

УДК 004.658

РАЗРАБОТКА СИСТЕМЫ ПРОЕКТИРОВАНИЯ БАЗ ДАННЫХ ДЛЯ СОХРАНЕНИЯ И ОБРАБОТКИ ТЕХНИЧЕСКОЙ ИНФОРМАЦИИ.

К.В. КАБАРГИНА^{1*}, Л.Б. КАЩЕЄВ²

¹ магістрант кафедри САІТ, НТУ «ХПІ», Харків, УКРАЇНА

² доцент кафедри САІТ, канд. техн. наук, НТУ «ХПІ», Харків, УКРАЇНА

* email: kabargina.katerina@gmail.com

В процессе информатизации общества происходит проникновение информационных технологий во все сферы жизнедеятельности общества. Вычислительная техника все более широко внедряется в системы управления технологическими процессами. Отличительная особенность таких систем управления – необходимость выполнения сложных видов обработки больших объемов информации в крайне ограниченные сроки.

База данных – это единственное, вместительное хранилище разнообразных данных и описаний их структур, которое после своего определения, осуществляемого отдельно и независимо от приложений, используется одновременно многими приложениями.

Кроме данных база данных может содержать средства, позволяющие каждому из пользователей оперировать только теми данными, которые входят в его компетенцию. В результате взаимодействия данных, содержащихся в базе, с методами, доступными конкретным пользователям, образуется информация, которую они потребляют и на основании которой в пределах собственной компетенции производят ввод и редактирование данных.

Существует 3 типа базы данных:

1) реляционные: информация хранится в виде таблиц. На каждый объект отводится 1 строка таблицы – запись, характеристики (атрибуты) объектов расположены по столбцам и называются полями записи. Одно из полей – «первичный ключ» – содержит уникальную информацию о каждом объекте, чаще всего это идентификационный номер. Записи одной таблицы могут быть связаны с записями других таблиц, связь направлена от произвольного поля первой таблицы («вторичный» или «внешний» ключ) к первичным ключам других таблиц, тип связи – «один к одному», «один ко многим», «многие ко многим»;

2) иерархические (древовидные): на более высоких уровнях расположены группы элементов более низких уровней. Одной группе, как правило, соответствуют несколько элементов более низких уровней (тип связи – «один ко многим»). Типичный пример: файловая система компьютера;

3) сетевые: элементы, имеющие какие-то общие свойства, связаны друг с другом в некое подобие сети, тип связи – «многие ко многим».

Реляционные СУБД сейчас являются наиболее популярными, в которых данные хранятся в реляционных таблицах, и для них разработан стандарт языка обработки – SQL.

Целью данной работы является создание базы данных без базы данных (без MDF файла). База данных заполняется в текстовом файле, то есть мы составляем виртуально базу данных в памяти. На машину не нужно ставить сервер и устанавливать MS SQL.

Реляционные базы данных, у них большое количество данных, а если мы имеем не очень много данных, то возникает идея: а можно ли сделать проще?

База данных без сервера означает локальная база. Под сервером можно понимать серверную операционную систему (ОС), в качестве которой могут выступать любые ОС, поддерживающие сетевую работу, в таком случае возможна работа в режиме файл-сервера. Это означает, что (одно) многопользовательский доступ обеспечивается за счет совокупности файловых возможностей ОС, возможностей многопользовательской работы используемой машины баз данных, например, BDE, MS Jet и др. и особенностей такого программирования. Если под сервером понимать сервер баз данных (InterBase, Oracle, MS SQL и пр.), то это клиент-серверные технологии, применяемые при сложных задачах.

В базе бывает очень много записей, бывает очень мало. Если мало записей, подключать реляционные таблицы типа Oracle или MS SQL это нецелесообразно, потому что если мы создаем консольный справочник без интернета, то мы 95% возможностей не используем. Но, возможности визуальной среды Visual Studio в частности C#, в которой есть внутренняя база данных (data table) и внутренние компоненты связи, именно то к чему мы подключались (серверу), а так мы подключаемся именно к его временной виртуальной СУБД, создаваемой видео компонентами.

Преимущества:

- 1) запуск без установки;
- 2) запуск из любой папки;
- 3) картинок в базе нет (используются ссылки на конкретный файл).

Недостаток всего один и очень большой: когда мы зайдем за большое число, все что умеет база данных, мы сделать не сможем.

Такой подход очень удобен, и очень прост в применении. Однако использование данного подхода может вызвать трудности в разработке программы с большим количеством данных.

Список літератури:

1. Serverless Architectures [Електронний ресурс] / Fowler, M. // Режим доступу: http://informatika.edusite.ru/lezione9_42.htm
2. Serverless Architectures [Електронний ресурс] / Fowler, M. // Режим доступу: <https://moodle.kstu.ru/mod/book/view.php?id=11573>