

4. Шесть правил проектирования базы данных

Чем больше власти, тем больше ответственности. Проектировщик БД должен разработать набор таблиц с подходящей структурой. Если сделать это как следует, то экономится масса времени и сил в будущем. Хорошо спроектированные БД легко усовершенствовать, с ними легче работать, они приводят к гораздо меньшему числу трудно разрешимых проблем в случае извлечения информации.

Разработка хорошей БД – это искусство, требующее опыта. Для достижения наилучших результатов предлагаются следующие рекомендации при создании собственных тестовых БД.

Правило 1. Выбирать подходящие имена полей

В программе Access не установлено особых правил для выбора используемых имен полей. Она позволяет включить 64 символа в создаваемое имя. Тем не менее, имена полей важны. Важно выбрать подходящее имя с самого начала.

Далее приведено несколько советов:

- Сделать его простым и коротким. Имя поля должно быть настолько коротко, насколько это возможно. Длинные имена утомительно набирать, в них больше шансов сделать ошибку, их труднее втиснуть в формы и отчеты.
- Пользоваться заглавными буквами. Нельзя сказать, что это "железное" правило, но можно сделать заглавной первую букву каждого слова и затем слить слова вместе для формирования имени поля. Нужно избегать пробелов.
- Не включать в имя поля имя таблицы.
- Не использовать для имени поля слово *Name*. Помимо того, что это труднопроизносимое имя, слово *Name* – ключевое в программе Access.

Правило 2. Разбивать данные

Не включать слишком большую порцию данных в одно поле. Нужно в каждом поле хранить элементарную порцию данных.

Существует множество оснований для разделения информации на отдельные поля. Прежде всего, это устраняет некоторые типы ошибок. В поле *Name* имя можно ввести несколькими разными способами, например, "Фамилия, Имя" или

"Имя Фамилия". Разбиение имени устраняет эти проблемы, способные создать затруднения при попытке использовать данные в автоматизированной задаче какого-либо вида, к примеру, объединение сообщений электронной почты. Гораздо важнее то, что значительно легче работать с данными, которые разделены на маленькие порции. После разделения поля *Name* на поля *Фамилия* и *Имя* можно сортировать и искать информацию, основываясь на одной порции этой информации, чего нельзя было бы сделать в противном случае. Аналогично следует разделить сведения об адресе на несколько столбцов, таких как улица, город, штат или регион и страна – в этом случае гораздо легче найти всех, кто живет, например, в Кемерово.

Правило 3. Хранить все детали в одном месте

Часто одна таблица применяется в решении многих задач. Некоторые из полей таблицы могут использоваться только от случая к случаю. Но все равно следует их все содержать в одной таблице.

Правило 4. Избегать дублирования данных

Когда заполняется таблица полями, иногда возникает желание включить в нее несвязанную информацию. Дублирование данных, неэффективно. Легко вообразить таблицу с сотнями похожих записей, бесполезно расходуя дисковое пространство и повторяющих одни и те же значения снова и снова. Но эта неприятность – мелочь, по сравнению с затратами на обновление подобной информации и возможностью возникновения противоречивости данных.

Правило 5. Избегать избыточной информации

Другой тип несвязанной информации – избыточные данные. Информация, которая уже есть где-то в БД или даже в той же таблице, иногда в слегка иной форме избыточна. Как и в случае дублирования данных, такая избыточность может порождать противоречивость данных.

Вычисляемые данные – самый распространенный тип избыточной информации.

Правило 6. Включать поле Код

Как уже известно, программа Access автоматически создает поле *Код*, когда разрабатывается таблица в *Режиме таблицы*, и назначает его первичным ключом создаваемой таблицы. Но даже, когда изучен *Конструктор*, все равно нужно вставлять поле *Код* в создаваемые таблицы. В поле *Код* используется тип данных *Счетчик*, в этом случае Access автоматически заполнит его числами и отведет ему роль первичного ключа.

В некоторых ситуациях в таблицу может быть включено уникальное поле, которое можно использовать как первичный ключ. Вставляя поле *Код*, всегда увеличивается степень свободы действий. Поле *Код* не придется менять никогда. Другая информация, даже имена и номера социального обеспечения могут измениться. И если используются связи между таблицами, программа Access копирует первичный ключ в другие таблицы. Если первичный ключ меняется, возникает необходимость искать его значение в разных местах БД.