_Lecturer

DROP PROC IF EXISTS Count_of_Exams_for_Group_and_Course

DROP PROC IF EXISTS Modification_Dates_and_Auditory

DROP PROC IF EXISTS Report_to_1

DROP PROC IF EXISTS Report_to_2

CREATE TABLE FORM (-- это экз или консультация

 IDForm INT PRIMARY KEY IDENTITY,

 Name_of_Form NVARCHAR(20) NOT NULL UNIQUE

)

```
CREATE TABLE DEPARTMENTS (    -- кафедра где учишься
    IDDepartment INT PRIMARY KEY IDENTITY,
    Name_of_Department NVARCHAR(50) NOT NULL UNIQUE
)
```

```
CREATE TABLE DEGREE_ALL (    -- должность препода (старший викладач,
    доцент, професор)
    IDDegree INT PRIMARY KEY IDENTITY,
    Name_of_Degree NVARCHAR(50) NOT NULL UNIQUE
)
```

```
CREATE TABLE AUDITORIES (
    IDAuditory INT PRIMARY KEY IDENTITY,
    Name_of_Auditory NVARCHAR(10) NOT NULL UNIQUE
)
```

```
CREATE TABLE DISCIPLINES (
    IDDiscipline INT PRIMARY KEY IDENTITY,
    Name_of_Discipline NVARCHAR(50) NOT NULL UNIQUE
)
```

```
CREATE TABLE GROUPS (
    IDGroup INT PRIMARY KEY IDENTITY,
    Name_of_Group NVARCHAR(10) NOT NULL UNIQUE,
    Year_of_Admission INT NOT NULL,
    DepartmentID INT NOT NULL,
    CONSTRAINT DepartmentID_ FOREIGN KEY (DepartmentID) REFERENCES
    DEPARTMENTS(IDDepartment) ON DELETE CASCADE
)
```

```

CREATE TABLE LECTURERS (    -- препод
    IDLecturer INT PRIMARY KEY IDENTITY,
    Surname NVARCHAR(50)NOT NULL,
    FirstName NVARCHAR(50)NOT NULL,
    Patronymic NVARCHAR(50),
    DegreeID INT NOT NULL,
    CONSTRAINT DegreeID_ FOREIGN KEY (DegreeID) REFERENCES
DEGREE_ALL(IDDegree) ON DELETE CASCADE
)

```

```

CREATE TABLE DEANERY (
    IDDeanery INT PRIMARY KEY IDENTITY,
    Dates DATE,
    GroupID INT NOT NULL,
    FormID INT NOT NULL,
    LecturerID INT NOT NULL,
    AuditoryID INT NOT NULL,
    DisciplineID INT NOT NULL,
    CONSTRAINT GroupID_ FOREIGN KEY (GroupID) REFERENCES
GROUPS(IDGroup) ON DELETE CASCADE,
    CONSTRAINT FormID_ FOREIGN KEY (FormID) REFERENCES FORM(IDForm) ON
DELETE CASCADE,
    CONSTRAINT LecturerID_ FOREIGN KEY (LecturerID) REFERENCES
LECTURERS(IDLecturer) ON DELETE CASCADE,
    CONSTRAINT AuditoryID_ FOREIGN KEY (AuditoryID) REFERENCES
AUDITORIES(IDAuditory) ON DELETE CASCADE,
    CONSTRAINT DisciplineID_ FOREIGN KEY (DisciplineID) REFERENCES
DISCIPLINES(IDDiscipline) ON DELETE CASCADE,
    CONSTRAINT UniqueSession UNIQUE (Dates, GroupID, FormID,
LecturerID, AuditoryID, DisciplineID)
)

```

```
DBCC CHECKIDENT('FORM',RESEED,1)
```

```
GO
```

```
DBCC CHECKIDENT('DEPARTMENTS',RESEED,1)
```

```
GO
```

```
DBCC CHECKIDENT('DEGREE_ALL',RESEED,1)
```

```
GO
```

```
DBCC CHECKIDENT('AUDITORIES',RESEED,1)
```

```
GO
```

```
DBCC CHECKIDENT('DISCIPLINES',RESEED,1)
```

```
GO
```

```
DBCC CHECKIDENT('GROUPS',RESEED,1)
```

```
GO
```

```
DBCC CHECKIDENT('LECTURERS',RESEED,1)
```

```
GO
```

```
DBCC CHECKIDENT('DEANERY',RESEED,1)
```

```
GO
```

```
INSERT INTO FORM(Name_of_Form) VALUES ('Консультация'), ('Экзамен')
```

```
INSERT INTO DEPARTMENTS(Name_of_Department)
```

```
VALUES ('Прикладная физика'),
```

```
        ('Физика энергетических систем'),
```

```
        ('Информационная безопасность'),
```

```
        ('Математические методы защиты информации'),
```

```
        ('Физико-технические средства защиты информации')
```

```
INSERT INTO DEGREE_ALL(Name_of_Degree) VALUES ('Старший преподаватель'),  
('Доцент'), ('Профессор')
```

```
INSERT INTO AUDITORIES(Name_of_Auditory)  
VALUES ('112-7'), ('114-7'), ('116-7'), ('107-7'), ('БФ-1'), ('308-  
4-1'), ('215-11')
```

```
INSERT INTO DISCIPLINES(Name_of_Discipline)  
VALUES ('Математический анализ'),  
('Физика'),  
('Случайные процессы'),  
('Мат. методы принятия решения'),  
('Классическая механика'),  
('Дифф. уравнения'),  
('Мат. статистика'),  
('Алгебра и геометрия'),  
('Программирование')
```

```
INSERT INTO GROUPS(Name_of_Group, Year_of_Admission, DepartmentID)  
VALUES ('ФИ-71', 2017, (SELECT IDDepartment FROM DEPARTMENTS WHERE  
IDDepartment=3)),  
('ФИ-72', 2017, (SELECT IDDepartment FROM DEPARTMENTS WHERE  
IDDepartment=3)),  
('ФИ-73', 2017, (SELECT IDDepartment FROM DEPARTMENTS WHERE  
IDDepartment=4)),  
('ФИ-74', 2017, (SELECT IDDepartment FROM DEPARTMENTS WHERE  
IDDepartment=4)),  
('ФИ-61', 2016, (SELECT IDDepartment FROM DEPARTMENTS WHERE  
IDDepartment=3)),  
('ФИ-62', 2016, (SELECT IDDepartment FROM DEPARTMENTS WHERE  
IDDepartment=4)),  
('ФИ-82', 2018, (SELECT IDDepartment FROM DEPARTMENTS WHERE  
IDDepartment=4)),
```

```
        ('ФФ-81', 2018, (SELECT IDDepartment FROM DEPARTMENTS WHERE  
IDDepartment=1)),
```

```
        ('ФФ-82', 2018, (SELECT IDDepartment FROM DEPARTMENTS WHERE  
IDDepartment=2)),
```

```
        ('ФФ-73', 2017, (SELECT IDDepartment FROM DEPARTMENTS WHERE  
IDDepartment=5))
```

```
INSERT INTO LECTURERS(Surname, FirstName, Patronymic, DegreeID)
```

```
VALUES ('Дороговцев', 'Андрей', 'Анатольевич', (SELECT IDDegree  
FROM DEGREE_ALL WHERE IDDegree=3)),
```

```
        ('Кравцов', 'Олег', 'Васильевич', (SELECT IDDegree FROM  
DEGREE_ALL WHERE IDDegree=2)),
```

```
        ('Южакова', 'Анна', 'Алексеевна', (SELECT IDDegree FROM  
DEGREE_ALL WHERE IDDegree=2)),
```

```
        ('Куссуль', 'Наталья', 'Николаевна', (SELECT IDDegree FROM  
DEGREE_ALL WHERE IDDegree=3)),
```

```
        ('Смирнов', 'Сергей', 'Анатольевич', (SELECT IDDegree FROM  
DEGREE_ALL WHERE IDDegree=2)),
```

```
        ('Яковлев', 'Сергей', 'Владимирович', (SELECT IDDegree FROM  
DEGREE_ALL WHERE IDDegree=2)),
```

```
        ('Стасюк', 'Александр', 'Сергеевич', (SELECT IDDegree FROM  
DEGREE_ALL WHERE IDDegree=1))
```

```
INSERT INTO DEANERY(Dates, GroupID, FormID, LecturerID, AuditoriumID,  
DisciplineID)
```

```
VALUES ('2020-01-08', 2, 1, 1, 1, 1),
```

```
        ('2020-01-09', 2, 2, 1, 1, 1),
```

```
        ('2020-01-11', 2, 1, 2, 3, 3),
```

```
        ('2020-01-12', 2, 2, 2, 2, 3),
```

```
        ('2020-01-10', 3, 1, 4, 1, 5),
```

```
        ('2020-01-11', 3, 2, 4, 2, 5),
```

```
        ('2020-01-13', 1, 1, 7, 6, 9),
```

```
        ('2020-01-14', 1, 2, 7, 6, 9),
```



```

('2020-01-14', 5, 1, 3, 5, 8),
('2020-01-15', 5, 2, 3, 4, 8),
('2020-01-15', 7, 1, 5, 2, 6),
('2020-01-16', 7, 2, 5, 1, 6),
('2020-01-18', 9, 1, 6, 4, 7),
('2020-01-19', 9, 2, 6, 4, 7),
('2020-01-19', 10, 1, 3, 7, 2),
('2020-01-20', 10, 2, 3, 7, 2),
('2020-01-19', 2, 1, 5, 1, 9),
('2020-01-20', 2, 2, 5, 2, 9)

```

GO

CREATE VIEW Main_Deanery

AS SELECT *

FROM DEANERY dn INNER JOIN (GROUPS g INNER JOIN DEPARTMENTS dp ON
dp.IDDepartment=g.DepartmentID) ON g.IDGroup=dn.GroupID

INNER JOIN FORM f ON f.IDForm=dn.FormID INNER JOIN (LECTURERS l
INNER JOIN DEGREE_ALL dg ON dg.IDDegree=l.DegreeID) ON
l.IDLecturer=dn.LecturerID

INNER JOIN AUDITORIES a ON a.IDAuditory=dn.AuditoryID INNER JOIN
DISCIPLINES ds ON ds.IDDiscipline=dn.DisciplineID

GO -- Розклад іспитів для однієї довільної кафедри

CREATE PROC Schedule_for_Department

@Name_of_Department NVARCHAR(50)

AS

SELECT Name_of_Department, Name_of_Group, Name_of_Discipline,
Dates, Surname, FirstName, Patronymic, Name_of_Auditory, Name_of_Form
FROM Main_Deanery

```
WHERE Name_of_Form='экзамен' AND  
Name_of_Department=@Name_of_Department
```

```
GO
```

```
--Schedule_for_Department 'Математические методы защиты информации'
```

```
GO -- Розклад іспитива для однієї довільної групи
```

```
CREATE PROC Schedule_for_Group
```

```
@Name_of_Group NVARCHAR(10)
```

```
AS
```

```
SELECT Name_of_Group, Name_of_Discipline, Dates, Surname,  
FirstName, Patronymic, Name_of_Auditory, Name_of_Form FROM Main_Deanery
```

```
WHERE Name_of_Form='экзамен' AND Name_of_Group=@Name_of_Group
```

```
GO
```

```
--Schedule_for_Group 'ФИ-72'
```

```
--GO
```

```
--Schedule_for_Group 'ФИ-73'
```

```
--GO
```

```
--Schedule_for_Group 'ФИ-61'
```

```
--GO
```

```
--Schedule_for_Group 'ФИ-71'
```

```
--GO
```

```
--Schedule_for_Group 'ФИ-82'
```

```
--GO
```

```
--Schedule_for_Group 'ФФ-73'
```

```
--GO
```

```
--Schedule_for_Group 'ФФ-82'
```

GO -- Перевірити, чи проходить у кожної групи між двома іспитами не менше трьох днів

CREATE PROC Checking_Days_between_2_Exams

 @Name_of_Group NVARCHAR(10)

AS

 IF EXISTS (SELECT * FROM (SELECT DATEDIFF(DAY, LastDates, Dates) x
FROM (

 SELECT Dates, LAG(Dates, 1) OVER (ORDER BY Dates) LastDates FROM
Main_Deanery

 WHERE Name_of_Group=@Name_of_Group AND Name_of_Form='экзамен') x) x
WHERE x>3)

 PRINT('Для группы "' + @Name_of_Group + '" проходит не меньше
трёх дней между двумя экзаменами.')

 ELSE

 PRINT('Для группы "' + @Name_of_Group + '" проходит меньше трёх
дней между двумя экзаменами.')

GO

--Checking_Days_between_2_Exams 'ФИ-72'

GO -- Переконатися, що у кожного викладача в день не більше одного іспиту

CREATE PROC Checking_1_Exam_per_1_Day_for_1_Lecturer

 @Surname NVARCHAR(50),

 @FirstName NVARCHAR(50),

 @Patronymic NVARCHAR(50)

AS

 IF EXISTS (SELECT * FROM (SELECT COUNT(*) AS Counts FROM
Main_Deanery

 WHERE Surname=@Surname AND FirstName=@FirstName AND
Patronymic=@Patronymic AND Name_of_Form='экзамен' GROUP BY Surname) x
WHERE Counts>1)

 PRINT('Для преподавателя "' + @Surname + ' ' + @FirstName + '
' + @Patronymic + ' ' + '" больше одного экзамена в день.')

```

ELSE

    PRINT('Для преподавателя "' + @Surname + ' ' + @FirstName + '
' + @Patronymic + '" не больше одного экзамена в день.')

GO

--Checking_1_Exam_per_1_Day_for_1_Lecturer 'Дороговцев', 'Андрей',
'Анатолевич'


GO -- Порахувати кількість іспитів у кожній групі і по курсу в цілому
CREATE PROC Count_of_Exams_for_Group_and_Course
AS

    SELECT Name_of_Group, COUNT(*) AS Count_Exams_of_Group FROM
Main_Deanery

    WHERE Name_of_Form='экзамен' GROUP BY Name_of_Group


    SELECT COUNT(*) AS Count_Exams_Generally FROM Main_Deanery
    WHERE Name_of_Form='экзамен'

GO

--Count_of_Exams_for_Group_and_Course


GO -- Зміна дати і аудиторії, в якій проводить іспит зазначений викладач
або зазначена група.
CREATE PROC Modification_Dates_and_Auditory
    @LecturerSurname NVARCHAR(50),
    @LecturerFirstname NVARCHAR(50),
    @LecturerPatronymic NVARCHAR(50),
    @GroupName NVARCHAR(10),
    @LastDate DATE,
    @NewDate DATE,
    @LastAuditory NVARCHAR(10),

```

```

        @NewAuditory NVARCHAR(10)

AS

        UPDATE DEANERY

                SET AuditoryID=(SELECT IDAuditory FROM Main_Deanery WHERE
Name_of_Auditory=@NewAuditory)

                WHERE AuditoryID=(SELECT IDAuditory FROM Main_Deanery WHERE
Name_of_Auditory=@LastAuditory)

        UPDATE DEANERY

                SET Dates=@NewDate WHERE Dates=@LastDate AND

                (LecturerID=(SELECT IDLecturer FROM Main_Deanery WHERE
Surname=@LecturerSurname AND FirstName=@LecturerFirstname AND
Patronymic=@LecturerPatronymic)

                OR GroupID=(SELECT IDGroup FROM Main_Deanery WHERE
Name_of_Group=@GroupName))

GO

--Modification_Dates_and_Auditory 'Дороговцев', 'Андрей', 'Анатольевич',
'', '2020-01-09', '2020-01-16', '112-7', '116-7'

--GO

--Schedule_for_Group 'ФИ-72'

--GO

--Modification_Dates_and_Auditory '', '', '', 'ФИ-73', '2020-01-11',
'2020-01-16', 'БФ-1'

--GO

--Schedule_for_Group 'ФИ-73'


GO -- Розклад іспитів на кафедрі, впорядковане за категоріями викладачів
(старший викладач, доцент, професор)

CREATE PROC Report_to_1

        @Name_of_Department NVARCHAR(50)

AS

```

```
        SELECT Name_of_Department, Name_of_Group, Name_of_Discipline,  
Dates, Surname, FirstName, Patronymic, Name_of_Auditory, Name_of_Form  
FROM Main_Deanery
```

```
        WHERE Name_of_Form='экзамен' AND  
Name_of_Department=@Name_of_Department
```

```
        ORDER BY Name_of_Form
```

```
GO
```

```
--Report_to_1 'Информационная безопасность'
```

```
GO -- Інформація виду: "Викладач - дисципліна - консультація (дата, час,  
аудиторія) - іспит (дата, час, аудиторія)"
```

```
CREATE PROC Report_to_2
```

```
AS
```

```
        SELECT Surname, Name_of_Discipline, Dates AS Date_of_Consultation,  
Name_of_Auditory AS Name_of_Auditory_for_Consul FROM Main_Deanery
```

```
        WHERE Name_of_Form='консультація' ORDER BY Dates
```

```
        SELECT Surname, Name_of_Discipline, Dates AS Date_of_Examination,  
Name_of_Auditory AS Name_of_Auditory_for_Exam FROM Main_Deanery
```

```
        WHERE Name_of_Form='экзамен' ORDER BY Dates
```

```
GO
```

```
--Report_to_2
```



