

*Старушенкова Екатерина Евгеньевна, преподаватель  
факультета довузовской подготовки и среднего профессионального  
образования Мордовский Государственный Университет им. Н. П. Огарева,  
Республика Мордовия, г. Саранск*

## ОСНОВНЫЕ ЭТАПЫ ПРОЕКТИРОВАНИЯ БАЗ ДАННЫХ

**Аннотация:** В данной статье подробно описана технология создания базы данных, приведена основная терминология, используемая при работе с базами данных: определения, классификации баз данных, их представления, модели данных. А так же приведена поэтапная схема разработки, заполнения и работы с базой данных.

**Ключевые слова:** СУБД, система СПО, студенты, обучающиеся, база данных.

**Annotation:** This article describes in detail the technology for creating a database, provides the basic terminology used when working with databases: definitions, classification of databases, their presentation, data models. As well as a phased diagram of the development, filling and working with the database.

**Keywords:** DBMS, open source system, students, students, database.

База данных в общем смысле представляет собой определенную форму, которая используется для структурированного хранения информации.

В настоящее время существует достаточно много классификаций баз данных, к примеру, по технологиям хранения все имеющиеся базы данных можно разделить на два типа:

1. Базы данных, которые размещаются непосредственно в памяти одного персонального компьютера, называют централизованными;

2. Базы данных, состоящие из нескольких частей и размещенные в памяти разных персональных компьютеров, называют распределенными.

Не смотря на многообразие операционных систем, ни в одной нет средства для работы с базами данных. Именно поэтому для грамотной работы с базами данных, а конкретнее для их редактирования и модифицирования были созданы системы управления базами данных [3].

По аналогии с операционной системой, основной частью систем управления базами данных является программа управления для автоматической работы процессов, имеющих доступ к основным частям баз данных, именуемая ядром. При запуске системы управления базами данных ядро выполняет одни из главных функций: организует параллельное выполнение запросов, управляет очередями операций при выполнении, но при этом постоянно находится в основной памяти.

Современные авторы в своих научных работах выделяют определенные разновидности информационных моделей, которые характеризуются в зависимости от среды и времени использования.

Иерархическая модель данных представляет собой дерево, которое имеет родительские и дочерние блоки, каждый из которых связан между собой определенным типом связи, применимым в базах данных.

Однако данная модель будет иметь как достоинства, так и недостатки.

К плюсам модели можно отнести простоту модели в использовании, ведь дерево – это наглядно, удобно и доступно.

Однако, самым большим недостатком, который чаще всего заставляет отказаться от данной модели является невозможность быстрого поиска «ответа» на заданный «вопрос», ведь если дерево имеет большое количество ветвей, то необходимо совершить большое количество переходов, на что не всегда есть время у программиста.

Следующей моделью данных именуют сетевой, в свою очередь она разделяется на система файл-сервер и система клиент-сервер [2].

Система файл-сервер представляет собой один персональный компьютер, который хранит в себе все файлы базы данных, а непосредственно доступ осуществляется при помощи персональных компьютеров, именуемые рабочими станциями. И непосредственно эти рабочие станции выбирают информацию с вычислительной машины и обрабатывают ее.

Характерной особенностью системы файл-сервер является общий коллективный доступ к базе данных, хранимой на файловом сервере. Все данные запрашиваются рабочими станциями из хранилища и обрабатываются при помощи систем управления базами данных.

В системе клиент-сервер кроме хранения базы данных на центральную машину ложатся и функции обработки данных, а на клиентских машинах выполняется только представление информации.

Однако, имеются и недостатки. Что произойдет, если вдруг сервер перестанет отвечать? Казалось бы, проблема не страшная. Но, если в этот момент программист обновлял данные в базе данных или извлекал их, то это грозит минимум их потерей.

И третья модель данных, разработанная по времени позже всех, но по мнениям программистов, наиболее простая и эффективная – реляционная.

Реляционная модель представляет собой таблицы, содержащие в себе данные. Данные представлены в таком виде для того, что бы пользователям было удобно осуществлять поиск, редактирование, модифицирование. Таблицы состоят из строк и столбцов, каждый из которых имеет свои имена.

### **Этапы работы с базами данных**

Для того, что бы пользователь создал базу данных, которая была проста и понятна в использовании не только ему, но и человеку, который будет с ней работать позже; а так же для более корректного выполнения технического задания, если разработчик работает на предприятии, существуют определенные требования, предъявляемые к базам данных [1].

**1 этап.** Построение модели базы данных.

Для построения модели необходимо провести анализ предметной области, по которой будет создаваться уже имеющаяся база данных, либо проанализировать ту базу данных, которая дается разработчику. Определить, какие данные будет содержать база данных, откуда разработчик будет их получать, сам ли он будет вносить их в базу данных. Необходимо определить, какие задачи будет выполнять база данных, так как для определенного функционала необходима особенная структура базы данных.

## **2 этап.** Выбор системы управления базы данных.

Достаточно просто будет работать программисту, если ему для работы дают практически готовую базу данных, с выбранными целями, задачами, структурами и уже выбранной системой управления базой данных. Будет немного сложнее, если разработчик не умеет работать с уже выбранной системой управления базой данных.

Допустим, нам необходимо самостоятельно выбрать систему управления базами данных.

Что необходимо учесть при выборе? В первую очередь, это требования заказчика, так как мы выполняем, прежде всего, его задание. Второй нюанс, который нужно учитывать – то, кто будет в последующем работать с данной базой данных. Если заполнять ее будет не разработчик, то необходимо учесть навыки программиста и использовать современные и популярные СУБД. Третий нюанс – непосредственно наши навыки. Мы не сможем создать базу данных при помощи СУБД Oracle, если имеем навыки работы только с MS Access.

Наиболее оптимальной СУБД для новичков, по моему мнению, является MS SQL Server. Данный продукт официальный продукт корпорации Microsoft, является свободнораспространяемым и понятным [4].

## **3 этап.** Создание структуры таблиц базы данных.

На данном этапе разработчику необходимо определить типы данных, которые будут хранить таблицы, задать размеры всех полей в каждой таблице и определить ключи.

#### **4 этап.** Нормализация.

В настоящее время авторы учебников и учебных пособий отмечают, что нормализовать базу данных значит избавить таблицы базы данных от избыточности. Иными словами в таблицах, согласно первой нормальной форме, не должно быть многозначно определенных значений полей. Каждая таблица обязательно должна иметь уникальный идентификатор, простыми словами – первичный ключ. Почему ключ так важен? С помощью ключа пользователь легко найдет точную информацию о типах сущности или об объектах таблиц. И, для завершения нормализации, необходимо привести таблицу к третьей нормальной форме, а именно исключить из полей таблиц информацию, которая влияет на другие поля. В MS Excel, эти поля называют вычисляемыми. То есть, можно сказать, что пользователю необходимо убрать «вычисляемые» поля.

**5 этап.** Редактирование и модифицирование данных, содержащихся в таблицах.

Ввести данные можно в базу данных двумя способами: ввести в уже имеющуюся таблицу или при помощи форм.

А непосредственно редактирование и модифицирование происходит при помощи запросов.

#### **6 этап.** Определение связей.

Далеко не каждая база данных ограничивается одной таблицей. Для того, что бы база данных была наиболее полной и правильной необходимо «связать» таблицы между собой. Связь в базе данных это взаимосвязь между сущностями в конкретной предметной области, а в нашем случае сущности представлены в таблицах. По необходимости пользователь может выбрать один из трех имеющихся типов связей «один к одному», «один ко многим», «многие ко многим». Однако, стоит учесть, что пользователь не может выбрать связь по типу «как захочется», важно оценить ситуацию в базе данных, учесть смысл дублирования записей и так далее.

#### **7 этап.** Вывод информации.

Одним из плюсов работы с базами данных является возможность вывода отчета пользователю. Отчеты включают в себя только необходимую информацию, чаще всего выбирая ее из разных таблиц, структурировать ее понятным для пользователя образом.

Исходя из вышесказанного, можно сделать вывод, что базы данных это новое, до конца не изученное направление, которое может помочь обычному пользователю; для программиста же базы данных являются неотъемлемой частью профессиональной деятельности. И для того, что бы работать с базой данных, необходимо придерживаться правил, нарушение которых, конечно, не несут в себе ошибок, как в математике. Отступление от правил грозит пользователю избыточностью файлов и данных, большими затратами ресурсов персонального компьютера, что вследствие может привести к неисправности базы данных; сервера, если база данных располагается именно там; а так же персональных компьютеров и их системы, на которые будет ложиться нагрузка.

#### **Библиографический список:**

1. Диго С. М. Базы данных: проектирование и использование / С. М. Диго. – М.: Финансы и статистика, 2005. – 153 с.
2. Конноли Т. Базы данных: проектирование, реализация, сопровождение / Т. Конноли, К. Бегг, А. Страчан. – М.: Вильямс, 2003. – 327 с.
3. Илюшечкин, В. М. Основы использования и проектирования баз данных. Учебник / В.М. Илюшечкин. - М.: Юрайт, 2014. - 214 с.
4. Старушенкова, Е.Е. Особенности изучения баз данных в системе среднего профессионального образования / Е.Е.Старушенкова // Научное образование. –2019. – №3. – с. 31-33.