МВД Украины

Национальный университет внутренних дел

Факультет управления и информатики

Кафедра информационных систем и технологий в деятельности ОВД

**Курсовая работа**

**по дисциплине: Организация баз данных и знаний**

**Создание журнала посещаемости занятий**

Выполнили:

студенты гр. 418-к

Баришанская А.Л., Вакула Т.А.

Проверил:

доц. каф. ИСиТ в ОВД

п\п-к милиции Танянский С.С.

Харьков

2009

**Содержание**

# Введение

1. Анализ предметной области

Описание задачи

Ограничения ведения базы данных

Постановка задачи

1. Проектирование структуры базы данных

Определение функциональных зависимостей

Разработка структуры базы данных.

Организация запросов к базе данных

1. Организация ведения базы данных

#### Заключение

**Введение**

Для построения БД будем использовать СУБД Access, которая является легкой для самостоятельного освоения, удобной для создания структуры таблиц и определение свойств атрибутов. Диалоговая среда СУБД дает возможность пользователю практически без использования языка программирование устанавливать разные виды поддержки целостности данных.

С помощью конструктора СУБД Access разработаны экранные формы для ввода данных облегчающие режим заполнения с помощь всплывающими полей и заданной последовательностью переходов по полям. Все запросы активизируются кнопками. При этом информация, полученная с их помощью, представлена в экранных формах удобных для просмотра и обработки.

Понятие структура физической БД включает: формат физической записи, кластеры записей, методы размещения и доступа к физическим записям.

БД состоит из файлов, структура которых отвечает всем представленным к ней требованиям, обеспечивает оптимальный вариант решения задачи и не несет избыточной информации. Все файлы БД обладают следующими свойствами: функциональная полнота, минимальная избыточность, целостность, непротиворечивость данных, восстанавливаемость, согласованность, безопасность, эффективность, логическая и физическая независимость, расширяемость, дружественность пользовательского интерфейса, реализуемость средствами конкретного инструментария.

Для улучшения поиска, просмотра, сортировки записей в файле они индексируются в соответствии с их ключевыми полями. Результирующие данные работы задачи можно предоставить конечному пользователю в виде отчета или ведомости, выводимых на экран, в файл или принтер, что позволяет конечному пользователю работать с удобной ему формой отчетности*.*

**1. Анализ предметной области**

**1.1 Описание задачи**

Необходимо создать БД, которая хранит журнал посещаемости занятий. Как элементы данных в БД должна помещаться следующая информация:

Тип занятий (лекция, практика, л/р);

Предмет (БД, Информатика);

Преподаватель (Танянский, Горелов, Струков, Лановой);

Кафедра (информатики)

Фамилия студента

Имя студента

Отчество студента

Дата проведения занятий

Признак посещаемости занятий студентами (был, не был)

В результате анализа предметной области выделим в качестве первичного ключа атрибуты ПРЕДМЕТ, ПРЕПОДАВАТЕЛЬ, ТИП ЗАНЯТИЙ, так как по каждому предмету один и тот же тип занятий должен вести один преподаватель, каждый преподаватель относится только к одной кафедре. Фамилия, Имя, Отчество также выделим в виде ключа , так как не может быть полных однофамильцев.

Таким образом, база данных, полученная на основании заданных атрибутов, будет иметь схему представленную на рисунке 1, где подчеркнутые атрибуты являются первичным ключом.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Тип занятий | Предмет | Преподаватель | Тип занятий | Фамилия | Имя | Отчество | Дата | Признак |

Рисунок 1. Схема БД журнала посещаемости занятий.

**1.2 Ограничения ведения базы данных**

В процессе ведения БД необходимо поддерживать соответствия (целостность) между введенными данными на основе требований обусловленных предметной областью.

Для рассмотренной задачи определим соответствия между атрибутами:

1. Тип занятий и предмет определяет преподавателя;
2. Преподаватель определяет кафедру;
3. ФИО, дата, предмет, тип занятий определяют признак посещаемости;

Таким образом, каждый тип занятий по одному предмету должен вести один преподаватель. Преподаватель относится к одной кафедре. Если студент в определенное время был на определенном предмете, значит он не мог не быть, он был. Подобным образом поддержка соответствий должна быть реализована для каждого заданного ограничения.

Кроме этого, хранение данных в одной таблицы при заданных ограничениях является избыточной.

Так, например, фамилии будут повторятся столько раз, сколько данный студент посещал предметы, дата будет повторятся столько раз, сколько в данный день было проведено занятий, предметы будут повторятся в зависимости от типа занятий.

Структура БД, состоящая из одной таблицы, как представлена на рисунке 1, не дает возможность сохранять все данные.

**1.3 Постановка задачи**

Проведенный анализ предметной области показал, что ведение данных в одной таблице не отвечает некоторым требованиям, предъявляемым к реляционным БД по причине, описанным в предыдущем разделе.

Таким образом, для решения задачи посещаемости занятий необходимо представить ее структуру в виде нескольких таблиц, каждая, из которой содержит отдельный факт предметной области.

Например, информация о преподавателе (ФИО, кафедра), о загрузке преподавателя (тип занятий , предмет, преподаватель), и о признаке посещаемости (тип занятий, предмет, ФИО, дата, признак). При этом БД должна представлять собою целостную систему, то есть пользователь должный иметь возможность в любой момент времени получить всю (любую) информацию, которая хранится в БД.

**2. Проектирование структуры базы данных.**

база данное аccess журнал

**2.1 Определение функциональных зависимостей**

На основании рассмотренных требований к БД (раздел 1.2) и поставленной задачи (раздел 1.3) формализуем ограничения на данные в виде функциональных зависимостей.

1. Тип занятий, предмет **→** Преподаватель
2. Предмет **→** Кафедра
3. ФИО, дата, предмет, тип занятий **→** Признак посещаемости

**2.2 Разработка структуры базы данных**

Для исключения возможных аномалий описанных в разделе 1.2 необходимо нормализовать БД, то есть привести ее к нормальной форме. Заданные ограничения в виде функциональных зависимостей (раздел 2.1.) позволяют построить третью нормальную форму (3НФ), которая устранит нежелательные свойства ведения БД.

Очевидно, что представленный набор атрибутов (рисунок 1) соответствует первой нормальной форме (1НФ). Воспользуемся определением полной функциональной зависимости [1,2] и построим вторую нормальную форму (2НФ).

Таким образом, БД будет иметь вид представленный на рисунке 2.

*Таблица 2 Таблица 1 Таблица 3*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Преподаватель** | 1 | **Тип занятий** | 1 ∞ | **Тип занятий** |
| Кафедра | ∞ | **Предмет** | 1 ∞ | **Предмет** |
|  |  | Преподаватель |  | **Фамилия** |
|  |  |  |  | **Имя** |
|  |  |  |  | **Отчество** |
|  |  |  |  | **Дата** |
|  |  |  |  | Признак |

Рисунок 2. Структура БД в 2НФ.

При этом функциональные зависимости будут соответствовать таблицам, следующим образом:

1. таблицы 1 соответствуют функциональные зависимости

Тип занятий, предмет **→** Преподаватель

2. таблицы 2 соответствуют функциональные зависимости

* Предмет **→** Кафедра

3. таблицы 3 соответствуют функциональные зависимости

* ФИО, дата, предмет, тип занятий **→** Признак посещаемости

Ключевые атрибуты в полученных таблицах определенные на основе заданных функциональных зависимостей между атрибутами. При этом тип связи между всеми таблицами соответствует «один ко многим», так как связные атрибуты у одной таблицы являются первичным ключом, а у другой нет.

Для полученной схемы БД определим свойства каждой таблицы (рисунок 3).

*Таблица 1 Таблица 3*

|  |  |
| --- | --- |
| Тип занятий | Тип данных: *текстовый*  Размер поля: *50*  Обязательное поле *: да*  Пустые строки*: нет*  Индексированное поле*: нет* |
| Предмет | Тип данных: *текстовый*  Размер поля: *50*  Обязательное поле *: да*  Пустые строки*: нет*  Индексированное поле*: нет* |
| Фамилия | Тип данных: *текстовый*  Размер поля: *50*  Обязательное поле *: да*  Пустые строки*: нет*  Индексированное поле*: нет* |
| Имя | Тип данных: *текстовый*  Размер поля: *50*  Обязательное поле *: да*  Пустые строки*: нет*  Индексированное поле*: нет* |
| Отчество | Тип данных: *текстовый*  Размер поля: *50*  Обязательное поле *: да*  Пустые строки*: нет*  Индексированное поле*: нет* |
| Дата | Тип данных: *текстовый*  Размер поля: *50*  Обязательное поле *: да*  Пустые строки*: нет*  Индексированное поле*: нет* |
| Признак | Тип данных: *текстовый*  Размер поля: *50*  Обязательное поле *: да*  Пустые строки*: нет*  Индексированное поле*: нет* |

|  |  |
| --- | --- |
| Тип занятий | Тип данных: *текстовый*  Размер поля: *50*  Обязательное поле *: да*  Пустые строки*: нет*  Индексированное поле*: нет* |
| Предмет | Тип данных: *текстовый*  Размер поля: *50*  Обязательное поле *: да*  Пустые строки*: нет*  Индексированное поле*: нет* |
| Преподаватель | Тип данных: *текстовый*  Размер поля: *50*  Обязательное поле *: да*  Пустые строки*: нет*  Индексированное поле*: нет* |

*Таблица 2*

|  |  |
| --- | --- |
| Преподаватель | Тип данных: *текстовый*  Размер поля: *50*  Обязательное поле *: да*  Пустые строки*: нет*  Индексированное поле*: нет* |
| Кафедра | Тип данных: *текстовый*  Размер поля: *50*  Обязательное поле *: да*  Пустые строки*: нет*  Индексированное поле*: нет* |

**2.3 Организация запросов к базе данных**

Для отчетных форм сформулируем запросы к БД.

1. Сделать выборку по ФИО, когда, кто пропустил занятие, по какому предмету и какой тип занятия попущен (получить на экране таблицу: предмет, тип занятия, дата).
2. Показать кафедру и прдметы, которые менее посещаемы.
3. Показать где работает каждый преподаватель
4. Информация о преподавателях (какой преподаватель ведет какой предмет и какой тип занятий)

Для получения требуемой информации сформулированы запросы на SQL

1. SELECT Фамилия, Имя, Отчество FROM Пользователь
2. PARAMETERS [Введите название темы] CHAR(50);

SELECT Фамилия, Имя, Отчество FROM Учет INNER JOIN Тема ON Учет.Тема=Тема.Тема WHERE Тема.Тема=[Введите название темы]

1. SELECT Фамилия, Имя, Отчество, COUNT([Название сайта или статьи]) AS "Количество сайтов или статей"

FROM Учет INNER JOIN Адрес ON Учет.[WWW адрес]=Адрес.[WWW адрес] GROUP BY Фамилия, Имя, Отчество

1. SELECT Подразделение, COUNT([Название сайта или статьи]) AS "Количество сайтов или статей"

FROM (Учет INNER JOIN Адрес ON Учет.[WWW адрес]=Адрес.[WWW адрес]) INNER JOIN Пользователь ON Учет.Фамилия=Пользователь.Фамилия AND Учет.Имя=Пользователь.Имя AND Учет.Отчество=Пользователь.Отчество

GROUP BY Подразделение HAVING COUNT ([Название сайта или статьи])>10

1. PARAMETERS [Введите название подразделения] CHAR(50);

SELECT Пользователь.Фамилия, Пользователь.Имя, Пользователь.Отчество, Подразделение

FROM (Учет INNER JOIN Адрес ON Учет.[WWW адрес]=Адрес.[WWW адрес]) INNER JOIN Пользователь ON Учет.Фамилия=Пользователь.Фамилия AND Учет.Имя=Пользователь.Имя AND Учет.Отчество=Пользователь.Отчество

WHERE Подразделение<> [Введите название подразделения] AND [Название сайта или статьи] IN

(SELECT [Название сайта или статьи]

FROM (Учет INNER JOIN Адрес ON Учет.[WWW адрес]=Адрес.[WWW адрес]) INNER JOIN Пользователь ON Учет.Фамилия=Пользователь.Фамилия AND Учет.Имя=Пользователь.Имя AND Учет.Отчество=Пользователь.Отчество

WHERE Подразделение= [Введите название подразделения])

**3. Организация ведения базы данных**

Данная информационная система предназначена для преподавателей (узнать когда и кто пропустил занятие) и для студентов (узнать на какой кафедре работает данный преподаватель, какой предмет ведёт преподаватель). На главной форме нанесены кнопки, которые позволяют открывать запросы. В нижней части главной формы расположена таблица студентов. На главной форме нанесены кнопки, при нажатии которыхпользователь может получить необходимую информацию (информация о преподавателях, место работы). Для того, чтобы осуществить переход от одной записи к другой, необходимо воспользоваться клавишами перехода . Для выхода необходимо нажать кнопку . Для закрытия программы необходимо нажать кнопку



.**Заключение**

В результате выполнения курсовой работы была разработанная структура БД ЖУРНАЛА ПОСЕЩАЕМОСТИ ЗАНЯТИЙ, определенны свойства атрибутов и поддержка целостность данных. Полученная структура обеспечивает независимое сохранение и ведение данных о студентах, преподавателях. Предлагаемая структура гарантирует исключения ряда аномалий, таких как избыточность данных, добавление и удаление различных категорий данных независимо друг от друга. Такая гарантия подтверждается тем, что структура разработана на основании формального аппарата нормализации.

Предлагаемые запросы показывают, что, используя язык SQL возможно получение информации из БД, формируя естественно-языковые конструкции любой сложности.

В качестве основных результатов курсовой работы можно выделить следующее:

1. Проанализирована предметная область и сформулирована задача для разработки.
2. Определены ограничения ведения данных в виде функциональных зависимостей между атрибутами.
3. Предложенные запросы к данным реализованы на SQL с использованием встроенных итоговых функций.
4. Разработанный интерфейс обеспечивает пользователя удобным вводом данных и просмотром выводимой информации.

На сегодняшний день очень большое значение уделяется выбору инструментария, отвечающего требованиям разработчиков. Современная жизнь насыщена информацией, которая пронизывает все стороны человеческой деятельности. Для хранения информации и поиска нужных данных требуется создание все большего количества баз и банков данных.

Не последнее место в списке СУБД, предлагаемому к выбору, занимает Access, благодаря тому, что оно, обладая всеми чертами классической СУБД, является также и удобной системой в разработке приложений, работающих с БД.

Отличительной чертой Access является объектно-ориентированный язык программирования, развитые визуальные средства разработки, поддержка стандартных протоколов обмена данных.

Современные информационные технологии позволяют экономить труд и время не только программистов, но и специалистов по эксплуатации и поддержке программных продуктов. Для удовлетворения возрастающих требований к обработке данных революционными темпами развиваются системы клиент – сервер, и роль Visual FoxPro в таких системах занимает значительное место.

Access может эффективно использоваться совместно с приложениями Internet не только из-за мощного процессора данных и удобного языка программирования MS SQL Server.

**Список использованных источников**

1. Дейт К. Введение в системы баз данных. М.: “Вильямс” 2001.
2. Т. Коннолли, К. Бегг, А. Страчан Базы данных: проектирование, реализация и сопровождение. Терия и практика, 2-е изд.: Пер. с англ.: Уч. Пос. – М.: Издательский дом “Вильямс”, 2000.
3. Д.Грофф, П.Вайнберг*.* SQL: полное руководство. - BHV-Киев, 1999.