Инструкционная карта № 9.

Тема: «Основные понятия и элементы баз данных»

В древности люди были примитивными и грубыми, они не чистили зубы, убого одевались и хранили данные в текстовых или битовых файлах. Но это время безвозвратно ушло – появились базы данных.

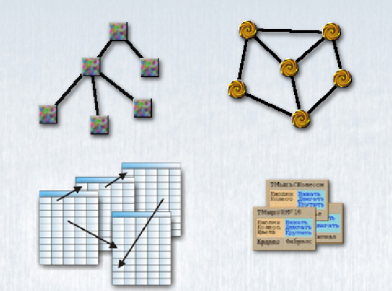
По сути таблицы баз данных не очень сильно отличаются от используемых ранее текстовых файлов данных – те же записанные подряд значения. Однако прежде чем стать таблицами, простые файлы прошли длительную эволюцию, обросли всяческими усовершенствованиями и новшествами, облегчающими работу с ними.

Почувствуйте разницу между обычным использованием служебных файлов и работой с базами данных.

Многие приложения хранят свои данные на диске. Но эти данные (в том числе и сохраненные пользователем во время работы) несут вспомогательную роль – они позволяют модифицировать работу приложения, временно хранить рабочие данные и т.п. Одно приложение обычно хранит множество предназначенных только для него файлов.

При работе с базами данных дело обстоит совсем по-другому. Здесь основным является сама база, а служебными являются приложения, которые работают с ней. С одной базой может работать множество рассчитанных только на нее приложений.

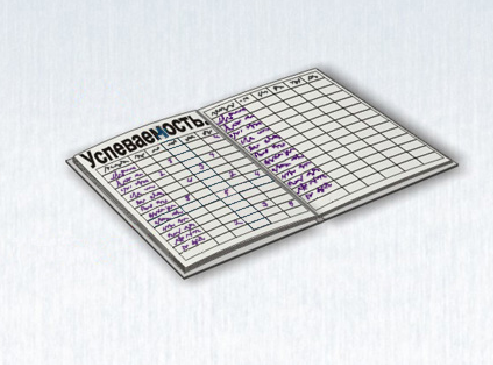
Одни и те же данные можно упорядочить различными способами. В соответствии с этим существует несколько моделей организации баз данных. Иерархические, сетевые, реляционные, объектно-ориентированные. Каждая модель имеет свои достоинства и недостатки, однако в настоящее время наиболее используемой моделью для персональных компьютеров является реляционная модель баз данных. В дальнейшем будет рассматриваться именно эта модель.



Примером реляционной базы данных может служить школьный классный журнал. Журнал состоит из таблиц, в которые вносятся различные данные.

Каждая таблица посвящена какому-то конкретному связанному набору данных и состоит из строк и столбцов. Столбцы посвящены однотипным данным. Строки – конкретным записям по каждому из пунктов. В базах данных столбцы принято называть полями, а строки – записями.

Таблицы в журнале связаны: фамилии учеников фигурируют в разных таблицах, и за счет этого осуществляется связь. При необходимости, используя эту связь, можно узнать средний балл ученика, выяснить, в каком районе города живет больше всего отличников по биологии из данного класса, и т.д.



Поскольку данные, хранимые в базе, обычно имеют достаточно сложную структуру, то в простой таблице их хранить невозможно – не удается передать тонкости структуры. Однако, используя несколько связанных таблиц, можно обойти трудности и создать базу любой сложности.

Таблицы связываются с помощью ключей. Ключом называется поле или несколько полей таблицы, данные в которых однозначно определяет запись. Ключом может служить, к примеру, фамилия ученика. Однако это не очень удобный ключ, поскольку при наличии однофамильцев нарушается однозначность. Обычно в качестве ключа используется специальное поле – индивидуальный номер (или код) записи.

Из двух связываемых таблиц одна является главной, а другая – подчиненной. В одном из полей подчиненной таблицы содержится значение ключевого поля главной таблицы. Таким образом, таблицы связаны, и структура установлена.

Наличие связей, однако, приводит к дополнительным ограничениям. Например, удаление записи в главной таблице должно влечь за собой каскадное удаление или переработку в подчиненной таблице всех записей, в которых имеется ссылка на удаляемую запись. Подобные проблемы могут решаться различными способами. Они могут быть разрешены на физическом уровне таблицы, могут и программно.

Кроме ключевых полей, можно задать также индексные поля таблицы. В отличие от ключа, данные в индексном поле могут и повторяться, то есть однозначность не обязательна. Индексы помогают при сортировке – как и ключи, они ускоряют доступ к записям таблицы.



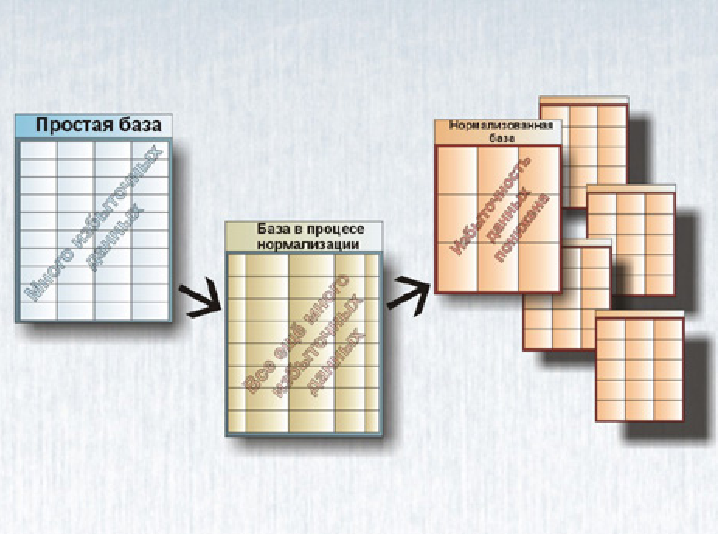
Как отмечалось, для реляционных баз данных разработан некоторый теоретический аппарат. К нему относится, в частности, понятие нормализации баз данных.

Нормализация происходит в процессе проектирования баз данных и состоит из последовательного приведения данных в первую нормальную форму, затем во вторую, в третью и т.д.

Каждая нормальная форма – это набор некоторых требований к структуре данных, которые последовательно понижают избыточность данных. Это особенно актуально для больших баз со сложной структурой.

Стартуя из одной общей таблицы, в которой содержатся все поля базы, в процессе нормализации происходит ее разбиение на несколько таблиц, модифицируются поля.

Несмотря на кажущуюся сложность этого процесса, все эти преобразования носят вполне интуитивно понятный характер. Часто новичок, создавая разумно организованную базу данных, и не подозревает, что то, что он создал, вполне соотносится с теорией и соответствует, к примеру, третьей нормальной форме.



При обработке записей в таблице может использоваться два способа доступа.

В небольших локальных базах данных используется **навигационный** способ. Этот способ заключается в последовательной обработке каждой отдельной записи.

При работе с большими базами обработка ведется с помощью SQL-запросов. Этот способ называется **реляционным**, и в этом случае обрабатывается сразу группа записей.

При работе с базами данных часто необходимо произвести несколько операций подряд, причем недопустим разрыв этой последовательности. Например, при переносе записи из одной таблицы в другую запись сначала удаляется в первой, а затем записывается во второй таблице. Если при этом произойдет сбой, к примеру, перебой электропитания, то может быть нарушена целостность базы данных.

Во избежание подобных случаев используется механизм **транзакций**. Несколько операций объединяются в транзакцию, и если все операции проведены успешно, то изменения базы данных подтверждаются, если же хотя бы одна операция неуспешна, то результаты действия всех операций транзакции отменяются, и база данных возвращается в исходное корректное состояние.

**Бизнес-правила** – еще один механизм, поддерживающий корректность баз данных и облегчающий работу с ними. С помощью бизнес-правил можно задавать диапазон допустимых значений, значения по умолчанию, запрет пустого значения и т.д.

