
Лабораторная работа № 6

Система управления базами данных MS Access

1. Цель работы

Освоить порядок работы в СУБД Microsoft Access 2013.

2. Общие сведения

Компьютеры позволяют хранить и сортировать большие объёмы информации. Причём есть возможность накопления не только текстовых и графических документов (рисунки, чертежи, фотографии, географические карты), но и страниц глобальной сети, звуковых и видео файлов. Эти возможности реализуются с помощью баз данных.

База данных (БД) — совокупность данных, организованных по определённым правилам, предусматривающим общие принципы описания, хранения и манипулирования данными, которые относятся к конкретной предметной области.

Под **данными** понимается информация, представленная в виде, пригодном для обработки автоматическими средствами при возможном участии человека.

Под **предметной областью** понимается однородная часть реального мира, которая представляет интерес для конкретного исследования.

В качестве примеров простейших БД можно назвать телефонный справочник, расписание движения поездов, сведения о сотрудниках предприятия, каталог товаров и цен на них (прайс-лист), алфавитный или предметный каталог книг в библиотеке.

Главное достоинство электронных БД — возможность быстрого поиска и отбора информации, а также простая генерация (создание) отчёта по заданной форме. Например, по номерам зачёток легко определить фамилии студентов или по фамилии писателя составить список его произведений.

Пользователей баз данных можно разделить на три категории: конечные пользователи (те, кто вводят, извлекают и используют данные), программисты и системные аналитики (те, кто пишут прикладные программы обработки данных, определяют логическую структуру БД) и администраторы.

Администратор базы данных — это специалист, отвечающий за разработку требований к базе данных во время её проектирования, успешную реализацию БД в процессе её создания, эффективное использование и сопровождение БД в процессе эксплуатации. Администратор взаимодействует с конечными пользователями и программистами в процессе проектирования

БД, контролирует её работоспособность, отвечает за реорганизацию и своевременное обновление информации, удаление устаревших данных и за восстановление разрушенных данных, за обеспечение безопасности и целостности данных.

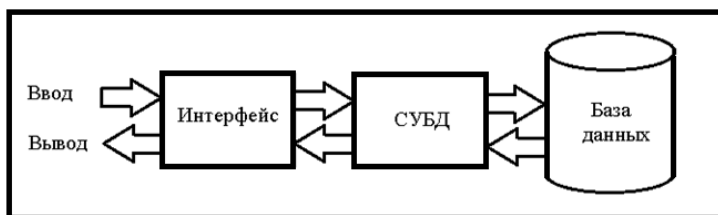
Под **безопасностью** данных понимают защиту данных от случайного или преднамеренного несанкционированного доступа к ним лиц, не имеющих на это права.

Под **целостностью** понимается возможность восстановления данных в случае возникновения сбоев в работе вычислительной техники. Если БД содержит данные, используемые многими пользователями, то очень важно, чтобы данные и связи между ними не разрушались.

Программисты и системные аналитики, создавая БД, стремятся упорядочить информацию по различным признакам (реквизитам, атрибутам), для того чтобы можно было извлекать из БД информацию с необходимым сочетанием признаков.

В современной технологии использования баз данных предполагается, что создание базы данных, её поддержка и обеспечение доступа пользователей к ней осуществляется с помощью специального программного обеспечения — систем управления базами данных.

Системы управления базами данных (СУБД) — программы, которые обеспечивают создание БД и манипуляцию данными. СУБД позволяют вводить, отбирать и редактировать данные. Они предоставляют средства для извлечения данных по определённому критерию (требованию, правилу). СУБД дают возможность конечным пользователям осуществлять непосредственное управление данными, а программистам и системным аналитикам быстро разрабатывать более совершенные программные средства их обработки.



Рассмотрим существующие классификации баз данных.

По технологии обработки данных БД подразделяются на централизованные и распределённые.

Централизованная БД хранится в памяти одной ЭВМ.

Распределённая БД состоит из нескольких частей (возможно, пересекающихся или даже дублирующих друг друга), хранящихся на различных ЭВМ вычислительной сети.

По способу установления связей между данными различают реляционные, иерархические и сетевые БД.

Реляционная БД является простейшей и наиболее привычной формой представления данных в виде таблиц. В теории множеств таблице соответствует термин «отношение» (relation), который и дал название этой БД. Для неё имеется развитый математический аппарат — реляционное исчисление и реляционная алгебра, в которых определены такие математические операции, как объединение, вычитание, пересечение, соединение и др.

Существенный вклад в разработку БД этого типа сделал американский учёный Е. Кодд (Е. Codd).

Достоинством реляционной БД является сравнительная простота инструментальных средств, необходимых для её поддержки, недостатком — жёсткость структуры данных (невозможность, например, задания строк таблицы произвольной длины) и зависимость скорости её работы от размера базы данных. Для многих операций, определённых в такой БД, может оказаться необходимым просмотр всей БД.

Иерархическая БД предполагает наличие связей между данными, имеющими какой-либо общий признак. В иерархической БД такие связи могут быть отражены в виде дерева-графа, где возможны только односторонние связи от старших вершин к младшим. Это ускоряет доступ к необходимой информации, но только если все возможные запросы отражены в структуре дерева. Никакие иные запросы на извлечение информации не будут удовлетворены.

Указанный недостаток снят в **сетевой БД**, в которой (по крайней мере, теоретически) возможно реализовать связи «всех данных со всеми». Поскольку на практике это осуществить невозможно, прибегают к некоторым ограничениям. Использование иерархической и сетевой БД ускоряет доступ к информации в базе данных. Каждый элемент данных должен содержать ссылки на некоторые другие элементы.

Сведения о некоторых СУБД приведены в таблице.

Название СУБД	Тип БД
MS Access	реляционная
Cache	иерархическая
IDS	сетевая
FoxPro	реляционная
IMS	иерархическая
Oracle	реляционная

Рассмотрим основные понятия и компоненты реляционных БД (например, MS Access), которые в настоящее время имеют наибольшее коммерческое использование.

Реляционная БД ориентирована на организацию данных в виде двумерных таблиц-отношений. Любая таблица обладает следующими свойствами:

- каждый элемент таблицы — это один элемент данных;
- все столбцы в таблице однородные, т. е. все элементы в столбце имеют одинаковые длину и тип (числовой, текстовый и т. д.);
- каждый столбец имеет уникальное имя;
- одинаковые строки в таблице отсутствуют;
- порядок следования строк и столбцов может быть произвольным.

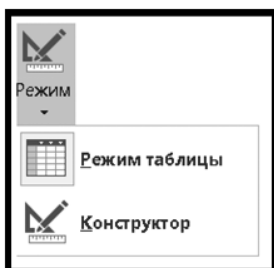
Таблица — это набор данных по конкретной теме (предметной области), например, сведения о студентах высшего учебного заведения. Данные в таблице располагаются в столбцах (**полях**) и строках (**записях**).

					Поле 1	Поле 2	Поле 3
Запись 1							
Запись 2							
Запись 3							

Поле — это элементарная единица логической организации данных, которая соответствует отдельной, неделимой единице информации — **атрибуту**. Каждому полю даётся имя поля (идентификатор поля внутри записи), например, «Фамилия».

Запись — это совокупность логически связанных полей.

В реляционном подходе к построению баз данных используется терминология теории отношений. Столбец таблицы со значениями соответствующего атрибута называется **доменом**, а строка со значениями разных атрибутов — **кортежем**.



Итак, для реляционных БД существует несколько равноправных терминов: столбец может называться полем или доменом, а строка — записью или кортежем.

Рассмотрим основные возможности реляционной БД MS Access на примере небольшой учебной базы данных. Базу данных можно создавать или в режиме «Таблицы», или в режиме «Конструктор».

При создании новой БД нужно указывать, какой тип данных вводится в данное поле: текстовый, числовой, денежный, логический и т.п. Режим «Конструктор» позволяет детально настроить типы данных, которые заносятся в каждом поле.

На следующем рисунке показан процесс записи данных в режиме «Таблицы».

The screenshot shows a table named 'Таблица1' in 'Table View'. The table has four columns: 'Зачётка', 'Фамилия', 'Имя', and 'Щелкните для добавления'. The data entered is as follows:

Зачётка	Фамилия	Имя	Щелкните для добавления
96120	Петров	Юрий	ав Краткий текст
96121			12 Число
96135			Денежный
*	0		Дата и время

После заполнения всех полей будет получена таблица, где указаны сведения о пяти студентах. Таблице дадим имя «Студенты».

The screenshot shows a table named 'Студенты' in 'Table View'. The table has six columns: 'Зачётка', 'Фамилия', 'Имя', 'Отчество', 'Пол', and 'Группа'. The data entered is as follows:

Зачётка	Фамилия	Имя	Отчество	Пол	Группа
96120	Петров	Юрий	Александрович	м	БТ-61
96121	Галкина	Ангелика	Юрьевна	ж	БТ-63
96135	Иванов	Андрей	Сергеевич	м	БТ-63
96136	Козлова	Елена	Васильевна	ж	БТ-62
96137	Сидорова	Светлана	Николаевна	ж	БТ-61

В режиме конструктора эта таблица выглядит так:

The screenshot shows the 'Table Design View' for the 'Студенты' table. The columns and their data types are as follows:

Имя поля	Тип данных
Зачётка	Числовой
Фамилия	Короткий текст
Имя	Короткий текст
Отчество	Короткий текст
Пол	Короткий текст
Группа	Короткий текст

Аналогично можно создать таблицу «Сессия», в которой содержатся оценки студентов по четырём дисциплинам. В поле «Результат» помещены средние значения оценок студентов по результатам сдачи четырёх экзаменов.

Зачётка	Математике	Физика	Информатика	Графика	Результат
96120	4	5	4	5	4,5
96121	4	4	5	4	4,25
96135	5	5	5	5	5
96136	5	4	5	5	4,75
96137	4	5	4	4	4,25

Из примеров видно, что каждое поле имеет уникальное (единственное в данной таблице) имя. Все поля таблицы «Сессия» имеют числовой тип данных.

Каждая запись должна однозначно идентифицироваться (определяться) уникальным **ключом записи**. В общем случае ключи записи бывают двух видов: первичный (уникальный) и вторичный.

Первичный ключ — это одно или несколько полей, однозначно идентифицирующих запись. Если первичный ключ состоит из одного поля, он называется **простым**, если из нескольких полей — **составным** ключом.

В приведённых выше таблицах простым первичным ключом является атрибут «Зачётка». В таблице «Студенты» можно было попытаться использовать в качестве простого первичного ключа атрибут «Фамилия». Однако не

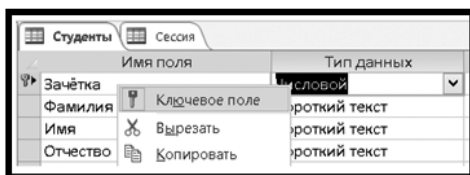
исключена возможность существования однофамильцев среди студентов. В этом случае атрибут «Фамилия» не сможет играть роль ключа, однозначно определяющего каждую запись. В качестве ключей часто используют инвентарные, табельные

номера, электронные адреса, номера ICQ, паспортные номера и серии или просто порядковые номера записей.

Ключевое поле задаётся с помощью контекстного меню (щелчок правой кнопкой по выбранному полю в режиме «Конструктор»).

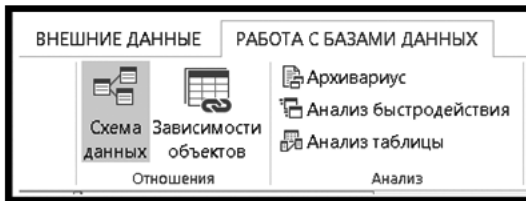
Вторичный ключ — это такое поле, значение которого может повторяться в нескольких записях, т. е. он не является уникальным. Если по значению первичного ключа может быть найден один-единственный экземпляр записи, то по вторичному ключу — несколько записей.

В реляционной БД содержится, как правило, несколько таблиц с различными сведениями. Разработчик БД может установить связи между отдельными таблицами. При создании связей используют ключевые поля. После установления связей появляется возможность создания запросов, форм и отчётов, в которые помещаются данные из нескольких связанных между собой таблиц.



Пусть в создаваемой базе данных имеется ещё одна таблица с названием «Стипендия», с помощью которой начисляется стипендия в зависимости от среднего балла за сессию (в процентах от максимальной стипендии). Ключевым полем в этой таблице является столбец с названием «Код».

Код	Результат	Процент
1	5	100
2	4,75	75
3	4,5	50
4	4,25	0

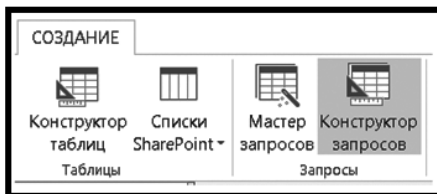


Следующие два рисунка иллюстрируют процесс создания связей между этими тремя таблицами. Вначале нужно на ленте (закладка «Работа с базами данных») выбрать пункт «Схема данных». Затем в открывшемся окне

расположить таблицы и установить связи между одноимёнными полями.



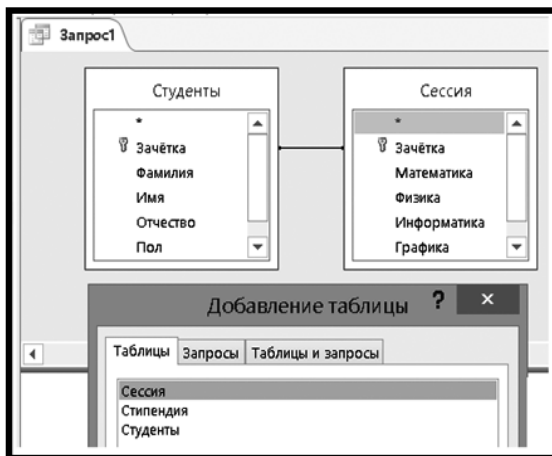
Для отбора данных из БД, удовлетворяющих определённым условиям, создаётся запрос.



Запрос — это инструкция (команда) для манипуляции данными в соответствии с определёнными условиями (критериями). Под манипуляцией понимаются операции выборки необходимой информации, изменения, удаления и добавления данных.

Запросы создаются с по-

мощью «Конструктора запросов» или «Мастера запросов». Например, на вкладке «Создание» можно выбрать пункт «Конструктор запросов» и с помощью диалогового окна «Добавить таблицы» выбрать нужные таблицы и заполнить бланк запроса.



На следующем рисунке показана форма (бланк) запроса-выборки, предназначенного для отбора из созданной БД оценок по математике у студентов группы БТ-61, а на следующем рисунке — результаты сделанной выборки.

Поле:	Фамилия	Группа	Математика
Имя таблицы:	Студенты	Студенты	Сессия
Сортировка:			
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:		"БТ-61"	
или:			

БТ-61_Математика		
Фамилия	Группа	Математике
Петров	БТ-61	4
Сидорова	БТ-61	4

Запрос можно формировать с использованием логических (булевых) операций И (AND), ИЛИ (OR), НЕ (NOT). Например, если требуется выбрать из БД сведения о результатах сдачи экзамена по математике студентами групп БТ-61 и БТ-62, то необходимо запрос изменить следующим образом:

Поле:	Фамилия	Группа	Математика
Имя таблицы:	Студенты	Студенты	Сессия
Сортировка:			
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:		"БТ-61"	
или:		"БТ-62"	

В этом случае из БД будут отобраны данные с помощью логической операции ИЛИ и на экране появятся сведения о студентах двух групп — БТ - 61 и БТ - 62.

Фамилия	Группа	Математика
Петров	БТ-61	4
Козлова	БТ-62	5
Сидорова	БТ-61	4

Логическая операция И используется для решения следующей задачи. Пусть требуется выбрать из БД фамилии студентов группы БТ-63, сдавших математику с оценкой 5. На следующем рисунке показано,

как формируется запрос с использованием логической операции И.

Поле:	Фамилия	Группа	Математика
Имя таблицы:	Студенты	Студенты	Сессия
Сортировка:			
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:		"БТ-63"	5

Результаты отбора приведены на следующем рисунке. Рассматриваемые примеры умышленно выбраны простыми, для того чтобы можно было проверить полученный результат даже без использования ЭВМ.

Фамилия	Группа	Математика
Иванов	БТ-63	5

Таблица с результатами запроса может использоваться при дальнейшей обработке данных. В запросе на выборку могут использоваться не только таблицы БД, но и таблицы, полученные в результате выполненных запросов.

В запросах можно производить вычисления. Например, для подсчёта среднего балла нужно при формировании запроса вначале просуммировать оценки по четырём предметам, а затем результат разделить на четыре.

Поле:	Фамилия	Средняя_оценка: ((Математика)+[Физика]+[Информатика]+[Графика])/4
Имя таблицы:	Студенты	
Групповая операция:	Группировка	Группировка
Сортировка:		
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Чтобы осуществить отбор записей из базы данных по фамилии (или по имени), нужно в качестве условия отбора использовать запись типа: Like("Фамилия"):

Поле:	Фамилия	Имя	Группа	Информатика
Имя таблицы:	Студенты	Студенты	Студенты	Сессия
Сортировка:				
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:	Like ("Петров")			
или:		Like ("Светлана")		

Форма позволяет отобразить данные из одной или нескольких таблиц и вывести их на экран, используя стандартный или созданный пользователем макет. При этом формы могут воссоздавать привычные для конечного пользователя документы. Формы используются не только для вывода данных из БД, но также (и, пожалуй, чаще) и для ввода данных. На рисунке представлена форма, позволяющая установить средний балл каждого студента (см. поле «Результат»). Содержание формы изменяется пользователем в зависимости от стоящей перед ним задачи.

Фамилия	<input type="text" value="Петров"/>
Зачётка	<input type="text" value="96120"/>
Группа	<input type="text" value="БТ-61"/>
Результат	<input type="text" value="4,5"/>

Отчёт содержит ту информацию, извлечённую из БД, которая должна быть представлена в виде итогового документа. Обычно отчёт представляется в напечатанном на бумаге виде (в отличие от таблиц, запросов и форм, которые чаще всего отображаются лишь на экране дисплея).

3. Задания на выполнение лабораторной работы

Создать базу данных, состоящую из двух таблиц: **Студенты** и **Сессия**. Исходные данные взять из таблиц 1 и 2.

Таблица 1. Студенты

Зачетк	Фамилия	Имя	Отчество	Пол	Группа
987651	Иванов	Иван	Иванович	М	1-Э-3
987652	Сидорова	Исидора	Сидоровна	Ж	1-Э-3
987653	Петров	Петр	Петрович	М	1-Э-2
987654	Галкина	Галина	Геннадьевна	Ж	1-Э-1
987655	Александров	Александр	Александрович	М	1-Э-1
987656	Воробьева	Вероника	Витальевна	Ж	1-Э-2
987657	Сергеев	Сергей	Сергеевич	М	1-Э-3
987658	Сорокина	Светлана	Семеновна	Ж	1-Э-2
987659	Григорьев	Григорий	Григорьевич	М	1-Э-1
987660	Орлова	Ольга	Викторовна	Ж	1-Э-1

Таблица 2. Сессия

Зачетка	Фамилия	Физика	Математика	Информати	Философия
987651	Иванов	4	4	4	4
987652	Сидорова	5	5	5	5
987653	Петров	3	3	2	3
987654	Галкина	5	4	5	4
987655	Александров	5	3	5	3
987656	Воробьева	4	3	4	2
987657	Сергеев	3	3	4	4
987658	Сорокина	4	4	3	3
987659	Григорьев	2	4	3	4
987660	Орлова	4	2	5	3

Сформировать запросы на отбор информации из базы данных в соответствии с номером варианта (табл.3).

Задания 1, 2, 3 и 23 одинаковые для всех вариантов.

В заданиях 24...33 нужно связать между собой две таблицы и запрос «Средний балл».

Таблица 3

Варианты	Задания
1	1, 2, 3, 4, 7, 9, 13, 17, 21, 23, 24, 28, 32, 34, 36
2	1, 2, 3, 5, 8, 10, 14, 18, 22, 23, 25, 29, 33, 35, 37
3	1, 2, 3, 6, 7, 11, 15, 19, 21, 23, 26, 30, 32, 34, 36
4	1, 2, 3, 4, 8, 12, 16, 20, 22, 23, 27, 31, 33, 35, 37
5	1, 2, 3, 5, 7, 9, 13, 17, 21, 23, 24, 28, 32, 34, 36
6	1, 2, 3, 6, 8, 10, 14, 18, 22, 23, 25, 29, 33, 35, 37
7	1, 2, 3, 4, 7, 11, 15, 19, 21, 23, 26, 30, 32, 34, 36
8	1, 2, 3, 5, 8, 12, 16, 20, 22, 23, 27, 31, 33, 35, 37
9	1, 2, 3, 6, 7, 9, 13, 17, 21, 23, 24, 28, 32, 34, 36
10	1, 2, 3, 4, 8, 10, 14, 18, 22, 23, 25, 29, 33, 35, 37
11	1, 2, 3, 5, 7, 11, 15, 19, 21, 23, 26, 30, 32, 34, 36
12	1, 2, 3, 6, 8, 12, 16, 20, 22, 23, 27, 31, 33, 35, 37
13	1, 2, 3, 4, 7, 9, 13, 17, 21, 23, 24, 28, 32, 34, 36
14	1, 2, 3, 5, 8, 10, 14, 18, 22, 23, 25, 29, 33, 35, 37
15	1, 2, 3, 6, 7, 8, 14, 18, 22, 23, 25, 29, 33, 34, 36
16	1, 2, 3, 6, 8, 13, 17, 21, 23, 24, 28, 31, 33, 34, 37

3.1. Задание 1. Создание запроса «Студенты группы 1-Э-1»

Используя таблицу **Студенты**, создать запрос, по которому из базы данных будут отобраны фамилии и имена студентов группы 1-Э-1. В результатах выборки должен быть обязательно указан номер студенческой группы.

3.2. Задание 2. Создание запроса «Фамилии студентов»

Используя таблицу **Студенты**, создать запрос, по которому из базы данных будут отобраны фамилии и имена мужчин всех групп. В результатах должен быть указан пол студентов.

3.3. Задание 3. Создание запроса «Студентки группы 1-Э-2»

Используя таблицу **Студенты**, создать запрос, по которому из базы данных будут отобраны фамилии и имена студенток (женщин) группы 1-Э-2.

3.4. Задание 4. Создание запроса «Оценки по физике студентов группы 1-Э-1»

Используя связанные таблицы **Студенты** и **Сессия**, создать запрос, по которому из базы данных будут отобраны фамилии, имена, номера зачёток и оценки по физике всех студентов группы 1-Э-1 (мужчин и женщин).

3.5. Задание 5. Создание запроса «Оценки по математике студентов группы 1-Э-2»

Используя связанные таблицы **Студенты** и **Сессия**, создать запрос, по которому из базы данных будут отобраны фамилии, имена, номера зачёток и оценки по математике студентов группы 1-Э-2.

3.6. Задание 6. Создание запроса «Оценки по информатике студентов группы 1-Э-3»

Используя связанные таблицы **Студенты** и **Сессия**, создать запрос, по которому из базы данных будут отобраны фамилии, имена, номера зачёток и оценки по информатике студентов группы 1-Э-3.

3.7. Задание 7. Создание запроса «Оценки по философии студенток группы 1-Э-1»

Используя связанные таблицы **Студенты** и **Сессия**, создать запрос, по которому из базы данных будут отобраны фамилии, имена, номера зачёток и оценки по философии студенток (женщин) группы 1-Э-1.

3.8. Задание 8. Создание запроса «Оценки по философии студентов группы 1-Э-2»

Используя связанные таблицы **Студенты** и **Сессия**, создать запрос, по которому из базы данных будут отобраны фамилии, имена, номера зачёток и оценки по философии студентов (мужчин) группы 1-Э-2.

3.9. Задание 9. Создание запроса «Оценка «хорошо» по математике»

Используя связанные таблицы **Студенты** и **Сессия**, создать запрос, по

которому из базы данных будут отобраны фамилии, имена, номера зачёток студентов, получивших оценку хорошо (4) по математике.

3.10. Задание 10. Создание запроса «Оценка «удовлетворительно» по философии»

Используя связанные таблицы **Студенты** и **Сессия**, создать запрос, по которому из базы данных будут отобраны фамилии, имена, номера зачёток студентов, получивших оценку удовлетворительно (3) по философии.

3.11. Задание 11. Создание запроса «Оценка «отлично» по информатике»

Используя связанные таблицы **Студенты** и **Сессия**, создать запрос, по которому из базы данных будут отобраны фамилии, имена, номера зачёток студентов, получивших оценку отлично (5) по информатике.

3.12. Задание 12. Создание запроса «Оценка «неудовлетворительно» по физике»

Используя связанные таблицы **Студенты** и **Сессия**, создать запрос, по которому из базы данных будут отобраны фамилии, имена, номера зачёток студентов, получивших оценку неудовлетворительно (2) по физике.

3.13. Задание 13. Создание запроса «Оценка «удовлетворительно» по физике и математике»

Используя связанные таблицы **Студенты** и **Сессия**, создать запрос, по которому из базы данных будут отобраны фамилии, имена, номера зачёток студентов, получивших оценку удовлетворительно (3) по двум предметам: физике и математике.

3.14. Задание 14. Создание запроса «Оценка «хорошо» по философии и информатике»

Используя связанные таблицы **Студенты** и **Сессия**, создать запрос, по которому из базы данных будут отобраны фамилии, имена, номера зачёток студентов, получивших оценку хорошо (4) одновременно по двум предметам: философии и информатике.

3.15. Задание 15. Создание запроса «Оценка «отлично» по физике и информатике»

Используя связанные таблицы **Студенты** и **Сессия**, создать запрос, по которому из базы данных будут отобраны фамилии, имена, номера зачёток студентов, получивших оценку отлично (5) одновременно по двум предметам: физике и информатике.

3.16. Задание 16. Создание запроса «Оценка «хорошо» по физике и философии»

Используя связанные таблицы **Студенты** и **Сессия**, создать запрос, по которому из базы данных будут отобраны фамилии, имена, номера зачёток студентов, получивших оценку хорошо (4) по двум предметам: и по физике, и по философии.

3.17. Задание 17. Создание запроса «Оценка «неудовлетворительно» по физике или философии»

Используя связанные таблицы **Студенты** и **Сессия**, создать запрос, по которому из базы данных будут отобраны фамилии, имена, номера зачёток студентов, получивших оценку неудовлетворительно (2) по одному из двух предметов: по физике или философии.

3.18. Задание 18. Создание запроса «Оценка «неудовлетворительно» по математике или информатике»

Используя связанные таблицы **Студенты** и **Сессия**, создать запрос, по которому из базы данных будут отобраны фамилии, имена, номера зачёток студентов, получивших оценку неудовлетворительно (2) по одному из двух предметов: по математике или информатике.

3.19. Задание 19. Создание запроса «Оценка «неудовлетворительно» по физике или информатике»

Используя связанные таблицы **Студенты** и **Сессия**, создать запрос, по которому из базы данных будут отобраны фамилии, имена, номера зачёток студентов, получивших оценку неудовлетворительно (2) по одному из двух предметов: по физике или информатике.

3.20. Задание 20. Создание запроса «Оценка «неудовлетворительно» по математике или философии»

Используя связанные таблицы **Студенты** и **Сессия**, создать запрос, по которому из базы данных будут отобраны фамилии, имена, номера зачёток студентов, получивших оценку неудовлетворительно (2) по одному из двух предметов: по математике или философии.

3.21. Задание 21. Создание запроса «Оценка «отлично» по всем предметам»

Используя связанные таблицы **Студенты** и **Сессия**, создать запрос, по которому из базы данных будут отобраны фамилии, имена, номера зачёток студентов, получивших отличные (5) оценки по всем предметам.

3.22. Задание 22. Создание запроса «Оценка «хорошо» по всем предметам»

Используя связанные таблицы **Студенты** и **Сессия**, создать запрос, по которому из базы данных будут отобраны фамилии, имена, номера зачёток студентов, получивших хорошие (4) оценки по всем предметам.

3.23. Задание 23. Создание запроса «Средний балл»

Используя таблицу **Сессия**, создать запрос с именем **Средний балл** для расчёта среднего балла каждого студента по результатам сдачи четырёх экзаменов. Запрос обязательно должен содержать поле **Зачётка**, которое впоследствии будет использовано для связывания нескольких таблиц.

3.24. Задание 24. Создание запроса «Средний балл 4»

Используя связанные таблицы **Студенты**, **Сессия** и запрос **Средний балл**, создать запрос, по которому из базы данных будут отобраны фамилии, имена, номера зачёток, номера групп студентов, имеющих средний балл 4.

3.25. Задание 25. Создание запроса «Средний балл 3,25»

Используя связанные таблицы **Студенты**, **Сессия** и запрос **Средний балл**, создать запрос, по которому из базы данных будут отобраны фамилии, имена, номера зачёток, номера групп студентов, имеющих средний балл 3,25.

3.26. Задание 26. Создание запроса «Средний балл 3,5»

Используя связанные таблицы **Студенты**, **Сессия** и запрос **Средний балл**, создать запрос, по которому из базы данных будут отобраны фамилии, имена, номера зачёток, номера групп студентов, имеющих средний балл 3,5.

3.27. Задание 27. Создание запроса «Средний балл 2,75»

Используя связанные таблицы **Студенты**, **Сессия** и запрос **Средний балл**, создать запрос, по которому из базы данных будут отобраны фамилии, имена, номера зачёток, номера групп студентов, имеющих средний балл 2,75.

3.28. Задание 28. Создание запроса «Успеваемость студента Петрова»

Используя связанные таблицы **Студенты**, **Сессия** и запрос **Средний балл**, создать запрос, по которому из базы данных будут отобраны: оценка по физике, средний балл и номер группы студента Петрова.

3.29. Задание 29. Создание запроса «Успеваемость студента Иванова»

Используя связанные таблицы **Студенты**, **Сессия** и запрос **Средний балл**, создать запрос, по которому из базы данных будут отобраны оценка по математике, средний балл и номер группы студента Иванова.

3.30. Задание 30. Создание запроса «Успеваемость студентки Орловой»

Используя связанные таблицы **Студенты**, **Сессия** и запрос **Средний балл**, создать запрос, по которому из базы данных будут отобраны оценка по информатике, средний балл и номер группы студентки Орловой.

3.31. Задание 31. Создание запроса «Успеваемость студентки Сидоровой»

Используя связанные таблицы **Студенты**, **Сессия** и запрос **Средний балл**, создать запрос, по которому из базы данных будут отобраны оценка по философии, средний балл и номер группы студентки Сидоровой.

3.32. Задание 32. Создание запроса «Средний балл более 4»

Используя связанные таблицы **Студенты**, **Сессия** и запрос **Средний балл**, создать запрос, по которому из базы данных будут отобраны фамилии, имена студентов, имеющих средний балл более 4.

3.33. Задание 33. Создание запроса «Средний балл менее 3,75»

Используя связанные таблицы **Студенты**, **Сессия** и запрос **Средний балл**, создать запрос, по которому из базы данных будут отобраны фамилии, имена студентов, имеющих средний балл менее 3,75.

3.34. Задание 34. Создание запроса «Отчество Иванович»

Используя таблицу **Студенты** определить фамилию, имя и номер зачётки студента, если известно, что его отчество Иванович.

3.35. Задание 35. Создание запроса «Отчество Викторовна»

Используя таблицу **Студенты** определить фамилию, имя и номер зачётки студентки, если известно, что её отчество Викторовна.

3.36. Задание 36. Создание запроса на удаление «Зачётка 987660»

Удалить с помощью запроса из таблицы **Сессия** запись о студенте, у которого номер зачётки 987660.

3.37. Задание 37. Создание запроса на удаление «Фамилия Орлова»


Удалить из таблицы **Сессия** запись о студентке по фамилии Орлова.

4. Порядок выполнения лабораторной работы

С помощью конструктора нужно создать заготовку для таблицы **Студенты**, указав имена полей и типы используемых данных в соответствии с таблицей 1. Следует обратить внимание на то, что в каждой таблице существует ключевое поле, с помощью которого можно отличить одну запись от других. Ключевое поле выделяется с помощью специального значка (ключа), расположенного слева от имени поля (см. поле **Зачётка**).

Чтобы установить этот значок нужно выделить строку и использовать соответствующую кнопку на панели инструментов. Эту же операцию можно выполнить с помощью контекстного меню, которое вызывается щелчком правой кнопки мыши по имени выбранного поля.

Таблица 4. Студенты

	Имя поля	Тип данных
	Зачетка	Числовой
	Фамилия	Короткий текст
	Имя	Короткий текст
	Отчество	Короткий текст
	Пол	Короткий текст
	Группа	Короткий текст

Свойства типа данных можно изменить с помощью списка, который вызывается щелчком мыши в поле **Тип данных**.

Таблица 5

Имя поля	Тип данных
Зачетка	Числовой
Фамилия	Короткий текст
Имя	Длинный текст
Отчество	Числовой
Пол	Дата и время
Группа	Денежный
	Счетчик
	Логический

Созданную в конструкторе заготовку нужно заполнить в соответствии с таблицей 1.

Для создания таблицы **Сессия** нужно с помощью конструктора сформировать заголовки шести полей (см. табл.6).

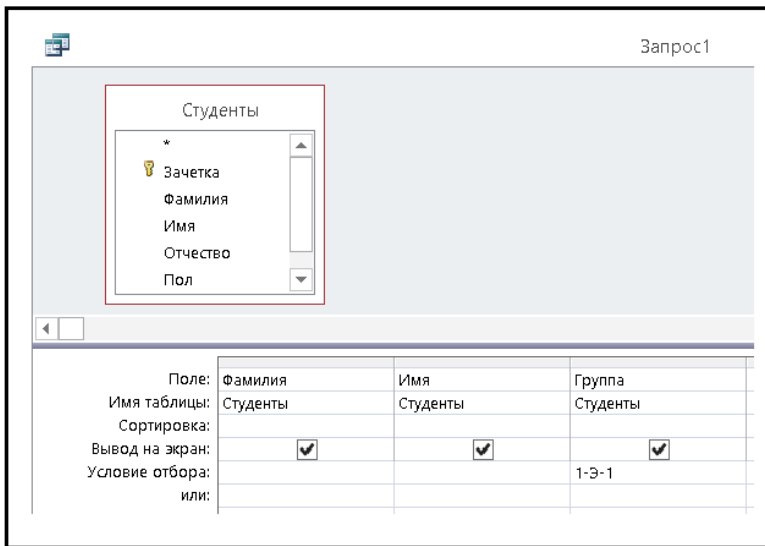
Таблица 6

Имя поля	Тип данных
Зачетка	Числовой
Фамилия	Короткий текст
Физика	Числовой
Математика	Числовой
Информатика	Числовой
Философия	Числовой

Новую таблицу **Сессия** следует заполнить в соответствии с табл. 2.

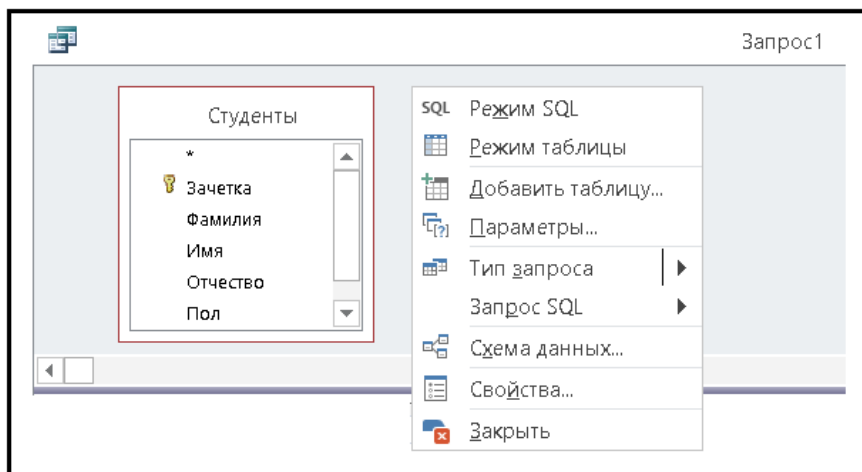
4.1. Методические указания к п. 3.1

Для формирования запроса следует выбрать на вкладке **СОЗДАНИЕ** команду **Конструктор запросов** и, используя таблицу **Студенты**, заполнить форму в соответствии со следующим рисунком.



Следует обратить внимание на то, что при формировании запросов ключевые слова должны быть записаны так же, как и в базе данных (лишние пробелы и грамматические ошибки недопустимы). При заполнении базы данных конкретными значениями нельзя вводить дополнительные пробелы, так как они воспринимаются приложением как отдельные символы.

Чтобы сохранить созданный запрос нужно с помощью контекстного меню закрыть запрос и в момент закрытия дать ему необходимое имя. Целесообразно имена давать так, чтобы номера, входящие в имя, совпадали с номерами заданий. Например, Запрос1.



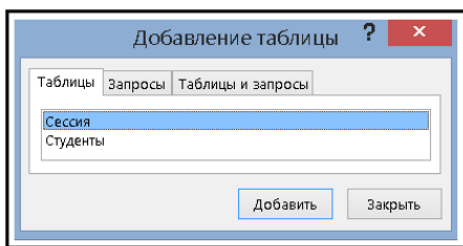
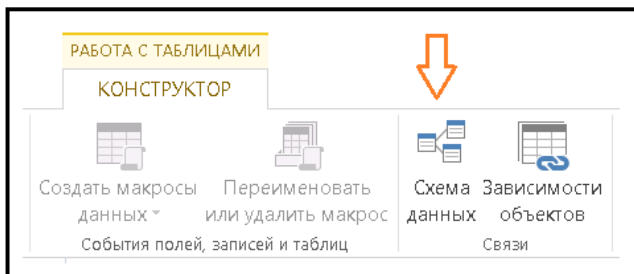
Если созданный запрос запустить на выполнение, то будет получен результат, показанный на рисунке.

Фамилия	Имя	Группа
Галкина	Галина	1-Э-1
Александров	Александр	1-Э-1
Григорьев	Григорий	1-Э-1
Орлова	Ольга	1-Э-1

Аналогично выполняются задания 2 и 3. В случае появления затруднений в выполнении заданий следует использовать теоретический материал, приведенный в разделе **Общие сведения**.

4.2. Методические указания к п. 3.4

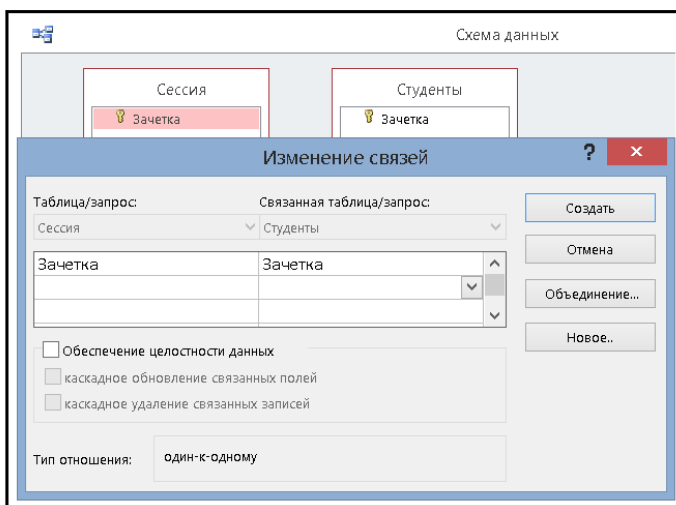
Чтобы связать таблицы нужно на вкладке **КОНСТРУКТОР** выбрать команду **Схема данных**.



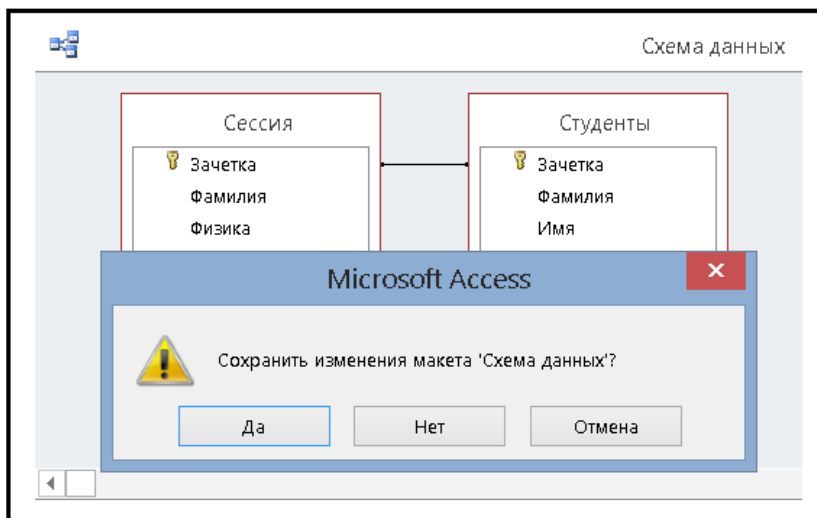
С помощью диалогового окна **Добавление таблицы** разместить в окне **Схема данных** нужные таблицы, например, **Студенты** и **Сессия**. Для этого следует использовать кнопку **Добавить**.

Для связывания этих двух таблиц нужно использовать

ключевое поле **Зачётка**.



Сделанные изменения макета **Схема данных** следует сохранить.



4.3. Методические указания к п. 3.23

Для подсчёта средних баллов нужно при формировании запроса вначале просуммировать оценки по четырём предметам, а затем результат разделить на четыре.

В СУБД MS Access это делается так. В очередном свободном заголовке поля делается следующая запись:

Средний балл:

$$=((\text{Математика})+(\text{Физика})+(\text{Информатика})+(\text{Философия}))/4$$

Пример выполнения расчётов с помощью MS Access показан на следующем рисунке. В данном случае поле получит имя **Средний балл**.

Поле:	Фамилия	Средняя_оценка: ((Математика)+[Физика]+[Информатика]+[Графика])/4
Имя таблицы:	Студенты	
Групповая операция:	Группировка	Группировка
Сортировка:		
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

4.4. Методические указания к п.п. 3.34 и 3.35

Чтобы осуществить отбор записей из базы данных по фамилии (по имени или по отчеству) нужно в качестве условия отбора использовать запись:

Like *Фрагмент строки*.

Например, чтобы найти сведения о студенте по фамилии Александров достаточно сформировать запрос, у которого в поле **Фамилия** имеется ко-

манда: Like *Алекс*. Понятно, что результат не изменится если вместо фрагмента фамилии будет полностью записана фамилия.

Если сформировать запрос так, как показано на рисунке, то после выполнения запроса будут выведены сведения о студентах Александрове и Петрове.

Поле:	Фамилия	Имя
Имя таблицы:	Сессия	Студенты
Сортировка:		
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:		Like "*Алекс*"
или:	Like ""*Петр*"	

Фамилия	Имя	Отчество
Петров	Петр	Петрович
Александров	Александр	Александрович

4.5. Методические указания к п.п. 3.36 и 3.37

Запрос на удалении данных из базы данных безвозвратно уничтожает хранящуюся информацию. Целесообразно перед выполнением этого задания сделать копию созданной БД. Формирование запроса осуществляется стандартно: на вкладке **СОЗДАНИЕ** выбирается команда **Конструктор запросов**. Затем добавляется таблица **Сессия**. На вкладке **КОНСТРУКТОР** выбирается команда **Удаление**.

Access

ВНЕШНИЕ ДАННЫЕ РАБОТА С БАЗАМИ ДАННЫХ

+ Добавление

↻ Обновление

✖ Удаление

⊞ Объединение

⊞ Перекрестный

⊞ К серверу

⊞ Управление

Тип запроса

Тип запроса: удаление

Создание запроса, который удаляет данные, соответствующие указанным условиям, из существующей таблицы.

РАБОТА С ЗАПРОСАМИ

КОНСТРУКТОР

+ Отобразить таблицу

Поле:	Математика
Имя таблицы:	Сессия
Удаление:	Условие
Условие отбора:	2
или:	

Далее выбирается необходимое поле и указывается соответствующее условие. На рисунке показан

запрос, предназначенный для удаления записи из таблицы **Сессия**, которая содержит неудовлетворительную оценку по дисциплине **Математика**.

4. Требования к отчёту

Отчёт оформляется в электронном виде. Он должен содержать запросы, сформированные в соответствии с номером варианта.

6. Контрольные вопросы

- 6.1. Дайте определение термину «база данных».
- 6.2. Что такое данные?
- 6.3. Как классифицируются базы данных по способу установления связей между данными?
- 6.4. Что такое СУБД?
- 6.5. Как по-другому называется строка реляционной БД?
- 6.6. Какие синонимы существуют для термина «столбец»?
- 6.7. Чем отличаются централизованная и распределённая базы данных?
- 6.8. Что называется запросом?
- 6.9. Что такое первичный ключ?
- 6.10. Какие логические операции используются при отборе данных?
- 6.11. Как в запросе выполнить вычисления?
- 6.12. Как нужно сформировать условие отбора, чтобы из реляционной БД выбрать информацию по известной фамилии человека?
- 6.13. Для чего используется форма?

7. Список литературы

1. Алексеев А.П. Информатика 2015 [Текст]: учеб. пособие/ Алексеев А. П. – М: СОЛОН-Пресс, 2015. – 400 с. ISBN 978-5-91359-158-6
2. Алексеев А.П. Информатика 2007 [Текст]: учеб. пособие для вузов/ Алексеев А. П. – М.: СОЛОН-ПРЕСС, 2007. - 608 с.- (Библиотека студента).
3. Макарова Н.В., Волков В.Б. Информатика: Учебник для вузов. – СПб.: Питер, 2011. – 576 с.