

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования

«Кубанский государственный университет»
Филiaal в г. Новороссийске

Кафедра информатики и математики

ОТЧЕТ

по учебной практике

Выполнил студент 2 курса
ОФО направления
38.03.05 Бизнес-информатика



Е.И. Голубов

подпись

Руководитель практики
Канд. физ.-мат. наук, доцент



И.Г. Рзун

Оценка, подпись

Новороссийск 2016

СОДЕРЖАНИЕ

Введение	3
1 Описание предметной области и основы работы с базами данных	6
2 Разработка базы данных для кондитерских фабрик России	17
Заключение.....	22
Список использованных источников.....	23

ВВЕДЕНИЕ

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

Жизненный цикл любого программного продукта, в том числе и системы управления базой данных, состоит из стадий проектирования, реализации и эксплуатации.

Естественно, наиболее значительным фактором в жизненном цикле приложения, работающего с базой данных, является стадия проектирования. От того, насколько тщательно продумана структура базы, насколько четко определены связи между ее элементами, зависит производительность системы и ее информационная насыщенность, а значит - и время ее жизни.

Обычно БД создается для хранения и доступа к данным, содержащим сведения о некоторой предметной области, то есть некоторой области человеческой деятельности или области реального мира. Всякая БД должна представлять собой систему данных о предметной области. БД, относящиеся к одной и той же предметной области, в различных случаях содержат более или менее детализированную информацию о ней, причем таким способом, который заведомо исключает ненужную избыточность. В хорошо спроектированной базе данных избыточность данных исключается, и вероятность сохранения противоречивых данных минимизируется. Таким образом, создание баз данных преследует две основные цели: понизить избыточность данных и повысить их надежность.

Хорошо спроектированная база данных:

- Удовлетворяет всем требованиям пользователей к содержимому базы данных. Перед проектированием базы необходимо провести обширные исследования требований пользователей к функционированию базы данных.

- Гарантирует непротиворечивость и целостность данных. При проектировании таблиц нужно определить их атрибуты и некоторые правила, ограничивающие возможность ввода пользователем неверных значений. Для верификации данных перед непосредственной записью их в таблицу база данных должна осуществлять вызов правил модели данных и тем самым гарантировать сохранение целостности информации.

- Обеспечивает естественное, легкое для восприятия структурирование информации. Качественное построение базы позволяет делать запросы к базе более "прозрачными" и легкими для понимания; следовательно, снижается вероятность внесения некорректных данных и улучшается качество сопровождения базы.

- Удовлетворяет требованиям пользователей к производительности базы данных. При больших объемах информации вопросы сохранения производительности начинают играть главную роль, сразу "высвечивая" все недочеты этапа проектирования.

Современная жизнь требует от нас максимум работы и минимум затраченного времени, минимум расчетов и максимум результатов и достижений. Но, когда речь идет о крупном предприятии, то разговоры об остатке времени не может и быть. Именно поэтому сегодня практически все предприятия, которые достигают успеха, а значит и имеют большую сферу деятельности, заказывают создание баз данных, которые максимально уменьшают время работы и могут обслуживать крупные предприятия. Такое программное обеспечение - это лучший способ защиты, стабильной работы и финансового учета предприятия.

На фирме, где уже созданы базы данных, директор имеет возможность постоянно наблюдать за течением работ и деятельностью своего персонала, держать под контролем финансовое положение. А в наше время такая стабилизированная и контролируемая работа любой фирмы - это важный шаг для прибыли.

Самое главное достоинство базы данных - это то, что все данные, хранятся и обрабатываются централизованно. Это означает, что не надо вести отдельный учет каждой отрасли предприятия, поскольку все необходимые данные хранятся в одном и том же хранилище.

Во многих городах России существует либо кондитерская фабрика, либо филиал этой фабрики. Фабрика имеет название, которое уникально. Имеет также дату введения в строй, т. е. дату открытия фабрики. В зависимости от владельца, фабрики разделяются по типу: открытое акционерное общество (ОАО), закрытое акционерное общество (ЗАО), госпредприятие (ГП), частный предприниматель (ЧП).

Основной задачей кондитерской фабрики есть производство и сбыт кондитерских изделий. Для этого фиксируется дата производства и дата поставки продукта (каждый продукт имеет свой уникальный код), количество товара, его тип.

Чтобы точно фиксировать поставки продуктов магазин (потребитель) имеет № накладной (уникален), количество поступившего товара, цену за единицу товара, дату поставки.

Это позволяет систематизировать работу фабрик: фиксировать дату выпуска и дату сбыта продукции, ее тип, название; легко узнать данные магазина (потребителя), которой произвел импорт товара. Также с помощью этих данных можно быстро и удобно использовать информацию о поставках.

1 Описание предметной области и основы работы с базами данных

Логически в современной реляционной СУБД можно выделить наиболее внутреннюю часть – ядро СУБД (часто его называют DataBaseEngine), компилятор языка БД (обычно SQL), подсистему поддержки времени выполнения, набор утилит. В некоторых системах эти части выделяются явно, в других – нет, но логически такое разделение можно провести во всех СУБД.

СУБД Access корпорации Microsoft® обладает исключительно высокими скоростными характеристиками и в этом отношении заметно выделяется среди других интерпретирующих систем. Набор команд и функций, предлагаемых разработчикам программных продуктов в среде Microsoft® Access 2000, по мощи и гибкости отвечает любым современным требованиям к представлению и обработке данных. Здесь может быть реализован максимально удобный, гибкий и эффективный пользовательский интерфейс. Система также обладает средствами быстрой генерации форм, отчетов и меню, поддерживает язык SQL.

Автоматизированная информационная система (АИС) — совокупность программно-аппаратных средств, предназначенных для автоматизации деятельности, связанной с хранением, передачей и обработкой информации.

АИС являются, с одной стороны, разновидностью информационных систем (ИС), с другой — автоматизированных систем (АС), вследствие чего их часто называют ИС или АС.

АИС может быть определена как комплекс автоматизированных информационных технологий, предназначенных для информационного обслуживания – организованного непрерывного технологического процесса подготовки и выдачи потребителям научной, управленческой и др. информации, используемой для принятия решений, в соответствии с нуждами для поддержания эффективной деятельности.

Классическими примерами автоматизированных информационных систем являются банковские системы, автоматизированные системы управления предприятиями, системы резервирования авиационных или железнодорожных билетов и т. д.

Основной причиной создания и развития АИС является необходимость ведения учёта информации о состоянии и динамике объекта, которому посвящена система. На основании информационной картины, создаваемой системой, руководители различного звена могут принимать решения об управляющих воздействиях с целью решения текущих проблем.

Учётные данные системы могут быть подвергнуты автоматической обработке для последующего тактического и стратегического анализа с целью принятия управленческих решений большего горизонта действия.

· Побочными, возможными, но не гарантированными эффектами от использования системы могут выступать:

- повышение производительности работы персонала;
- улучшение качества обслуживания клиентов;
- снижение трудоемкости и напряженности труда персонала;
- снижение количества ошибок в его действиях.

Мною была выбрана СУБД Microsoft Access. Microsoft Office Access или просто Microsoft Access — реляционная СУБД корпорации Microsoft. Имеет широкий спектр функций, включая связанные запросы, связь с внешними таблицами и базами данных. Благодаря встроенному языку VBA, в самом Access можно писать приложения, работающие с базами данных. Основные компоненты MS Access:

- построитель таблиц;
- построитель экранных форм;

- построитель SQL-запросов (язык SQL в MS Access не соответствует стандарту ANSI);

- построитель отчётов, выводимых на печать.

Они могут вызывать скрипты на языке VBA, поэтому MS Access позволяет разрабатывать приложения и БД практически «с нуля» или написать оболочку для внешней БД.

MS Access является файл-серверной СУБД и потому применима лишь к маленьким приложениям. Отсутствует ряд механизмов, необходимых в многопользовательских БД, таких, например, как триггеры. Опыт показывает [источник не указан 55 дней], что даже для проектов на 5-20 пользователей предпочтительно использовать клиент-серверные решения.

Существенно расширяет возможности MS Access по написанию приложений механизм связи с различными внешними СУБД: "связанные таблицы" (связь с таблицей СУБД) и "запросы к серверу" (запрос на диалекте SQL, который "понимает" СУБД). Также MS Access позволяет строить полноценные клиент-серверные приложения на СУБД MS SQL Server. При этом имеется возможность совместить с присущей MS Access простотой инструменты для управления БД и средства разработки.

На сегодняшний день наиболее часто используются три модели данных: иерархическая, сетевая и реляционная. Кроме них существуют и другие модели, например модель данных, основанная на инвертированных списках или объектно-ориентированная, однако они не имеют широкого распространения, так как базы на инвертированных списках использовались на заре развития СУБД, а объектно-ориентированные базы данных ещё не до конца изучены. Таким образом, выбор сокращается до трёх вышеназванных моделей данных.

Иерархические базы данных. Этот вид баз данных одним из первых получил широкое распространение и стал промышленно использоваться.

Иерархическая БД состоит из упорядоченного набора деревьев; более точно, из упорядоченного набора нескольких экземпляров одного типа дерева. Тип дерева состоит из одного "корневого" типа записи и упорядоченного набора из нуля или более типов поддеревьев (каждое из которых является некоторым типом дерева). Тип дерева в целом представляет собой иерархически организованный набор типов записи. Примерами типичных операторов манипулирования иерархически организованными данными могут быть следующие операторы:

- Найти указанное дерево БД;
- Перейти от одного дерева к другому;
- Перейти от одной записи к другой внутри;
- Перейти от одной записи к другой в порядке обхода иерархии;
- Вставить новую запись в указанную позицию;
- Удалить текущую запись.

Одним из основных преимуществ иерархической модели данных является скорость поиска по базе.

Типичным представителем (наиболее известным и распространенным) является InformationManagementSystem (IMS) фирмы IBM. Первая версия появилась в 1968 г. До сих пор поддерживается много баз данных, что создает существенные проблемы с переходом, как на новую технологию БД, так и на новую технику.

Сетевая модель данных. Сетевой подход к организации данных является расширением иерархического подхода. В иерархических структурах запись-потомок должна иметь в точности одного предка; в сетевой структуре данных потомок может иметь любое число предков.

Сетевая БД состоит из набора записей и набора связей между ними, а если говорить более точно: из набора экземпляров каждого типа из заданного в схеме БД набора типов записи и набора экземпляров каждого типа из заданного набора типов связи.

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 два условия:

- Каждое экземпляр типа P является предком только в одном экземпляре L ;
- Каждый экземпляр C является потомком не более чем в одном экземпляре L .

На формирование типов связи не накладываются особые ограничения; возможны, например, ситуации:

а) Тип записи потомка в одном типе связи $L1$ может быть типом записи предка в другом типе связи $L2$ (как в иерархии).

б) Данный тип записи P может быть типом записи предка в любом числе типов связи.

в) Данный тип записи P может быть типом записи потомка в любом числе типов связи.

г) Может существовать любое число типов связи с одним и тем же типом записи предка и одним и тем же типом записи потомка; и если $L1$ и $L2$ два типа связи с одним и тем же типом записи предка P и одним и тем же типом записи потомка C , то правила, по которым образуется родство, в разных связях могут различаться.

д) Типы записи X и Y могут быть предком и потомком в одной связи и потомком и предком - в другой.

е) Предок и потомок могут быть одного типа записи.

Примерный набор операций может быть таковым:

- Найти конкретную запись в наборе однотипных записей (инженера Сидорова);
- Перейти от предка к первому потомку по некоторой связи (к первому сотруднику отдела 310);
- Перейти к следующему потомку в некоторой связи (от Сидорова к Иванову);
- Перейти от потомка к предку по некоторой связи (найти отдел Сидорова);
- Создать новую запись;
- Уничтожить запись;
- Модифицировать запись;
- Включить в связь;
- Исключить из связи;
- Переставить в другую связь и т.д.

К достоинствам сетевой СУБД можно отнести возможность экономии памяти за счет разделения подобъектов.

Типичным представителем является Integrated Database Management System (IDMS) компании Cullinet Software Inc., предназначенная для использования на машинах основного класса фирмы IBM под управлением большинства операционных систем. Архитектура системы основана на

предложениях Data Base Task Group (DBTG) Комитета по языкам программирования Conference on Data Systems Languages (CODASYL), организации, ответственной за определение языка программирования Кобол.

Описанные выше модели данных относятся к так называемым ранним СУБД. У этих моделей есть существенные недостатки так то:

- Слишком сложно пользоваться;
- Фактически необходимы знания о физической организации;
- Прикладные системы зависят от этой организации;
- Их логика перегружена деталями организации доступа к БД.

В условия современного развития компьютерной техники, когда с базами данных всё чаще работают непрофессионалы, делает эти СУБД весьма сложными для обслуживания.

Инфологическая модель должна включать такое формализованное описание предметной области, которое легко будет "читаться" не только специалистами по базам данных.

Инфологическое проектирование, прежде всего, связано с попыткой представления семантики предметной области в модели БД. Реляционная модель данных в силу своей простоты и лаконичности не позволяет отобразить семантику, то есть смысл предметной области.

Проблема представления семантики давно интересовала разработчиков, и в семидесятых годах было предложено несколько моделей данных, названных семантическими моделями. К ним можно отнести семантическую модель данных, предложенную Хаммером (Hammer) и Мак-Леонам (McLeon) в 1981 году, функциональную модель данных Шипмана (Shipman), также созданную в 1981 году, модель "сущность—связь", предложенную Ченом (Chen) в 1976 году, и ряд других моделей. У всех моделей были свои положительные и

отрицательные стороны, но испытание временем выдержала только последняя. И в настоящий момент именно модель Чена "сущность—связь", или "Entity Relationship", стала фактическим стандартом при инфологическом моделировании баз данных.

Модель «сущность-связь» называют также «ER-моделью» (essence-сущность, relation-связь).

При проектировании БД информацию обычно размещают в нескольких таблицах. Таблицы при этом связывают с семантикой информации. В реляционной СУБД для указания связей в таблице производят операции их связывания. Рассмотрим наиболее часто встречаемые бинарные связи:

1. Связи вила 1:1 образуется в случае, когда все поля записи основной таблицы и дополнительной таблицы являются ключевыми.

2. Связь 1:М может быть в случае, когда одной записи основной таблицы соответствует несколько записей дополнительной таблицы.

3. Связь М:1 может быть тогда, когда нескольким записям основной таблицы ставится в соответствии одна запись дополнительной.

4. Связь М:М возникает в том случае когда нескольким записям основной таблицы соответствует несколько записей дополнительной. В реляционной БД связь М:М реализуется через дополнительные таблицы.

Реляционная модель баз данных была предложена сотрудником фирмы ИВМ Э. Кодом в начале 70-х годов. Будучи математиком, он предложил использовать для обработки данных аппарат теории множеств (объединение, пересечение, разность и Декартово произведение). Он показал, что любое представление данных сводится к совокупности двумерных таблиц особого вида, известных в математике как отношения.

Одна из главных идей заключается в том, что связи между данными должны устанавливаться в соответствии с их внутренними логическими взаимоотношениями. В реляционной модели одной командой могут обрабатываться целые файлы.

Реляционная БД представляет собой информацию об объекте, представленную в виде двумерного массива - таблицы объединенных определенными связями.

Концептуальная модель представляет объекты и их взаимосвязи без указания способов их физического хранения. Таким образом, концептуальная модель является, по существу, моделью предметной области.

Концептуальная модель включает описания предметов и их взаимосвязей, представляющих интерес в рассматриваемой предметной области и выявляемых в результате анализа данных.

После анализа данных, характеризующих предметную область, отдельные данные объединяются в таблицы. Нужно определить оптимальный набор таблиц, который обладает лучшими свойствами при включении, удалении и изменении данных, а также позволяет сделать избыточность данных и время ответа на запросы минимальными.

Атрибут значение, которого идентифицируется кортежами (строками таблицы) называется ключом. Отношение может содержать и несколько ключей, один из которых объявляется первичным. Первичные ключи не могут обновляться. Все прочие ключи отношений являются возможными ключами.

Если в отношении кортеж идентифицируется соединением значений нескольких атрибутов, то такой ключ называется составным.

Атрибут представляющие собой копии ключей других отношений называется внешним ключом. Реляционная модель накладывает на внешние ключи ограничения для обеспечения целостности данных. Это означает, что к

каждому значению внешнего ключа должны соответствовать строки в связываемых отношениях.

В разрабатываемой БД сущность табельный номер сотрудника будет являться ключом для атрибутов сотрудники, отпуск, увольнение, командировка, трудовой договор и повышение квалификации (перевод).

Атрибут сотрудника так же имеет уникальные поля, такие как номер паспорта и ИНН, но номер паспорта не может быть ключом, так как номер паспорта может меняться, а ИНН может являться ключевым, но нам удобнее использовать как ключ табельный номер.

Для атрибута табель рабочего времени ключом будет являться две сущности, номер сотрудника и период, то есть ключ будет составным.

Нормализация – разбиение таблицы на две или более, обладающие лучшими свойствами включения, изменения или удаления данных. окончательная цель нормализации сводится к получению такого проекта БД в котором каждый факт появляется лишь в одном месте, то есть исключена избыточность информации.

Нормализация отношений – формальный аппарат ограничений, на формирование отношений которого позволяет устранить дублирование, обеспечить непротиворечивость хранимых в базе данных, уменьшить трудозатраты на ведение БД.

Кодом выведено три нормальные формы и предложен механизм, позволяющий любое отношение преобразовать к третьей нормальной форме. Приведем наши отношения к третьей нормальной форме.

Первая НФ: Отношение называется нормализованным или приведенным к первой нормальной форме тогда и только тогда, когда все его атрибуты простые (неделимые). Таблица находится в первой нормальной форме тогда и только тогда, когда ни одна из ее строк не содержит в любом ее поле более

одного значения, и не одно из ее ключевых полей не пусто. Для того чтобы привести наши отношения к первой нормальной форме надо сущность ФИО разбить на три отдельные (Фамилия, Имя, Отчество). Так же следует вынести в отдельную таблицу структурное подразделение, должности и наименование фирмы, чтобы не допустить избыточности данных. В отдельную таблицу выносятся приказы по личному составу и производственные приказы, так как нумерация у приказов общая. Атрибуты место проживания по паспорту и фактическое место проживания не требуют разбиения так как используются один раз.

Вторая НФ: Таблица находится во второй нормальной форме, если она удовлетворяет определению первой нормальной формы и все ее поля, не входящие в первичный ключ, связаны полной функциональной зависимостью с первичным ключом. Для того чтобы наши отношения привести во вторую нормальную форму надо вынести все начальников отдела в отдельную таблицу.

Третья НФ: Таблица находится в третьей нормальной форме, если она удовлетворяет определению второй нормальной формы и ни одно из ее не ключевых полей не зависит функционально от любого другого не ключевого поля. Отношения, представленные в данной БД приведены к третьей нормальной форме.

2 Разработка базы данных для кондитерских фабрик России

Назначением данной прикладной системы «База кондитерских фабрик России» является автоматизация обработки информации. Данная система является очень эффективной для упрощения и повышения надежности обработки и хранения информации.

В системе реализованы функции ввода, редактирования и просмотра и оценки данных о фабриках, магазинах, продукции. В запросах данных используются условия, что увеличивает гибкость системы.

В данной системе рассматривается предметная область «База данных кондитерских фабрик России». В ходе разработки базы данных были выделены такие объекты, как Фабрики, Продукция, Магазины, Города, Форма собственности, Поставляемая продукция. Объекты в базе данных представляются таблицами.

Таблица - Фабрика

Имя поля	Тип данных	Ключи
Название фабрики	Текстовый	
Год введения в строй	Дата/Время	
Телефон	Числовой	
ID_Фабрики	Счетчик	Первичный
ID_Города	Числовой	Вторичный

Таблица - Продукция

Имя поля	Тип данных	Ключи
Наименование	Текстовый	
Тип	Текстовый	
Расфасовка	Числовой	
ID_Фабрики	Числовой	Вторичный
ID_Продукции	Счетчик	Первичный

Таблица - Города

Имя поля	Тип данных	Ключи
Город	Текстовый	
ID_Города	Счетчик	Первичный
ID_Фабрики	Числовой	Вторичный

Таблица – Поставляемая продукция

Имя поля	Тип данных	Ключи
Объем	Числовой	
Цена	Денежный	
Продукция	Текстовый	
ID_Поставляемой продукции	Счетчик	Первичный
ID_Магазина	Счетчик	Вторичный

Таблица – Форма собственности

Имя поля	Тип данных	Ключи
Форма собственности	Текстовый	
ID_Формы собственности	Счетчик	Первичный
ID_Фабрики	Числовой	Вторичный

Базовые таблицы: Фабрики, Продукция, Магазины, Форма собственности, Города.

Таблицы-справочники: Поставляемая продукция

Связи между таблицами приведены на рисунке 1.



Рис. 1 - Схема данных

Максимальная цена поставленной продукции : запрос на выборку

The query editor shows the following query structure:

- Произция** (1) joined to **Магазин** (∞).
- Магазин** (1) joined to **Поставляе...** (∞).

The **Поставляе...** table is configured with the following settings:

Поле:	Номер магазина(ч)	Наименование	Цена
Имя таблицы:	Магазин	Произция	ПоставляемаяПро
Групповая операция:	Группировка	Группировка	Max
Сортировка:			
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:			

Рис. 2 - Максимальная цена поставленной продукции

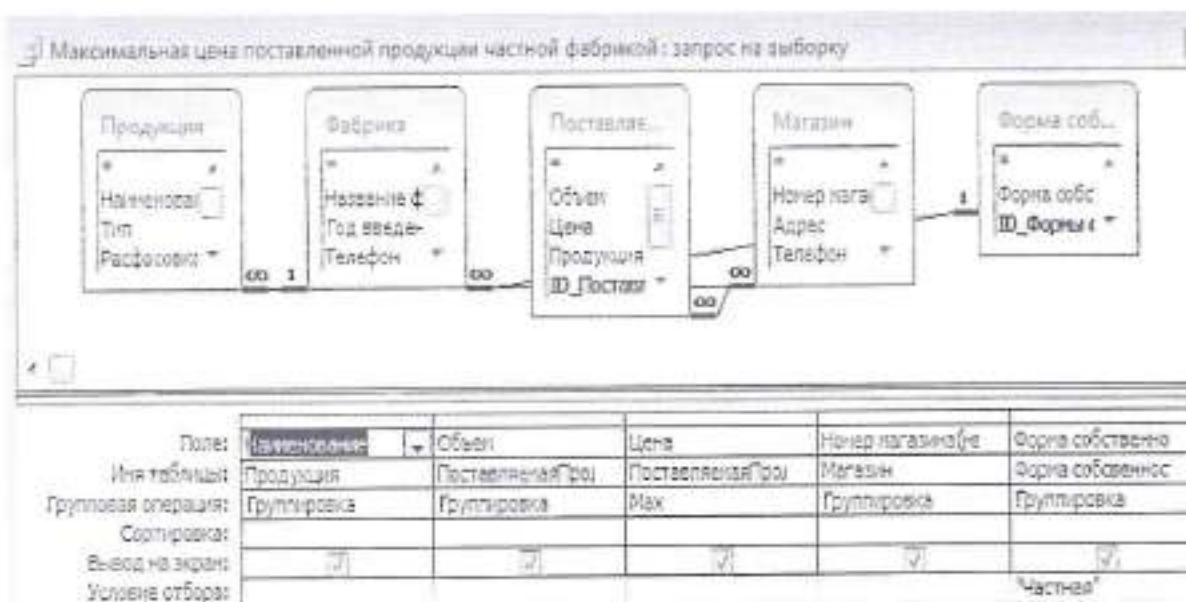


Рис. 3 - Максимальная цена у частной фабрики

Для связывания отдельных объектов в разработанной системе были использованы макросы, которые представляют собой небольшие программы на языке макрокоманд системы Microsoft Access.

В разработанной системе были применены макрокоманды на открытие и закрытие форм.

Главное меню : макрос

Имя макроса	Макрокоманда
Фабрика	ОткрытьФорму
Продукция	ОткрытьФорму
Поставленная прс	ОткрытьФорму
Город	ОткрытьФорму
Магазин	ОткрытьФорму
Форма собственнс	ОткрытьФорму
ФабрикаЗакреть	Закреть
ПродукцияЗакреть	Закреть
Поставленная прс	Закреть
ГородЗакреть	Закреть
МагазинЗакреть	Закреть
Форма собственнс	Закреть

Рис. 4 - Макросы

ЗАКЛЮЧЕНИЕ

Результаты работы представляют собой прикладную систему «База кондитерских фабрик России». В процессе создания была разработана концептуальная модель данных, которая сохраняет целостность данных, все таблицы приведены к третьей нормальной форме, определены связи между таблицами и условия на значения определенных полей. Создан графический интерфейс, представленный формами. Также разработаны различные запросы на выборку данных из таблиц, получение разнообразной информации о продукции, магазинах, фабриках. Объекты связаны между собой с помощью макросов.

Данная система является эффективным средством для упрощения и повышения надежности обработки информации о кондитерских фабриках России.

В качестве инструментария разработки использована СУБД Microsoft Accesses 2003. Для функционирования системы необходим компьютер IBM PC с процессором Intel Pentium и ОЗУ объемом не менее 32 Mb, операционная система Windows 95/98/XP, пакет Office 2003.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Медведкова, И.Е. Базы данных / И.Е. Медведкова, Ю.В. Бугаев, С.В. Чикунов. – Воронеж, 2014. – 6 с.
2. Гушин, А.Н. Базы данных: учебно-методическое пособие / А.Н. Гушин. – М, 2015. – 50 с.
3. Чурбанова, О.В. Базы данных и знаний. Проектирование баз данных в Microsoft Access: учебно-методическое пособие / О.В. Чурбанова, А.Л. Чурбанов. – Архангельск, 2015. – 73 с.
4. Карпова, Т.С. Базы данных: модели, разработка, реализация: учебное пособие / Т.С. Карпова. – М, 2016. – 96 с.
5. Щелокова, С.А. Базы данных: учебное пособие / С.А. Щелокова. – Оренбург, 2014. – 166 с.
6. Гушин, А.С. Базы данных: учебник / А.С. Гушин. – М, 2014. – 146 с.
7. Абросимова, М.А. Базы данных: проектирование и создание программного приложения в СУБД MS Access: практикум / М.А. Абросимова. – Уфа, 2014. – 43 с.
8. Черячукин, В.В. Право интеллектуальной собственности на программы для ЭВМ и базы данных в Российской Федерации и зарубежных странах: учебное пособие / В.В. Черячукин. – М, 2015. – 114 с.
9. Баженова, И.Ю. Основы проектирования приложений баз данных / И.Ю. Баженова. – М, 2016. – 46 с.
10. Королёв, В.Т. Технология ведения баз данных: учебное пособие / В.Т. Королёв, Е.А. Контарёв, А.М. Черных. – М, 2015. – 75 с.
11. Сирант, О.В. Работа с базами данных / О.В. Сирант, Т.А Коваленко. – М, 2016. – 32 с.
12. Щелоков, С.А. Разработка и создание баз данных средствами СУБД Access и SQL Server / С.А. Щелоков. – Оренбург, 2014. – 90 с.

13. Микляев, И.А. Универсальные объектно-ориентированные базы данных на реляционной платформе: Монография / И.А. Микляев. – Архангельск, 2014. – 180 с.
14. Рощина, Я.М. Основы моделирования экономического поведения домохозяйств на базе данных RLMS-HSE: лекции / Я.М. Рощина. – М, 2015. – 131 с.
15. Громов, Ю.Ю. Управление данными: учебник / Ю.Ю. Громов, О.Г. Иванова, А.В. Яковлев, В.Г. Однолько. – Тамбов, 2015. – 58 с.
16. Бессарабов, Н.В. Модели и смыслы данных в Cache и Oracle / Н.В. Бессарабов. – М, 2016. – 155 с.
17. Громов, Ю.Ю. Управление данными: учебное пособие / Ю.Ю. Громов, О.Г. Иванова, А.В. Яковлев, В.Г. Однолько. – Тамбов, 2014. – 56 с.
18. Войтович, И.Д. Нанозлектронная элементная база информатики. Качественно новые направления / И.Д. Войтович, В.М. Корсунский. – М, 2016. – 287 с.
19. Васюков, О.Г. Управление данными: учебно-методическое пособие / О.Г. Васюков. – Самара, 2014. – 61 с.
20. Уткин, В.Б. Информационные системы и технологии в экономике: учебник / В.Б. Уткин, К.В. Балдин. – М, 2015. – 73 с.