
МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ТЕХНОЛОГИЧЕСКИЙ ИНСТИТУТ
(ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ)»

Кафедра ресурсосберегающих технологий

Осипенко У.Ю.

Информационные технологии

Учебное пособие
для студентов заочной формы обучения

Санкт-Петербург
2011

ВВЕДЕНИЕ

Задания к контрольным работам разбиты на варианты исходя из последней цифры номера зачетки.

Последняя цифра зачетки	Номер варианта
0	1
1	2
2	3
3	4
4	5
5	6
6	7
7	8
8	9
9	10

Отчет к контрольной работе должен быть представлен преподавателю распечатанным на принтере, а его копия вместе с файлом базы данных MS Access для второй и третьей контрольных работ в электронном виде (на любом носителе информации: дискете; CD-диске; и пр.).

На титульном листе указывается фамилия, имя, отчество студента, номер учебной группы, номер контрольной работы, номер варианта и ставится личная подпись студента.

Контрольная работа №1

Подготовьте реферат на тему соответствующую варианту из списка ниже.

Вариант 1. Состояние и перспективы развития информационных технологий в химической промышленности.

Вариант 2. Состояние и тенденции развития информационных технологий в вашей профессиональной деятельности.

Вариант 3. Технологии тестирования степени профессиональной пригодности абитуриентов.

Вариант 4. Этапы и тенденции развития информационных технологий в системах управления технологическим процессом.

Вариант 5. Динамика использования и развития автоматизированных обучающих систем.

Вариант 6. Интернет и становление систем дистанционного обучения.

Вариант 7. Использование информационных технологий в административном управлении.

Вариант 8. Использование информационных технологий в образовании и подготовке специалистов.

Вариант 9. Состояние и динамика использования систем дистанционного обучения в ВУЗах.

Вариант 10. Динамика развития мирового рынка информации и информационных услуг.

Контрольная работа №2

Разработка информационно-логической модели реляционной базы данных.

Большинство современных баз данных строится на основе реляционных (от английского relation – математическое название таблицы) систем, т.е. таких систем, для которых выполняются два условия:

1. Данные воспринимаются пользователем как таблицы. Здесь речь идет о логическом представлении данных, а не физическом.

2. В распоряжении пользователя имеются три оператора, которые генерируют «новые» таблицы из «старых». Эти операторы: select (выбор), project (проекция) и join (объединение).

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 виртуальную таблицу. Для задания связи таблицы должны иметь поля с одинаковыми именами или хотя бы с одинаковыми форматами данных. Связь между таблицами устанавливает отношения между совпадающими значениями в этих полях. Такая организация данных позволяет уменьшить избыточность хранимых данных, упрощает их ввод и организацию запросов и отчетов. В Access можно задать три вида связей между таблицами:

- один ко многим;
- многие ко многим;
- один к одному.

Связь «Один ко многим» - наиболее часто используемый тип связи между таблицами. В такой связи каждой записи в таблице «А» может соответствовать несколько записей в таблице «В» (поля с этими записями называют «внешними ключами»), а запись в таблице «В» не может иметь более одной соответствующей ей записи в таблице «А».

При связи «Многие ко многим» одной записи в таблице «А» может соответствовать несколько записей в таблице «В», а одной записи в таблице «В» - несколько записей в таблице «А». Такая схема реализуется только с помощью третьей (связующей) таблицы, ключ которой состоит по крайней мере из двух полей, одно из которых является общим с таблицей «А», а другое - общим с таблицей «В».

При связи «Один к одному» запись в таблице «А» может иметь не более одной связанной записи в таблице «В» и наоборот. Этот тип связи используют не очень часто, поскольку такие данные могут быть помещены в одну таблицу. Связь с отношением «Один к одному» применяют для разделения очень широких таблиц, для отделения части таблицы в целях ее защиты, а также для сохранения сведений, относящихся к подмножеству записей в главной таблице.

Тип создаваемой связи зависит от полей, для которых определяется связь:

- связь «Один ко многим» создается в том случае, когда только одно из полей является ключевым или имеет уникальный индекс, т. е. значения в нем не повторяются;
- связь «Один к одному» создается в том случае, когда оба связываемых поля являются ключевыми или имеют уникальные индексы;

- связь «Многие ко многим» фактически представляет две связи типа один ко многим через третью таблицу, ключ которой состоит, по крайней мере, из двух полей, общих для двух других таблиц.

Целостность данных означает систему правил, используемых в СУБД Access для поддержания связей между записями в связанных таблицах (таблиц, объединенных с помощью связи), а также обеспечивает защиту от случайного удаления или изменения связанных данных. Контролировать целостность данных можно, если выполнены следующие условия:

- связанное поле (поле, посредством которого осуществляется связь) одной таблицы является ключевым полем или имеет уникальный индекс;

- связанные поля имеют один тип данных. Здесь существует исключение. Поле счетчика может быть связано с числовым полем, если оно имеет тип «Длинное целое»;

- обе таблицы принадлежат одной базе данных Access. Если таблицы являются связанными, то они должны быть таблицами Access. Для установки целостности данных база данных, в которой находятся таблицы, должна быть открыта. Для связанных таблиц из баз данных других форматов установить целостность данных невозможно.

Первичный ключ – это один или несколько столбцов, идентифицирующих строку отношения. Первичный ключ должен быть уникальным. Например, для отношения в таблице «Студент» столбец (атрибут) «НомерЗачетки» является уникальным ключом (unique key), поскольку имеет место взаимно однозначное соотношение между значением атрибута «НомерЗачетки» и строкой отношения «Студент». Таким образом, запрос, предписывающий найти все строки в отношении «Студент», в которых значение атрибута «НомерЗачетки» равно 04343, выдаст единственную строку.

Ключ, построенный на основании столбца «Группа» является неуникальным.

Из приведенных данных таблицы 1 в методических указаниях может создаться впечатление, что атрибуты «НомерЗачетки», «Фамилия» и «E_mail» являются уникальными идентификаторами. Но для того, чтобы решить, так ли это на самом деле, одних приведенных здесь данных недостаточно. О том, является ли тот или иной столбец уникальным, разработчики должны спросить у пользователей или у других специалистов в данной области. Хороший пример – столбец «Фамилия». Скорее всего, это случайность, что все фамилии в наших данных оказались разными. Если спросить пользователей, они скажут, что значения в столбце «Фамилия» в отношении «Студент» не всегда является уникальным.

Таблица 1 – «Студент»

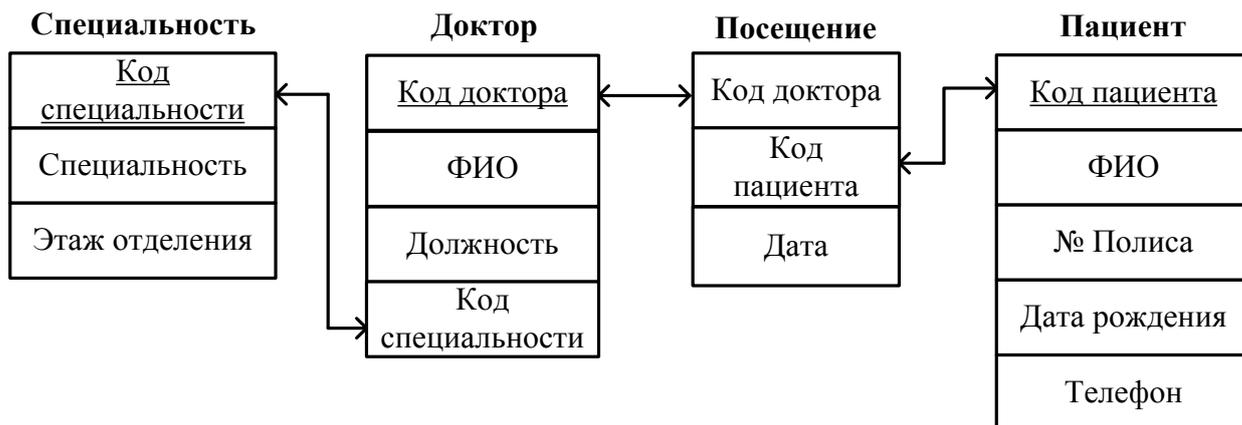
НомерЗачетки	Фамилия	Имя	Группа	E_mail	Телефон
04413	Иванов	Иван	112	ii@somewhere.com	236-9956
04343	Лопухов	Дмитрий	876	lopux@swhere.com	755-7755
04786	Кобылова	Мария	458	korova@milkyway.muu	777-7777
03652	Гопников	Сидор	267	yaya@gopnikcity.da	345-7645
07768	Дылдова	Зинаида	468	shpala@RusRailways.ru	487-9857
03478	Быдлов	Будулай	746	myaso@freshmeat.ru	543-3456
03872	Пузанова	София	112	puzzo@liposuction.net	567-3465
07653	Сатанов	Люцифер	666	boss@HotVacation.com	666-6669

В реляционной модели СУБД все связи представляются путем помещения первичного ключа одной таблицы в другую таблицу. Такой ключ называется внешним или вторичным ключом (foreign key).

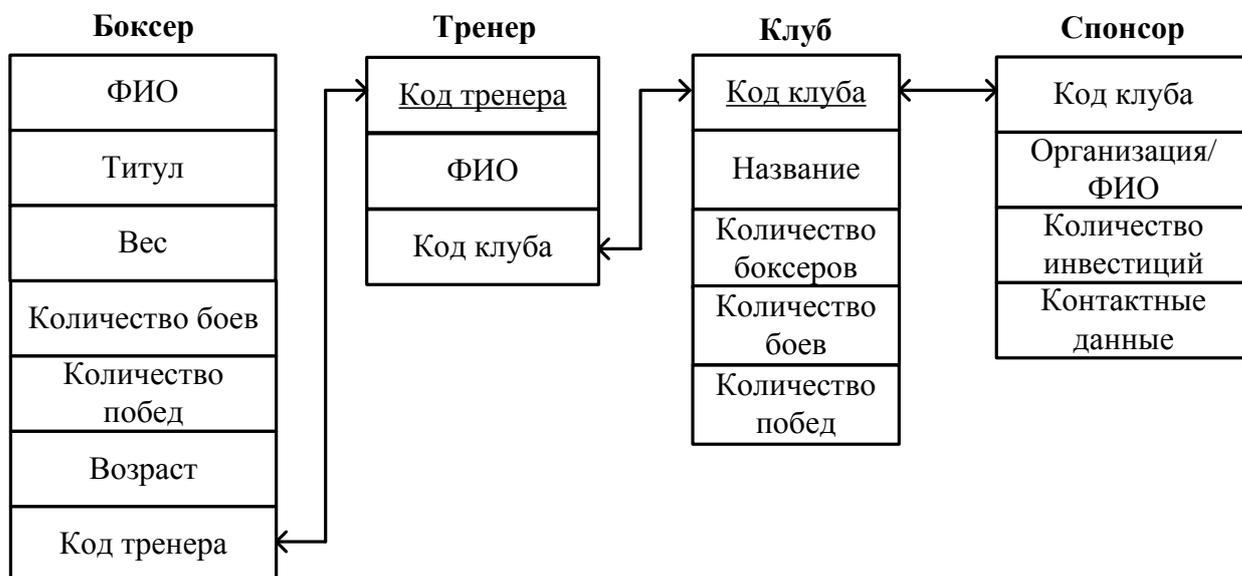
Задание

Исходя из представленной логической схемы, разработать с помощью средств MS Access базу данных и заполнить таблицы не менее чем 6-ю строками данных. В отчете отобразить все этапы создания, заполнения и связывания Таблиц.

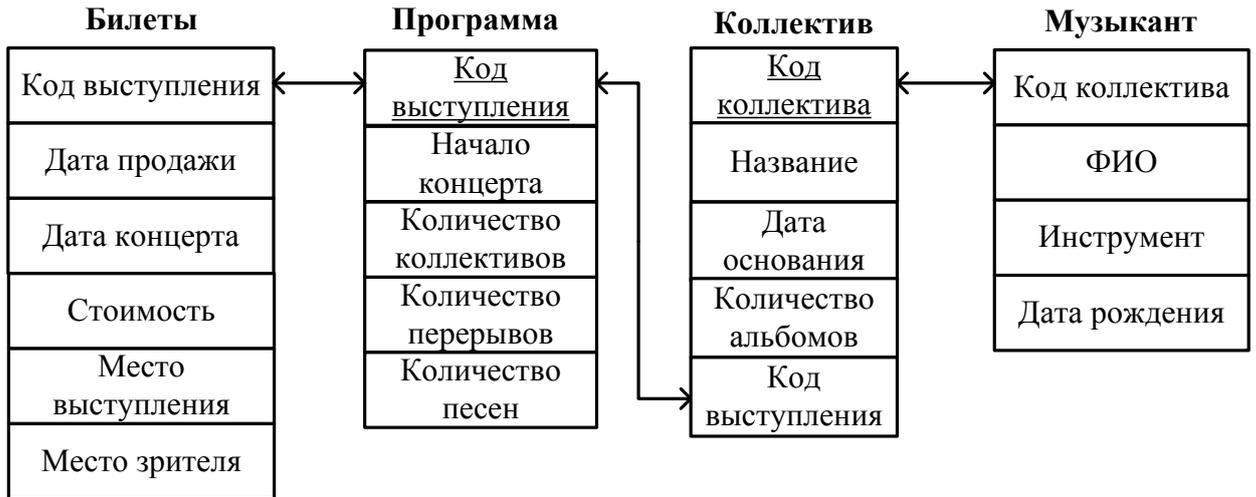
Вариант 1 – Логическая модель базы данных «Поликлиника».



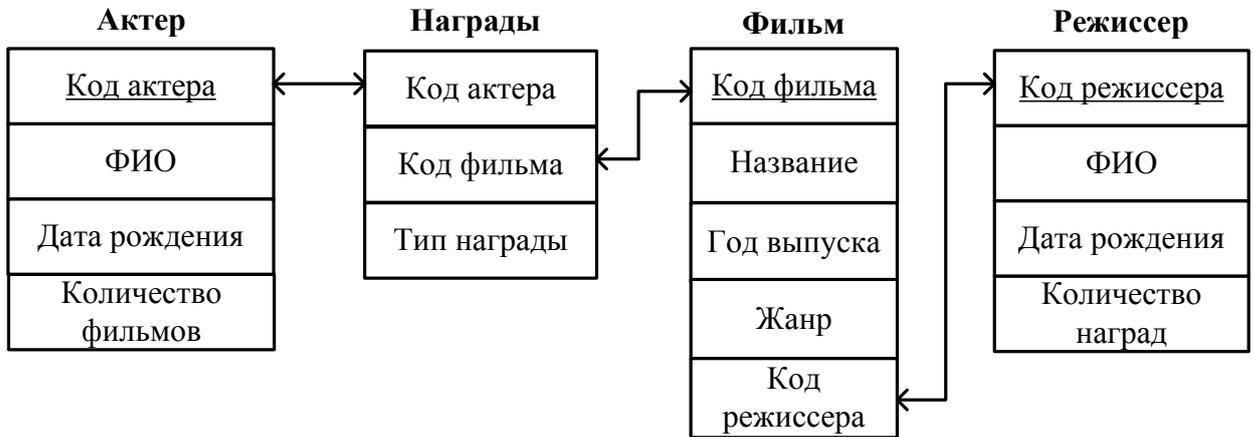
Вариант 2 – Логическая модель базы данных «Бокс».



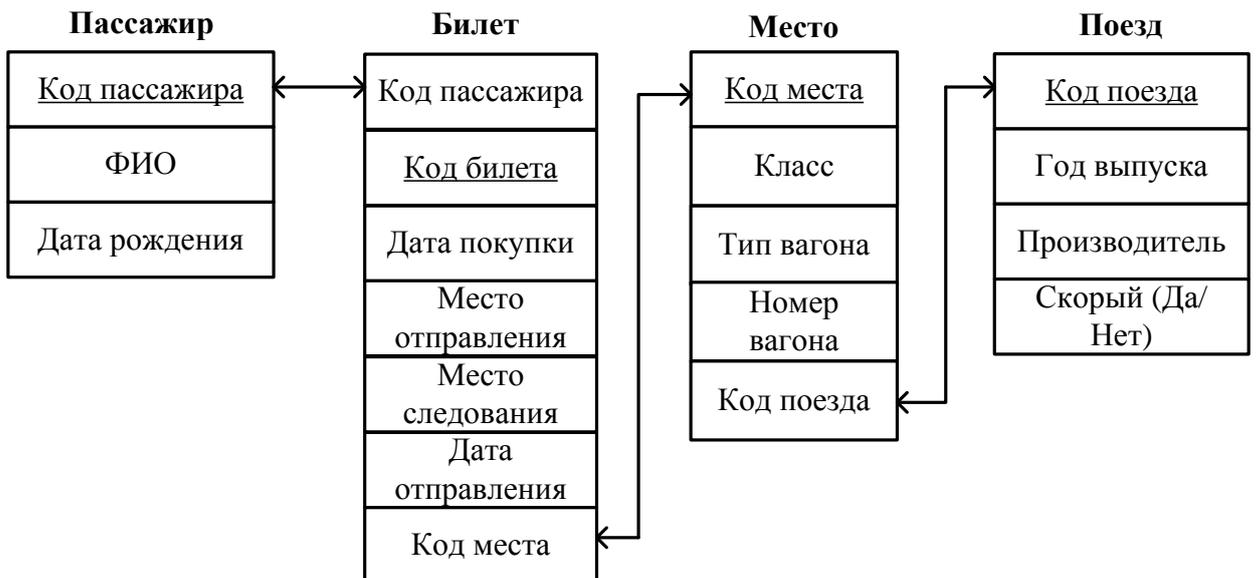
Вариант 3 – Логическая модель базы данных «Концерт».



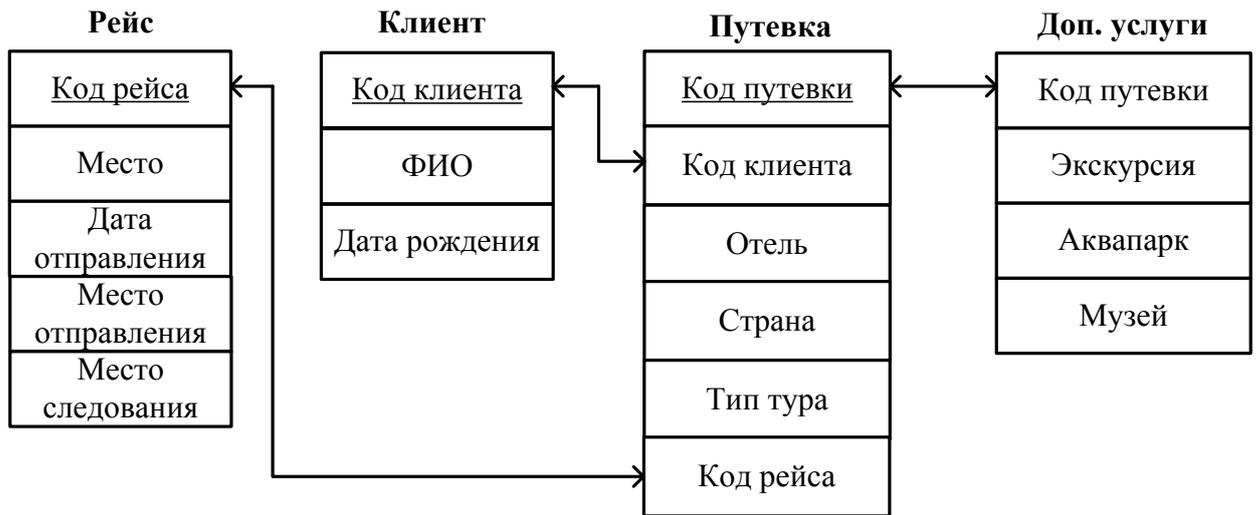
Вариант 4 – Логическая модель базы данных «Кинофильм».



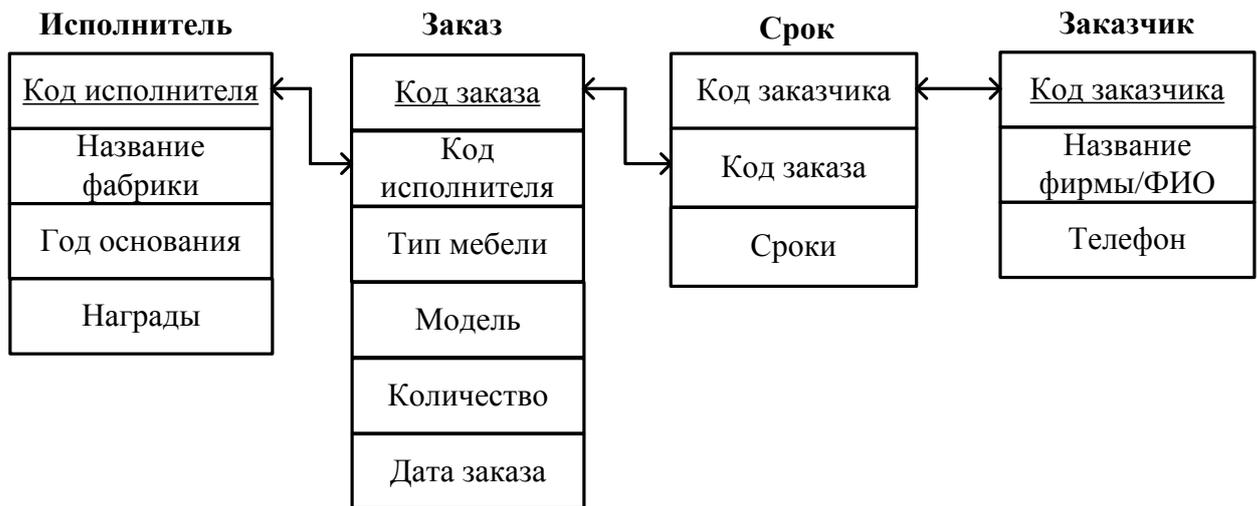
Вариант 5 – Логическая модель базы данных «Поездка».



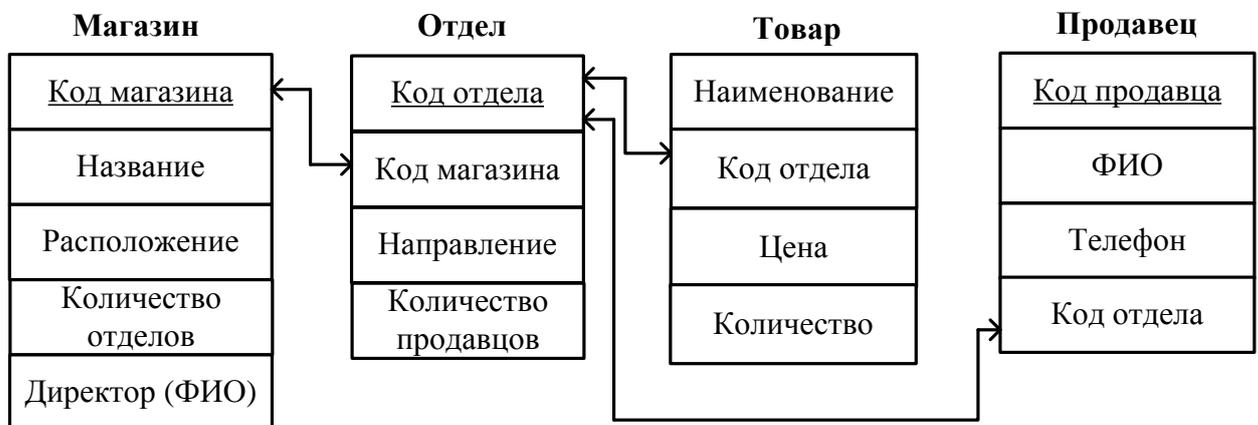
Вариант 6 – Логическая модель базы данных «Путевка».



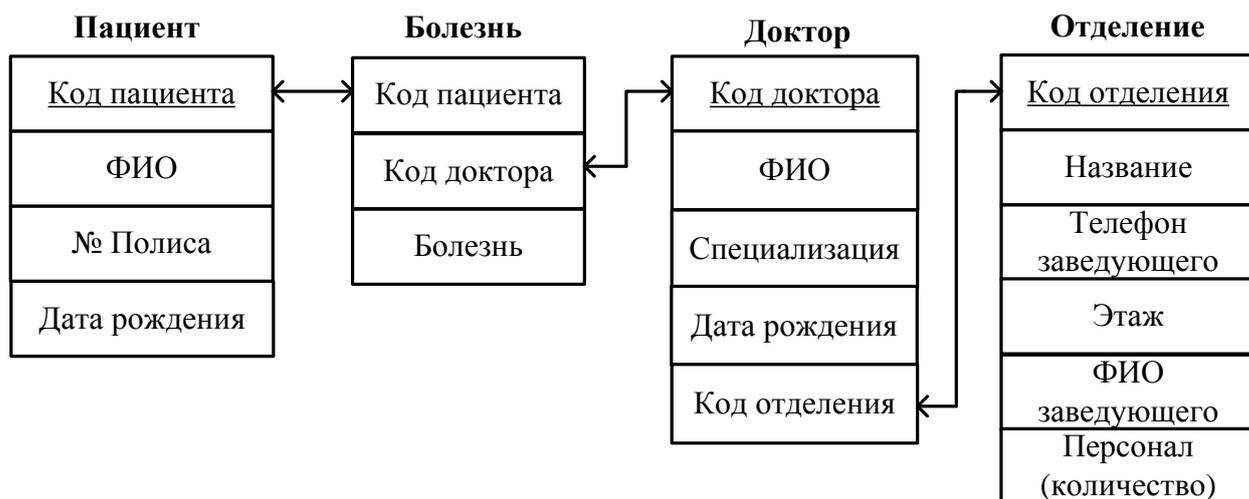
Вариант 7 – Логическая модель базы данных «Заказ мебели».



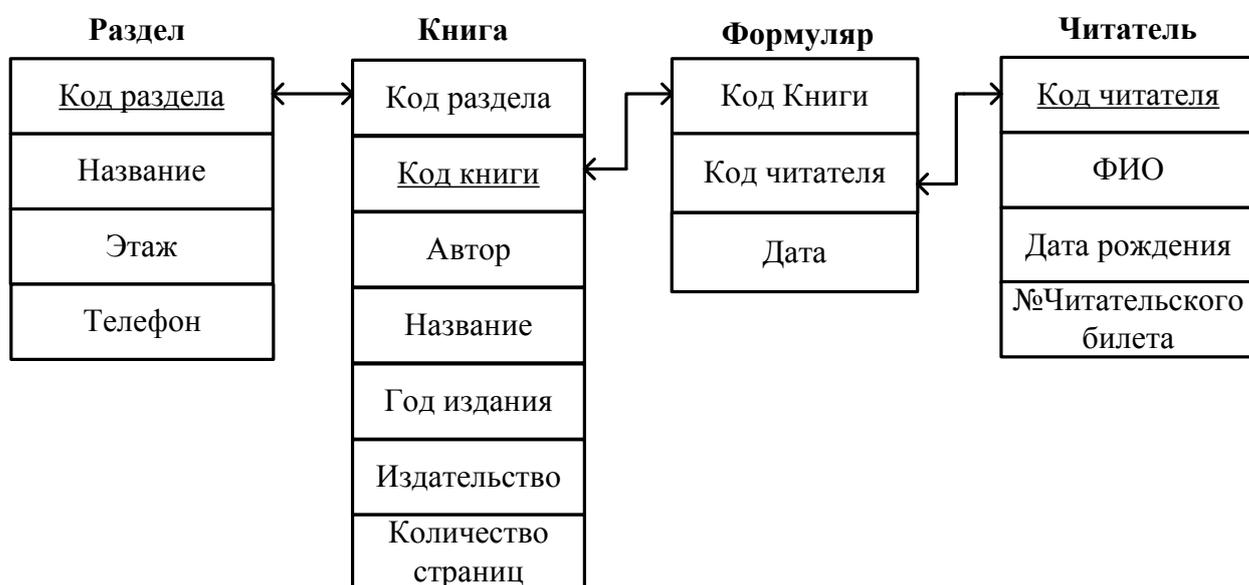
Вариант 8 – Логическая модель базы данных «Магазин».



Вариант 9 – Логическая модель базы данных «Больница».



Вариант 10 – Логическая модель базы данных «Библиотека».



Технология работы.

1. Для создания новой базы данных:

- загрузить Access, и создать новую базу данных;
- задайте соответствующее варианту имя базы данных и выберите место сохранения.

2. Для создания таблицы базы данных:

- в окне базы данных выбрать вкладку «Таблицы», а затем щелкнуть по кнопке «Создать» (для MS Office 2007 – вкладка «Создание», кнопка «Таблица»);
- в окне «Новая таблица» выбрать пункт «Конструктор» и щелкнуть по кнопке «ОК» (для MS Office 2007 – щелкнуть правой кнопкой мыши по созданной таблице в левом меню и выбрать пункт «Конструктор»). В результате проделанных операций открывается окно таблицы в режиме конструктора, в котором следует определить поля таблицы.

3. Для определения полей таблицы:

- ввести строку столбца «Имя поля» имя первого поля (например для таблицы «Раздел» ввести имя первого столбца «Код раздела» для 10 варианта);
- в строке столбца «Тип данных» щелкнуть по кнопке списка и выбрать тип данных «Счетчик» для первичного ключа и «Текст», «Дата» или «Числовое» для остальных в зависимости от смысла.
- Выбрать для таблицы ключевое поле, если это необходимо. Для этого нужно щелкнуть правой кнопкой мыши по нужному полю и выбрать пункт «Ключевое поле»;
- В режиме «Конструктора» можно также удалить лишние поля.

4. Для сохранения таблицы:

- выбрать пункт меню «Файл, Сохранить» (для MS Office 2003 – таблица сохраняется также при ее закрытии, для MS Office 2007 – таблица сохраняется при входе в режим «Конструктора»);
- в диалоговом окне «Сохранение» ввести имя таблицы;
- щелкнуть по кнопке «ОК».

5. Для того чтобы организовать связь между таблицами, надо создать во второй таблице столбец, соответствующий первичному ключу в первой таблице. Это будет вторичный ключ.

Например, нужно связать таблицу «Студент» и таблицу «Группа» (рис. 2), при чем нужно учесть, что для одной группы должно быть много студентов, а не для одного студента много групп.

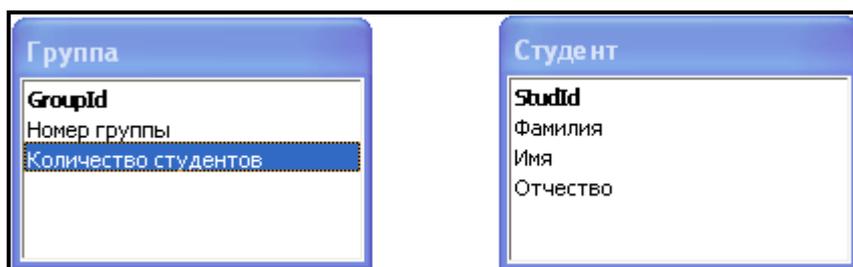


Рисунок 2 – Несвязанные таблицы

Поэтому создадим в таблице «Студент» новое поле с названием «GroupId» и типом данных Integer. Это поле и будет вторичным ключом. На основании введенного атрибута можно организовать связь между рассматриваемыми таблицами. Технология организации связей имеет свои особенности для различных СУБД.

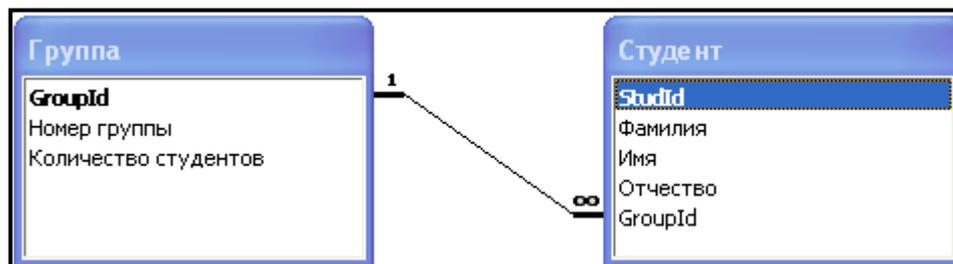


Рисунок 4 – Связанные таблицы «один-ко-многим»

Теперь таблица «Группа» связана с таблицей «Студент» связью «один-ко-многим» (т.е. в одной группе может быть любое количество студентов, а каждый студент, может обучаться только в одной группе).

Для того, чтобы организовать такую связь, необходимо:

- Сначала необходимо на панели меню найти кнопку со всплывающим названием «Схема данных» (для MS Office 2003 находится справа верхней панели меню, для MS Office 2007 – находится на вкладке «Работа с базами данных»);
- Затем нужно добавить на схему все таблицы.
- Связь между таблицами осуществляется простым перетаскиванием первичного ключа из первой таблицы во внешний ключ второй таблицы. В открывшемся окне необходимо обязательно проставить галочку «Обеспечить целостность данных».
- Если при попытке связать таблицы появляется сообщение об ошибке, следует проверить тип данных внешнего ключа (должно быть «Числовой», если первичный ключ «Счетчик» или «Числовой»). Во избежание ошибок лучше заполнять таблицы после их связывания и стоит учесть, что поля внешнего ключа не должны содержать данных, которых не содержат поля первично ключа (например, если первичный ключ содержит значения 1, 3, 5, то в полях вторичного ключа могут многократно повторяться числа 1, 3 и 5, но не должно быть чисел 0, 2, 4 и других).

Контрольная работа №3

Формирование запросов

Запросы являются мощным средством обработки данных, хранимых в таблицах Access . С помощью запросов можно просматривать, анализировать и изменять данные из нескольких таблиц. Они также используются в качестве источника данных для форм и отчетов. Запросы позволяют вычислять итоговые значения и выводить их в компактном формате, подобном формату электронной таблицы, а также выполнять вычисления над группами записей.

Запросы можно создавать самостоятельно и с помощью «мастеров». «Мастера запросов» автоматически выполняют основные действия в зависимости от ответов пользователя на поставленные вопросы. Самостоятельно разработать запросы можно в режиме «конструктора».

В Access можно создавать следующие типы запросов:

- запрос на выборку;
- запрос с параметрами;
- перекрестный запрос;
- запрос на изменение (запрос на удаление, обновление и добавление записей, на создание таблицы);
- запросы SQL (запросы на объединение, запросы к серверу, управляющие запросы, подчиненные запросы).

Запрос на выборку используется наиболее часто. При его выполнении данные, удовлетворяющие условиям отбора, выбираются из одной или из нескольких таблиц и выводятся в определенном порядке.

Можно также использовать запрос на выборку, чтобы сгруппировать записи для вычисления сумм, средних значений, пересчета и других действий.

Запрос с параметрами – это запрос, при выполнении которого в его диалоговом окне пользователю выдается приглашение ввести данные, на основе которых будет выполняться запрос. Например, часто требуются данные о том, какие дисциплины ведут преподаватели. Чтобы не создавать отдельные запросы по каждому преподавателю, можно создать один запрос с параметрами, где в качестве параметра будет использоваться фамилия преподавателя. При каждом вызове этого запроса предлагается ввести фамилию преподавателя, а затем на экран будут выведены все поля, которые вы указали в запросе, например фамилия, имя, отчество преподавателя и читаемая им дисциплина.

Язык SQL (Structured Query Language - структурированный язык запросов) ориентирован на операции с данными, представленными в виде логически взаимосвязанных совокупностей таблиц. Особенность предложений этого языка состоит в том, что они ориентированы в большей степени на конечный результат обработки данных, чем на процедуру этой обработки. SQL сам определяет, где находятся данные, какие индексы и даже наиболее эффективные последовательности операций следует использовать для их получения: не надо указывать эти детали в запросе к базе данных.

Все операторы языка запроса делятся:

- команда определения данных
- команды манипулирования данными

SELECT - (выбрать) данные из указанных столбцов и (если необходимо) выполнить перед выводом их преобразование в соответствии с указанными выражениями и (или) функциями

FROM - (из) перечисленных таблиц, в которых расположены эти столбцы

WHERE - (где) строки из указанных таблиц должны удовлетворять указанному перечню условий отбора строк

GROUP BY- (группируя по) указанному перечню столбцов с тем, чтобы получить для каждой группы единственное агрегированное значение, используя во фразе SELECT SQL-функции SUM (сумма), COUNT (количество), MIN (минимальное значение), MAX (максимальное значение) или AVG (среднее значение)

HAVING - (имея) в результате лишь те группы, которые удовлетворяют указанному перечню условий отбора групп

Задание

Используя созданную базу данных в MS Access создать и выполнить запрос исходя из варианта. В отчете отобразить все этапы создания запросов и результаты их выполнения.

Вариант 1

На основе таблиц созданной базы данных создать простые запросы на выборку.

1. Запрос «Доктор»

Через «Мастер запросов» создать простой запрос на выборку из таблицы «Доктор», в котором должны отображаться поля «ФИО», «Должность».

2. Запрос «Доктора по отделениям»

Через «Конструктор запросов» создать простой запрос на выборку из таблиц «Доктор» и «Специальность», в котором должны отображаться поля «ФИО», «Специальность» и «Этаж отделения».

3. Запрос «Пациент»

Через «Режим SQL» создать простой SQL запрос на выборку из таблицы «Пациент», в котором должны отображаться поля «ФИО», «Телефон».

4. Запрос «Посещение»

Через «Режим SQL» используя лекции создать простой SQL запрос на выборку из таблиц «Пациент» и «Посещение», в котором должны отображаться поля «ФИО», «№ Полиса», «Дата» посещения.

Вариант 2

На основе таблиц созданной базы данных создать простые запросы на выборку.

1. Запрос «Боксер»

Через «Мастер запросов» создать простой запрос на выборку из таблицы «Боксер», в котором должны отображаться поля «ФИО», «Титул» и «Количество побед».

2. Запрос «Боксер и тренер»

Через «Конструктор запросов» создать простой запрос на выборку из таблиц «Боксер» и «Тренер», в котором должны отображаться поля «ФИО», «Титул» и «Количество побед» и «ФИО» тренера.

3. Запрос «Клубы»

Через «Режим SQL» создать простой SQL запрос на выборку из таблицы «Клуб», в котором должны отображаться поля «Название», «Количество побед».

4. Запрос «Тренера клуба»

Через «Режим SQL» используя лекции создать простой SQL запрос на выборку из таблиц «Тренер» и «Клуб», в котором должны отображаться поля «ФИО», «Название», «Количество побед».

Вариант 3

На основе таблиц созданной базы данных создать простые запросы на выборку.

1. Запрос «Программа»

Через «Мастер запросов» создать простой запрос на выборку из таблицы «Программа», в котором должны отображаться поля «Код выступления», «Количество песен».

2. Запрос «Музыканты коллектива»

Через «Конструктор запросов» создать простой запрос на выборку из таблиц «Музыкант» и «Коллектив», в котором должны отображаться поля «ФИО», «Инструмент» и «Название коллектива».

3. Запрос «Билет»

Через «Режим SQL» создать простой SQL запрос на выборку из таблицы «Билет», в котором должны отображаться поля «Дата», «Стоимость», «Место выступления».

4. Запрос «Выступление коллектива»

Через «Режим SQL» используя лекции создать простой SQL запрос на выборку из таблиц «Коллектив» и «Программа», в котором должны отображаться поля «Название», «Начало концерта», «Количество песен».

Вариант 4

На основе таблиц созданной базы данных создать простые запросы на выборку.

1. Запрос «Актер»

Через «Мастер запросов» создать простой запрос на выборку из таблицы «Актер», в котором должны отображаться поля «ФИО», «Дата рождения».

2. Запрос «Награды актера»

Через «Конструктор запросов» создать простой запрос на выборку из таблиц «Актер» и «Награды», в котором должны отображаться поля «ФИО», «Тип награды».

3. Запрос «Режисер»

Через «Режим SQL» создать простой SQL запрос на выборку из таблицы «Режисер», в котором должны отображаться поля «ФИО», «Количество наград».

4. Запрос «Фильм»

Через «Режим SQL» используя лекции создать простой SQL запрос на выборку из таблиц «Фильм» и «Режиссер», в котором должны отображаться поля «Название», «Год выпуска», «ФИО» режиссера.

Вариант 5

На основе таблиц созданной базы данных создать простые запросы на выборку.

1. Запрос «Пассажир»

Через «Мастер запросов» создать простой запрос на выборку из таблицы «Пассажир», в котором должны отображаться поля «ФИО», «Код пассажира».

2. Запрос «Пассажир поезда»

Через «Конструктор запросов» создать простой запрос на выборку из таблиц «Пассажир» и «Билет», в котором должны отображаться поля «ФИО», «Дата отправления», «Место следования» и «Код места».

3. Запрос «Поезда»

Через «Режим SQL» создать простой SQL запрос на выборку из таблицы «Поезд», в котором должны отображаться поля «Год выпуска», «Производитель», «Скорый».

4. Запрос «Вагоны поезда»

Через «Режим SQL» используя лекции создать простой SQL запрос на выборку из таблиц «Вагон» и «Поезд», в котором должны отображаться поля «Тип вагона», «Номер вагона», «Код поезда», «Скорый».

Вариант 6

На основе таблиц созданной базы данных создать простые запросы на выборку.

1. Запрос «Клиент»

Через «Мастер запросов» создать простой запрос на выборку из таблицы «Клиент», в котором должны отображаться поля «ФИО», «Код клиента».

2. Запрос «Путевка»

Через «Конструктор запросов» создать простой запрос на выборку из таблиц «Клиент» и «Путевка», в котором должны отображаться поля «ФИО», «Страна», «Отель» и «Тип тура».

3. Запрос «Рейс»

Через «Режим SQL» создать простой SQL запрос на выборку из таблицы «Рейс», в котором должны отображаться поля «Дата отправления», «Место следования», «Место».

4. Запрос «Услуги»

Через «Режим SQL» используя лекции создать простой SQL запрос на выборку из таблиц «Путевка» и «Доп. услуги», в котором должны отображаться поля «Страна», «Отель», «Экскурсия», «Код услуги».

Вариант 7

На основе таблиц созданной базы данных создать простые запросы на выборку.

1. Запрос «Все исполнители»

Через «Мастер запросов» создать простой запрос на выборку из таблицы «Исполнитель», в котором должны отображаться поля «Код исполнителя», «Название фабрики».

2. Запрос «Исполнитель заказа»

Через «Конструктор запросов» создать простой запрос на выборку из таблиц «Заказ» и «Исполнитель», в котором должны отображаться поля «Код заказа», «Тип мебели», «Количество» и «Название фабрики».

3. Запрос «Рейс»

Через «Режим SQL» создать простой SQL запрос на выборку из таблицы «Рейс», в котором должны отображаться поля «Дата отправления», «Место следования», «Место».

4. Запрос «Услуги»

Через «Режим SQL» используя лекции создать простой SQL запрос на выборку из таблиц «Путевка» и «Доп. услуги», в котором должны отображаться поля «Страна», «Отель», «Экскурсия», «Код услуги».

Вариант 8

На основе таблиц созданной базы данных создать простые запросы на выборку.

1. Запрос «Магазин»

Через «Мастер запросов» создать простой запрос на выборку из таблицы «Магазин», в котором должны отображаться поля «Название», «Расположение».

2. Запрос «Отделы магазинов»

Через «Конструктор запросов» создать простой запрос на выборку из таблиц «Отдел» и «Магазин», в котором должны отображаться поля «Код отдела», «Направление», «Название» магазина.

3. Запрос «Продавец»

Через «Режим SQL» создать простой SQL запрос на выборку из таблицы «Продавец», в котором должны отображаться поля «ФИО», «Код отдела».

4. Запрос «Услуги»

Через «Режим SQL» используя лекции создать простой SQL запрос на выборку из таблиц «Путевка» и «Доп. услуги», в котором должны отображаться поля «Страна», «Отель», «Экскурсия», «Код услуги».

Вариант 9

На основе таблиц созданной базы данных создать простые запросы на выборку.

1. Запрос «Пациент»

Через «Мастер запросов» создать простой запрос на выборку из таблицы «Пациент», в котором должны отображаться поля «ФИО», «№ Полиса».

2. Запрос «Болезнь пациента»

Через «Конструктор запросов» создать простой запрос на выборку из таблиц «Пациент» и «Болезнь», в котором должны отображаться поля «ФИО», «№ Полиса», «Болезнь».

3. Запрос «Доктора»

Через «Режим SQL» создать простой SQL запрос на выборку из таблицы «Доктор», в котором должны отображаться поля «ФИО», «Специальность».

4. Запрос «Доктора отделений»

Через «Режим SQL» используя лекции создать простой SQL запрос на выборку из таблиц «Доктор» и «Отделение», в котором должны отображаться поля «ФИО», «Специальность», «Наименование» отделения, «Этаж».

Вариант 10

На основе таблиц созданной базы данных создать простые запросы на выборку.

1. Запрос «Книги»

Через «Мастер запросов» создать простой запрос на выборку из таблицы «Книга», в котором должны отображаться поля «Автор», «Название».

2. Запрос «Формуляр»

Через «Конструктор запросов» создать простой запрос на выборку из таблиц «Формуляр» и «Читатель», в котором должны отображаться поля «Дата», «Код книги», «ФИО», «№ Читальского билета».

3. Запрос «Разделы»

Через «Режим SQL» создать простой SQL запрос на выборку из таблицы «Раздел», в котором должны отображаться поля «Название», «Телефон».

4. Запрос «Книги по разделам»

Через «Режим SQL» используя лекции создать простой SQL запрос на выборку из таблиц «Книга» и «Раздел», в котором должны отображаться поля «Автор», «Название» книги, «Название» раздела, «Этаж».

Технология работы.

Для создания нового запроса надо в окне базы данных выбрать вкладку «Запросы» и щелкнуть по кнопке «Создать». Откроется окно «Новый запрос». В окне следует выбрать один из пяти пунктов:

- конструктор,
- простой запрос,
- перекрестный запрос,
- повторяющиеся записи,
- записи без подчиненных.

Конструктор позволит создать любой тип запроса, но этот режим рекомендуется пользователям, уже имеющим некоторый опыт создания запросов. «Простой запрос» позволит создать с помощью «Мастера» запрос на выборку из определенных полей таблиц или других запросов.

При выполнении запроса на выборку Access извлекает записи из таблиц и формирует результирующий набор данных. Он выглядит, как таблица, хотя и не является ею. Результирующий набор данных является «динамическим» (или виртуальным) набором записей и не хранится в базе данных.

После закрытия запроса результирующий набор данных прекращает свое существование. Хотя сам по себе динамический набор данных больше не существует, данные, которые в нем содержались, остаются в базовых таблицах.

При сохранении запроса остается только структура запроса – перечень таблиц, список полей, порядок сортировки, ограничения на записи, тип запроса и т. д. При сохранении в базе данных запрос по сравнению с результирующим

набором данных имеет ряд преимуществ:

- на физическом носителе информации (обычно это жесткий диск) требуется меньший объем пространства;
- запрос может использовать обновленные версии любых записей, измененных со времени последнего запуска запроса.

При каждом выполнении запрос обращается к базовым таблицам и снова создает результирующий набор данных. Поскольку сам по себе результирующий набор данных не сохраняется, запрос автоматически отображает любые изменения, происшедшие в базовых таблицах с момента последнего запуска этого запроса (даже в реальном времени в многопользовательской среде).

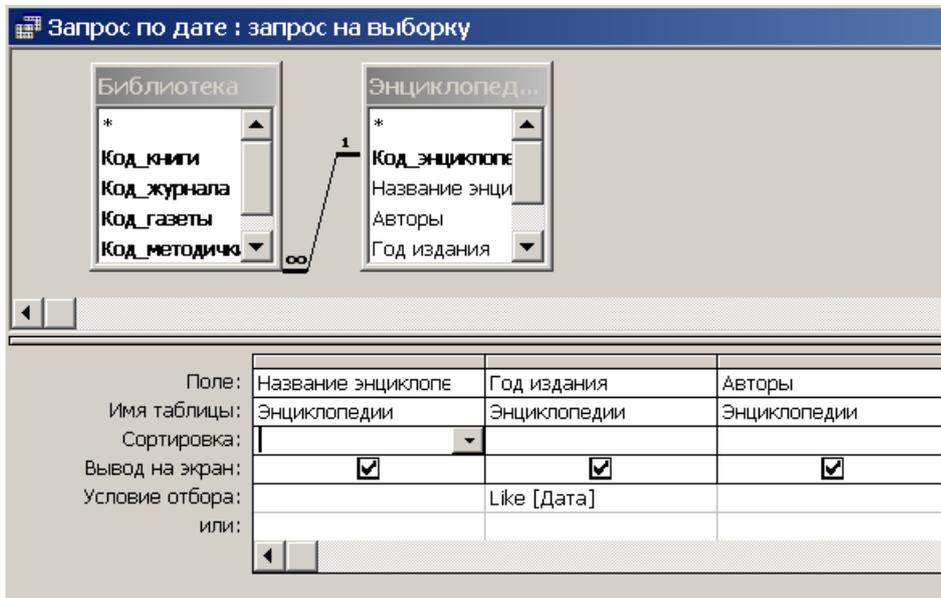
Для сохранения запроса следует выполнить следующие действия.

Выполнить команду «Файл, Сохранить» или щелкнуть по кнопке «Сохранить» на панели инструментов. Если запрос сохраняется впервые, то следует указать новое имя запроса в диалоговом окне «Сохранение».

Чтобы создать запрос в режиме *SQL* нужно создать запрос в режиме конструктора и щелкнув по пустому полю в открывшемся окне правой кнопкой мыши, выбрать «Режим SQL».

Например:

Запрос, который просит ввести дату для того, чтобы по году издания книги узнать ее название и автора,



а читаться он будет следующим образом: Необходимо выдать три поля: Название энциклопедии, год издания и автора из таблицы Энциклопедии с запросом года издания этой книги, до выдачи запроса.

```

Запрос по дате : запрос на выборку
SELECT Энциклопедии.[Название энциклопедии], Энциклопедии.[Год издания], Энциклопедии.Авторы
FROM Энциклопедии INNER JOIN Библиотека ON Энциклопедии.Код_энциклопедии = Библиотека.Код_энциклопедии
WHERE (((Энциклопедии.[Год издания]) Like [Дата]);

```

Другой запрос, который показывает методичку с максимальным количеством страниц,

Запрос по методичкам : перекрестный запрос

Методички

- *
 - Код_методичк
 - Название метс
 - Год издания
 - Составители

Поле:	Код_методички	Составители	Название методичк	Год издания	Кол-во страниц	Итоговое значение
Имя таблицы:	Методички	Методички	Методички	Методички	Методички	Методички
Групповая операция:	Группировка	Группировка	Группировка	Группировка	Max	Max
Перекрестная таблица:	Заголовки строк	Заголовки строк	Заголовки строк	Заголовки столбцов	Значение	Заголовки строк
Сортировка:						
Условие отбора:						
или:						

будет выглядеть следующим образом: Выдать запрос с полями (Составители, название методички, год издания, кол-во страниц и итоговое значение) из т. Методички .

```

Запрос по методичкам : перекрестный запрос
TRANSFORM Max(Методички.[Кол-во страниц]) AS [Max-Кол-во страниц]
SELECT Методички.Код_методички, Методички.Составители, Методички.[Название методички], Max(Методички.[Кол-во страниц]) AS [Итоговое значение Кол-во страниц]
FROM Методички
GROUP BY Методички.Код_методички, Методички.Составители, Методички.[Название методички]
PIVOT Методички.[Год издания];

```

Еще один запрос. который выдает поля (Название книги, год издания, кол-во страниц и автора книги) из т.Книги для книг, которые издавались после 1900 года, читается следующим образом:

Запрос по книгам : запрос на выборку

Библиотека

- * Код_книги
- Код_журнала
- Код_газеты
- Код_методичк

Книги

- * Код_книги
- Название книги
- Автор книги
- Год издания

∞ ————— 1

Поле:	Название книги	Год издания	Кол-во страниц	Автор книги
Имя таблицы:	Книги	Книги	Книги	Книги
Сортировка:				
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:		<1900		
или:				

Выдать запрос по книгам, которые попадают в условие отбора, а которые не попадают не выдавать.

Запрос по книгам : запрос на выборку

```

SELECT Книги.[Название книги], Книги.[Год издания], Книги.[Кол-во страниц], Книги.[Автор книги]
FROM Книги INNER JOIN Библиотека ON Книги.Код_книги = Библиотека.Код_книги
WHERE (((Книги.[Год издания])<1900));

```