

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«ИРКУТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра мировой экономики

Допускаю к защите

Руководитель _____

подпись

В.В.Надршин

И.О.Фамилия

Создание базы данных «Музыкальные записи»

наименование темы

КУРСОВАЯ РАБОТА

по дисциплине

Информационные технологии в менеджменте

Выполнил
студент группы

МПб – 13 - 2

Шифр группы

подпись

М.А.Шульга

И.О.Фамилия

Нормоконтроль

подпись

В.В.Надршин

И.О.Фамилия

Курсовая работа защищена с оценкой _____

Иркутск, 2014

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«ИРКУТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра мировой экономики

**ЗАДАНИЕ
НА КУРСОВУЮ РАБОТУ**

По курсу Информационные технологии в менеджменте

Студенту Шульге М.А.
(фамилия, инициалы)

Тема проекта

Создание базы данных «Музыкальные записи»

Исходные данные

Создать БД в Access «Музыкальные записи». Основную и вспомогательные таблицы создать с помощью Конструктора, определив необходимые поля и типы данных, связать таблицы с помощью Схемы данных, создать параметрически универсальные запросы, формы по каждой таблице и отчеты на основе созданных запросов.

Рекомендуемая литература

Ломтадзе В.В., Шишкина Л.П. Практическая информатика – Иркутск: Изд-во ИрГТУ, 2012. – 200 с.

Дата выдачи задания « 15 » марта 2014 г.

Дата представления работы руководителю « 15 » мая 2014 г.

Руководитель курсовой работы _____ **В.В.Надршин**
подпись

Содержание

Введение.....	4
1 Технология баз данных	5
1.1 История баз данных	5
1.2 Предназначение баз данных	6
1.3 Функции СУБД.....	6
1.4 Три модели баз данных	9
2 Microsoft Access.....	11
3 База данных «Музыкальные записи».....	133
3.1 Таблицы.....	133
3.2 Схема данных	14
3.3 Формы	14
3.4 Запросы	15
3.5 Отчеты.....	16
Заключение	17
Список литературы	198

Введение

В настоящее время среди разработчиков баз данных (БД) большой популярностью пользуется реляционная СУБД ACCESS, входящая в состав пакета Microsoft Office. Дружественный интерфейс и простота настройки, эффективные средства создания таблиц, форм, запросов, интеграция с другими приложениями пакета, средства организации работы с базами данных и защита информации – вот далеко не полный перечень достоинств этого приложения.

Основные функции СУБД – это описание структуры базы данных, обработка данных и управление данными.

База данных – это совокупность сведений о реальных объектах, процессах, событиях или явлениях, относящихся к определённой теме или задаче, организованная таким образом, чтобы обеспечить удобное представление этой совокупности, как в целом, так и любой её части.

Любая СУБД позволяет выполнять четыре простейшие операции с данными:

- добавить в таблицу одну или несколько записей;
- удалить из таблицы одну или несколько записей;
- обновить значения некоторых полей в одной или нескольких записях;
- найти одну или несколько записей, удовлетворяющих заданному условию.

Для выполнения этих операций используется механизм запросов. Результатом выполнения запросов является либо отобранное по определённым критериям множество записей, либо изменение в таблицах.

1 Технология баз данных

1.1 История баз данных

История возникновения и развития технологий баз данных может рассматриваться как в широком, так и в узком смысле.

С широкой точки зрения понятие истории баз данных обобщается до истории любых средств, с помощью которых человечество хранило и обрабатывало данные. В таком смысле могут упоминаться, например, средства учёта царской казны и налогов в древнем Шумере (4000 г. до н. э.), узелковая письменность инков — кипу, клинописи, содержащие документы Ассирийского царства и т. п. Следует помнить, что недостатком этого подхода является размывание понятия «база данных» и фактическое его слияние с понятиями «архив» и даже «письменность».

История баз данных в узком смысле рассматривает базы данных в традиционном (современном) понимании. Эта история начинается с 1955 года, когда появилось программируемое оборудование обработки записей. Программное обеспечение этого времени поддерживало модель обработки записей на основе файлов. Для хранения данных использовались перфокарты.

Оперативные сетевые базы данных появились в середине 1960-х. Операции над оперативными базами данных обрабатывались в интерактивном режиме с помощью терминалов. Простые индексно-последовательные организации записей быстро развились к более мощной модели записей, ориентированной на наборы. За руководство работой Data Base Task Group (DBTG), разработавшей стандартный язык описания данных и манипулирования данными, Чарльз Бахман получил Тьюринговскую премию. В это же время в сообществе баз данных COBOL была проработана концепция схем баз данных и концепция независимости данных.

Следующий важный этап связан с появлением в начале 1970-х реляционной модели данных, благодаря работам Эдгара Ф. Кодда. Работы Кодда открыли путь к тесной связи прикладной технологии баз данных с математикой и логикой. За свой вклад в теорию и практику Эдгар Ф. Кодд также получил премию Тьюринга.

Сам термин база данных (англ. database) появился в начале 1960-х годов, и был введён в употребление на симпозиумах, организованных фирмой SDC (System Development Corporation) в 1964 и 1965 годах, хотя понимался сначала в довольно узком смысле, в контексте систем искусственного интеллекта. В широкое употребление в современном понимании термин вошёл лишь в 1970-е годы.

1.2 Предназначение баз данных

База данных — объективная форма представления и организации совокупности данных (статей, расчетов и т. д.), систематизированных таким образом, чтобы эти данные могли быть найдены и обработаны с помощью ЭВМ. Данные эти представляют на машиночитаемых носителях информацию в форме, пригодной для оперативного использования с помощью ЭВМ (компьютера). Они должны удовлетворять информ. потребностям многих пользователей, быть структурированными и связанными между собой.

Пользователями базы данных могут быть различные прикладные программы, программные комплексы и специалисты предметной области, выступающие в роли потребителей или источников данных, называемые конечными пользователями.

В наше время базы данных имеют значение так называемого банка данных, в котором хранятся данные, относящиеся к той или иной сфере жизни человека.

Банк данных — совокупность данных, экономической информации, чаще всего хранящейся в компьютерах и предназначенной для использования несколькими пользователями.

Система управления базами данных (СУБД) — специализированная программа (чаще комплекс программ), предназначенная для организации и ведения базы данных. Для создания и управления информационной системой СУБД необходима в той же степени, как для разработки программы на алгоритмическом языке необходим транслятор.

1.3 Функции баз данных

К основным функциям СУБД принято относить следующие:

- управление данными во внешней памяти;
- управление буферами оперативной памяти;
- управление транзакциями;
- журнализация и восстановление БД после сбоев;
- поддержка языков БД.

Функция управления данными во внешней памяти включает обеспечение необходимых структур внешней памяти как для хранения данных, непосредственно входящих в БД, так и для служебных целей, например, для ускорения доступа к данным в некоторых случаях (обычно для этого используются индексы). В некоторых реализациях СУБД активно используются возможности существующих файловых систем, в других

работа производится вплоть до уровня устройств внешней памяти. Но подчеркнем, что в развитых СУБД пользователи в любом случае не обязаны знать, использует ли СУБД файловую систему, и если использует, то как организованы файлы. В частности, СУБД поддерживает собственную систему именования объектов БД.

СУБД обычно работают с БД значительного размера, по крайней мере этот размер обычно существенно больше доступного объема оперативной памяти. Понятно, что если при обращении к любому элементу данных будет производиться обмен с внешней памятью, то вся система будет работать со скоростью устройства внешней памяти. Практически единственным способом реального увеличения этой скорости является буферизация данных в оперативной памяти. При этом, даже если операционная система производит общесистемную буферизацию (как в случае ОС UNIX), этого недостаточно для целей СУБД, которая располагает гораздо большей информацией о полезности буферизации той или иной части БД. Поэтому в развитых СУБД поддерживается собственный набор буферов оперативной памяти с собственной дисциплиной замены буферов. Существует отдельное направление СУБД, которое ориентировано на постоянное присутствие в оперативной памяти всей БД. Это направление основывается на предположении, что в будущем объем оперативной памяти компьютеров будет настолько велик, что позволит не беспокоиться о буферизации. Пока эти работы находятся в стадии исследований.

Транзакция – это последовательность операций над БД, рассматриваемых СУБД как единое целое. Либо транзакция успешно выполняется, и СУБД фиксирует (СОММИТ) изменения БД, произведенные этой транзакцией, во внешней памяти, либо ни одно из этих изменений никак не отражается на состоянии БД. Понятие транзакции необходимо для поддержания логической целостности БД. Если вспомнить наш пример информационной системы с файлами СОТРУДНИКИ и ОТДЕЛЫ, то единственным способом не нарушить целостность БД при выполнении операции приема на работу нового сотрудника является объединение элементарных операций над файлами СОТРУДНИКИ и ОТДЕЛЫ в одну транзакцию. Таким образом, поддержание механизма транзакций является обязательным условием даже однопользовательских СУБД (если, конечно, такая система заслуживает названия СУБД). Но понятие транзакции гораздо более важно в многопользовательских СУБД.

Также основным требованием к СУБД является надежность хранения данных во внешней памяти. Под надежностью хранения понимается то, что СУБД должна быть в состоянии восстановить последнее согласованное состояние БД после любого аппаратного или программного сбоя. Обычно рассматриваются два возможных вида аппаратных сбоев:

- Мягкие сбои, которые можно трактовать как внезапную остановку работы компьютера (например, аварийное выключение питания),

- Жесткие сбои, характеризуемые потерей информации на носителях внешней памяти.

Примерами программных сбоев могут быть: аварийное завершение работы СУБД (по причине ошибки в программе или в результате некоторого аппаратного сбоя) или аварийное завершение пользовательской программы, в результате чего некоторая транзакция остается незавершенной. Первую ситуацию можно рассматривать как особый вид мягкого аппаратного сбоя; при возникновении последней требуется ликвидировать последствия только одной транзакции.

В любом случае для восстановления БД нужно располагать некоторой дополнительной информацией. Другими словами, поддержание надежности хранения данных в БД требует избыточности хранения данных, причем та часть данных, которая используется для восстановления, должна храниться особо надежно. Наиболее распространенным методом поддержания такой избыточной информации является ведение журнала изменений БД.

Журнал - это особая часть БД, недоступная пользователям СУБД и поддерживаемая с особой тщательностью (иногда поддерживаются две копии журнала, располагаемые на разных физических дисках), в которую поступают записи обо всех изменениях основной части БД. В разных СУБД изменения БД журналируются на разных уровнях: иногда запись в журнале соответствует некоторой логической операции изменения БД (например, операции удаления строки из таблицы реляционной БД), иногда - минимальной внутренней операции модификации страницы внешней памяти; в некоторых системах одновременно используются оба подхода.

Во всех случаях придерживаются стратегии "упреждающей" записи в журнал (так называемого протокола Write Ahead Log - WAL). Грубо говоря, эта стратегия заключается в том, что запись об изменении любого объекта БД должна попасть во внешнюю память журнала раньше, чем измененный объект попадет во внешнюю память основной части БД. Известно, что если в СУБД корректно соблюдается протокол WAL, то с помощью журнала можно решить все проблемы восстановления БД после любого сбоя.

Самая простая ситуация восстановления - индивидуальный откат транзакции. Строго говоря, для этого не требуется общесистемный журнал изменений БД. Достаточно для каждой транзакции поддерживать локальный журнал операций модификации БД, выполненных в этой транзакции, и производить откат транзакции путем выполнения обратных операций, следуя от конца локального журнала. В некоторых СУБД так и делают, но в большинстве систем локальные журналы не поддерживают, а индивидуальный откат транзакции выполняют по общесистемному журналу, для чего все записи от одной транзакции связывают обратным списком (от конца к началу).

При мягком сбое во внешней памяти основной части БД могут находиться объекты, модифицированные транзакциями, не закончившимися к моменту сбоя, и могут отсутствовать объекты, модифицированные

транзакциями, которые к моменту сбоя успешно завершились (по причине использования буферов оперативной памяти, содержимое которых при мягком сбое пропадает). При соблюдении протокола WAL во внешней памяти журнала должны гарантированно находиться записи, относящиеся к операциям модификации обоих видов объектов. Целью процесса восстановления после мягкого сбоя является состояние внешней памяти основной части БД, которое возникло бы при фиксации во внешней памяти изменений всех завершившихся транзакций и которое не содержало бы никаких следов незаконченных транзакций. Для того, чтобы этого добиться, сначала производят откат незавершенных транзакций (undo), а потом повторно воспроизводят (redo) те операции завершенных транзакций, результаты которых не отображены во внешней памяти. Этот процесс содержит много тонкостей, связанных с общей организацией управления буферами и журналом.

Для работы с базами данных используются специальные языки, в целом называемые языками баз данных. В ранних СУБД поддерживалось несколько специализированных по своим функциям языков. Чаще всего выделялись два языка - язык определения схемы БД (SDL - Schema Definition Language) и язык манипулирования данными (DML - Data Manipulation Language). SDL служил главным образом для определения логической структуры БД, т.е. той структуры БД, какой она представляется пользователям. DML содержал набор операторов манипулирования данными, т.е. операторов, позволяющих заносить данные в БД, удалять, модифицировать или выбирать существующие данные. Мы рассмотрим более подробно языки ранних СУБД в следующей лекции.

В современных СУБД обычно поддерживается единый интегрированный язык, содержащий все необходимые средства для работы с БД, начиная от ее создания, и обеспечивающий базовый пользовательский интерфейс с базами данных. Стандартным языком наиболее распространенных в настоящее время реляционных СУБД является язык SQL (Structured Query Language). В нескольких лекциях этого курса язык SQL будет рассматриваться достаточно подробно, а пока мы перечислим основные функции реляционной СУБД, поддерживаемые на "языковом" уровне (т.е. функции, поддерживаемые при реализации интерфейса SQL).

1.4 Три модели баз данных

Существует три модели баз данных:

I. Иерархическая - Иерархическая структура представляет совокупность элементов, связанных между собой по определенным правилам. Объекты, связанные иерархическими отношениями, образуют ориентированный граф (перевернутое дерево). К основным понятиям иерархической структуры

относятся: уровень, элемент (узел), связь. Узел – это совокупность атрибутов данных, описывающих некоторый объект. На схеме иерархического дерева узлы представляются вершинами графа. Каждый узел на более низком уровне связан только с одним узлом, находящимся на более высоком уровне. Иерархическое дерево имеет только одну вершину (корень дерева), не подчиненную никакой другой вершине и находящуюся на самом верхнем (первом) уровне. Зависимые (подчиненные) узлы находятся на втором, третьем и т.д. уровнях. Количество деревьев в базе данных определяется числом корневых записей. К каждой записи базы данных существует только один (иерархический) путь от корневой записи.

II. Реляционная - логическая модель данных, прикладная теория построения баз данных, которая является приложением к задачам обработки данных таких разделов математики как теории множеств и логика первого порядка. На реляционной модели данных строятся реляционные базы данных. Реляционная модель данных включает следующие компоненты:

- Структурный аспект (составляющая) — данные в базе данных представляют собой набор отношений.

- Аспект (составляющая) целостности — отношения (таблицы) отвечают определенным условиям целостности. РМД поддерживает декларативные ограничения целостности уровня домена (типа данных), уровня отношения и уровня базы данных.

- Аспект (составляющая) обработки (манипулирования) — РМД поддерживает операторы манипулирования отношениями (реляционная алгебра, реляционное исчисление). Кроме того, в состав реляционной модели данных включают теорию нормализации. Термин «реляционный» означает, что теория основана на математическом понятии отношение (relation). В качестве неформального синонима термину «отношение» часто встречается слово таблица. Необходимо помнить, что «таблица» есть понятие нестрогое и неформальное и часто означает не «отношение» как абстрактное понятие, а визуальное представление отношения на бумаге или экране. Некорректное и нестрогое использование термина «таблица» вместо термина «отношение» нередко приводит к недопониманию. Наиболее частая ошибка состоит в рассуждениях о том, что РМД имеет дело с «плоскими», или «двумерными» таблицами, тогда как таковыми могут быть только визуальные представления таблиц. Отношения же являются абстракциями, и не могут быть ни «плоскими», ни «неплоскими».

III. Сетевая модель данных - логическая модель данных, являющаяся расширением иерархического подхода, строгая математическая теория, описывающая структурный аспект, аспект целостности и аспект обработки данных в сетевых базах данных. Разница между иерархической моделью данных и сетевой состоит в том, что в иерархических структурах запись-потомок должна иметь в точности одного предка, а в сетевой структуре данных у потомка может иметься любое число предков. Сетевая БД состоит из набора экземпляров определенного типа записи и набора экземпляров

определенного типа связей между этими записями. Тип связи определяется для двух типов записи: предка и потомка. Экземпляр типа связи состоит из одного экземпляра типа записи предка и упорядоченного набора экземпляров типа записи потомка. Для данного типа связи L с типом записи предка P и типом записи потомка C должны выполняться следующие два условия:

- каждый экземпляр типа записи P является предком только в одном экземпляре типа связи L;
- каждый экземпляр типа записи C является потомком не более чем в одном экземпляре типа связи L.

2 Microsoft Access

Microsoft Access - это интерактивная реляционная СУБД (relational database management system - RDBMS) для WINDOWS. Это программа, которую используют для хранения и извлечения данных в зависимости от отношений, которые были установлены. Работа с ней упрощена посредством манипулятора мыши. Графические возможности оболочки производят большое впечатление при изготовлении высококачественных отчетов и распечаток. Все это благодаря поддержке True-type шрифтов и встраивания OLE-объектов (Object Linking and Embedding) в рамках среды WINDOWS. OLE-объект представляет собой ссылку на определенную информацию, которая остается в своей первоначальной форме. OLE-объектом может быть EXCEL - таблица, Paintbrush - иллюстрация или Sound - файл.

Новшеством программы Access является Cue Cards - Система суфлирования. Это обучающая система, предоставляющая пользователю рекомендации по выходу из реальных ситуаций при решении прикладных задач. При знании английского языка пользователь всегда сможет получить подсказку о том, что следует делать далее.

Access также предоставляет в распоряжение пользователя механизмы работы с базами данных различных форматов. К примеру, можно прямо обращаться к базам данных dBASE, Paradox или Vtrieve без конвертирования их в формат, используемый Access. В состав пакета Access также входит язык Access Basic (встроенный диалект языка Visual Basic), дающий возможность формирования специализированных систем управления базами данных.

Основными понятиями или объектами этой системы являются: таблицы, запросы, формуляры, отчеты, макросы и модули. И конечно же главным понятием будет база данных (database).

База данных это набор данных, который связан с определенной темой или назначением. Когда создается база данных в Microsoft Access, создается один файл, который содержит все таблицы, запросы, формуляры, отчеты и

другие объекты, которые помогают использовать информацию в базе данных.

Microsoft Access создает пустой файл базы данных и открывает окно базы данных Database window. Созданную базу данных можно наполнить объектами различного рода и выполнять операции с ними. Но с базой данных можно выполнять операции как с неделимым образованием. Все операции такого рода - операции управления базой данных - сосредоточены в меню File прикладного окна Access или в окне базы данных.

Microsoft Access высветит окно соответствующей базы данных. При открытии базы данных можно ограничить возможные операции с базой данных только чтением и ограничить круг пользователей, имеющих доступ к базе данных только собственной персоной. Такие ограничения могут оказаться полезными, если база данных эксплуатируется в многопользовательской системе. Если необходимо только просмотреть базу данных и содержащиеся в ней объекты, но не изменять их, то при открытии базы данных необходимо активизировать контрольный индикатор Read Only (только чтение).

Если необходимо предотвратить изменения данных и объектов данной базы данных со стороны других пользователей, то при открытии базы данных необходимо установить опцию Exclusive (исключительные права).

Базу данных можно подвергнуть сжатию, для этого необходимо выбрать в меню File директиву Compact Database (сжать базу данных). Access создает копию базы данных и по-новому размещает ее на диске. После выбора сжимаемого файла в диалоговом окне сжатия базы данных появляется диалоговое окно цели сжатия (Database to Compact Into), в котором можно выбрать имя для сжатой версии базы данных. Если используется, то же самое имя, и база данных оказывается успешно сжатой, то по завершении сжатия Access автоматически уничтожает исходную(несжатую) версию.

Базу данных можно зашифровать для защиты секретной информации от просмотра. Это осуществляется с помощью директивы Encrypt Database (расшифровать - Decrypt Database). В появившемся диалоговом окне выберите подлежащую шифрованию/дешифрованию базу данных и щелкните на кнопке ОК. В результате откроется диалоговое окно целевого файла, где следует указать имя файла, в который будет помещен результат шифрования/дешифрования. Если используется, то же самое имя и база данных была успешно зашифрована/расшифрована, то Access автоматически заменяет оригинальный файл на полученный в результате преобразования.

В некоторых ситуациях база данных может оказаться поврежденной. Для восстановления базы данных служит директива Repair Database (восстановить базу данных) в меню File. Эту операцию рекомендуется применять в целях профилактики так как Access не замечает, что база данных была повреждена. В диалоговом окне ремонта нужно ввести имя, директорию и накопитель поврежденной(или подозреваемой в повреждении)

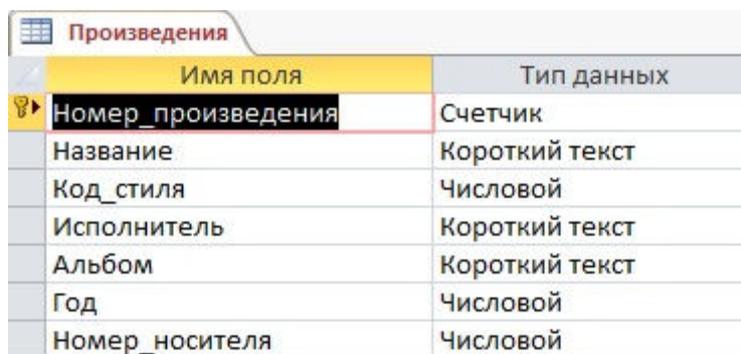
базы данных и нажать кнопку ОК. После этого Access займется "починкой" заданной базы данных и спасет то, что еще можно спасти.

3 База данных «Музыкальные записи»

3.1 Таблицы

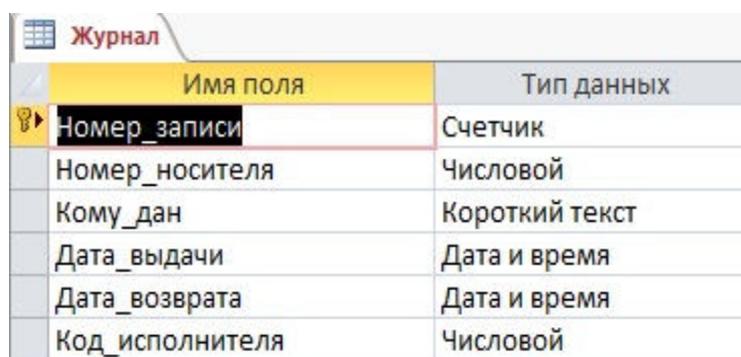
Для создания БД «Музыкальные записи» нам нужны следующие таблицы:

- 1) Основные: *Произведения* и *Журнал*. (рис. 1, 2)
- 2) Вспомогательные: *Стили*, *Носители* и *Исполнители*. (рис. 3, 4, 5).



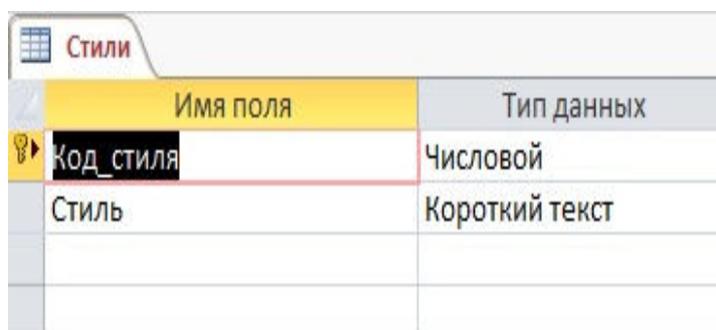
Имя поля	Тип данных
Номер_произведения	Счетчик
Название	Короткий текст
Код_стиля	Числовой
Исполнитель	Короткий текст
Альбом	Короткий текст
Год	Числовой
Номер_носителя	Числовой

Рисунок 1. – Поля таблицы «Произведения»



Имя поля	Тип данных
Номер_записи	Счетчик
Номер_носителя	Числовой
Кому_дан	Короткий текст
Дата_выдачи	Дата и время
Дата_возврата	Дата и время
Код_исполнителя	Числовой

Рисунок 2. – Поля таблицы «Журнал»



Имя поля	Тип данных
Код_стиля	Числовой
Стиль	Короткий текст

Рисунок 3. – Поля таблицы «Стили»

Носители			
Номер_носителя	Тип_носителя	Полка	Щелкните для добавления
1	Audio CD	1	
2	Мр3 CD	2	
3	Аудио кассета	3	
4	Пластинка	4	
*	0	0	

Рисунок 4. – Поля таблицы «Носители»

Исполнители			
Код_исполнителя	Исполнитель	Лучшее произведе	Щелкните для добавления
1	Metallica	Creeping Death	
2	Pink Floyd	Money	
3	Eric Clapton	I Shot a Sheriff	
4	The Offspring	Have You Ever	
*	(№)		

Рисунок 5. Поля таблицы «Исполнители»

3.2 Схема данных

Схема данных отображает связь таблиц в базе данных «Музыкальные записи». (рис. 6)

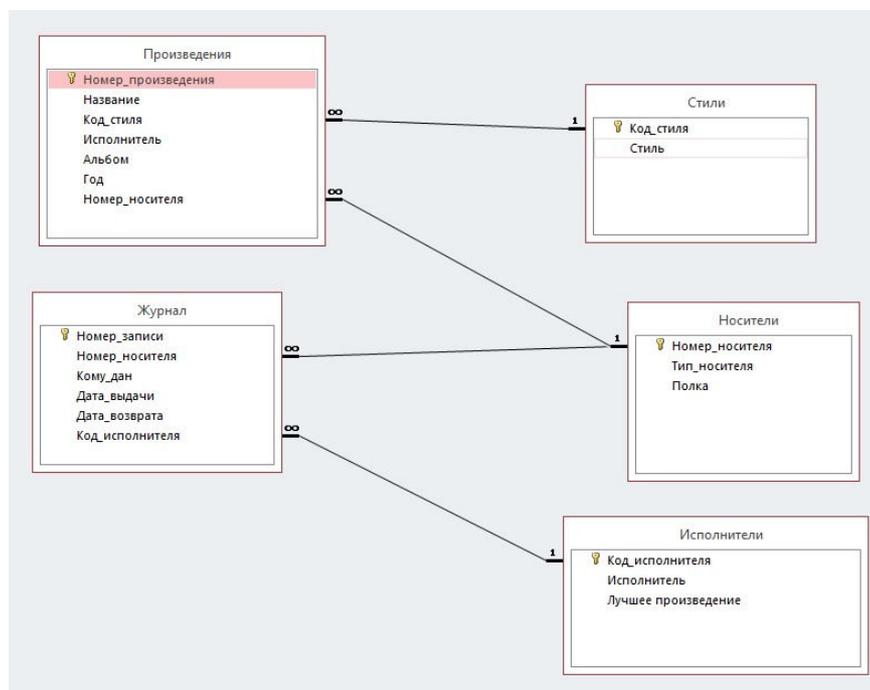


Рисунок 6. Схема данных

3.3 Формы

В базе данных выгоднее создавать, так называемые, формы для отображения произведений, их стиля, типа носителя, исполнителя и т.д. (рис. 7)

Рисунок 7. Форма «Произведения»

3.4 Запросы

Запросы позволяют выбирать данные из одной или нескольких связанных таблиц. Результатом выполнения запроса является результирующая таблица, которая наряду с другими таблицами может быть использована при обработке данных. С помощью запросов можно также обновлять, удалять или добавлять данные в таблицы. Основным назначением запросов является отбор данных по критериям поиска.

В базе данных музыкальные записи созданы запросы:

1. Произведения за период (вы задаете интересующий вас год или временной промежуток, в результате получаете результирующую таблицу, на которой видны нужные записи.) (рис. 8).
2. Должники (этот запрос показывает список людей, которые одолжили записи на определенных носителях и не вернули их.) (рис. 9).

Номер_произведения	Название	Стиль	Исполнитель	Альбом	Год	Номер_носителя	Полка
1	Fight Fire With Fire	Трэш-метал	Metallica	Ride the Lighting	1984	1	1
2	Ride the Lighting	Трэш-метал	Metallica	Ride the Lighting	1984	1	1
3	For Whom the Bell Tolls	Трэш-метал	Metallica	Ride the Lighting	1984	1	1
4	Fade to Black	Баллада	Metallica	Ride the Lighting	1984	1	1
5	Trapped Under Ice	Трэш-метал	Metallica	Ride the Lighting	1984	1	1
6	Creeping Death	Трэш-метал	Metallica	Ride the Lighting	1984	1	1
7	The Call of Ktulu	Инструментальный рок	Metallica	Ride the Lighting	1984	1	1
8	Escape	Трэш-метал	Metallica	Ride the Lighting	1984	1	1
9	Enter Sandman	Трэш-метал	Metallica	Black Album	1991	3	3

Рисунок 8. Запрос «Произведения за период»

Номер_записи	Номер_носителя	Полка	Кому_дан	Дата_выдачи	Дата_возврата
1	1	1	Шульга П.А.	11.02.2004	
4	1	1	Домбровская Е.В.	14.12.2013	
6	1	1	Решетников К.А.	16.05.2002	
*	(№)				

Рисунок 9. Запрос «Должники»

3.5 Отчеты

Отчеты предназначены для формирования выходных документов и вывода их на печать. По своим свойствам и структуре отчеты во многом подобны формам. Основное их отличие заключается в том, что в отчете отображаются все данные и в них предусмотрена возможность группировать данные по различным критериям. Отчеты в отличие от форм могут содержать специальные элементы оформления, характерные для печати документов: колонтитулы, номера страниц и т.д.

В базе данных музыкальные записи созданы отчеты:

1. Произведения за период (на основе запроса см пункт «Запросы»)
2. Стили (отчёт создан что бы показать записи определённых стилей, для удобства выбора определенных композиций для прослушивания)

Произведения

Альбом		461 Ocean Boulevard					
произведения	Название	Стиль	Исполнитель	Год	Номер_носителя	Тип_носителя	Полка
31	Motherless Children	Блюз	Eric Clapton	1974	2		2
32	Give Me Strength	Блюз	Eric Clapton	1974	2		2
33	Willie And The Hand Jive	Блюз	Eric Clapton	1974	2		2
34	Get Ready	Блюз-рок	Eric Clapton	1974	2		2
35	I Shot a Sheriff	Блюз-рок	Eric Clapton	1974	2		2
36	I Can't Hold Out	Блюз	Eric Clapton	1974	2		2
37	Please Be With Me	Блюз-рок	Eric Clapton	1974	2		2
38	Let It Grow	Блюз	Eric Clapton	1974	2		2
39	Steady Rollin Man	Блюз-рок	Eric Clapton	1974	2		2
40	Mainline Florida	Блюз	Eric Clapton	1974	2		2
Min				1974			
Max				1974			

Рисунок 10. Отчет «Произведения за период»

Стиль	Название	Исполнитель	Альбом	Год	Тип носителя	Полка
Трэш-метал						
	11 Holier Than Thou	Metallica	Black Album	1991	Аудио кассета	3
	1 Fight Fire With Fire	Metallica	Ride the Lightning	1984	Audio CD	1
	9 Enter Sandman	Metallica	Black Album	1991	Аудио кассета	3
	10 Sad But True	Metallica	Black Album	1991	Аудио кассета	3
	13 Wherever I May Roam	Metallica	Black Album	1991	Аудио кассета	3
	14 Don't Tread On Me	Metallica	Black Album	1991	Аудио кассета	3
	15 Through the Never	Metallica	Black Album	1991	Аудио кассета	3
	17 Of Wolf and Man	Metallica	Black Album	1991	Аудио кассета	3
	18 The God That Failed	Metallica	Black Album	1991	Аудио кассета	3
	19 My Friend of Misery	Metallica	Black Album	1991	Аудио кассета	3
	5 Trapped Under Ice	Metallica	Ride the Lightning	1984	Audio CD	1
	2 Ride the Lightning	Metallica	Ride the Lightning	1984	Audio CD	1
	20 The Struggle Within	Metallica	Black Album	1991	Аудио кассета	3
	3 For Whom the Bell Tolls	Metallica	Ride the Lightning	1984	Audio CD	1
	8 Escape	Metallica	Ride the Lightning	1984	Audio CD	1
	6 Creeping Death	Metallica	Ride the Lightning	1984	Audio CD	1

Рисунок 11. Отчет «Стили»

Заключение

Подобная база данных эффективно выполняет поставленные задачи, которые пользователь задает, что делает СУБД незаменимым инструментом в жизни и работе пользователя. Приобретая навыки пользования СУБД, люди значительно облегчают себе задачу ведения учета. Создание запросов, форм и отчетов в базах данных делает восприятие сложной информации легче.

Представленная база данных может использоваться для учета музыкальных записей как в магазине, так и дома.

Список использованной литературы

1. Ломтадзе В.В., Шишкина Л.П. Практическая информатика. – Иркутск: изд-во ИрГТУ. – 2012. – 200 с.
2. Бояринцева Т.П., Воропаева Е.Ф., Дмитриенко Т.А., Шишкина Л.П. Лабораторный практикум по информатике. Расширенные возможности Excel. - Иркутск: изд-во ИрГТУ. - 2003. – 71 с.
3. ru.wikipedia.org

