

Е. А. СИДОРОВА, А. В. ДОЛГОВА

ОСНОВЫ БАЗ ДАННЫХ

ОМСК 2020

Министерство транспорта Российской Федерации
Федеральное агентство железнодорожного транспорта
Омский государственный университет путей сообщения

Е. А. Сидорова, А. В. Долгова

ОСНОВЫ БАЗ ДАННЫХ

Утверждено методическим советом университета
в качестве учебно-методического пособия
к выполнению самостоятельной работы

Омск 2020

УДК 004.65 (075.8)
ББК 32.973-018.2я73
С34

Основы баз данных: Учебно-методическое пособие к выполнению самостоятельной работы / Е. А. Сидорова, А. В. Долгова; Омский гос. ун-т путей сообщения. Омск, 2020. 23 с.

Учебно-методическое пособие разработано в соответствии с рабочими программами дисциплин информационного профиля с учетом требований ФГОС ВО последнего поколения.

Приведены основные понятия баз данных, их классификация. Рассмотрены требования к организации компьютерной базы данных и этапы ее проектирования. Приведены краткие сведения о системах управления базами данных и языке структурированных запросов *SQL*. Представлены контрольные и тестовые вопросы.

Предназначено для самостоятельной работы студентов всех направлений подготовки (специальностей) очной и заочной форм обучения по дисциплинам, изучающим базы данных и системы управления ими.

Библиогр.: 5 назв. Табл. 3. Рис. 9.

Рецензенты: доктор техн. наук, профессор В. Н. Горюнов;
доктор техн. наук, профессор В. А. Нехаев.

ОГЛАВЛЕНИЕ

Введение.....	5
1. Понятие базы данных	6
2. Классификация БД. Модели данных.....	6
2.1. Классификация БД.....	6
2.2. Иерархическая модель данных.....	7
2.3. Сетевая модель данных	8
2.4. Реляционная модель данных	9
3. Требования к организации компьютерной БД.....	10
4. Объекты БД.....	11
5. Межтабличные связи и ключевые поля.....	13
6. Проектирование БД.....	15
7. Системы управления базами данных	16
8. Язык SQL.....	16
9. Особенности работы с файлами БД	18
10. СУБД Microsoft Access	19
11. Контрольные вопросы	20
12. Примеры тестовых вопросов	20
Библиографический список	22

ВВЕДЕНИЕ

Базы данных являются одним из основных компонентов современных информационных систем, которые в настоящее время широко применяются в различных областях человеческой деятельности. Главными задачами организации баз данных являются хранение информации и эффективный доступ к ней. Решение этих задач обеспечивают системы управления базами данных, к которым относится программа *Microsoft Access*, входящая в состав распространенного пакета *Microsoft Office*. Отличительными особенностями этой программы являются удобный интерфейс и наличие интерактивных средств разработки приложений.

В настоящем учебно-методическом пособии рассмотрены основные понятия баз данных, приведена их классификация. Представлены требования к организации компьютерной базы данных и этапы ее проектирования. Приведены краткие сведения о системах управления базами данных и языке структурированных запросов *SQL*. Представлены контрольные и тестовые вопросы.

Библиографический список, приведенный в конце пособия, содержит литературу для углубленного изучения материала по рассматриваемой тематике.

1. ПОНЯТИЕ БАЗЫ ДАННЫХ

База данных (БД) – это совокупность структурированных взаимосвязанных данных, организованная по определенным правилам, которые предусматривают общие принципы описания, хранения и обработки данных. Обычно БД создается для конкретной предметной области. Например, БД Пенсионного фонда, БД системы продажи железнодорожных билетов, БД отдела кадров предприятия и т. п.

Реальный или представляемый объект, информация о котором должна сохраняться в базе данных и должна быть доступной, называется **сущностью**.

2. КЛАССИФИКАЦИЯ БД. МОДЕЛИ ДАННЫХ

2.1. Классификация БД

Базы данных классифицируются по следующим признакам.

1) По характеру информации: *фактографические* (картотеки) и *документальные* (архивы).

Фактографическая БД содержит краткие сведения об описываемых объектах, представленные в строго определенном формате. Например, БД книжного фонда библиотеки (для хранения библиографических сведений о каждой книге) или БД отдела кадров предприятия (для хранения анкетных сведений работников).

Документальная БД содержит обширную информацию самого разного типа: текстовую, графическую, звуковую, мультимедийную. Например, БД законодательных актов (тексты законов) или БД музыкальных записей (тексты и ноты песен, справочная информация об авторах и исполнителях, звуковые записи и видеоклипы).

Современные информационные технологии постепенно стирают границу между фактографическими и документальными БД. Существуют средства, позволяющие легко подключать любой документ (текстовый, графический, звуковой) к фактографической базе данных.

Далее будут рассматриваться только фактографические БД.

2) По способу хранения данных: *централизованные* (все части БД хранятся на одном компьютере) и *распределенные* (различные части одной БД хранятся

на множестве компьютеров, объединенных в сеть). Например, информацию в Интернете можно рассматривать как распределенную БД.

3) По структуре организации (модели) данных: *иерархические, сетевые и реляционные*.

Моделью данных называется формальная теория представления и обработки данных. Она является ядром любой БД и включает в себя три аспекта:

аспект структуры – методы описания типов и логических структур данных в БД;

аспект манипуляции – методы манипулирования данными;

аспект целостности – методы описания и поддержки целостности БД.

2.2. Иерархическая модель данных

Для иерархических структур характерна подчиненность объектов нижнего уровня объектам верхнего уровня. К основным понятиям иерархической структуры относятся *узел, уровень и связь*.

Узел – это совокупность атрибутов (свойств) данных, описывающих некоторый объект. На схеме иерархического дерева (рис. 1) узлы представляются вершинами графа, которые расположены на определенных **уровнях**. Иерархическое дерево на самом верхнем уровне имеет только одну вершину (корень дерева), не подчиненную никакой другой вершине. Зависимые (подчиненные) узлы находятся на расположенных ниже уровнях.

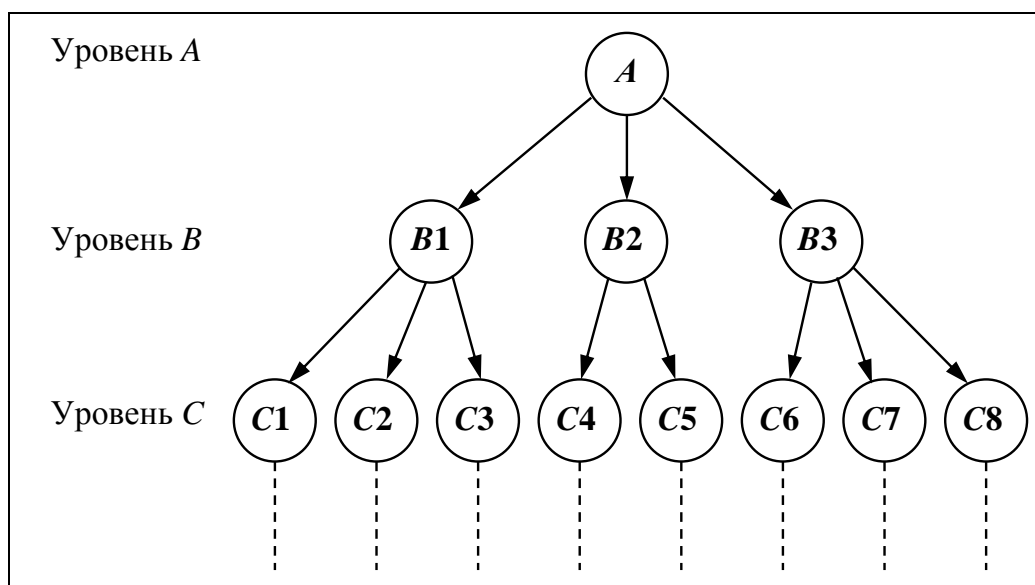
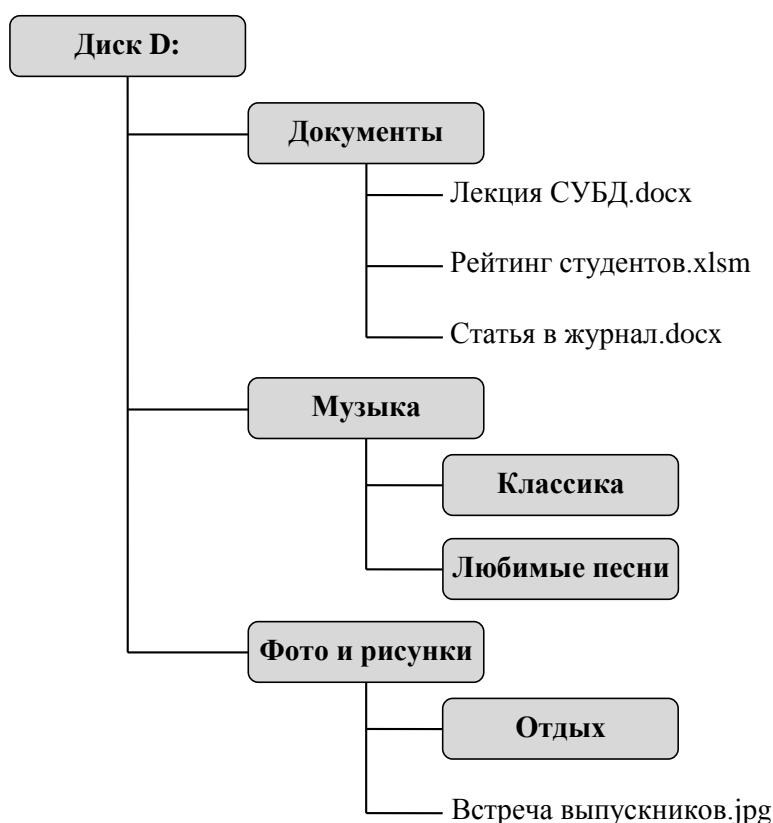


Рис. 1. Структура иерархической модели данных

Графически изображаемое взаимодействие между двумя узлами называется **связью**.

В иерархической структуре каждый узел на более низком уровне связан только с одним узлом, находящимся на более высоком уровне. К каждому узлу существует только один (иерархический) путь от корня дерева, тип связи между верхними и нижними объектами характеризуется как «один-ко-многим».

Примером иерархической модели данных является файловая структура на диске компьютера ([рис. 2](#)).



[Рис. 2.](#) Пример иерархической модели данных

2.3. Сетевая модель данных

Сетевая структура данных предполагает несколько уровней взаимосвязанных объектов. В ней любой объект может быть и главным, и подчиненным. В сетевой структуре при тех же основных понятиях (узел, уровень, связь) каждый элемент может быть связан с любым другим элементом на любом уровне (тип связи между ними «многие-ко-многим»), на каждом уровне может находиться любое число объектов ([рис. 3](#)). Пример сетевой модели данных «Хобби студентов» приведен на [рис. 4](#).

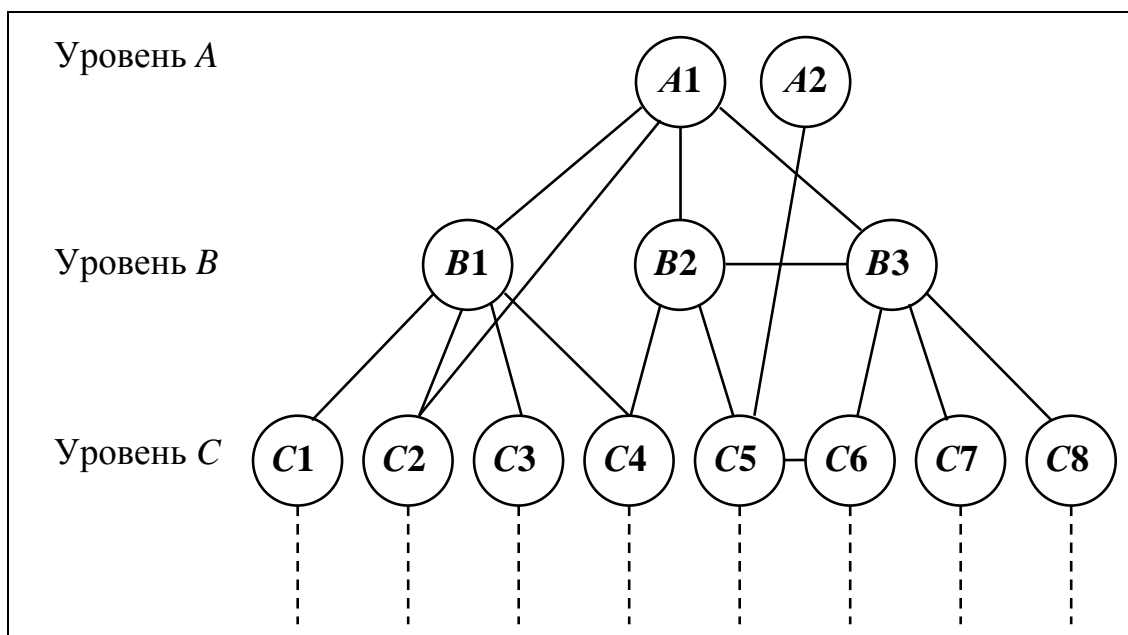


Рис. 3. Структура сетевой модели данных



Рис. 4. Пример сетевой модели данных

2.4. Реляционная модель данных

Иерархические и сетевые модели данных не могут быть реализованы с помощью наиболее распространенных программных средств, поэтому компьютерные БД основаны на реляционной модели.

Реляционная (от англ. *relation* – отношение) модель данных представляет собой совокупность взаимосвязанных двумерных таблиц. Каждая таблица представляет собой набор столбцов, объединяющих сведения о логически самостоятельной части предметной области. Каждая строка в таблице содержит информацию, относящуюся только к одному определенному объекту. Пример реляционной модели данных «Студенты университета» представлен на рис. 5.

В дальнейшем будут рассматриваться только реляционные БД.

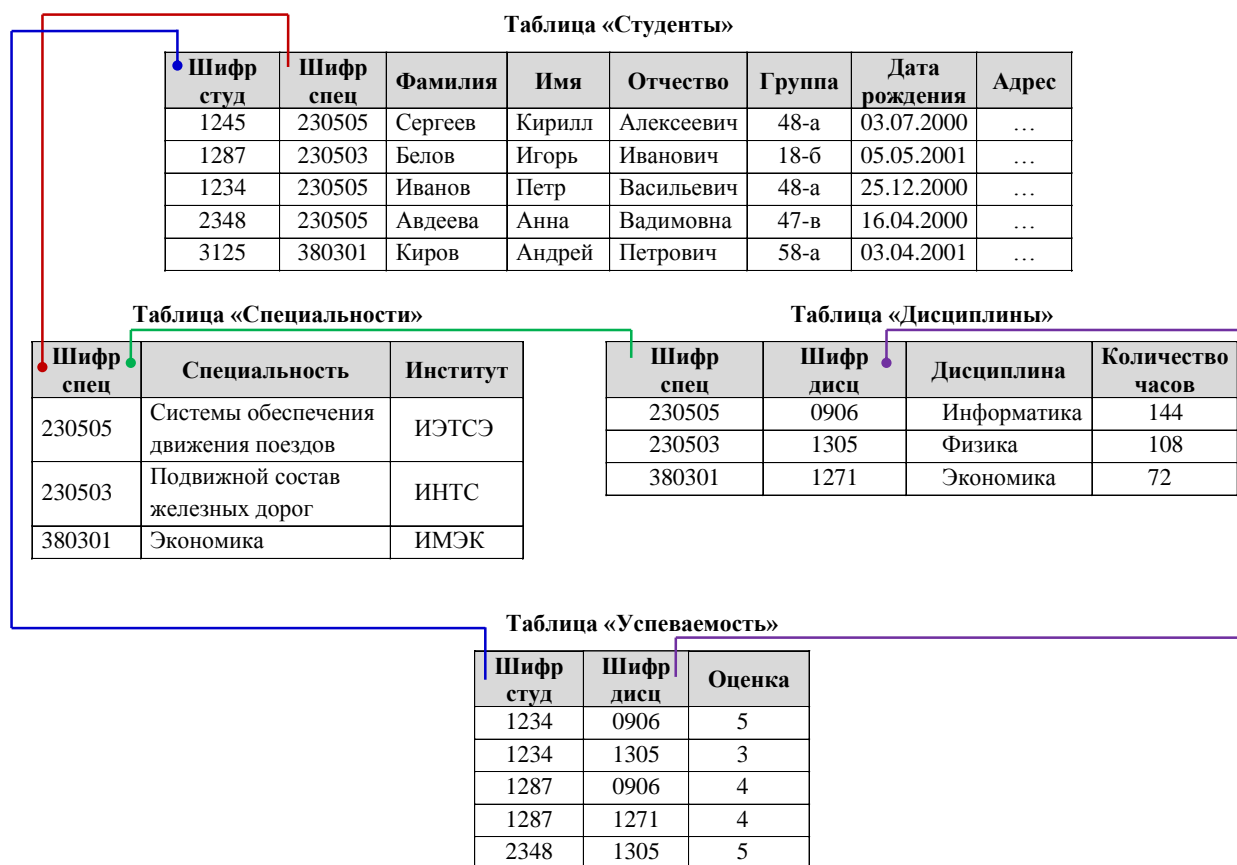


Рис. 5. Пример реляционной модели данных

3. ТРЕБОВАНИЯ К ОРГАНИЗАЦИИ КОМПЬЮТЕРНОЙ БД

Для обеспечения эффективного и надежного функционирования компьютерной БД при ее организации должны быть выполнены следующие требования:

1) *неизбыточность данных* (любые данные хранятся в БД в одном экземпляре, дублирование данных приводит к нерациональному использованию памяти ЭВМ и может привести к противоречивости данных и неверной их обработке);

2) *совместное использование данных* (одни и те же данные могут быть использованы несколькими пользователями при решении различных задач);

3) *расширяемость БД* (возможность увеличения числа однотипных данных, введения в БД новых типов объектов или новых взаимосвязей);

4) *простота работы с БД* (наличие удобного интерфейса);

5) *эффективность доступа к БД* (максимальная скорость доступа при ограничениях на объем оперативной памяти ЭВМ);

6) *целостность БД*, означающая ее сохраняемость физическую (на носителях информации) и логическую (обеспечение непротиворечивости данных);

7) *независимость данных от использующих их программ* (отсутствие необходимости вносить изменения в уже существующие программы при модификациях БД);

8) *секретность данных* (защита БД от несанкционированного доступа).

4. ОБЪЕКТЫ БД

Основными объектами БД являются **таблицы**. Без них БД существовать не может. Если в таблице нет данных (пустая таблица), то это все равно полноценная таблица. Пустая таблица содержит информацию о *структуре таблицы*, которая определяет методы занесения данных и хранения их в базе. Структуру таблицы составляют столбцы, которые в теории баз данных называют **полями**, и строки, называемые **записями**. Поле содержит однородные данные обо всех объектах (например, фамилии студентов группы), а запись – атрибуты (характеристики) одного объекта (например, набор сведений о конкретном студенте). Каждое поле в таблице характеризуется именем, типом данных в нем и набором свойств.

Имя поля показывает, как следует обращаться к данным в нем при автоматических операциях с БД. В имени поля символ точки недопустим.

Тип поля определяется типом данных в нем. Наиболее распространенные типы данных приведены в [табл. 1](#).

Таблица 1

Основные типы данных в полях БД

Тип данных	Назначение
1	2
Текстовый	Хранение текста длиной до 255 символов
Числовой	Хранение цифровой информации
Денежный	Хранение данных о денежных суммах
Дата/время	Хранение дат и времени. Этот тип предоставляет дополнительные возможности по обработке данных, например, позволяет вычислить количество дней между двумя датами или определить дату, отстоящую от заданной на определенный срок
Логический	Хранение логических данных, которые могут принимать только два значения, например, «Да» или «Нет», «Истина» или «Ложь». В БД поля с логическим типом данных, как правило, представляются в виде флажков

1	2
Счетчик	Специальный тип данных для уникальных (не повторяющихся в поле) натуральных чисел с автоматическим наращиванием, он используется обычно для порядковой нумерации записей
<i>MEMO</i>	Хранение больших объемов текста (до 64 Кбайт). В поле <i>MEMO</i> хранится не сам текст, а указатель на него, поэтому упорядочение данных по этому полю невозможно
<i>OLE</i>	Хранение ссылок на объекты (например, файлы изображений, звукозаписей, электронных таблиц, текстовых документов и др.), которые могут быть включены в БД с использованием <i>OLE</i> ¹ -протокола обмена данными

Если в поле содержатся цифровые данные, с которыми не предполагается выполнение математических действий, то такое поле обычно задается как текстовое (например, поле, содержащее номер телефона).

Основные свойства полей БД приведены в табл. 2.

Таблица 2

Основные свойства полей БД

Свойство поля	Назначение
Размер	Предельная длина данных в поле
Маска ввода	Служит для ускорения и контроля ввода информации и характеризует формат ввода данных в поле, например, для ввода даты можно использовать маску <i>00.00.00</i> , для номера телефона – <i>000-000-00-00</i>
Условие на значение	Ограничение, используемое для проверки правильности ввода данных, например, если в поле допускаются только положительные числа, можно ввести условие на значение <i>>0</i>
Обязательное поле	Означает, что при наполнении БД это поле должно быть заполнено обязательно, например, для списка сотрудников это может быть личный шифр
Индексированное поле	Означает, что по данному полю осуществляется упорядочение данных, в связи с чем все операции, связанные с поиском или сортировкой данных этого поля, существенно ускоряются

¹ *OLE* – технология связывания и внедрения объектов.

Помимо таблиц БД может содержать и другие объекты: запросы, формы, отчеты и др.

Запросы – это специальные средства для выбора данных из таблиц и представления их в удобном виде. С помощью запросов выполняют отбор данных, их сортировку и фильтрацию. Кроме того, запросы позволяют преобразовывать данные, создавать новые таблицы, выполнять автоматическое наполнение таблиц данными и простейшие вычисления и т. д. Особенность запросов состоит в том, что они черпают данные из базовых таблиц и создают на их основе временную результирующую таблицу, называемую *моментальным снимком*.

Формы – это средство для ввода, отображения (с применением специальных средств оформления) и редактирования данных. Обычно форма представляет собой бланк, подлежащий заполнению, на котором дополнительно можно разместить элементы управления для работы с другими объектами БД.

Отчеты – это объекты БД, предназначенные в первую очередь для вывода данных на печать. В связи с этим режимы подготовки отчетов содержат команды для группировки выводимых данных, вычисления итогов, а также специальные элементы оформления, характерные для печатных документов (заголовки, колонтитулы, примечания и т. п.).

Макросы служат для автоматического выполнения группы команд.

Модули предназначены для выполнения сложных программных действий.

5. МЕЖТАБЛИЧНЫЕ СВЯЗИ И КЛЮЧЕВЫЕ ПОЛЯ

Эффективное функционирование БД невозможно без связывания таблиц между собой. Связи между таблицами называются **реляционными отношениями**. Основное их назначение заключается в обеспечении целостности данных и автоматизации задач обслуживания БД. Например, реляционные отношения позволяют одновременно отображать родственные сведения об объекте из нескольких таблиц.

Выделяют следующие типы межтабличных связей:

1) *один-к-одному* – каждая запись таблицы А не может быть связана более чем с одной записью таблицы Б (например, паспортные данные соответствуют только одному человеку);

2) *один-ко-многим* (или *многие-к-одному*) – одна запись в таблице А может быть связана со многими записями таблицы Б (например, один студент имеет оценки по разным дисциплинам) (см. рис. 5);

3) *многие-ко-многим* – каждая запись в таблице А может быть связана со многими записями в таблице Б, а каждая запись в таблице Б – со многими записями в таблице А (например, у каждого студента есть преподаватели по разным дисциплинам, а у каждого преподавателя много студентов).

Две таблицы БД связываются между собой с помощью **ключевых полей** (ключей), которые определены в обеих таблицах и однозначно идентифицируют конкретный объект. При этом типы данных связываемых полей должны быть одинаковыми или совместимыми (например, тип «Счетчик» совместим с числовым полем, имеющим размер «Длинное целое»).

В качестве ключей могут выступать:

1) одно поле (простой ключ). Одним из вариантов такого ключа является поле с типом данных «Счетчик»;

2) комбинация полей (составной ключ). Как правило, она применяется при реализации связи типа «многие-ко-многим».


Ключи бывают двух типов: первичные и вторичные (внешние).

Первичный ключ – это одно или несколько полей в таблице, комбинация значений которых однозначно определяет каждую запись в таблице. Первичный ключ используется для связывания таблицы с внешними ключами в других таблицах. Поле первичного ключа не должно быть пустым, значения этого поля в строках таблицы не могут повторяться.

Вторичный (внешний) ключ – это одно или несколько полей в таблице, содержащих ссылку на поле или поля первичного ключа в другой таблице. В поле вторичного ключа совпадение значений в строках таблицы допускается.

Из двух логически связанных таблиц одну называют таблицей первичного ключа или **главной** таблицей, а другую – таблицей вторичного (внешнего) ключа или **подчиненной** таблицей. Ключевые поля и установленные связи позволяют сопоставить родственные записи из обеих таблиц и совместно вывести их в форме, отчете или запросе.

Графическое отображение логической структуры БД и межтабличных связей называется **схемой данных** (рис. 6). Обычно в ней на концах каждой линии связи отображается количество записей в таблице, имеющих одинаковые

значения ключевого поля – 1 (одна запись) и ∞ (много записей), поле первичного ключа отмечается символом .

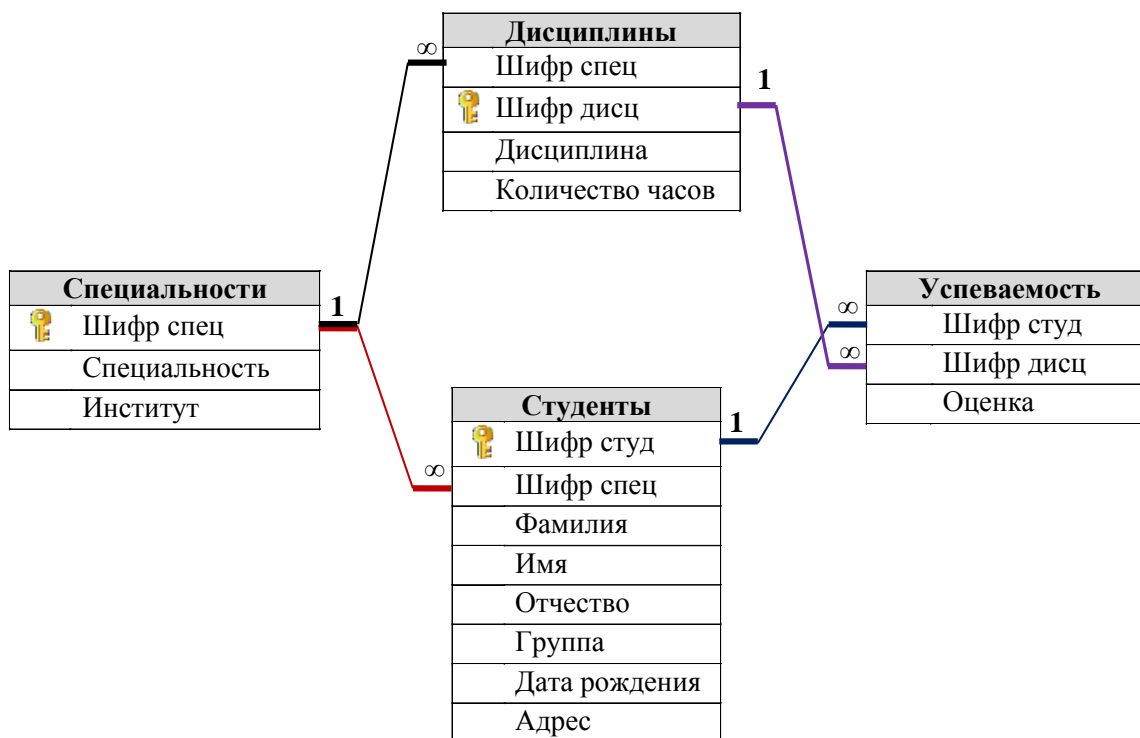


Рис. 6. Пример схемы данных

6. ПРОЕКТИРОВАНИЕ БД

Процесс проектирования БД подразделяется на два этапа: теоретический и практический.

Теоретический этап проектирования БД включает в себя следующие действия:

- создание общего списка полей с указанием их имен, типов и свойств;
- распределение полей по таблицам;
- определение ключевых полей, по которым таблицы связываются между собой;
- определение связей между таблицами, т. е. составление схемы данных.

Практический этап проектирования БД предполагает ее реализацию на компьютере с помощью выбранного программного средства и включает в себя такие операции:

- создание таблиц и связывание их между собой;
- ввод пробных данных;
- создание форм, отчетов и запросов для операций с введенными данными.

7. СИСТЕМЫ УПРАВЛЕНИЯ БАЗАМИ ДАННЫХ

Для реализации БД на компьютере существуют специальные программы – **системы управления базами данных (СУБД)**. Они обеспечивают структурирование информации, размещение ее в таблицах и манипулирование данными. В настоящее время насчитывается более 50 различных СУБД для компьютеров. Из них наиболее распространенными являются *MS SQL Server*, *Oracle Database*, *Informix*, *Sybase*, *IBM DB2*, *FoxPro*, *MS Access*, бесплатные *MySQL*, *PostgreSQL*, *FireBird* и др.

Основные компоненты архитектуры СУБД:

1) *язык описания данных (ЯОД)* – предназначен для описания типов данных, их структур и взаимосвязей;

2) *язык манипулирования данными (ЯМД)* (или язык запросов) – выполняет операции с БД (наполнение, обновление, удаление, выборку информации).

Основные функции СУБД:

1) *непосредственное управление данными во внешней памяти* (для ускорения доступа к данным);

2) *управление буферами оперативной памяти* (в СУБД поддерживается собственный набор буферов оперативной памяти);

3) *управление транзакциями* (**транзакция** – это последовательность операций над БД, рассматриваемая как единое целое. Либо транзакция успешно выполняется, и СУБД фиксирует изменения БД во внешней памяти, либо ни одно из этих изменений никак не отражается на состоянии БД. Понятие транзакции необходимо для поддержания логической целостности БД);

4) *журнализация* (обеспечение надежности хранения данных во внешней памяти, т. е. возможности восстановления последнего согласованного состояния БД после аппаратного или программного сбоя);

5) *поддержка языков БД*.

8. ЯЗЫК SQL

Для создания, модификации и управления данными в реляционных базах данных применяется универсальный компьютерный язык *SQL* (от англ. *Structured Query Language* – язык структурированных запросов). Целью разработки языка *SQL* было создание простого непроцедурного языка для пользова-

телей, не имеющих навыков программирования. Основные команды языка *SQL* приведены в табл. 3.

Таблица 3

Основные команды языка *SQL*

Команда	Назначение
<i>SELECT</i>	Выбор записей из одной или нескольких таблиц БД
<i>INSERT</i>	Добавление новых записей в таблицы БД
<i>UPDATE</i>	Обновление информации в записях таблицы БД
<i>DELETE</i>	Удаление записей из таблицы БД

В командах языка *SQL* могут присутствовать служебные слова, детализирующие запрос на получение информации и имеющие следующее назначение:

FROM – имя таблицы для отбора данных;

WHERE – условие отбора записей из таблиц БД;

ORDER BY – порядок сортировки записей таблицы БД (по умолчанию применяется упорядочение данных по возрастанию).

В общем случае запрос на языке *SQL* состоит из нескольких частей, называемых *предложениями*, и имеет вид:

SELECT Поле

FROM Таблица

WHERE Условие

ORDER BY Сортировка

Предложения *SQL*-запроса можно записывать в одной строке.

Если в имени поля присутствует символ пробела (например, «Шифр спец»), то при написании *SQL*-запроса это имя заключают в квадратные скобки. Текстовые данные в *SQL*-запросе записывают в кавычках, а даты помещают между знаками решетки (#).

Если в *SQL*-запросе фигурируют поля из разных таблиц, то имя каждого поля записывают с указанием имени соответствующей таблицы, разделяя их символом точки, например, *Студенты.Фамилия, Специальности.Институт*.

П р и м е р. При помощи языка *SQL* сформировать запрос, посредством которого выбрать в БД (см. рис. 5) из таблицы «Студенты» значения полей

«Группа», «Фамилия», «Дата рождения» для обучающихся по специальности 230505. Результаты упорядочить по группе, а затем по фамилии.

SQL-запрос, удовлетворяющий условию задачи, будет иметь вид:

```
SELECT Группа, Фамилия, [Дата рождения]
FROM Студенты
WHERE [Шифр спец] = "230505"
ORDER BY Группа, Фамилия.
```

Результат выполнения этого *SQL*-запроса представлен на [рис. 7](#).

Группа ▾	Фамилия ▾	Дата рождения ▾
47-в	Авдеева	16.04.2000
48-а	Иванов	25.12.2000
48-а	Сергеев	03.07.2000

[Рис. 7](#). Результат выполнения *SQL*-запроса

9. ОСОБЕННОСТИ РАБОТЫ С ФАЙЛАМИ БД

БД представляют собой файлы, работа с которыми отличается от работы с файлами других типов. Любые изменения обычного файла проходят под контролем операционной системы и должны быть сохранены перед его закрытием. Если обычный файл закрыть без сохранения изменений, то на диске останется старое содержимое файла.

В СУБД для сохранения информации используется двойной подход. Глобальные операции (изменение структуры БД, создание новых таблиц или других объектов) происходят при сохранении файла БД, и в них участвует операционная система, которая предупреждает пользователя о таких изменениях. Но операции по изменению данных, не затрагивающие структуру БД, максимально автоматизированы, *выполняются без предупреждения* (в обход операционной системы), новые данные сохраняются немедленно автоматически. Следовательно, в отличие от других приложений в СУБД отказаться от выполненных изменений данных невозможно. Поэтому все серьезные изменения производят с копией БД, чтобы в случае необходимости можно было восстановить ее первоначальное состояние.

10. СУБД MICROSOFT ACCESS

Большой популярностью в настоящее время пользуется СУБД *MS Access*, входящая в пакет *MS Office*. Имена файлов БД, начиная с версии *MS Access 2010*, имеют стандартное расширение **.accdb*. *MS Access* позволяет быстро и эффективно обновлять и анализировать данные, осуществлять поиск нужных данных, печатать отчеты, диаграммы и почтовые наклейки. В *MS Access* существует много способов управления: система меню, панели инструментов, контекстные меню, мышь и комбинации «горячих» клавиш.

Вся работа с БД в *MS Access* выполняется через окно контейнера БД, откуда осуществляется доступ ко всем объектам – таблицам, запросам, формам, отчетам. Работа с объектами может осуществляться визуально или программным путем.

Выделяют следующие средства создания основных объектов БД в *MS Access*:

- 1) ручные (разработка объектов в режиме *Конструктора*);
- 2) автоматизированные (разработка с помощью *программ-мастеров*);
- 3) автоматические (средства ускоренной разработки простейших объектов).

MS Access в режиме *Конструктора* объектов БД предоставляет разные бланки табличного вида. Заполнять их удобно, пользуясь скрытыми элементами управления (например, раскрывающимися списками ▾ или построителями выражений ...), для вывода которых выполняют щелчок мышью на соответствующем пустом поле (рис. 8).

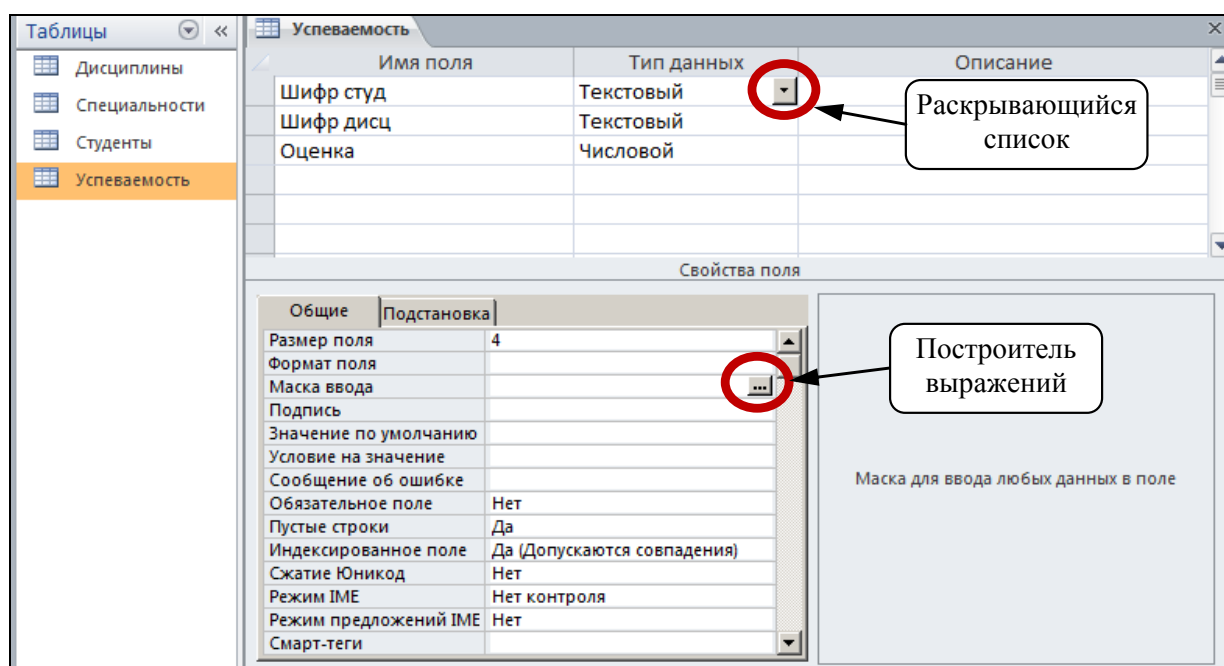


Рис. 8. Пример бланка *MS Access* в режиме *Конструктора таблиц*

На [рис. 9](#) представлен заполненный в режиме *Конструктора запросов* бланк, реализующий условие задачи примера в [разд. 8](#).

Поле:	Группа	Фамилия	Дата рождения	Шифр спец
Имя таблицы:	Студенты	Студенты	Студенты	Студенты
Сортировка:	по возрастанию	по возрастанию		
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Условие отбора:				"230505"

[Рис. 9](#). Пример бланка *MS Access* в режиме *Конструктора запросов*

Основные приемы работы с БД в *MS Access* целесообразно осваивать, непосредственно решая конкретную задачу. С этой целью рекомендуется выполнить лабораторные работы, подробное описание которых представлено в лабораторном практикуме по работе в СУБД *Microsoft Access 2010* [4, 5].

11. КОНТРОЛЬНЫЕ ВОПРОСЫ

- 1) Что называется базой данных?
- 2) Перечислите модели данных и дайте им краткую характеристику.
- 3) Назовите основные объекты базы данных.
- 4) Что включает в себя теоретический этап проектирования базы данных?
- 5) Назовите основные типы межтабличных связей.
- 6) Что в теории баз данных называется транзакцией?
- 7) В чем заключаются особенности работы с файлами БД?

12. ПРИМЕРЫ ТЕСТОВЫХ ВОПРОСОВ

Вопрос № 1 (несколько верных ответов)

Укажите объекты базы данных *MS Access*...

Варианты ответов:

- 1) запросы;
- 2) таблицы;
- 3) ключи;
- 4) формы;
- 5) шаблоны.

Вопрос № 2 (один верный ответ)

Реляционная база данных НЕ может существовать без ...

Варианты ответов:

- 1) модулей;
- 2) отчетов;
- 3) таблиц;
- 4) форм;
- 5) запросов.

Вопрос № 3 (один верный ответ)

Запись в таблице базы данных *MS Access* – это ...

Варианты ответов:

- 1) поле в таблице;
- 2) строка в таблице;
- 3) имя поля;
- 4) маска ввода.

Вопрос № 4 (один верный ответ)

Столбцы таблицы базы данных *MS Access* называются ...

Варианты ответов:

- 1) полями;
- 2) записями;
- 3) структурой;
- 4) формами.

Вопрос № 5 (один верный ответ)

Содержит ли какую-либо информацию таблица базы данных *MS Access*, в которой нет ни одной записи?

Варианты ответов:

- 1) пустая таблица не содержит никакой информации;
- 2) пустая таблица содержит информацию о структуре таблицы;
- 3) пустая таблица содержит информацию о будущих записях;
- 4) таблица без записей существовать не может.

Вопрос № 6 (один верный ответ)

Содержит ли какую-либо информацию таблица базы данных *MS Access*, в которой нет полей?

Варианты ответов:

- 1) содержит информацию о структуре базы данных;
- 2) не содержит никакой информации;
- 3) таблица без полей существовать не может;
- 4) содержит информацию о будущих записях.

Библиографический список

1. Гущин А. Н. Базы данных: Учебно-методическое пособие / А. Н. Гущин. М.: Директ-Медиа, 2015. 311 с.
2. Трофимов В. В. Информатика: В 2 т.: Учебник / В. В. Трофимов. М.: Юрайт, 2019. Т. 1. 553 с.
3. Советов Б. Я. Базы данных: Учебник / Б. Я. Советов, В. В. Цехановский, В. Д. Чертовской. М.: Юрайт, 2019. 420 с.
4. Сидорова Е. А. Лабораторный практикум по работе в СУБД MS Access 2010 / Е. А. Сидорова, Н. А. Насташук, А. В. Долгова / Омский гос. ун-т путей сообщения. Омск, 2015. Ч. 1. 33 с.
5. Сидорова Е. А. Лабораторный практикум по работе в СУБД MS Access 2010 / Е. А. Сидорова, Н. А. Насташук, А. В. Долгова / Омский гос. ун-т путей сообщения. Омск, 2015. Ч. 2. 40 с.

Учебное издание

СИДОРОВА Елена Анатольевна,
ДОЛГОВА Анна Владимировна

ОСНОВЫ БАЗ ДАННЫХ

Учебно-методическое пособие

Редактор Н. А. Майорова

Подписано в печать 10.06.2020. Формат $60 \times 84 \frac{1}{16}$.
Офсетная печать. Бумага офсетная. Усл. печ. л. 1,5. Уч.-изд. л. 1,7.
Тираж 50 экз. Заказ .

**

Редакционно-издательский отдел ОмГУПСа
Типография ОмГУПСа

*

644046, г. Омск, пр. Маркса, 35