



УДК 621.317

ПРОЕКТИРОВАНИЕ СИСТЕМЫ УЧЕТА ПОСЕЩАЕМОСТИ СТУДЕНТАМИ ЗАНЯТИЙ И НАУЧНОЙ ЛАБОРАТОРИИ

Г.А. Столбовой, М.Г. Попов, М.В. Козырева, А.И. Киллер

Южно-Российский Государственный Политехнический Университет (НПИ) имени М.И. Платова,
г. Новочеркасск

Для построения базы данных (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 программирования устанавливать разные виды поддержки целостности данных.

DESIGNING THE ATTENDANCE ACCOUNTING SYSTEM STUDENTS OF LESSONS AND RESEARCH LABORATORY

G.A. Stolbovoy, M.G. Popov, M.V. Kozyreva, A.I. Killer

Platov South-Russian State Polytechnic University (NPI), Novocherkassk

To build a database (DB), we will use the Access DBMS, which is easy for self-development, convenient for creating the table structure and defining attribute properties. The interactive environment of the DBMS allows the user to install various types of data integrity support with virtually no programming language.

С помощью конструктора СУБД Access разработаны экранные формы для ввода данных облегчающие режим заполнения с помощью 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, дружелюбность 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.

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.

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

Для рассмотренной задачи определим соответствия между атрибутами:

Тип занятий и предмет определяет преподавателя;

Преподаватель определяет кафедру;

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

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 должен иметь возможность в любой момент времени получить всю (любую) информацию, которая хранится в БД.

На основании рассмотренных требований к БД и поставленной задачи формализуем ограничения на данные в виде функциональных зависимостей.

Тип занятий, предмет → Преподаватель

Предмет → Кафедра



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 отчетных форм сформируем запросы к БД.

Сделать выборку по ФИО, когда, кто пропустил занятие, по какому предмету и какой тип занятия попушен (получить на экране таблицу: предмет, тип занятия, дата).

Показать кафедру и предметы, которые менее посещаемы.

Показать где работает каждый преподаватель

Информация о преподавателях (какой преподаватель ведет какой предмет и какой тип занятий)

Для получения требуемой информации сформулированы запросы на SQL

SELECT Фамилия, Имя, Отчество FROM Пользователь

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]

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

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

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])



В результате была разработана структура БД журнала посещаемости занятий, определены свойства атрибутов и поддержка целостности 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 работы можно выделить следующее:

Проанализирована предметная область и сформулирована задача для разработки.

Определены ограничения ведения данных в виде функциональных зависимостей между атрибутами.

Предложенные запросы к данным реализованы на SQL с использованием встроенных итоговых функций.

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.