

Краевое государственное автономное
профессиональное образовательное учреждение
«Кунгурский колледж агротехнологий и управления»

МЕТОДИЧЕСКИЕ УКАЗАНИЯ
ПО ВЫПОЛНЕНИЮ ПРАКТИЧЕСКИХ И ЛАБОРАТОРНЫХ РАБОТ
ПО ДИСЦИПЛИНЕ
ОП.08 ОСНОВЫ ПРОЕКТИРОВАНИЯ БАЗ ДАННЫХ
09.02.07 «Информационные системы и программирование»

2023 г.

Рассмотрено на заседании МК
информационных дисциплин от
«30» августа 2023 г.

Председатель МК
 А.В.Атушкина

Утверждаю
Зам. директора


Л.И.Петрова

Организация-разработчик: ГБПОУ «ККАТУ»

Составитель: А.В.Атушкина

Пояснительная записка

Методические указания по выполнению лабораторных и практических работ по ОП.08 Основы проектирования баз данных разработаны в соответствии с рабочей программой дисциплины и предназначены для приобретения необходимых практических навыков и закрепления теоретических знаний, полученных обучающимися при изучении профессионального модуля, обобщения и систематизации знаний перед дифференцированным зачетом.

Методические указания предназначены для обучающихся специальности 09.02.07 Информационные системы и программирование.

Дисциплина ОП.08 Основы проектирования баз данных относится к общепрофессиональному циклу, изучается на 2 курсе и при его изучении отводится значительное место выполнению практических работ.

Освоение содержания ОП.08 Основы проектирования баз данных во время выполнения практических работ обеспечивает достижение обучающимися следующих **результатов**:

Код	Наименование общих компетенций
ОК 1	Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам.
ОК 2	Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности.
ОК 4	Работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами.
ОК 5	Осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста.
ОК 9	Использовать информационные технологии в профессиональной деятельности.
ОК 10	Пользоваться профессиональной документацией на государственном и иностранном языке.

Рабочая программа дисциплины предусматривает проведение практических работ в объеме 38 часов.

Порядок выполнения практической работы

- записать название работы, ее цель в тетрадь;

- выполнить основные задания в соответствии с ходом работы;
- выполнить индивидуальные задания.

Рекомендации по оформлению практической работы

Задания выполняются обучающимися по шагам. Необходимо строго придерживаться порядка действий, описанного в практической работе.

Результаты выполнения практических работ необходимо сохранять в своей папке на компьютере или USB – накопителе.

В случае пропуска занятий обучающийся осваивает материал самостоятельно в свободное от занятий время и сдает практическую работу с пояснениями о выполнении.

Требования к технике безопасности при выполнении лабораторных/практических работ

Вводный инструктаж

Инструкция №1 ПРАВИЛА ПОВЕДЕНИЯ СТУДЕНТОВ В КАБИНЕТЕ ИНФОРМАТИКИ

Кабинет информатики относится к кабинетам повышенной опасности, находясь в нем необходимо соблюдать требования настоящей инструкции.

1. Не заходите в кабинет без разрешения преподавателя.
2. Во время перемены все студенты выходят в коридор. В кабинете кроме преподавателя могут находиться только дежурные.
3. Запрещается находиться в кабинете в верхней одежде, грязной обуви.
4. Не бегайте по кабинет - можете получить травму или повредить оборудование.
5. Не бросайте мусор в кабинете, этим вы создаете пожарную опасность.
6. Будьте внимательны и дисциплинированы, точно выполняйте указания преподавателя.
7. Не приступайте к выполнению работы без разрешения преподавателя.
8. Не пользуйтесь электрическими розетками для шалости, это опасно для вашей жизни.

Инструктаж №2 ПРАВИЛА РАБОТЫ СТУДЕНТОВ ЗА КОМПЬЮТЕРОМ ПЕРЕД ВЫПОЛНЕНИЕМ РАБОТЫ необходимо выслушать преподавателя о ее содержании и ходе выполнения.

ВО ВРЕМЯ РАБОТЫ

1. Компьютер можно включать только после разрешения преподавателя.

2. ЗАПРЕЩАЕТСЯ:

- прикасаться к проводам, лишенных изоляции;
- включать компьютер со снятым корпусом;
- производить подключение устройств к включенному компьютеру;
- прикасаться руками и острыми предметами к экрану монитора, внутренним частям компьютера;
- есть за компьютером;
- включать компьютер сразу же после его выключения (необходимо подождать 10-15 секунд).

3. Обнаружив неисправность в электрических устройствах, находящихся под напряжением, необходимо немедленно отключить источник электропитания и сообщить об этом преподавателю.

4. Не оставляйте рабочего места без разрешения преподавателя.

ПОСЛЕ ОКОНЧАНИЯ РАБОТЫ

1. Корректно завершите работу компьютера.

2. Сдай рабочее место преподавателю.

Инструктаж №3 ПРАВИЛА ПОЖАРНОЙ БЕЗОПАСНОСТИ

1. В кабинете должны, быть средства для тушения пожара: огнетушитель.

2. Кабинет должен содержаться в чистоте. Весь сгораемый мусор следует систематически выносить.

3. В кабинете не допускается хранение легковоспламеняющихся жидкостей.

4. ЗАПРЕЩАЕТСЯ:

- допускать к работе студентов, не ознакомленных с правилами техники безопасности;
- оставлять без присмотра включенное в электрическую сеть оборудование;
- подвешивать электропроводку на гвоздях, использовать электропровода с поврежденной изоляцией, некалиброванные предохранители, обертывать электрические лампы бумагой или тканью, подвешивать стенды, таблицы и пр. на электропровода;
- работа в кабинете с нагревательными приборами;
- производить сложный ремонт компьютерной техники.

5. По окончании работы необходимо тщательно осмотреть помещение, устранить все недочеты, отключить напряжение электросети с помощью рубильника.

6. В случае возникновения пожара необходимо:

- отключить напряжение электрической сети;
- немедленно эвакуировать студентов;
- принять меры по тушению пожара;
- сообщить о пожаре по телефону 01 или 112.

Перечень практических работ

Пр.р.№1. Создание связей между таблицами.

Пр.р.№2. Создание простой БД в режиме «Конструктор».

Пр.р.№3. Создание ER - моделей

Пр.р.№4. Создание файла базы данных и таблиц

Пр.р.№5. Работа с многотабличной базой данных

Пр.р.№6. Создание запросов на выборку и перекрестных запросов

Пр.р.№7. Создание форм, многотабличных форм и отчетов

Пр.р.№8. Вычисляемые поля

Пр.р.№9. Создание запросов

Пр.р.№10. Работа с меню: создание, модификация, активация и удаление

Пр.р.№11. Создание интерфейса (входной формы).

Пр.р.№12. Управление внешним видом

Пр.р.№13. Поиск данных в таблице. Установка даты и вывод записей на экран.

Пр.р.№14. Создание и печать отчетов

Пр.р.№15. Добавление записей при помощи языка SQL

Пр.р.№16. Редактирование записей при помощи языка SQL

Пр.р.№17. Выборка данных из таблиц при помощи языка SQL

Пр.р.№18. Управление транзакциями с помощью языка SQL

Критерии оценки выполнения практических работ

Оценки	Критерии оценок
«5»	- обучающийся подбирает необходимые для выполнения предлагаемых работ источники знаний (литература, материалы, инструменты), показывает необходимые для проведения практической работы теоретические знания. Правильно оформлена практическая работа, соблюдена технологическая последовательность выполнения данного вида работ. Работа оформлена аккуратно.
«4»	- практическая работа выполняется обучающимся в полном объёме и самостоятельно. Обучающийся использует указанные преподавателем источники информации. Могут быть неточности и небрежность в оформлении работы. Работа показывает знания обучающимися основного теоретического материала, но имеются незначительные ошибки при оформлении практической части работы.
«3»	- обучающийся выполняет и оформляет практическую работу полностью с помощью преподавателя или хорошо подготовленных и уже выполнивших на «отлично» данную работу других обучающихся.
«2»	- практическая работа не выполнена полностью за отведенное время по неуважительной причине.

Перечень учебных изданий, Интернет-ресурсов, дополнительной литературы

Основные печатные издания

1. Федорова Г.Н. Основы проектирования баз данных: учебник. – М.: Издательский центр «Академия», 2021. – 224 с.

Основные электронные издания

1. Илюшечкин В.М. Основы использования и проектирования баз данных: учебник для среднего профессионального образования / В.М. Илюшечкин,

испр. и доп. – М.: Издательство Юрайт, 2021. – 213 с. – (Профессиональное образование). – ISBN 978-5-534-01283-5. – Текст: электронный // Образовательная платформа Юрайт [сайт]. – URL: <https://urait.ru/bcode/471698> (дата обращения: 20.08.2023).

Пример

Практическая работа № 5

Работа с многотабличной базой данных

Цель работы:


1. *Проектирование структуры базы данных. Создать вариант базы данных с двумя таблицами: “Список” и “Группы”.*
2. *Конструирование пустых таблиц базы данных.*
3. *Создание схемы базы данных. В данном случае таблицы связаны связью один-ко-многим. Это значит, что в таблице “Группы” каждое значение связанного поля может встречаться только один раз, а в таблице “Список” – несколько раз (несколько человек могут быть из одной группы). Связи следует устанавливать при пустых таблицах.*
4. *Ввод данных в таблицы. Создать форму для ввода данных.*


Порядок работы.

Задание 1. Создание базы данных – Учебная база.

1. *Запустите программу СУБД Microsoft Access. Для этого выполните:
Пуск – Все программы – Microsoft office – Microsoft office Access 2010.*
2. *Выберите команду Новая база данных. Затем введите имя файла – Учебная база и нажмите кнопку Создать.*

Задание 2. Создание таблицы “Группы”.

1. *Создайте таблицу под именем “Группы” с помощью конструктора таблиц. Для этого выполните команду: Создание – конструктор таблиц.*
2. *Заполните Имя поля следующими данными (заголовками столбцов): Учебная группа, Преподаватель и соответственно Тип данных для них: Учебная группа – ЧИСЛОВОЙ, Преподаватель – ТЕКСТОВЫЙ. Сделайте поле “Учебная группа” ключевым, установив курсор на имя поля и щелкнув по кнопке **Ключевое поле** .*


3. Сохраните таблицу под именем “Группы”, щелкнув по кнопке **Сохранить** .
4. Закройте таблицу.

Задание 3. Создание таблицы “Список”.

1. Создайте таблицу под именем “Список” с помощью конструктора таблиц. Для этого выполните команду: Создание – конструктор таблиц.

2. Заполните Имя поля следующими данными (заголовками столбцов): Код, Фамилия, Имя,

Отчество, Год рождения, Класс, Учебная группа. Выберите из ниспадающего списка числовой тип данных для поля “Код”. Сделайте поле “Код” ключевым, установив курсор на имя поля и щелкнув по кнопке


Ключевое поле . Тип данных полей “Фамилия”, “Имя”, “Отчество” – текстовый, полей “Год рождения”, “Школа”, “Класс” – числовой.

Установите тип данных поля “Учебная группа” числовой. Общие свойства поля не меняйте. Выберите вкладку **Подстановка**, тип


элемента управления – **Поле со списком**, источник строк – **Группы**.

3. Заполните таблицу данными:

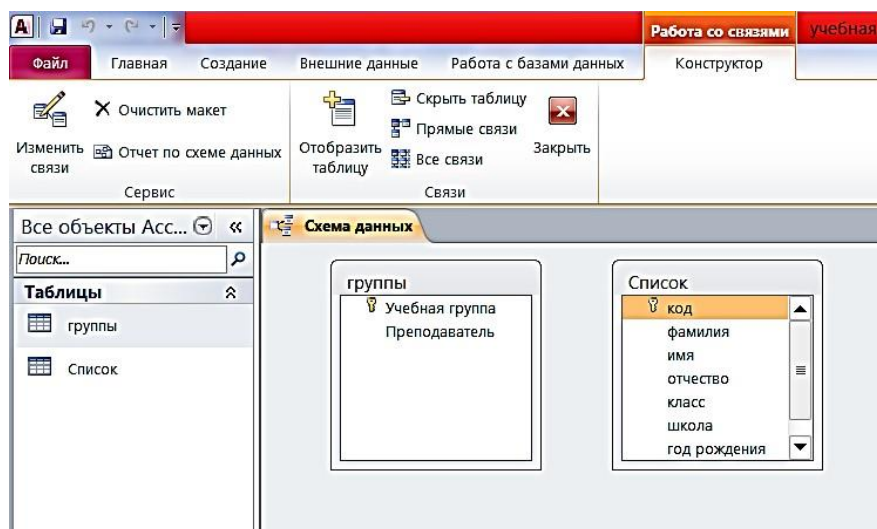
Код	Фамилия	Имя	Отчество	Год рождения	Школа	Класс	Учебная группа
1	Чернова	Наталья	Алексеевна	2001	5	9	101
2	Куликов	Клим	Алексеевич	1999	2	10	103
3	Архипов	Виктор	Александрович	2000	5	9	101
4	Баранкова	Александра	Николаевна	1998	5	11	104
5	Новоселов	Алексей	Иванович	2000	3	9	105

4. Сохраните таблицу под именем “Список”, щелкнув по кнопке **Сохранить** .
5. Закройте таблицу.

Задание 4. Создание схемы данных.

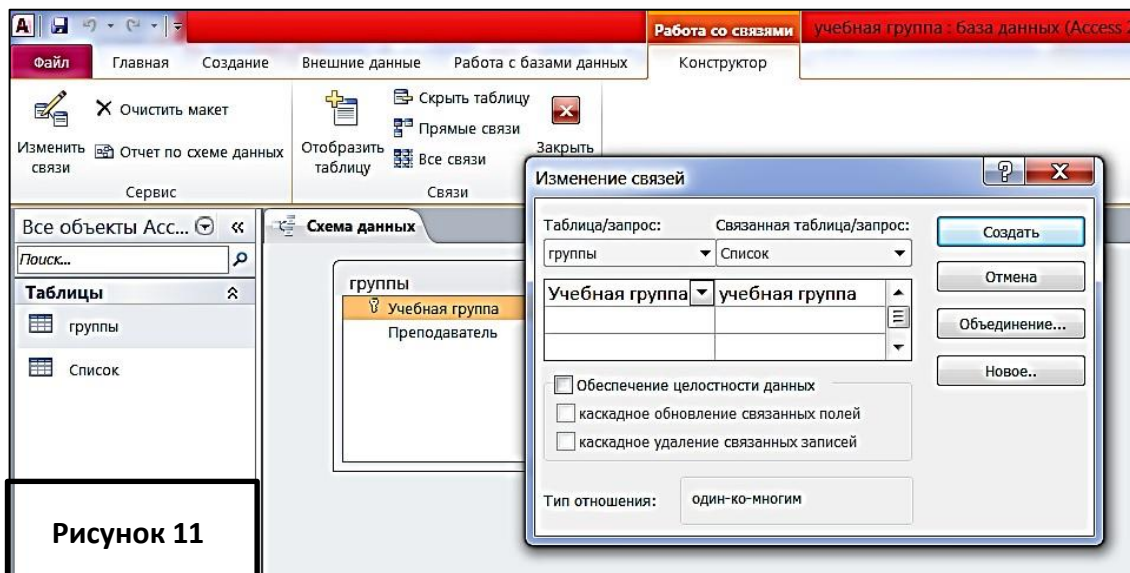
1. В ленточном меню выберите вкладку **Работа с базами данных**, щелкните по кнопке **Схема данных** . Появится окно **Схема данных**.

2. В появившемся окне **Добавление таблицы** выделите таблицу “**Группы**” и щелкните по кнопке **Добавить**, выделите таблицу “**Список**” и щелкните по кнопке **Добавить**. В окне **Схема данных** появится условный вид этих таблиц. Щелкните по кнопке **Закреть** окна **Добавление таблицы**.



3. Увеличьте окно таблицы “**Список**” так, чтобы были видны все поля.

4. Установите курсор мыши на имя поля “**Учебная группа**” в таблице “**Группы**” и, не отпуская кнопку мыши, перетащите курсор мыши на поле “**Учебная группа**” таблицы “**Список**”. Отпустите кнопку мыши. Появится диалоговое окно **Изменение связей**, представленное на рисунке (Рис.11):



5. Установите флажок **Обеспечение целостности данных**. Это невозможно будет сделать, если типы полей “Учебная группа” заданы не одинаково.

6. Установите флажок **каскадное обновление связанных полей**. Это приведет к тому, что при изменении номера группы в таблице “Группы” автоматически изменится соответствующий номер в таблице “Список”.

7. Установите флажок **каскадное удаление связей**. Это приведет к тому, что при удалении записи с номером группы из таблицы “Группы”, будут удалены все записи из таблицы “Список”, в которых находились соответствующие номера групп.

8. Щелкните по кнопке **Создать**. Появится связь “один-ко-многим”. Схема данных представлена на рисунке (Рисунок 12).

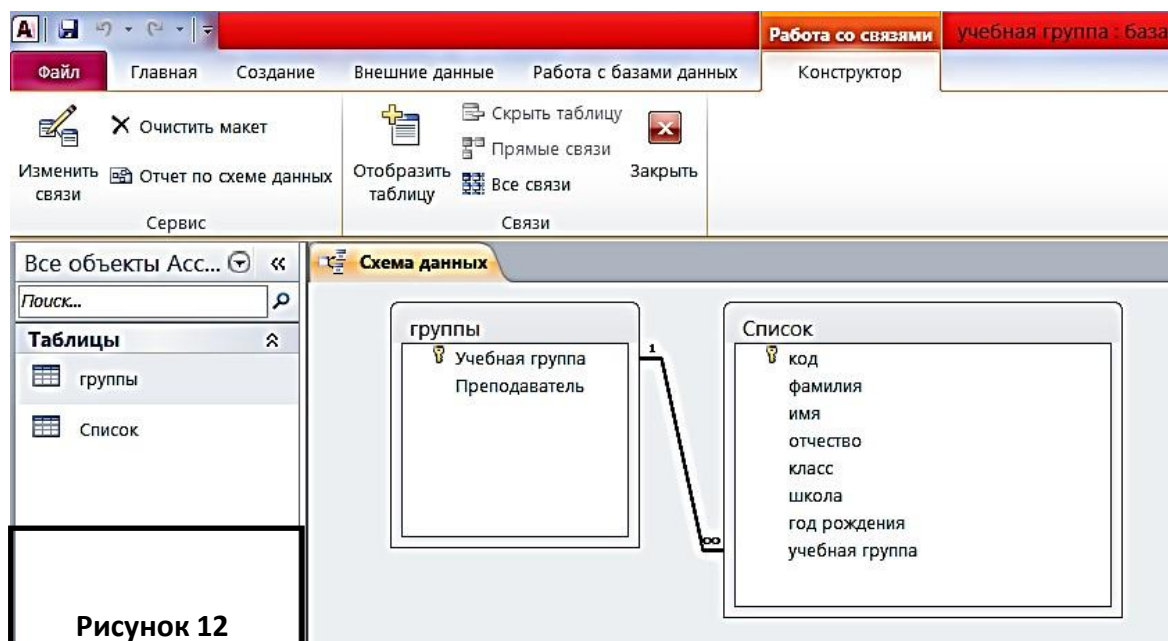



Рисунок 12

9. Закройте схему данных, щелкнув по кнопке  в верхнем правом углу окна и ответив утвердительно на вопрос о сохранении схемы данных.

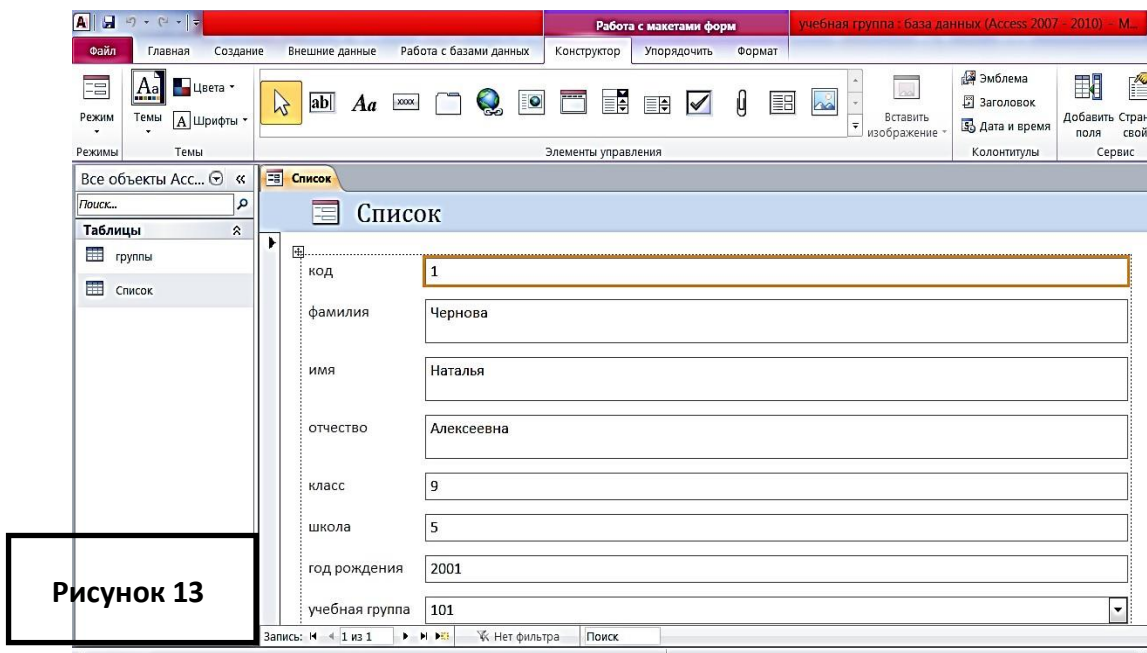
Задание 5. Заполнение таблицы “Группы” значениями.

1. Откройте таблицу “Группы” в режиме Таблицы.
2. Заполните ее записями из таблицы 1. Таблица 1

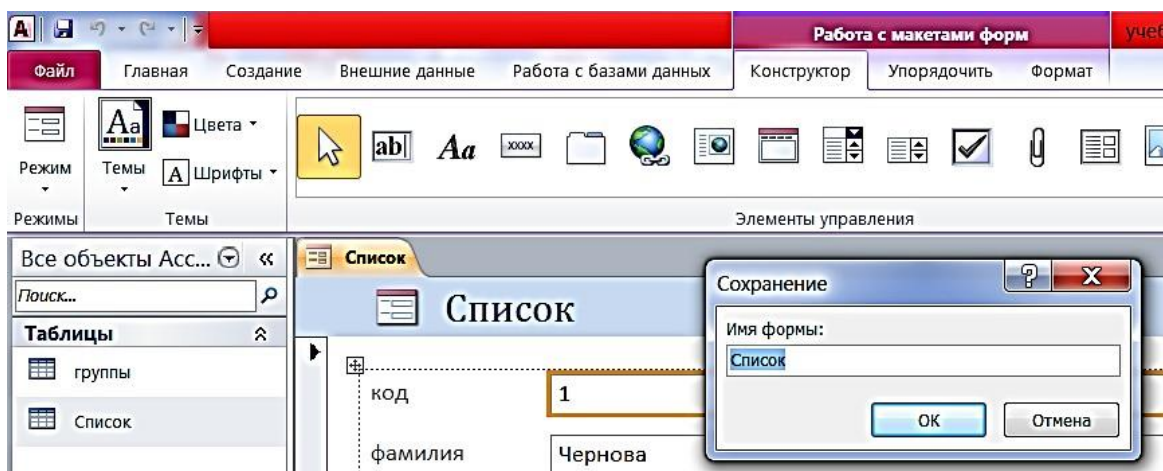
Учебная группа	Преподаватель
101	Смирнова З.В.
102	Вильданова Р.Ш.
103	Зайцева С.А.
104	Зазулина И.В.
105	Друк Н.Н.

Задание 6. Создание формы для ввода данных.

1. В окне базы данных выделите имя таблицы **Список** и выполните команду **Создание – Форма**.
2. Появится форма для просмотра данных таблицы, представленная на рисунке (Рисунок 13).



3. Закройте форму, присвоив ей имя Список.



Задание 7. Проверка каскадного обновления связанных полей.

1. Откройте таблицу "Группы".
2. Исправьте номера учебных групп на 201, 202, 203, 204, 205. Сохраните таблицу.
3. Откройте таблицу "Список". Убедитесь, что значения поля "Учебная группа" изменились. Закройте таблицу "Список".

Задание 8. Проверка каскадного удаления связанных полей.

1. Откройте таблицу “Группы”.
2. Удалите первую запись (Выделите первую строку, правой кнопкой мыши вызовите контекстное меню и выберите команду Удалить запись).
3. Ответьте утвердительно на запрос об удалении. Закройте таблицу “Группы”.
4. Откройте таблицу “Список”. Убедитесь, что исчезли записи с номером группы 201.

Задание 9. Завершение работы с программой Access.

1. Выполните команду Файл – Выход.
Если вы производили редактирование в базе данных, появится вопрос о сохранении изменений. Ответьте утвердительно.

Практическая работа № 17

Выборка данных из таблиц при помощи языка SQL

Цель: изучить принципы работы с базой данных, изучить спецификации запроса языка баз данных SQL, получить практические навыки составления и содержательной интерпретации запросов выборки данных (операторов SELECT), а также их выполнения с использованием СУБД MS Access.

Простейшие SELECT-запросы

Оператор SELECT (выбрать) языка SQL является самым важным и самым часто используемым оператором. Он предназначен для выборки информации из таблиц базы данных. Упрощенный синтаксис оператора SELECT выглядит следующим образом.

```
SELECT [DISTINCT] <список атрибутов>  
FROM <список таблиц>  
[WHERE <условие выборки>]  
[ORDER BY <список атрибутов>]  
[GROUP BY <список атрибутов>]  
[HAVING <условие>]  
[UNION <выражение с оператором SELECT>];
```


В квадратных скобках указаны элементы, которые могут отсутствовать в запросе. Ключевое слово *SELECT* сообщает базе данных, что данное предложение является запросом на извлечение информации. После слова *SELECT* через запятую перечисляются наименования полей (список атрибутов), содержимое которых запрашивается.

Обязательным ключевым словом в предложении-запросе *SELECT* является слово *FROM* (из). За ключевым словом *FROM* указывается список разделенных запятыми имен таблиц, из которых извлекается информация. Например,

```
SELECT NAME,SURNAME  
FROM STUDENT;
```

Любой *SQL*-запрос должен заканчиваться символом «*;*» (точка с запятой). Приведенный запрос осуществляет выборку всех значений полей *NAME* и *SURNAME* из таблицы *STUDENT*.

Его результатом является таблица следующего вида:

NAME	SURNAME
Иван	Иванов
Петр	Петров
Вадим	Сидоров
Борис	Кузнецов
Ольга	Зайцева
Андрей	Павлов
Павел	Котов
Артем	Лукин
Антон	Петров
Вадим	Белкин

Порядок следования столбцов в этой таблице соответствует порядку полей *NAME* и *SURNAME*, указанному в запросе, а не их порядку во входной таблице *STUDENT*.

Если необходимо вывести значения всех, столбцов таблицы, то можно вместо перечисления их имен использовать символ «***» (звездочка).

```
SELECT *  
FROM STUDENT;
```

В данном случае результатом выполнения запроса будет вся таблица *STUDENT*.

Еще раз обратим внимание на то, что получаемые в результате *SQL*-запроса таблицы не в полной мере отвечают определению реляционного

отношения. В частности, в них могут оказаться кортежи (строки) с одинаковыми значениями атрибутов.

Например, запрос «Получить список названий городов, где проживают студенты, сведения о которых находятся в таблице *STUDENT*», можно записать в следующем виде.

```
SELECT CITY FROM STUDENT;
```

Его результатом будет таблица:

CITY
Орел
Курск
Москва
Брянск
Липецк
Воронеж
Белгород
Воронеж
NULL
Воронеж

Видно, что в таблице встречаются одинаковые строки (выделены жирным шрифтом). Для исключения из результата *SELECT*-запроса повторяющихся записей используется ключевое слово *DISTINCT* (отличный).

Если запрос *SELECT* извлекает множество полей, то *DISTINCT* исключает дубликаты строк, в которых значения всех выбранных полей идентичны.

Предыдущий запрос можно записать в следующем виде.

```
SELECT DISTINCT CITY  
FROM STUDENT;
```

В результате получим таблицу, в которой дубликаты строк исключены.

CITY
Орел
Курск
Москва
Брянск
Липецк
Воронеж
Белгород
NULL

Ключевое слово *ALL* (все), в отличие от *DISTINCT*, оказывает противоположное действие, то есть при его использовании повторяющиеся строки включаются в состав выходных данных.

Режим, задаваемый ключевым словом *ALL*, действует по умолчанию, поэтому в реальных запросах для этих целей оно практически не используется.

Использование в операторе *SELECT* предложения, определяемого ключевым словом *WHERE* (где), позволяет задавать выражение условия (предикат), принимающее значение истина или ложь для значений полей строк таблиц, к которым обращается оператор *SELECT*. Предложение *WHERE* определяет, **какие строки** указанных таблиц должны быть выбраны. В таблицу, являющуюся результатом запроса, включаются только те строки, для которых условие (предикат), указанное в предложении *WHERE*, принимает значение истина.

Пример

Написать запрос, выполняющий выборку имен (*NAME*) всех студентов с фамилией (*SURNAME*) Петров, сведения о которых находятся в таблице *STUDENT*.

```
SELECT SURNAME, NAME
FROM STUDENT
WHERE SURNAME = 'Петров';
```

Результатом этого запроса будет таблица:

SURNAME	NAME
Петров	Петр
Петров	Антон

В задаваемых в предложении *WHERE* условиях могут использоваться операции сравнения, определяемые операторами = (равно), > (больше), < (меньше), >= (больше или равно), <= (меньше или равно), <> (не равно), а также логические операторы *AND*, *OR* и *NOT*.

Например, запрос для получения имен и фамилий студентов, обучающихся на третьем курсе и получающих стипендию (размер стипендии больше нуля), будет выглядеть таким образом:

```
SELECT NAME, SURNAME  
FROM STUDENT  
WHERE KURS = 3 AND STIPEND > 0;
```

Результат выполнения этого запроса имеет вид:

SURNAME	NAME
Петров	Петр
Лукин	Артем

Упражнения

Вопрос 11, 12 оформить в тетради и предоставить на проверку

Чтобы создать *SQL*-запрос, нажмите вкладку *Создать/Конструктор запросов*, после чего переключитесь в режим *SQL*.

1. Напишите запрос для вывода идентификатора (номера) предмета обучения, его наименования, семестра, в котором он читается, и количества отводимых на этот предмет часов для всех строк таблицы *SUBJECT*.
2. Напишите запрос, позволяющий вывести все строки таблицы *EXAM_MARKS*, в которых предмет обучения имеет номер (*SUBJ_ID*), равный 22.
3. Напишите запрос, выбирающий все данные из таблицы *STUDENT*, расположив столбцы таблицы в следующем порядке: *KURS*, *SURNAME*, *NAME*, *STIPEND*.
4. Напишите запрос *SELECT*, который выводит наименование предмета обучения (*SUB_NAME*) и количество часов (*HOURL*) для каждого предмета (*SUBJECT*) в 4-м семестре (*SEMESTER*).

5. Напишите запрос, позволяющий получить из таблицы EXAM_MARKS значения столбца MARK (экзаменационная оценка) для всех студентов, исключив из списка повторение одинаковых строк.
6. Напишите запрос, который выводит список фамилий студентов, обучающихся на третьем и последующих курсах.
7. Напишите запрос, выбирающий данные о фамилии, имени и номере курса для студентов, получающих стипендию больше 140.
8. Напишите запрос, выполняющий выборку из таблицы SUBJECT названий всех предметов обучения, на которые отводится более 30 часов.
9. Напишите запрос, который выполняет вывод списка университетов, рейтинг которых превышает 300 баллов.
10. Напишите запрос к таблице STUDENT для вывода списка фамилий (SURNAME), имен (NAME) - и номера курса (KURS) всех студентов со стипендией, большей или равной 100, и живущих в Воронеже.
11. Какие данные будут получены в результате выполнения запроса?

```
SELECT *
FROM STUDENT
WHERE (STIPEND < 100 OR
NOT (BIRTHDAY >= '10/03/1980'
AND STODENT_ID > 1003));
```
12. Какие данные будут получены в результате выполнения запроса?

```
SELECT *
FROM STUDENT
WHERE NOT ((BIRTHDAY = '10/03/1980' OR STIPEND > 100)
AND STUDENT_ID >= 1003);
```

Практическая работа № 7

Проектирование форм для работы с данными связанных таблиц

Цель: научиться конструировать и использовать формы для работы с данными нескольких взаимосвязанных таблиц – многотабличные формы.

Проектирование формы для работы со связанными данными выполняется на основе схемы данных, состоящей из нормализованных таблиц, связанных одно/многочисленными отношениями.

В процессе проектирования формы целесообразно выполнить приведенную ниже последовательность действий:

1. *Определить таблицу, которая будет источником записей основной части формы.*

2. Выбрать таблицы, которые будут источниками записей, загружаемых в поля основной таблицы.

3. Выбрать способ реализации для источников записей.

Задание 1. Проектирование формы для работы с данными двух связанных таблиц

Разработаем форму для таблицы **ПОКУПАТЕЛЬ**. Таблица **ПОКУПАТЕЛЬ** будет источником записей основной части формы.

1. На основе схемы данных определим данные для конструирования формы (рис.1).

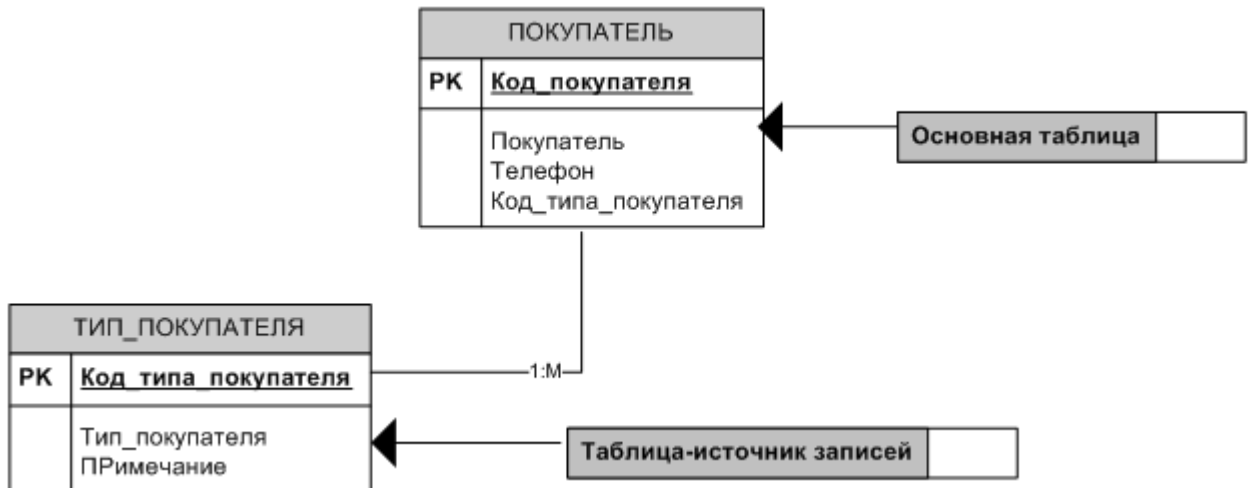


Рис.1. Подсхема данных для конструирования формы на базе таблицы **ПОКУПАТЕЛЬ**

Источник записей для поля **КОД_ТИПА_ПОКУПАТЕЛЯ** – таблица **ТИП_ПОКУПАТЕЛЯ**. Реализацию источника записей удобно выполнить с помощью элементов управления **СПИСОК** или **ПОЛЕ СО СПИСКОМ**.

2. Создадим форму в режиме **КОНСТРУКТОРА** для таблицы **ПОКУПАТЕЛЬ** (рис.2).

3. Переместим из списка полей в область данных формы поля **ПОКУПАТЕЛЬ** и **ТЕЛЕФОН**.

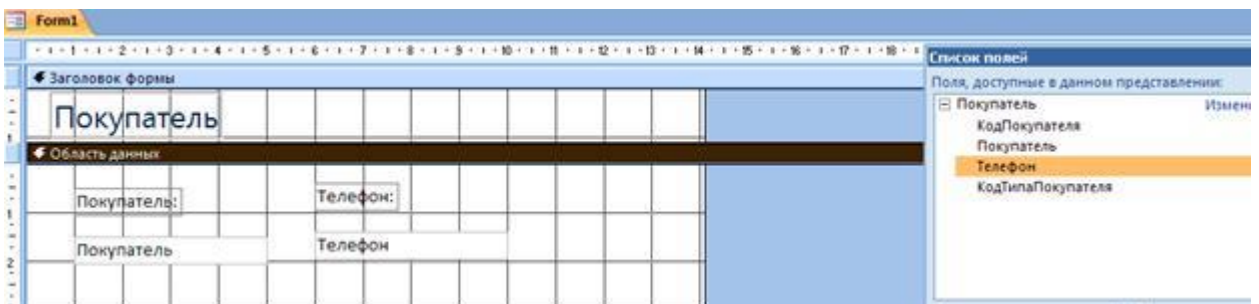


Рис.2. Создание формы **ПОКУПАТЕЛЬ**

4. Щелкните на кнопке **ПОЛЕ СО СПИСКОМ** на панели инструментов.

5. Перетащите поле **ТИП_ПОКУПАТЕЛЯ** из списка в область данных формы.

6. Появится диалоговое окно Мастера. На первом шаге выберите режим «Объект будет использовать значения из таблицы или запроса» (рис.3).

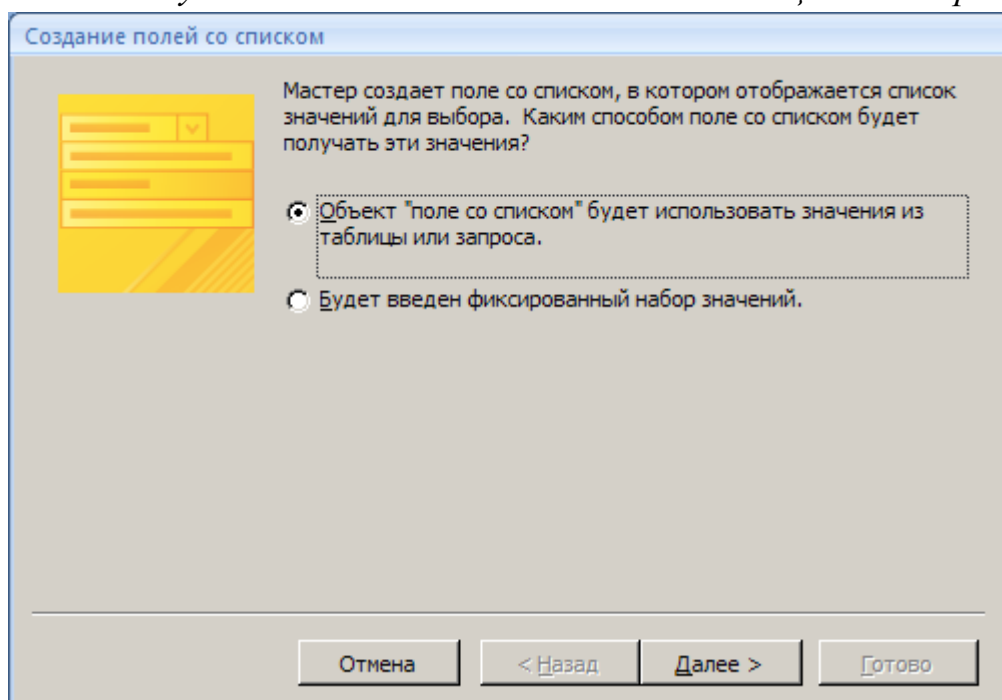


Рис.3. Первый шаг Мастера создания списков

7. На следующем шаге Мастера вам необходимо указать название таблицы-источника записей. В нашем случае укажите таблицу ТИП_ПОКУПАТЕЛЯ.

8. В следующем диалоге вам предлагают определиться с составом будущего списка. В нашем случае удобнее перенести оба поля: КОД_ТИПА является полем для связи таблиц, а поле ТИП_ПОКУПАТЕЛЯ содержит смысловую информацию для пользователя (рис. 4).

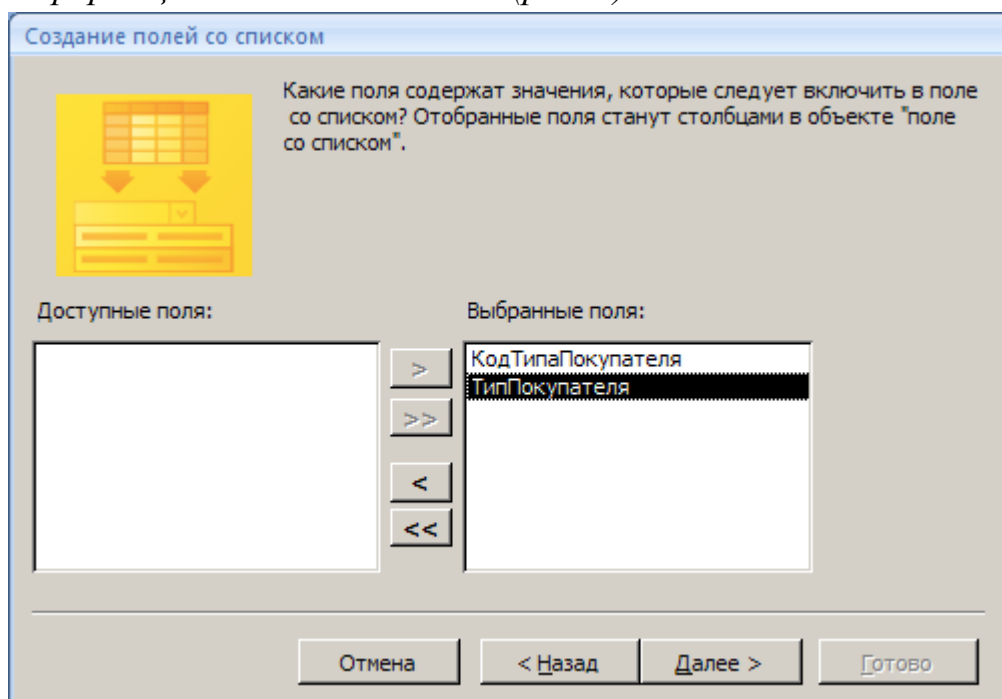


Рис.4. Выбор полей для списка

9. На следующем шаге **МАСТЕР** предполагает вам скрыть ключевой столбец. Можно согласиться с предложением, это удобно.

10. И, наконец, последний шаг **МАСТЕРА**, – в каком поле таблицы **ПОКУПАТЕЛЬ** следует сохранить значения из таблицы **ТИП_ПОКУПАТЕЛЬ**. Если названия полей совпадают (у нас это не так), то обычно, **МАСТЕР** делает верное предположение, иначе вам необходимо выбрать из списка нужное поле (**ТИП_ПОКУПАТЕЛЯ**).

11. Результат должен быть примерно таким, какой показан на рис.5.

12. Внесите несколько записей с помощью новой формы.

13. Убедитесь, что записи попадают в таблицу **ПОКУПАТЕЛЬ**.

14. Добавьте кнопку для удаления текущей записи (рис.5).

Рис.5. Окончательный вид формы для 2-х связанных таблиц

Самостоятельно

1. Разработайте и выполните реализацию форм для таблиц **ПОЛКА** (рис.6) и **ТОВАР**.

2. При конструировании форм используйте элементы управления **СПИСОК** и **ПОЛЕ СО СПИСОКОМ**.

Рис.6. Использование элемента управления **СПИСОК**

Упражнение 2. Проектирование многотабличных форм

Многотабличная форма создается для работы с данными нескольких взаимосвязанных таблиц. Многотабличная форма может состоять из **основной части** и одной или нескольких **подчиненных** включаемых форм, то есть быть составной.

Проектирование многотабличных форм выполняется на основе схемы данных, состоящей из нормализованных таблиц, связанных одно/многочисленными отношениями.

В процессе проектирования формы целесообразно выполнить приведенную ниже последовательность действий:

- Определить таблицы, которые будут источником записей основной части формы.
- Выбрать таблицы, которые будут источниками записей подчиненных форм, включаемых в составную форму.
- Определить состав полей каждой из форм.
- Выбрать способы реализации подчиненных форм и источников записей.

Многотабличная форма может быть создана в режиме КОНСТРУКТОРА или с помощью МАСТЕРА форм. Наиболее технологичным является использование КОНСТРУКТОРА.

1. На основе схемы данных определим данные для конструирования формы (рис.7).



Рис.7. Подсхема данных для конструирования формы на базе таблицы ЗАКАЗ

Источник записей для основной части формы ЗАКАЗ – поля таблицы ЗАКАЗ. Поля таблицы ПУНКТ_ЗАКАЗА являются источниками записей подчиненной формы. Для поля основной части ПОКУПАТЕЛЬ источником записей будет таблица ПОКУПАТЕЛЬ. Для поля ФОРМА_ОПЛАТЫ источником записей – поля таблицы ФОРМА_ОПЛАТЫ.

Реализацию списков записей для полей **ПОКУПАТЕЛЬ** и **ФОРМА_ОПЛАТЫ** в основной части формы удобно выполнить с помощью элементов управления **ПОЛЕ СО СПИСКОМ**.

Реализацию полей подчиненной формы удобно выполнить с помощью вложенной формы. Следовательно, необходимо предварительно разработать подчиненную форму **ПУНКТ_ЗАКАЗА**.

2. В режиме **КОНСТРУКТОРА** создайте форму для таблицы **ПУНКТ_ЗАКАЗА** (рис.8).

3. Для поля **ТОВАР** источником записей является таблица **ТОВАР**. Реализацию удобно выполнить с помощью элемента **ПОЛЕ СО СПИСКОМ** (рис.8).



Рис.8. Форма **ПУНКТ_ЗАКАЗА** в режиме **КОНСТРУКТОРА**

4. Вносить записи с помощью созданной формы неудобно, так как нужно помнить номер заказа. Эта форма должна быть вложенной формой для формы **Заказ**. Форму **ПунктЗаказа** можно просмотреть в режиме **ВИД** (рис.9).

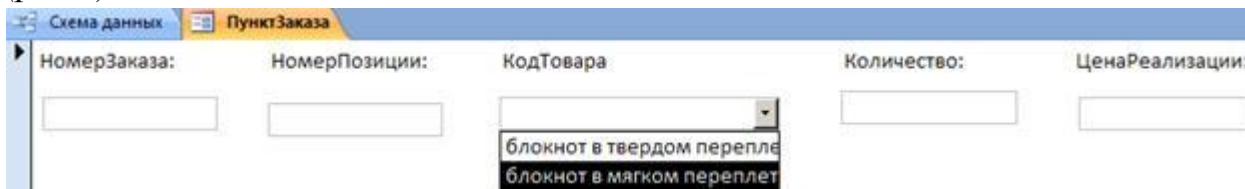


Рис.9. Окончательный вид подчиненной формы **ПУНКТ_ЗАКАЗА**

Прежде, чем начать создавать основную часть формы **ЗАКАЗ**, внесем некоторые изменения в нашу базу данных. Иногда используемые элементы интерфейса и способы реализации полей таблиц в форме **Access** позволяют сократить количество таблиц в схеме базы данных.

В таблице **ЗАКАЗ** поле **ФОРМА_ОПЛАТЫ** имеет всего два значения: наличная оплата и безналичная оплата. В **КОНСТРУКТОРЕ** форм на панели инструментов есть элемент **ПЕРЕКЛЮЧАТЕЛЬ**, имеющий два состояния: **1/0**, **ДА/НЕТ**, **ВКЛ/ВЫКЛ**. Использование элемента **ПЕРЕКЛЮЧАТЕЛЬ** для реализации поля **ФОРМА_ОПЛАТЫ** позволит:

— Хранить значения поля **ФОРМА_ОПЛАТЫ** в числовом формате, а не в текстовом формате (например, **1** – наличная, **0** – безналичная), что является экономией памяти.

— Удалить из схемы таблицу ФОРМА_ОПЛАТЫ (комментарии к значениям поля можно вынести непосредственно в форму).

— Реализовать понятный и привлекательный интерфейс формы.

5. Удалите из базы данных таблицу ФОРМА_ОПЛАТЫ.

6. В режиме КОНСТРУКТОРА создайте форму для таблицы ЗАКАЗ (рис.13).

7. Перенесите поля НОМЕР_ЗАКАЗА, ДАТА_ЗАКАЗА, ОПЛАЧЕНО в область данных формы из списка полей.

8. С помощью свойств поля НОМЕР_ЗАКАЗА это поле можно сделать недоступным для ввода (поле типа счетчик), как показано на рис.15.

9. Сконструируйте поле ПОКУПАТЕЛЬ с помощью элемента ПОЛЕ СО СПИСКОМ (источник записей – таблица ПОКУПАТЕЛЬ).

10. Для создания поля ФОРМА_ОПЛАТЫ щелкните мышью по элементу ГРУППА ПЕРЕКЛЮЧАТЕЛЕЙ на панели инструментов КОНСТРУКТОРА и затем покажите в форме место для помещения этого элемента.

11. Откроется первое диалоговое окно МАСТЕРА создания переключателей. Введите подписи для переключателей (рис.10).

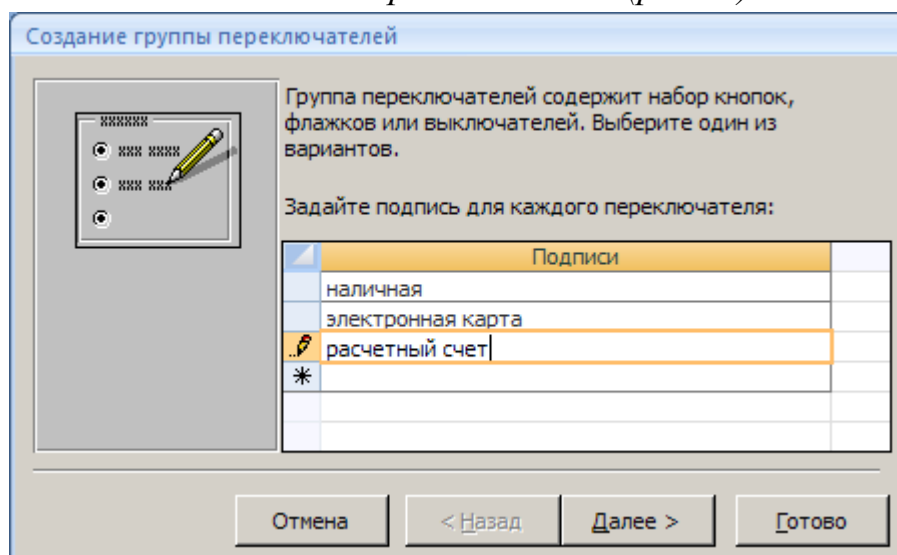


Рис.10. Мастер создания элементов ПЕРЕКЛЮЧАТЕЛИ

12. В следующем диалоге можно выбрать один из переключателей в качестве переключателя, выбранного по умолчанию.

13. В следующем окне диалога вам будет предложено установить числовые значения, которые будут сохранять в поле ФОРМА_ОПЛАТЫ в таблице ЗАКАЗ. Предположим, что наличная форма оплаты – это число 1, безналичная – число 0 (рис.11).

14. Далее надо указать в каком поле будут сохранены значения 1 или 0 (рис.12).

Создание группы переключателей

Задать переключатель, используемый по умолчанию?

Да, выбор по умолчанию:

Нет

Отмена < Назад Далее > Готово

Рис.11. Создание переключателей в форме

Создание группы переключателей

Имеется возможность сохранить выбранное значение в поле или использовать его позднее для выполнения определенной задачи, например, для печати отчета.

Как предполагается использовать выбранное значение?

Сохранить значение для дальнейшего применения

Сохранить значение в поле:

Отмена < Назад Далее > Готово

Рис.12. Сохранение значений переключателей

15. Следующий шаг определит дизайн вашей группы переключателей.

16. Последний шаг диалога. Введите название вашей группы переключателей, например – форма оплаты.

Заголовок формы


Заказ

Область данных

ДатаЗаказа:	КодПокупателя	ФормаОплаты	Оплачено
ДатаЗаказа	КодПокупателя	<input type="checkbox"/> наличная <input type="checkbox"/> электронная карта <input type="checkbox"/> расчетный счет	<input checked="" type="checkbox"/>

Примечание формы

Рис.13. Форма ЗАКАЗ в режиме КОНСТРУКТОРА

Добавление подчиненной формы в основную форму может быть с помощью кнопки ПОДЧИНЕННАЯ ФОРМА/ОТЧЕТ на панели элементов режима КОНСТРУКТОР  (рис.14).

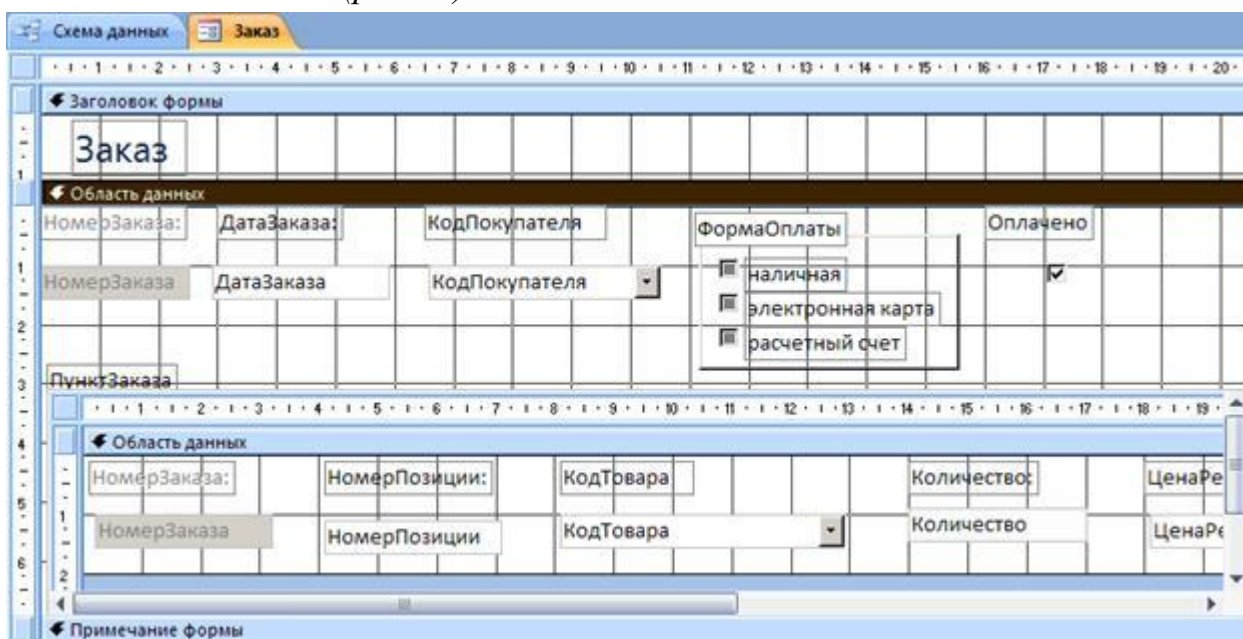


Рис.14. Размещение подчиненной формы в основной форме

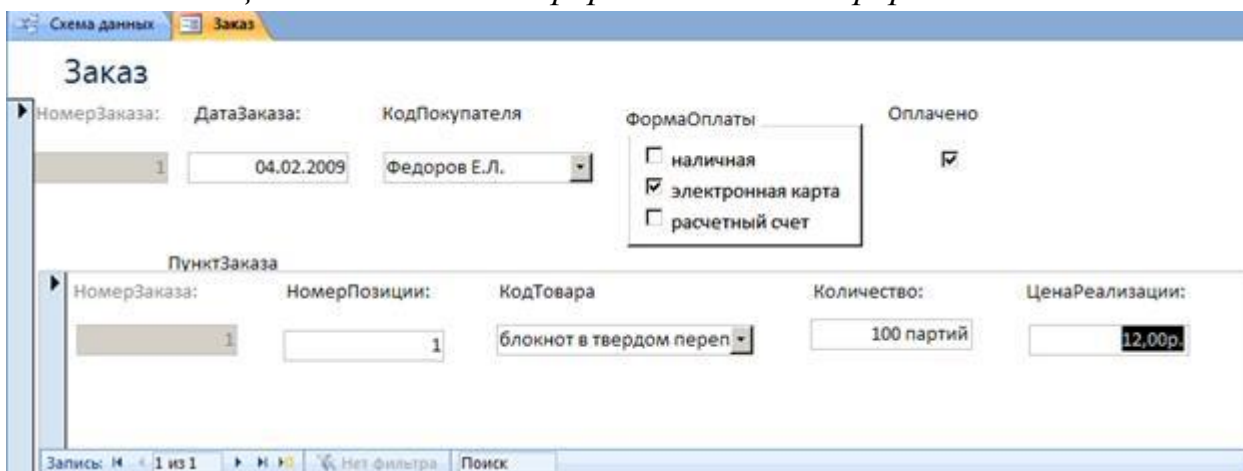


Рис.15. Окончательный вид многотабличной формы ЗАКАЗ


17. Проверьте работоспособность многотабличной формы (рис.15). Введите несколько заказов, каждый из которых состоит из нескольких позиций[2].

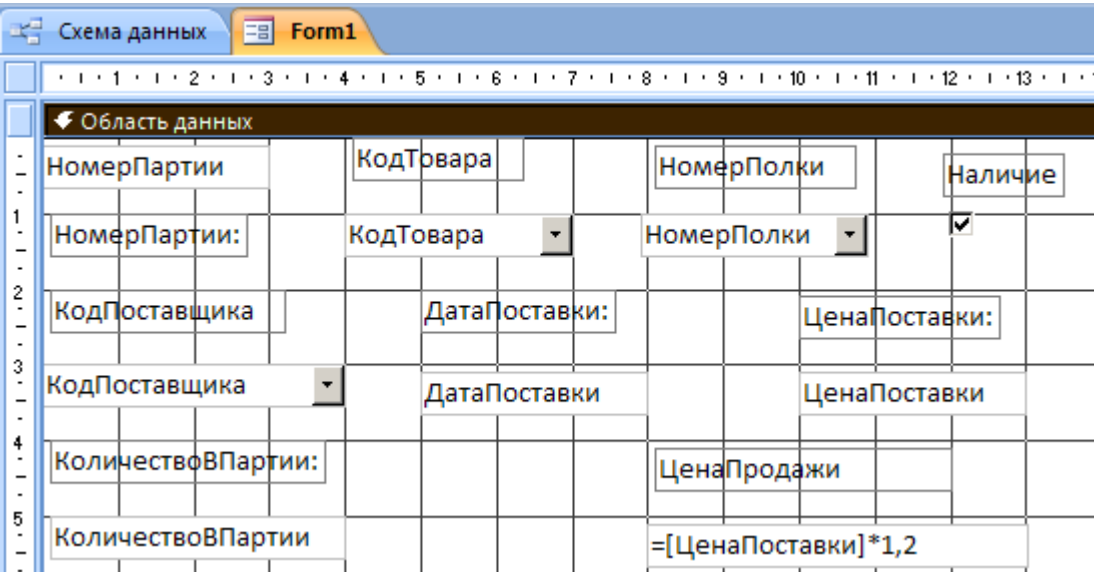
18. Проверьте, что введенные в форму данные помещаются в таблицы ЗАКАЗ и ПУНКТ_ЗАКАЗА.

Самостоятельно

1. Добавьте в форму ЗАКАЗ кнопки для выполнения следующих операций:

- Удаления текущей записи.
- Открытия формы ПОКУПАТЕЛЬ.
- Обновления состояния записей формы.

2. В свойствах поля ЦЕНА_РЕАЛИЗАЦИИ в подчиненной форме установите формат ДЕНЕЖНЫЙ (рис.15).
3. Проверьте возможность добавления нового покупателя непосредственно из формы ЗАКАЗ. Убедитесь, что эти записи попадают в таблицу ПОКУПАТЕЛЬ.
4. Добавьте в форму ТОВАР подчиненную формы, созданную на основе таблицы ПАРТИЯ_ТОВАРА.
5. В форме ПАРТИЯ_ТОВАРА создайте вычисляемое поле ЦЕНА_ПРОДАЖИ (рис. 16) с помощью ПОСТРОИТЕЛЯ ВЫРАЖЕНИЙ (кнопка  на панели инструментов КОНСТРУКТОРА формы) и элемента управления ПОЛЕ из панели элементов **abl**.



The screenshot shows the Microsoft Access Form Designer interface. At the top, there are tabs for 'Схема данных' and 'Form1'. Below the tabs is a horizontal ruler with grid lines numbered 1 through 14. The main area is titled 'Область данных' and contains a data table with the following fields and values:

НомерПартии	КодТовара	НомерПолки	Наличие
НомерПартии:	КодТовара	НомерПолки	<input checked="" type="checkbox"/>
КодПоставщика	ДатаПоставки:	ЦенаПоставки:	
КодПоставщика	ДатаПоставки	ЦенаПоставки	
КоличествоВПартии:		ЦенаПродажи	
КоличествоВПартии		=[ЦенаПоставки]*1,2	

Рис.16. Создание вычисляемого поля в форме

На рис.17 показан примерный вид формы для таблицы ТОВАР в режиме КОНСТРУКТОРА формы.

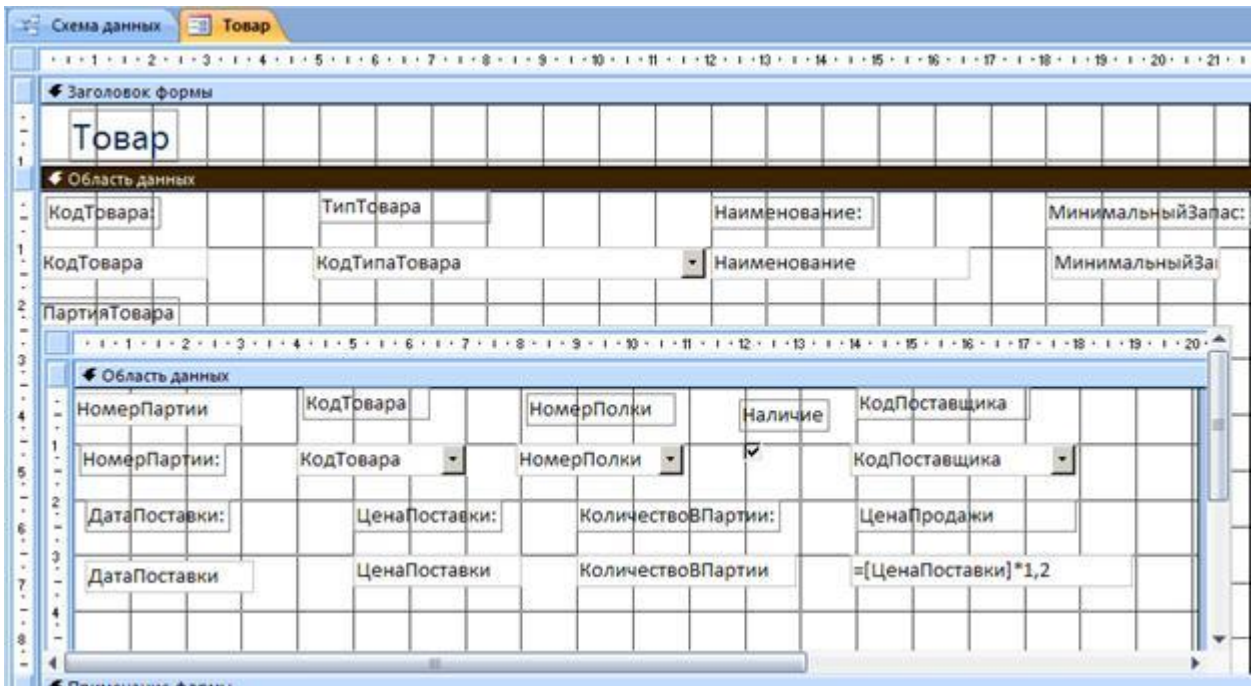


Рис.17. Вид формы ТОВАР в режиме КОНСТРУКТОРА

6. Заполните все таблицы вашей базы данных с помощью созданных форм.
7. Добавьте при необходимости кнопки для удаления записей, кнопки для открытия форм-справочников из многотабличных форм.