

Федеральное государственное бюджетное образовательное учреждение
высшего образования

«Санкт-Петербургский государственный педиатрический медицинский
университет»

Министерства здравоохранения Российской Федерации

УТВЕРЖДЕНО

Учебно-методическим советом

«__»_____2020 г.,

протокол № __

Проректор по учебной работе,
председатель Учебно-методического
совета

профессор

Орел В.И.

П.Е.Котиков, В.В.Решетников, А.А.Тихомирова

АНАЛИЗ ДАННЫХ

Санкт-Петербург

2020

УДК 614.1
ББК51.1

Котиков П.Е., Решетников В.В., Тихомирова А.А. Базы данных. Учебно-методическое пособие. Санкт-Петербург: издание СПбГПМУ, 2019.-56 с.

Учебно-методическое пособие предназначено для студентов первого и второго курсов Санкт-Петербургского государственного педиатрического медицинского университета. Содержание учебно-методического пособия соответствует содержанию учебной программы дисциплины «Медицинская информатика», преподаваемой на первом и втором курсах указанного высшего учебного заведения.

Пособие содержит материалы для практических занятий по освоению современных технологий хранения, обработки медицинских данных и управления данными в практической деятельности.

УТВЕРЖДЕНО УЧЕБНО-МЕТОДИЧЕСКИМ СОВЕТОМ СПбГПМУ

©СПбГПМУ, 2020

Оглавление

ВВЕДЕНИЕ	6
Раздел 1. Теоретические основы баз данных	7
Раздел 2. Практическая работа в СУБД «ACCESS	14
Раздел 3. Задания для самостоятельного выполнения	27
Литература	43
I. Издания, доступные в ЭБС «Консультант студента»:	43
II. Другие издания	44
ПРИЛОЖЕНИЯ	45

ВВЕДЕНИЕ

Пособие содержит материалы для практических занятий по освоению современных технологий баз данных.

Учебно- методическое пособие предназначено для студентов первого и второго курсов Санкт-Петербургского государственного педиатрического медицинского университета. Материал учебно-методического пособия соответствует содержанию рабочей программ дисциплины «Медицинская информатика» для основных специальностей.

В разделе 1 пособия приведен основной теоретический материал, знание которого востребовано практикой работы с базами данных в практической деятельности специалиста-врача.

Задания, представленные в разделах 2 и 3 пособия, предназначены для использования при проведении практических занятий. Выполнение этих заданий предполагает применение пакета программ «Microsoft Office».

Изучение тем, связанных с базами данных завершается выполнением студентами самостоятельной работы по одной из тем (раздел 3 пособия). В приложении 2 показаны на снимках экранов отдельные примеры из студенческих работ. В приложении 1 приведен примерный состав вопросов тестов текущего контроля.

Радел 1. Теоретические основы баз данных

РЕЛЯЦИОННЫЕ БАЗЫ ДАННЫХ И РАБОТА С НИМИ В СРЕДЕ СУБД MS ACCESS РЕЛЯЦИОННЫЕ БАЗЫ ДАННЫХ

Реляционная модель баз данных, базирующаяся на математической *теории отношений*, была предложена американским ученым Эдгаром Коддом в начале 70-х гг. XX века.

1.1. СТРУКТУРА РЕЛЯЦИОННЫХ ДАННЫХ

Важнейшие понятия, используемые при описании реляционных баз данных, - *домен, кортеж и отношение*.

Эти понятия можно пояснить на конкретном примере представления в реляционной базе данных сущности реального мира «*Пациент*».

Домен представляет собой множество элементов, обладающих некими общими свойствами. Общие свойства элементов домена проявляются в том, что эти элементы в компьютере представляются данными одного типа.

В нашем примере используются домены с названиями *Коды пациентов, Коды прививок, Вид прививки, Дата плановой прививки и Дата фактической прививки*.

Кортеж представляет собой последовательность таких элементов, при которых первый элемент берется (копируется) из первого домена, второй - из второго домена и т. д.

Отношение представляет собой множество однотипных кортежей. Это означает, что все кортежи отношения формируются по одной схеме: их первые элементы берутся (копируются) из первого домена, вторые - из второго домена и т. д. Элементы домена обладают некими общими свойствами. Кроме того, предполагается, что над элементами одного домена определены допустимые для них операции. Например, для числового домена определены арифметические операции, для домена, содержащего строковые элементы, предусмотрена операция *конкатенации* (сцепления) и т. д. Предполагается также, что элементы одного домена могут сравниваться на равенство.

На практике вместо термина *отношение* часто применяют термин *таблица*, вместо *кортеж* - *строка* или *запись*.

Имена столбцов называются *атрибутами*, а индивидуальные значения, появляющиеся в отдельных кортежах, - *значениями атрибутов*.

Атрибуты могут располагаться в любом порядке, но от их переупорядочения смысл отношения не меняется. Атрибутам могут присваиваться произвольные названия. Однако необходимо учитывать, что в одном отношении не может быть атрибутов с одинаковыми названиями.

Кроме того, если атрибуты, а точнее было бы сказать значения атрибутов, двух отношений принадлежат одному домену, то им могут присваиваться как

одинаковые, так и различные имена.

Все случаи, когда атрибуты из разных отношений берутся из одного домена и имеют разные имена, будут каждый раз специально оговариваться.

На практике вместо термина *атрибут* могут использоваться термины *столбец* или *поле* (соответственно, вместо *значение атрибута* в кортеже - *значение столбца* или *значение поля* в записи).

Так как у множества элементы различимы, т. е. отличаются друг от друга, то у отношения нет двух одинаковых кортежей, так как отношение - это множество, элементами которого являются кортежи. На практике в таблице могут встречаться две идентичные записи - это одно из отличий отношений от таблиц, которые встречаются в бумажных документах.

1.2. РЕЛЯЦИОННАЯ БАЗА ДАННЫХ И ЕЕ СВОЙСТВА

Реляционная база данных - совокупность отношений, содержащих всю информацию, которая должна храниться в БД.

В одной базе данных не должно быть отношений с одинаковыми именами.

База данных *Пациенты-Прививки* содержит три класса информации.

Информация *первого класса* - информация о пациентах с видом и датой прививки. К такой информации относят фамилию пациента и адрес проживания. Эта информация содержится в отношении *Пациенты*, степень которого равна двум, а мощность - трем.

Информация *второго класса* - информация о прививках. К ней относят код прививки, являющийся уникальным и вид прививки, не являющийся уникальными. Такая информация содержится в отношении *Прививки*.

Информация *третьего класса* - информация о кодах и видах прививки для каждого пациента. Эта информация содержится в отношении *Пациенты*, степень которого равна трем, а мощность - пяти. Структура таблицы, используемой для хранения отношения, довольно проста, поскольку все записи имеют одинаковый формат.

Обычно предполагается, что в реляционной базе данных значения атрибутов кортежей являются атомарными (неделимыми), а не множественными. Если отношение удовлетворяет условию атомарности значений атрибутов, то говорят, что отношение находится в первой нормальной форме (**1НФ**).

В первой нормальной форме отношения значения всех атрибутов отношения являются атомарными (неделимыми).

Отметим, что таблица с неатомарными значениями в ячейках легко приводится к требуемому виду путем тривиального дублирования значений атомарных полей.

Число отношений в БД и конкретные атрибуты, приписываемые каждому отношению, определяются в процессе проектирования. После стадии проектирования БД начинается процесс ее реализации на компьютере с помощью системы управления базами данных (СУБД).

Система управления базами данных - программное обеспечение, с помощью которого пользователи могут определять, создавать и поддерживать базу данных в актуальном состоянии, а также осуществлять к ней контролируемый

доступ.

В настоящее время разработано много СУБД, предназначенных для работы с реляционными базами данных. В данном учебно-методическом пособии изучаются методы работы с реляционной БД в среде СУБД MS Access.

Отметим, что в среде (справочной системе) СУБД MS Access используется термин таблица вместо термина отношение.

Важнейшими понятиями, связанными с отношениями реляционной базы данных, являются понятия первичного и внешнего ключа отношения.

Первичный ключ - минимальный набор атрибутов (или один атрибут) отношения, который может быть использован для однозначной идентификации конкретного кортежа.

Зная, что отношение не может иметь два одинаковых кортежа, можно сделать вывод, что отношение всегда имеет первичный ключ (в крайнем случае, им может быть полный набор его атрибутов). Однако при определении первичного ключа должно соблюдаться требование «минимальности». Это значит, что если произвольный единичный атрибут исключить из первичного ключа, оставшихся атрибутов будет недостаточно для однозначной идентификации отдельных кортежей.

Примечание. Иногда бывает несколько вариантов для выбора первичного ключа. В этом случае говорят, что у отношения есть несколько *потенциальных ключей*. Один из потенциальных ключей назначается в качестве первичного ключа.

Примечание. На практике могут встречаться одинаковые фамилии. В этом случае в отношение добавляется атрибут, значения которого могут использоваться как первичный ключ. Например, пациентам приписываются коды с уникальными значениями.

Простой первичный ключ - первичный ключ, состоящий из одного атрибута.

Составной первичный ключ - первичный ключ, состоящий из двух или более атрибутов.

С первичным ключом связано понятие *целостности сущности*.

При *целостности сущности* атрибуты, входящие в первичный ключ, не должны содержать отсутствующих значений. Иногда отсутствующие значения обозначаются специальным определителем - *Null*.

Данное свойство первичного ключа вытекает из его определения. Так как первичный ключ однозначно определяет кортежи отношения, то значения остальных атрибутов отношения *зависят* от значений первичного ключа. Если допустить присутствие определителя *Null* в любом атрибуте первичного ключа, то это означает, что не все его атрибуты нужны для уникальной идентификации кортежей, что противоречит определению первичного ключа.

Если один или несколько атрибутов имеются в двух отношениях, то это обычно указывает на определенную *связь* между кортежами этих отношений.

Примечание. В СУБД MS Access имеются связи двух видов: связи ссылочной

целостности и связи-соединения. Сейчас мы определим *связи ссылочной целостности*. Далее, при описании запросов с использованием *Конструктора запросов*, будут рассмотрен второй вид связи - *связи-соединения*.

Внешний ключ - атрибут или набор атрибутов в отношении, который является первичным ключом в отношении ***B***.

Отношение ***B*** называется *главным отношением (главной таблицей)* относительно отношения ***A*** по указанному внешнему ключу.

Примечание. На практике встречаются ситуации, когда отношение ***B*** может совпадать с отношением ***A***. В таких случаях отношение ***A*** называется *самоссылающимся*. Более подробно об этом изложено в литературных источниках [11] и [13].

Внешний ключ обычно используется для установления между двумя таблицами, так называемой *связи ссылочной целостности*.

Ссылочная целостность означает, что если в отношении имеется внешний ключ, то его значение должно соответствовать значению первичного ключа в каком-либо кортеже главного отношения.

Ссылочная целостность требует, чтобы в отношении не было таких значений атрибутов, входящих во внешний ключ, которых нет в первичном ключе главного отношения. Такая согласованность значений внешних и первичных ключей является своеобразным обручем, скрепляющим воедино всю базу данных.

Обеспечение целостности сущностей и ссылочной целостности необходимо для того, чтобы база данных обладала так называемой *реляционной целостностью*.

Иногда для указания атрибутов, из которых состоят кортежи отношения, используется следующая форма записи: после имени отношения в скобках указывают атрибуты отношения.

Ссылочная целостность таблицы **Пациенты** по ее внешним ключам в среде СУБД MS Access представляется соответствующими линиями связи в схеме данных. Напомним, что у таблицы **Пациенты** два внешних ключа. В схеме данных линиями связываются внешний ключ одной таблицы и соответствующий первичный ключ другой (главной) таблицы. Символы «1» и «∞» на линиях связи указывают, что это связи типа «*один ко многим*». В данном случае это означает, что один пациент может сделать много видов прививок.

Примечания:

СУБД MS Access поддерживает также связи ссылочной целостности типа «*один к одному*». Такая связь устанавливается, когда у одной таблицы связываемое поле является первичным ключом, а у второй таблицы связываемое поле либо является первичным ключом, либо не является первичным ключом, но имеет только одно допустимое значение. Подробнее об этом будет сказано далее. Возможные в реальности связи типа «*многие ко многим*», когда одной записи

таблицы **A** соответствуют несколько записей таблицы **B** и, наоборот, одной записи таблицы **B** соответствуют несколько записей таблицы **A**, сводятся к связям типа «один ко многим» таблиц **A** и **B** с третьей дополнительной (связующей) таблицей. В число полей третьей таблицы включаются поля первичных ключей таблиц **A** и **B**, которые, таким образом, становятся внешними ключами третьей таблицы.

К свойствам отношений в реляционной БД в целом можно отнести следующие:

- отношение имеет имя, которое отличается от имен всех других отношений рассматриваемой БД;

- любое значение каждого атрибута является атомарным;
- каждый атрибут в отношении имеет уникальное имя;
- все значения атрибута берутся из одного и того же домена;
- порядок следования атрибутов не имеет никакого значения;
- в одном отношении нет дубликатов кортежей;
- теоретически порядок следования кортежей в отношении не имеет никакого значения.

База данных должна обладать реляционной целостностью - целостностью каждой сущности и ссылочной целостностью.

1.3. ЯЗЫКИ ЗАПРОСОВ К РЕЛЯЦИОННЫМ БАЗАМ ДАННЫХ

Для того, чтобы пользователь мог выполнить с помощью СУБД вышеперечисленные действия с БД, обычно ему предоставляются языковые средства. В настоящее время существует много искусственных языков, предназначенных для описания пользователем необходимых ему действий по манипулированию данными в БД. Однако стандартом для реляционных БД является язык *SQL (Structured Query Language* - язык структурированных запросов), который в большей или меньшей степени поддерживает любая СУБД, претендующая на звание «реляционной».

Необходимо отметить, что в названии языка *SQL* присутствует слово *Query* - *запрос*. Это связано с весьма важным обстоятельством, которому не во всех руководствах по работе с БД уделяется должное внимание. Среди всех многочисленных функций СУБД *главная* - *выдавать ответы на поступающие запросы*. Дело в том, что обычно базы данных содержат огромный объем информации, который пользователь не может в целом «переварить» и осмыслить. Он только может работать с отдельными «срезами» информации и извлекать их из БД с помощью запросов.

Хотя *SQL* и задумывался как средство работы конечного пользователя, в конце концов, он стал настолько сложным, что превратился в инструмент программиста, так объем опубликованного стандарта языка *SQL* более 2000 страниц.

Для пользователей, не являющихся профессиональными программистами, в

СУБД Microsoft Access предусмотрен адаптированный язык запросов *QBE* (Query by Example), реализованный в виде *Конструктора запросов*. Важнейшими характеристиками реляционных языков является их селективная мощность и простота их изучения и использования.

Селективная мощность реляционных языков - относительная (сравнительная) характеристика языков, определяющая их возможности для получения требуемой информации из базы данных. В качестве эталона при сравнении селективной мощности различных реляционных языков обычно принимается реляционная алгебра Кодда.

1.4. ОПЕРАЦИИ РЕЛЯЦИОННОЙ АЛГЕБРЫ

Реляционная алгебра Кодда содержит восемь операций: четыре теоретико-множественные операции (*объединения, пересечения, разности и декартова произведения*) и четыре специальные реляционные операции (*выборки, проекции, соединения и деления*).

Многие авторы считают, что язык является *реляционно полным*, если позволяет для любого конечного набора отношений R_1, R_2, \dots, R_n в первой нормальной форме определить любое отношение, выводимое из R_1, R_2, \dots, R_n с помощью выражений реляционной алгебры Кодда. Другими словами, выразительная мощность реляционно полного языка не должна уступать реляционной алгебре Кодда.

Операндами и результатами операций реляционной алгебры Кодда являются отношения. Операции выборки и проекции являются унарными, поскольку они работают с одним отношением. Другие операции работают с парами отношений, и поэтому их называют бинарными операциями.

В среде СУБД MS Access все восемь операций реляционной алгебры Кодда могут быть реализованы на языке *QBE*, т. е. с помощью *Конструктора запросов*. Другими словами, язык *QBE* является реляционно полным и его селективная мощность не меньше селективной мощности реляционной алгебры Кодда.

Примечание. На самом деле селективная мощность языка *QBE* больше селективной мощности реляционной алгебры Кодда, так как в конструкторе запросов СУБД MS Access имеются дополнительные возможности (дополнительные операции) по работе с таблицами реляционной базы данных:

- возможность работы с частью значения атрибута (подстрокой атрибута-строки);
- возможность установления между таблицами левого и правого внешнего соединения;
- возможность создания перекрестного запроса, результатом которого является таблица, в которой на уровень названий атрибутов выносятся конкретные значения атрибутов, т. е. данные переносятся на уровень метаданных. Метаданные - это данные о данных.

Наиболее важными и часто используемыми при выполнении запросов к реляционной базе данных являются операции выборки, проекции, соединения и

декартова произведения.

Декартово произведение (Cartesian product)

Операцию обозначают следующим образом: $T = R \times S$. Синтаксис: $R \text{ TIMES } S$.

Входные отношения R и S могут иметь разный состав атрибутов. Результирующее отношение T включает все атрибуты исходных отношений. Кортежами отношения T являются всевозможные попарные конкатенации («сцепления») кортежей отношений R и S . Таким образом можно сделать вывод, что степень результирующего отношения T равна сумме степеней исходных отношений R и S , а мощность результирующего отношения T равна произведению мощностей исходных отношений R и S .

Выборка (Selection)

Операцию *Выборка* обозначают формулой: $T = \sigma_c(R)$. Синтаксис: $R \text{ WHERE } c$

На входе используется одно отношение R , результат - новое отношение T , имеющее тот же состав атрибутов. Результирующее отношение T содержит подмножество кортежей исходного отношения R , удовлетворяющих условию выборки, которое обозначено буквой c .

Проекция (Projection)

Обозначить операцию *Проекция* можно следующим образом: $T = \pi_{A1, A2, \dots, An}(R)$.

Операция проекции производит выборку из каждого кортежа отношения-операнда R значений атрибутов $A1, A2, \dots, An$, входящих в список, и удаляет из полученной таблицы повторяющиеся строки.

Соединение (Join)

Операция *соединение (join)* имеет несколько модификаций: *тетасоединение*, *эквисоединение*, которое еще называют «соединение по эквивалентности», *естественное соединение*, *левое внешнее соединение*, *правое внешнее соединение* и *полное внешнее соединение*.

В СУБД Access могут устанавливаться связи-соединения, позволяющие непосредственно реализовать *эквисоединение*, *естественное соединение*, *левое внешнее соединение* и *правое внешнее соединение*.

Эквисоединение. Обозначение операции можно представить следующим образом: $T = R \bowtie S$.

Отношения-операнды R и S могут иметь разный состав атрибутов. Однако в R должны быть атрибуты $A1, A2, \dots, An$, а в S - атрибуты $B1, B2, \dots, Bn$ такие, что $A1$ и $B1$ определены на домене $D1$, $A2$ и $B2$ - на домене $D2$, ..., An и Bn определены на домене Dn . Таким образом, значения атрибутов из каждой пары могут быть сравнены на совпадение (равенство). Это условие сравнения на совпадение (равенство) обозначено буквой c .

Естественное соединение (обозначение $T = R \bowtie S$) аналогично *эквисоединению*, но есть несколько существенных отличий:

- предполагается, что атрибуты каждой пары Ai и Bi ($1 \leq i \leq n$), значения которых проверяются на совпадение, должны иметь одинаковые имена.

в результирующем отношении T из каждой пары атрибутов A_i и B_i ($1 \leq i \leq n$) остается только один атрибут. Так как атрибуты A_i и B_i ($1 \leq i \leq n$) имеют одинаковые имена, то не возникает вопросов, какой атрибут из R сравнивается на совпадение с атрибутом из S , и какое имя дать атрибуту, остающемуся в результирующем отношении T .

Степень результирующего отношения T естественного соединения равна сумме степеней отношений-операндов R и S минус количество общих атрибутов n .

Примечание. Естественное соединение отношений R и S можно получить из их эквисоединения, применив к последнему операцию проекции. Можно показать, что в данном случае результат операции проекция не содержит кортежей-дубликатов.

На практике из всех видов операций соединения чаще всего используется естественное соединение.

Поэтому, если иное специально не оговорено, то под операцией соединения обычно понимают операцию естественного соединения.

Левым внешним соединением называется соединение, при котором записи таблицы R , не имеющие совпадающих значений в общих столбцах таблицы S , также включаются в результирующую таблицу T . Предполагается, что поля отношений R и S , по которым производится соединение, имеют одинаковые имена, как и при естественном соединении.

Для обозначения отсутствующих значений во второй таблице S используется пустое (неопределенное) значение *Null*.

Аналогично определяется *правое внешнее соединение*.

Необходимо отметить, что из восьми операций реляционной алгебры только пять являются примитивами - выборка, проекция, декартово произведение, объединение и разность. Остальные три операции - пересечение, соединение и деление - могут быть определены через первые пять. Соответствующие формулы (аналогичные формуле 2) приводятся во многих литературных источниках по реляционным базам данных [12, с. 130, 131].

1.5. МЕТОД «СУЩНОСТЬ-СВЯЗЬ» ЛОГИЧЕСКОГО ПРОЕКТИРОВАНИЯ (ER-МЕТОД). ОСНОВНЫЕ ЭТАПЫ ПРОЕКТИРОВАНИЯ БАЗЫ ДАННЫХ

Создание базы данных в среде системы управления базами данных Access (СУБД Access) предполагает выполнение следующих основных этапов:

- концептуальное проектирование;
- логическое проектирование;
- физическое проектирование;
- использование БД (заполнение БД оперативной информацией и формирование запросов и отчетов).

Концептуальное проектирование - процедура конструирования информационной модели, не зависящей от условий реализации БД. Таким

образом, сконструированная на данном этапе информационная модель не зависит ни от СУБД, ни от средств вычислительной техники.

Концептуальное проектирование БД выполняется на основе анализа информационных потоков, использования классификаторов и систем кодирования, определения диапазона действия и области применения БД, выяснения состава ее пользователей, сбора и анализа требований пользователей.

В настоящем пособии не рассматривается методика проведения концептуального проектирования. Предполагается, что к моменту начала логического проектирования базы данных сконструирована информационная модель рассматриваемой предметной области.

На этапе логического проектирования информационная модель уточняется с учетом типа создаваемой БД (реляционной, сетевой или иерархической). В настоящее время реляционные модели базы данных практически повсеместно вытеснили все другие типы моделей. В СУБД Access реализована именно реляционная база данных.

Процесс физического проектирования БД предполагает выполнение в среде выбранной СУБД следующих работ:

- описание логической структуры каждой таблицы;
- описание связей между таблицами, входящими в одну БД;
- первоначальное заполнение справочников БД необходимой нормативно-справочной информацией.

Подробные сведения об этапах жизненного цикла БД изложены в работах К. Дж. Дейта, Т. Коннолли и др. [11-13].

Подчеркнем, что концептуальное проектирование БД не связано с какой-либо конкретной СУБД, а этап логического проектирования зависит только от типа СУБД (сетевой, иерархической или реляционной). Однако способ представления результатов концептуального проектирования зависит от используемого метода логического проектирования. Используемые термины «отношение» и «атрибут» относятся к реляционной СУБД, каковой и является СУБД Access.

Поскольку одна из целей нашего курса состоит в изучении технологии применения СУБД Access для реализации реляционных баз данных, то последние два этапа создания баз данных предполагают знание пользователем именно этой СУБД.

1.6. КОНЦЕПЦИЯ ER-МЕТОДА ЛОГИЧЕСКОГО ПРОЕКТИРОВАНИЯ

Существуют различные подходы представления информационной модели рассматриваемой предметной области как результата концептуального проектирования и, соответственно, как набора исходных данных для логического проектирования. В настоящем пособии рассматривается *ER-метод* логического проектирования. Данный метод предполагает, что в результате выполнения концептуального проектирования БД должно быть достигнуто следующее:

- выделены все сущности, информация о которых должна содержаться в

искомой БД;

- определены основные атрибуты для каждой сущности;
- назначен ключевой атрибут для каждой сущности;
- сформулированы связи между выделенными сущностями;
- выявлены условия применения выделенных сущностей на данном предприятии.

Таким образом, информационная модель предметной области должна включать указанные пять компонентов.

В качестве определения *ER-метода* можно принять следующее:

- основу *ER-метода* составляют понятия «сущность», «связь» и «атрибуты»;
- суть *ER-метода* состоит в наборе формализованных процедур и правил, позволяющих получить полный набор таблиц и определить структуру каждой таблицы на основе приведенных пяти компонентов информационной модели предметной области.

Особенностью *ER-метода* является то, что для одной и той же проблемы разные проектировщики могут получать различные наборы сущностей и связей. Определение лучшего из этих наборов может быть вопросом личного предпочтения. Само проектирование БД не является полностью формализуемым процессом, но *ER-метод* привносит в процесс проектирования достаточно много четких процедур.

1.7. ОСНОВНЫЕ ТЕОРЕТИЧЕСКИЕ ПОНЯТИЯ

Сущность определяется как некоторый объект, представляющий интерес, информация о котором должна храниться в БД. Этот объект должен иметь *экземпляры*, отличающиеся друг от друга и допускающие однозначную идентификацию. Сущность представляется, как правило, именем существительным. Примерами сущностей могут служить машины, банковские счета, учебные заведения, служащие и договоры.

Кроме выделения сущностей на этапе концептуального проектирования обычно определяют обязательный минимум свойств (атрибутов) каждого объекта.

Атрибут есть свойство сущности.

Например, атрибутами (свойствами) сущности *Пациент* являются *ФИО*, *Код пациента*, *Адрес*, *Телефон* и т. д.

Среди множества атрибутов каждого объекта (сущности) должен присутствовать так называемый ключевой атрибут или ключ сущности, однозначно идентифицирующий конкретный экземпляр данной сущности.

Атрибут, или набор атрибутов, используемый для однозначной идентификации экземпляра сущности, называется *ключом сущности*.

Для сущности *Пациент* ключом сущности является номер пациента. Определенное значение этого номера однозначно указывает на конкретного пациента.

На этапе логического проектирования определяются все таблицы (*отношения*) БД и полный список их *атрибутов*. В некоторых книгах для таблиц (отношений), получаемых на этапе логического проектирования,

используется термин «*информационный объект*».

Связь представляет собой соединение двух или более сущностей. При поиске связей в основном следует полагаться на то обстоятельство, что связь обычно выражается глаголом.

Характеристики связи во многом определяются условиями применения сущностей. *Условия применения* - это правила использования выделенных для БД объектов.

1.8. ER-ДИАГРАММЫ И ХАРАКТЕРИСТИКИ СВЯЗИ

В результате концептуального проектирования было выявлено, что двумя главными объектами, или *сущностями*, представляющими в данном случае интерес, являются *Пациенты* и *Прививки*.

Сущность *Пациенты* характеризуется такими атрибутами, как *Код пациента*, *ФИО*, *Дата рождения*, *Телефон* и др. Для дальнейшего рассмотрения важно лишь то, что атрибут *Код пациента* является ключом сущности *Пациенты*. Это означает, что значение атрибута *Код пациента* однозначно определяет экземпляр сущности *Пациенты*.

Сущность *Прививки* обладает следующими атрибутами: *Код прививки*, *Вид прививки* и т. д. Ключом сущности является атрибут *Код прививки*.

На диаграммах ER-типа сущности представляются в виде прямоугольников, а связи - в виде ромбов. Непосредственно под прямоугольным блоком каждой сущности выписывается и выделяется подчеркиванием ключ этой сущности: *Код пациента* для сущности *Пациенты* и *Код прививки* для сущности *Прививки*.

В большинстве случаев для определения набора *отношений* проектируемой БД используются диаграммы *ER-типа*, а не диаграммы *ER-экземпляров*.

Важной характеристикой связи между двумя (и более) сущностями является *степень связи*.

Степень связи указывает на количество сущностей, охваченных данной связью. *Связь*, существующая между сущностями *Пациенты* и *Прививки*, называется *бинарной*, поскольку она связывает только две сущности. Связи более высокого порядка, существующие между *n* сущностями, называются *n-сторонними (n-арными)*. Бинарные связи встречаются наиболее часто и во всех примерах настоящего пособия рассматриваются только бинарные связи.

Показатель кардинальности описывает количество возможных связей для каждого из экземпляров рассматриваемых сущностей.

Наиболее распространенными являются бинарные связи с показателями кардинальности «*один к одному*» (1:1), «*один ко многим*» (1:n) или «*многие к одному*» (n:1), «*многие ко многим*» (m:n).

1.9. МЕТОДИКА ПРИМЕНЕНИЯ ER-МЕТОДА

В качестве исходных данных для выполнения логического проектирования БД на этапе концептуального проектирования должно быть определено

следующее:

- набор сущностей;
- предварительный перечень атрибутов для каждой сущности (основные атрибуты);
- ключевой атрибут для каждой сущности;
- набор связей между сущностями;
- описание условий применения объектов (сущностей) в данной ситуации.

ER-метод логического проектирования предполагает выполнение следующего:

• если в результате концептуального проектирования БД были выделены документы-сущности, то необходимо каждую из них представить в виде двух сущностей: *Заголовок документа* и *Строка документа*;

• определение показателя кардинальности для каждой из связей (на основе производственных условий использования сущностей и, если это необходимо, диаграммы *ER-экземпляров*);

- определение класса принадлежности каждой сущности в каждой связи;
- построение диаграммы *ER-типа*;
- определение всех отношений БД, их атрибутов и первичных ключей.

Также следует отметить роль диаграмм *ER-типа* и *ER-экземпляров* при проектировании базы данных.

Построение диаграммы *ER-типа* при проектировании БД дает возможность выбрать правило генерации отношений. Кроме того, совокупность диаграмм *ER-типа* для всех связей проектируемой БД представляет собой компактное и наглядное представление структуры БД. Такое изображение структуры БД полезно как для предварительного знакомства с БД, так и для процесса ее модификации.

Диаграмма *ER-экземпляров* дает возможность определить показатель кардинальности связи и класс принадлежности каждой сущности для рассматриваемой связи. Для того чтобы построить диаграмму необходимо выполнить следующее:

- Определить, сколько линий (одна или больше) может исходить из точки в левой части диаграммы.
- Определить, сколько линий (одна или больше) может входить в точку правой части диаграммы.
- Уточнить, может ли в левой части диаграммы быть точка, из которой не выходит ни одна линия.
- Определить, может ли в правой части диаграммы быть точка, в которую не входит ни одна линия.

При этом действуют следующие правила:

1. Если точка в левой части диаграммы может быть соединена только с одной точкой в правой части, то связь будет иметь показатель ...: 1:
2. Если точка в левой части диаграммы может быть соединена с несколькими точками в правой части, то связь будет иметь показатель ...: *n*:
3. Если точка в правой части диаграммы может быть соединена только с одной точкой в левой части, то связь будет иметь показатель 1: ...:

4. Если точка в правой части диаграммы может быть соединена с несколькими точками в левой части, то связь будет иметь показатель n :

5. Если в левой части диаграммы может существовать точка, несвязанная ни с одной точкой в правой части, то класс принадлежности левой сущности является необязательным:

6. Если в правой части диаграммы может существовать точка, несвязанная ни с одной точкой в левой части, то класс принадлежности правой сущности является необязательным.

Раздел 2. Практическая работа в СУБД «ACCESS»

Система управления базами данных Microsoft Office Access

Microsoft Office Access - псевдореляционная СУБД (Система управления реляционными базами данных) корпорации Microsoft. Имеет широкий спектр функций, включая связанные запросы, связь с внешними таблицами и базами данных. Благодаря встроенному языку VBA (Visual Basic for Applications), в Access можно писать приложения, работающие с базами данных. MS Access является файл-серверной СУБД и потому применима лишь к маленьким приложениям. Отсутствует ряд механизмов, необходимых в многопользовательских БД, таких, например, как триггеры. Для обучения работе в программе MS Access студентам СПбГПМА предлагается готовая информационно-логическая модель учета детских прививок.

2.1. ЗАПУСК MICROSOFT OFFICE ACCESS

- Нажмите кнопку **Пуск**.
- Выберите пункт **Все Программы – Microsoft Office**.
- Выберите символ, запускающий **Microsoft Office Access** (ключ).

2.2. СОЗДАНИЕ ФАЙЛА НОВОЙ БАЗЫ ДАННЫХ

При запуске MS Access на экране автоматически открывается диалоговое окно, предлагающее создать новую базу данных или открыть уже существующую.

- Для создания новой базы данных нажмите **Новая база данных**.
- Дайте имя новой базе данных, для чего введите в поле **Имя файла**: *«Ваша фамилия и № группы»*. (Имя файла может содержать до 225 знаков.)
- Нажмите кнопку **Создать**. Расширение файла **.accdb** добавляется автоматически. Таким образом, полное имя файла будет типа: *«Петров 110. accdb»*.

2.3. СОЗДАНИЕ ТАБЛИЦЫ БАЗЫ ДАННЫХ В РЕЖИМЕ КОНСТРУКТОР


Сначала необходимо создать структуры таблиц, входящих в новую базу.

- Нажмите **Создание** и затем **Конструктор Таблиц**. Перед Вами диалоговое окно, в котором необходимо создать структуру Вашей таблицы, то есть задать имена полей и типы хранящихся в ней данных.
- Начните с названия полей. Первое поле - **Код пациента**, впишите его в первую строку. Нажмите клавишу перемещения курсора вниз или клавишу **Enter**. Во вторую строку впишите название следующего поля – **ФИО пациента**. Аналогично введите названия следующих полей: **Дата рождения**, **Почтовый адрес**, **Телефон**, **Страховка**, **Диагноз**, **Примечание**.

- Перейдите к указанию типа данных полей для того, чтобы программа быстрее и лучше могла обрабатывать информацию.
- Когда в таблице создается новое поле, программа автоматически выбирает тип данных **Текстовый**. Необходимый тип данных для каждого созданного Вами поля можно определить по следующей ниже таблице.

Типы полей в Microsoft Office Access

Тип данных	Описание
Текстовый	Может состоять из цифр, букв или их комбинаций, используется в полях, не требующих вычислений. Может содержать до 255 символов.
Поле МЕМО	Используется для хранения длинных текстов (до 65535 символов).
Числовой	В поле этого типа могут вводиться только цифры.
Дата/время	Тип данных для полей, в которых указываются дата и время.
Счетчик	Используется для абонентных номеров. Число, увеличивающееся на определенную или случайную величину.
Логический	Указывается в качестве контроля для окошка с флажком, который может иметь только два значения Да и Нет.
Гиперссылка	Текст или комбинация текста и цифр, для адреса гиперссылки – то есть имени файла, страницы в Интернет и т.п.
Мастер подстановок	Тип данных, извлекающий значения из другой таблицы или списка.
Денежный	Для денежных и числовых значений.

- Поместите курсор под рубрикой **Тип данных** в строке **Код пациента**. В правой части поля появляется направленная вниз стрелка. Щелкните по ней, чтобы появился список альтернатив. Выберите тип данных **Текстовый**.
- Тип данных **Текстовый** поля **Дата рождения** замените типом **Дата / Время**. Щелкните по направленной вниз стрелке и в списке альтернатив выберите соответствующий тип.
- Поставьте курсор на строку **«Формат поля»** в нижней части окна. В открывшемся окне выберите **«краткий формат даты»**.
- Тип данных **Текстовый** поля **Телефон** замените типом **Числовой**. Для этого типа данных необходимо ввести маску ввода. Для типа данных **Числовой** маска ввода вводится с клавиатуры и записывается как: **000-00-00**.
- Тип поля **Примечание** замените типом **Поле Мемо**. Тип данных **Поле Мемо** используется для большого объема записей.
- Поле **Код пациента** отметьте как **Ключевое поле** для обеспечения уникальности значения. Для этого поместите курсор в любое место поля **Код пациента**. Нажмите **Конструктор** и далее **Ключевое поле**. При этом слева от надписи **Код пациента** появится символ ключа .
- Сохраните созданную структуру таблицы под именем **Пациенты**.

- По аналогии с таблицей **Пациенты** создайте еще таблицы и сохраните их под именами **Прививки**, **План прививок** и **Регистрация прививок**.
- Выберите тип данных **Текстовый** для поля **Код прививки** в таблице **Прививки**. Для остальных полей таблиц типы данных выберите самостоятельно, ориентируясь на информацию, содержащуюся в колонках таблицы.

2.4. СОЗДАНИЕ СВЯЗЕЙ МЕЖДУ ТАБЛИЦАМИ И ЗАПОЛНЕНИЕ ТАБЛИЦ

В нашем примере таблицы последовательно связаны одна с другой. Таблицы **План прививок** и **Регистрация прививок** связаны с таблицами **Пациенты** и **Прививки**. Поля **Код пациента** и **Код прививки** в таблицах **План прививок** и **Регистрация прививок** могут содержать только те значения, которые встречаются в поле **Код пациента** в таблице **Пациенты** и **Код прививки** в таблице **Прививки**.

Создайте связи между таблицами, выбрав **Работа с базами данных - Схема данных**. В диалоговом окне, появившемся на экране, выделите таблицы, между которыми будет установлена связь. Щелкните по кнопке **Добавить**, после чего закройте это диалоговое окно.

- Выделите мышкой поле **Код пациента** в таблице **Пациенты** и, удерживая нажатой левую кнопку мыши, перетащите его на поле **Код пациента** в таблицу **План прививок**. Связь создается между одинаковыми полями таблиц. В диалоговом окне **Изменение связей**, появившемся на экране, включите опцию **Обеспечение целостности данных**. При этом Access будет отслеживать совпадение номеров в обеих таблицах. Включите также опции **Каскадное обновление связанных полей** и **Каскадное удаление связанных полей**, для внесения и удаления исправлений во всех связанных таблицах, после чего щелкните по кнопке **Создать**.
- Убедитесь, что между двумя таблицами появилась корректная связь в виде соединительной линии между одинаковыми полями, на концах которой у таблицы **Пациенты** появился значок **1**, а у таблицы **План прививок** – значок ∞ .
- По аналогии создайте связи между таблицами **Пациенты** и **Регистрация прививок**, а также между таблицей **Прививки** и таблицами **План прививок** и **Регистрация прививок**. Закройте окно **Схема данных** и подтвердите ее сохранение.

Результат формирования схемы данных показан на рисунке 1.

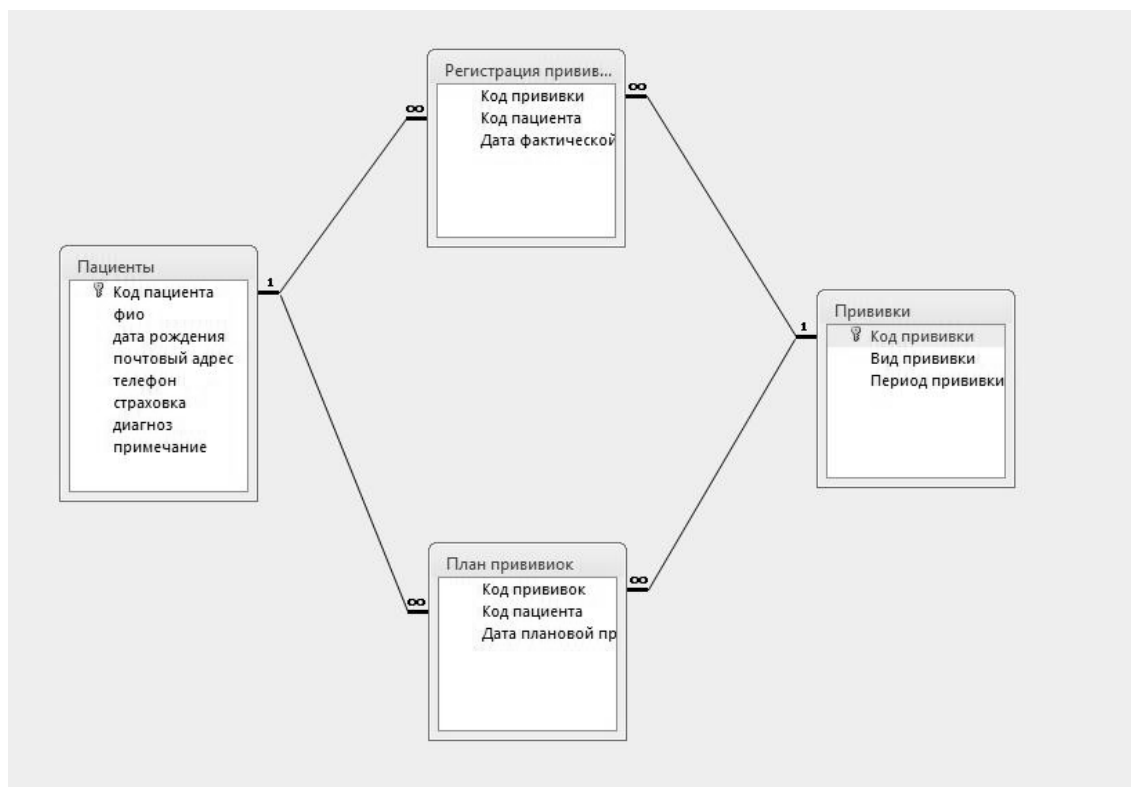


Рис.1 Схема данных

Откройте и заполните таблицы в соответствии с предлагаемыми ниже образцами.

Пациенты

Код пациента	ФИО пациента	Дата рождения	Почтовый адрес	Телефон	Страховка	Диагноз	Примечание
01	Сидоров Олег	01.09.1994	ул. Ленина 54	113-45-66	Росно	Диарея	В стадии обострения
02	Исаков Сергей	01.03.1994	ул. Марата 42	119-21-34	Lloyd's	Аллергия	Лекарственная аллергия
03	Петров Иван	01.04.1994	ул. Седова 6	113-45-78	Росно	Бронхит	В стадии обострения
04	Иванов Петр	01.05.1995	ул. Салова 5	111-55-59	Росно	Аллергия	Лекарственная аллергия
05	Каверин Михаил	01.05.1996	пр. Энгельса 14	112-35-62	Lloyd's	Диарея	В стадии обострения

06	Карелов Егор	04.06.199 7	ул. Есенина 37	111-25-6 3	Lloyd's	Диарея	В стадии обострения
07	Селина Жанна	09.10.200 7	пр. Лесной 32	113-15-6 4	Росно	Диарея	В стадии обострения
08	Сорокин Лев	11.12.200 7	пр. Энгельса 61	112-75-6 6	Lloyd's	Диарея	Лекарствен- ная аллергия
09	Жукова Кира	14.07.200 8	ул. Седова 33	114-85-6 7	Lloyd's	Пневмония	В стадии обострения
10	Розова Дарья	15.03.200 8	ул. Павлова 10	114-95-6 9	Росно	Диарея	Лекарствен- ная аллергия

Прививки

Код прививки	Вид прививки	Период прививания
011	Гепатит В	6 - 12 часов
012	Гепатит В	1 месяц
013	Гепатит С	6 месяцев
021	БЦЖ	3 - 7 дней
022	БЦЖ	7 лет
023	БЦЖ	14 лет
031	АКДС	3 - 6 месяцев
032	Полиомиелит	18 месяцев
033	ММР II	7 лет

План прививок

Код прививки	Код пациента	Дата плановой прививки
011	01	01.09.1994
012	01	02.10.1994
013	01	02.03.1995
021	01	04.09.1995
022	01	01.03.1995
023	01	01.02.2008
031	01	01.12.1994
032	01	01.12.1995
033	01	01.12.2009

011	02	04.09.1994
012	02	04.09.2001
013	02	02.03.1995
021	02	01.09.1994
022	02	02.10.1994
023	02	01.02.2008
011	03	01.12.1994
012	03	02.10.1994
013	03	08.06.2003
011	05	30.05.1996
012	05	29.06.1996
013	05	28.02.1997
021	05	20.03.2002
022	05	24.03.2009
023	05	21.03.2016
031	05	29.11.1996
032	05	29.01.1998
033	05	28.12.2003
011	06	04.06.1997
012	06	04.07.1997
021	06	11.06.1997
022	06	04.06.2004
011	07	10.10.2007
012	07	11.11.2007
013	07	12.05.2008
011	08	11.12.2007
012	08	12.01.2008
013	08	11.07.2008
011	10	16.03.2008

Регистрация прививок

Код прививки	Код пациента	Дата фактической прививки
011	01	01.09.1994
011	05	01.05.1996

011	06	04.06.1997
011	07	09.10.2007
011	08	11.12.2004
011	10	15.03.2008
012	01	04.09.2001
012	05	02.05.1996
012	06	07.07.1997
012	07	11.11.2007
012	08	12.01.2005
013	01	08.12.2004
013	05	01.07.1996
013	07	12.06.2008
021	01	10.08.1996
021	02	16.03.2008
021	03	01.05.1996
021	05	04.05.1996
021	06	11.06.1997
022	01	12.09.1997
022	02	21.06.2008
022	03	04.06.1998
022	05	06.08.1996
022	06	04.06.2004
023	01	15.01.2008
023	02	01.09.2008
023	03	09.10.2008
031	01	11.12.2007
031	05	01.03.1996
032	01	15.03.2008
032	05	02.05.1996

2.5. СОЗДАНИЕ ЗАПРОСОВ

Запросы используются для просмотра, изменения и анализа данных из одной или нескольких таблиц. Наиболее распространенный тип запроса - запрос на выборку. Запрос этого типа извлекает запрашиваемую информацию из одной или нескольких таблиц базы данных и отображает результат в сводной таблице.

Запрос можно создать в режиме **Конструктор** и с помощью **Мастера запросов**.

2.5.1. СОЗДАНИЕ ЗАПРОСА В РЕЖИМЕ КОНСТРУКТОРА

Создайте запрос, содержащий поля: **Код пациента** и **ФИО пациента** из таблицы **Пациенты**, **Вид прививки** из таблицы **Прививки**, **Дата плановой прививки** из таблицы **План прививок** и **Дата фактической прививки** из таблицы **Регистрация прививок**.

- Щелкните **Создание – Конструктор запросов**. В окне **Добавление таблицы** выделите *все* таблицы, на которых будет базироваться запрос, щелкните по кнопке **Добавить** и закройте окно. Выберите поля, которые будут необходимы в новом запросе.
- Сохраните созданный Вами запрос.
- Откройте созданный Вами запрос в режиме **Конструктор** и установите *критерий*, с помощью которого можно указать, какого рода информацию должен предоставить запрос (*горизонтальный фильтр*). В зависимости от поля, для которого будет указан критерий, Вы получите более конкретизированную информацию.
- Получите всю указанную в запросе информацию для пациента под кодовым номером пять. Поместите курсор в строку **Условие отбора** ниже поля **Код пациента** и введите число **05**.
- Перейдите в режим **Таблицы** и посмотрите, как выглядит запрос.

2.5.2. СОЗДАНИЕ ЗАПРОСА ПРИ ПОМОЩИ МАСТЕРА ЗАПРОСОВ

- При помощи **Мастера запросов** создайте запрос, содержащий поля: **Код пациента**, **ФИО пациента**, **Телефон**, **Вид прививки**, **Дата плановой прививки** и **Дата фактической прививки**.
- Щелкните **Создание – Мастер запросов**. Убедитесь, что выбран способ создания **Простой запрос** и щелкните по кнопке **ОК**. В отдельной строке **Таблицы и запросы**, щелкнув по направленной вниз стрелке, из появившегося списка выберите таблицу **Пациенты**. Ниже в левом окне **Доступные поля** появится список полей указанной таблицы. Выберите поле **Код пациента** и, нажав кнопку с символом « > », переведите его в правое окно **Выбранные поля**. Аналогично добавьте поля **ФИО пациента** и **Телефон**.
- Щелкните по направленной вниз стрелке в строке **Таблицы и запросы** и выберите таблицу **Прививки**. Переведите в правое окно **Выбранные поля** поле **Вид прививки**.
- Аналогично переведите в правое окно **Выбранные поля** поле **Дата плановой прививки** и поле **Дата фактической прививки** из соответствующих таблиц.

- Два раза нажмите кнопку **Далее**. Введите название «**Сведения о прививках**» для создаваемого запроса и нажмите кнопку **Готово**.
- Посмотрите созданный запрос, перейдя в режим таблицы.

2.5.3. СОЗДАНИЕ ЗАПРОСА, СОДЕРЖАЩЕГО ИТОГОВЫЕ ЗНАЧЕНИЯ

- Создайте с помощью мастера запрос, для подсчета количества прививок, сделанных каждому пациенту, реализуя алгоритм **Простой запрос**.
- Выберите поля **Вид Прививки**, **Дата фактической прививки** из соответствующих таблиц и нажмите кнопку **Далее**. Выберите **итоговый отчет**, установив флажок «**Подсчет числа записей**», затем выберите интервал группировки дат по годам и нажмите **Готово**.
- Посмотрите созданный запрос, перейдя в режим таблицы.
- Перейдите в режим **Конструктор**. Задайте параметр отбора, записав в строку **Условие отбора**: [**Год прививки**:].
- При переходе в режим просмотра запроса появляется диалоговое окно с просьбой указать параметр отбора.

2.5.4. СОЗДАНИЕ ПАРАМЕТРИЧЕСКОГО ЗАПРОСА

Параметрический запрос создается для выделения необходимой информации, например на заданную дату.

- Создайте параметрический запрос на известную дату. Сначала создайте запрос в режиме **Конструктор**, включив следующие поля: **ФИО пациента**, **Вид прививки** и **Дата фактической прививки**. Поставьте курсор в строку **Условие отбора** и впишите выражение [**Дата прививки**:]. Закройте запрос, сохранив его.
- Откройте созданный запрос. При открытии этого запроса появится окно, в котором укажите известную дату. После нажатия **ОК** появится запрос на заданную дату.

2.5.5. СОЗДАНИЕ ПЕРЕКРЕСТНОГО ЗАПРОСА

Перекрестный запрос создается в два этапа. Сначала необходимо создать обычный запрос, включающий необходимые поля из трех таблиц. На основе этого запроса создается перекрестный запрос, содержащий информацию обо всех сделанных прививках каждому пациенту и итоговые цифры.

1. Создайте обычный запрос, выбрав из соответствующих таблиц поля: **ФИО пациента**, **Вид прививки** и **Дата фактической прививки**. Закройте его, присвоив ему имя **Перекрестный запрос_1**.
2. Создайте перекрестный запрос на основе предыдущего запроса. Щелкните **Мастер Запросов - Перекрестный запрос**.

3. В окне **Создание перекрестных таблиц** щелкните по кнопке **Запросы** и из списка запросов выберите **Перекрестный запрос_1** и нажмите кнопку **Далее**.
4. Из окна **Доступные поля** переведите поле **ФИО пациента** в окно **Выбранные поля**, используемое в качестве заголовков строк и нажмите **Далее**.
5. В следующем окне выберите поле **Вид прививки** для использования в качестве заголовков столбцов и нажмите **Далее**.
6. Для поля **Дата прививки** выберите функцию **Последний** и нажмите **Далее**. Задайте имя запроса - **Перекрестный запрос** и нажмите **Готово**.
7. Просмотрите созданный запрос.

2.5.6. СОЗДАНИЕ ЗАПРОСА «ПОВТОРЯЮЩИЕСЯ ЗАПИСИ»

С помощью запроса **Повторяющиеся записи** можно получить список пациентов, которым уже сделана прививка. Можно также выявить и исключить повторные случайные записи.

- Создайте запрос **Повторяющиеся записи**, содержащий список пациентов, которым сделаны одинаковые прививки. Для этого в **Мастере запросов** выберите **Повторяющиеся записи** и нажмите **ОК**.
- В окне **Поиск повторяющихся записей** выберите таблицу **Регистрация прививок** и нажмите **Далее**. Выберите поле **Код прививки** и нажмите **Далее**. Выберите **Код пациента** и нажмите **Готово**.
- Посмотрите созданный запрос.

2.5.7. СОЗДАНИЕ ЗАПРОСА «ЗАПИСИ БЕЗ ПОДЧИНЕННЫХ»

Запрос с записью без подчиненных выбирает в таблице записи, для которых не существует связанных записей в таблице, например, пациентов, которым не сделаны прививки.

- Выберите **Мастер запросов – Записи без подчиненных**.
- Выберите таблицу **Пациенты** и нажмите **Далее**. В следующем диалоговом окне выберите таблицу **Регистрация прививок** и нажмите **Далее**. В следующем окне выберите поле **Код пациента** из таблицы и нажмите **Далее**. В следующем окне выберите поля **ФИО пациента** и **Телефон** и нажмите **Далее** и **Готово**.
- Посмотрите созданный запрос.

2.5.8. СОЗДАНИЕ ЗАПРОСА С ИСПОЛЬЗОВАНИЕМ ПОСТРОИТЕЛЯ ВЫРАЖЕНИЙ

Запрос для определения количества несделанных прививок создается в три этапа. Первый запрос определяет количество запланированных прививок с момента рождения пациента, второй запрос определяет количество прививок, фактически сделанных пациентам, в третьем запросе строится выражение для расчета прививок, не сделанных пациентам.

- Создайте итоговый запрос для определения количества запланированных прививок, используя таблицы **Пациенты** и **План прививок**. Из указанных таблиц выберите поля **ФИО пациента** и **Дата плановой прививки**. Сохраните запрос, присвоив ему имя «**План**».
- Создайте итоговый запрос для определения количества прививок, фактически сделанных пациентам, используя таблицы **Пациенты** и **Регистрация прививок**. Из этих таблиц выберите поля **ФИО пациента** и **Дата фактической прививки**. Сохраните запрос, присвоив ему имя «**Факт**».
- Создайте запрос в режиме **Конструктор**, используя 2 ранее созданных запроса. Из этих запросов выберите поля **ФИО пациента**, **Count –план прививок**, **Count –регистрация прививок**.
- Создайте выражение в четвертом столбце для расчета несделанных прививок. Поставьте курсор в четвертом столбце и нажмите на кнопке **Построитель**. В диалоговом окне **Построителя выражений** в первом столбце щелкните по запросу **План**. Во втором столбце появятся поля этого запроса. Дважды кликните **Count–план прививок**. В верхней части окна появится следующее выражение: **[План]![Count-план прививок]**. Ту же операцию повторите с запросом **Факт**, выделив поле **Count–регистрация прививок**. Между этими выражениями поставьте знак *минус* и вставьте из раскрывшегося списка функцию **Nz**. Таким образом в верхней части окна появится окончательное выражение: **[План]![Count-план прививок] – Nz([Факт]![Count-регистрация прививок])**.
- Нажмите **ОК**. Данное выражение появится в первой строчке четвертого столбца. Перейдите в режим таблицы и посмотрите созданный запрос.

? Задание для самостоятельной работы: *создайте запрос, для подсчета количества прививок, сделанных клиентам каждой страховой компании.*

2.6. СОЗДАНИЕ ФОРМЫ В РЕЖИМАХ МАСТЕР ФОРМ И КОНСТРУКТОР

Для того чтобы представить запрашиваемую информацию для консультанта, для пациента или для клиента более удобным образом, используются формы.

При создании формы наиболее оптимальным будет вариант использовать **Мастер форм** для создания заготовки формы и затем сделать окончательная правку создаваемой формы в режиме конструктора.

- Создайте форму для таблицы с помощью **Мастера форм**. Щелкните **Создание – Другие формы – Мастер форм**.

- Выберите все поля из таблицы **Пациенты**, поле **Код прививки** из таблицы **Регистрация прививок**, поле **Вид прививки** из таблицы **Прививки** и поле **Дата прививки** из таблицы **Регистрация прививок**. Нажмите **Далее**.
- Выберите табличный вид формы и нажмите **Далее**, выберите необходимый стиль оформления, задайте имя создаваемой формы - **Прививки** и нажмите **Готово**.
- Перейдите в режим **Конструктор**. Создайте заголовок формы. Для этого подведите курсор к горизонтальной черте между заголовком формы и областью данных, нажмите левую кнопку мыши и, удерживая ее нажатой, растяните область заголовка.
На ленте выберите элемент «надпись» **Aa** и поместите его в области **Заголовок формы**. Введите текст «**Сведения о прививках**». Выберите крупный шрифт, **размер 16** и расположите заголовок по центру формы. Нажмите клавишу **Enter**.
- Создайте поле со списком страховок. Если пройти курсором по кнопкам ленты, то будет появляться название каждой команды. Нажмите на кнопку **Поле со списком** и измененным курсором выделите поле на свободном месте формы. В окне мастера выделите *фиксированный набор значений* и нажмите **Далее**. В столбец введите названия страховых компаний и нажмите **Далее**. Сохраните этот список в поле **Страховка** и нажмите **Далее** и **Готово**.
- Перейдите в режим **Формы** и просмотрите полученную форму.
- Перейдите в режим **Конструктор** и создайте кнопки управления формой:

- а) кнопка перехода на предыдущую запись,*
- б) кнопка перехода на следующую запись,*
- в) кнопка закрытия формы.*

На панели элементов найдите команду **Кнопка** и измененным курсором выделите место для кнопки в виде квадрата. Далее, следуя советам мастера, создайте все требуемые кнопки.

? Задание для самостоятельной работы: *создайте примеры форм Сводная диаграмма и Сводная таблица.*

2.7. СОЗДАНИЕ ОТЧЕТА

Отчеты представляют собой эффективный способ представления данных в печатном формате. Возможность управления размером и внешним видом всех элементов отчета позволяет отобразить необходимую информацию из базы данных желаемым образом.

- Создайте отчет по данным имеющихся таблиц. Щелкните **Создание - Мастер отчетов**.

- Выберите поля **ФИО пациента**, **Дата рождения**, **Почтовый адрес**, **Телефон**, **Вид прививки**, **Дата плановой прививки**, **Дата фактической прививки** и нажмите кнопку **Далее**.
В следующем окне добавьте уровень группировки по полю **Дата фактической прививки**. Нажмите кнопку **Группировка...**, задайте группировку по **кварталам** и нажмите кнопку **ОК**. Нажмите кнопку **Далее**.
Задайте сортировку по фамилии пациента, нажмите **Далее**. Выберите требуемый вид отчета, нажмите кнопку **Далее**. Выберите требуемый стиль отчета, нажмите кнопку **Далее**. Задайте имя отчета **Прививки** и нажмите кнопку **Готово**.
- Перейдите в режим **Конструктор** и задайте цвет шрифта заголовка **синий на желтом фоне**.
- Расположите поле **ФИО пациента** в центре отчета, ниже - название прививки. Все остальные данные поместите в строчку под этими полями.

? **Задание для самостоятельной работы:** *создайте отчет, содержащий данные о пациентах, приписанных к разным страховым компаниям с группировкой по названию компании.*

2.8. СОЗДАНИЕ МАКРОСА

Макрос - это набор из одной или нескольких команд, выполняющих определенные действия. Макросы позволяют автоматизировать часто выполняемые в Access последовательности действий.

- Щелкните **Создание – Макрос**.
- В открывшемся диалоговом окне поставьте курсор на первое поле в столбец с названием **Макрокоманда**.
- Щелкните по направленной вниз стрелке справа в столбце. В открывшемся списке доступных макрокоманд выберите команду **ОткрытьФорму**.
- Внизу открывается диалоговое окно **Аргументы макрокоманды**.
- Щелкните мышью в поле **Имя формы**.
- Щелкните по направленной вниз стрелке.
- В списке альтернатив выберите форму **Прививки**, это делается для того, чтобы выполнить дополнительные установки, когда точно известно, какая форма будет использоваться (например, запрос или другая форма представления данных).
- Щелкните мышью в поле **Макрокоманда** ниже строки **ОткрытьФорму**, выберите макрокоманду **НаЗапись**.
- Поместите курсор в поле **Запись** и щелкните на направленной вниз стрелке.
- Выберите аргумент **Новая**. После этой команды макрос сначала откроет форму **Прививки**, а затем перейдет на новую запись.
- Введите еще одну макрокоманду: напоминание пользователю о том, какая информация требуется при вводе данных о новых клиентах. Это будет происходить с помощью диалогового окна. Чтобы это осуществить

поместите курсор на следующую строку макрокоманд ниже строки **НаЗапись**.

- Выберите макрокоманду **Сообщение**.
- Поместите курсор в поле **Сообщение** в области аргументов макрокоманд.
- Введите следующий текст: *“Не забудьте спросить номер телефона и записать его!”*
- Нажмите клавишу **Tab**.
- Убедитесь, что Ваши макрокоманды стоят в следующей последовательности: **ОткрытьФорму, НаЗапись, Сообщение**.
- Сохраните макрос. При сохранении макроса желательно использовать имена, описывающие, что делает макрос при запуске.
- Введите имя макроса: **Прививка**.
- Щелкните по кнопке **ОК**.
- Теперь убедитесь, что созданный Вами макрос функционирует.
- Нажмите **Выполнить**.

Радел 3. Задания для самостоятельного выполнения

3.1. ТРЕБОВАНИЯ К ВЫПОЛНЕННЫМ ЗАДАНИЯМ:

Требования к таблицам:

1. Таблицы, запросы и фильтры должны полностью соответствовать заданию.
2. Каждая таблица должна содержать как минимум, заданное в квадратных скобках
число записей.
3. Информация, вводимая в таблицы должна содержать осмысленную информацию,
соответствующую заданию.
4. Типы данных полей таблиц должны соответствовать хранимой в них информации.
5. В первичных таблицах поля связи должны иметь тип данных “Счётчик”, а во вторичных таблицах – тип данных “Числовой”.

Требования к формам и отчётам:

6. Для каждой таблицы, должна быть создана ленточная форма.
7. На всех формах для таблиц необходимо отобразить все поля, кроме полей связи в
первичных таблицах (поля, имеющие тип данных «Счётчик»), а поля связи во вторичных таблицах отобразить при помощи «Выпадающих списков» или «Простых списков» (таким образом, вместо кодов связи должны отображаться значения из первичных таблиц, соответствующие этим кодам).
8. На всех формах для таблиц необходимо разместить кнопки перемещения по

записям, добавления и удаления записей, кнопку открытия табличной формы, кнопку закрытия формы и кнопки, отображающие отчёт с той же информацией что и на форме.

9. Для каждой таблицы, запроса и фильтра должны быть созданы табличная форма и отчёт.

10. В верхней части каждой формы, отчёта и страницы, должна быть надпись-заголовок с именем таблицы, запроса или фильтра.

11. На табличных формах должна быть реализована сортировка данных по выбранному полю, поиск информации и фильтрация по заданному полю.

12. На всех табличных формах для запросов и фильтров необходимо отобразить все поля, кроме полей связи.

13. На всех табличных формах необходимо разместить кнопки перемещения по записям, кнопку поиска информации, кнопку закрытия формы и кнопки, отображающие отчёт и страницу с той же информацией что и на форме.

14. Во все отчёты поместить все поля из таблиц запросов и фильтров, кроме полей связи.

15. Поля отдельных записей в отчётах выделить цветом и рамкой.

18. Создать главную кнопочную форму.

19. В верхнюю часть главной кнопочной формы поместить надпись с названием БД.

20. В центре разместить вкладки “Формы” и “Отчёты”.

21. На соответствующие вкладки поместить кнопки для открытия всех форм и отчётов. На кнопках или рядом с ними должны быть поясняющие надписи.

22. Создать форму «О программе».

23. Создать форму «Заставка».

24. В нижней части главной кнопочной формы разместить кнопки для выхода из программы, открытия формы о программе.

25. Сделать форму «Заставка» стартовой.

! Примечание В базе данных должна обеспечиваться автоматически непротиворечивость и целостность данных (контроль целостности данных в схеме данных должен быть включен!).

ЗАДАНИЕ №1: БД АПТЕКА

Таблицы: 1) Сотрудники (Код сотрудника, ФИО, Возраст, Пол, Адрес, Телефон, Паспортные данные, Код должности)[10 записей].

2) Должности (Код должности, Наименование должности, Оклад, Обязанности, Требования)[5 записей].

3) Склад (Код ингредиента, Наименование ингредиента, Дата выпуска, Объём, Срок годности, Стоимость, Поставщик)[10 записей].

4) Лекарство (Код лекарства, Наименование лекарства, Код ингредиента 1, Объём

ингредиента 1, Код ингредиента 2, Объём ингредиента 2, Код ингредиента 3, Объём ингредиента 3, Стоимость, Время приготовления)[10 записей].

5) Заказ (Дата, Время, ФИО заказчика, Телефон, Код лекарства 1, Код лекарства 2, Код лекарства 3, Стоимость, Отметка о выполнении, Код сотрудника)[10 записей].
Запросы: 1) Отдел кадров (Связывает таблицы "Сотрудники" и "Должности" по полю "Код должности").
2) Лекарства (Связывает таблицы "Лекарство" и "Склад" по полям "Код ингредиента", "Код ингредиента 1", "Код ингредиента 2" и "Код ингредиента 3").
3) Заказ (Связывает таблицы "Заказ", "Лекарство" и "Сотрудники" по полям "Код лекарства", "Код лекарства 1", "Код лекарства 2", "Код лекарства 3" и "Код сотрудника").
Фильтры: 1) Фильтры для отображения сотрудников отдельных должностей (На основе запроса "Отдел кадров").
2) Фильтры отображения ингредиентов определённых поставщиков (На основе таблицы "Склад").
3) Фильтры выполненных и невыполненных заказов (На основе запроса "Заказы").

ЗАДАНИЕ №2: БД ПРИВИВКИ

Таблицы: 1) Сотрудники (Код сотрудника, ФИО, Возраст, Пол, Адрес, Телефон, Паспортные данные, Код должности)[10 записей].
2) Должности (Код должности, Наименование должности, Оклад, Обязанности, Требования)[5 записей].
3) Прививки (Код прививки, Наименование прививки, Минимальный срок действия, Код вакцины, Дополнительные условия)[5 записей].
4) Вакцина (Код вакцины, Наименование)[3 записи].
5) Пациенты (ФИО, Адрес, Телефон, Паспортные данные, Дата прививки, Дата окончания действия, Код прививки, Код вакцины, Отметка о прививке, Код сотрудника)[10 записей].
Запросы: 1) Отдел кадров (Связывает таблицы "Сотрудники" и "Должности" по полю "Код должности").
2) Прививки (Связывает таблицы "Прививки" и "Вакцина" по полю "Код вакцины").
3) Пациенты (Связывает таблицы "Пациенты", "Прививки" и "Сотрудники" по полям "Код прививки" и "Код сотрудника").
Фильтры: 1) Фильтры для отображения сотрудников отдельных должностей (На основе запроса "Отдел кадров").
2) Фильтры для отображения прививок по вакцинам (На основе запроса

"Прививки").

3) Фильтры для отображения пациентов с отдельными прививками (На основе запроса "Прививки").

4) Фильтры для отображения пациентов с истекшими и неистекшими сроками действия прививок (На основе запроса "Пациенты").

5) Вывести данные о расходе доз вакцин.

ЗАДАНИЕ №3: БД БОЛЬНИЦЫ.

Таблицы: 1) Сотрудники (Код сотрудника, ФИО, Возраст, Пол, Адрес, Телефон, Паспортные данные, Код должности)[10 записей].

2) Должности (Код должности, Наименование должности, Оклад, Обязанности, Требования)[5 записей].

3) Лекарства (Код лекарства, Наименование, Показания, Противопоказания, Упаковка, Стоимость)[5 записей].

4) Болезни (Код болезни, Наименование, Симптомы, Продолжительность, Последствия, Код лекарства 1, Код лекарства 2, Код лекарства 3)[10 записей].

5) Пациенты (ФИО пациента, Возраст, Пол, Адрес, Телефон, Дата обращения, Код болезни, Код сотрудника, Результат лечения)[10 записей].

Запросы: 1) Отдел кадров (Связывает таблицы "Сотрудники" и "Должности" по полю

"Код должности").

2) Болезни (Связывает таблицы "Болезни" и "Лекарства" по полю "Код лекарства", "Код лекарства 1", "Код лекарства 2" и "Код лекарства 3").

3) Пациенты (Связывает таблицы "Пациенты", "Болезни" и "Сотрудники" по полям "Код болезни" и "Код сотрудника").

Фильтры: 1) Фильтры для отображения сотрудников отдельных должностей (На основе запроса "Отдел кадров").

2) Фильтры для отображения болезней с одинаковыми симптомами (На основе запроса "Болезни").

3) Фильтры для отображения пациентов с одинаковыми болезнями (На основе запроса "Пациенты").

ЗАДАНИЕ №4: БД РАЗМЕЩЕНИЕ БОЛЬНЫХ.

Таблицы: 1) Сотрудники (Код сотрудника, ФИО, Возраст, Пол, Адрес, Телефон, Паспортные данные, Код должности)[10 записей].

2) Должности (Код должности, Наименование должности, Оклад, Обязанности, Требования)[5 записей].

3) Палаты (Код палаты, Наименование, Вместимость, Описание, Код сотрудника)[5 записей].

4) Заболевания (Код заболевания, Наименование, Описание)[5 записей].

5) Пациенты (ФИО, Паспортные данные, Дата размещения, Дата убытия, Код палаты, Код услуги 1, Код услуги 2, Код услуги 3, Код сотрудника)

Запросы: 1) Отдел кадров (Связывает таблицы "Сотрудники" и "Должности" по полю "Код должности").

2) Палаты (Связывает таблицы "Сотрудники" и "Палаты" по полю "Код сотрудника").

3) Пациенты (Связывает таблицы "Пациенты", "Палаты", "Заболевания" и "Сотрудники"

по полям "Код палаты", "Код заболевания", "Код услуги 1", "Код услуги 2", "Код

услуги 3" и "Код сотрудника").

Фильтры: 1) Фильтры для отображения сотрудников отдельных должностей (На основе

запроса "Отдел кадров").

2) Фильтры для отображения пациентов в разных палатах (На основе запроса "Пациенты").

3) Вывести палаты различной вместимости (На основе запроса "Палаты").

4) Вывести данные по загруженности палат.

ЗАДАНИЕ №5: БД БИБЛИОТЕКА МЕДУНИВЕРСИТЕТА.

Таблицы: 1) Сотрудники (Код сотрудника, ФИО, Возраст, Пол, Адрес, Телефон, Паспортные данные, Код должности)[10 записей].

2) Должности (Код должности, Наименование должности, Оклад, Обязанности, Требования)[5 записей].

3) Издательства (Код издательства, Наименование, Город, Адрес)[5 записей].

4) Жанры (Код жанра, Наименование, Описание) [5 записей].

5) Книги (Код книги, Наименование, Автор, Код издательства, Год издания, Код

жанра) [10 записей].

6) Читатели (Код читателя, ФИО, Дата рождения, Пол, Адрес, Телефон, Паспортные данные) [10 записей].

7) Выданные книги (Код книги, Код читателя, Дата выдачи, Дата возврата, Отметка о возврате, Код сотрудника) [10 записей].

Запросы: 1) Отдел кадров (Связывает таблицы "Сотрудники" и "Должности" по полю "Код должности").

"Код должности").

2) Каталог (Связывает таблицы "Книги", "Издательства" и "Жанры" по полям "Код издательства" и "Код жанра").

3) Книги на руках (Связывает таблицы "Выданные книги", "Книги", "Читатели" и "Сотрудники" по полям "Код книги", "Код читателя" и "Код сотрудника")

Фильтры: 1) Фильтры для отображения сотрудников отдельных должностей (На основе

запроса "Отдел кадров").

2) Фильтры для отображения книг отдельных издательств (На основе запроса "Каталог").

- 3) Фильтры для отображения книг отдельных авторов (На основе запроса "Каталог").
- 4) Фильтры для отображения книг отдельных годов издания (На основе запроса "Каталог").
- 5) Фильтры для отображения сданных и не сданных книг (На основе запроса "Книги на руках").
- 6) Фильтры для отображения книг на руках отдельных читателей (На основе запроса "Книги на руках").

ЗАДАНИЕ №6: БД АВТОПАРК СКОРОЙ ПОМОЩИ.

Таблицы: 1) Сотрудники (Код сотрудника, ФИО, Возраст, Пол, Адрес, Телефон, Паспортные данные, Код должности)[10 записей].

2) Должности (Код должности, Наименование должности, Оклад, Обязанности, Требования)[5 записей].

3) Марки (Код марки, Наименование, Технические характеристики, Стоимость, Специфика)[5 записей].

4) Тарифы (Код тарифа, Наименование, Описание, Стоимость)[5 записей].

5) Дополнительные услуги (Код услуги, Наименование, Описание услуги, Стоимость)[5 записей].

6) Автомобили (Код автомобиля, Код марки, Регистрационный номер, Номер кузова, Номер двигателя, Год выпуска, Пробег, Код сотрудника-шофёра, Дата последнего ТО, Код сотрудника-механика, Специальные отметки)[10 записей].

7) Вызовы (Дата, Время, Телефон, Откуда, Куда, Код тарифа, Код услуги, Код автомобиля, Код сотрудника-оператора)[10 записей].

Запросы: 1) Отдел кадров (Связывает таблицы "Сотрудники" и "Должности" по полю "Код должности").

2) Автопарк (Связывает таблицы "Автомобили", "Марки" и "Сотрудники" по полю "Код марки" и "Код сотрудника").

3) Список вызовов (Связывает таблицы "Вызовы", "Тарифы", "Услуги", "Автомобили" и "Сотрудники" по полю "Код тарифа", "Код услуги", "Код автомобиля" и "Код сотрудника-диспетчера").

Фильтры: 1) Фильтры для отображения сотрудников отдельных должностей (На основе запроса "Отдел кадров").

2) Фильтры для отображения автомобилей отдельных годов выпуска (На основе запроса "Автопарк").

3) Фильтры для отображения автомобилей отдельных марок (На основе запроса "Автопарк").

4) Фильтры для отображения вызовов по отдельным тарифам (На основе запроса "Список вызовов").

5) Фильтры для отображения вызовов по отдельным датам (На основе запроса

“Список вызовов”).

ЗАДАНИЕ №7: БД САНАТОРНОЕ ЛЕЧЕНИЕ.

Таблицы: 1) Сотрудники (Код сотрудника, ФИО, Возраст, Пол, Адрес, Телефон, Паспортные данные, Код должности)[10 записей].

2) Должности (Код должности, Наименование должности, Оклад, Обязанности, Требования)[5 записей].

3) Виды отдыха (Код вида, Наименование, Описание, Ограничения)[5 записей].

4) Санатории (Код санатория, Наименование, Страна, Город, Адрес, Телефон, профиль, Контактное лицо)[10 записей].

5) Дополнительные услуги (Код услуги, Наименование, Описание, Цена) [5 записей].

6) Клиенты (Код клиента, ФИО, Дата рождения, Пол, Адрес, Телефон, Паспортные данные)[5 записей].

7) Путёвки (Дата начала, Дата окончания, Продолжительность, Код санатория, Код

вида, Код услуги 1, Код услуги 2, Код услуги 3, Код клиента, Код сотрудника, Отметка о бронировании, Отметка об оплате)[10 записей].

Запросы: 1) Отдел кадров (Связывает таблицы "Сотрудники" и "Должности" по полю

"Код должности").

2) Список путёвок (Связывает таблицы "Путёвки", "Санатории", "Виды отдыха",

"Дополнительные услуги", "Клиенты" и "Сотрудники" по полям "Код санатория",

"Код вида", "Код услуги", "Код услуги 1", "Код услуги 2", "Код услуги 3", "Код клиента" и "Код сотрудника").

Фильтры: 1) Фильтры для отображения сотрудников отдельных должностей (На основе

запроса "Отдел кадров").

2) Фильтры для отображения путёвок по отдельным видам отдыха (На основе запроса "Список путёвок").

3) Фильтры для отображения путёвок в отдельные страны (На основе запроса "Список путёвок").

4) Фильтры для отображения путёвок в отдельные санатории (На основе запроса "Список путёвок").

5) Фильтры для отображения забронированных и не забронированных путёвок (На основе запроса "Список путёвок").

6) Фильтры для отображения оплаченных и не оплаченных путёвок (На основе запроса "Список путёвок").

7) Фильтры для отображения заказанных и не заказанных путёвок (На основе запроса "Список путёвок").

ЗАДАНИЕ №8: БД КОМПАНИЯ МЕДСТРАХОВАНИЯ.

Таблицы: 1) Сотрудники (Код сотрудника, ФИО, Возраст, Пол, Адрес, Телефон, Паспортные данные, Код должности)[10 записей].

2) Должности (Код должности, Наименование должности, Оклад, Обязанности, Требования)[5 записей].

3) Риски (Код риска, Наименование, Описание, Средняя вероятность)[5 записей].

4) Виды полисов (Код вида полиса, Наименование, Описание, Условия, Код риска 1, Код риска 2, Код риска 3)[5 записей].

5) Группы клиентов (Код группы, Наименование, Описание)[5 записей].

6) Клиенты (Код клиента, ФИО, Дата рождения, Пол, Адрес, Телефон, Паспортные данные, Код группы)[10 записей].

7) Полисы (Номер полиса, Дата начала, Дата окончания, Стоимость, Сумма выплаты, Код вида полиса, Отметка о выплате, Отметка об окончании, Код клиента, Код сотрудника)[10 записей].

Запросы: 1) Отдел кадров (Связывает таблицы "Сотрудники" и "Должности" по полю

"Код должности").

2) Риски полисов (Связывает таблицы "Виды полисов" и "Риски" по полям "Код риска", "Код риска 1", "Код риска 2", "Код риска 3").

3) Список клиентов (Связывает таблицы "Клиенты" и "Группы клиентов" по полю "Код группы").

4) Список полисов (Связывает таблицы "Полисы", "Виды полисов", "Клиенты" и "Сотрудники" по полям "Код вида полиса", "Код клиента" и "Код сотрудника").

Фильтры: 1) Фильтры для отображения сотрудников отдельных должностей (На основе запроса "Отдел кадров").

2) Фильтры для отображения клиентов отдельных групп (На основе запроса "Список клиентов").

3) Фильтры для отображения полисов отдельных видов (На основе запроса "Список полисов").

4) Фильтры для отображения полисов по которым производились выплаты и по которым не производились выплаты (На основе запроса "Список полисов").

5) Фильтры для оконченных и неоконченных полисов (На основе запроса "Список полисов").

ЗАДАНИЕ №9: БД МЕДПОМОЩЬ.

Таблицы: 1) Сотрудники (Код сотрудника, ФИО, Возраст, Пол, Адрес, Телефон, Паспортные данные, Код должности)[10 записей].

2) Должности (Код должности, Наименование должности, Оклад, Обязанности, Требования)[5 записей].

- 3) Заболевания (Код заболевания, Наименование, Описание)[5 записей].
 - 4) Обращения (Код обращения, Наименование, Описание)[5 записей].
 - 5) Категории пациентов (Код категории, Наименование, Замечания)[5 записей].
 - 6) Дополнительные услуги (Код услуги, Наименование, Описание, Цена)[5 записей].
 - 7) Пациенты (Код пациента, ФИО, Пол, Дата рождения, Возраст, Рост, Вес, Количество детей, Семейное положение, Вредные привычки, Описание, Код заболевания Код обращения, Код категории, Адрес, Телефон, Паспортные данные, Информация о) [10 записей].
 - 8) Услуги (Код пациента, Дата, Код услуги 1, Код услуги 2, Код услуги 3, Стоимость, Код сотрудника)[10 записей].
- Запросы: 1) Отдел кадров (Связывает таблицы "Сотрудники" и "Должности" по полю "Код должности").
- 2) Список пациентов (Связывает таблицы "Пациентов", "Заболевания", "обращения" и "Категории пациентов" по полям "Код заболевания", "Код обращения" и "Код категории").
 - 3) Список обращений (Связывает таблицы "Обращения", "Пациенты", "Дополнительные услуги" и "Сотрудники" по полям "Код пациента", "Код услуги", "Код услуги 1", "Код услуги 2", "Код услуги 3" и "Код сотрудника").
- Фильтры: 1) Фильтры для отображения сотрудников отдельных должностей (На основе запроса "Отдел кадров").
- 2) Фильтры для отображения пациентов по заболеваниям (На основе запроса "Список пациентов").
 - 3) Фильтры для отображения пациентов по обращениям (На основе запроса "Список пациентов").
 - 4) Фильтры для отображения клиентов отдельных категорий (На основе запроса "Список пациентов").
 - 5) Фильтры для отображения клиентов по вредным привычкам (На основе запроса "Список пациентов").
 - 6) Фильтры для отображения пациентов по семейному положению (На основе запроса "Список пациентов").

ЗАДАНИЕ №10: БД СЕРВИС-ЦЕНТР МЕДОБОРУДОВАНИЯ.

- Таблицы: 1) Сотрудники (Код сотрудника, ФИО, Возраст, Пол, Адрес, Телефон, Паспортные данные, Код должности)[10 записей].
- 2) Должности (Код должности, Наименование должности, Оклад, Обязанности, Требования)[5 записей].
 - 3) Запчасти (Код запчасти, Наименование, Функции, Цена)[5 записей].
 - 4) Ремонтируемые модели (Код модели, Наименование, Тип, Производитель, Технические характеристики, Особенности)[5 записей].

5) Виды неисправностей (Код вида, Код модели, Описание, Симптомы, Методы ремонта, Код запчасти 1, Код запчасти 2, Код запчасти 3, Цена работы)[5 записей].

6) Обслуживаемые объекты (Код объекта, Наименование, Адрес, Телефон)[5 записей].

7) Заказы (Дата заказа, Дата возврата, ФИО заказчика, Серийный номер, Код вида неисправности, Код объекта, Отметка о гарантии, Срок гарантии ремонта, Цена, Код сотрудника)[10 записей].

Запросы: 1) Отдел кадров (Связывает таблицы "Сотрудники" и "Должности" по полю "Код должности").

2) Список неисправностей (Связывает таблицы "Виды неисправностей", "Ремонтируемые модели" и "Запчасти" по полям "Код модели", "Код запчасти", "Код запчасти 1", "Код запчасти 2", "Код запчасти 3").

3) Список заказов (Связывает таблицы "Заказы", "Виды неисправностей", "Обслуживаемые магазины" и "Сотрудники" по полям "Код вида неисправности", "Код объекта" и "Код сотрудника").

Фильтры: 1) Фильтры для отображения сотрудников отдельных должностей (На основе запроса "Отдел кадров").

2) Фильтры для отображения видов неисправностей отдельных моделей (На основе запроса "Список неисправностей").

3) Фильтры для отображения видов неисправностей отдельных типов устройств (На основе запроса "Список неисправностей").

4) Фильтры для отображения видов неисправностей моделей отдельных производителей (На основе запроса "Список неисправностей").

5) Фильтры для отображения заказов отдельных объектов (На основе запроса "Список заказов").

6) Фильтры для отображения заказов отдельных неисправностей (На основе запроса "Список заказов").

7) Фильтры для отображения гарантийных и не гарантийных заказов (На основе запроса "Список заказов").

ЗАДАНИЕ №11: БД МЕДУНИВЕРСИТЕТ.

Таблицы: 1) Сотрудники (Код сотрудника, ФИО, Возраст, Пол, Адрес, Телефон, Паспортные данные, Код должности)[10 записей].

2) Должности (Код должности, Наименование должности, Оклад, Обязанности, Требования)[5 записей].

3) Студенты (ФИО, Дата рождения, Пол, Адрес, ФИО отца, ФИО матери, Код группы, Дополнительная информация) [10 записей].

4) Группы (Код группы, Код сотрудника-на, Код вида, Количество учеников, Буква, Год обучения, Год создания)[5 записей].

5) Виды групп (Код вида, Наименование, Описание)[5 записей].

- 6) Дисциплины (Код дисциплины, Наименование, Описание, Код сотрудника-преподавателя)[10 записей].
- 7) Расписание (Дата, День недели, Код группы, Код дисциплины, Время начала, Время окончания)[10 записей].
- Запросы: 1) Отдел кадров (Связывает таблицы "Сотрудники" и "Должности" по полю "Код должности").
- 2) Список студентов (Связывает таблицы "Студенты" и "Группы" по полю "Группы").
- 3) Список групп(Связывает таблицы "Группы", "Виды групп" и "Сотрудники" по полям "Код вида" и "Код сотрудника").
- 4) Список дисциплин (Связывает таблицы "дисциплины" и "Сотрудники" по полю "Код сотрудника").
- 5) Расписание занятий (Связывает таблицы "Расписание", "Группы" и "Дисциплины" по полям "Код группы" и "Код дисциплины").
- Фильтры: 1) Фильтры для отображения сотрудников отдельных должностей (На основе запроса "Отдел кадров").
- 2) Фильтры для отображения групп различных годов обучения (На основе запроса "Список групп").
- 3) Фильтры для отображения расписания для отдельных групп и дат (На основе запроса "Расписание занятий").
- 4) Фильтры для отображения отдельных видов групп (На основе запроса "Список групп").
- 5) Фильтры для отображения студентов отдельных групп (На основе запроса "Список студентов ").
- 6) Фильтры для отображения дисциплин отдельных преподавателей (На основе запроса "Список дисциплин").

ЗАДАНИЕ №12: БД ДОСТАВКА ЛЕКАРСТВ.

- Таблицы: 1) Сотрудники (Код сотрудника, ФИО, Возраст, Пол, Адрес, Телефон, Паспортные данные, Код должности)[10 записей].
- 2) Должности (Код должности, Наименование должности, Оклад, Обязанности, Требования)[5 записей].
- 3) Виды автомобилей (Код вида автомобиля, Наименование, Описание)[5 записей].
- 4) Марки автомобилей (Код марки, Наименование, Технические характеристики, Описание) [5 записей].
- 5) Виды грузов (Код вида груза, Наименование, Код вида автомобиля для транспортировки, Описание)[5 записей].
- 6) Грузы (Код груза, Наименование, Код вида груза, Срок годности,

Особенности)[5 записей].

7) Автомобили (Код автомобиля, Код марки, Код вида автомобиля, Регистрационный номер, Номер кузова, номер двигателя, Год выпуска, Код сотрудника-водителя, Дата последнего ТО, Код сотрудника-механика)[5 записей].

8) Рейсы (Код автомобиля, Заказчик, Откуда, Куда, Дата отправления, Дата прибытия, Код груза, Цена, Отметка об оплате, Отметка о возвращении, Код сотрудника)[10 записей].

Запросы: 1) Отдел кадров (Связывает таблицы "Сотрудники" и "Должности" по полю "Код должности").

2) Транспортировка (Связывает таблицы "Виды грузов" и "Виды автомобилей" по полю "Код вида автомобиля").

3) Перевозимые грузы (Связывает таблицы "Грузы" и "Виды грузов" по полю "Код вида груза").

4) Автопарк (Связывает таблицы "Автомобили", "Марки автомобилей", "Виды автомобилей" и "Сотрудники" по полям "Код марки", "Код вида автомобиля" и "Код сотрудника").

5) Заказы (Связывает таблицы "Рейсы", "Автомобили", "Грузы" и "Сотрудники"

по полям "Код автомобиля", "Код груза" и "Код сотрудника").

Фильтры: 1) Фильтры для отображения сотрудников отдельных должностей (На основе запроса "Отдел кадров").

2) Фильтры для отображения отдельных видов грузов (На основе запроса "Перевозимые грузы").

3) Фильтры для отображения отдельных видов автомобилей (На основе запроса "Автопарк").

4) Фильтры для отображения заказов по перевозке отдельных грузов (На основе запроса "Заказы").

5) Фильтры для отображения заказов отдельных заказчиков (На основе запроса "Заказы").

6) Фильтры для отображения оплаченных и не оплаченных заказов (На основе запроса "Заказы").

7) Фильтры о вернувшихся и не вернувшихся из рейса автомобилей (На основе запроса "Заказы").

ЗАДАНИЕ №13: БД ОПТОВОГО СКЛАДА ЛЕКАРСТВ.

Таблицы: 1) Сотрудники (Код сотрудника, ФИО, Возраст, Пол, Адрес, Телефон, Паспортные данные, Код должности)[10 записей].

2) Должности (Код должности, Наименование должности, Оклад, Обязанности, Требования)[5 записей].

3) Товары (Код товара, Код типа, Производитель, Наименование, Условия

хранения, Упаковка, Срок годности) [10 записей].

4) Типы товаров (Код типа, Наименование, Описание, Особенности) [5 записей].

5) Поставщики (Код поставщика, Наименование, Адрес, Телефон, Код поставляемого товара 1, Код поставляемого товара 2, Код поставляемого товара 3) [5 записей].

6) Заказчики (Код заказчика, Наименование, Адрес, Телефон, Код потребляемого товара 1, Код потребляемого товара 2, Код потребляемого товара 3) [5 записей].

7) Склад (Дата поступления, Дата заказа, Дата отправки, Код товара, Код поставщика, Код заказчика, Способ доставки, Объём, Цена, Код сотрудника) [10 записей].

Запросы: 1) Отдел кадров (Связывает таблицы "Сотрудники" и "Должности" по полю "Код должности").

2) Список товаров (Связывает таблицы "Товары" и "Типы товаров" по полю "Код типа").

3) Список поставщиков (Связывает таблицы "Поставщики" и "Товары" по полям

"Код товара", "Код поставляемого товара 1", "Код поставляемого товара 2" и "Код поставляемого товара 3").

4) Список заказчиков (Связывает таблицы "Заказчики" и "Товары" по полям "Код товара", "Код потребляемого товара 1", "Код потребляемого товара 2" и "Код потребляемого товара 3").

5) Заказы (Связывает таблицы "Склад", "Товары", "Поставщики", "Заказчики" и "Сотрудники" по полям "Код товара", "Код поставщика", "Код заказчика" и "Код сотрудника").

Фильтры: 1) Фильтры для отображения сотрудников отдельных должностей (На основе запроса "Отдел кадров").

2) Фильтры для отображения отдельных типов товаров (На основе запроса "Список товаров").

3) Фильтры товаров отдельных поставщиков (На основе запроса "Заказы").

4) Фильтры товаров отдельных заказчиков (На основе запроса "Заказы").

5) Фильтры товаров по отдельным способам доставки (На основе запроса "Заказы").

ЗАДАНИЕ №14: БД СКОРАЯ ПОМОЩЬ.

Таблицы: 1) Сотрудники (Код сотрудника, ФИО, Возраст, Пол, Адрес, Телефон, Паспортные данные, Код должности)[10 записей].

2) Должности (Код должности, Наименование должности, Оклад, Обязанности, Требования)[5 записей].

- 3) Виды работ (Код вида, Наименование, Описание, Цена работы, Код материала
1, Код материала 2, Код материала 3)[5 записей].
- 4) Материалы (Код материала, Наименование, Упаковка, Описание, Цена) [5 записей].
- 5) Бригады (Код бригады , Код сотрудника 1, Код сотрудника 2, Код сотрудника
3) [5 записей].
- 6) Пациенты (Код пациента, ФИО, Адрес, Телефон, Паспортные данные)[5 записей].
- 7) Вызовы (Код вызова, Код вида работ, Код бригады, Содержание работ, Время начала,
Время окончания, Отметка о завершении, Ложный вызов, Код сотрудника) [10 записей].
- Запросы: 1) Отдел кадров (Связывает таблицы "Сотрудники" и "Должности" по полю
"Код должности").
- 2) Список работ (Связывает таблицы "Виды работ" и "Материалы" по полям
"Код материала", "Код материала 1", "Код материала 2" и "Код материала 3").
- 3) Список бригад (Связывает таблицы "Бригады" и "Сотрудники" по полям
"Код
сотрудника", "Код сотрудника 1", "Код сотрудника 2" и "Код сотрудника 3").
- 4) Список вызовов (Связывает таблицы "Вызовы", "Виды работ", "Бригады" и
"Сотрудники" по полям "Код вида", "Код бригады" и "Код сотрудника").
- Фильтры: 1) Фильтры для отображения сотрудников отдельных должностей
(На основе
запроса "Отдел кадров").
- 2) Фильтры для отображения отдельных видов работ (На основе запроса
"Список работ").
- 3) Фильтры вызовов на конкретные работы (На основе запроса "Список
вызовов").
- 4) Фильтры для отображения вызовов отдельных пациентов (На основе запроса
"Список вызовов").
- 5) Фильтры на вызовов, выполняемые отдельными бригадами (На основе
запроса "Список вызовов").
- 6) Фильтры для завершённых и не завершённых работ по вызовам (На основе
запроса
"Список вызовов").
- 7) Фильтры для исполненных и ложных вызовов (На основе запроса
"Список вызовов").

ЗАДАНИЕ №15: БД СТОМАТОЛОГИЯ.

Таблицы: 1) Сотрудники-руководители (Код сотрудника-руководителя, ФИО, Возраст, Пол, Адрес, Телефон, Паспортные данные, Код должности)[10 записей].

2) Должности (Код должности, Наименование должности, Оклад, Обязанности, Требования)[5 записей].

3) Виды услуг (Код вида услуги, Наименование, Описание, Цена)[5 записей].

4) Виды работ (Код вида, Наименование, Описание)[5 записей].

5) Врачи (Код врача, ФИО, Пол, Дата рождения, Адрес проживания, Телефон, Паспортные данные, Код вида работ, Цена, Дополнительная информация)[10 записей].

6) Пациенты (Код пациента, ФИО, Пол, Дата рождения, Адрес проживания, Телефон, Паспортные данные, Код вида работ, Количество обращений, Отметка о выполнении, Оплата, Дополнительные пожелания)[10 записей].

7) Договоры (Дата заключения, Код продавца, Код пациента, Сумма, Стоимость услуг, Код вида услуги, Код сотрудника-руководителя)[10 записей].

Запросы: 1) Отдел кадров (Связывает таблицы "Сотрудники" и "Должности" по полю

"Код должности").

2) Работы (Связывает таблицы "Врачи" и "Виды работ" по полю "Код вида работ").

3) Обслуживание (Связывает таблицы "Пациенты" и "Виды работ" по полю "Код вида работ").

4) Заключённые договора (Связывает таблицы "Договоры", "Врачи", "Пациенты", "Услуги" и "Сотрудники-руководители" по полям "Код врача", "Код пациента", "Код услуги" и "Код сотрудника-руководителя").

Фильтры: 1) Фильтры для отображения сотрудников отдельных должностей (На основе запроса "Отдел кадров").

2) Фильтры продаваемых работ различных видов (На основе запроса "Работы").

3) Фильтры покупаемого обслуживания различных видов (На основе запроса "Обслуживание").

4) Фильтры договоров, заключённых отдельными сотрудниками-руководителями (На основе запроса "Заключённые договора").

ЗАДАНИЕ №16: БД КОМПЬЮТЕРНОЕ ОБСЛУЖИВАНИЕ ПОЛИКЛИНИКИ

Таблицы: 1) Сотрудники (Код сотрудника, ФИО, Возраст, Пол, Адрес, Телефон, Паспортные данные, Код должности)[10 записей].

2) Должности (Код должности, Наименование должности, Оклад, Обязанности, Требования)[5 записей].

3) Виды комплектующих (Код вида, Наименование, Описание)[15 записей].

4) Комплектующие (Код комплектующего, Код вида, Марка, Фирма производитель, Страна производитель, Дата выпуска, Характеристики, Срок гарантия, Описание, Цена)[15 записей].

5) Заказчики (Код заказчика, ФИО, Адрес, Телефон)[10 записей].

6) Услуги (Код услуги, Наименование, Описание, Стоимость)[5 записей].

7) Заказы (Дата заказа, Дата исполнения, Код заказчика, Код комплектующего 1,

Код комплектующего 2, Код комплектующего 3, Доля предоплаты, Отметка об оплате, Отметка об исполнении, Общая стоимость, Срок общей гарантии, Код услуги 1, Код услуги 2, Код услуги 3, Код сотрудника)[10 записей].

Запросы: 1) Отдел кадров (Связывает таблицы "Сотрудники" и "Должности" по полю

"Код должности").

2) Список комплектующих (Связывает таблицы "Комплектующие" и "Виды комплектующих" по полю "Код вида").

3) Список заказов (Связывает таблицы "Заказы", "Заказчики", "Комплектующие", "Услуги" и "Сотрудники" по полям "Код заказчика", "Код комплектующего", "Код комплектующего 1", "Код комплектующего 2", "Код комплектующего 3", "Код услуги", "Код услуги 1", "Код услуги 2", "Код услуги 3" и "Код сотрудника").

Фильтры: 1) Фильтры для отображения сотрудников отдельных должностей (На основе запроса "Отдел кадров").

2) Фильтр для отображения комплектующих отдельных видов (На основе запроса "Список комплектующих").

3) Фильтры для отображения заказов отдельных заказчиков (На основе запроса "Список заказов").

4) 3) Фильтры для отображения заказов по датам заказа (На основе запроса "Список заказов").

Литература

І. Издания, доступные в ЭБС «Консультант студента»:

1. Зарубина Т.В., Медицинская информатика [Электронный ресурс]: учебник / Зарубина Т.В. [и др.] - М. : ГЭОТАР-Медиа, 2018. - 512 с. - ISBN 978-5-9704-4573-0 - Режим доступа:
<http://www.studentlibrary.ru/book/ISBN9785970445730.html>
2. Толстяков Р.Р., Информатика [Электронный ресурс]: учеб. пособие / Р.Р. Толстяков, Т.Ю. Забавникова, Т.В. Попова - М. : ФЛИНТА, 2019. - 112 с. - ISBN 978-5-9765-1593-2 - Режим доступа:
<http://www.studentlibrary.ru/book/ISBN9785976515932.html>
3. Когаловский М.Р., Перспективные технологии информационных систем [Электронный ресурс] / М.Р. Когаловский - М. : ДМК Пресс, 2018. - 287 с. - ISBN 978-5-93700-042-2 - Режим доступа:
<http://www.studentlibrary.ru/book/ISBN9785937000422.html>
4. Хроленко А.Т., Современные информационные технологии для гуманитария [Электронный ресурс] / Хроленко А.Т. - М. : ФЛИНТА, 2018. - 128 с. - ISBN 978-5-9765-0023-5 - Режим доступа:
<http://www.studentlibrary.ru/book/ISBN9785976500235.html>
5. Харитонов Е.А., Теоретические и практические вопросы дисциплины "Информатика" [Электронный ресурс]: учебное пособие / Харитонов Е. А. - Казань : Издательство КНИТУ, 2017. - 140 с. - ISBN 978-5-7882-2108-3 - Режим доступа: <http://www.studentlibrary.ru/book/ISBN9785788221083.html>
6. Омельченко В.П., Информатика [Электронный ресурс]: учебник / Омельченко В.П., Демидова А.А. - М. : ГЭОТАР-Медиа, 2017. - 384 с. - ISBN 978-5-9704-4633-1 - Режим доступа:
<http://www.studentlibrary.ru/book/ISBN9785970446331.html>
7. Кравченко Ю.А., Информационные и программные технологии. Часть 1. Информационные технологии [Электронный ресурс]: учебное пособие / Кравченко Ю. А. - Ростов н/Д : Изд-во ЮФУ, 2017. - 112 с. - ISBN 978-5-9275-2495-2 - Режим доступа:
<http://www.studentlibrary.ru/book/ISBN9785927524952.html>
8. Тушко Т.А., Информатика [Электронный ресурс]: учеб. пособие / Т.А. Тушко, Т.М. Пестунова - Красноярск : СФУ, 2017. - 204 с. - ISBN 978-5-7638-3604-2 - Режим доступа: <http://www.studentlibrary.ru/book/ISBN9785763836042.html>
9. Березовская Е.А., Имитационное моделирование [Электронный ресурс]: учеб. пособие / Березовская Е. А. - Ростов н/Д : Изд-во ЮФУ, 2018. - 76 с. - ISBN 978-5-9275-2426-6 - Режим доступа:
<http://www.studentlibrary.ru/book/ISBN9785927524266.html>
10. Парфенова Е.В., Информационные технологии [Электронный ресурс] Электронное издание на основе: Информационные технологии: лаб. практикум / Е.В. Парфенова - М. : МИСиС, 2018. - 56 с. - Режим доступа:

II. Другие издания:

11. Дейт, К. Дж. Введение в системы баз данных / К. Дж. Дейт. - 8-е изд.: [пер. с англ.]. - М.: Вильямс, 2005, - 1328 с.
12. Коннолли, Т. Базы данных: проектирование, реализация и сопровождение. Теория и практика / Т. Коннолли, К. Бегг, А. Страчан: 2-е изд. : [пер. с англ.]. - М.: Вильямс, 2000. - 1120 с.
13. Джексон, Г. Проектирование реляционных баз данных для использования с микроЭВМ: [пер. с англ.] / Г. Джексон. - М. : Мир, 1991. - 252 с.

ПРИЛОЖЕНИЯ

ПРИЛОЖЕНИЕ 1.

ВОПРОСЫ КОНТРОЛЬНЫХ ТЕСТОВ

1. В чем отличие базы данных от базы знаний?
2. Информационная система это
3. Что такое ссылочная целостность данных?
4. К какому классу информационных систем относится каталог в библиотеке?
5. К какому классу информационных систем относится сеть Интернет?
6. К какому классу информационных систем относится программа Microsoft Excel?
7. Что такое «плоские базы данных»?
8. Где в БД хранятся данные?
9. Где в БД можно изменять данные?
10. Запись это
11. Поле это
12. Что НЕ является характеристикой поля?
13. Ключевое поле это
14. Индексное поле это
15. Какое из ниже перечисленных утверждений НЕ верно?
16. Тип данных это
17. Значение поля это
18. В таблице имеется поле "ФИО". Какой тип данных необходимо задать для этого поля?
19. Запросы предназначены для
20. Фильтры предназначены для
21. Чем отличается запрос от фильтра?
22. Существует запрос "Отдел кадров", связывающий таблицы "Сотрудники" и "Должности". Какая из таблиц в запросе является первичной?
23. Какому условию должны удовлетворять поля связи в запросе?
24. Какого вида связи НЕ существует в запросе?
25. Запросы с какими видами связей называют простыми?
26. Если в БД есть хотя бы две связанные таблицы то она называется
27. Каких видов полей НЕ существует?
28. Таблицы запросы и фильтры образуют....
29. Формы предназначены для....
30. Формы могут отображать содержимое
31. Формы состоят из
32. Форма с кнопками, появляющаяся первой после запуска БД это

- 33. Форма с кнопками для открытия всех других форм это
- 34. Форма заставки это
- 35. Форма, отображающая таблицу это
- 36. Какого вида форм НЕ существует в БД?
- 37. Отчёты предназначены для
- 38. Где в отчёте располагаются номера страниц?
- 39. Без чего НЕ может быть создан отчёт?
- 40. Какой объект НЕЛЬЗЯ поместить в отчёт?

ПРИЛОЖЕНИЕ 2.
СНИМКИ ЭКРАНОВ НЕКОТОРЫХ ВЫПОЛНЕННЫХ ЗАДАНИЙ

Сервис-Центер медоборудования

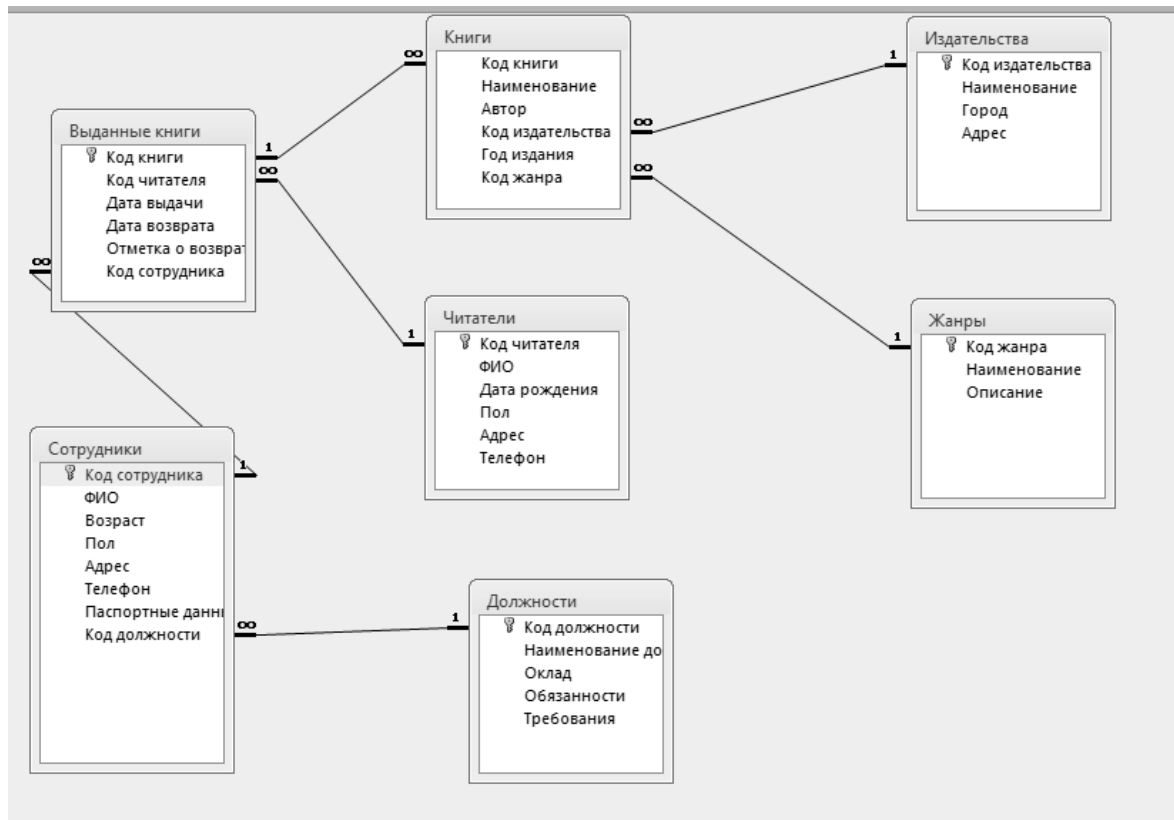
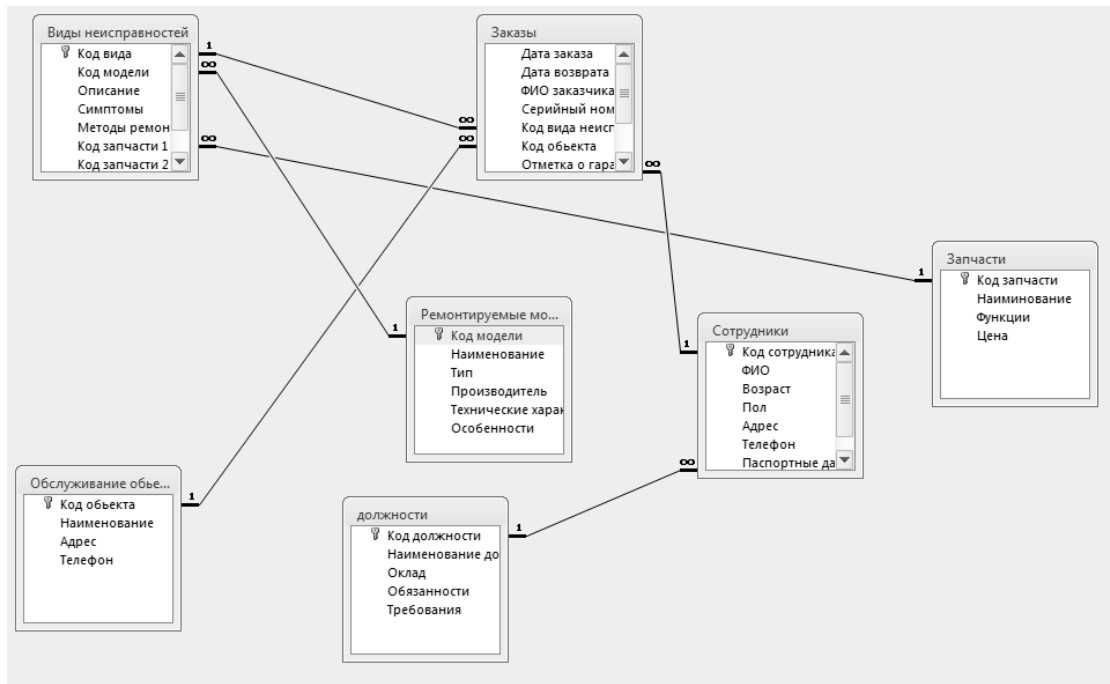


Выйти

О программе

Главная кнопочная





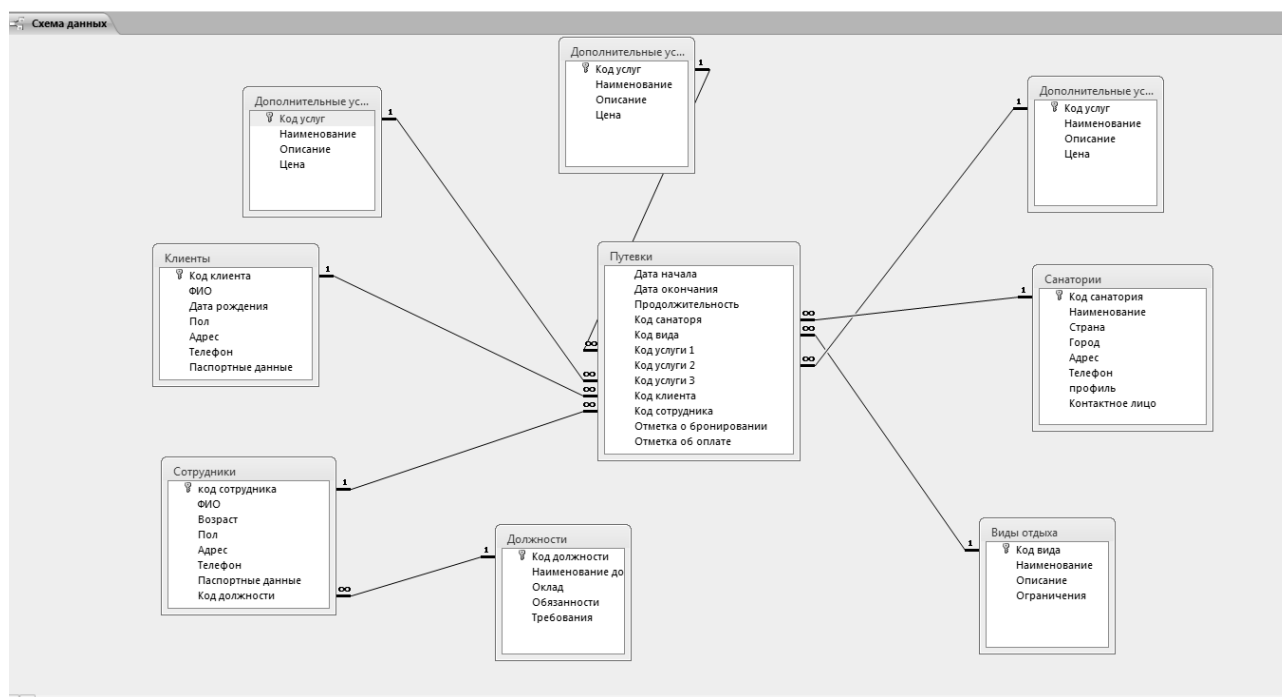
Заболевания

Код	Заболевания	Наименование	Описание
1	11	Корь	Острая вирусная инфекция
2	12	Краснуха	Эпидемическое вирусное заболевание
3	13	Желтуха	Желтушное окрашивание кожи и слизистых
4	14	ОРВИ	Острая респираторная вирусная инфекция
5	15	ОРЗ	Острое респираторное заболевание

7 января 2020 г.

Стр. 1 из 1

Палаты Запрос			
Код сотрудника	Код палаты	Наименование	Вместимость
201	101	Палата интенс	4
206	102	Палата повыш	1
204	103	Палата детей	5
208	104	Палата взросл	5
209	105	Палата	4
*			



Путевки

Дата начала:

01.12.2019

Дата окончания:

29.12.2019

Продолжительность:

29

Код услуги 3:

1

Код клиента:

1

Код сотрудника:

1

▶

◀

🔍

Дата начал	Дата оконч	Продолжит	Код санатор	Код вида	Код услуги	Код услуги	Код услуги	Код клиент	Код сотруд	Отметка о с	Отметка об
01.12.2019	29.12.2019	29	2	1	5	2	1	1	1	+	+
01.12.2019	29.12.2019	29	5	4	2	5	4	6	10	+	+
01.12.2019	29.12.2019	29	1	4	4	3	1	2	11	+	+
01.12.2019	29.12.2019	29	3	3	1	5	2	5	2	+	+
01.12.2019	29.12.2019	29	6	2	2	4	1	2	3	+	+
01.12.2019	29.12.2019	29	8	1	4	2	3	1	4	+	+
01.12.2019	29.12.2019	29	8	5	5	4	1	3	4	+	+
01.12.2019	29.12.2019	29	9	2	3	1	5	6	5	+	+
01.12.2019	29.12.2019	29	10	5	1	2	3	2	8	+	+
01.12.2019	29.12.2019	29	7	1	5	1	4	4	9	+	+

аписы: 1 из 10 ▶ ◀ 🔍 Без фильтра Поиск

Кнопочная форма

Медуниверситет

☐ Группы

☐ Сотрудники

☐ Должности

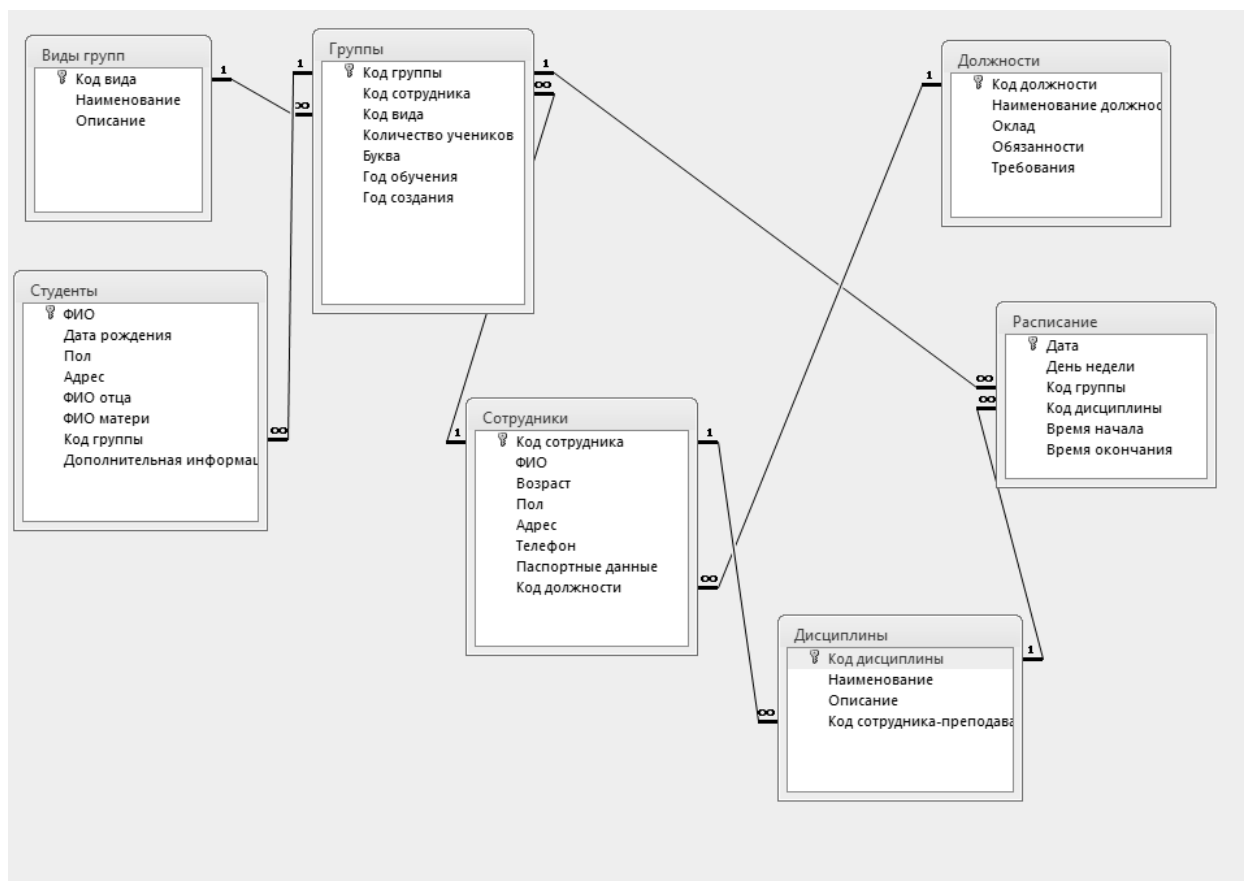
☐ Студенты

☐ Группы

☐ Виды групп

☐ Дисциплины

☐ Расписание



Студенты подчиненная форма			
ФИО	П	Адрес	
Бедрина Екатерина Сергеевна	Ж	Ул. Тверская 26	
Беклешова Юлия Вадимовна	Ж	Ул. Карла Маркса 15	
Булдаков Альберт Альбертови	М	Ул. Победы 10	
Ващенко Ирина Владиславовн	Ж	Ул. Моховая 4	
Высоцкая Наталья Владимиро	Ж	Ул. Полярные зори 36	
Галахов Дмитрий Станиславов	М	Ул. Держинского 17	
Дмитренко Анастасия Викторо	Ж	Ул. Шалимова 45	
Дунин Степан Иванович	М	Ул. Кирова 46	
Малаев Аврелий Дмитриевич	М	Ул. Папанина 1	
Михнев Никандр Алексеевич	М	Ул. Кирова 20	
*			

Сотрудники

Сотрудники

ФИО

Крюкова Екатерина Евгеньевна

Телефон

246-77-67

Паспортные данные

4714-5467

Дисциплины

	Наименование	Описание
	Философия	кафедра гуманитарных наук
*		

Записи: 1 из 1

Нет фильтра

Поиск

Сотрудники

Табличная форма Студенты

Табличная форма Студенты

ФИО

Бедрина Екатерина Сергеевна

Дата рождения

15.10.2000

Пол

Ж

Адрес

Ул. Тверская 26

ФИО отца

Бедрин Сергей Максимович

ФИО матери

Бедрина Ирина Степановна

Код группы

201

Дополнительная информация

Бюджет