

Міністерство освіти і науки України

НТУУ «Київський політехнічний інститут»

Фізико-технічний інститут

Бази даних та інформаційні системи

Розрахунково-графічна робота

Варіант №19

**Виконав:**

Студент III курсу ФТІ

групи ФБ-41

Соколов Юрій

**Перевірив:**

Коломицев М.В.

### *Завдання*

**Предметна область** - деканат інституту.

**Задачі, що вирішує інформаційна система** - контроль відвідувань занять.

**Завдання на обробку даних:**

1. Створити впорядковані списки:

- Студенти - групи;

- ПІБ студентів, які пропустили лабораторні роботи з предмету ХХ;

- Табель відвідуваності по предмету УУ в групі УУ за осінній семестр;

2. обчислення:

- Кількість прогульників у групі ХХ за січень;

- Сумарне число пропусків занять по групах і предметів.

3. корекція:

- Видалення даних за минулий навчальний рік;

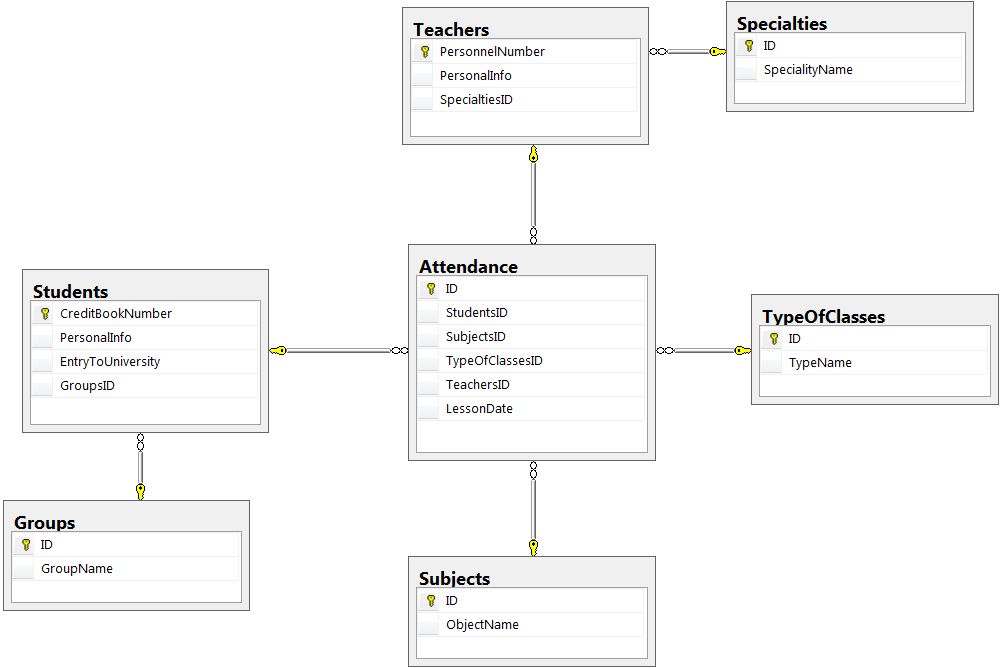
4. Звіти виду:

- Табель відвідуваності за семестр - угруповання з предметів і групам; підсумок - сумарне число пропусків;

- Повідомлення кожному студенту - прогульникові про явку в деканат.

### Діаграма потоків даних (DFD):

**C:\Users\Admin\Desktop\GA7gG8R.png**

**Побудова ER- діаграми: **

**Побудова реляційної моделі:**

1. Модель знаходиться в 1НФ, бо всі атрибути атомарні;
2. Модель знаходиться в 2НФ, бо не містить неповних функціональних залежностей не ключових атрибутів від атрибутів первинного ключа;
3. Модель знаходиться в 3НФ, бо не містить транзитивних залежностей.

**Отримана модель:**

Students = (CreditBookNumber, PersonalInfo, EntryToUniversity, GroupsID)

Specialties = (ID, SpecialityName)

Groups = (ID, GroupName)

Subjects= (ID, ObjectName)

Teachers= (PersonnelNumber, PersonalInfo)

TypeOfClasses= (ID, TypeName)

Attendance= (ID, StudentsID, SubjectsID, TypeOfClassesID, TeachersID, LessonDate)

**Створення таблиць:**

* Студенти:

create table Students(CreditBookNumber varchar(7) CONSTRAINT StudentsPrimary PRIMARY KEY, PersonalInfo varchar(50) NOT NULL,

EntryToUniversity date NOT NULL, GroupsID decimal(9,0) NOT NULL)

* Спеціальності:

CREATE TABLE Specialties(ID decimal(9,0) CONSTRAINT SpecialtiesPrimary PRIMARY KEY, SpecialityName varchar(50) NOT NULL)

* Групи:

CREATE TABLE Groups(ID decimal(9,0) CONSTRAINT GroupsPrimary PRIMARY KEY, GroupName varchar(6) NOT NULL)

* Предмети:

create table Subjects(ID decimal(9,0) CONSTRAINT SubjectsPrimary PRIMARY KEY, ObjectName varchar(35) NOT NULL)

* Тип заняття:

create table TypeOfClasses(ID decimal(9,0) CONSTRAINT TypeOfClassesPrimary PRIMARY KEY, TypeName varchar(20) NOT NULL)

* Відвідування занять:

create table Attendance(ID decimal(9,0) CONSTRAINT AttendancePrimary PRIMARY KEY, StudentsID varchar(7) NOT NULL, SubjectsID decimal(9,0) NOT NULL,

TypeOfClassesID decimal(9,0) NOT NULL, TeachersID varchar(12) NOT NULL, LessonDate date NOT NULL)

**Створення обмежень:**

**FOREIGN KEYS**

alter table Students

add constraint StudentsGroupsForeign FOREIGN KEY (GroupsID) REFERENCES Groups(ID)

alter table Attendance

add constraint AttendanceStudentsForeign FOREIGN KEY (StudentsID) REFERENCES Students(CreditBookNumber) ON DELETE CASCADE

alter table Attendance

add constraint AttendanceSubjectsForeign FOREIGN KEY (SubjectsID) REFERENCES Subjects(ID) ON DELETE CASCADE

alter table Attendance

add constraint AttendanceTypeOfClassesForeign FOREIGN KEY (TypeOfClassesID) REFERENCES TypeOfClasses(ID) ON DELETE CASCADE

alter table Attendance

add constraint AttendanceTeachersForeign FOREIGN KEY (TeachersID) REFERENCES Teachers(PersonnelNumber) ON DELETE CASCADE

alter table Teachers

add SpecialtiesID decimal(9,0) NOT NULL

alter table Teachers

add constraint TeachersSpecialtiesForeign FOREIGN KEY (SpecialtiesID) REFERENCES Specialties(ID) ON DELETE CASCADE

**Обмеження на введення даних:**

alter table Students

ADD CONSTRAINT PersonalInfoLike CHECK (PersonalInfo NOT LIKE '%[^a-z]% %[^a-z]%');

alter table Students

ADD CONSTRAINT CreditBookNumberLike CHECK (CreditBookNumber LIKE '[a-z][a-z]-[0-9][0-9][0-9][0-9]');

alter table Groups

ADD CONSTRAINT GroupNameLike CHECK (GroupName LIKE '[a-z][a-z]-[1-9][1-9]');

alter table Groups

ADD CONSTRAINT GroupIDCheck CHECK (ID>0);

alter table Specialties

ADD CONSTRAINT SpecialityNameLike CHECK (SpecialityName NOT LIKE '%[^a-z]%');

alter table Specialties

ADD CONSTRAINT SpecialtiesIDCheck CHECK (ID>0);

alter table Subjects

ADD CONSTRAINT SubjectsIDCheck CHECK (ID>0);

alter table Subjects

ADD CONSTRAINT ObjectNameLike CHECK (ObjectName NOT LIKE '%[^a-z]% %[^a-z]%');

alter table Teachers

ADD CONSTRAINT PersonnelNumberLike CHECK (PersonnelNumber LIKE '%[0-9][0-9][0-9][0-9][0-9][0-9]%');

alter table Teachers

ADD CONSTRAINT PersonalInfoTeachersLike CHECK (PersonalInfo NOT LIKE '%[^a-z]% %[^a-z]%');

alter table TypeOfClasses

ADD CONSTRAINT TypeOfClassesIDCheck CHECK (ID>0);

alter table TypeOfClasses

ADD CONSTRAINT TypeNameLike CHECK (TypeName NOT LIKE '%[^a-z]%');

**1. Створити впорядковані списки:**

- Студенти – групи:

CREATE VIEW StudentsGroup

AS

SELECT CreditBookNumber, PersonalInfo, GroupName FROM Students join Groups A ON GroupsID=A.ID

- ПІБ студентів, які пропустили лабораторні роботи з предмету ХХ:

USE Deanery

GO

CREATE PROC AttendanceLab

@ObjectName varchar(35)

AS

IF EXISTS(Select \* from Subjects

WHERE ObjectName=@ObjectName)

IF EXISTS(Select \* from Subjects A join Attendance ON A.ID=SubjectsID

WHERE ObjectName=@ObjectName)

SELECT PersonalInfo, LessonDate FROM Students join Attendance ON CreditBookNumber=StudentsID join Subjects A ON SubjectsID=A.ID

WHERE ObjectName=@ObjectName AND TypeOfClassesID=3

ELSE

PRINT 'Пропусков лабораторных работ по этому предмету не найдено!'

ELSE

PRINT 'Этого предмета нет в базе данных'

- Табель відвідуваності по предмету УУ в групі УУ за осінній семестр:

USE Deanery

GO

CREATE PROC TabelAutumn

@ObjectName varchar(35),

@GroupName varchar(6)

AS

IF EXISTS(Select \* from Subjects

WHERE ObjectName=@ObjectName)

IF EXISTS(Select \* from Groups

WHERE GroupName=@GroupName)

select PersonalInfo, count(StudentsID) as [Количество Пропусков] from Students join Attendance ON CreditBookNumber=StudentsID

join Subjects A ON SubjectsID=A.ID join Groups B ON GroupsID=B.ID

where GroupName=@GroupName AND ObjectName=@ObjectName AND DATEPART(mm,LessonDate)>='09' AND DATEPART(mm,LessonDate)<='12'

group by PersonalInfo

having count(StudentsID) IN (select count(StudentsID) from Attendance

group by StudentsID)

ELSE

PRINT 'Этой групы нет в базе данных!'

ELSE

PRINT 'Этого предмета нет в базе данных!'

- Звіт відвідування занять :

CREATE VIEW AttendanceOtchet

AS

SELECT CreditBookNumber, PersonalInfo, GroupName, ObjectName, TypeName, TeachersID, LessonDate

FROM Students join Attendance ON CreditBookNumber=StudentsID join Groups G ON GroupsID=G.ID join Subjects S ON SubjectsID=S.ID

join TypeOfClasses T ON TypeOfClassesID=T.ID

**2. Обчислення:**

- Кількість прогульників у групі ХХ за січень;

CREATE PROC ZADANIYA4

@GroupName varchar(6)

AS

IF EXISTS(Select \* from Groups WHERE GroupName=@GroupName)

select COUNT(\*) AS [Количество прогульщиков за январь] FROM (Select PersonalInfo, COUNT(PersonalInfo) AS [Количество прогулов]

From AttendanceOtchet A

where EXISTS (Select PersonalInfo from AttendanceOtchet B

WHERE A.PersonalInfo=B.PersonalInfo) AND GroupName=@GroupName AND DATEPART(mm,LessonDate)='01'

GROUP BY PersonalInfo) AS Tab

ELSE

PRINT 'Этой групы нет в базе данных!'

- Сумарне число пропусків занять по групах і предметів:

CREATE PROC CountProgulov

@GroupName varchar(6),

@ObjectName varchar(35)

AS

IF EXISTS(Select \* from Groups WHERE GroupName=@GroupName)

IF EXISTS(Select \* from Subjects WHERE ObjectName=@ObjectName)

select COUNT(\*) AS [Количество прогульщиков по данному предмету] FROM (Select PersonalInfo, COUNT(PersonalInfo) AS [Количество прогулов]

From AttendanceOtchet A

where EXISTS (Select PersonalInfo from AttendanceOtchet B

WHERE A.PersonalInfo=B.PersonalInfo) AND GroupName=@GroupName AND ObjectName=@ObjectName

GROUP BY PersonalInfo) AS Tabl

ELSE

PRINT 'Этого предмета нет в базе данных!'

ELSE

PRINT 'Этой групы нет в базе данных!'

**3. Корекція:**

- Видалення даних за минулий навчальний рік;:

use Deanery

go

CREATE TRIGGER ClearAttendance

ON Attendance

FOR DELETE

AS

DECLARE @DATE varchar(10)

SELECT @DATE=DATEPART(mm,GETDATE())

IF @DATE='07'

BEGIN

DELETE FROM Attendance WHERE ID IN (SELECT ID FROM DELETED)

PRINT('Таблица успешно очищена.')

END

ELSE

BEGIN

RAISERROR('Учебный год ещё не закончен', 16, 1)

ROLLBACK TRAN

END

**4. Звіти виду:**

- Табель відвідуваності за семестр - угруповання з предметів і групам; підсумок - сумарне число пропусків:

CREATE VIEW SumProgul

AS

Select GroupName, ObjectName, COUNT(\*) AS [Суммарное число прогулов] From AttendanceOtchet

GROUP BY GroupName, ObjectName

HAVING COUNT(\*) IN (Select COUNT(PersonalInfo) AS [Количество прогулов]

From AttendanceOtchet A

where EXISTS (Select PersonalInfo from AttendanceOtchet B

WHERE A.PersonalInfo=B.PersonalInfo)

GROUP BY GroupName, ObjectName)

- Повідомлення кожному студенту - прогульникові про явку в деканат:

CREATE PROC TIZVEZDA

@PersonalInfo varchar(50)

AS

IF EXISTS(Select \* from Students WHERE PersonalInfo=@PersonalInfo)

IF EXISTS(Select \* from AttendanceOtchet WHERE PersonalInfo=@PersonalInfo)

PRINT 'Вас ждут в деканате!!!'

ELSE

PRINT 'Все хорошо.'

ELSE

PRINT 'В базе данных нету студента с таким именем.'