

Обычно есть две категории пользователей, которые работают с базами данных. Первая категория — это дизайнеры. Их задача - разработать структуру таблиц базы данных и согласовать ее с заказчиком. Помимо таблиц, дизайнеры разрабатывают другие объекты базы данных, предназначенные, с одной стороны, для автоматизации работы с базой данных, а с другой - для ограничения функциональности работы с базой данных (при потребности по соображениям безопасности). Дизайнеры не заполняют базу данных конкретными данными (заказчик возможно предположить их конфиденциальными и не предоставлять посторонним). Исключением является опытное заполнение модельными данными на этапе отладки объектов базы данных.

Вторая группа, работающих с базами данных, — это пользователи. Они получают исходную базу данных от проектировщиков и занимаются ее наполнением и обслуживанием. Как правило, пользователи не имеют доступа к управлению структурой базы данных – исключительно к данным, да и то не ко всем, а к тем, работа с какими предусмотрена на конкретном рабочем месте.

Соответственно, СУБД имеет два режима работы: проектный и пользовательский. Первый режим специализирован для создания или изменения структуры базы данных и создания ее объектов. Во втором режиме ранее специализированные объекты используются для заполнения базы данных или получения из нее данных.

Объекты базы данных.

Таблицы являются основными объектами любой базы данных. Во-первых, таблицы хранят все данные, доступные в базе данных, а во-вторых, таблицы хранят структуру базы данных (поля, их типы и свойства).

Эти объекты используются для извлечения информации из таблиц и предоставления их пользователю комфортным способом. Запросы

используются для выполнения таких операций, как выбор данных, сортировка и фильтрация. С помощью запросов вы можете выполнять преобразования данных в соответствии с заданным алгоритмом, создавать новые таблицы, автоматически заполнять таблицы данными, импортированными из других источников, выполнять простые вычисления в таблицах и многое другое.

Если запросы — это специальные инструменты для отбора и анализа данных, то формы — это инструменты для ввода данных. Их смысл равен – дать юзеру вероятность заполнять только те поля, что ему необходимо заполнить. В то же время специальные элементы управления (счетчики, выпадающие списки, переключатели, флажки и т.) могут быть размещены в форме для автоматизации ввода. Превосходства формуляров исключительно бесспорны при вводе данных из заполненных формуляров. В данном случае бланк выполнен графическими средствами таким образом, что повторяет дизайн бланка – это значительно упрощает работу наборщика, сокращает его утомляемость и предотвращает типографские ошибки.

По своим свойствам и структуре отчеты во многом похожи с формами, но они предназначены только для заключения данных, вдобавок для вывода не на экран, а на принтер. В этом отношении отчеты различаются тем, что в них приняты особые меры для группировки выходных данных и вывода особых элементов оформления, характерных для печатных документов.

Это специальные объекты базы данных, реализованные в последних версиях СУБД Microsoft Access (начиная с Access 2000). Однако вернее называть их страницами доступа к данным. Физически это особый объект, выполненный в HTML-коде, расположенный на веб-странице и передаваемый клиенту вместе с ней. Сам по себе этот объект не является базой данных, но включает

компоненты, после которые передаваемая веб-страница подключается к базе данных, оставшейся на сервере. Используя эти компоненты, гость веб-сайта сможет просматривать записи базы данных в полях страницы доступа. Таким образом, страницы доступа к данным обеспечивают интерфейс между клиентом, сервером и базой данных, размещенной на сервере. Эта база данных не обязательно должна являться базой данных Microsoft Access. Страницы доступа, сделанные средствами Microsoft Access, вдобавок позволяют вам работать с базами данных Microsoft SQL Server.

Макросы и модули.

Эти категории объектов специализированы как для автоматизации повторяющихся действий при работе с СУБД, да и для создания новых функций через программирования. В СУБД Microsoft Access макросы состоят из последовательности внутренних команд СУБД и являются одним из средств автоматизации работы с базой данных. Модули формируются с использованием внешнего языка программирования, в данном случае Visual Basic для приложений. Это одно из средств, с помощью которого разработчик базы данных сможет внедрить в нее нестандартный функционал, удовлетворить своеобразные требования заказчика, нарастить эффективность системы управления, а также уровень ее безопасности.

Проектирование базы данных.

Методически конкретно начинать работать с карандашом и листом бумажки в руках, не прибегая к помощи компьютера. На данном шаге в этом попросту нет необходимости. Неоптимальные решения и прямые ошибки, заложенные на этапе проектирования, очень трудно устранить позже, потому этот шаг является основополагающим.

1) Разработка технического задания.

Техническое задание на проектирование базы данных должно быть предоставлено заказчиком. Однако для этого он обязан владеть подходящей терминологией и знать, хотя бы в общих чертах, технические способности

основной СУБД. К сожалению, на практике такая ситуация появляется не всегда. Поэтому, естественно, применяются следующие подходы:

Продемонстрируйте заказчику работу похожей базы данных, после чего они согласуют спецификацию отличий;

Если аналога нет, они узнают круг задач и потребности заказчика, после чего помогают ему организовать техническое задание.

При подготовке технического задания они составляют:

- Список исходных данных, с которыми работает клиент;
- Список выходных данных, необходимых заказчику для управления структурой его компании;
- Перечень выходных данных, которые не являются важными заказчику, но которые он должен предоставить прочим организациям (структурам более высокого уровня, органам статистического учета, другим административным и надзорным организациям).

В то же время весьма важно не ограничивать связь с ведущим отделом заказчика, а проводить обсуждения со всеми службами и отделами, которые могут быть поставщиками данных в базу данных или их потребителями.

## 2) Разработка структуры базы данных.

Выяснив ключевую часть данных, которые потребляет или доставляет клиент, вы можете приступить к созданию структуры базы данных, то есть структуры ее основных таблиц.

Работа начинается с составления общего списка полей – он может содержать десятки или даже сотни позиций.

В соответствии с типом данных, размещаемых в каждом поле, установите особенно пригодный тип для каждого поля.

Далее поля общего перечня распределяются в соответствии с базовыми таблицами. На первом шаге распределение выполняется на функциональной основе. Цель состоит в том, дабы обеспечить ввод данных в одну таблицу, по возможности, в рамках одного отдела или, что еще лучше, на одном рабочем месте.

В каждой из таблиц выделено главное поле. Таким образом, выберите поле, в котором данные не могут быть повторены. Например, для таблицы данных обучающихся таким полем может служить индивидуальный шифр учащихся. Для таблицы, содержащей расписание занятий, такое поле может отсутствовать, но его можно создать, искусственно объединив поля "Время занятий" и "Номер аудитории". Эта комбинация не повторяется, поскольку не принято проводить два разных занятия в одном классе одновременно. Если в таблице вообще нет полей, которые можно было бы использовать в качестве ключевых слов, вы постоянно можете ввести дополнительное поле типа счетчика – по определению, оно не может содержать повторяющихся данных.

Используя карандаш и бумагу, набросайте соединения между таблицами. Такой чертеж именуется схемой данных. Существует несколько видов вероятных связей между таблицами. Наиболее общераспространенными являются отношения "один ко многим" и "один к одному". Связь между таблицами организована на основе общего поля, и одна из таблиц должна иметь ключевое поле, то есть на стороне "один" должно быть ключевое поле, включающее уникальные, неповторяющиеся значения. Значения на стороне "много" могут повторяться.

"Бумажный" этап работы над техническим предложением завершается разработкой схемы данных. Эту схему возможно согласовать с заказчиком, а затем начать к прямому созданию базы данных.

Следует помнить, что в ходе разработки проекта заказчику наверняка придут в голову новые идеи. На всех стадиях проектирования он стремится охватить целой системой все новые подразделения и службы предприятия. Возможность эластичного использования его пожеланий во многом определяется квалификацией разработчика базы данных. Если схема данных скомпилирована правильно, ввести новые таблицы к базе данных не составит труда. Если конструкция базы данных нерациональна, разработчик сможет столкнуться с серьезными проблемами и вступить в конфликт с заказчиком.

Противоречия между поставщиком и заказчиком постоянно указывают на недостаточную квалификацию подрядчика. Именно поэтому шаг предварительного проектирования базы данных следует считать основным. Его успех зависит от того, насколько удобной станет база данных и будут ли пользователи с ней работать. Если отмечается, что пользователи базы данных "саботируют" ее работу и выбирают работать классическими методами, это свидетельствует не о низкой квалификации пользователей, а о недостаточной квалификации разработчика базы данных.

На этом этапе заканчивается подготовительное проектирование базы данных, и на последующем этапе начинается ее непосредственная разработка. С этого момента вам следует начать работать с СУБД.