

Студ. Деликатная М.М.

Науч. рук. доц. Жилияк Н. А.

(кафедра информационных систем и технологий, БГТУ)

### БАЗА ДАННЫХ «СТУДЕНТЫ»

При компьютерной обработке информации упорядоченные данные принято хранить в базах данных – файлах, использование которых вместе со специальными программными средствами позволяет пользователю как просматривать необходимую информацию, так и, по мере необходимости, манипулировать ею, например, добавлять, изменять, копировать, удалять, сортировать и т.д [1]. Цель данной работы – создание базы данных для более эффективной работы деканата. В этой базе будет храниться информация о студентах и посещенных ими мероприятиях за все время обучения. Проектирование базы данных происходило в четыре этапа [2]: анализ предметной области, определение целей, выбор системы управления базами данных (СУБД) и проектирование базы данных.

Анализ предметной области. В базе данных «Студенты» будет храниться информация мероприятиях, которые посетили студенты за время обучения в университете, а также информация о самих студентах. В данной базе будет предоставлена возможность изменения и обработки информации, а также возможность выполнения запросов и получение отчетов.

Особенности базы (рисунок 1):

- каждый факультет имеет несколько специальностей;
- на каждой специальности имеется несколько групп;
- каждая группа имеет список студентов;
- каждый студент участвовал в одном или нескольких мероприятиях.

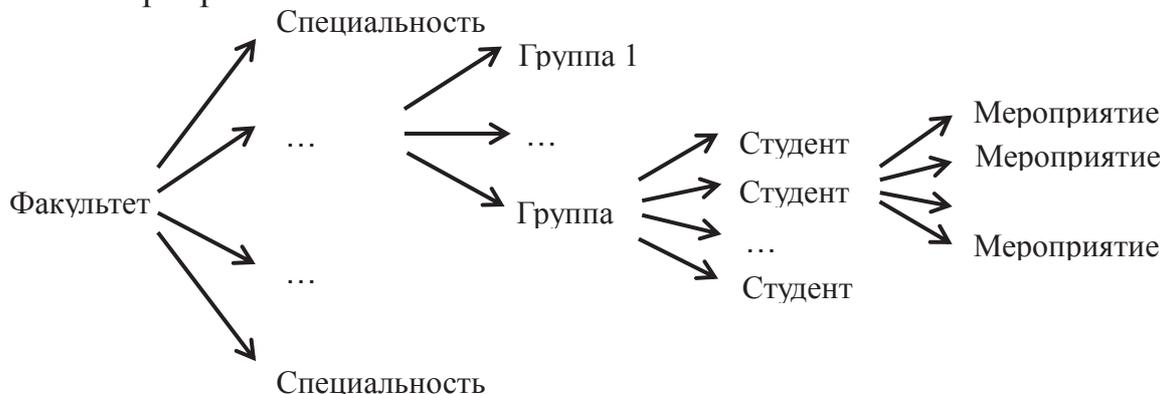


Рисунок 1 – Особенности базы

База данных является реляционной. В ней данные хранятся в таблицах, которые связаны отношениями. Каждая запись в таблице соответствует отдельной сущности, а количество строк записей в таблице неограниченно.

В базе данных «Студенты» выделено пять сущностей: *факультеты, специальности, группы, список студентов и список мероприятий*. Для каждой выбраны атрибуты и определены типы связей.

Сущность «*Список мероприятий*»:

- код мероприятия;
- название мероприятия;
- код студента.

Сущность «*Список студентов*»:

- код студента;
- фамилия;
- имя;
- отчество;
- курс;
- номер подгруппы;
- форма обучения;
- код группы.

Сущность «*Список студентов*» связана с сущностью «*Список мероприятий*» связью один ко многим.

Сущность «*Группы*»:

- аббревиатура;
- код специальности;
- код группы.

Сущность «*Группы*» связана с сущностью «*Список студентов*» связью один ко многим.

Сущность «*Специальности*»:

- аббревиатура;
- код факультета;
- код специальности.

Сущность «*Специальности*» связана с сущностью «*Группы*» связью один ко многим.

Сущность «*Факультеты*»:

- аббревиатура;
- код факультета.

Таблица «*Факультеты*» связана с таблицей «*Специальности*» связью один ко многим.

Определение целей. БД создаётся для декана факультета и преподавателей. Она должна отображать сведения о студентах и мероприятиях, которые они посетили. Также должен осуществляться поиск данных, а также реализация запросов и отчетов. Запросы должны быть направлены на получение информации о студенте и информации о посещенных им мероприятиях.

Выбор СУБД. Система управления базами данных – совокупность программных и лингвистических средств общего или специального назначения, обеспечивающих управление созданием и использованием баз данных [3].

Данная база создавалась с помощью системы управления реляционными базами данных (СУРБД) *Microsoft SQL Server*. Основной используемый язык запросов, который компания *Microsoft* позиционирует как диалект языка *SQL* для своей фирменной СУРБД — *Transact-SQL* [4].

Проектирование БД. С учетом нормализации и составленных отношений, база данных будет выглядеть следующим образом, представленным на рисунке 2:

```

USE Студенты;

CREATE TABLE Факультеты
(
    "Код_факультета" INT IDENTITY(1,1) PRIMARY KEY,
    "Аббревиатура" NVARCHAR(40) NOT NULL,
);

CREATE TABLE Специальности
(
    "Код_специальности" INT IDENTITY(1, 1) PRIMARY KEY,
    "Аббревиатура" NVARCHAR(30) NOT NULL,
    "Код_факультета" INT FOREIGN KEY REFERENCES "Факультеты"(Код_факультета) NOT NULL
);

CREATE TABLE Группы
(
    "Код_группы" INT IDENTITY(1, 1) PRIMARY KEY,
    "Аббревиатура" NVARCHAR(30) NOT NULL,
    "Код_специальности" INT FOREIGN KEY REFERENCES "Специальности"(Код_специальности) NOT NULL
);

CREATE TABLE Список_студентов
(
    "Код_студента" INT IDENTITY(1, 1) PRIMARY KEY,
    "Фамилия" NVARCHAR(30) NOT NULL,
    "Имя" NVARCHAR(30) NOT NULL,
    "Отчество" NVARCHAR(30) NOT NULL,
    "Курс" NVARCHAR(30) NOT NULL,
    "Подгруппа" NVARCHAR(30) NOT NULL,
    "Форма_обучения" NVARCHAR(30) NOT NULL,
    "Код_группы" INT FOREIGN KEY REFERENCES "Группы"(Код_группы) NOT NULL
);

CREATE TABLE Список_мероприятий
(
    "Код_мероприятия" INT IDENTITY(1, 1) PRIMARY KEY,
    "Название_мероприятия" NVARCHAR(30) NOT NULL,
    "Код_студента" INT FOREIGN KEY REFERENCES "Список_студентов"(Код_студента) NOT NULL
);

```

Рисунок 2 – Проектирование БД

Таким образом, было создано пять таблиц. Все значения полей таблиц являются неделимыми, не имеется совпадающих записей. Все неключевые поля в таблицах зависят только от ключа таблицы и не зависят друг от друга. Это значит, что все таблицы БД приведены к 3 нормальной форме.

В каждой таблице имеется первичный ключ (*primary key*), который является идентификатором объекта. Каждый идентификатор – это счетчик (*identity(1,1)*), который автоматически увеличивается на единицу при добавлении в таблицу новой записи. Таблицы также включают поля с символьным типом данных (*nvarchar*). Для установления связей между данными в таблицах были созданы внешние ключи (*foreing key*). Все поля таблицы должны быть обязательно заполнены (значение *NOT NULL*).

#### ЛИТЕРАТУРА

1. Пинкус, М. Введение в базы данных [Электронный ресурс] / М. Пинкус. – Режим доступа: <http://onmcsso.narod.ru/db/>.
2. Карпова, И. П. Проектирование реляционных баз данных [Электронный ресурс] / И. П. Карпова. – Режим доступа: <http://rema44.ru/resurs/study/dbprj/dbprj.html>.
3. Эталонная модель управления данными: ГОСТ Р ИСО/МЭК ТО 10032-2007. Введен 27.12.07. – Москва: Стандартинформ, 2009. – с.3.
4. Ицик, Б. Г. Microsoft SQL Server 2012. Основы T-SQL / Б. Г. Ицик. – М. : Эксмо, 2015. – 400 с.

УДК 378.147

Студ. А. В. Бабако

Науч.рук.: доц., к.т.н. Н. Н. Пустовалова  
(кафедра информационных систем и технологий, БГТУ),  
доц., к.п.н. Н. П. Коровкина

(кафедра автоматизации производственных процессов и электротехники, БГТУ)  
**ОБУЧАЮЩАЯ ПРОГРАММА ДЛЯ ИЗУЧЕНИЯ ФАЙЛОВОГО  
ВВОДА/ВЫВОДА НА ЯЗЫКЕ C/C++**

Приложение для изучения файлового ввода/вывода на языке C/C++ реализовано в виде двух этапов обучения. Первый этап представлен в виде теста, на втором этапе пользователь должен отвечать на различные вопросы по теме. Для перехода с первого этапа на следующий необходимо правильно выполнить больше 70%