

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

по выполнению лабораторных работ
по дисциплине «Базы данных»
направление 38.03.05 «Бизнес-информатика»
направленность (профиль) «Информационная бизнес-аналитика и цифровые
технологии»

Введение

Самостоятельная работа - планируемая учебная, учебно-исследовательская, научно-исследовательская работа студентов, выполняемая во внеаудиторное (аудиторное) время по заданию и при методическом руководстве преподавателя, но без его непосредственного участия (при частичном непосредственном участии преподавателя, оставляющем ведущую роль за работой студентов).

Самостоятельная работа студентов в ВУЗе является важным видом учебной и научной деятельности студента. Самостоятельная работа студентов играет значительную роль в рейтинговой технологии обучения.

Целью данной дисциплины является формирование набора (ОПК-3 и ОПК-5) компетенций будущего бакалавра по направлению подготовки 38.03.05 «Бизнес-информатика».

Задачами дисциплины «Базы данных» являются: систематизация и закрепление полученных теоретических знаний и практических умений студентов; углубление и расширение теоретических знаний; формирование умений использовать нормативную, правовую, справочную документацию и специальную литературу; развитие познавательных способностей и активности студентов: творческой инициативы, самостоятельности, ответственности и организованности; формирование самостоятельности мышления, способностей к саморазвитию, самосовершенствованию и самореализации; развитие исследовательских умений; использование материала, собранного и полученного в ходе самостоятельных занятий на семинарах, на практических и лабораторных занятиях, при написании курсовых и выпускной квалификационной работ, для эффективной подготовки к итоговым зачетам и экзаменам.

Дисциплина «Базы данных» входит в Блок 1 обязательных дисциплин учебного плана 2026 года по направлению 38.03.05 «Бизнес информатика». Ее освоение происходит в 4 семестре.

Содержание

Лабораторная работа № 1.....	4
Лабораторная работа № 2.....	8
Лабораторная работа № 3.....	17
Лабораторная работа № 4.....	20
Лабораторная работа № 5.....	24
Лабораторная работа № 6.....	27
Лабораторная работа №8.....	31
Лабораторная работа №9.....	36
Лабораторная работа №10.....	39
Лабораторная работа №11.....	43
Лабораторная работа №12.....	47
Лабораторная работа №13.....	50
Лабораторная работа №14.....	54
Лабораторная работа №15.....	58
Лабораторная работа №16.....	66

Лабораторная работа № 1. Знакомство с СУБД

Формируемые компетенции или их части

№ п/п	Содержание компетенции	Шифр
1	Способен управлять процессами создания и использования продуктов и услуг в сфере информационно-коммуникационных технологий, в том числе разрабатывать алгоритмы и программы для их практической реализации	ОПК-3
2	Способен организовывать взаимодействие с клиентами и партнерами в процессе решения задач управления жизненным циклом информационных систем и информационно-коммуникационных технологий	ОПК-5

Теоретическая часть

Базы данных — это совокупность сведений (о реальных объектах, процессах, событиях или явлениях), относящихся к определенной теме или задаче, организованная таким образом, чтобы обеспечить удобное представление этой совокупности, как в целом, так и любой ее части.

Почти все современные системы основаны на реляционной (relational) модели управления базами данных. **Реляционная база данных** представляет собой множество взаимосвязанных таблиц, каждая из которых содержит информацию об объектах определенного типа. Название «реляционная» связано с тем, что каждая запись содержит информацию, относящуюся только к одному объекту. В таких базах данные не дублируются, а связываются по определенным полям.

Можно выделить три основные функции СУБД:

- **определение данных** (Datadefinition) - вы можете определить, какая именно информация будет храниться в вашей базе данных, задать структуру данных и их тип (например, максимальное количество цифр или символов), а также указать, как эти данные связаны между собой. В некоторых случаях вы можете также задать форматы и критерии проверки данных;

- **обработка данных** (Datamanipulation) - данные можно обрабатывать самыми различными способами. Можно объединять данные с другой связанной с ними информацией и вычислять итоговые значения;

- **управление данными** (Datacontrol) - вы можете указать, кому разрешено знакомиться с данными, корректировать их или добавлять новую информацию. Можно также определить правила коллективного пользования данными.

Система управления базами данных MicrosoftAccess является одним из самых популярных приложений в семействе настольных СУБД.

Запуск MicrosoftAccess. Создание базы данных

Для того, чтобы запустить MicrosoftAccess необходимо:

1. Нажать кнопку *Пуск* на Панели задач в нижней части рабочего стола.
2. Открыть в главном меню пункт *Программы*.
3. Выбрать программу *MicrosoftAccess*.

Выбрать пустой шаблон действие 1 задать имя файла действие 2 и нажать кнопку создать.

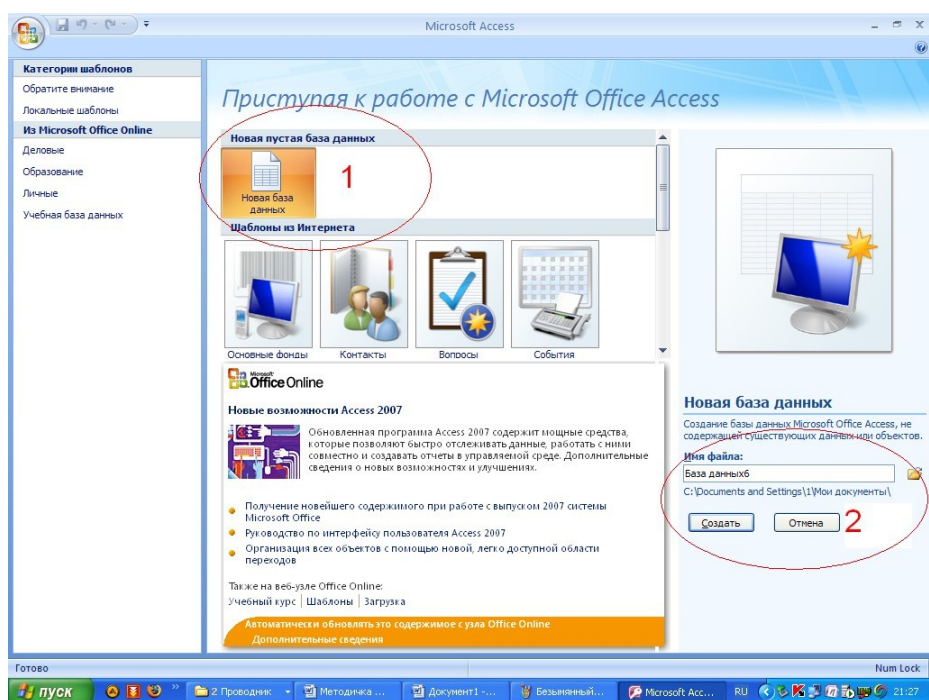


Рисунок 1 - Создание базы данных

Под *базой данных* (БД) понимают хранилище структурированных данных, при этом данные должны быть непротиворечивы, минимально избыточны и целостны.

Реляционные БД представляют связанную между собой совокупность таблиц-сущностей базы данных (ТБД). Связь между таблицами может находить свое отражение в структуре данных, а может только подразумеваться, то есть присутствовать на неформализованном уровне. Каждая таблица БД представляется как совокупность строк и столбцов, где строки соответствуют экземпляру объекта, конкретному событию или явлению, а столбцы - атрибутам (признакам, характеристикам, параметрам) объекта, события, явления.

При практической разработке БД таблицы-сущности зовутся таблицами, строки-экземпляры - записями, столбцы-атрибуты - полями ТБД.

Одно из важнейших достоинств реляционных баз данных состоит в том, что можно хранить логически сгруппированные данные в разных таблицах и задавать *связи* между ними, объединяя их в единую базу. Такая организация данных позволяет уменьшить избыточность хранимых данных, упрощает их ввод и организацию запросов и отчетов.

Понятие первичного ключа

В каждой таблице БД может существовать первичный ключ. Под первичным ключом понимают поле или набор полей, однозначно (уникально) идентифицирующих запись. Первичный ключ должен быть минимально достаточным: в нем не должно быть полей, удаление которых из первичного ключа не отразится на его уникальности.

Таблица 1 - Данные отношения «Преподаватель»

Таб. №	ФИО	Уч. степень	Уч. звание	Код кафедры
101	Андреев А.П.	Д-р техн. наук	Профессор	01
102	Апухтин И.С.	Канд. техн. наук	Доцент	01
103	Глухов И.Л.	Канд. техн. наук	Доцент	01
104	Сеченов Ю.Б.	Канд. техн. наук	Доцент	01

В качестве первичного ключа в таблице «Преподаватель» может выступать только «Таб. №», значения других полей могут повторяться внутри данной таблицы.

Правила нормализации баз данных, и чисто практические соображения должны побудить разработчика **всегда определять первичный ключ для таблицы базы данных.**

Реляционные отношения (связи) между таблицами базы данных

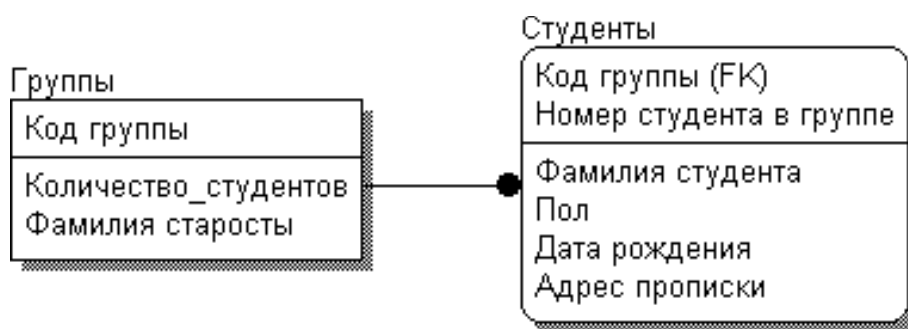
Между двумя или более таблицами базы данных могут существовать отношения подчиненности. Отношения подчиненности определяют, что для каждой записи главной таблицы *{master, называемой еще родительской}* может существовать одна или несколько записей в подчиненной таблице *{detail, называемой еще дочерней}*. Существует три разновидности связей между таблицами базы данных:

- «один-ко-многим»,
- «один-к-одному»,
- «многие-ко-многим».

Отношение «один-ко-многим» имеет место, когда одной записи родительской таблицы может соответствовать несколько записей в дочерней таблице.

Связь "один-ко-многим" является самой распространенной для реляционных баз данных.

В широко распространенной нотации структуры баз данных IDEF1X отношение «один-ко-многим» изображается путем соединения таблиц линией, которая на стороне дочерней таблицы оканчивается кружком или иным символом. Поля, входящие в первичный ключ для данной ТБД, всегда расположены сверху и отчеркнуты от прочих



полей линией.

Рисунок 2 - Отношение один-ко многим

Отношение «один-к-одному» имеет место, когда одной записи в родительской таблице соответствует одна запись в дочерней таблице.

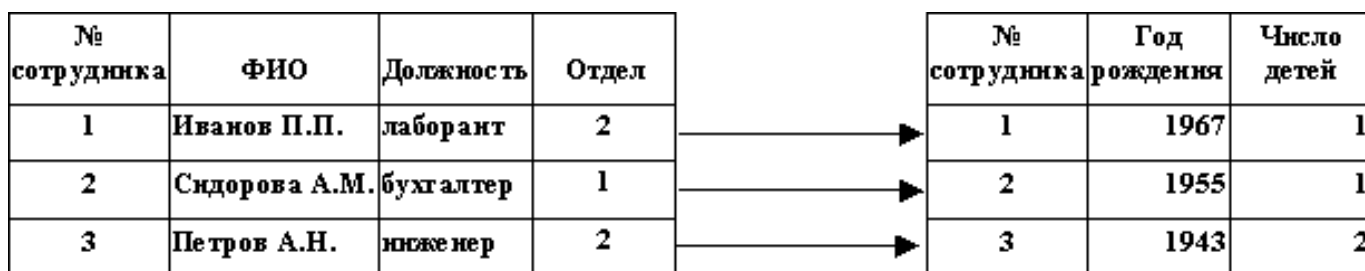


Рисунок 3 - Отношение один-к одному

Данное отношение используют, если не хотят, чтобы таблица БД «не распухала» от второстепенной информации.

Задание.

Спроектировать базу данных согласно вариантам:

Вариант 1. Деятельность торговой фирмы.

В базе данных учесть следующие признаки: дату, количество, наименование, тип, цену проданного товара, покупателя, его фирму, город, телефон.

Вариант 2. Деятельность предприятия по сборке изделий.

В базе данных учесть следующие признаки: наименование, тип, цену продажи некоторого изделия, количество дней на его сборку, количество компонент в изделии, описание, изготовитель компонент, тип и стоимость каждого компонента.

Вариант 3. Деятельность стола заказов.

В базе данных учесть следующие признаки: дату получения и исполнения заказа, скидку на заказ, количество и цену товара, вошедшего в заказ, имя клиента, его расчетный счет и величину кредита.

Вариант 4. Оплата коммунальных услуг.

В базе данных учесть следующие признаки: фамилию квартиросъемщика, его адрес, жилую площадь, число проживающих в квартире, дату и период оплаты коммунальных услуг, стоимость одного квадратного метра жилплощади, стоимость потребления холодной воды на одного проживающего.

Вариант 5. Работа фирмы с поставщиками.

В базе данных учесть следующие признаки: дату продажи некоторого товара, количество, цену, скидку при продаже и налог на продажу, а также поставщиков товара, страну и наличие лицензии на продажу.

Вариант 6. Начисление зарплаты.

В базе данных учесть следующие признаки: фамилию, адрес, телефон сотрудника, дату его рождения и дату устройства на работу, дату, вид и количество в часах выполненной работы, описание выполненной работы, тип освобождения от налога, нижнюю и верхнюю границы оплаты одного часа.

Вариант 7. Деятельность бюро добрых услуг.

В базе данных учесть следующие признаки: вид услуги, ее описание и стоимость, дату оказания этой услуги, скидку при оплате в зависимости от социального положения клиента, имя и место проживания клиента.

Вариант 8. Оплата междугородних телефонных разговоров.

В базе данных учесть следующие признаки: дату и время, продолжительность телефонного разговора, город, с которым состоялся разговор, фамилию, адрес, номер телефона клиента, тарифы городов и скидки на время разговора в течение суток.

Вариант 9. Поваренная книга.

В базе данных учесть следующие признаки: названия и типы блюд, описание компонент блюда с указанием количества в граммах, калорийности и стоимости 1 грамма, количества жиров, углеводов и белков в 1 грамме компонента.

Вариант 10. Книжная палата.

В базе данных учесть следующие признаки: дату и количество проданных книг, название, автора, издательство, тематику, цену проданной книги, сведения об авторе: фамилию, пол, дату рождения.

Контрольные вопросы (перечень вопросов по теме, на которые студент обязан знать ответы) и /или тестовые задания

1. Дайте определение базы данных.
2. Чем поле отличается от записи?
3. Какие основные функции СУБД и что это такое?
4. Опишите возможности СУБД MS Access.
5. Какие объекты входят в состав файла базы данных MS Access?
6. Какие ограничения на имена полей, элементов управления и объектов действуют в MS Access?
7. Чем отличаются режимы работы с объектами базы данных в MS Access: оперативный режим, режим конструктора?
8. Опишите, какие типы данных могут иметь поля в MS Access. Каков их предельный размер?

Лабораторная работа № 2. Отношения и воздействия

Формируемые компетенции или их части

№ п/п	Содержание компетенции	Шифр
1	Способен управлять процессами создания и использования продуктов и услуг в сфере информационно-коммуникационных технологий, в том числе разрабатывать алгоритмы и программы для их практической реализации	ОПК-3
2	Способен организовывать взаимодействие с клиентами и партнерами в процессе решения задач управления жизненным циклом информационных систем и информационно-коммуникационных технологий	ОПК-5

Отношение «многие-ко-многим» имеет место, когда:

а) записи в родительской таблице может соответствовать больше одной записи в дочерней таблице; б) записи в дочерней таблице может соответствовать больше одной записи в родительской таблице. Например, каждый студент изучает несколько дисциплин. Каждая дисциплина изучается несколькими студентами.

Группа	№ студента	ФИО студента	Код дисциплины	Название дисциплины
ИДМ-107	1	Иванов	1	Информатика
ИДМ-107	2	Петров	2	Математика
ИДМ-107	3	Сидоров	3	История

рис. 4 - Отношение многие-ко многим

Реляционные СУБД (в частности Access) не поддерживают связи «многие-ко-многим» на уровне индексов и ссылочной целостности. Считается, что всякую связь «многие-ко-многим» можно заменить на одну или более связей «один-ко-многим».

Например,

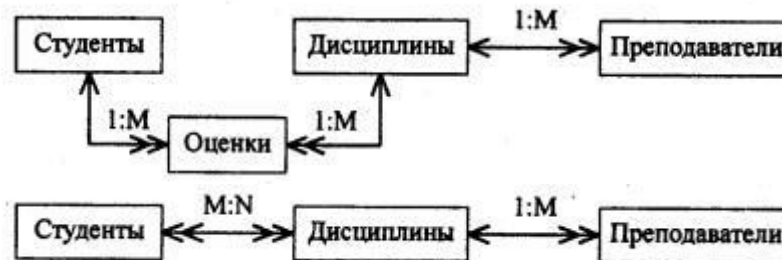


Рисунок 5 - Трансформации связи многие-ко многим

Ссылочная целостность и каскадные воздействия

Рассмотрим наиболее часто встречающуюся в базах данных связь «один-ко-многим». Как можно заметить, дочерняя и родительская таблицы связаны между собой по общему полю «Шифр группы». Назовем это поле **полем связи**.

Таблица «Группа»

Шифр группы	Кол-во студ-в	Прох. балл
101	30	4,50
102	32	4,50
103	29	4,80

Таблица «Студент»

Шифр группы	№ студ.	ФИО	Год рожд.	Прох. балл
101	01	Аристов Р.П.	1979	4,25
101	02	Борисова С.А.	1979	4,25
102	01	Макова Н.В.	1978	4,75

Рисунок 6 - Отношение один-ко многим

Возможны два вида изменений, которые приведут к утере связей между записями в родительской и дочерней таблицах:

- изменение значения поля связи в записи родительской таблицы без изменения значений полей связи в соответствующих записях дочерней таблицы;
- изменение значения поля связи в одной из записей дочерней таблицы без соответствующего изменения значения полей связи в родительской и дочерней таблицах.

Таблица «Группа»

Шифр группы	Кол-во студ-в	Прох. балл
A-101	30	4,50
102	32	4,50
103	29	4,80

Таблица «Студент»

Шифр группы	№ студ.	ФИО	Год рожд.	Прох. балл
101	01	Аристов Р.П.	1979	4,25
101	02	Борисова С.А.	1979	4,25
102	01	Макова Н.В.	1978	4,75

Таблица «Группа»

Шифр группы	Кол-во студ-в	Прох. балл
101	30	4,50
102	32	4,50
103	29	4,80

Таблица «Студент»

Шифр группы	№ студ.	ФИО	Год рожд.	Прох. балл
A-101	01	Аристов Р.П.	1979	4,25
101	02	Борисова С.А.	1979	4,25
102	01	Макова Н.В.	1978	4,75

Рисунок 7 - Примеры нарушения целостности базы данных

И в первом, и втором случаях наблюдается *нарушение целостности базы данных*, поскольку информация в ней становится *недоверенной*. Следовательно, нужно блокировать действия, которые нарушают целостность связей между таблицами, которую называют *ссылочной целостностью*.

Чтобы предотвратить потерю ссылочной целостности, используется механизм *каскадных изменений*. Он состоит в обеспечении следующих требований:

- необходимо запретить изменение поля связи в записи дочерней таблицы без синхронного изменения полей связи в родительской таблице;
- при изменении поля связи в записи родительской таблице, следует синхронно изменить значения полей связи в соответствующих записях дочерней таблицы;
- при удалении записи в родительской таблице, следует удалить соответствующие записи в дочерней таблице.

Необходимость разрешения или запрещения каскадных изменений обычно реализуется в СУБД при определении связей между таблицами. Собственно, таким образом, и происходит создание ссылочной целостности.

Понятие внешнего ключа

Для обеспечения ссылочной целостности в дочерней таблице создается внешний ключ. Во внешний ключ входят поля связи дочерней таблицы. Для связей типа "один-ко-многим" внешний ключ по составу полей должен совпадать с первичным ключом родительской таблицы.

Индексы и методы доступа

Индексы представляют собой механизмы быстрого доступа к данным в таблицах БД.

Сущность индексов состоит в том, что они хранят значения индексных полей (т.е. полей, по которым построен индекс) и указатель на запись в таблице.

При **последовательном методе доступа** для выполнения запроса к таблице БД просматриваются *все* записи таблицы, от первой до последней.

При **индексно-последовательном методе доступа** для выполнения запроса к таблице БД указатель в индексе устанавливается на первую строку, удовлетворяющую условию запроса (или его части), и считывается запись из таблицы по хранящемуся на нее в индексе указателю.

Определение первичных и внешних ключей таблиц БД приводят к созданию индексов по полям, объявленным в составе первичных или внешних ключей.

Нормализация таблиц при проектировании БД

При проектировании структуры новой БД определяют сущности (объекты, явления) предметной области, которые должны найти свое отражение в базе данных. В конечном итоге анализ предметной области должен привести к созданию *эскиза* БД. Сначала желательно изобразить *сущности* и *связи* между ними. Как правило, каждой сущности в БД соответствует *таблица*. Затем - в эскизе второго порядка - для каждой таблицы БД приводится список *атрибутов* - *полей* записи.

На этом этапе процесс проектирования структур БД является процессом творческим, неоднозначным, с другой стороны, узловые его моменты могут быть формализованы.

Одной из таких формализаций является требование, согласно которому реляционная база данных должна быть нормализована. Процесс *нормализации* имеет своей целью устранение избыточности данных и заключается в приведении к третьей нормальной форме (3НФ).

Первая нормальная форма (1НФ) требует, чтобы каждое поле таблицы БД:

- было неделимым;
- не содержало повторяющихся групп.

Неделимость поля означает, что значение поля не должно делиться на более мелкие значения. Например, если в поле "Подразделение" содержится название факультета и название кафедры, требование неделимости не соблюдается и необходимо из данного поля выделить или название факультета, или кафедры в отдельное поле. Повторяющимися являются поля, содержащие одинаковые по смыслу значения. Например, если требуется получить статистику сдачи экзаменов по предметам, можно создать поля для хранения данных об оценке по каждому предмету. Однако в этом случае мы имеем дело с повторяющимися группами.

Вторая нормальная форма (2НФ) требует, чтобы все поля таблицы зависели от первичного ключа, то есть, чтобы первичный ключ однозначно определял запись и не был избыточен. Те поля, которые зависят только от части первичного ключа, должны быть выделены в составе отдельных таблиц.

Третья нормальная форма (3НФ) требует, чтобы значение любого поля таблицы, не входящего в первичный ключ, не зависело от значения другого поля, не входящего в первичный ключ.

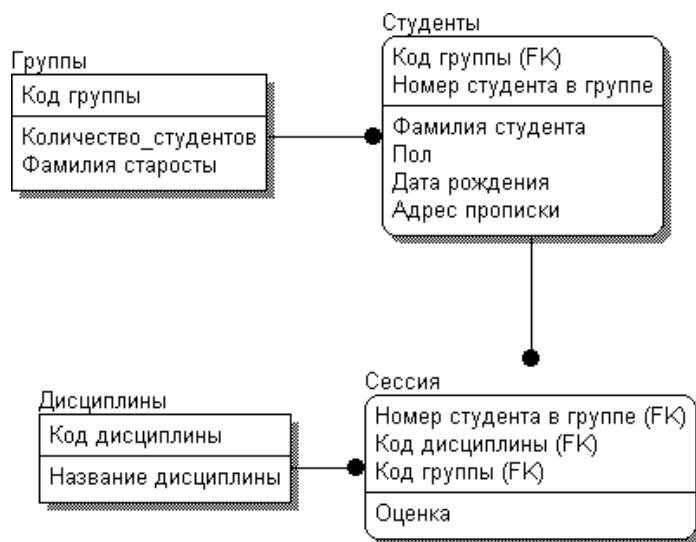


Рисунок 8 - Пример логической модели базы данных "Сессия"

Основным средством хранения информации в СУБД Access являются плоские таблицы, состоящие из строк (записей) и именованных столбцов (полей). Каждое поле записи содержит одну характеристику объекта и имеет строго определенный тип данных (например, текстовая строка, число, дата).

Задание.

Спроектировать базу данных согласно вариантам:

Вариант 1. Музыкальная коллекция.

В базе данных учесть следующие признаки: дату, количество и стоимость проданного альбома, страну, авторов слов и музыки, исполнителя, длительность каждой композиции в альбоме.

Вариант 2. Videотека.

В базе данных учесть следующие признаки: дату продажи видеокассеты, название фильма, страну, режиссера, тематику фильма, наличие Оскаров, дату выпуска фильма, стоимость кассеты, информацию о покупателе: возраст, пол, социальное положение.

Вариант 3. Олимпийские игры.

В базе данных учесть следующие признаки: номер, символ олимпиады, город проведения, наличие в городе гор или моря, дату открытия и закрытия, число видов спорта, по которым проводятся соревнования, команды-участницы, количество спортсменов в команде, число завоеванных золотых, серебряных и бронзовых медалей каждой командой.

Вариант 4. Учебный процесс.

В базе данных учесть следующие признаки: фамилии студентов, дату рождения, курс, дату сдачи, оценку и название предмета для каждого студента, для каждого предмета указать число часов на изучение, код предмета: гуманитарный

блок, математический или профессиональный и кафедру, которая ведет данный предмет.

Вариант 5. Учебная нагрузка преподавателя.

В базе данных учесть следующие признаки: фамилию, должность, звание преподавателя, кафедру, на которой он работает, название предмета, который он ведет, для предмета указать название, длительность в часах, код предмета:

гуманитарный блок, математический или профессиональный, для каждой должности и звания указать стоимость часа.

Вариант 6. Продажа билетов на самолеты.

В базе данных учесть следующие признаки: дату продажи билета, номер рейса, дату вылета рейса, номер места, фамилию пассажира, его социальное положение, данные по типу самолета, обслуживающего рейс: тип самолета, стоимость

билета, конечный пункт, продолжительность маршрута, экипаж, квалификация командира, его возраст и стаж полетов.

Вариант 7. Автобусный парк.

В базе данных учесть следующие признаки: дату подачи и дату исполнения заявки, продолжительность и вид поездки, число участников, сумма аванса, выплаченного за поездку, марка выделенного автобуса, его техническое состояние,

количество мест, стоимость билета, налоги и скидки в зависимости от вида поездки.

Вариант 8. Финансовое состояние вузов.

В базе данных учесть следующие признаки: число госбюджетных и хоздоговорных студентов в каждой студенческой группе, число преподавателей на факультете, среднюю стоимость обучения одного студента на факультете, среднюю зарплату преподавателя по университету, название университета, город, фамилию ректора, объем госбюджетных поступлений и дотаций из местного бюджета для каждого университета.

Вариант 9. Областное УВД.

В базе данных учесть следующие признаки: дату совершения и дату раскрытия преступления, вид и тяжесть преступления, описать участников: фамилию, дату рождения, вид участия, описать примененное оружие: марку, страну

изготовления, за кем числится.

Вариант 10. Фирма по продаже подержанных автомобилей.

В базе данных учесть следующие признаки: дату продажи, продавца, вид оплаты, данные о покупателе: фамилию, пол, возраст, социальное положение, информацию об автомобиле: марка, цвет, изготовитель, дата изготовления,

техническое состояние, мощность двигателя.

Контрольные вопросы (перечень вопросов по теме, на которые студент обязан знать ответы) и /или тестовые задания

1. Каково назначение справочной системы MS Access? Чем отличается поиск подсказки на вкладках: Содержание, Мастер ответов и Указатель?

2. Что такое выражения в MS Access? Какие бывают выражения и для чего они используются?

3. Какие особенности в записи различных операндов выражений: имя поля, число, текст?
4. Каково назначение строителя выражений?
5. С какой целью выполняется проектирование базы данных и в чем оно заключается?
6. Какие операции с данными в таблице базы данных вы знаете?
7. Каково назначение сортировки данных в таблице? Какие бывают виды сортировки?
8. Что подразумевает понятие абстрагирование в СУБД?

Лабораторная работа № 3. Построение ER-диаграммы (логическая модель) Формируемые компетенции или их части

№ п/п	Содержание компетенции	Шифр
1	Способен управлять процессами создания и использования продуктов и услуг в сфере информационно-коммуникационных технологий, в том числе разрабатывать алгоритмы и программы для их практической реализации	ОПК-3
2	Способен организовывать взаимодействие с клиентами и партнерами в процессе решения задач управления жизненным циклом информационных систем и информационно-коммуникационных технологий	ОПК-5

Теоретическая часть

Построение базы данных предметной области «Расписание экзаменов». Проектируемая система должна выполнять следующие действия:

- Хранить информацию о студентах, предметах, экзаменах (по какому предмету, какой группе и когда сдается), группах.
 - Печатать расписание для каждой группы, списки групп, результаты экзаменов.
- Разработаем ER-модель данной предметной области.

Примечание: исключительно ради рассмотрения различных вариантов связей и атрибутов примем за истину следующие условия:

- 1) дата рождения студента может быть не указана;
- 2) один и тот же экзамен сразу могут сдавать несколько групп.

Студент, предмет, группа, экзамен - явные кандидаты на сущность.

Связи между сущностями:

- «каждый студент должен обучаться в одной группе», «группа может состоять из нескольких студентов»,
- «каждая группа может сдавать несколько экзаменов», «каждый экзамен может проводиться у одной или нескольких групп»,
- «каждый экзамен должен сдаваться по одному предмету», «по каждому предмету могут сдаваться несколько экзаменов».

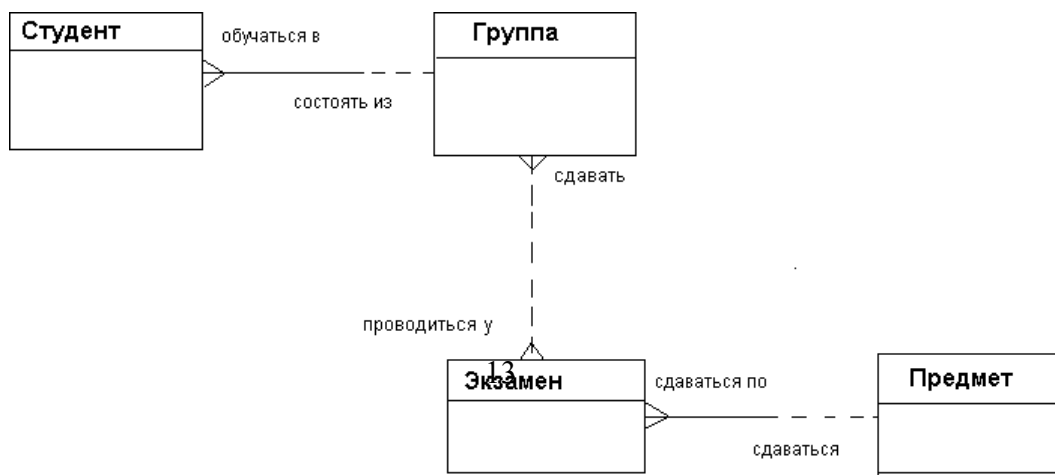


Рисунок 9 - Пример логической модели базы данных

Атрибуты сущностей:

1. Атрибутами студента могут быть *фамилия, имя, отчество, датарождения*.
2. Атрибутами группы могут быть *номер группы и форма обучения* (дневная или заочная).
3. Атрибутом экзамена может быть *дата* его сдачи.
4. Атрибутом предмета может быть его *название*.

Однако *оценки* за экзамен не могут быть атрибутом никакой из имеющихся сущностей, а, по-видимому, представляют собой отдельную сущность.

Связи:

- «каждый студент может получить несколько оценок», «каждая оценка должна быть поставлена одному студенту»,
 - «каждый экзамен может оцениваться несколькими оценками», «каждая оценка должна быть поставлена за один экзамен»
5. Атрибутом оценки может быть ее *значение*.

Отсюда

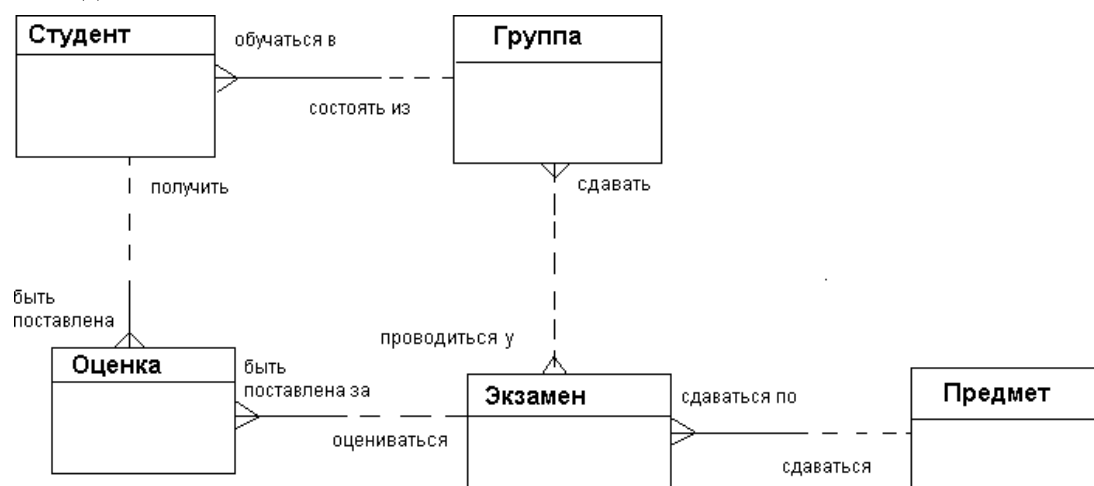


Рисунок 11 - Пример логической модели базы данных

Сущности группа и экзамен связаны друг с другом отношением много-ко-многим. Такая связь должна быть разделена на две связи типа один ко многим. Для этого требуется дополнительная сущность. Введем сущность «**Строка расписания экзаменов**». Связи:

- «каждая строка расписания экзаменов должна включать одну группу», «каждая группа может упоминаться в одной или нескольких строках расписания»;
- «каждая строка расписания экзаменов должна включать один экзамен», «каждый экзамен может упоминаться в одной или нескольких строках расписания».

Отсюда



Рисунок 12 - Пример логической модели базы данных

То есть мы получаем **концептуальную ER-диаграмму**.

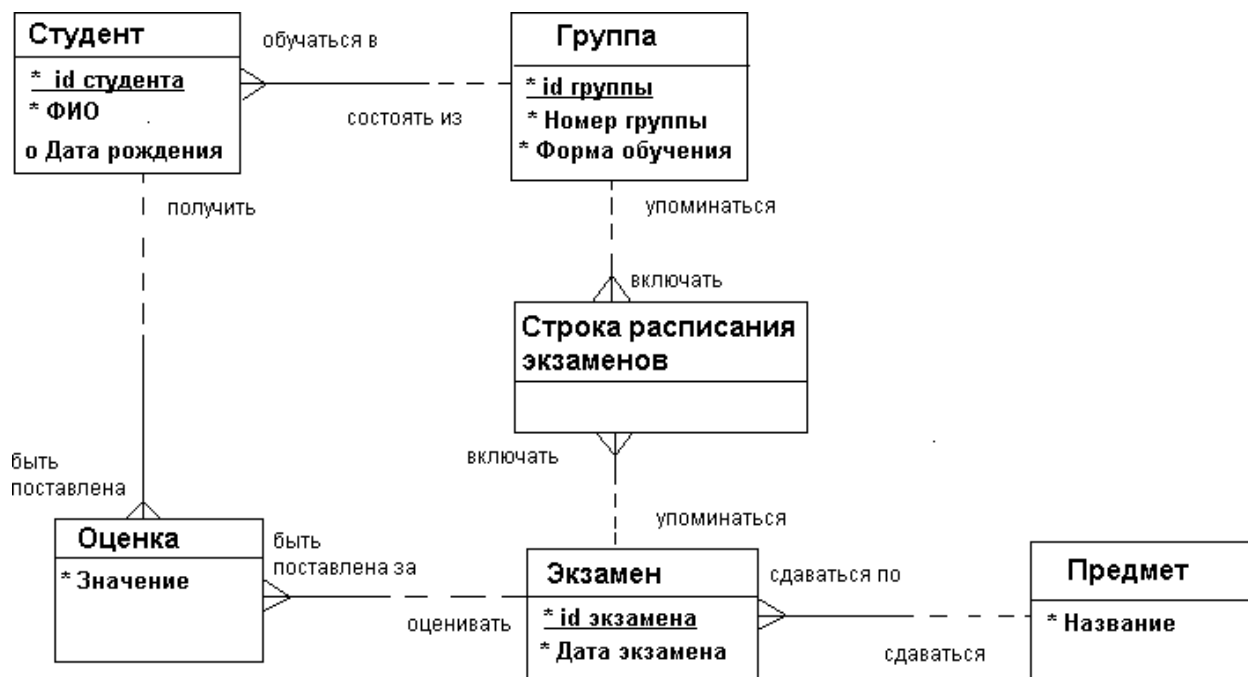


Рисунок 13 - Пример концептуальной ER-диаграммы

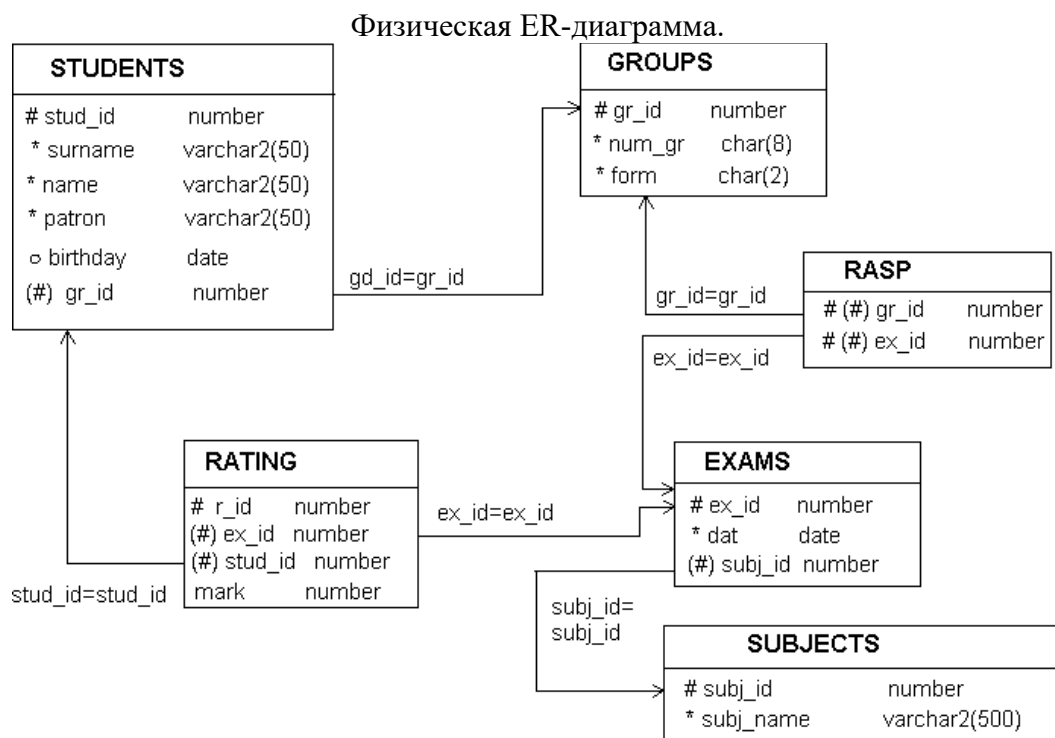


Рисунок 14 - Пример физической модели базы данных

ER-диаграммы рекомендуется разрабатывать в среде онлайн-редактора деловой графики <https://www.draw.io>. Допускается также использование CASE-средства AllFusionERwinDataModeler, редактора деловой графики MS Visio.

Задание.

1. Институт (деканаты, кафедры, учебный отдел).
 - * Студенты: паспортные данные, адрес, дата зачисления, номер приказа, факультет, группа, является ли старостой, кафедра (специализация), изучаемые (изученные) предметы, оценки, задолженности, стипендия.
 - * Учебные курсы: название, факультет(ы), групп(ы), кафедра, семестр(ы), форма отчётности, число часов.
 - * Преподаватели: паспортные данные, адрес, телефон, фотография, кафедра, должность, учёная степень, начальник (зав. кафедрой), предмет(ы), число ставок, зарплата.
2. Библиотека института.
 - * Книги: авторы, название, раздел УДК, раздел (техническая, общественно-политическая и т.п.), место и год издания, издательство, количество страниц, иллюстрированность, цена, дата покупки, номер сопроводительного документа (чек, счёт/накладная), вид издания (книги, учебники, брошюры, периодические издания), инвентарный номер (есть только для книг и некоторых учебников), длительность использования читателями (год, две недели, день), электронная версия книги или ее реферата (отсканированный текст).
 - * Читатели: номер читательского билета, ФИО, год рождения, адрес, дата записи, вид (студент, аспирант, преподаватель, сотрудник), курс, номер группы, названия взятых книг и даты их выдачи.
3. Отдел кадров и бухгалтерия некоторой компании.
 - * Сотрудники: ФИО, паспортные данные, фотография, дом. и моб. телефоны, отдел, комната, раб. телефоны (в т.ч. местный), подчинённые сотрудники, должность, тип(ы) работы, задание(я), проект(ы), размер зарплаты, форма зарплаты (почасовая, фиксированная).

* Отделы: название, комната, телефон(ы), начальник, размер финансирования, число сотрудников.

* Проекты: название, дата начала, дата окончания, размер финансирования, тип финансирования (периодический, разовый), задачи и их исполнители, структура затрат и статьи расходов.

4. Отдел поставок некоторого предприятия.

* Поставщики: название компании, ФИО контактного лица, расчётный счёт в банке, телефон, факс, поставляемое оборудование (материалы), даты поставок (по договорам и реальные), метод и стоимость доставки.

* Сырьё: тип, марка, минимальный запас на складе, время задержки, цена, продукты, при производстве которых используется, потребляемые объёмы (необходимый, реальный, на единицу продукции).

5. Технологический отдел некоторого предприятия.

* Производственные процессы: продукты, объёмы их производства, необходимые материалы, количества разных видов материалов на единицу продукции, отходы производства; используемое оборудование и его тип, даты ввода оборудования в строй, сроки амортизации, производительность оборудования; человеческие ресурсы (сколько всего и сколько по производству единицы продукции — сколько необходимо и сколько реально).

* Материалы: тип (категория), марка, является ли сырьём (или производится на предприятии), потребляемые объёмы (в т.ч. на единицу конечной продукции), в рамках каких технологических процессов используется, цена.

6. Отдел продаж некоторой фирмы.

* Клиенты: название компании, ФИО контактного лица, адрес выставления счёта, адрес доставки, телефон, факс.

* Заказы: тип заказа (покупка, гарантийный ремонт, негарантийный ремонт), общая стоимость, скидка, товар(ы), их изготовители, модели (марки), серийные номера, описание неисправностей, необходимые ресурсы, клиент, дата получения заказа, срок завершения, дата выставления счёта и его оплаты, метод оплаты, дата поставки, метод и стоимость доставки.

* Ресурсы: ФИО, отдел(ы) и телефон(ы) исполнителя(ей), число рабочих часов для выполнения заказа, ставка зарплаты, ответственный за выполнение заказа, необходимое оборудование и расходные материалы, их количество и стоимость, а также наличие материалов на складе.

7. Магазин (внутренний учет).

* Клиенты: юридическое или физическое лицо, ФИО, адрес, телефон, адрес выставления счёта, вид и номер карточки, факс.

* Продажи: наименования, модели (марки) и серийные номера товаров, поставка из магазина или со склада, количество и общая стоимость товаров, размер скидки, тип скидки, форма оплаты (наличными, оплата счёта, по карточке), необходимость доставки, стоимость и тип доставки, адрес доставки.

* Товары: категория, модель, название производителя, адрес производителя, цена, количество в магазине и на складе.

Контрольные вопросы (перечень вопросов по теме, на которые студент обязан знать ответы) и /или тестовые задания

1. В чем отличие ERD от DFD?

2. Какие основные цели преследуют ER-диаграммы?

3. Какие основные элементы используются в ER-диаграммах?

4. Какие способы изображения элементов на ER-диаграммах вы знаете?

Лабораторная работа № 4. Построение ER-диаграммы (физическая модель)

Формируемые компетенции или их части

п/п	№	Содержание компетенции	Шифр
.	1	Способен управлять процессами создания и использования продуктов и услуг в сфере информационно-коммуникационных технологий, в том числе разрабатывать алгоритмы и программы для их практической реализации	ОПК-3
.	2	Способен организовывать взаимодействие с клиентами и партнерами в процессе решения задач управления жизненным циклом информационных систем и информационно-коммуникационных технологий	ОПК-5

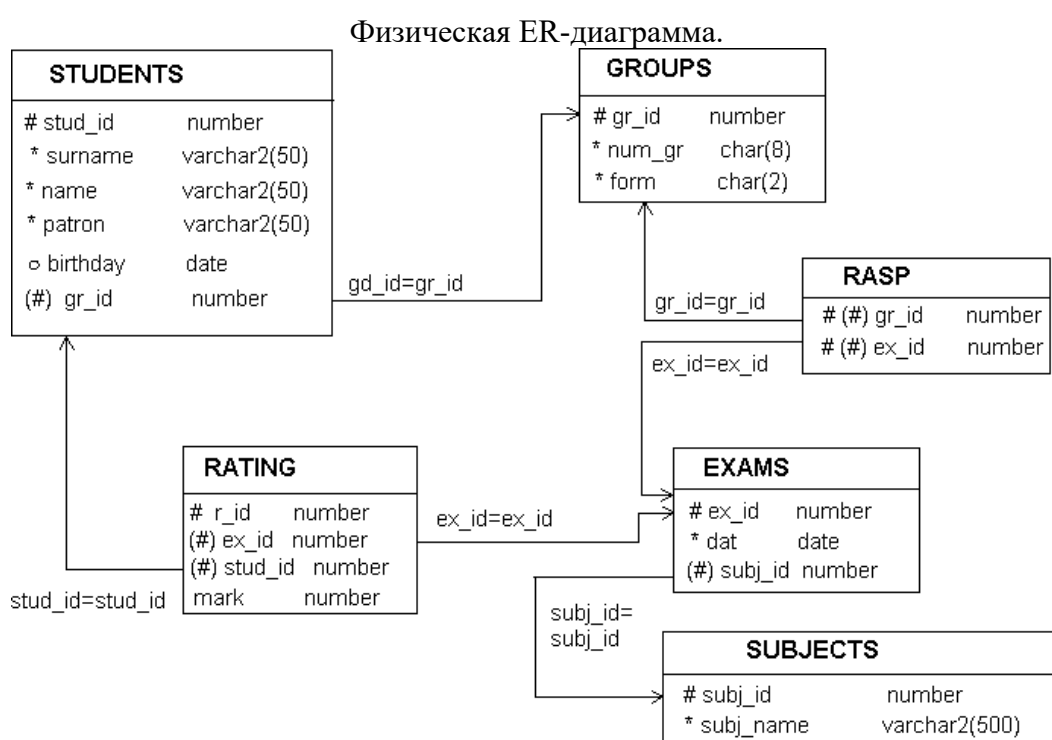


Рисунок 14 - Пример физической модели базы данных

ER-диаграммы рекомендуется разрабатывать в среде онлайн-редактора деловой графики <https://www.draw.io>. Допускается также использование CASE-средства AllFusionERwinDataModeler, редактора деловой графики MS Visio.

Задание.

1. Электронный магазин (информация для клиентов).
 - * Товары: категория, модель, производитель, цены (в т.ч. средняя и минимальная), есть ли в наличии, описание, характеристики, внешний вид; магазины, где можно купить товар, их телефоны и адреса; аксессуары, их цены и где их купить.
 - * Магазины: название, компания-владелец, её юридический адрес и home-site, контактные телефоны, адрес, схема проезда, эмблема; товары и цены на них; рекламная информация: некоторые товары с фотографиями, описаниями и ценами, основные отделы (категории товаров).
2. Пункт проката видеозаписей (внутренний учет).
 - * Видеокассеты: идентификационный номер видеокассеты, тип видеокассет, дата его создания, компания- поставщик, число штук данного типа (общее, в магазине,

выдано в настоящее время, выдано всего, выдано в среднем за месяц), общая длительность записей; записи видеокассет: название, длительность, категория, год выпуска и производитель (оригинала).

* Клиенты: ФИО, паспортные данные, адрес, телефон; заказы, т.е. взятые видеокассеты (сейчас и в прошлом): номер, дата выдачи, дата возвращения, общая стоимость заказа.

3. Пункт проката видеозаписей (информация для клиентов).

* Видеокассеты: краткое описание, внешний вид (этикетка), марка (пустой) видеокассеты, цена за единицу прокатного времени (например: 1 день, 3 дня, неделя), есть ли в наличии, общая длительность записей; записи на видеокассете: название, длительность, жанр (категория), тема, год и страна выпуска (оригинала), кинокомпания, описание, актеры, режиссер.

* Заказы: идентификационные номера и названия выданных видеокассет, дата выдачи, дата возвращения (продления), общая стоимость заказа, возвращены ли кассеты заказа.

4. Кинотеатры (информация для зрителей).

* Фильмы: название, описание, жанр (категория), длительность, популярность (рейтинг, число проданных билетов в России и в мире), показывается ли сейчас (сегодня, на текущей неделе), в каких кинотеатрах показывается, цены на билеты (в т.ч. средние).

* Кинотеатры: название, адрес, схема проезда, описание, число мест (в разных залах, если их несколько), акустическая система, широкоэкранность, фильмы и цены на них: детские и взрослые билеты в зависимости от сеанса (дневной, вечерний и т.п.) и от категории мест (передние, задние и т.п.); сеансы показа фильмов (дата и время начала).

5. Ресторан (информация для посетителей).

* Меню: дневное или вечернее, список блюд по категориям.

* Блюда: цена, название, вид кухни, категории (первое, второе и т.п.; мясное, рыбное, салат и т.п.), является ли вегетарианским, компоненты блюда, время приготовления, есть ли в наличии.

* Компоненты блюд: тип (гарнир, соус, мясо и т.п.), калорийность, цена, рецепт, время приготовления, есть ли в наличии, ингредиенты (продукты) и их расходы на порцию.

6. Аналитический отдел некоторой компании (поиск и анализ публикаций).

* Категории: название, тип (область исследований, область приложений и т.п.), родительская категория, дочерние категории, связанные по смыслу категории (с пояснениями о связях), найденные публикации.

* Публикации: название, тип (газетная, книжная, web и т.п.), название, тип, адрес и телефон источника (газета, книга, сайт и т.п.), выходные данные (date-line), язык, реферат, ключевые слова, категории (с указанием степени уверенности отнесения к ним), текст и его тип (обычный, DOC, HTML, отсканированные картинки и т.п.), обзор.

* Задачи: тип задачи (классификация или поиск), сотрудник (создавший категорию или нашедший публикацию, ответственный за категорию или публикацию и т.п.), завершена ли работа над задачей.

7. Аналитический отдел некоторой компании (анализ рынка технологий — например, по публикациям — см. №13).

* Организация: название, тип (промышленная, финансовая, торговая, исследовательская и т.п.), категория(и), организация-владелец (акционеры), страна, контактная информация; договорные отношения с другими организациями.

* Технология (продукт): название, категория(и), организация-разработчик и производитель(и), использующие организации.

* Человек: фамилия, имя, тип (начальник, менеджер, создатель технологии и т.п.), организация(и), в которой работает, контактная информация.

8. Компания по (разработке и) сопровождению программного обеспечения.

* Ошибка (bug): краткое и полное описание, срок поступления информации об ошибке, её источник (пользователь, тестировщик) и его координаты, уровень ошибки (критическая, важная, незначительная и т.п.), категория функциональности (интерфейс, данные, расчетный алгоритм, другое, неизвестная категория), часть проекта, модуль (пакет), программист, ответственный за модуль, программист, ответственный за исправление ошибки, срок исправления (необходимый и реальный), исправлена ли, проверено ли исправление тестировщиком.

Контрольные вопросы (перечень вопросов по теме, на которые студент обязан знать ответы) и /или тестовые задания

1. Как на ER-диаграммах указываются ключевые атрибуты. Приведите пример ER- диаграммы с зависимостью один ко-многим, где одна из сущностей обязательна, а вторая нет.
2. Какие типы связей вы знаете?
3. Какими свойствами характеризуются связи?

Лабораторная работа № 5. Построение базы данных в СУБД Access.

Формируемые компетенции или их части

№ п/п	Содержание компетенции	Шифр
1	Способен управлять процессами создания и использования продуктов и услуг в сфере информационно-коммуникационных технологий, в том числе разрабатывать алгоритмы и программы для их практической реализации	ОПК-3
2	Способен организовывать взаимодействие с клиентами и партнерами в процессе решения задач управления жизненным циклом информационных систем и информационно-коммуникационных технологий	ОПК-5

Теоретическая часть

Для создания новой таблицы необходимо открыть базу данных, перейти на вкладку Создать и выбрать желаемый пункт для создания таблицы.

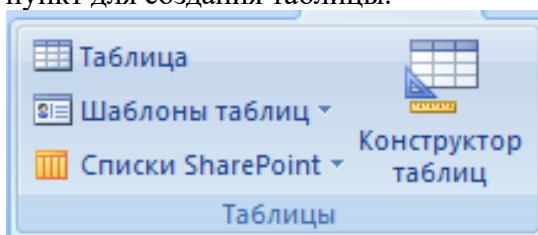


Рисунок 14 - Окно выбора способа создания новой таблицы в СУБД MS Access

В Access используются три способа создания таблиц: путем ввода данных (by entering data), с помощью Мастера создания таблиц (by using wizard) и с помощью Конструктора таблиц (in Design view), который является наиболее универсальным.

В режиме Конструктора таблицы создаются путем задания имен полей, их типов и свойств. Чтобы создать таблицу в режиме Конструктора, необходимо:

1. Щелкнуть левой кнопкой мыши на Вкладке **Создание Конструктор таблиц**.

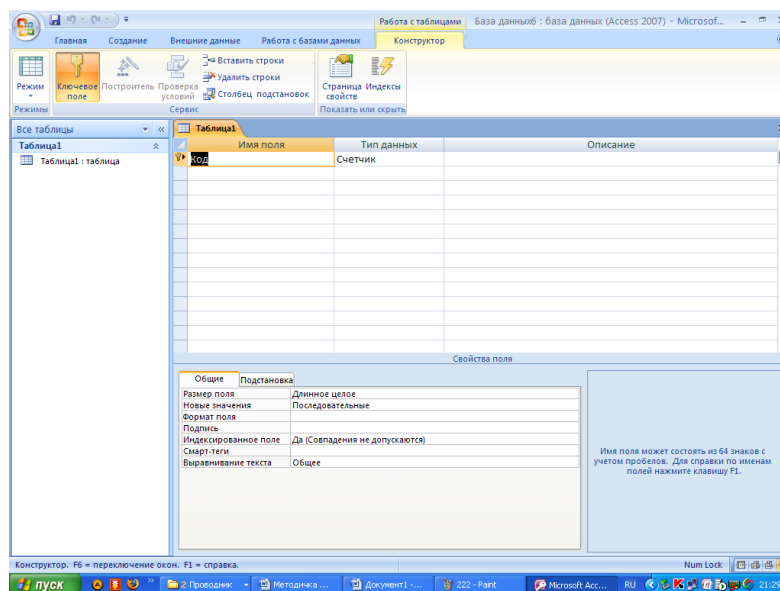


Рисунок 15 - Окно создания новой таблицы в режиме Конструктора

2. В окне Конструктора таблиц в столбец **Имя поля** (Field Name) ввести имена полей создаваемой таблицы.

3. В столбце **Тип данных** (Data Type) для каждого поля таблицы выбрать из раскрывающегося списка тип данных, которые будут содержаться в этом поле.

4. В столбце **Описание** (Description) можно ввести описание данного поля (не обязательно).

5. В нижней части окна Конструктора таблиц на вкладках **Общие** (General) и **Подстановка** (Look up) ввести свойства каждого поля или оставить значения свойств, установленные по умолчанию.

6. После описания всех полей будущей таблицы нажать кнопку **Заккрыть** (в верхнем правом углу окна таблицы).

7. На вопрос **Сохранить изменения макета или структуры таблицы <имя_таблицы>?** (Do you want to save changes to the design of table<имя_таблицы>?), нажать кнопку Да (Yes).

8. В окне **Сохранить как** (Save As) в поле Имя таблицы (Table Name) ввести имя создаваемой таблицы и нажать кнопку ОК.

9. В ответ на сообщение **Ключевые поля не заданы** (The re is no primary key defined) и вопрос **Создать ключевое поле сейчас?** (Do you want to create a primary key now?) нажмите кнопку Да (Yes) если ключевое поле необходимо, или кнопку Нет (No) если такого не требуется.

10. После указанных действий в списке таблиц в окне базы данных появятся имя и значок новой таблицы. Ввести данные в созданную таблицу можно, открыв таблицу в режиме Таблицы.

В Microsoft Access имеются следующие типы данных:

□ **Текстовый (Text)** — символьные или числовые данные, не требующие вычислений. Размер текстового поля задается с помощью свойства **Размер поля** (Field Size), в котором указывается максимальное количество символов, которые могут быть введены в данное поле.

□ **Поле МЕМО (MEMO)** — поле МЕМО предназначено для ввода текстовой информации, по объему превышающей 255 символов. Этот тип данных отличается от типа **Текстовый (Text)** тем, что в таблице хранятся не сами данные, а ссылки на блоки данных, хранящиеся отдельно. За счет этого ускоряется обработка таблиц (сортировка, поиск и т.п.).

□ **Числовой (Number)** — числовой тип применяется для хранения числовых

данных, используемых в математических расчетах.

- Дата/Время (Date/Time) — тип для представления даты и времени.
- Денежный (Currency) — тип данных, предназначенный для хранения данных, точность представления которых колеблется от 1 до 4 десятичных знаков.
- Счетчик (AutoNumber) — поле содержит 4-байтный уникальный номер, определяемый Microsoft Access автоматически для каждой новой записи либо случайным образом, либо путем увеличения предыдущего значения на 1. Значения полей типа счетчика обновлять нельзя.
- Логический (Yes/No) — логическое поле, которое может содержать только два значения, интерпретируемых как Да/Нет, Истина/Ложь, Включено/Выключено.
- Поле объекта OLE (OLE object) — содержит ссылку на OLE-объект (лист Microsoft Excel, документ Microsoft Word, звук, рисунок и т. п.). В поле объекта OLE могут храниться произвольные данные, в том числе и данные нескольких типов. Это позволяет обойти основное ограничение реляционных баз данных, которое требует, чтобы в каждом поле хранились данные только одного типа.
- Гиперссылка (Hyperlink) — дает возможность хранить в поле ссылку, с помощью которой можно сослаться на произвольный фрагмент данных внутри файла или Web-страницы на том же компьютере, в Интернет.

При установке курсора на какое-либо из заполняемых полей таблицы в левой нижней части экрана появляются свойства этого поля. Причем свойства задаваемого поля зависят от принятого для него типа данных.

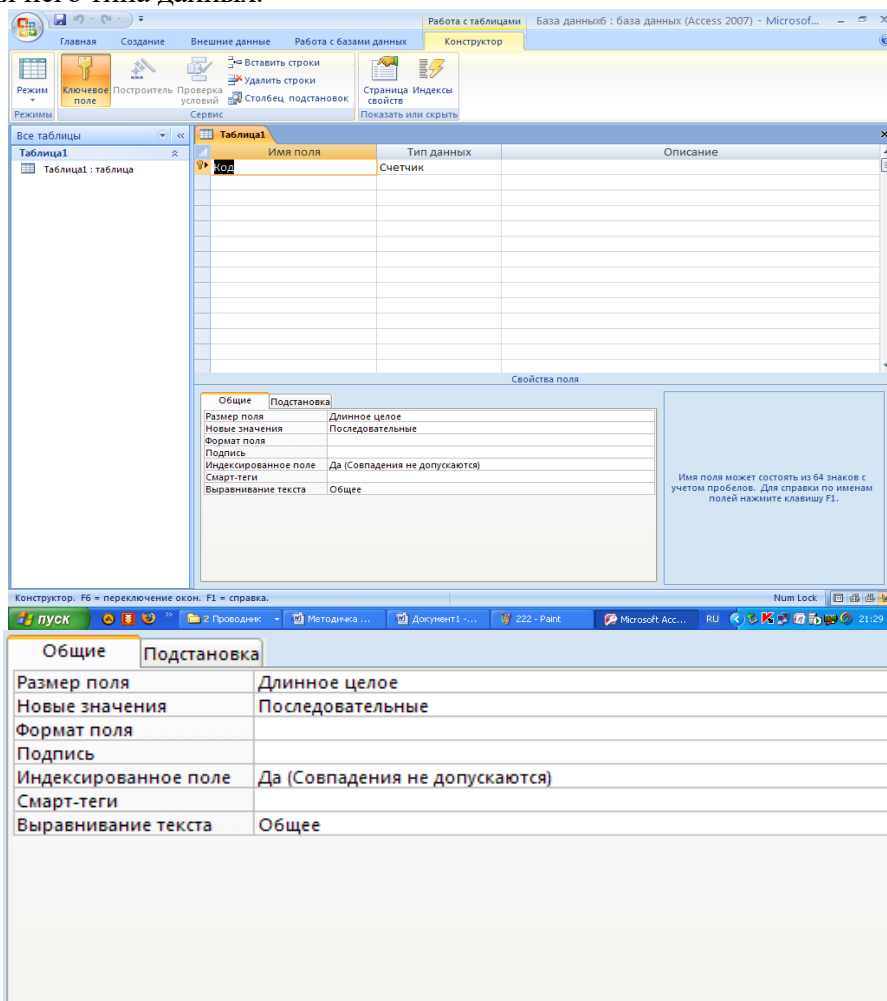


Рисунок 16 - Установка свойств полей таблицы

Свойство **Размер поля** зависит от размера хранящейся в нем информации. Например, для ввода данных в поле

Фамилия работника достаточно 20 символов.

Для большинства типов данных характерно свойство поля - **Подпись** (Caption). С помощью этого свойства можно задать названия полей таблицы, которые выводятся в различных режимах (в надписях, присоединенных к элементам управления формы, в заголовке столбца в режиме Таблицы и т.д.). Кроме того, для большинства типов данных существует свойство **Обязательное поле** (Required), которое определяет необходимость ввода данных в это поле.

Свойство **Формат поля** (Format) указывает формат отображения данных из поля в режиме Таблицы. Для определения формата полей текстового типа используются специальные символы форматирования. Для числовых полей значение формата можно выбрать из раскрывающегося списка. Для логических полей можно выбрать из списка следующие варианты: Да/Нет (Yes/No), Истина/Ложь (True/False), Вкл/Выкл (On/Off).

С помощью свойства **Маска ввода** (InputMask) указывается маска, позволяющая автоматизировать проверку ввода символов в поле. Она применяется к таким полям, как номер телефона, дата и т. д. Задавать маску ввода можно вручную или с помощью Мастера.

Проще всего научиться работать с маской ввода с помощью мастера по созданию масок ввода. Для создания маски с помощью мастера необходимо щелкнуть на соответствующее поле, затем - по ячейке свойства **Маска ввода**, расположенной в нижней части этого окна. Вы увидите справа небольшую кнопку с тремя точками - кнопку построителя.

Нажмите эту кнопку, чтобы воспользоваться помощью мастера по созданию масок ввода. Выберите один из предлагаемых стандартных масок и нажмите далее.

Свойство **Индексированное поле** (Indexed) определяет, является ли данное поле индексированным, и если является, то в каком режиме. Существуют два режима индексирования: *Совпадения допускаются* (Duplicates OK) и *Совпадения не допускаются* (No duplicates). В первом случае поле может содержать повторяющиеся значения, во втором — нет.

Два свойства, которые тоже определены для большинства полей, позволяют выполнять проверку данных, вводимых в поле:

□ **Условие на значение** (ValidationRule) — свойство определяет условие (ограничение), накладываемое на вводимые в это поле данные. При несоответствии вводимых данных указанному условию выдается сообщение об ошибке. Для создания условия на значение с помощью мастера необходимо щелкнуть на соответствующее поле, затем - по ячейке свойства **Условие на значение**, расположенной в нижней части этого окна. Вы увидите справа небольшую кнопку с тремя точками - кнопку построителя. Нажмите эту кнопку, на экране появится построитель выражения, с помощью которого вы сможете задать любое условие.

□ **Сообщение об ошибке** (ValidationText) — свойство определяет то сообщение, которое будет выдаваться пользователю, если при вводе данных не соблюдается условие, указанное в свойстве Условие на значение (ValidationRule). На пример, "Введенное значение выходит за рамки заданного диапазона".

Задание.

1. Разработать структуру базы данных согласно вариантам задания 1.
2. Создать таблицы в среде Microsoft Access. Для каждого элемента данных определить имя, тип данных, свойства: размер, условия на значение, маску ввода, сообщение об ошибке, значение по умолчанию и т.д. Определить для таблиц первичные ключи.

3. Организовать связь между таблицами.

4. Заполнить таблицы данными.

Контрольные вопросы (перечень вопросов по теме, на которые студент обязан знать ответы) и /или тестовые задания

1. Какие структуры ACCESS вы знаете?
2. Для чего служит структура "таблица"?
3. Для чего данным задаются различные типы?
4. Что можно настроить в свойствах поля таблицы?

5. Как отсортировать данные в таблицах по возрастанию и убыванию?

Лабораторная работа № 6. Нормализация отношений

Формируемые компетенции или их части

№ п/п	Содержание компетенции	Шифр
1	Способен управлять процессами создания и использования продуктов и услуг в сфере информационно-коммуникационных технологий, в том числе разрабатывать алгоритмы и программы для их практической реализации	ОПК-3
2	Способен организовывать взаимодействие с клиентами и партнерами в процессе решения задач управления жизненным циклом информационных систем и информационно-коммуникационных технологий	ОПК-5

Теоретическая часть

Ввод различных выражений в Microsoft Access возможен не только вручную, но и может быть облегчен с помощью **Построителя выражений**.

Построитель выражений вызывается всякий раз, когда в поле свойства объекта Access вы щелкните кнопку **Построить**  на строке главного меню или нажимаете кнопку Построителя (кнопка с тремя точками, например, в строке свойства **Условие назначения**).

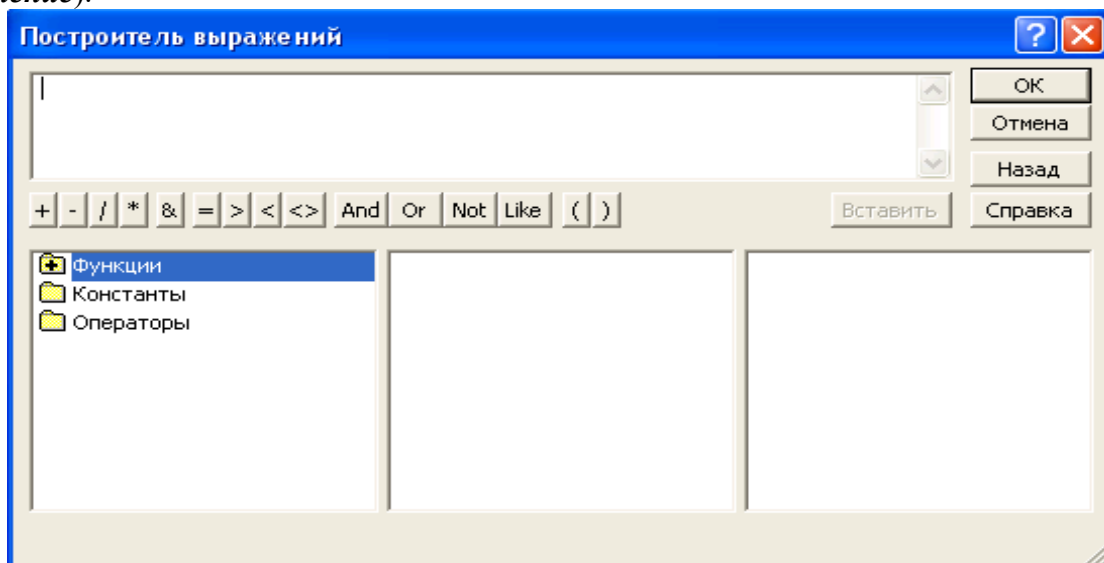


Рисунок 17 - Окно Построителя выражений

С помощью удобных клавиш, расположенных в Построителе можно задавать любое выражение, в том числе использовать различные формулы для вычислений и применять встроенные функции Access. Введение выражений с помощью Построителя позволяет также наглядно определить правильность ввода и тем самым избежать неточностей и ошибок при наборе текста и формул.

Создать поле-список и сделать более простым ввод значений в поле таблицы позволяет **операция подстановки**. Используя эту операцию, при последующем заполнении таблицы данными можно выбирать значения поля из списка. Список значений может быть как фиксированным, так и содержаться в таблице или запросе. Сформировать столбец

подстановок для поля помогает **Мастер подстановок** (LookupWizard), который находится в поле **Тип данных**.

Проверить наличие подстановки можно, открыв вкладку **Подстановка**, в свойствах поля.

Выше неоднократно упоминалось понятие ключевого поля. **Ключевое поле** — это одно или несколько полей, комбинация значений которых однозначно определяет каждую запись в таблице. Если для таблицы определены ключевые поля, то Microsoft Access предотвращает дублирование или ввод пустых значений в ключевое поле. Ключевые поля используются для быстрого поиска и связи данных из разных таблиц при помощи запросов, форм и отчетов.

Операция сортировки данных используется всегда для удобства нахождения нужной информации. Когда на экране (или на бумаге) отображается таблица, гораздо легче найти нужную строку, если эти строки упорядочены. Вы привыкли к тому, что табличные данные упорядочены по алфавиту, по дате, по увеличению или уменьшению значений в столбцах, содержащих числа. Но в разных ситуациях мы хотели бы сортировать строки по разным признакам (столбцам таблицы). В идеале это должно выполняться легким движением руки. Именно так и позволяет делать Access. По умолчанию, когда таблица открывается в режиме Таблицы, она упорядочивается по значению ключевого поля. Если ключевое поле для таблицы не определено, записи выводятся в порядке их ввода в таблицу. Если нужно отсортировать записи по значению другого поля, достаточно установить курсор на любую строку соответствующего столбца и нажать одну из кнопок на панели инструментов: **Сортировка по возрастанию** или **Сортировка по убыванию**.

Создание Связей и индексов:

Каждая БД представляет обычно несколько таблиц, число которых может достигать, в общем случае, до десятков и сотен. При этом часто оказывается, что в разных таблицах хранятся одинаковые данные.

Для связывания полей необязательно совпадение их имен, но обязательно совпадение их типов.

Связи между таблицами можно устанавливать двумя путями. Первый путь - *графический*. Войдите в Схему данных, и выберите таблицы для установления связей. Далее следует выбрать в главной таблице поле для связи, нажать левую кнопку мыши и перетащить поле во вторую таблицу. Отпустить левую кнопку мыши над тем полем подчиненной таблицы, с которым устанавливается связь.

Второй путь - создание связей через **Мастер подстановок**, который активизируется при выборе типа данных в конструкторе таблицы.

Чтобы изменить связь нужно войти в Схему данных (ярлык на главной панели инструментов), щелкнуть правой кнопкой мыши по линии связи и войти в диалоговое окно *Изменить связь*, установив обеспечение целостности данных, каскадное обновление и каскадное удаление записей связанных таблиц.

Если отношение между таблицами «один-ко-многим», то слева из списка *Таблица/запрос* выбирается главная таблица и поле в этой таблице, а справа из списка *Связанная таблица/запрос* - подчиненная и соответственно поле в ней. Если отношение «один-к-одному», то порядок таблиц значения не имеет. Если вы уже устанавливали связь графически, то все поля в списке уже выбраны, и нужно только определить правила ссылочной целостности. Для этого устанавливают флажок *Обеспечение целостности данных*.

Когда создается новая связь, можно также воспользоваться кнопкой *Новое* и в окне *Создание* ввести имена связываемых таблиц и имена полей, используемых для связи. Нажать кнопку *ОК*.

Индексы - это внутренняя таблица, состоящая из двух столбцов: значения выражения, содержащего все поля, включенные в индекс, и местоположение каждой записи таблицы с данным значением индексного выражения. Допустим, вы часто

осуществляете поиск Сотрудников в таблице Сотрудники по Фамилии, Имени и Отчеству. Если индекс отсутствует, то Access просматривает все записи вашей таблицы. Эта операция выполняется быстро только при небольшом количестве записей. Если вы создадите индекс, то Access его использует для прямого поиска записей.

Индексы бывают *по одному полю* и *составные*. Для создания индекса по одному полю откройте таблицу в режиме Конструктора и откройте поле, для которого вы хотите создать индекс. Щелкните по ячейке Свойства Индексированное поле, расположенное в нижней части окна таблицы и установите значение Да. Если поле, по которому вы индексируете имеет повторяющиеся значения, то следует выбрать Да (совпадения допускаются). Однако Access также может создать индекс, содержащий только уникальные значения данного поля - для этого нужно выбрать в списке значений Да (совпадения не допускаются). Access автоматически создает такой индекс по первичному ключу таблицы.

Чтобы создать индекс по нескольким полям, откройте таблицу Сотрудники в режиме Конструктора, затем нажмите кнопку Индексы на панели инструментов в режиме Конструктора.

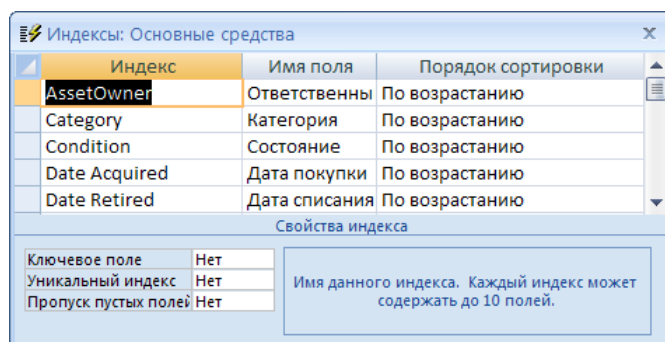


Рисунок 18 - Диалоговое окно создания Индекса

В столбце *Индекс* заполняется название вашего индекса, в столбце *Имя поля* указывается наименование индексируемого поля, в столбце *Порядок сортировки* можно указать, как вы хотите отсортировать данные, по возрастанию или по убыванию.

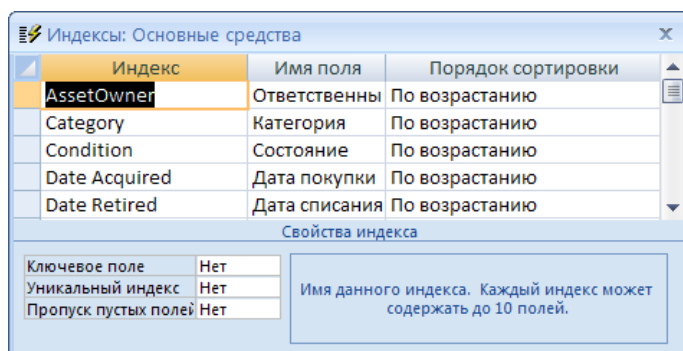


Рисунок 19 - Создание нескольких индексов

Для построения составного индекса поместите курсор на пустую строку и введите с клавиатуры имя индекса. В столбце *Имя поля* окна индексы выберите необходимое поле, пользуясь раскрывающимся списком, затем, не указывая никакого нового названия индекса, поместите курсор на следующую строку столбца *Имя поля* и укажите следующее поле, которое будет включено в составной индекс.

Задание.

1. Разработать структуру базы данных согласно вариантам задания 1.
2. Создать таблицы в среде Microsoft Access. Для каждого элемента данных определить имя, тип данных, свойства: размер, условия на значение, маску ввода, сообщение об ошибке, значение по умолчанию и т.д. Определить для таблиц

первичные ключи.

3. Организовать связь между таблицами.
4. Заполнить таблицы данными.

Контрольные вопросы (перечень вопросов по теме, на которые студент обязан знать ответы) и /или тестовые задания

1. Для чего служат связи?
2. Виды связей. Способы создания связей.
3. Назначение индексов.
4. Виды индексов.
5. Как создать составной индекс?

Лабораторная работа № 7. Создание запросов. Окно запроса

Формируемые компетенции или их части

п/п	№ Содержание компетенции	Шифр
.	1 Способен управлять процессами создания и использования продуктов и услуг в сфере информационно-коммуникационных технологий, в том числе разрабатывать алгоритмы и программы для их практической реализации	ОПК-3
.	2 Способен организовывать взаимодействие с клиентами и партнерами в процессе решения задач управления жизненным циклом информационных систем и информационно-коммуникационных технологий	ОПК-5

Теоретическая часть

Таблицы только хранят данные, но необходимо иметь возможность выбрать заданные данные из нескольких таблиц. Именно для этого служат **запросы на выборку**.

В **запросах на выборку** данные могут: отбираться по многим критериям; сортироваться; с ними могут производиться вычислительные операции.

Запрос это временная таблица. Это значит, что данные в них не хранятся постоянно, а только временно вызываются из таблиц, по заранее заданному шаблону, в момент активизации запроса. Таким образом, в базе данных постоянно хранится только шаблоны вызова данных (временные таблицы удаляются после закрытия запроса), а сама информация не дублируется.

Запросы позволяют:

1. формировать сложные критерии для выбора записей из одной или нескольких таблиц;
2. указывать поля, которые должны быть отображены для выбранных записей;
3. редактировать группы записей, удовлетворяющих определенным критериям;
4. выполнять вычисления с использованием выбранных данных.

Для создания запроса необходимо:

1. Открыть свою базу данных.
2. Перейти на вкладку Запросы
3. Во вкладке Создание выбрать команду Конструктор запросов. После нажатия этой опции появится окно конструктора запроса с диалоговым окном добавления таблиц. Окно добавления таблиц можно также вызвать командой Добавить таблицу из меню Запрос.

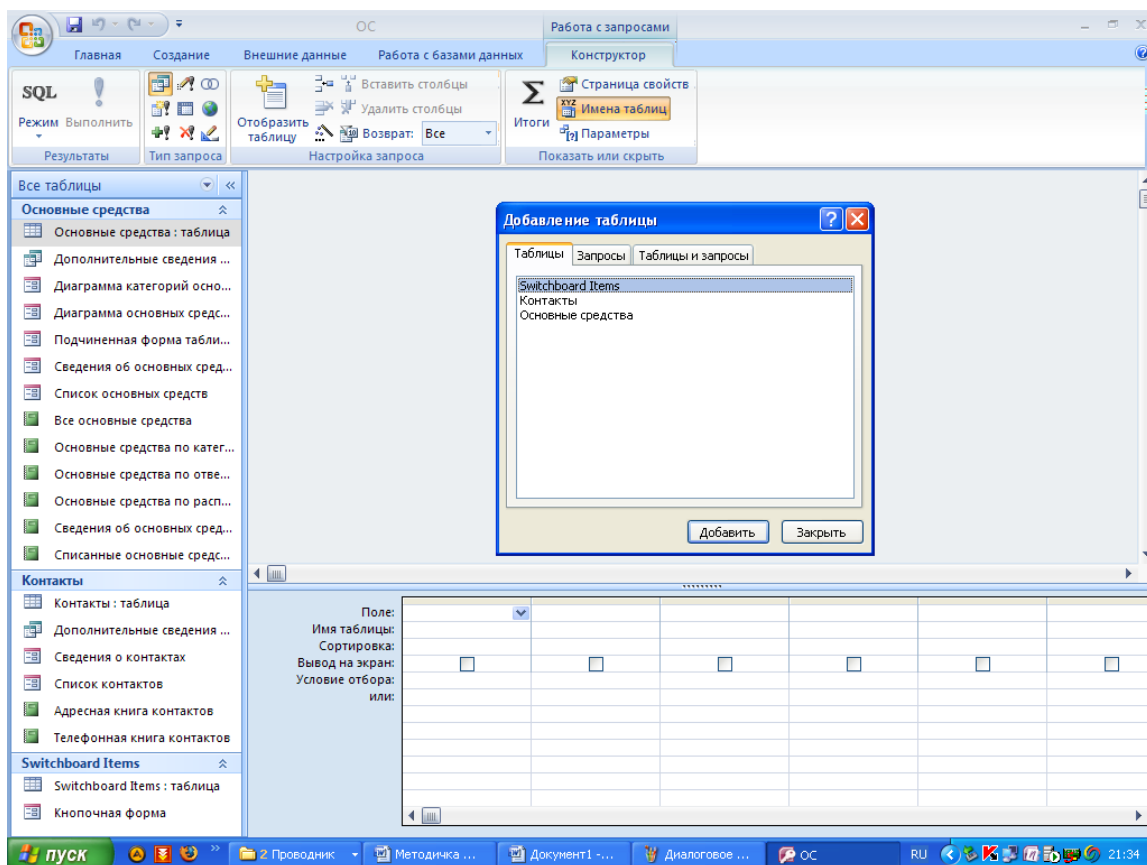


Рисунок 20 - Окно создания запроса в режиме Конструктора

4. Добавить в запрос необходимые таблицы
5. Убедиться, что между добавленными таблицами установлены связи.
6. После добавления таблиц нажать кнопку **Заккрыть** в окне **Добавление таблицы**.
7. Затем нужно указать, какие поля из базовых таблиц будут отображаться в запросе. Для этого, выделить нужное поле в таблице-источнике, подвести указатель мыши к выделенному полю, нажать на левую кнопку мыши и перетащить поле в нужное место бланка запроса.

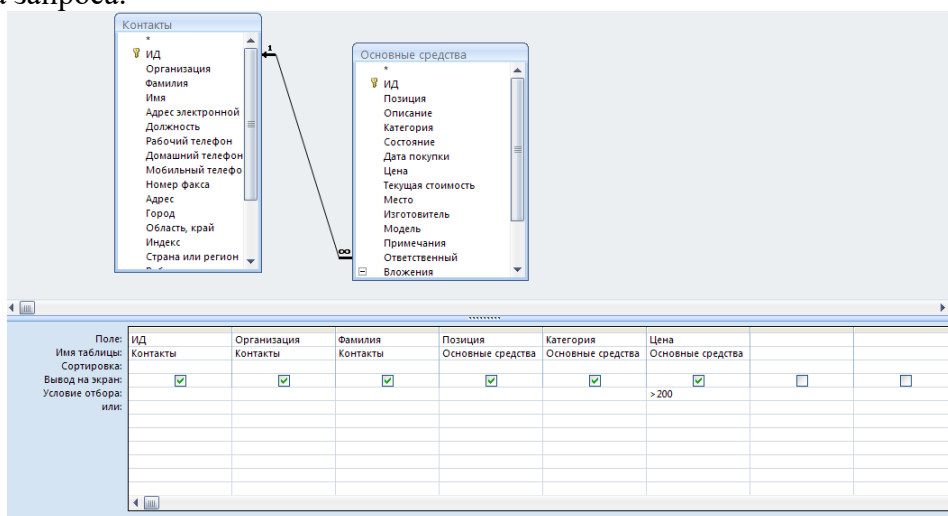


Рисунок 21 - Окно формирования запроса на выборку

8. В строке **Условие отбора** указывается необходимые условия для запроса. Например, при задании отобрать товары в ценовой категории Больше 200 в Условии отбора ставится >200

9. Сохранение запроса для дальнейшего использования производится нажатием на панели инструментов кнопки Сохранить. Далее СУБД запросит имя сохраняемого запроса. Целесообразно, чтобы оно имело смысловую нагрузку, что облегчит дальнейшее использование запроса.

Чтобы удалить лишнюю или внесенную по ошибке базовую таблицу из запроса, необходимо выделить ее, щелкнув на любом месте в списке ее полей, и нажать клавишу **Delete**. Чтобы удалить поле из запроса, выделите нужный столбец в бланке запроса, а затем нажмите клавишу **Delete**.

Самое главное в запросе - возможность использования критериев выборки, которые вводятся в строку Условие отбора. Можно выделить следующие типы запросов на основе критериев выборки:

□ **Выборка по строгому совпадению.** В строку Условие отбора для определенного поля вводится одно из значений, существующих в таблице. Например, название конкретного товара или название фирмы, города.

Данные запросы можно параметризовать, т.е. вводить условия отбора в виде параметра при каждом запуске запроса, что устраняет необходимость предварительно его модификации. Для параметризации необходимо в строке Условие отбора вместо самого условия ввести текст приглашения на его ввод по формату [текст приглашения]. Например, [Введите наименование организации].

Поле:	Организация	Позиция	Категория	Дата покупки	Цена
Имя таблицы:	Контакты	Основные средства	Основные средства	Основные средства	Основные средств
Сортировка:					
Вывод на экран:	☒	☒	☒	☒	☒
Условие отбора:	[Введите наименование организации]				
или:					

Рисунок 22 - Окно создания параметризованного запроса

Затем запрос сохраняется обычным способом. Чтобы оценить результаты сформированного запроса, необходимо его запустить двойным щелчком мыши.

Задание.

Для всех вариантов задания 1 сформировать и выполнить следующие запросы:

1. Запрос-выборку по одной таблице с использованием параметров.
2. Запрос-выборку по нескольким таблицам и запросам с использованием вычисляемых полей.
3. Запрос на обновление данных.
4. Запрос на удаление данных.

5. Запрос на создание таблицы.
6. Итоговый запрос.
7. Перекрестный запрос.

Предложенные ниже варианты запросов попытаться выполнить с помощью запросов по образцу QBE, в противном случае реализовать SQL-запросы.

Варианты заданий

Вариант 1.

1. Определить покупателя, который купил максимальное количество товаров.
2. Для каждой покупки рассчитать общую стоимость.
3. Определить сумму продажи для каждого месяца.
4. Определить покупателей, купивших товаров на сумму, превышающую среднюю сумму покупок всех покупателей.
5. Определить тип, товаров которого куплено больше всего.

Вариант 2.

1. Для каждого вида изделия рассчитать его стоимость.
2. Найти изделия, в состав которых входит больше всего компонентов.
3. Определить компоненты, которые входят в большее число изделий.
4. Вычислить прибыль от продажи каждого типа продукции.
5. Найти изделие, на сборку которого уходит дней больше, чем в среднем на сборку изделий.

Вариант 3.

1. Определить заказ, на выполнение которого ушло больше всего дней.
2. Определить клиентов, стоимость заказов которых превысила их кредит.
3. Рассчитать стоимость каждого заказа с учетом скидки.
4. Определить клиента, который купил больше всего товаров.
5. Определить город, где живет клиент, чаще других оформляющий заказы.

Вариант 4.

1. Рассчитать для каждого квартиросъемщика квартплату.
2. Определить задолжников по квартплате за каждый месяц.
3. Определить дом с максимальной жилой площадью.
4. Определить дом с максимальной плотностью населения.
5. Жильцам, просрочившим оплату жилья, назначить пени 1% за каждый просроченный день.

Вариант 5.

1. Рассчитать общую стоимость товара с учетом транспортных расходов, скидки и налога.
2. Определить прибыль от продажи за каждый месяц
3. Определить страну, в которой изготовлены компоненты, вошедшие в товар, пользующийся наибольшей популярностью.
4. Определить самый дешевый компонент, поступающий без лицензии.
5. Определить товар, в состав которого входят компоненты с максимальными транспортными расходами.

Вариант 6.

1. Вычислить оплату труда каждого сотрудника как нижняя грань оплаты, если сотрудник работает меньше года, средняя оплата, если сотрудник работает от года до пяти лет, и верхняя грань - более пяти лет.
2. Для каждого сотрудника рассчитать его ежемесячный заработок.
3. Найти сотрудника, который работает дольше других.
4. Вычислить сумму налога, которую фирма платит каждый месяц.
5. Определить сотрудников, ежемесячная оплата которых оказалась больше средней.

Вариант 7.

1. Определить услугу, пользующую наибольшей популярностью.
2. Для каждого клиента рассчитать стоимость услуг с учетом социального положения и скидок.
3. Определить доход фирмы от предоставленных услуг за каждый месяц.
4. Определить, жители города или села чаще всего обращаются в фирму.
5. Рассчитать количество и сумму предоставленных населению услуг по категориям, определенным социальным происхождением клиентов.

Вариант 8.

1. Для каждого клиента вычислить сумму оплаты междугородних разговоров.
2. Определить город, с которым чаще всего разговаривают клиенты.
3. Определить клиента, который говорит по телефону чаще и дольше других.
4. Определить время суток, на которое приходится больше всего разговоров.
5. Определить день, в который телефонная линия была занята меньше всего.

Вариант 9.

1. Вычислить стоимость и калорийность каждого блюда.
2. Определить блюдо из супов с наименьшим содержанием жиров.
3. Определить компоненты самого дорогого блюда.
4. Найти компонент, который входит в большинство блюд.
5. Определить содержание жира, белков и углеводов в самом дорогом блюде самого дешевого в среднем типа блюд.

Вариант 10.

1. Определить тематику, по которой продается больше всего книг.
2. По каждому месяцу вычислить сумму продаж.
3. Определить, книги каких авторов пользуются наибольшей популярностью, авторов мужчин или авторов-женщин.
4. Определить дни, когда было продано книг больше, чем обычно (т.е. больше среднего).
5. Какие по тематике книги пишут молодые авторы.

Контрольные вопросы (перечень вопросов по теме, на которые студент обязан знать ответы) и /или тестовые задания

1. Для чего служат запросы? Назовите типы запросов.
2. Как создать в запросе вычисляемое поле?
3. Параметризованный запрос, назначение и порядок выполнения.

Лабораторная работа № 8. Создание запросов. Выборка

Формируемые компетенции или их части

№ п/п	Содержание компетенции	Шифр
1.	Способен управлять процессами создания и использования продуктов и услуг в сфере информационно-коммуникационных технологий, в том числе разрабатывать алгоритмы и программы для их практической реализации	ОПК-3
2.	Способен организовывать взаимодействие с клиентами и партнерами в	ОПК-5

	процессе решения задач управления жизненным циклом информационных систем и информационно-коммуникационных технологий	
--	--	--

Теоретическая часть

При запуске параметризованного запроса появляется диалоговое окно (рис. 23), в котором пользователь должен ввести собственно условие отбора и нажать клавишу ОК.

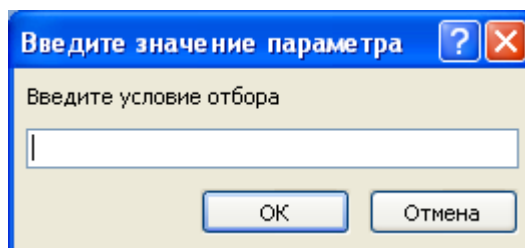


Рисунок 23 - Диалоговое окно запуска параметризованного запроса

- **Выборка по строгому несовпадению.** В этом случае в выборку отбираются все записи таблицы, кроме записей, содержащих значение, указанное в строке Условие отбора. Для реализации данного запроса перед значением вводится префикс Not или <>. Например, Not "МТФ" в поле Факультет запроса к таблице СТУДЕНТ приведет к выборке всех студентов вуза, кроме студентов МТФ.
- **Выборка по неточному совпадению.** Для выборки записей в условиях неполноты знания требуемых значений используется оператор Like<условие>.

В списке можно указывать сразу диапазон символов. Например, **Like**"[В-М]*" - выбираются все студенты, ФИО которых начинается на буквы от "В" до "М". Длина ФИО произвольная.

- **Выборка по диапазону.** Для формирования данных условий выбора используются операторы сравнения >, >=, <, <= и <>. Операции сравнения могут связываться логическими операциями And (И) и Or (ИЛИ). Для этих же целей используется оператор диапазона Between<нижнее_значение>and<верхнее_значение>. Например, выбор книг стоимостью от 100 до 200 рублей может быть реализован через ввод в запросе условия в поле Стоимость в виде >=100 and<=200 или Between100 and200.

Перечень значений в условии выборки можно задать и оператором **In**(значение, значение, ...). Например, выбор студентов факультетов МТФ или ФАПУ можно реализовать, указав в поле *Факультет* запроса условие **In**("МТФ", "ФАПУ"). Это же условие можно записать и через операцию ИЛИ: "МТФ" **or**"ФАПУ". Также можно указать одно название факультета в строке **Условие отбора** (см. рис. 2.2), а второе в следующей строке **или**. Число строк **или** не ограничено.

В выражениях отбора также можно использовать знаки математических операций +, -, /, * и неограниченное число круглых скобок. Сложные выражения в условиях отбора могут формироваться с помощью соответствующего построителя, который вызывается кнопкой на панели инструментов.

- **Запрос с вычислениями.** Такой запрос позволяет получить дополнительную информацию в процессе выборки, например, стоимость всей партии товара при хранимой в таблице информации о количестве товара и стоимости единицы его продукции. Для этого в строку Поле пустого столбца заносят выражение для вычисления по следующему формату:

<Название_формируемого_поля>:<выражение>.

Например: Стоимость партии:[Товар]![количество товара]*[стоимость единицы товара].

- **Запрос с групповыми операциями.** Рассмотренные запросы анализируют отдельные записи таблицы. Вместе с тем, СУБД Access позволяет находить интегральные показатели для групп записей в таблице. Каждая такая группа характеризуется одинаковым значением по какому-то полю, например, одинаковым названием факультета или семейным положением. Для перехода в данный режим запросов необходимо в панели инструментов нажать клавишу Групповые операции, что приведет к появлению в бланке запроса новой второй строки с одноименным названием. В ячейках данной строки указывается или режим группировки по некоторому полю (опция Группировка), или название групповой операции:

- * **Sum**- сумма значений
- * **Avg**- среднее значение по данному полю для всей группы;
- * **Count**- число записей в данной группе;
- * **Max** - максимальное значение поля в каждой группе;
- * **Min** - минимальное значение поля в каждой группе;
- * **First** - первое значение данного поля в каждой группе;
- * **Last** - последнее значение данного поля в каждой группе и др.

Опции выбора вызываются нажатием кнопки раскрытия в требуемой ячейке.

При запуске запроса СУБД разбивает таблицу на группы, число которых равно числу существующих значений в группируемом поле, и реализует для каждой группы требуемую операцию, т.е. число строк в выборке равно числу групп.

Рассмотренные запросы не изменяют содержимое исходной таблицы. Для реализации подобных действий СУБД Access использует четыре следующих запроса:

- **Запрос-создание новой таблицы.** Предназначен для сохранения результатов запроса в виде новой таблицы.

Исходно формируется обычный запрос на выборку необходимой информации из таблицы. После проверки результатов его выполнения производится возврат в режим конструктора запросов. Далее нажимается кнопка **Тип запроса**  на панели инструментов или выбирается команда главного меню

Запрос. В появившемся списке выбирается опция **Создание таблицы**, после чего СУБД запрашивает её имя. Указывается имя создаваемой таблицы и нажимается кнопка **ОК**. Непосредственно запрос на создание запускается нажатием кнопки на панели инструментов.

В окне **Таблицы БД** появляется пиктограмма созданной таблицы.

- **Запрос-добавление выборки в другую таблицу.** Выборку можно добавить к другой таблице, одностипной по структуре или с изменением структуры выборки. Для этого необходимо сформировать обычный запрос и оценить результаты его выполнения. Далее следует вернуться в режим конструктора запроса.

Нажимается кнопка **Тип запроса** на панели инструментов или выбирается команда главного меню **Запрос**. В появившемся списке выбирается опция **Добавление**, после чего СУБД запрашивает имя таблицы, к которой будет добавлена выборка. Последний шаг - нажатие кнопки **ОК**.

Выборку можно добавлять и к таблицам других БД, что определяется установкой соответствующих переключателей в окне ввода имени целевой таблицы.

Если структура выборки и целевой таблицы не совпадают, то в целевую таблицу добавляются значения только тех полей выборки, имена которых совпадают с именами полей целевой таблицы.

- *Запрос-удаление.* С помощью запросов можно удалить часть или все записи из таблицы.

Для этого необходимо сформировать обычный запрос и оценить результаты его выполнения. Далее следует вернуться в режим конструктора запроса.

Нажимается кнопка **Тип запроса** на панели инструментов или выбирается команда главного меню **Запрос**. В появившемся списке выбирается опция **Удаление**, после чего в бланке запроса появляется новая третья строка с именем **Удаление**, куда можно вводить дополнительные условия на выборку удаляемых записей. Последний шаг - нажатие кнопки **ОК**.

- *Запрос-обновление.* С помощью запросов можно обновлять в единой операции некоторые или все значения выбранных полей.

Для этого необходимо сформировать обычный запрос и оценить результаты его выполнения. Далее следует вернуться в режим конструктора запроса.

Нажимается кнопка **Тип запроса** на панели инструментов или выбирается команда главного меню **Запрос**. В появившемся списке выбирается опция **Обновление**, после чего в бланке запроса появляется новая третья строка с именем **Обновление**. В ней задаются новые значения полей таблицы, в том числе и вычисляемые выражения. Далее запрос запускается на выполнение кнопкой . СУБД указывает число модифицируемых записей и просит подтвердить изменения кнопкой **ОК**. Пользователь на этом этапе еще может отказаться от модификации значений в таблице.

В предыдущих лабораторных работах мы уже использовали так называемый **Построитель выражений**. Его используют для построения сложных выражений в Access. Так, если вам в работе потребуется применить встроенную функцию, щелкните в построителе выражений по папке **Функции** в самом левом списке, чтобы вывести категории функций.

Например, в одном из заданий вам необходимо вывести в поле запроса Фамилию + инициалы. Для этого следует использовать встроенную функцию **Left** текстового типа.

Затем вам необходимо набрать выражение

ФИО:

[Сотрудники]![ФАМИЛИЯ]+ "пробел" + Left([Сотрудники]!

[Имя];1)+ "точка" + Left([Сотрудники]![Отчество]

;1)+ "точка"

В кавычках ставится реальный пробел и реальная точка.

Таким образом, в вашем запросе будут отражаться Иванов И.И.

Задание.

Для всех вариантов задания 1 сформировать и выполнить следующие запросы:

1. Запрос-выборку по одной таблице с использованием параметров.
2. Запрос-выборку по нескольким таблицам и запросам с использованием вычисляемых полей.
3. Запрос на обновление данных.
4. Запрос на удаление данных.
5. Запрос на создание таблицы.
6. Итоговый запрос.
7. Перекрестный запрос.

Предложенные ниже варианты запросов попытаться выполнить с помощью запросов по образцу QBE, в противном случае реализовать SQL-запросы.

Варианты заданий

Вариант 1.

1. Вычислить сумму продаж по каждому месяцу.
2. Определить страну, выпустившую самый долгозвучающий диск.
3. Определить песню, пользующуюся наибольшей популярностью.
4. Составить рейтинг исполнителей по каждому месяцу.
5. Определить автора слов, написавшего больше всех песен.

Вариант 2.

1. Определить сумму продаж по каждому месяцу.
2. Определить, какой тип покупателей чаще других покупает видеокассеты.
3. Какой самый старый фильм был продан за последний месяц.
4. Определить страну, завоевавшую своими фильмами больше сего Оскаров.
5. Какие по тематике фильмы смотрит молодежь.

Вариант 3.

1. Для каждой олимпиады рассчитать отношения числа завоеванных медалей к числу участников.
2. Определить команду, для которой отношение числа завоеванных золотых медалей к числу участников больше, чем аналогичный показатель олимпиады.
3. Определить команды, которые чаще других участвовали в олимпиадах.
4. Определить команды, которые по числу завоеванных медалей на протяжении всех олимпиад попадали в первую тройку.
5. Найти олимпиады, символы которых совпадали.

Вариант 4.

1. Определить предмет, по которому нет двоек.
2. Определить студентов, сдавших успешно экзамены и набравших в сумме часов больше, чем среднее число часов по всем студентам.
3. Определить блок дисциплин, средняя оценка по которым самая высокая.
4. Определить кафедру, по предметам которой получено больше всего двоек студентами младших курсов.
5. Найти студентов, сдавших все экзамены успешно, если их день рождения пришелся на период сдачи экзаменов.

Вариант 5.

1. Вычислить зарплату каждого преподавателя.
2. Определить блок дисциплин, которые читают самые квалифицированные преподаватели.
3. Определить кафедру, для которой отношение числа предметов к числу преподавателей самое большое.
4. Определить семестр и кафедры, на которые приходится учебная нагрузка в часах, больше средней по кафедрам.
5. Определить предмет, который читается в большинстве семестров.

Вариант 6.

1. Вычислить прибыль от каждого рейса.
2. Определить рейсы до заданного пункта, на которые остались свободные места.
3. Найти отношение количества рейсов дальнего следования, которые выполняют квалифицированные команды экипажей, к общему числу рейсов дальнего следования.
4. Какой экипаж имеет больше всего налетов, по количеству и по продолжительности.
5. Какие пассажиры по своему социальному положению летают чаще других.

Вариант 7.

1. Определить водителей с плохим техническим состоянием автобуса чаще других отправляющихся в рейс.
2. Вычислить прибыль от поездок за каждый месяц.
3. Какие поездки пользуются наибольшей популярностью.
4. Для каждой организации рассчитать долг с учетом стоимости поездок и выплаченного аванса.

5. Определить водителя, совершившего больше всего поездок.

Вариант 8.

1. Для каждого вуза рассчитать объем свободных наличных средств.
2. Определить факультеты с самым большим отношением числа студентов к числу преподавателей.
3. Определить вуз с самой низкой средней стоимостью обучения одного студента.
4. Найти вуз с самым большим числом хоздоговорных студентов.
5. Определить, какая сумма приходится на каждого студента.

Вариант 9.

1. Определить, преступления какого вида раскрываются быстрее других.
2. Оружие какой страны наиболее часто используется в преступлениях.
3. Определить сколько преступлений и какого вида приходится на каждую возрастную группу.
4. В каком городе преступления раскрываются быстрее, чем в других городах.
5. В какой месяц было совершено больше всего преступлений.

Вариант 10.

1. В каком месяце была продана самая дорогая из старых машин.
2. Машины какой страны пользуются популярностью у молодежи.
3. Определить сумму продаж за каждый месяц.
4. Определить, какая возрастная группа покупает в среднем самые дорогие автомобили.
5. Какая группа по социальному положению предпочитает при расчете кредитные карточки.

Контрольные вопросы (перечень вопросов по теме, на которые студент обязан знать ответы) и /или тестовые задания

1. Как осуществляется выборка сотрудников по фамилии в диапазоне, например, от В до М?
2. Для чего используют запросы на добавление и удаление?
3. Каково назначение групповых операций при формировании запроса?
4. Для чего используют встроенные функции? Назовите перечень операций, позволяющий их использовать.

Лабораторная работа № 9. Введение ограничений целостности базы данных в СУБД Access

Формируемые компетенции или их части

№ п/п	Содержание компетенции	Шифр
1.	Способен управлять процессами создания и использования продуктов и услуг в сфере информационно-коммуникационных технологий, в том числе разрабатывать алгоритмы и программы для их практической реализации	ОПК-3
2.	Способен организовывать взаимодействие с клиентами и партнерами в процессе решения задач управления жизненным циклом информационных систем и информационно-коммуникационных технологий	ОПК-5

Теоретическая часть

Целью обеспечения целостности данных является предотвращение появления непарных записей, ссылающихся на несуществующие записи. Обеспечение целостности данных включается для конкретного отношения между таблицами. В результате Access отменяет для этого отношения все действия, которые могут нарушить целостность данных. Это означает, что будет отменено как обновление, изменяющее целевой объект ссылки, так и удаление такого целевого объекта. Сведения о том, как настроить в Access распространение операций обновления и удаления таким образом, чтобы в результате изменялись и все связанные строки.

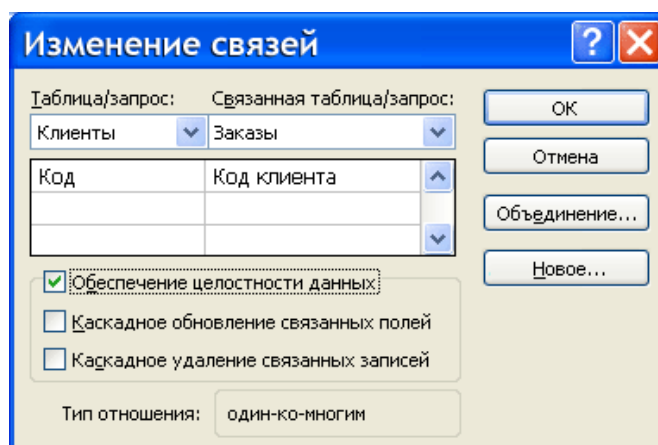


Рисунок 24 - Диалоговое окно **Изменение связей**

Включение и отключение обеспечения целостности данных

1. На вкладке **Работа с базами данных** в группе **Отношения** нажмите кнопку **Схема данных**.
2. На вкладке **Конструктор** в группе **Связи** нажмите кнопку **Все связи**. Будут отображены все таблицы с отношениями, а также соответствующие линии связи. Обратите внимание на то, что скрыты таблицы (таблицы, для которых установлен флажок **скрытый** в диалоговом окне **Свойства**) и их отношения не от

ображаются, если в диалоговом окне **Параметры переходов** выбран параметр "Показывать скрытые объекты".

3. Щелкните линию отношения, которое вы хотите изменить. При выделении линия связи становится толще.

4. Дважды щелкните линию связи. Откроется диалоговое окно

Изменение связей.

5. Выберите или отмените параметр **Обеспечение целостности данных**.

6. Внесите в отношение необходимые изменения и нажмите кнопку **ОК**.

В режиме обеспечения целостности данных действуют перечисленные ниже правила.

- Не допускается ввод в поле внешнего ключа связанной таблицы значений, отсутствующих в поле первичного ключа главной таблицы, поскольку это приводит к появлению потерянных записей.

- Не допускается удаление записи из главной таблицы, если в связанной таблице существуют связанные с ней записи. Например, невозможно удалить запись из таблицы "Сотрудники", если в таблице "Заказы" имеются заказы, относящиеся к данному сотруднику. Однако можно удалить главную запись и все связанные записи одним действием, установив флажок Каскадное удаление связанных записей.

- Не допускается изменение значения первичного ключа в главной таблице, если это приведет к появлению потерянных записей. Например, нельзя изменить номер заказа в таблице "Заказы", если в таблице "Сведения о заказах" имеются строки, относящиеся к этому заказу. Однако можно обновить главную запись и все связанные записи одним действием, установив флажок "Каскадное обновление связанных полей".

Примечания: Если при включении обеспечения целостности данных возникли трудности, обратите внимание на то, что должны выполняться условия, перечисленные ниже.

- Общее поле главной таблицы должно быть первичным ключом или иметь уникальный индекс.

- Общие поля должны иметь одинаковый тип данных. Единственное исключение — поле типа "Счетчик" можно связать с полем типа "Числовой", если его свойство Размер поля имеет значение Длинное целое.

- Обе таблицы существуют в одной базе данных Access. Обеспечение целостности данных нельзя включить для присоединенных таблиц. Однако если исходные таблицы имеют формат Access, можно открыть базу данных, в которой они хранятся, и включить обеспечение целостности данных в этой базе.

Задание каскадных параметров

Иногда возникает ситуация, в которой требуется изменить значение только на стороне "одной" отношения. В этом случае необходимо, чтобы Access автоматически обновил все затронутые строки в ходе выполнения одной операции. Тогда обновление будет полностью завершено, а база данных не будет находиться в несогласованном состоянии, когда некоторые строки обновлены, а другие — нет. Этой проблемы можно избежать с помощью параметра Access "Каскадное обновление связанных полей". Если при включении обеспечения целостности данных был включен параметр "Каскадное обновление связанных полей", то при последующем обновлении первичного ключа автоматически будут обновлены все связанные с ним поля.

Также может потребоваться удаление строки и всех связанных с ней записей — например, записи в таблице "Поставщики" и всех связанных с этим поставщиком заказов. Для этого в Access предназначен параметр "Каскадное удаление связанных записей". Если включить обеспечение целостности данных и установить флажок **Каскадное удаление связанных записей**, при удалении записи, содержащей первичный ключ, будут автоматически удалены все записи, связанные с этим первичным ключом.

Включение и отключение каскадного обновления и каскадного удаления

1. На вкладке **Работа с базами данных** в группе **Отношения** нажмите кнопку **Схема данных**.
2. На вкладке **Конструктор** в группе **Связи** нажмите кнопку **Все связи**. Будут отображены все таблицы с отношениями, а также соответствующие линии связи. Обратите внимание на то, что скрыты таблицы (таблицы, для которых установлен флажок **скрытый** в диалоговом окне **Свойства**) и их отношения не отображаются, если в диалоговом окне **Параметры переходов** не выбран параметр "Показывать скрытые объекты".
3. Щелкните линию отношения, которое вы хотите изменить. При выделении линия связи становится толще.
4. Дважды щелкните линию связи. Откроется диалоговое окно **Изменение связей**.
5. Установите флажок **Обеспечение целостности данных**.
6. Установите флажок **Каскадное обновление связанных полей**, **Каскадное удаление связанных записей** или оба эти флажка.
7. Внесите в отношение необходимые изменения и нажмите кнопку **ОК**.

Задание.

1. С помощью механизма каскадных изменений обеспечить поддержку ссылочной целостности Вашей базы данных.
2. Обеспечить семантическую поддержку целостности Вашей базы данных. Для каждого поля базы данных задать следующие (подходящие по смыслу) виды декларативных ограничений целостности:
 - ограничения целостности атрибута: значение по умолчанию, задание обязательности или необязательности значений (Null), задание условий на значения атрибутов;
 - задание значения по умолчанию.

Контрольные вопросы (перечень вопросов по теме, на которые студент обязан знать ответы) и /или тестовые задания

1. Основные сведения о фильтрах. Элементы управления.
2. Виды фильтров.
3. Построение фильтров.

Лабораторная работа № 10. Создание SQL-запросов (select) Формируемые компетенции или их части

№ п/п	Содержание компетенции	Шифр
1.	Способен управлять процессами создания и использования продуктов и услуг в сфере информационно-коммуникационных технологий, в том числе разрабатывать алгоритмы и программы для их практической реализации	ОПК-3
2.	Способен организовывать взаимодействие с клиентами и партнерами в процессе решения задач управления жизненным циклом	ОПК-5

	информационных систем и информационно-коммуникационных технологий	
--	--	--

Теоретическая часть

Запрос SQL — это запрос, создаваемый при помощи инструкций SQL. Язык SQL (StructuredQueryLanguage) используется при создании запросов, а также для обновления и управления реляционными БД. В среде MS Access, когда пользователь создает запрос в режиме конструктора запроса (с помощью построителя запросов), MS Access автоматически создает эквивалентную инструкцию SQL. Фактически, для большинства свойств запроса, доступных в окне свойств в режиме конструктора, имеются эквивалентные предложения или параметры языка SQL, доступные в режиме SQL. При необходимости, пользователь имеет возможность просматривать и редактировать инструкции SQL в режиме SQL. После внесения изменений в запрос в режиме SQL его вид в режиме конструктора может измениться. Некоторые запросы, которые называют запросами SQL, невозможно создать в бланке запроса. Для запросов к серверу, управляющих запросов и запросов на объединение необходимо создавать инструкции SQL непосредственно в окне запроса в режиме SQL. Для подчиненного запроса пользователь должен ввести инструкцию SQL в строку Поле или Условие отбора в бланке запроса. Синтаксиса написания SQL-предложений: в описании команд слова, написанные прописными латинскими буквами, являются зарезервированными словами SQL; фрагменты SQL-предложений, заключенные в фигурные скобки и разделенные символом « », являются альтернативными. При формировании соответствующей команды для конкретного случая необходимо выбрать одну из них; фрагмент описываемого SQL-предложения, заключенный в квадратные скобки [], имеет необязательный характер и может не использоваться; многоточие ..., стоящее перед закрывающейся скобкой, говорит о том, что фрагмент, указанный в этих скобках, может быть повторен;

2 Описание команд SQL Выборка записей Инструкция SELECT. При выполнении инструкции SELECT СУБД находит указанную таблицу или таблицы, извлекает заданные столбцы, выделяет строки, соответствующие условию отбора, и сортирует или группирует результирующие строки в указанном порядке в виде набора записей. Синтаксис команды:

SELECT [предикат] { * | таблица.* | [таблица.]поле_1 [AS псевдоним_2] [, [таблица.]поле_2[AS псевдоним_2] [, ...]] } FROM выражение [, ...] [WHERE...] [GROUP BY...] [HAVING...] [ORDER BY...] где предикат — один из следующих предикатов отбора: ALL, DISTINCT, DISTINCTROW, TOP.

Данные ключевые слова используются для ограничения числа возвращаемых записей. Если они отсутствуют, по умолчанию используется предикат ALL; * указывает, что результирующий набор записей будет содержать все поля заданной таблицы или таблиц. Следующая инструкция отбирает все поля из таблицы «Студенты»:

SELECT * FROM Студенты;

таблица — имя таблицы, из которой выбираются записи; поле_1, поле_2 — имена полей, из которых должны быть отобраны данные; псевдоним_1, псевдоним_2 — ассоциации, которые станут заголовками столбцов вместо исходных названий полей в таблице; выражение — имена одной или нескольких таблиц, которые содержат необходимые для отбора записи; предложение GROUP BY в SQL-предложении объединяет записи с одинаковыми значениями в указанном списке полей в одну запись.

Если инструкция SELECT содержит статистическую функцию SQL, например Sum или Count, то для каждой записи будет вычислено итоговое значение; предложение HAVING определяет, какие сгруппированные записи, выданные в результате выполнения запроса, отображаются при использовании инструкции SELECT с предложением GROUP BY. После того как записи результирующего набора будут сгруппированы с 3 помощью предложения GROUP BY, предложение HAVING отберет те из них, которые удовлетворяют условиям отбора, указанным в предложении HAVING; предложение ORDER BY позволяет отсортировать записи, полученные в результате запроса, в порядке

возрастания или убывания на основе значений указанного поля или полей. Следует отметить, что инструкции SELECT не изменяют данные в базе данных. Приведем минимальный синтаксис инструкции SELECT: SELECT поля FROM таблица. Если несколько таблиц, включенных в предложение FROM, содержат одноименные поля, перед именем такого поля следует ввести имя таблицы и оператор « . » (точка). Предположим, что поле «Номер_группы» содержится в таблицах «Студенты» и «Группы». Следующая инструкция SQL отберет поле «Номер_группы» и «ФИО_студента» из таблицы «Студенты» и «ФИО_куратора» из таблицы «Группы» при номере группы, равном 432-1:

SELECT Группы.Номер_группы, Группы.ФИО_куратора, Студенты.ФИО_студента
FROM Группы, Студенты WHERE Группы.Номер_группы = Студенты.Номер_группы
AND

На рисунке 25 приведен пример выполнения данного запроса. Таблицы БД СТУДЕНТЫ

Номер_зачет- ной книжки	ФИО_студента	Дата рождения	Место рождения	Но- мер группы
1992412-11	Карасев А.А.	27.08.75	г. Чита	412-1
1992432-11	Данилов О. В.	27.08.75	г. Алматы	432-1
1992432-12	Раевский А. И.	20.05.75	г. Бишкек	432-1
1992432-22	Глазов О.А	04.07.75	г. Киров	432-1

ГРУППЫ

Номер_Группы	ФИО_куратора
412-1	Самойлов С.С.
432-1	Авдеев Р.М

Результат выполнения запроса

Номер_группы	ФИО_куратора	ФИО_студента
432-1	Авдеев Р.М	Данилов О. В.
432-1	Авдеев Р.М	Раевский А. И.
432-1	Авдеев Р.М	Глазов О.А

Рисунок 25 - Пример выполнения запроса на выборку

Помимо обычных знаков сравнения (=, <=, >=, <>) в языке SQL в условии отбора используются ряд ключевых слов:

Isnotnull — выбрать только непустые значения;

IsNull — выбрать только пустые значения;

Between ... And определяет принадлежность значения выражения указанному диапазону.

Синтаксис:

выражение [Not] Between значение_1 And значение_2 ,

где выражение — выражение, определяющее поле, значение которого проверяется на принадлежность к диапазону;

значение_1, значение_2 – выражения, задающие границы диапазона.

Если значение поля, определенного в аргументе выражение, попадает в диапазон, задаваемый аргументами значение_1 и значение_2 (включительно), то оператор Between...And возвращает значение True; в противном случае возвращается значение False.

Логический оператор Not позволяет проверить противоположное условие: что выражение находится за пределами диапазона, заданного с помощью аргументов значение_1 и значение_2.

Оператор Between...And часто используют для проверки: попадает ли значение поля в указанный диапазон чисел.

В следующем примере выдается список студентов, получающих стипендию от 800 до 900 рублей:

SELECT ФИО_студента, Размер_стипендии

FROM Студенты

WHERE Размер_стипендии Between 800 And 900

На рисунке 26 приведен результат выполнения запроса. СТУДЕНТЫ

Номер зачетной книжки	ФИО студента	Размер стипендии
1992412-11	Карасев А.А.	900
1992432-11	Данилов О. В.	800
1992432-12	Раевский А. И.	950
1992432-22	Глазов О.А	850

Результирующий набор данных

ФИО студента	Размер стипендии
Карасев А.А.	900
Данилов О. В.	800
Глазов О.А	850

Рисунок 26 - Результат выполнения запроса на выборку с использованием операторов Between...And

Если выражение, значение_1 или значение_2 имеет значение Null, оператор Between...And возвращает значение Null.

Оператор Like используется для сравнения строкового выражения с образцом. Синтаксис: выражение Like «образец», где выражение — выражение SQL, используемое в предложении WHERE;

образец — строка, с которой сравнивается выражение.

Оператор Like используется для нахождения в поле значений, соответствующих указанному образцу. Для аргумента образец можно задавать полное значение (например, Like «Иванов») или использовать подстановочные знаки для поиска диапазона значений (например, Like "Ив*"). Приведем перечень подстановочных символов и пример их использования в языке Jet SQL согласно документации Microsoft.

Параметры оператора Like

Тип совпадения	Образец	Совпадение (True)	Несовпадение (False)
Несколько символов	a*a	aa, aBa, aBBBa	aBC
	ab	abc, AABb, Xab	aZb, bac
Специальный символ	a[*]a	a*a	aaa
Несколько символов	ab*	abcdefg, abc	cab, aab
Одиночный символ	a?a	aaa, a3a, aBa	aBBBa
Одиночная цифра	a#a	a0a, a1a, a2a	aaa, a10a
Диапазон символов	[a-z]	f, p, j	2, &
Вне диапазона	[!a-z]	9, &, %	b, a
Не цифра	[!0-9]	A, a, &, ~	0, 1, 9
Комбинированное выражение	a[!b-m]#	An9, az0, a99	abc, aj0

Задание

Все запросы, создаваемые в рамках данной лабораторной работы, должны быть реализованы на языке SQL без помощи построителя запросов. Для создания нового SQL-запроса в окне базы данных нажмите кнопку Создание запроса в режиме конструктора и не выбирая таблицы для запроса нажмите кнопку Вид на панели инструментов.

1. Используя инструкцию CREATE TABLE, создайте запрос на создание новой таблицы для выбранной ранее предметной области, 20 содержащую пять полей, различных типов данных, определив в запросе первичный ключ и проиндексировав соответствующие поля, используя предложение CONSTRAINT. Для запуска запроса нажмите кнопку Запуск на панели инструментов. После чего создайте запрос на создание еще одной таблицы, содержащей внешний ключ по отношению к первичному ключу предыдущей таблицы. Запустите запрос, после чего проверьте, отразились ли изменения в схеме данных.

2. Используя команду SELECT, создайте запрос на выборку записей из двух (или более) таблиц, используя правила внешнего и внутреннего соединения, а также различные условия отбора и сортировки.

3. Используя команду UPDATE, создайте запрос на обновление данных в созданных ранее таблицах.

4. Используя команду INSERT INTO, создайте запросы на добавление группы записей и одной записи в существующую таблицу.

5. Используя команду CREATE INDEX, создайте запрос на создание нового индекса, используя различные условия на значения индексов (IGNORE NULL, DISALLOW NULL, PRIMARY), а также типы сортировки.

6. Используя команду DROP, создайте запросы на удаление таблицы и индекса, созданных ранее в БД.

7. Сохраните все созданные запросы в базе данных.

Контрольные вопросы (перечень вопросов по теме, на которые студент обязан знать ответы) и /или тестовые задания

1. Понятие запроса. Бланк запроса. Виды запросов.

2. Установка свойств полей в запросе. Условия отбора в запросе. Создание вычисляемых полей в запросе.

**Лабораторная работа № 11. Создание SQL-запросов. Внутреннее соединение
Формируемые компетенции или их части**

№ п/п	Содержание компетенции	Шифр
1.	Способен управлять процессами создания и использования продуктов и услуг в сфере информационно-коммуникационных технологий, в том числе разрабатывать алгоритмы и программы для их практической реализации	ОПК-3
2.	Способен организовывать взаимодействие с клиентами и партнерами в процессе решения задач управления жизненным циклом информационных систем и информационно-коммуникационных технологий	ОПК-5

Теоретическая часть

Операция INNER JOIN объединяет записи из двух таблиц, если связующие поля этих таблиц содержат одинаковые значения.

Синтаксис операции:

FROM таблица_1 INNER JOIN таблица_2 ON таблица_1.поле_1

оператор таблица_2.поле_2

где таблица_1, таблица_2 — имена таблиц, записи которых подлежат объединению;

поле_1, поле_2 — имена объединяемых полей.

Поля должны иметь одинаковый тип данных и содержать данные одного рода, однако эти поля могут иметь разные имена; оператор — любой оператор сравнения: "=", ">", "<"

4)

Предикат ALL используется для отбора в главном запросе только тех записей, которые удовлетворяют сравнению со всеми записями, отобранными в подчиненном запросе. Если в предыдущем примере предикат ANY заменить предикатом ALL, результат запроса будет включать только тех студентов, у которых средний балл больше 4. Это условие является значительно более жестким. Предикат IN используется для отбора в главном запросе только тех записей, которые содержат значения, совпадающие с одним из отобранных подчиненным запросом.

Следующий пример возвращает сведения обо всех студентах, средний балл которых за семестр был больше 4.

```
SELECT *  
FROM Студенты  
WHERE Номер_зачетной_книжкиin (SELECT Номер_зачетной_книжки FROM  
Успеваемость WHERE оценка > 4)
```

Предикат NOT IN используется для отбора в главном запросе только тех записей, которые содержат значения, не совпадающие ни с одним из отобранных подчиненным запросом. Предикат EXISTS (с необязательным зарезервированным словом NOT) используется в логическом выражении для определения того, должен ли подчиненный запрос возвращать какие-либо записи. В подчиненном запросе можно использовать псевдонимы таблиц для ссылки на таблицы, перечисленные в предложении FROM, расположенном вне подчиненного запроса.

В следующем примере отбираются фамилии и имена студентов, чья стипендия равна или больше средней стипендии студентов, обучающихся в той же группе.

В данном примере таблица СТУДЕНТЫ получает псевдоним C1:

```
SELECT Фамилия, Имя, Номер_группы, Стипендия  
FROM СТУДЕНТЫ AS C1  
WHERE Стипендия >= (SELECT Avg(Стипендия)  
FROM СТУДЕНТЫ  
WHERE C1.Номер_группы = СТУДЕНТЫ.Номер_группы)  
OrderbyНомер_группы;
```

В последнем примере зарезервированное слово AS не является обязательным. Некоторые подчиненные запросы можно использовать в перекрестных запросах как предикаты (в предложении WHERE). Подчиненные запросы, используемые для вывода результатов (в списке SELECT), нельзя использовать в перекрестных запросах. Создание новой таблицы Инструкция CREATE TABLE создает новую таблицу и используется для описания ее полей и индексов. Если для поля добавлено ограничение NOT NULL, то при добавлении новых записей это поле должно содержать допустимые данные.

Синтаксис:

```
CREATE TABLE таблица (поле_1 тип [(размер)] [NOT NULL] [индекс_1] [, поле_2  
тип [(размер)] [NOT NULL] [индекс_2] [, ...]] [, CONSTRAINT составной Индекс [, ...]]),
```

где таблица — имя создаваемой таблицы; поле_1, поле_2 — имена одного или нескольких полей, создаваемых в новой таблице.

Таблица должна содержать хотя бы одно поле; тип — тип данных поля в новой таблице;

размер — размер поля в символах (только для текстовых и двоичных полей);

индекс_1, индекс_2 — предложение CONSTRAINT, предназначенное для создания простого индекса; составной Индекс — предложение CONSTRAINT, предназначенное для создания составного индекса.следующем примере создается новая таблица с двумя полями:

```
CREATE TABLE Студенты (Номер_зачетной_книжкиinteger PRIMARY KEY,  
ФИО_студента TEXT (50), Место_рождения TEXT (50));
```

В результате выполнения этого запроса будет создана таблица со следующей схемой (рис. 27):

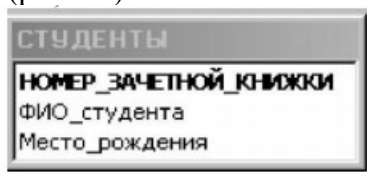


Рисунок 27 - Схема таблицы СТУДЕНТЫ

Предложение CONSTRAINT используется в инструкциях ALTER TABLE и CREATE TABLE для создания или удаления индексов.

Существуют два типа предложений CONSTRAINT: для создания простого индекса (по одному полю) и для создания составного индекса (по нескольким полям).

Синтаксис: простой индекс:

CONSTRAINT имя {PRIMARY KEY|UNIQUE | NOT NULL} составной индекс:
CONSTRAINT имя {PRIMARY KEY (ключевое_1[, ключевое_2 [, ...]]) | UNIQUE (уникальное_1[, уникальное_2 [, ...]]) | NOT NULL (непустое_1[, непустое_2 [, ...]]) | FOREIGN KEY (ссылка_1[, ссылка_2 [, ...]]) REFERENCES внешняя Таблица [(внешнее Поле_1 [, внешнее Поле_2 [, ...]])],

где имя — имя индекса, который следует создать;

ключевое_1, ключевое_2 — имена одного или нескольких полей, которые следует назначить ключевыми;

уникальное_1, уникальное_2 — имена одного или нескольких полей, которые следует включить в уникальный индекс;

непустое_1, непустое_2 — имена одного или нескольких полей, в которых запрещаются значения Null;

ссылка_1, ссылка_2 — имена одного или нескольких полей, включенных во внешний ключ, которые содержат ссылки на поля в другой таблице;

внешняя Таблица — имя внешней таблицы, которая содержит поля, указанные с помощью аргумента внешнееПоле;

внешнее Поле_1, внешнее Поле_2 — имена одного или нескольких полей во внешней Таблице, на которые ссылаются поля, указанные с помощью аргумента ссылка_1, ссылка_2. Это предложение можно опустить, если данное поле является ключом внешней Таблицы. Предложение CONSTRAINT позволяет создать для поля индекс одного из двух описанных ниже типов:

1) уникальный индекс, использующий для создания зарезервированное слово UNIQUE. Это означает, что в таблице не может быть двух записей, имеющих одно и то же значение в этом поле. Уникальный индекс создается для любого поля или любой группы полей. Если в таблице определен составной уникальный индекс, то комбинация значений включенных в него полей должна быть уникальной для каждой записи таблицы, хотя отдельные поля и могут иметь совпадающие значения;

2) ключ таблицы, состоящий из одного или нескольких полей, использующий зарезервированные слова PRIMARY KEY. Все значения ключа таблицы должны быть уникальными и не значениями Null. Кроме того, в таблице может быть только один ключ. Для создания внешнего ключа можно использовать зарезервированную конструкцию FOREIGN KEY. Если ключ внешней таблицы состоит из нескольких полей, необходимо использовать предложение CONSTRAINT. При этом следует перечислить все поля, содержащие ссылки на поля во внешней таблице, а также указать имя внешней таблицы и имена полей внешней таблицы, на которые ссылаются поля, перечисленные выше, причем в том же порядке. Однако, если последние поля являются ключом внешней таблицы, то указывать их необязательно, поскольку ядро базы данных считает, что в качестве этих полей следует использовать поля, составляющие ключ внешней таблицы.

В следующем примере создается таблица ЗАДОЛЖЕННОСТЬ_ЗА_ОБУЧЕНИЕ с единственным полем НОМЕР_ЗАЧЕТНОЙ_КНИЖКИ и внешним ключом fl_i, связанным с полем НОМЕР_ЗАЧЕТНОЙ_КНИЖКИ в таблице СТУДЕНТЫ:

```
CREATE TABLE Задолженность_за_обучение (Код_задолженностиinteger  
PRIMARY KEY, Номер_зачетной_книжкиinteger,
```

```
CONSTRAINT fl_i FOREIGN KEY (Номер_зачетной_книжки)
```

```
REFERENCES Студенты (Номер_зачетной_книжки));
```

Внешний вид схемы БД, состоящей из таблиц СТУДЕНТЫ и ЗАДОЛЖЕННОСТЬ_ЗА_ОБУЧЕНИЕ, представлен на рисунок 28.

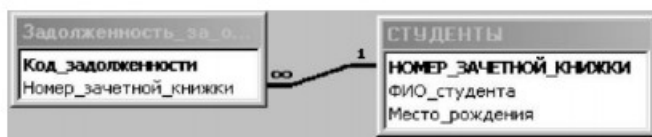


Рисунок 28 - Схема данных

Изменение структуры таблицы Инструкция ALTER TABLE изменяет структуру таблицы, созданной с помощью инструкции CREATE TABLE.

Синтаксис:

ALTER TABLE таблица {ADD {COLUMN поле тип[(размер)] [NOT NULL] [CONSTRAINT индекс] | CONSTRAINT составной Индекс} | DROP {COLUMN поле | CONSTRAINT имя Индекса} }

где таблица — имя изменяемой таблицы;

поле— имя поля, добавляемого в таблицу или удаляемого из нее;

тип — тип данных поля;

размер — размер поля;

индекс — индекс для поля;

составной Индекс — описание составного индекса, добавляемого к таблице;

имя Индекса — имя составного индекса, который следует удалить.

С помощью инструкции ALTER TABLE существующую таблицу можно изменить несколькими способами:

1) добавить новое поле в таблицу с помощью предложения ADD COLUMN. В этом случае необходимо указать имя поля, его тип и размер.

Например, следующая инструкция добавляет в таблицу СТУДЕНТЫ текстовое поле ПРИМЕЧАНИЯ длиной 50 символов: ALTER TABLE Студенты ADD COLUMN Примечания TEXT(50) Если для поля добавлено ограничение NOT NULL, то при добавлении новых записей это поле должно содержать допустимые данные;

2) добавить составной индекс с помощью зарезервированных слов ADD CONSTRAINT;

3) удалить поле с помощью зарезервированных слов DROP COLUMN. В этом случае необходимо указать только имя поля;

4) удалить составной индекс с помощью зарезервированных слов DROP CONSTRAINT. В этом случае указывается только имя составного индекса, следующее за зарезервированным словом CONSTRAINT. Создание индекса с помощью инструкции CREATE INDEX CREATE INDEX создает новый индекс для существующей таблицы.

Синтаксис команды:

CREATE [UNIQUE] INDEXиндексONтаблица (поле [ASC|DESC][, поле [ASC|DESC], ...]) [WITH { PRIMARY | DISALLOWNULL | IGNORENULL }]

гдеиндекс — имясоздаваемогоиндекса;

таблица — имя существующей таблицы, для которой создается индекс;

поле — имена одного или нескольких полей, включаемых в индекс. Для создания простого индекса, состоящего из одного поля, вводится имя поля в круглых скобках сразу после имени таблицы. Для создания составного индекса, состоящего из нескольких полей, перечисляются имена всех этих полей. Для расположения элементов индекса в убывающем порядке используется зарезервированное слово DESC; в противном случае будет принят порядок по возрастанию. Чтобы запретить совпадение значений индексированных полей в разных записях, используется зарезервированное слово UNIQUE.

Необязательное предложение WITH позволяет задать условия на значения. Например: с помощью параметра DISALLOW NULL можно запретить значения Null в индексированных полях новых записей; параметр IGNORE NULL позволяет запретить включение в индекс записей, имеющих значения Null в индексированных полях;

зарезервированное слово PRIMARY позволяет назначить индексированные поля ключом. Такой индекс по умолчанию является уникальным, следовательно, зарезервированное слово UNIQUE можно опустить. Удаление таблицы/индекса Инструкция DROP удаляет существующую таблицу из базы данных или удаляет существующий индекс из таблицы.

Синтаксис:

DROP {TABLE таблица | INDEX индекс ON таблица}

где таблица — имя таблицы, которую следует удалить или из которой следует удалить индекс;

индекс — имя индекса, удаляемого из таблицы.

Задание

Все запросы, создаваемые в рамках данной лабораторной работы, должны быть реализованы на языке SQL без помощи построителя запросов. Для создания нового SQL-запроса в окне базы данных нажмите кнопку Создание запроса в режиме конструктора и не выбирая таблицы для запроса нажмите кнопку Вид на панели инструментов.

1. Используя инструкцию CREATE TABLE, создайте запрос на создание новой таблицы для выбранной ранее предметной области, 20 содержащую пять полей, различных типов данных, определив в запросе первичный ключ и проиндексировав соответствующие поля, используя предложение CONSTRAINT. Для запуска запроса нажмите кнопку Запуск на панели инструментов. После чего создайте запрос на создание еще одной таблицы, содержащей внешний ключ по отношению к первичному ключу предыдущей таблицы. Запустите запрос, после чего проверьте, отразились ли изменения в схеме данных.

2. Используя команду SELECT, создайте запрос на выборку записей из двух (или более) таблиц, используя правила внешнего и внутреннего соединения, а также различные условия отбора и сортировки.

3. Используя команду UPDATE, создайте запрос на обновление данных в созданных ранее таблицах.

4. Используя команду INSERT INTO, создайте запросы на добавление группы записей и одной записи в существующую таблицу.

5. Используя команду CREATE INDEX, создайте запрос на создание нового индекса, используя различные условия на значения индексов (IGNORE NULL, DISSALLOW NULL, PRIMARY), а также типы сортировки.

6. Используя команду DROP, создайте запросы на удаление таблицы и индекса, созданных ранее в БД.

7. Сохраните все созданные запросы в базе данных.

Контрольные вопросы (перечень вопросов по теме, на которые студент обязан знать ответы) и /или тестовые задания

1. Свойства полей. Размер поля. Формат поля. Маска ввода. Подпись. Условие на значение.
2. Итоговые запросы. Перекрестный запрос. Многотабличные запросы.
3. Модификация данных с помощью запроса.

Лабораторная работа № 12. Создание SQL-запросов. Удаление таблиц Формируемые компетенции или их части

№ п/п	Содержание компетенции	Шифр
1.	Способен управлять процессами создания и использования продуктов и услуг в сфере информационно-коммуникационных технологий, в том числе разрабатывать алгоритмы и программы для их практической реализации	ОПК-3
2.	Способен организовывать взаимодействие с клиентами и партнерами в	ОПК-5

	процессе решения задач управления жизненным циклом информационных систем и информационно-коммуникационных технологий	
--	--	--

Теоретическая часть

Прежде чем удалить таблицу или удалить из нее индекс, необходимо ее закрыть. Следует отметить, что таблица удаляется из базы данных безвозвратно. Удаление записей Инструкция DELETE создает запрос на удаление записей из одной или нескольких таблиц, перечисленных в предложении FROM и удовлетворяющих предложению WHERE. Синтаксис команды: DELETE [Таблица.*] FROM таблица WHERE условие Отбора где Таблица — необязательное имя таблицы, из которой удаляются записи; таблица — имя таблицы, из которой удаляются записи; условие Отбора — выражение, определяющее удаляемые записи.

С помощью инструкция DELETE можно осуществлять удаление большого количества записей. Данные из таблицы также можно удалить и с помощью инструкции DROP, однако при таком удалении теряется структура таблицы. Если же применить инструкцию DELETE, удаляются только данные. При этом сохраняются структура таблицы и все остальные ее свойства, такие, как атрибуты полей и индексы. Запрос на удаление удаляет записи целиком, а не только содержимое указанных полей. Нельзя восстановить записи, удаленные с помощью запроса на удаление. Чтобы узнать, какие записи будут удалены, необходимо посмотреть результаты запроса на выборку, использующего те же самые условие отбора в предложении Where, а затем выполнить запрос на удаление. Добавление записей Инструкция INSERT INTO добавляет запись или записи в таблицу.

Синтаксис команды: а) запрос на добавление нескольких записей:

INSERT INTO назначение [(поле_1[, поле_2[, ...]])] SELECT [источник.]поле_1[, поле_2[, ...] FROM выражение

б) запрос на добавление одной записи:

INSERT INTO назначение [(поле_1[, поле_2[, ...]])] VALUES (значение_1[, значение_2[, ...])

где назначение — имя таблицы или запроса, в который добавляются записи;

источник — имя таблицы или запроса, откуда копируются записи;

поле_1, поле_2 — имена полей для добавления данных, если они следуют за аргументом «Назначение»;

имена полей, из которых берутся данные, если они следуют за аргументом источник;

выражение — имена таблицы или таблиц, откуда вставляются данные.

Это выражение может быть именем отдельной таблицы или результатом операции INNER JOIN, LEFT JOIN или RIGHT JOIN, а также сохраненным запросом; значение_1, значение_2 — значения, добавляемые в указанные поля новой записи. Каждое значение будет вставлено в поле, занимающее то же положение в списке: значение_1 вставляется в поле_1 в новой записи, значение_2 — в поле_2 и т.д. Каждое значение текстового поля следует заключать в кавычки (' '), для разделения значений используются запятые. Инструкцию INSERT INTO можно использовать для добавления одной записи в таблицу с помощью запроса на добавление одной записи, описанного выше. В этом случае инструкция должна содержать имя и значение каждого поля записи. Нужно определить все поля записи, в которые будет помещено значение, и значения для этих полей. Если поля не определены, в недостающие столбцы будет вставлено значение по умолчанию или значение Null. Записи добавляются в конец таблицы. Инструкцию INSERT INTO можно также использовать для добавления набора записей из другой таблицы или запроса с помощью предложения SELECT ... FROM, как показано выше в запросе на добавление нескольких записей.

В этом случае предложение SELECT определяет поля, добавляемые в указанную таблицу НАЗНАЧЕНИЕ. Инструкция INSERT INTO является необязательной, однако, если она присутствует, то должна находиться перед инструкцией SELECT. Запрос на добавление записей копирует записи из одной или нескольких таблиц в другую таблицу. Таблицы, которые содержат добавляемые записи, не изменяются. Вместо добавления существующих записей из другой таблицы, можно указать значения полей одной новой записи с помощью предложения VALUES. Если список полей опущен, предложение VALUES должно содержать значение для каждого поля таблицы; в противном случае инструкция INSERT не будет выполнена.

Можно использовать дополнительную инструкцию INSERT INTO с предложением VALUES для каждой добавляемой новой записи. Обновление данных Инструкция UPDATE создает запрос на обновление, который изменяет значения полей указанной таблицы на основе заданного условия отбора. Синтаксис команды: UPDATE таблица SET новое Значение WHERE условие Отбора; где таблица — имя таблицы, данные в которой следует изменить; новое Значение — выражение, определяющее значение, которое должно быть вставлено в указанное поле обновленных записей; условие Отбора — выражение, отбирающее записи, которые должны быть изменены. При выполнении этой инструкции будут изменены только записи, удовлетворяющие указанному условию.

Инструкцию UPDATE особенно удобно использовать для изменения сразу нескольких записей или в том случае, если записи, подлежащие изменению, находятся в разных таблицах. Одновременно можно изменить значения нескольких полей. Следующая инструкция SQL увеличивает стипендию студентов группы 422-1 на 10 %:

```
UPDATE Студенты SET Стипендия = стипендия * 1.1
```

```
WHERE Номер_группы = '422-1';
```

Запрос на объединение

Операция UNION создает запрос на объединение, который объединяет результаты нескольких независимых запросов или таблиц.

Синтаксис команды:

```
[TABLE] запрос_1 UNION [ALL] [TABLE] запрос_2 [UNION[ALL] [TABLE]
запрос_n [...]]
```

где запрос_1-п — инструкция SELECT или имя сохраненной таблицы, перед которым стоит зарезервированное слово TABLE. В одной операции UNION можно объединить в любом наборе результаты нескольких запросов, таблиц и инструкций SELECT. В следующем примере объединяется существующая таблица СТУДЕНТЫ и инструкции SELECT:

```
TABLE Студенты UNION ALL SELECT * FROM Абитуриенты
```

```
WHERE Общий_балл > 22;
```

По умолчанию повторяющиеся записи не возвращаются при использовании операции UNION, однако в нее можно добавить предикат ALL, чтобы гарантировать возврат всех записей. Кроме того, такие запросы выполняются несколько быстрее. Все запросы, включенные в операцию UNION, должны отбирать одинаковое число полей; при этом типы данных и размеры полей не обязаны совпадать. Псевдонимы необходимо использовать только в первом предложении SELECT, в остальных они пропускаются.

В каждом аргументе «Запрос» допускается использование предложения GROUP BY или HAVING для группировки возвращаемых данных. В конец последнего аргумента «Запрос» можно включить предложение ORDER BY, чтобы отсортировать возвращенные данные. Основные различия MicrosoftJet SQL и ANSI SQL Рассмотрим различия двух диалектов языка SQL MicrosoftJet SQL и ANSI SQL, описанные в руководстве разработчика приложений баз данных на СУБД MicrosoftAccess: языки SQL-ядра базы данных MicrosoftJet и ANSI SQL имеют разные наборы зарезервированных слов и типов данных; разные правила применимы к оператору Between...And, имеющему следующий синтаксис: выражение [NOT] Between значение_1 And значение_2. В языке SQL

MicrosoftJet значение_1 может превышать значение_2; в ANSI SQL, значение_1 должно быть меньше или равно значению_2;

разные подстановочные знаки используются с оператором Like. Так, в языке SQL MicrosoftJet любой одиночный символ изображается знаком «?», а в ANSI SQL знаком «_», любое число символов в языке SQL MicrosoftJet изображается знаком «*», а в ANSI SQL — знаком «%»; язык SQL-ядра MicrosoftJet обычно предоставляет пользователю большую свободу. Например, разрешается группировка и сортировка по выражениям. Особые средства языка SQL MicrosoftJet В языке SQL MicrosoftJet поддерживаются следующие дополнительные средства: инструкция TRANSFORM, предназначенная для создания перекрестных запросов; дополнительные статистические функции, такие как StDev и VarP; описание PARAMETERS, предназначенное для создания запросов с параметрами. Средства ANSI SQL, не поддерживаемые в языке SQL MicrosoftJet В языке SQL MicrosoftJet не поддерживаются следующие средства ANSI SQL: инструкции, управляющие защитой, такие как COMMIT, GRANT и LOCK; зарезервированное слово DISTINCT в качестве описания аргумента статистической функции (например, нельзя использовать выражение SUM(DISTINCT имя Столбца); предложение LIMIT TO nn ROWS, используемое для ограничения количества строк, возвращаемых в результате выполнения запроса. Для ограничения количества возвращаемых запросом строк можно использовать только предложение WHERE.

Задание

Все запросы, создаваемые в рамках данной лабораторной работы, должны быть реализованы на языке SQL без помощи построителя запросов. Для создания нового SQL-запроса в окне базы данных нажмите кнопку Создание запроса в режиме конструктора и не выбирая таблицы для запроса нажмите кнопку Вид на панели инструментов.

1. Используя инструкцию CREATE TABLE, создайте запрос на создание новой таблицы для выбранной ранее предметной области, 20 содержащую пять полей, различных типов данных, определив в запросе первичный ключ и проиндексировав соответствующие поля, используя предложение CONSTRAINT. Для запуска запроса нажмите кнопку Запуск на панели инструментов. После чего создайте запрос на создание еще одной таблицы, содержащей внешний ключ по отношению к первичному ключу предыдущей таблицы. Запустите запрос, после чего проверьте, отразились ли изменения в схеме данных.

2. Используя команду SELECT, создайте запрос на выборку записей из двух (или более) таблиц, используя правила внешнего и внутреннего соединения, а также различные условия отбора и сортировки.

3. Используя команду UPDATE, создайте запрос на обновление данных в созданных ранее таблицах.

4. Используя команду INSERT INTO, создайте запросы на добавление группы записей и одной записи в существующую таблицу.

5. Используя команду CREATE INDEX, создайте запрос на создание нового индекса, используя различные условия на значения индексов (IGNORE NULL, DISSALLOW NULL, PRIMARY), а также типы сортировки.

6. Используя команду DROP, создайте запросы на удаление таблицы и индекса, созданных ранее в БД.

7. Сохраните все созданные запросы в базе данных.

Контрольные вопросы (перечень вопросов по теме, на которые студент обязан знать ответы) и /или тестовые задания

1. Итоговые запросы. Перекрестный запрос. Многотабличные запросы.
2. Модификация данных с помощью запроса.

Лабораторная работа № 13. Создание концептуальной модели данных в среде автоматизированного проектирования. Выбор модели, атрибуты сущности.

Формируемые компетенции или их части

№ п/п	Содержание компетенции	Шифр
1.	Способен управлять процессами создания и использования продуктов и услуг в сфере информационно-коммуникационных технологий, в том числе разрабатывать алгоритмы и программы для их практической реализации	ОПК-3
2.	Способен организовывать взаимодействие с клиентами и партнерами в процессе решения задач управления жизненным циклом информационных систем и информационно-коммуникационных технологий	ОПК-5

Теоретическая часть

Каждая сущность является множеством подобных индивидуальных объектов, называемых экземплярами. Каждый экземпляр индивидуален и должен отличаться от всех остальных экземпляров. Атрибут выражает определенное свойство объекта. С точки зрения БД (физическая модель) сущности соответствует таблица, экземпляру сущности – строка в таблице, а атрибуту – колонка таблицы. Построение модели данных предполагает определение сущностей и атрибутов, т.е. необходимо определить, какая информация будет храниться в конкретной сущности или атрибуте. Сущность можно определить как объект, событие или концепцию, информация о которой должна сохраняться. Сущности должны иметь наименование с четким смысловым значением, именоваться существительным в единственном лице, не носить «технических» наименований и быть достаточно важными для того, чтобы их моделировать. Именование сущности в единственном числе облегчает в дальнейшем чтение модели. Фактически имя сущности дается по имени ее экземпляра. Каждая сущность должна быть полностью определена с помощью текстового описания. Каждый атрибут хранит информацию об определенном свойстве сущности, а каждый экземпляр сущности должен быть уникальным. Атрибут или группа атрибутов, которые идентифицируют сущность, называется первичным ключом. При установлении связей между сущностями атрибуты первичного ключа родительской сущности мигрируют в качестве внешних ключей в дочернюю сущность. Очень важно дать атрибуту правильное имя. Атрибуты должны именоваться в единственном числе и иметь четкое смысловое значение. Соблюдение этого правила позволяет частично решить проблему нормализации данных уже на этапе определения атрибутов. Связи Связь является логическим соотношением между сущностями. Каждая связь должна именоваться глаголом или глагольной фразой. Имя связи выражает некоторое ограничение или бизнес-правило и облегчает чтение построенной модели данных. Различают зависимые и независимые сущности. Тип сущности определяется ее связью с другими сущностями. Идентифицирующая связь устанавливается между независимой (родительский конец связи) и зависимой (дочерний конец связи) сущностями. При установлении идентифицирующей связи атрибуты первичного ключа родительской сущности переносятся в состав первичного ключа дочерней сущности. Эта операция дополнения атрибутов дочерней сущности при создании связи называется миграцией атрибутов. В дочерней сущности атрибуты помечаются как внешний ключ (FK). При установлении неидентифицирующей связи дочерняя сущность остается независимой, а атрибуты первичного ключа родительской сущности мигрируют в состав неключевых компонентов родительской сущности. Неидентифицирующая связь служит для связывания независимых сущностей. Имя связи – фраза, характеризующая отношение между родительской и дочерней сущностями. Для связи один-ко-многим

идентифицирующей или не идентифицирующей достаточно указать имя, характеризующее отношение от родительской к дочерней сущности. Тип связи (идентифицирующая/неидентифицирующая). Для неидентифицирующей связи можно указать обязательность. В случае обязательной связи атрибут внешнего ключа получит признак NOT NULL, несмотря на то, что внешний ключ не войдет в состав первичного ключа дочерней сущности. В случае необязательной связи внешний ключ может принимать значение NULL. Необязательная неидентифицирующая связь помечается прозрачным ромбиком со стороны родительской сущности. Правила ссылочной целостности – логические конструкции, которые выражают бизнес-правила использования данных и представляют собой правила вставки, замены и удаления. Информацию о предметной области суммируют составлением спецификаций по сущностям, атрибутам и отношениям с использованием графических диаграмм, в чем и заключается процесс моделирование данных. Основы работы в PowerDesigner Для запуска пакета PowerDesigner в меню программы (Windows) найдите папку Sybase и запустите файл PowerDesigner. Для создания концептуальной модели данных необходимо выбрать File/ New или на панели инструментов выбрать значок . Далее появится окно для выбора создаваемой модели (рис. 29), в котором надо выбрать ConceptualDataModel

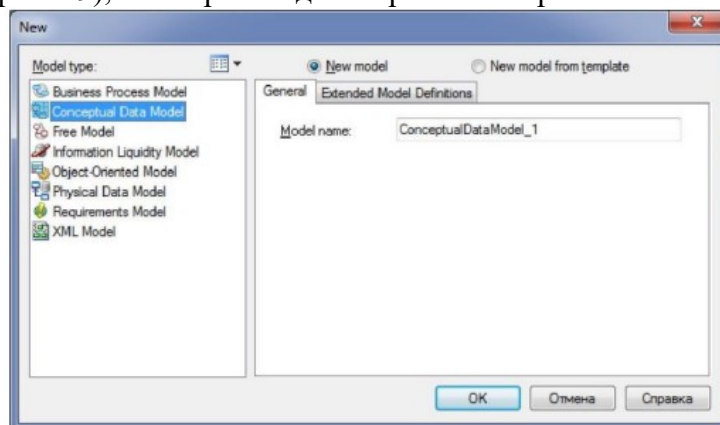


Рисунок 29 - Окно выбора модели.

После нажатия кнопки ОК появиться окно, в котором создается ER-диаграмма. Создание сущностей Для создания сущности, в панели Palette (рис. 30) нажмите кнопку с белым прямоугольником (с подсказкой Entity).



Рисунок 30 - Панель элементов с выбранным элементом сущность

Далее, поместите указатель мыши на рабочее поле в нужном месте и щелкните кнопкой мыши. Прямоугольник, изображающий сущность появится в указанном месте. При этом, курсор мыши на рабочем поле выглядит как выбранный элемент, т.о. можно создавать несколько выбранных элементов одного типа без повторного их выбора на панели элементов. Для того, чтобы изменить свойства созданной сущности, дважды щелкните на нее левой кнопкой мыши или нажмите правую кнопку и в выпавшем меню, выберите пункт Properties, в результате чего откроется окно свойств сущности (рис. 31).

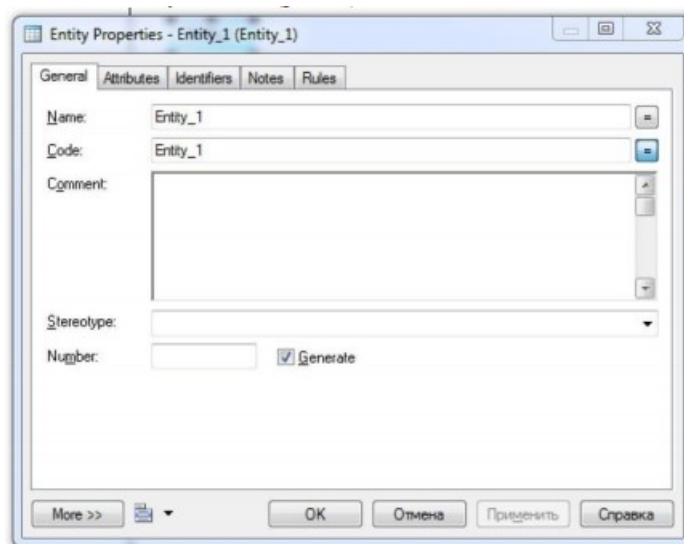


Рисунок 31 - Окно свойств сущности

В открывшемся окне пять вкладок. Зкладка General позволяет ввести следующие основные параметры: Name — имя сущности, которое будет видеть пользователь; Code — имя кода сущности, которое будет использоваться при генерации физической модели; Number — ограничение количества записей в таблице послегенерации физической модели; Comment — комментарий, предназначенный для улучшения понимания модели.

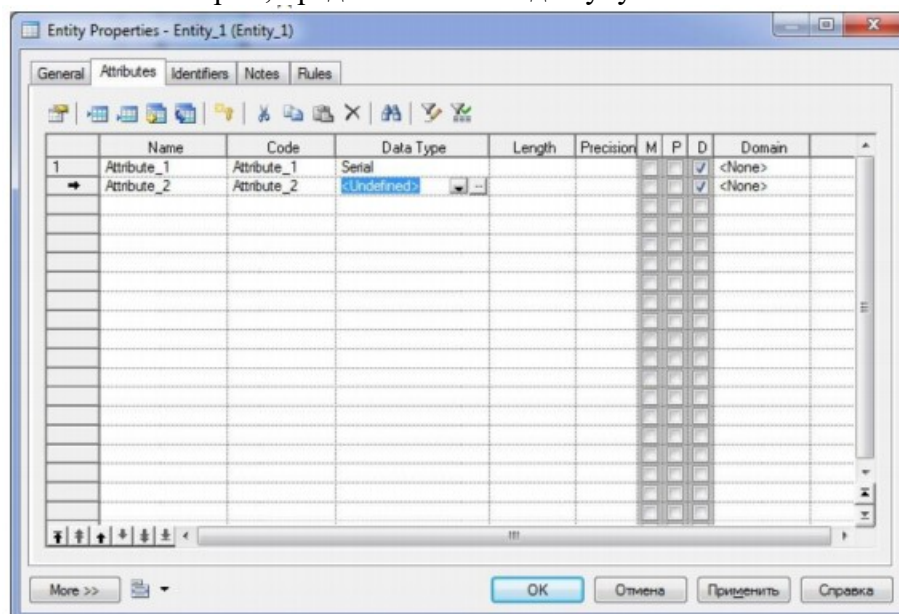


Рисунок 32 - Окно ввода атрибутов сущностей

Зкладка Attributes содержит таблицу (рис. 32) и позволяет определять атрибуты сущности: Name — имя атрибута, которое будет видеть пользователь; Code — имя кода атрибута, которое будет использоваться при генерации физической модели; Datatype — тип данных атрибута, который может быть выбран из выпадающего списка или вручную, щелкнув в поле Datatype; Domain — принадлежность к домену, если он определен. Использование доменов позволяет, определив один раз пользовательский тип данных, использовать его в дальнейшем при определении типа данных атрибута. О создании домена будет сказано ниже; M (mandatory) — обязательный атрибут, указывает может ли данный атрибут принимать неопределенные значения (обязательно ли данное поле для заполнения в таблице БД); P (PrimaryIdentifier) — первичный идентификатор сущности (в физической модели данных атрибут будет являться первичным ключом или его составной

частью); D (Displayed) — отображаемый, т.е. будет ли атрибут показываться в модели. Более полную информацию по свойствам атрибута можно получить, дважды щелкнув по полю, расположенному слева от поля с именем атрибута. Здесь можно вводить комментарий в поле Comment, задавать список значений для данного атрибута, определять верхние и нижние границы значений. Закладка Identifiers содержит автоматически заполняемую таблицу первичных идентификаторов сущности, но позволяет делать это вручную, когда необходимо создать суррогатный первичный идентификатор. Закладка Rules позволяет вводить необходимые правила на ввод значений в таблицу. Остальные закладки Notes, Versioninfo носят описательный характер для улучшения понимания модели. Для фиксации всех изменений необходимо нажать кнопку Apply.

Задание

1. Создайте концептуальную модель выбранной предметной области, учитывая следующие требования:

- a. Должно быть создано не менее 8-ми сущностей.
- b. В каждой сущности (за исключением справочников-классификаторов) должно быть создано не менее 5-ти атрибутов.

c. Необходимо использовать домены для определения типов данных атрибутов.

2. Выполните проверку качества созданной модели. Работа считается полностью выполненной, если при проверке модели не выдаются ошибки.

Контрольные вопросы (перечень вопросов по теме, на которые студент обязан знать ответы) и /или тестовые задания

1. Создание кнопок. Связывание форм с помощью командных кнопок.
2. Функции работы с датами в ACCESS.

Лабораторная работа № 14. Создание концептуальной модели данных в среде автоматизированного проектирования. Домен, окно концептуальной модели

Формируемые компетенции или их части

№ п/п	Содержание компетенции	Шифр
1.	Способен управлять процессами создания и использования продуктов и услуг в сфере информационно-коммуникационных технологий, в том числе разрабатывать алгоритмы и программы для их практической реализации	ОПК-3
2.	Способен организовывать взаимодействие с клиентами и партнерами в процессе решения задач управления жизненным циклом информационных систем и информационно-коммуникационных технологий	ОПК-5

Теоретическая часть

Домен Домен – это множество допустимых значений атрибута определенного типа данных. Домен определяется заданием стандартного типа данных, к которому относятся элементы домена и заданием произвольного логического выражения, применяемому к этому типу данных. Для создания домена необходимо в главном меню выбрав пункт Model, выбрать пункт Domains. На рис. 33 показано форма определения домена. Данная форма содержит аналогичные поля как для определения свойств атрибутов (в закладке Attributes в свойствах формы), а также дополнительные поля Length и Precision для описания длины и точности значений атрибутов.



Рисунок 35 - Панель элементов с выбранным элементом связь

Необходимо перевести курсор мыши на одну сущность и, нажав левую кнопку мыши и, не отпуская ее, перевести курсор на вторую сущность. Далее можно отпустить кнопку мыши – связь установлена. Для изменения свойств связи, необходимо дважды щелкнуть левой кнопкой мыши на линию связи (или нажать правую кнопку мыши и выпавшем меню выбрать пункт Properties). Откроется окно (рис. 36), с закладками: закладки Notes и Versioninfo используются для подробного описания связи. В закладке Rules можно задавать параметры ограничения связи; в закладке General указывается имя и код связи, а две кнопки с именами используемых сущностей позволяют вызвать окно со свойствами соответствующей сущности. Закладка Cardinalities позволяет указать вид связи (один–к–одному, один–ко–многим, многие–ко–многим и т.д.) и устанавливает свойства связи от Сущности1 к Сущности2 и наоборот: Mandatory определяет обязательность связи, показывая, что экземпляр Сущности1 (запись) может существовать только при наличии соответствующего экземпляра в Сущности2;

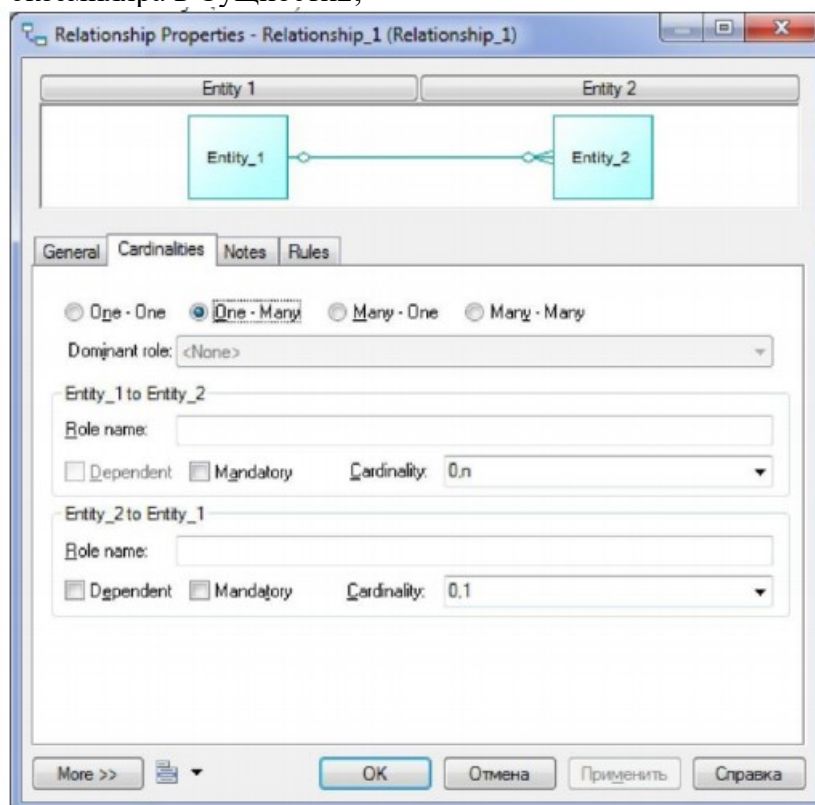


Рисунок 36 - Окно свойств связи

Dependent показывает, что каждый экземпляр Сущности1 отождествляется с экземпляром в Сущности2 (первичный ключ на стороне «один» при создании физической модели войдет в состав первичного ключа в таблице на стороне «многие»); Role – текст, описывающий связь от Сущности1 к Сущности2. Связи многие-ко-многим преобразуются в физической модели в промежуточные таблицы. После того, как созданы все сущности, указаны атрибуты и установлены все связи необходимо проверить концептуальную правильность построения концептуальной модели. Для этого необходимо выбрать в меню Tools/CheckModel (или нажать F4). Появится окно (рис. 37), в котором предлагается выбрать объекты для проверки. Package – система проверит правильность циклических связей. Domain – система проверит правильность заполнения доменов. Dataitems – проверять ли атрибуты. Entities – система проверит правильность создания сущностей. Entityattributes – проверка правильности свойств сущности Entityidentifier – проверка правильности идентификаторов сущности. Relationships – проверка связей.

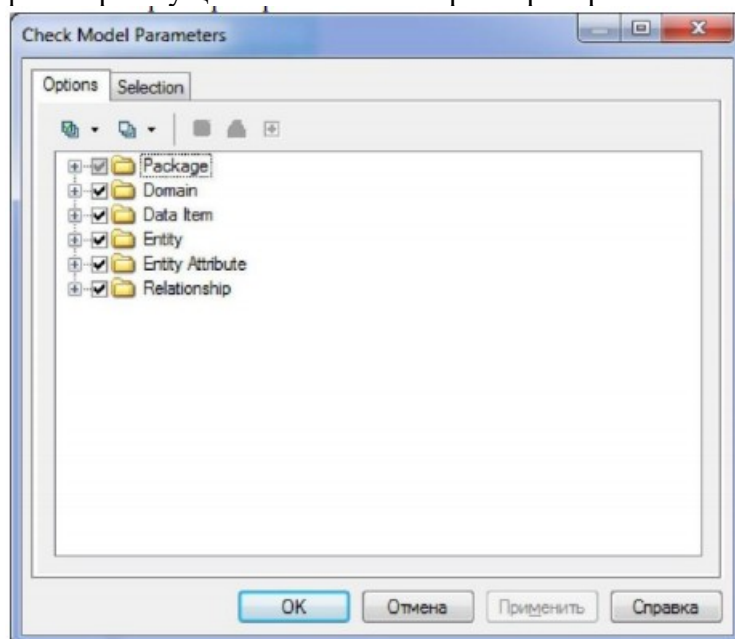


Рисунок 37 - Окно проверки концептуальной модели

После нажатия кнопки ОК система проверит всю концептуальную модель, выдаст ошибки (или предупреждения), если таковые имеются. Для просмотра сведений об ошибке необходимо дважды «щелкнуть» по ней кнопкой мыши.

Задание

1. Создайте концептуальную модель выбранной предметной области, учитывая следующие требования:

- a. Обязательно соблюдение 3-й нормальной формы для проектируемых сущностей.
- b. Имена полей должны быть представлены по-русски, коды полей должны быть написаны латиницей;
- c. В концептуальной модели данных не должно быть создано внешних ключей;
- d. Необходимо корректно задать имена связей.

2. Выполните проверку качества созданной модели. Работа считается полностью выполненной, если при проверке модели не выдаются ошибки.

Контрольные вопросы (перечень вопросов по теме, на которые студент обязан знать ответы) и /или тестовые задания

1. Построение отчетов.
2. Главная кнопочная форма.

Лабораторная работа № 15. Соединения таблиц

Формируемые компетенции или их части

№ п/п	Содержание компетенции	Шифр
1.	Способен управлять процессами создания и использования продуктов и услуг в сфере информационно-коммуникационных технологий, в том числе разрабатывать алгоритмы и программы для их практической реализации	ОПК-3
2.	Способен организовывать взаимодействие с клиентами и партнерами в процессе решения задач управления жизненным циклом информационных систем и информационно-коммуникационных технологий	ОПК-5

Теоретическая часть

Данные часто хранятся в несколько связанных таблицах. Для выбора данных используются разные методы соединения таблиц.

Когда выбор данных осуществляется из нескольких таблиц, в конструкции SELECT для каждого поля указывается таблица в виде <таблица>.<поле>. Если название поля уникальное, то можно его указать без таблицы, иначе это обязательно, чтобы избежать коллизий. Чтобы не повторить длинные названия таблиц, можно использовать псевдоним для таблиц. Псевдоним указывается в конструкции как FROM.

При неявном соединении таблиц формат конструкций FROM и WHERE имеет следующий вид:

FROM <таблица1> [псевдоним1], <таблица2> [псевдоним2]... [WHERE <условие_соединения> [AND <условие_поиска>]...]

Таким образом, если более одной таблицы присутствует в конструкции FROM, то их разделяют запятой.

Если условие соединения не указывать, тогда результат будет декартовым произведением, то есть для каждой строки одной из таблиц берутся все возможные сочетания строк из других таблиц.

Для явного соединения таблиц используется команда JOIN. У явного соединения есть следующие разновидности:

– Внутреннее соединение – осуществляется с помощью команды INNER JOIN. Из двух таблиц берутся только связанные строки. INNER JOIN имеет следующий формат записи:

<таблица1> INNER JOIN <таблица2> ON <таблица1>.<связующее_поле> = <таблица2>.<связующее_поле>.

При внутреннем соединении слово INNER можно пропустить.

– Внешнее соединение – осуществляется с помощью команды OUTER JOIN. OUTER JOIN имеет следующий формат записи:

<таблица1> LEFT | RIGHT | FULL OUTER JOIN <таблица2> ON <таблица1>.<связующее_поле> = <таблица2>.<связующее_поле>.

У внешнего соединения есть три разновидности:

– Левое внешнее соединение LEFT OUTER JOIN – из таблицы, название которой является левым операндом команды JOIN, выбираются все строки, из второй таблицы – только те записи, которые имеют связь с первой таблицей.

– Правое внешнее соединение RIGHT OUTER JOIN – из таблицы, название которой является правым операндом команды JOIN, выбираются все строки, из первой таблицы только те записи, которые имеют связь со второй таблицей.

– Полное внешнее соединение FULL OUTER JOIN – из обеих таблиц

выбираются все строки.

При использовании внешних соединений, слово OUTER можно пропустить.

– Перекрестное соединение CROSS JOIN – декартово произведение двух таблиц.

CROSS JOIN имеет следующий формат записи:

<таблица1> CROSS JOIN <таблица2>.

При выборе данных из трех и более таблиц, с помощью явного соединения, результат зависит от порядка соединения.

Используя соединение, можно связать таблицу с собой. При таком соединении псевдо- ним обязателен.

Набор данных, полученных из нескольких таблиц, не отличается от набора, получен- ного из одной таблицы. Ему тоже можно применить группировки и т.д.

1. Практическая часть

Даны следующие таблицы:

Таблица 1. Факультет

Аббревиатура	Название
Ен	Естественные науки
Гн	Гуманитарные науки
Ит	Информационные технологии
Фм	Физико-математический

Таблица 2. Кафедра

Шифр	Название	Факультет
вм	Высшая математика	ен
ис	Информационные системы	ит
мм	Математическое моделирование	фм
оф	Общая физика	ен
пи	Прикладная информатика	ит
эф	Экспериментальная физика	фм

Таблица 3. Сотрудник

Таб_номер	Шифр	Фамилия	Должность	Зарплата	Шеф
101	пи	Прохоров П.П.	зав. кафедрой	35 000,00 р.	101
102	пи	Семенов С.С.	преподаватель	25 000,00 р.	101
105	пи	Петров П.П.	преподаватель	25 000,00 р.	101
153	пи	Сидорова С.С.	инженер	15 000,00 р.	102
201	ис	Андреев А.А.	зав. кафедрой	35 000,00 р.	201
202	ис	Борисов Б.Б.	преподаватель	25 000,00 р.	201
241	ис	Глухов Г.Г.	инженер	20 000,00 р.	201
242	ис	Чернов Ч.Ч.	инженер	15 000,00 р.	202
301	мм	Басов Б.Б.	зав. кафедрой	35 000,00 р.	301
302	мм	Сергеева С.С.	преподаватель	25 000,00 р.	301
401	оф	Волков В.В.	зав. кафедрой	35 000,00 р.	401
402	оф	Зайцев З.З.	преподаватель	25 000,00 р.	401
403	оф	Смирнов С.С.	преподаватель	15 000,00 р.	401
435	оф	Лисин Л.Л.	инженер	20 000,00 р.	402
501	вм	Кузнецов К.К.	зав. кафедрой	35 000,00 р.	501
502	вм	Романцев Р.Р.	преподаватель	25 000,00 р.	501
503	вм	Соловьев С.С.	преподаватель	25 000,00 р.	501
601	эф	Зверев З.З.	зав. кафедрой	35 000,00 р.	601
602	эф	Сорокина С.С.	преподаватель	25 000,00 р.	601
614	эф	Григорьев Г.Г.	инженер	20 000,00 р.	602

Таблица 4. Специальность

Номер	Направление	Шифр
01.03.04	Прикладная математика	мм
09.03.02	Информационные системы и технологии	ис
09.03.03	Прикладная информатика	пи
14.03.02	Ядерные физика и технологии	эф
38.03.05	Бизнес-информатика	ис

Таблица 5. Дисциплина

Код	Объем	Название	Исполнитель
101	320	Математика	вм
102	160	Информатика	пи
103	160	Физика	оф
202	120	Базы данных	ис
204	160	Электроника	эф
205	80	Программирование	пи
209	80	Моделирование	мм

Таблица 6. Заявка

Номер	01.03.04			09.03.02						09.03.03					14.03.02				38.03.05			
Код	101	205	209	101	102	103	202	205	209	101	102	103	202	205	101	102	103	204	101	103	202	209

Таблица 7. Зав_кафедрой

Таб_номер	101	201	301	401	501	601
Стаж	15	18	20	10	18	8

Таблица 8. Инженер

Таб_номер	153	241	242	435	614
Специальность	электроник	электроник	программист	электроник	программист

Таблица 9. Преподаватель

Таб_номер	Звание	Степень
101	профессор	д. т.н.
102	доцент	к. ф.-м. н.
105	доцент	к. т.н.
201	профессор	д. ф.-м. н.
202	доцент	к. ф.-м. н.
301	профессор	д. т.н.
302	доцент	к. т.н.
401	профессор	д. т.н.
402	доцент	к. т.н.
403	ассистент	—
501	профессор	д. ф.-м. н.
502	профессор	д. ф.-м. н.
503	доцент	к. ф.-м. н.
601	профессор	д. ф.-м. н.

Таблица 10. Студент

Рег_номер	Номер	Фамилия
10101	09.03.03	Николаева Н. Н.
10102	09.03.03	Иванов И. И.
10103	09.03.03	Крюков К. К.
20101	09.03.02	Андреев А. А.
20102	09.03.02	Федоров Ф. Ф.
30101	14.03.02	Бондаренко Б. Б.
30102	14.03.02	Цветков К. К.
30103	14.03.02	Петров П. П.
50101	01.03.04	Сергеев С. С.
50102	01.03.04	Кудрявцев К. К.
80101	38.03.05	Макаров М. М.
80102	38.03.05	Яковлев Я. Я.

Таблица 11. Экзамен

Дата	Код	Рег_номер	Таб_номер	Аудитория	Оценка
05.06.2015	102	10101	102	т505	4
05.06.2015	102	10102	102	т505	4
05.06.2015	202	20101	202	т506	4
05.06.2015	202	20102	202	т506	3
07.06.2015	102	30101	105	ф419	3
07.06.2015	102	30102	101	т506	4
07.06.2015	102	80101	102	м425	5
09.06.2015	205	80102	402	м424	4
09.06.2015	209	20101	302	ф333	3
10.06.2015	101	10101	501	т506	4
10.06.2015	101	10102	501	т506	4
10.06.2015	204	30102	601	ф349	5
10.06.2015	209	80101	301	э105	5
10.06.2015	209	80102	301	э105	4
12.06.2015	101	80101	502	с324	4
15.06.2015	101	30101	503	ф417	4
15.06.2015	101	50101	501	ф201	5
15.06.2015	101	50102	501	ф201	3
15.06.2015	103	10101	403	ф414	4
17.06.2015	102	10101	102	т505	5

Пример 1: Выбрать факультет и кафедры, используя неявное соединение. Результат отсортировать по алфавиту:

```
SELECT
Ф.Название AS Факультет
, К.Название AS Кафедра
FROM
Факультет Ф, Кафедра К
WHERE
Ф.Аббревиатура = К.Факультет ORDER BY
Факультет, Кафедра
```

Пример 2: Выбрать факультет и кафедры, используя явное соединение. Результат от- сортировать по алфавиту:

```
SELECT
Ф.Название AS Факультет
, К.Название AS Кафедра
FROM
Факультет Ф
INNER JOIN Кафедра К ON Ф.Аббревиатура = К.Факультет
ORDER BY
Факультет, Кафедра
```

Пример 3: Выбрать все факультеты и их кафедры, если существуют. Результат отсор- тировать по алфавиту:

```
SELECT
Ф.Название AS Факультет
, К.Название AS Кафедра
FROM
Факультет Ф
LEFT OUTER JOIN Кафедра К ON Ф.Аббревиатура = К.Факультет
ORDER BY
Факультет, Кафедра
```

Пример 4: Вывести из таблиц «Кафедра», «Специальность» и «Студент» данные о сту- дентах:

```
SELECT
С.Фамилия
, П.Направление
, К.Название AS Кафедра
FROM
Студент С
INNER JOIN Специальность П ON С.Номер = П.Номер INNER JOIN Кафедра К ON
П.Шифр = К.Шифр
```

Пример 5: Вывести для каждого сотрудника фамилию, должность, зарплату и фамилию его непосредственного руководителя:

```
SELECT
С.Фамилия
, С.Должность
, С.Зарплата
, П.Фамилия AS Руководитель
```

FROM

Сотрудник С

INNER JOIN Сотрудник П ON С.Шеф = П.Таб_номер

Пример 6: Вывести список студентов, сдавших хотя бы один экзамен. По правилам соединения, студенты, не сдававшие экзамены, в выборке представлены не будут:

SELECT

С.Фамилия

FROM

Студент С

INNER JOIN Экзамен Э ON С.Рег_номер = Э.Рег_номер

GROUP BY

С.Фамилия

Пример 7: Вывести из таблиц «Студент» и «Экзамен» учетные номера и фамилии студентов, а также количество сданных экзаменов и средний балл для каждого студента:

SELECT

С.Фамилия

, COUNT(Э.Оценка) AS [Количество экзаменов]

, AVG(Э.Оценка) AS [Средний балл]

FROM

Студент С

INNER JOIN Экзамен Э ON С.Рег_номер = Э.Рег_номер

GROUP BY

С.Фамилия

Пример 8: Вывести список заведующих кафедрами и их зарплаты, и стаж работы:

SELECT

С.Фамилия

, С.Зарплата

, З.Стаж

FROM

Сотрудник С

INNER JOIN Зав_кафедрой З ON С.Таб_номер = З.Таб_номер

Пример 9: Вывести список кандидатов и докторов физико-математических наук:

SELECT

С.Фамилия

, П.Степень

FROM

Сотрудник С

INNER JOIN Преподаватель П ON С.Таб_номер = П.Таб_номер

WHERE

П.Степень IN ('к.ф.-м.н.', 'д.ф.-м.н.')

Пример 10: Вывести название дисциплины, фамилию, должность и степень преподавателя, дату и место проведения экзаменов в хронологическом порядке:

SELECT DISTINCT

Д.Название AS Дисциплина

, С.Фамилия

, С.Должность

, П.Степень

```

, Э.Дата
, Э.Аудитория
FROM
Экзамен Э
INNER JOIN Дисциплина Д ON Э.Код = Д.Код
INNER JOIN Сотрудник С ON Э.Таб_номер = С.Таб_номер INNER JOIN
Преподаватель П ON Э.Таб_номер = П.Таб_номер
ORDER BY
Э.Дата

```

Пример 11: Вывести фамилию преподавателей и количество их экзаменов:

```

SELECT

```

```

С.Фамилия
, COUNT(Э.Дата) AS [Количество экзаменов]
FROM
Экзамен Э
INNER JOIN Сотрудник С ON Э.Таб_номер = С.Таб_номер
GROUP BY
С.Фамилия

```

Пример 12: Вывести список студентов, не сдавших ни одного экзамена: SELECT

```

С.Фамилия
FROM
Студент С
LEFT OUTER JOIN Экзамен Э ON С.Рег_номер = Э.Рег_номер
WHERE
Э.Рег_номер IS NULL

```

2. Задание

1. Вывести из таблиц «Кафедра», «Специальность» и «Студент» данные о студентах, которые обучаются на данном факультете (например, «ит»).
2. Вывести из таблиц «Кафедра», «Специальность» и «Сотрудник» данные о выпускающих кафедрах (факультет, шифр, название, фамилию заведующего). Выпускающей считается та кафедра, на которую есть ссылки в таблице «Специальность».
3. Вывести в запросе для каждого сотрудника номер и фамилию его непосредственного руководителя. Для заведующих кафедрами поле руководителя оставить пустым.
4. Вывести список студентов, сдавших минимум два экзамена.
5. Вывести список инженеров с зарплатой, меньшей 20000 руб.
6. Вывести список студентов, сдавших экзамены в заданной аудитории.
7. Вывести из таблиц «Студент» и «Экзамен» учетные номера и фамилии студентов, а также количество сданных экзаменов и средний балл для каждого студента только для тех студентов, у которых средний балл не меньше заданного (например, 4).
8. Вывести список заведующих кафедрами и их зарплаты, и степень.
9. Вывести список профессоров.
10. Вывести название дисциплины, фамилию, должность и степень преподавателя, дату и место проведения экзаменов в хронологическом порядке в заданном интервале даты.

11. Вывести фамилию преподавателей, принявших более трех экзаменов.
12. Вывести список студентов, не сдавших ни одного экзамена в указанной дате.

Лабораторная работа № 16. Агрегатные функции

Формируемые компетенции или их части

№ п/п	Содержание компетенции	Шифр
1.	Способен управлять процессами создания и использования продуктов и услуг в сфере информационно-коммуникационных технологий, в том числе разрабатывать алгоритмы и программы для их практической реализации	ОПК-3
2.	Способен организовывать взаимодействие с клиентами и партнерами в процессе решения задач управления жизненным циклом информационных систем и информационно-коммуникационных технологий	ОПК-5

Теоретическая часть

Агрегатные функции используются для получения обобщающих значений. Они, в отличие от скалярных функций, оперируют значениями столбцов множества строк. К агрегатным функциям относятся:

- SUM – вычисляет итог;
- MAX – возвращает наибольшее значение;
- MIN – возвращает наименьшее значение;
- AVG – вычисляет среднее значение;
- COUNT – вычисляет количество значений в столбце.

Аргументами функций выступают поля таблицы или результаты выражений над ними.

Вложенность не допускается.

Из агрегатных функций можно составлять любые выражения.

Для функций SUM и AVG столбец должен содержать числовые значения.

Для функций COUNT() можно указать аргумент * для подсчета всех строк без исключения.

По умолчанию вышеперечисленные пять функций учитывают все строки выборки для вычисления результата. Но выборка может содержать повторяющиеся значения. Если необходимо выполнить вычисления только над уникальными значениями, исключив из набора значений повторяющиеся данные, то для этого применяется оператор DISTINCT (кроме COUNT (*)). По умолчанию вместо DISTINCT применяется оператор ALL, который выбирает все строки. Так как этот оператор неявно подразумевается при отсутствии DISTINCT, то его можно не указывать.

Агрегатные функции можно применить не только на всю таблицу, но также на группу значений. Для этого применяется команда GROUP BY, которая пишется после WHERE. После команды GROUP BY перечисляется название столбцов, по которым следует группировать данные. Предложение GROUP BY указывает, что результаты запроса следует разделить на группы, применить агрегатную функцию по отдельности к каждой группе и получить для каждой группы одну строку результатов.

В качестве элемента группировки должен выступать любой возвращаемый элемент, указанный в предложении SELECT, кроме значений агрегатных функций.

Если столбец, по которому производится группировка, содержит значение NULL, то строки со значением NULL составят отдельную группу.

Команда HAVING <условие> применяется для фильтрации строк, возвращаемых при использовании предложения GROUP BY. HAVING пишется после GROUP BY, имеет такой формат, как WHERE, но в качестве значения используется значение, возвращаемое агрегат-ными функциями.

1. Практическая часть

Дана таблица **Страны**:

Название	Столица	Площадь	Население	Континент
Австрия	Вена	83858	8741753	Европа
Азербайджан	Баку	86600	9705600	Азия
Албания	Тирана	28748	2866026	Европа
Алжир	Алжир	2381740	39813722	Африка
Ангола	Луанда	1246700	25831000	Африка
Аргентина	Буэнос-Айрес	2766890	43847000	Южная Америка
Афганистан	Кабул	647500	29822848	Азия
Бангладеш	Дакка	144000	160221000	Азия
Бахрейн	Манама	701	1397000	Азия
Белиз	Бельмопан	22966	377968	Северная Америка
Белоруссия	Минск	207595	9498400	Европа
Бельгия	Брюссель	30528	11250585	Европа
Бенин	Порто-Ново	112620	11167000	Африка
Болгария	София	110910	7153784	Европа
Боливия	Сукре	1098580	10985059	Южная Америка
Ботсвана	Габороне	600370	2209208	Африка
Бразилия	Бразилиа	8511965	206081432	Южная Америка
Буркина-Фасо	Уагадугу	274200	19034397	Африка
Бутан	Тхимпху	47000	784000	Азия
Великобритания	Лондон	244820	65341183	Европа
Венгрия	Будапешт	93030	9830485	Европа

Название	Столица	Площадь	Население	Континент
Венесуэла	Каракас	912050	31028637	Южная Америка
Восточный Тимор	Дили	14874	1167242	Азия
Вьетнам	Ханой	329560	91713300	Азия

Пример 1: Вывести максимальную площадь стран: **SELECT**
MAX(Площадь) **AS** Макс_площадь
FROM
Страны

Пример 2: Вывести наименьшее население стран в Африке: **SELECT**
MIN(Население) **AS** Мин_население
FROM
Страны
WHERE
Континент = 'Африка'

Пример 3: Вывести суммарное население стран Северной и Южной Америки:
SELECT
SUM(Население) **AS** Суммарное_население
FROM
Страны
WHERE
Континент = 'Северная Америка' OR Континент = 'Южная Америка'

Пример 4: Вывести среднее население стран, кроме европейских. Результат округлить до двух знаков:
SELECT
ROUND(**AVG**(**CAST**(Население **AS** **FLOAT**)), 2) **AS** Среднее_население
FROM
Страны
WHERE
Континент != 'Европа'

Пример 5: Вывести количество стран, название которых начинается с буквы «С»:
SELECT
COUNT(*) **AS** Количество
FROM
Страны
WHERE
LEFT(Название, 1) = 'С'

Пример 6: Вывести количество континентов, где есть страны: **SELECT**
COUNT(**DISTINCT** Континент) **AS** Количество_Континентов
FROM
Страны

Пример 7: Вывести разницу населения между странами с наибольшим и наименьшим количеством граждан:
SELECT
MAX(Население) - **MIN**(Население) **AS** Разница

FROM

Страны

Пример 8: Вывести количество стран на каждом континенте. Результат отсортировать по количеству стран по убыванию:

SELECT

Континент

, *COUNT*(Название) *AS* Количество_Стран *FROM*

Страны *GROUP BY*

Континент *ORDER BY*

Количество_Стран *DESC*

Пример 9: Вывести количество стран по первым буквам в названии. Результат отсортировать в алфавитном порядке:

SELECT

LEFT(Название, 1) *AS* Первая_буква

, *COUNT*(Название) *AS* Количество_Стран

FROM

Страны

GROUP BY

LEFT(Название, 1) *ORDER BY*

Первая_буква

Пример 10: Вывести список континентов, где плотность населения больше, чем 100 чел. на кв. км:

SELECT

Континент

, *AVG*(*CAST*(Население *AS* *FLOAT*) / Площадь) *AS* Сред_Плотность

FROM

Страны

GROUP BY

Континент *HAVING*

AVG(*CAST*(Население *AS* *FLOAT*) / Площадь) > 100

Пример 11: Ожидается, что через 25 лет население Европы и Азии вырастет на 20%, Северной Америки и Африки на 50%, а остальных частей мира – на 70%. Вывести список континентов с прогнозируемым населением:

SELECT

Континент

, *CASE*

WHEN Континент *IN* ('Европа', 'Азия') *THEN* *FLOOR*(*SUM*(Население) * 1.2)

WHEN Континент *IN* ('Северная Америка', 'Африка') *THEN*

FLOOR(*SUM*(Насе-
ление) *

1.5) *ELSE* *FLOOR*(*SUM*(Население) * 1.7) *END AS* Суммарное_Население

Страны

FROM

GROUP BY

Континент

Пример 12: Вывести список континентов, где разница по населению между наибольшими и наименьшими странами не более в 1000 раз:

SELECT

Континент

FROM

Страны

GROUP BY

Континент

HAVING MAX(Население) <= 1000 * MIN(Население)

Пример 13: Вывести количество стран, у которых нет столицы (не введена в базу):
SELECT

COUNT(*) AS Количество

FROM

Страны

WHERE

Столица IS NULL

Пример 14: Вывести количество символов в самых длинных и коротких названиях стран и столиц:

SELECT

MAX(LEN(Название)) AS Дл_Название

, MAX(LEN(Столица)) AS Дл_Столица

, MIN(LEN(Название)) AS Кр_Название

, MIN(LEN(Столица)) AS Кр_Столица

FROM

Страны

Пример 15: Вывести список континентов, у которых средняя плотность среди стран с площадью более 1 млн. кв. км больше, чем 30 чел. на кв. км. Результат отсортировать по плотности по убыванию:

SELECT

Континент

, AVG(CAST(Население AS FLOAT) / Площадь) AS Плотность

FROM

Страны

WHERE

Площадь > 1000000 GROUP BY

Континент HAVING

AVG(CAST(Население AS FLOAT) / Площадь) > 30 ORDER BY

Плотность DESC

2. Задание

1. Вывести минимальную площадь стран.
2. Вывести наибольшую по населению страну в Северной и Южной Америке.
3. Вывести среднее население стран. Результат округлить до одного знака.
4. Вывести количество стран, у которых название заканчивается на «ан», кроме стран, у которых название заканчивается на «стан».

5. Вывести количество континентов, где есть страны, название которых начинается с буквы «Р».
6. Сколько раз страна с наибольшей площадью больше, чем страна с наименьшей площадью?
7. Вывести количество стран с населением больше, чем 100 млн. чел. на каждом континенте. Результат отсортировать по количеству стран по возрастанию.
8. Вывести количество стран по количеству букв в названии. Результат отсортировать по убыванию.
9. Ожидается, что через 20 лет население мира вырастет на 10%. Вывести список континентов с прогнозируемым населением:
10. Вывести список континентов, где разница по площади между наибольшими и наименьшими странами не более в 10000 раз:
11. Вывести среднюю длину названий Африканских стран.
12. Вывести список континентов, у которых средняя плотность среди стран с населением более 1 млн. чел. больше, чем 30 чел. на кв. км.

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

для обучающихся по организации и проведению самостоятельной работы
по дисциплине «Базы данных»
направление подготовки 38.03.05 Бизнес-информатика
направленность (профиль) «Информационная бизнес-аналитика и цифровые
технологии»

ВВЕДЕНИЕ

Самостоятельная работа - планируемая учебная, учебно-исследовательская, научно-исследовательская работа студентов, выполняемая во внеаудиторное (аудиторное) время по заданию и при методическом руководстве преподавателя, но без его непосредственного участия (при частичном непосредственном участии преподавателя, оставляющем ведущую роль за работой студентов).

Самостоятельная работа студентов в ВУЗе является важным видом учебной и научной деятельности студента. Самостоятельная работа студентов играет значительную роль в рейтинговой технологии обучения.

Целью данной дисциплины является формирование набора (ОПК-3 и ОПК-5) компетенций будущего бакалавра по направлению подготовки 38.03.05 «Бизнес-информатика».

Задачами дисциплины «Базы данных» являются: систематизация и закрепление полученных теоретических знаний и практических умений студентов; углубление и расширение теоретических знаний; формирование умений использовать нормативную, правовую, справочную документацию и специальную литературу; развитие познавательных способностей и активности студентов: творческой инициативы, самостоятельности, ответственности и организованности; формирование самостоятельности мышления, способностей к саморазвитию, самосовершенствованию и самореализации; развитие исследовательских умений; использование материала, собранного и полученного в ходе самостоятельных занятий на семинарах, на практических и лабораторных занятиях, при написании курсовых и выпускной квалификационной работ, для эффективной подготовки к итоговым зачетам и экзаменам.

Дисциплина «Базы данных» входит в Блок 1 обязательных дисциплин учебного плана 2026 года по направлению 38.03.05 «Бизнес информатика». Ее освоение происходит в 4 семестре.

Общая характеристика самостоятельной работы студента при изучении дисциплины «Базы данных»

Целью самостоятельной работы студентов является овладение фундаментальными знаниями, профессиональными умениями и навыками деятельности по профилю, опытом творческой, исследовательской деятельности. Самостоятельная работа студентов способствует развитию самостоятельности, ответственности и организованности, творческого подхода к решению проблем учебного и профессионального уровня.

Задачами СР являются:

систематизация и закрепление полученных теоретических знаний и практических умений студентов;

углубление и расширение теоретических знаний;

формирование умений использовать нормативную, правовую, справочную документацию и специальную литературу;

развитие познавательных способностей и активности студентов: творческой инициативы, самостоятельности, ответственности и организованности;

формирование самостоятельности мышления, способностей к саморазвитию, самосовершенствованию и самореализации;

развитие исследовательских умений;

использование материала, собранного и полученного в ходе самостоятельных занятий на семинарах, на практических и лабораторных занятиях, при написании курсовых и выпускной квалификационной работ, для эффективной подготовки к итоговым зачетам и экзаменам.

План-график выполнения самостоятельной работы

Коды реализуемых компетенций	Вид деятельности студентов	Итоговый продукт самостоятельной работы	Средства и технологии оценки	Объем часов, в том числе		
				СРС	Контактная работа с преподавателем	Всего
4 семестр						
ОПК-3 ОПК-5	Самостоятельное решение задач	база данных	Подготовка к экзамену	32	4	36
ОПК-3 ОПК-5	Подготовка к лабораторным работам	Комплект заданий	Зачетное задание	43	5	48
ОПК-3 ОПК-5	Лекционный материал	Конспект лекций	Собеседование	43	5	48
Итого за 4 семестр				154	14	168
Итого				154	14	168

Методические рекомендации по подготовке к лабораторным занятиям

Работа по подготовке к лабораторным занятиям и активное в них участие — одна из форм изучения программного материала курса. Она направлена на подготовку высококвалифицированных магистров. Подготовку к занятиям следует начинать с внимательного изучения соответствующих разделов учебных пособий и учебников, далее — следует изучать специальную литературу и источники, работать с картами, таблицами, схемами, наконец, написать доклад, если студент получил такое задание. Готовясь к занятиям

и принимая активное участие в их работе, студент проходит школу работы над источниками и литературой, получает навыки самостоятельной работы над письменным и устным сообщением (докладом), учится участвовать в дискуссиях, отстаивать свою точку зрения, формулировать и аргументировать выводы. Форма практических занятий во многом определяется его темой. Практика показывает, что основные формы занятий следующие: беседа на основе составленного преподавателем плана (она наиболее приемлема при обсуждении одного из теоретических вопросов по проблемам темы или монографии), коллоквиум по разделу учебника или одной из монографий (коллоквиум предполагает прежде всего проверку знаний по определенной теме, источникам, разделу курса); подготовка письменного доклада студентом, его устный доклад и обсуждение его на занятии.

В планы лабораторных занятий включены основные вопросы общего курса. В ходе занятий возможна их конкретизация и корректировка. При подготовке сообщений и докладов следует широко использовать опубликованные источники, исследовательскую литературу. Учебники и учебные пособия студент использует по своему выбору. Каждому студенту в течение года следует прочитать не менее двух монографий (или одну монографию и одну работу мемуарного характера), которые указаны в списке литературы или рекомендовано преподавателем из числа новых публикаций, составить краткий доклад и быть готовым к беседе по ним с преподавателем.

Методические рекомендации по изучению теоретического материала

Работа с книгой

При работе с книгой необходимо подобрать литературу, научиться правильно ее читать, вести записи. Для подбора литературы в библиотеке используются алфавитный и систематический каталоги.

Важно помнить, что рациональные навыки работы с книгой - это всегда большая экономия времени и сил.

Правильный подбор учебников рекомендуется преподавателем, читающим лекционный курс. Необходимая литература может быть также указана в методических разработках по данному курсу.

Изучая материал по учебнику, следует переходить к следующему вопросу только после правильного уяснения предыдущего, описывая на бумаге все выкладки и вычисления (в том числе те, которые в учебнике опущены или на лекции даны для самостоятельного вывода).

При изучении любой дисциплины большую и важную роль играет самостоятельная индивидуальная работа.

Особое внимание следует обратить на определение основных понятий курса. Студент должен подробно разбирать примеры, которые поясняют такие определения, и уметь строить аналогичные примеры самостоятельно. Нужно добиваться точного представления о том, что изучаешь. Полезно составлять опорные конспекты. При изучении материала по учебнику полезно в тетради (на специально отведенных полях) дополнять конспект лекций. Там же следует отмечать вопросы, выделенные студентом для консультации с преподавателем.

Выводы, полученные в результате изучения, рекомендуется в конспекте выделять, чтобы они при перечитывании записей лучше запоминались.

Опыт показывает, что многим студентам помогает составление листа опорных сигналов, содержащего важнейшие и наиболее часто употребляемые формулы и понятия. Такой лист помогает запомнить формулы, основные положения лекции, а также может служить постоянным справочником для студента.

Различают два вида чтения; первичное и вторичное. *Первичное* - это внимательное, неторопливое чтение, при котором можно остановиться на трудных местах. После него не должно остаться ни одного непонятого слова. Содержание не всегда может быть понятно после первичного чтения.

Задача *вторичного* чтения - полное усвоение смысла целого (по счету это чтение может быть и не вторым, а третьим или четвертым).

Правила самостоятельной работы с литературой

Как уже отмечалось, самостоятельная работа с учебниками и книгами (а также самостоятельное теоретическое исследование проблем, обозначенных преподавателем на лекциях) – это важнейшее условие формирования у себя научного способа познания. Основные советы здесь можно свести к следующим:

- Составить перечень книг, с которыми Вам следует познакомиться;
- Сам такой перечень должен быть систематизированным.
- Обязательно выписывать все выходные данные по каждой книге (при написании курсовых и дипломных работ это позволит очень сэкономить время).
- Разобраться для себя, какие книги (или какие главы книг) следует прочитать более внимательно, а какие – просто просмотреть.
- При составлении перечней литературы следует посоветоваться с преподавателями и научными руководителями (или даже с более подготовленными и эрудированными сокурсниками), которые помогут Вам лучше сориентироваться, на что стоит обратить большее внимание, а на что вообще не стоит тратить время...

• Естественное, все прочитанные книги, учебники и статьи следует конспектировать, но это не означает, что надо конспектировать «все подряд»: можно выписывать кратко основные идеи автора и иногда приводить наиболее яркие и показательные цитаты (с указанием страниц).

Чтение научного текста является частью познавательной деятельности. Ее цель – извлечение из текста необходимой информации. От того на сколько осознанна читающим собственная внутренняя установка при обращении к печатному слову (найти нужные сведения, усвоить информацию полностью или частично, критически проанализировать материал и т.п.) во многом зависит эффективность осуществляемого действия.

Выделяют **четыре основные установки в чтении научного текста**:

1. информационно-поисковый (задача – найти, выделить искомую информацию)
2. усваивающая (усилия читателя направлены на то, чтобы как можно полнее осознать и запомнить как сами сведения излагаемые автором, так и всю логику его рассуждений)
3. аналитико-критическая (читатель стремится критически осмыслить материал, проанализировав его, определив свое отношение к нему)
4. творческая (создает у читателя готовность в том или ином виде – как отправной пункт для своих рассуждений, как образ для действия по аналогии и т.п. – использовать суждения автора, ход его мыслей, результат наблюдения, разработанную методику, дополнить их, подвергнуть новой проверке).

Основные виды систематизированной записи прочитанного:

1. Аннотирование – предельно краткое связное описание просмотренной или прочитанной книги (статьи), ее содержания, источников, характера и назначения;
2. Планирование – краткая логическая организация текста, раскрывающая содержание и структуру изучаемого материала;
3. Тезирование – лаконичное воспроизведение основных утверждений автора без привлечения фактического материала;
4. Цитирование – дословное выписывание из текста выдержек, извлечений, наиболее существенно отражающих ту или иную мысль автора;
5. Конспектирование – краткое и последовательное изложение содержания прочитанного.

Конспект – сложный способ изложения содержания книги или статьи в логической последовательности. Конспект аккумулирует в себе предыдущие виды записи, позволяет всесторонне охватить содержание книги, статьи. Поэтому умение составлять план, тезисы, делать выписки и другие записи определяет и технологию составления конспекта.

Методические рекомендации по написанию доклада

Доклад – публичное сообщение, представляющее собой развернутое изложение на определенную тему, вид самостоятельной работы, который используется в учебных и внеаудиторных занятиях и способствует формированию навыков исследовательской работы, расширяет познавательные интересы, приучает критически мыслить.

Чтобы выступление было удачным, оно должно хорошо восприниматься на слух, быть интересным для слушателей. При выступлении приветствуется активное использование мультимедийного сопровождения доклада (презентация, видеоролики, аудиозаписи).

Общая структура доклада. Построение доклада включает три части: вступление, основную часть и заключение.

Вступление.

Формулировка темы доклада (она должна быть не только актуальной, но и оригинальной, интересной по содержанию).

Актуальность выбранной темы (чем она интересна, в чем заключается ее важность, почему учащимся выбрана именно эта тема).

Анализ литературных источников (рекомендуется использовать данные за последние 5 лет)

Основная часть. Состоит из нескольких разделов, постепенно раскрывающих тему. Возможно использование иллюстрации (графики, диаграммы, фотографии, карты, рисунки). Если необходимо, для обоснования темы используется ссылка на источники с доказательствами, взятыми из литературы (цитирование авторов, указание цифр, фактов, определений). Изложение материала должно быть связным, последовательным, доказательным.

Способ изложения материала для выступления должен носить конспективный или тезисный характер.

Заключение.

Подводятся итоги, формулируются главные выводы, подчеркивается значение рассмотренной проблемы, предлагаются самые важные практические рекомендации.

Требования к оформлению доклада.

Объем машинописного текста доклада должен быть рассчитан на произнесение доклада в течение 7 -10 минут (3-5 машинописных листа текста с докладом). Поэтому при подборе необходимого материала для доклада отбирается самое главное. В докладе должны быть кратко отражены главные моменты из введения, основной части и заключения. При подготовке конспекта доклада необходимо составить не только текст доклада, но и необходимый иллюстративный материал, сопровождающий доклад (основные тезисы, формулы, схемы, чертежи, таблицы, графики и диаграммы, фотографии и т.п.).

Методические указания по подготовке к экзамену

В период подготовки к экзамену студенты вновь обращаются к пройденному учебному материалу. При этом они не только скрепляют полученные знания, но и получают новые. Подготовка студента к экзамену включает в себя три этапа:

- самостоятельная работа в течение семестра;
- непосредственная подготовка в дни, предшествующие экзамену по темам курса;
- подготовка к ответу на вопросы, содержащиеся в билетах.

Литература для подготовки к экзамену рекомендуется преподавателем либо указана в учебно-методическом комплексе. Для полноты учебной информации и ее сравнения лучше использовать не менее двух учебников. Студент вправе сам придерживаться любой из представленных в учебниках точек зрения по спорной проблеме (в том числе отличной от преподавателя), но при условии достаточной научной аргументации.

Основным источником подготовки к экзамену является конспект лекций, где учебный материал дается в систематизированном виде, основные положения его детализируются, подкрепляются современными фактами и информацией, которые в силу новизны не вошли в

опубликованные печатные источники. В ходе подготовки к экзамену студентам необходимо обращать внимание не только на уровень запоминания, но и на степень понимания излагаемых проблем.

Экзамен проводится по билетам, охватывающим весь пройденный материал. По окончании ответа экзаменатор может задать студенту дополнительные и уточняющие вопросы. На подготовку к ответу по вопросам билета студенту дается 60 минут с момента получения им билета. Положительным также будет стремление студента изложить различные точки зрения на рассматриваемую проблему, выразить свое отношение к ней, применить теоретические знания по современным проблемам жилищного права.

Результаты экзамена объявляются студенту после окончания ответа в день сдачи.

Список рекомендуемой литературы

Основная литература:

1. Управление данными [Электронный ресурс] : учебник / Ю.Ю. Громов [и др.]. — Электрон. текстовые данные. — Тамбов: Тамбовский государственный технический университет, ЭБС АСВ, 2015. — 192 с. — 978-5-8265-1385-9. — Режим доступа: <http://www.iprbookshop.ru/63912.html>

2. Кузнецов С.Д. Введение в реляционные базы данных [Электронный ресурс] / С.Д. Кузнецов. — 2-е изд. — Электрон. текстовые данные. — М. : Интернет-Университет Информационных Технологий (ИНТУИТ), 2016. — 247 с. — 5-9556-00028-0. — Режим доступа: <http://www.iprbookshop.ru/73671.html>

Дополнительная литература:

1. Абросимова, М.А. Базы данных: проектирование и создание программного приложения в СУБД MS Access : практикум / М.А. Абросимова ; Министерство образования и науки Российской Федерации, Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования «Уфимский государственный университет экономики и сервиса», Кафедра «Информатика и ИКТ». - Уфа : Уфимский государственный университет экономики и сервиса, 2014. - 56 с. - Библиогр. в кн. ; То же [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=272367>

2. Чурбанова, О.В. Базы данных и знаний. Проектирование баз данных в MicrosoftAccess : учебно-методическое пособие / О.В. Чурбанова, А.Л. Чурбанов ; Министерство образования и науки Российской Федерации, Северный (Арктический) федеральный университет имени М.В. Ломоносова. - Архангельск : САФУ, 2015. - 152 с. : ил., схем., табл. - Библиогр. в кн. - ISBN 978-5-261-01029-6 ; То же [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=436230>

Интернет-ресурсы:

www.biblioclub.ru – Электронно-библиотечная система «Университетская библиотека онлайн».

www.iprbookshop.ru - Электронно-библиотечная система «IPRbooks»

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

по выполнению и защите курсовой работы
по дисциплине «Базы данных»
направление 38.03.05 «Бизнес-информатика»
направленность (профиль) «Информационная бизнес-аналитика и цифровые
технологии»

Содержание

Формулировка задания.....	77
Тематика курсовых работ.....	78
Структура работы.....	79
Общие требования к написанию и оформлению работы.....	79
Требования к оформлению курсовой работы.....	79
Последовательность выполнения задания.....	84
Критерии оценивания.....	84
Порядок защиты курсовой работы.....	86

Цель, задачи и реализуемые компетенции

Курсовая работа является самостоятельной исследовательской работой студента и представляет собой логически завершенное и оформленное научное исследование.

Цель — формирование у студента навыков научно-исследовательской работы, повышение уровня его профессиональной (теоретической и практической) подготовки, углубление знаний по учебной дисциплине, развитие интереса и навыков самостоятельной работы с научной и справочной литературой.

Новая модель образования, реализуемая в федеральном университете, освоение федеральных государственных образовательных стандартов предъявляют высокие требования к качеству подготовки специалистов. Выпускники университета по глубине усвоенных фундаментальных знаний и научному кругозору должны быть способны самостоятельно и высокопрофессионально решать производственные и научные задачи. Развитие навыков самостоятельной учебной, исследовательской и научной работы студентов происходит в процессе выполнения курсовых работ.

Выполнение курсовой работы имеет целью расширение знаний студентов, обучение методам теоретического анализа явлений и закономерностей науки, отработку навыков самостоятельного применения теоретических знаний к комплексному решению профессиональных задач, использования справочной литературы, методов математической обработки экспериментальных данных, компьютерных технологий. Системой курсовых работ (проектов) студент подготавливается к выполнению выпускной квалификационной работы.

В процессе выполнения курсовой работы студентом должны решаться следующие задачи:

- приобретение новых теоретических знаний в соответствии с темой работы и заданием руководителя;
- умение систематизировать, обобщать и логично излагать концепции, альтернативные точки зрения по исследуемой проблеме;
- развитие учебно-исследовательских и методических навыков, необходимых для системного научного анализа изучаемого явления;
- совершенствование профессиональной подготовки.

Основной целью курсовой работы по курсу «Базы данных» является систематизация, углубление и активное применение знаний по проектированию баз данных, закрепление знаний, полученных в лекционном курсе, а также на лабораторных занятиях.

Компетенции обучающегося, формируемые в результате написания курсовой работы – ОПК-3 и ОПК-5.

Формулировка задания

Темы курсовых работ утверждаются на заседании кафедры, ведущей дисциплины, по которым учебными планами предусмотрены курсовые проекты, в течение 2-х недель после начала семестра. Выписка из протокола заседания кафедры передается в дирекцию института, где формируется распоряжение об утверждении тем курсовых работ. В течение 10 дней после выхода распоряжения об утверждении тем курсовых работ специалист по учебно-методической работе дирекции вносит темы работ в автоматизированную базу университета.

Основные содержательные и процессуальные аспекты, необходимые для выполнения работы, оформляются кафедрой в заданиях по курсовому проектированию (Приложение 1). В заданиях необходимо четко сформулировать тему работы (проекта) и требования, определяющие ее объем, содержание, а также исходные данные.

Цель курсовой работы состоит в изучении теоретических основ темы исследования и применении инструментальных средств проектирования и создания БД.

Задачи

- изучить информационные источники по теме исследования;

- дать характеристику основным терминам и определениям по теме курсовой работы, привести классификацию;
- изучить инструментарий и привести пример практического применения в конкретной предметной области.

Перечень подлежащих разработке вопросов

- а) по теоретической части рассматриваются основные понятия БД;
- б) по проектной части рассматриваются проектирование и создание БД с помощью инструментальных средств.

Тематика курсовых работ

1. Сетевые базы данных и СУБД
2. Иерархические базы данных и СУБД
3. Реляционные базы данных и СУБД
4. Объектные базы данных и СУБД
5. Объектно-ориентированные базы данных и СУБД
6. Объектно-реляционные базы данных и СУБД
7. Архитектура систем управления базами данных (СУБД)
8. Система управления базами данных MS Access
9. Система управления базами данных MS SQL Server
10. Системы управления базами данных (СУБД): настройка, обработка запросов, оптимизация
11. Системы управления базами данных (СУБД): управление параллельным доступом, транзакции, способы решения проблем
12. Распределенные и параллельные базы данных.
13. Архитектура «клиент-сервер»
14. Методология проектирования БД: моделирование предметной области
15. Методология проектирования БД: концептуальное проектирование
16. Методология проектирования БД: логическое проектирование
17. Методология проектирования БД: физическое проектирование
18. Проектирование реляционной базы данных
19. Нормализация реляционной модели данных
20. Инфологическое моделирование.
21. Модель ER «сущность-связь»
22. Принципы построения: реляционной базы данных: реляционная алгебра и реляционное исчисление
23. Язык структурированных запросов SQL: использование языка SQL в прикладном программировании
24. Язык структурированных запросов SQL: оптимизация запросов в SQL
25. Язык структурированных запросов SQL: организация защиты данных с помощью SQL
26. Безопасность и администрирование БД
27. Распределенные базы данных: технология обработки данных OLAP
28. Распределенные базы данных: технология OLTP
29. Принципы построения распределенных хранилищ данных
30. Восстановление данных в СУБД MS Access и СУБД MS SQL Server
31. Проблемы параллелизма в СУБД
32. Проблемы безопасности баз
33. Проблемы целостности баз данных.
34. Проблемы создания и сжатия больших объемов информационных массивов, информационных хранилищ и складов данных
35. Разработка и сопровождение баз данных в MS SQL Server
36. Мультимедийные базы данных

Структура работы

Курсовая работа должна состоять из введения, теоретической части, эмпирической (практической, расчетно-графической) части, заключения, списка информационных источников и приложения. В отдельных случаях, в соответствии с тематикой работы (проекта), эмпирическая часть может отсутствовать.

Общие требования к написанию и оформлению работы

Курсовая работа должна содержать элементы новизны, наряду с фундаментальным аспектом должен быть проведен анализ современного состояния изучаемой проблемы, а также включенность в региональную проблематику. Задание по курсовой работе (проекту) необходимо индивидуализировать с учетом интересов и способностей студентов.

Титульный лист оформляется в соответствии с Приложением 2.

Во введении обосновывается актуальность выбранной темы исследования; отражаются объект, предмет, цель, задачи, методы, новизна, теоретическая и практическая значимость курсовой работы.

Теоретическая часть должна содержать анализ состояния изучаемой проблемы на основе обзора научной, научно-информационной, справочной литературы. Представленный материал должен быть логически связан с целью исследования. В параграфах теоретической части необходимо отражать отдельные компоненты проблемы и завершать их выводами.

Проектная часть включает в себя: характеристику предметной области, обоснование создания базы данных, разработку инфологической модели, концептуальной модели, физической модели базы данных; создание необходимых форм, запросов и отчетов; выводы.

Заключение содержит выводы, подтверждающие или опровергающие первоначальные предположения (гипотезы), перспективы дальнейшего изучения проблемы, связь с практикой, анализ реализации целей и задач исследования.

Список информационных источников должен быть составлен в соответствии с требованиями ГОСТа к оформлению библиографии.

Приложение содержит весь фактический материал экспериментальных исследований (анкеты, опросники, схемы, чертежи, расчетные материалы, карты, рисунки, ответы респондентов и т.д.).

Требования к оформлению курсовой работы

Курсовую работу (проект) рекомендуется представлять в объеме 1-2 печатных листа.

Текст работы должен быть машинописным на одной стороне белой бумаги формата А4 (210x297 мм). При выполнении работы необходимо соблюдать равномерную плотность, контрастность и четкость изображения по всей работе. В работе должны быть четкие, не расплывшиеся линии, буквы, цифры и знаки.

При выполнении работы на компьютере (в текстовом редакторе WORD) необходимо установить следующие параметры:

- размер левого поля - 30 мм, правого - 15 мм, верхнего и нижнего полей - 20мм;
- шрифт Times New Roman, размершрифта 14;
- выравнивание по ширине, первая строка равна отступ 1,25 см, межстрочный интервал – 1,5 (отступ справа/слева и интервал перед/после равны нулю, разбивка на страницы без запрета висячих строк).

Цвет шрифта - черный, подчеркивания и полужирный шрифт не допускаются. Разрешается использовать компьютерные возможности акцентирования внимания на определенных формулах, теоремах, применяя шрифты разной гарнитуры.

Все страницы курсовой работы, включая иллюстрации и приложения, нумеруются по порядку без пропусков и повторений. Первой страницей считается титульный лист, на котором номер страницы не ставится, на следующей странице ставится цифра «2» и т.д. В общую нумерацию входят список информационных источников и приложения.

Порядковый номер страницы проставляют в центре нижней части листа без точки.

Главы курсовой работы должны иметь порядковые номера в пределах всей работы, обозначенные арабскими цифрами и записанные с абзацного отступа (например, 1, 2, 3 и т.д.). Нумерация пунктов должна быть в пределах каждой главы, и номер пункта должен состоять из номеров главы и пункта в главе, разделенных точкой (например, 1.1, 1.2, 1.3 и т.д.).

Слово «Глава» не пишется.

Пункты, при необходимости, могут быть разбиты на подпункты, которые должны иметь порядковую нумерацию в пределах каждого пункта (например, 1.1.1, 1.1.2, 1.1.3 и т.д.).

В конце номера главы и пункта (подпункта) точка не ставится.

Каждую главу курсовой работы необходимо начинать с нового листа (страницы).

Заголовки глав следует располагать в середине строки без точки в конце и печатать прописными буквами, не подчеркивая.

Заголовки пунктов и подпунктов следует печатать с абзацного отступа с прописной буквы без точки в конце, выравнивая по ширине.

Переносы слов в заголовках не допускаются. Если заголовок состоит из двух предложений, их разделяют точкой. Расстояние после заголовка главы и структурного элемента - 2 свободные строки, между текстом и заголовком пункта - 2 свободные строки, между заголовком пункта и текстом - 1 свободная строка.

Внутри пунктов или подпунктов могут быть приведены перечисления. Перед каждой позицией перечисления следует ставить дефис (использование других символов не допускается), запись производится с абзацного отступа.

При необходимости ссылки в тексте работы на один из элементов перечисления вместо дефиса ставятся строчные буквы в порядке русского алфавита, начиная с буквы а (за исключением букв ё, з, й, о, ч, ь, ы, ь). Для дальнейшей детализации перечислений необходимо использовать арабские цифры, после которых ставится скобка, запись производится с абзацного отступа.

В тексте документа, за исключением формул, таблиц и рисунков, не допускается:

- применять математический знак минус (-) перед отрицательными значениями величин (следует писать слово «минус»);

- применять знак « Φ » для обозначения диаметра (следует писать слово «диаметр»). При указании размера или предельных отклонений диаметра на чертежах, помещенных в тексте документа, перед размерным числом следует писать знак « Φ »;

- применять без числовых значений математические знаки, например > (больше), < (меньше), = (равно), $\geq$ (больше или равно), $\leq$ (меньше или равно), $\neq$ (не равно), а также знаки № (номер), % (процент);

- применять индексы стандартов, технических условий и других документов без регистрационного номера.

Уравнения и формулы следует выделять из текста в отдельную строку. Выше и ниже каждой формулы или уравнения должна быть оставлена одна свободная строка.

В формулах в качестве символов следует применять обозначения, установленные соответствующими государственными стандартами. Пояснения символов и числовых коэффициентов, входящих в формулу, если они не пояснены ранее в тексте, должны быть приведены непосредственно под формулой. Пояснения каждого символа следует давать с новой строки в той последовательности, в которой символы приведены в формуле. Первая строка пояснения должна начинаться со слова «где» без двоеточия после него. Формулы, за

исключением формул, помещаемых в приложения, должны нумероваться сквозной нумерацией арабскими цифрами, которые записывают на уровне формулы справа в круглых скобках. Одну формулу обозначают - (1). Допускается нумерация формул в пределах главы. В этом случае номер формулы состоит из номера главы и порядкового номера формулы, разделенных точкой, например (3.1). Формулы, помещаемые в приложениях, должны нумероваться отдельной нумерацией арабскими цифрами в пределах каждого приложения с добавлением перед каждой цифрой обозначения приложения, например формула (B.1).

Формулы, следующие одна за другой и не разделенные текстом, разделяют запятой.

Если уравнение не уместится в одну строку, то оно должно быть перенесено после знака равенства (=) или после знаков плюс (+), минус (-), умножения (x), деления (:) или других математических знаков, причем знак в начале следующей строки повторяют. При переносе формулы на знаке, символизирующем операцию умножения, применяют знак «×».

Ссылки в тексте на порядковые номера формул дают в скобках, например, ... в формулах (5, 6).

Формулы в работе следует нумеровать порядковой нумерацией в пределах всей работы арабскими цифрами в круглых скобках в крайнем правом положении на строке.

$$p = \frac{(\overline{ЧП}_0 - \overline{ЧП}) * \gamma(z) + \sqrt{D \overline{ЧП}} * \beta(z)}{\overline{ЧП}} * 100, \quad (5)$$

$$z = \frac{(\overline{ЧП}_0 - \overline{ЧП})}{\sqrt{D \overline{ЧП}}}, \quad (6)$$

где p - риск инвестиционного проекта;

$\overline{ЧП}_0$ - планируемая величина чистой прибыли, руб.;

$\overline{ЧП}$ - среднее значение чистой прибыли при различных вариантах, руб.;

$D \overline{ЧП}$ - дисперсия чистой прибыли;

$\gamma(z)$ - функция Лапласа;

$\beta(z)$ - функция Гаусса.

Порядок изложения в документах математических уравнений такой же, как и формул.

Таблицы применяют для лучшей наглядности и удобства сравнения показателей. Название таблицы должно отражать ее содержание, быть точным, кратким.

Таблицы, за исключением таблиц приложений, следует нумеровать арабскими цифрами сквозной нумерацией. Таблицы каждого приложения обозначают отдельной нумерацией арабскими цифрами с добавлением перед цифрой обозначения приложения. Если в документе одна таблица, она должна быть обозначена «Таблица 1» или «Таблица B.1», если она приведена в приложении B.

Допускается нумеровать таблицы в пределах главы. В этом случае номер таблицы состоит из номера главы и порядкового номера таблицы в главе, разделенных точкой (Таблица 2.1, 2.2, 2.3 и т.д.).

Наименование таблицы следует помещать над таблицей слева, без абзацного отступа в одну строку с ее номером через тире.

Таблица

номер

— Название таблицы с большой буквы (точка в конце не ставится)

1	2	3	4

На все таблицы работы должны быть приведены ссылки в ее тексте, при ссылке следует писать слово «таблица» с указанием ее номера.

Заголовки граф и строк таблицы следует писать с прописной буквы в единственном числе, а подзаголовки граф - со строчной буквы, если они составляют одно предложение с заголовком, или с прописной буквы, если они имеют самостоятельное значение. В конце заголовков и подзаголовков таблиц точки не ставят. Если цифровые или иные данные в какой-либо строке таблицы не приводят, то в ней ставят прочерк.

Таблицы слева, справа и снизу ограничивают линиями. Допускается применять размер шрифта в таблице меньший, чем в тексте.

Разделять заголовки и подзаголовки боковика и граф диагональными линиями не допускается. Горизонтальные и вертикальные линии, разграничивающие строки таблицы, допускается не проводить, если их отсутствие не затрудняет пользование таблицей.

Заголовки граф, как правило, записывают параллельно строкам таблицы. При необходимости допускается перпендикулярное расположение заголовков граф.

Таблицу, в зависимости от ее размера, помещают под текстом, в котором впервые дана ссылка на нее, или на следующей странице, а при необходимости, в приложении к документу.

Допускается помещать таблицу вдоль длинной стороны листа документа.

Таблицу с большим числом строк допускается переносить на другой лист (страницу). При переносе части таблицы на другой лист (страницу) слово «Таблица», ее номер и наименование указывают один раз слева (без абзацного отступа) над первой частью таблицы, а над другими частями также слева пишут слова «Продолжение таблицы» и указывают номер таблицы.

Таблицу с большим количеством граф допускается делить на части и помещать одну часть под другой в пределах одной страницы. Если строки и графы таблицы выходят за формат страницы, то в первом случае в каждой части таблицы повторяется головка, во втором случае - боковик. При делении таблицы на части допускается ее головку или боковик заменять соответственно номером граф и строк. При этом нумеруют арабскими цифрами графы и (или) строки первой части таблицы. В рамках работы должно соблюдаться единообразие в оформлении переноса таблиц.

Если повторяющийся в разных строках графы таблицы текст состоит из одного слова, то его после первого написания допускается заменять кавычками; если из двух и более слов, то при первом повторении его заменяют словами «То же», а далее - кавычками. Ставить кавычки вместо повторяющихся цифр, марок, знаков, математических и химических символов не допускается.

Примечания приводят в документах, если необходимы пояснения или справочные данные к содержанию текста, таблиц или графического материала. Примечания следует помещать непосредственно после текстового, графического материала или в таблице, к которым относятся эти примечания.

Слово «Примечание» следует печатать с прописной буквы с абзаца.

Если примечание одно, то после слова «Примечание» ставится тире и примечание печатается тоже с прописной буквы. Одно примечание не нумеруют. Несколько примечаний нумеруют по порядку арабскими цифрами без проставления точки. Примечание к таблице помещают в конце таблицы над линией, обозначающей окончание таблицы.

Иллюстрации (чертежи, графики, схемы, диаграммы и др.) следует располагать в работе непосредственно после текста, в котором они упоминаются впервые, или на следующей странице.

Иллюстрации могут быть в компьютерном исполнении, в том числе и цветные. На все иллюстрации должны быть даны ссылки в работе.

Чертежи, графики, диаграммы, схемы, иллюстрации, помещаемые в курсовую работу, должны соответствовать требованиям государственных стандартов Единой системы конструкторской документации (ЕСКД).

Допускается выполнение чертежей, графиков, диаграмм, схем посредством использования компьютерной печати.

Количество иллюстраций должно быть достаточным для пояснения излагаемого текста. Иллюстрации, за исключением иллюстраций приложений, следует нумеровать арабскими цифрами сквозной нумерацией. Если рисунок один, то он обозначается «Рисунок 1».

Допускается нумеровать иллюстрации в пределах главы. В этом случае номер иллюстрации состоит из номера главы и порядкового номера иллюстрации в главе, разделенных точкой.

Иллюстрации должны иметь наименование. Слово «Рисунок» и его наименование располагают под иллюстрацией посередине строки: Рисунок 1.1 – Название рисунка (с большой буквы, точку в конце не ставить).

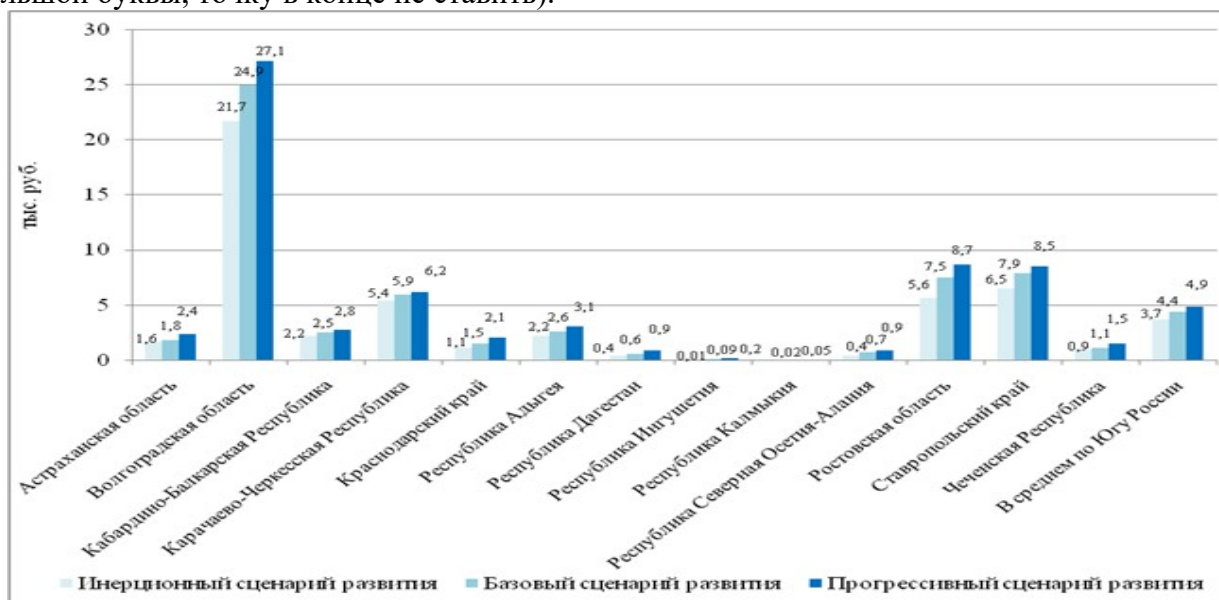


Рисунок 1 – Прогнозный выпуск инновационной продукции на душу населения в регионах Юга России в 2021 году, тыс. руб.

Иллюстрации каждого приложения обозначают отдельной нумерацией арабскими цифрами с добавлением перед цифрой обозначения приложения. Например: Рисунок А.3.

При ссылках на иллюстрации следует писать «... в соответствии с рисунком 2» при сквозной нумерации и «... в соответствии с рисунком 1.2» при нумерации в пределах главы.

Список литературы оформляется в соответствии с требованиями ГОСТа к оформлению библиографии; в нем указываются все использованные в работе источники литературы и документации (приложение М).

Допускаются следующие способы группировки библиографических записей: алфавитный, систематический или в порядке появления ссылок на источники в тексте. При алфавитном способе группировки все библиографические записи располагают по алфавиту фамилий авторов или первых слов заглавий документов. Библиографические записи произведений авторов-однофамильцев располагают в алфавите их инициалов. При систематической (тематической) группировке материала библиографические записи располагают в определенной логической последовательности в соответствии с принятой системой классификации.

Приложение оформляют как продолжение данного документа на последующих его листах или выпускают в виде самостоятельного документа.

Приложения могут быть обязательными и информационными.

Информационные приложения могут быть рекомендуемого или справочного характера.

В тексте документа на все приложения должны быть даны ссылки. Степень обязательности приложений при ссылках не указывается. Приложения располагают в порядке ссылок на них в тексте документа. Каждое приложение следует начинать с новой страницы с указанием наверху посередине страницы слова «Приложение» и его обозначения. Приложение

должно иметь заголовок, который записывают симметрично относительно текста с прописной буквы отдельной строкой.

Приложения обозначают заглавными буквами русского алфавита, начиная с А, за исключением букв Ё, З, Й, О, Ч, Ь, Ы, Ъ. После слова «Приложение» следует буква, обозначающая его последовательность. Допускается обозначение приложений буквами латинского алфавита, за исключением букв I и O.

В случае полного использования букв русского и латинского алфавитов допускается обозначать приложения арабскими цифрами.

Если в документе одно приложение, оно обозначается «Приложение А».

Приложения, как правило, выполняют на листах формата А4.

Текст каждого приложения, при необходимости, может быть разделен на разделы, подразделы, пункты, подпункты, которые нумеруют в пределах каждого приложения. Перед номером ставится обозначение этого приложения. Приложения должны иметь общую с остальной частью документа сквозную нумерацию страниц.

Все приложения должны быть перечислены в содержании курсовой работы (при наличии) с указанием их обозначений и заголовков.

Последовательность выполнения задания

Содержание основных этапов подготовки курсовой работы представлено в таблице:

№ п/п	Содержание блока	Число %	Срок выполнения (номер недели)	Рекомендуемая литература
1	Получения задания. Литературный обзор. Теоретическая часть	15%	1-5	Основная литература: 1,2,3 Дополнительная литература: 1-2
2	Аналитический обзор предметной области.	15%	6	Основная литература: 1,2,3
3	Практическая часть	45%	7-8	Дополнительная литература: 1-2
5	Оформление отчета	15%	9	Основная литература: 1,2,3
6	Защита работы	10%	10-11	
	Итого:	100%		

Критерии оценивания

Оценка «отлично» выставляется студенту, если в курсовой работе глубоко и прочно усвоен программный материал, исчерпывающе, последовательно, четко и логически стройно построен, тесно увязаны теория с практикой, решены поставленные задачами, вопросы. В работе использована литература, правильно обосновывается принятое решение, раскрыты приемы выполнения практических задач. Компетенции ПК-3 освоены на повышенном уровне.

Оценка «хорошо» выставляется студенту, если в курсовой работе глубоко и прочно усвоен программный материал, исчерпывающе, последовательно, четко и логически стройно построен, тесно увязаны теория с практикой, решены поставленные задачами, вопросы. В работе правильно обосновывается принятое решение, раскрыты приемы выполнения практических задач, но существуют ошибки в выполнении задач, неправильно записаны запросы, построены и спроектированы задачи. Компетенции ПК-3 освоены полностью на базовом уровне.

Оценка «удовлетворительно» выставляется студенту, если в курсовой работе допускаются неточности, недостаточно правильные формулировки, нарушения логической последовательности в изложении программного материала, неправильно выполнены практические работы. Если какая-либо часть курсовой работы является плагиатом, скомпилирована из фрагментов работ других авторов и носит несамостоятельный характер. Оформление работы совершенно не соответствует требованиям. Компетенции ПК-3 освоены частично на базовом уровне.

Оценка «неудовлетворительно» выставляется студенту, который не выполнил курсовую работу, либо выполнил с грубыми нарушениями требований, не раскрыл заявленную тему, не выполнил практической части работы. Содержание курсовой работы не соответствует ее теме. При написании работы не были использованы источники и литература. Компетенции ПК-3 не освоены.

Для подготовки к данному оценочному мероприятию необходимо заранее подготовиться, используя основную и дополнительную, методическую литературу, Интернет-ресурсы.

Критерии оценки навыков студентов

В ходе работы над курсовой работой продемонстрированы навыки закрепления и систематизации теоретических положений.

Широко использованы навыки научно-исследовательской деятельности.

Применялись навыки самостоятельного теоретического и практического исследования в соответствии с направлением обучения

Уровень навыков обработки, анализа и систематизации результатов исследований, как теоретического, так и практического характера.

Полученные результаты имеют практическую значимость в соответствующей области.

Критерии оценки подготовки курсовой работы

Работа с научной литературой, со справочниками и другими информационными источниками, в том числе электронными ресурсами, в полной мере соответствует уровню научного исследования.

Курсовая работа подготовлена в соответствии с требованиями, предъявляемыми к научно-исследовательской работе.

Мысли, выводы, результаты исследования изложены студентом научным языком, без художественных, просторечных, разговорных оборотов и фразеологизмов.

При подготовке курсовой работы студент провел масштабную работу с литературой и специальными источниками.

Уровень самостоятельности исследования подтвержден проверкой курсовой работы в системе "Антиплагиат".

Критерии оценки соответствия курсовой работы требованиям

В курсовой работе должен присутствовать анализ, проведена систематизация теоретических материалов по избранной теме. Введение должно быть написано с использованием научного аппарата.

Курсовая работа должна быть написана самостоятельно и содержать критическое осмысление изученных литературных и специальных источников.

Изложение материала в курсовой работе должно быть конкретным и соответствовать теме исследования. Курсовая работа должна быть насыщена фактическими данными, цитатами, таблично-графическим материалом, иметь сноски на использованные источники.

В заключении курсовой работы должны быть сформулированы выводы по результатам проведенного исследования в соответствии с поставленными задачами исследования.

Использованный материал из литературных, специальных, нормативно-правовых и электронных источников должен быть переработан студентом самостоятельно, увязан с исследуемой темой и изложен своими словами.

Курсовая работа, удовлетворяющая предъявленным требованиям, допускается к защите. Защита курсовой работы происходит на комиссии в составе заведующего кафедрой, руководителя курсовой работы и одного преподавателя кафедры.

При защите работы оцениваются:

- раскрытие целей и задач курсовой работы, актуальность, описание выполненной работы, основные выводы и предложения, разработанные студентом в процессе написания курсовой работы;
- умение студента формулировать свои мысли, передавать суть вопроса, доказывать и отстаивать свою точку зрения на проблему в процессе защиты.

Порядок защиты курсовой работы

Защита курсовой работы является обязательной формой проверки выполнения работы. Защита производится на заседании кафедры, научно-методического семинара кафедры, научной проблемной группы специальной комиссией, утверждаемой директором института, состоящей обычно из 3 преподавателей кафедры, при непосредственном участии руководителя, в присутствии студентов. Результаты наиболее интересных курсовых работ (проектов) могут быть доложены на научных конференциях. Публичная защита стимулирует научный интерес, творчество, ответственность студентов.

Защита состоит в коротком докладе студента по выполненной работе и в ответах на вопросы присутствующих на защите. Научный руководитель зачитывает отзыв на курсовую работу (проект) студента (Приложение 3).

Результаты защиты курсовой работы, согласно действующему Положению о текущем контроле и промежуточной аттестации в СКФУ, оцениваются дифференцированной отметкой по пятибалльной системе. Оценка курсовой работы заносится в зачетную книжку студента и зачетно-экзаменационную ведомость, составляемую в 2-х экземплярах, один из которых хранится на кафедре в течение всего срока обучения студента, другой представляется в дирекцию института (филиала).

Защита курсовых работ (проектов), предусмотренных учебным планом, проводится не позднее, чем за две недели до начала зачетно-экзаменационной сессии.

Студент, не представивший в установленный срок курсовую работу (проект) или не защитивший ее по неуважительной причине, считается имеющим академическую задолженность.

Список литературы, рекомендуемый к использованию

Основная литература:

1. Управление данными [Электронный ресурс] : учебник / Ю.Ю. Громов [и др.]. — Электрон. текстовые данные. — Тамбов: Тамбовский государственный технический университет, ЭБС АСВ, 2015. — 192 с. — 978-5-8265-1385-9. — Режим доступа: <http://www.iprbookshop.ru/63912.html>

2. Кузнецов С.Д. Введение в реляционные базы данных [Электронный ресурс] / С.Д. Кузнецов. — 2-е изд. — Электрон. текстовые данные. — М. : Интернет-Университет Информационных Технологий (ИНТУИТ), 2016. — 247 с. — 5-9556-00028-0. — Режим доступа: <http://www.iprbookshop.ru/73671.html>

Дополнительная литература:

1. Абросимова, М.А. Базы данных: проектирование и создание программного приложения в СУБД MS Access : практикум / М.А. Абросимова ; Министерство образования и

науки Российской Федерации, Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования «Уфимский государственный университет экономики и сервиса», Кафедра «Информатика и ИКТ». - Уфа : Уфимский государственный университет экономики и сервиса, 2014. - 56 с. - Библиогр. в кн. ; То же [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=272367>

2. Чурбанова, О.В. Базы данных и знаний. Проектирование баз данных в MicrosoftAccess : учебно-методическое пособие / О.В. Чурбанова, А.Л. Чурбанов ; Министерство образования и науки Российской Федерации, Северный (Арктический) федеральный университет имени М.В. Ломоносова. - Архангельск : САФУ, 2015. - 152 с. : ил., схем., табл. - Библиогр. в кн. - ISBN 978-5-261-01029-6 ; То же [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=436230>

Интернет-ресурсы:

www.biblioclub.ru – Электронно-библиотечная система «Университетская библиотека онлайн».

www.iprbookshop.ru - Электронно-библиотечная система «IPRbooks»

УТВЕРЖДАЮ
Заведующий кафедрой
Цифровых бизнес-технологий и систем учета

_____ А.А. Калашников

Институт экономики и управления
Кафедра цифровых бизнес-технологий и систем учета

Направление Бизнес-информатика

ЗАДАНИЕ на курсовую работу

студента Иванова Иван Ивановича
по дисциплине Базы данных

1. Тема работы Автоматизация управления эффективностью бизнеса на основе системы сбалансированных показателей
2. Цель состоит в изучении аспектов автоматизации управления эффективностью бизнеса на основе системы сбалансированных показателей.
3. Задачи
 - 3.1 выявить ключевые аспекты ССП как инструмента управления бизнеса;
 - 3.2 рассмотреть основные принципы ССП;
 - 3.3 охарактеризовать особенности ССП в части проектирование бизнеса, карты стратегии.
4. Перечень подлежащих разработке вопросов
 - а) по теоретической части рассматривается Сбалансированная система показателей как инструмент реализации стратегии
 - б) по проектной части рассматриваются особенности автоматизации управления эффективностью бизнеса.
5. Исходные данные
 - а) по литературным источникам Интернет-сайты, журналы, газеты
 - б) по вариантам, разработанным преподавателем

- в) иное данные информационных агентств, рейтинговых агентств
6. Список рекомендуемой литературы
 - 6.1 Д. Нортон, Р. Каплан. Сбалансированная система показателей. От

стратегии к действию. - М.: Олимп-Бизнес, 2012.

6.2 Д. Нортон, Р. Каплан. Стратегическое единство. Создание синергии организации с помощью сбалансированной системы показателей - М.: Вильямс, 2012.

6.3 Евченко А.В., Методы исследования систем управления [Текст]: Учеб.пособие. /А.В. Евченко, Э.Н. Кузьбожев/-Курск: Курск. Гос. техн. ун-т, 2001. 168 с.

6.4 Игнатьева А. В. Исследование систем управления [Текст] : учебное пособие для студентов вузов. - М. : ЮНИТИ-ДАНА, 2002. - 157 с.

6. Контрольные сроки предоставления отдельных разделов курсовой работы
- 25% - подготовительная часть «11» марта 2027 г.
 - 50% - теоретическая часть «31» марта 2027 г.
 - 75% - проектная часть «19» апреля 2027 г.
 - 100% - курсовая работа к защите – «13» мая 2027 г.

7. Срок защиты курсовой работы «24» мая 2027 г.

Дата выдачи задания «15» февраля 2027 г.

Руководитель курсовой работы
кандидат экономических наук,
доцент

А.А. Калашников

Задание принял к исполнению студент очной формы обучения 2 курса группы
БИН-б-о-26-1

И.И. Иванов

_____ (личная подпись)

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ
ФГАОУ ВО «СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
ИНСТИТУТ ЭКОНОМИКИ И УПРАВЛЕНИЯ
КАФЕДРА ЦИФРОВЫХ БИЗНЕС-ТЕХНОЛОГИЙ И СИСТЕМ УЧЕТА

КУРСОВАЯ РАБОТА

по дисциплине
«Базы данных»

на тему
«Автоматизация управления эффективностью бизнеса на основе системы
сбалансированных показателей»

Выполнил:

Иванов Иван Иванович
студент 2 курса
группы БИН-б-о-
очной формы обучения

Руководитель работы:

Калашников Александр
Александрович
доцент кафедры цифровых бизнес-
технологий и систем учета

Работа допущена к защите _____ .

Работа выполнена и
защищена с оценкой _____ Дата защиты _____

Члены комиссии: доцент кафедры _____ =
(должность) (подпись) (И.О. Фамилия)

доцент кафедры _____
доцент кафедры _____

Ставрополь, 2027 г.

Отзыв
на курсовую работу студента 2 курса
Иванова Иван Ивановича
«Автоматизация управления эффективностью бизнеса на основе системы
сбалансированных показателей»

Актуальность: курсовая работа посвящена изучению методик автоматизации управления эффективностью бизнеса на основе системы сбалансированных показателей. В настоящее время любая компания нуждается в бизнес-процессах, оптимально организованных для выживания в условиях глобальной конкуренции и динамичной бизнес среды. На сегодняшний день наиболее эффективным инструментом стратегического управления является Сбалансированная система показателей – ССП (BalancedScorecard - BSC).

С этих позиций особую актуальность приобретает изучение теоретических основ автоматизации управления бизнеса.

В первой главе рассматриваются основы целевого управления, ССП, ключевые аспекты, показатели.

Вторая глава посвящена непосредственно этапам разработки проекта, формулированию карты стратегии, построению карт распределения ответственности.

Выводы, сделанные в Заключение, соответствуют целям, поставленным во Введении.

Проанализирован значительный объем литературы – 34 источника.

За время работы студент проявил себя как творческий и инициативный исследователь. В процессе подготовки и написания работы студент соблюдал сроки выполнения календарного графика и проявил отличные навыки в работе с теоретическими источниками.

Таким образом, работа выполнена на высоком теоретическом и практическом уровне, соответствует требованиям, предъявляемым к курсовым работам, и заслуживает высокой оценки.

Научный руководитель
кандидат экономических наук,
доцент

«.....» 202_ г.