# Учет междугородних телефонных разговоров

# СОДЕРЖАНИЕ

ЗАДАНИЕ

ВВЕДЕНИЕ

I.Концептуальное проектирование

1.1Определение типов сущности

# 1.2 Определение типов связи

1.3Определение атрибутов и связывание их с типами сущностей и связей

1.4 Определение доменов атрибутов

1.5 Определение атрибутов, являющихся потенциальными, первичными и внешними ключами

1.6 Создание диаграммы "сущность — связь"

II. Логическое проектирование

2.1 Проверка связей

2.2 Проверка моделей с помощью правил нормализации

2.3 Построение окончательной диаграммы "Сущность- связь"

ЗАКЛЮЧЕНИЕ

СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

**ЗАДАНИЕ**

**Задание на выполнение курсовой работы** по дисциплине "Базы данных"

Спроектировать и разработать базу данных для предметной области "Учет междугородних телефонных разговоров", содержащей следующие атрибуты:

***Заказчик разговора***:Код абонента, Ф. И. О., Адрес, № телефона заказчика, Дата, Время.

***Оператор исходящего пункта***: Код заявки, Ф. И. О, Личный №, Связь прямая или нет, Зона, Тариф.

***Оператор входящего пункта***: Код заявки, Ф. И. О, Личный №,, Город приема заказа.

***Данные вызываемого абонента***: Ф. И. О., Адрес, № телефона.

***Заявки***: Код абонента, Код заявки, Вызываемый номер.

Разрабатываемая база данных должна соответствовать следующим бизнес-правилам:

1.         Заказчик может совершить звонок посредствам электронной АТС непосредственно напрямую вызываемому абоненту.

2.         Заказчик может воспользоваться услугами оператора телекома.

3.         Оператор исходящего пункта может соеденить закозчика напрямую с вызываемым абонентом либо воспользоваться услугами оператора телекома города вызываемого абонента.

Обеспечить реализацию следующих запросов:

1) Какой абонент, какому абоненту звонил;

2)Кто сколько наговорил.

**ВВЕДЕНИЕ**

Современный мир информационных технологий трудно представить себе без использования баз данных.

База данных — это совокупность данных, отражающая состояние объектов и их отношений в рассматриваемой предметной области.

Практически все системы в той или иной степени связаны с функциями долговременного хранения и обработки информации. Информация становится фактором, определяющим эффективность любой сферы деятельности. Увеличились информационные потоки и повысились требования к скорости обработки данных, и теперь уже большинство операций не может быть выполнено вручную, они требуют применения наиболее перспективных компьютерных технологий.

Любые административные решения требуют четкой и точной оценки текущей ситуации и возможных перспектив ее изменения. И если раньше в оценке ситуации участвовало несколько десятков факторов, которые могли быть вычислены вручную, то теперь таких факторов сотни и сотни тысяч.

База данных - это организованная структура, предназначенная для хранения информации.

Эти системы обеспечивают получение общих или детализированных отчетов по итогам работы; позволяют легко определять тенденции изменения важнейших показателей; обеспечивают получение информации, критической по времени, без существенных задержек; выполняют точный и полный анализ данных.

MS Access входит в состав самого популярного пакета Microsoft Office. Основные преимущества: знаком многим конечным пользователям и обладает высокой устойчивостью данных, прост в освоении, может использоваться непрофессиональным программистом, позволяет готовить отчеты из баз данных различных форматов.

Целью данной курсовой работы является разработка базы данных для автоматизации учета междугородних телефонных разговоров между абонентами как через оператора связи, так и напрямую с вызываемым абонентом.

**I.    КОНЦЕПТУАЛЬНОЕ ПРОЕКТИРОВАНИЕ**

Процесс концептуального проектирования - это когда администратор данных решает, какая именно информация должна храниться в базе данных, то есть указывает те типы сущностей, в которых заинтересован конечный пользователь, а также определяет диапазон информации об этих сущностях, которую необходимо записать.

Цель концептуального проектирования: создание локальной концептуальной модели данных предприятия на основе представления о предметной области каждого отдельного типа пользователей. Представление пользователя включает в себя данные, необходимые конкретному пользователю для принятия решений или выполнения некоторого задания. Пользователем может быть как отдельный работник, так и группы лиц. Обычно представление пользователя отражает некоторую функциональную область в общем поле деятельности предприятия. Представление пользователя может быть построено в виде диаграммы потоков данных, определяющих функциональные области и индивидуальные функции.

Каждая локальная концептуальная модель включает следующее:

1. типы сущности;

2. типы связи;

3. атрибуты;

4. домены атрибутов;

5. потенциальные ключи;

6. первичные ключи.

Эти модели называются инфологическими или семантическими и отражают естественный и удобный для разработчиков и других пользователей формы.

Инфологическая модель применяется на втором этапе проектирования БД, то есть после словесного описания предметной области, потому что процесс проектирования длительный, он требует обсуждений с заказчиком, со специалистами в предметной области. При разработке серьезных корпоративных информационных систем проект базы данных является тем фундаментом, на котором строится вся система в целом. Следовательно, инфологическая модель должна включать такое формализованное описание предметной области, которое легко будет "читаться" не только специалистами по базам данных. И это описание должно быть настолько емким, чтобы можно было оценить глубину и корректность проработки проекта БД.

В настоящий момент именно модель Чена "сущность—связь" стала фактическим стандартом при инфологическом моделировании баз данных. Общепринятым стало сокращенное название ER-модель. Модель "Сущность - связь" в наибольшей степени согласуется с концепцией объектно-ориентированного проектирования, которая в настоящий момент, несомненно, является базовой для разработки сложных программных систем.

**1.1      Определение типов сущности**

**Цель:** определение основных типов сущности, присутствующих в представлении данного пользователя о предметной области приложения. На этом этапе определяются основные объекты, которые могут интересовать пользователя. Эти объекты являются типами сущности, входящих моделей.

Сущность-это реальный или воображаемый объект, имеющий существенное значение для рассматриваемой предметной области.

Сущность имеет имя, уникальное в пределах моделируемой системы. Так как сущность соответствует некоторому классу однотипных объектов, то предполагается, что в системе существует множество экземпляров данной сущности. Объект, которому соответствует понятие сущности, имеет свой-набор атрибутов — характеристик, определяющих свойства данного представителя класса. При этом набор атрибутов должен быть таким, чтобы можно было различать конкретные экземпляры сущности.

Одно из общепринятых графических обозначений сущности — прямоугольник, в верхней части которого записано имя сущности, а ниже перечисляются атрибуты, причем ключевые атрибуты помечаются, например, подчеркиванием или специальным шрифтом.

Каждая сущность должна обладать некоторыми свойствами:

1. иметь уникальное имя;

2. обладать одним или несколькими атрибутами, которые либо принадлежат сущности, либо наследуются через связь;

3.обладать одним или нескольким атрибутами, которые однозначно идентифицируют каждый экземпляр сущности;

4. каждая сущность может обладать любым количеством связи с другими сущностями данной модели.

После выделения каждой сущности ей следует присвоить некоторое осмысленное имя, понятное пользователю. Если сущность известна пользователю под разными именами, все дополнительные имена рекомендуется определить как алиасы, также задокументировать.

|  |  |  |  |
| --- | --- | --- | --- |
| Тип сущности | Описание | Псевдонимы | Особенности использования |
| Заказчик | Сведения о заказчике | Заказчик | Заказчик может заказать несколько звонков |
| Получатель | Сведения о получателе | Получатель | Получатель может получать звонки от разных заказчиков |
| Заявка | Информация о звонке | Заявка, телефонный звонок | Заявка может поставляться несколькими заказчиками, каждый вид заявки может, каждая заявка может быть получена несколькими получателями. |

Рис.№1. Сведения о типах сущностей

- Чтобы идентифицировать типы сущностей необходимо изучить представление пользователей и извлечь все используемые в них существительные, или сочетание существительного и прилагательного. После чего выбираются самые крупные и важные объекты для нашей предметной области.

1.2      **Определение типов связи**

**Цель:** определение важнейших типов связи, существующих между сущностями, выделенными на предыдущем этапе. Для этого из представлений пользователей выбираются все выражения, в которых содержатся глаголы.

Связь **-** это ассоциирование сущности. Сущности, участвующие в связи, называются участниками.

Между сущностями могут быть установлены: бинарные ассоциации, показывающие, каким образом сущности соотносятся или взаимодействуют между собой. Связь может существовать между двумя разными сущностями или между сущностью и ей же самой (рекурсивная связь). Она показывает, как связаны экземпляры сущностей между собой. Если связь устанавливается между двумя сущностями, то она определяет взаимосвязь между экземплярами одной и другой сущности. Графическая интерпретация связи позволяет сразу прочитать смысл взаимосвязи между сущностями, она наглядна и легко интерпретируема. Связи делятся на три типа по множественности:

1. один-к-одному (1:1),

2. один-ко-многим (1:М),

3. многие-ко-многим (М:М).

Связь "один-к-одному" означает, что экземпляр одной сущности связан только с одним экземпляром другой сущности. Связь "один-ко-многим"(1: М) означает, что один экземпляр сущности, расположенный слева по связи, может быть связан с несколькими экземплярами сущности, расположенными справа по связи, а связь "многие-к-одному" (M:1) означает, что один экземпляр первой сущности может быть связан с несколькими экземплярами второй сущности, и наоборот, один экземпляр второй сущности может быть связан с несколькими экземплярами первой сущности.

Определим типы связей наших сущностей. Данные представлены в таблице, па рисунке №2.

|  |  |  |  |
| --- | --- | --- | --- |
| Тип сущности | Связь | Тип сущности | Кардинальность связи |
| Заказчик | Заказывает | Телефонный звонок | 1:М |
| Получатель | Получает | Телефонный звонок | М:1 |

Рис.№2. Сведения о типах связей

**1.3      0пределение атрибутов и связывание их с типами сущностей и связей**

**Цель:** связывание атрибутов с типами сущности и связи.

Атрибут - это любая характеристика сущности значимая для рассматриваемой предметной области и предназначенная для квалификации, идентификации, классификации, количественной характеристики или выражения состояния сущности.

Кроме того, в ER-модели допускается принцип категоризации сущностей. Это значит, что, как и в объектно-ориентированных языках программирования, вводится понятие подтипа сущности, то есть сущность может быть представлена в виде двух или более своих подтипов — сущностей, каждая из которых может иметь общие атрибуты и отношения и/или атрибуты и отношения, которые определяются однажды на верхнем уровне и наследуются на нижнем уровне.

Атрибуты бывают:

- простые;

- составные - состоят из простых атрибутов;

- однозначные - атрибуты, которые могут принимать единственное значение;

- многозначные - атрибуты, которые могут принимать много значений;

- производные - это атрибуты, значение которых могут быть установлены с помощью значений других атрибутов.

Необходимо выбрать все данные, описывающие сущности и связи, выделенные в создаваемой модели базы данных. Выбранные существительные представляют собой атрибут в том случае, если они описывают свойство, качество или характеристику некоторой сущности или связи.

Далее атрибуты документируют, то есть каждому выделенному атрибуту присваивают осмысленное имя. О каждом атрибуте в документацию помещают следующие сведения:

1. имя атрибута и его описание;

2. любые алиасы или синонимы, имеющиеся для данного атрибута;

3. тип данных и размерность значений;

4. значение, принимаемое для атрибута по умолчанию, если таковое имеется;

5. является ли атрибут обязательным;

6. является ли атрибут составным;

7. является ли атрибут производным;

8. является ли атрибут многозначным.

Сведения об атрибутах представлены в таблице на рис.№3.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Атрибуты | Тип данных | Простой | Синонимы | Описание |
| Заказчик разговора | | | | |
| Код абонента | счётчик |  |  | Код абонента, заказчика разговора |
| ФИО | текстовый |  | Личные данные абонента | Личные данные абонента |
| Адрес | Текстовый | - | Адрес абонента | Адрес абонента |
| №\_телефона\_заказчика | Числовой |  | №\_телефона\_заказчика | №\_телефона\_заказчика |
| Дата | Дата/время |  | Дата | Дата звонка |
| Время | Дата/время |  | Время | Время звонка |

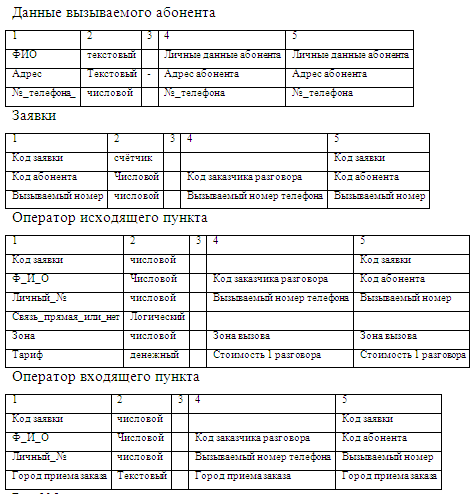


Рис.№3.

**1.4 0пределение доменов атрибутов**

**Цель:** определение доменов для всех атрибутов, присутствующих в каждой локальной концептуальной модели данных.

Домен - это набор значений, которые могут быть присвоены атрибутам сущности. Домены должны содержать следующие данные:

1. набор допустимых значений для атрибутов;

2. сведения о размере и формате каждого из полей атрибутов.

После выделения всех имеющихся доменов их документируют, присваивают осмысленные имена.

Сведения о доменах атрибутов представлены в таблице на рисунке №4.

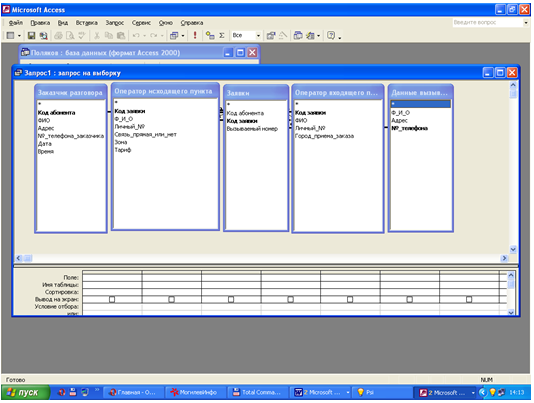


Рис.№4

**1.5      Определение атрибутов, являющихся потенциальными, первичными и внешними ключами**

**Цель:** определение всех потенциальных ключей для каждого типа сущности и если таких ключей окажется несколько выбор среди них первичного ключа. Также необходимо выделить внешние ключи для каждой сущности, если они имеются.

Выделяют следующие виды ключей:

1. потенциальный ключ - это атрибут, набор атрибутов, однозначно идентифицирующих каждый экземпляр сущности;

2. первичный ключ - потенциальный ключ, который выбран для идентификации экземпляров внутри сущностей (потенциальные ключи, не выбранные в качестве первичных, называются альтернативными);

3. внешний ключ - это атрибут или группа атрибутов дочерней сущности, которые соответствуют первичному ключу родительской сущности.

4. составной ключ - первичный ключ, состоящий из нескольких атрибутов.

При выборе первичного ключа необходимо принимать во внимание следующие рекомендации:

1. использовать потенциальный ключ с минимальным набором атрибутов;

2. использовать тот потенциальный ключ, вероятность изменения значений которого минимальна;

3. выбирать тот потенциальный ключ, который имеет минимальную вероятность потери уникальности значений в будущем;

4. остановить свой выбор на потенциальном ключе, с которым будет проще всего работать с точки зрения пользователей.

После выбора первичных, альтернативных и внешних ключей сведения о них документируются.

**1.6Создание диаграммы "сущность — связь"**

**Цель:** разработка ER - диаграммы, содержащей концептуальное отражение представлений пользователя о предметной области приложения.

Существует множество подходов к построению таких моделей: графовые модели, семантические сети, модель "сущность-связь" и т.д. Наиболее популярной из них оказалась модель *"сущность-связь "* или называемая ещё *ER-моделью.*

Моделирование предметной области при помощи модели "сущность-связь" базируется на использовании графических диаграмм.

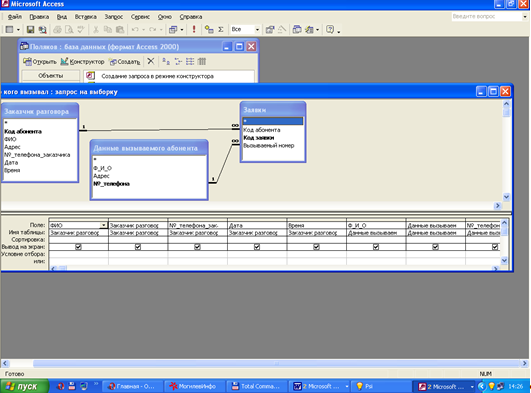


Рис.№5

**П. Логическое проектирование**

Логическое проектирование базы данных представляет собой процесс конструирования моделей информационной структуры предприятия, выполняемые в соответствии с требованиями выбранной схемы организации информации. Однако создаваемая логическая модель не зависит от особенностей конкретных СУБД и других физических условий реализации.

Основой анализа корректности схемы являются так называемые функциональные зависимости между атрибутами базы данных. Некоторые зависимости между атрибутами отношений являются нежелательными из-за побочных эффектов и аномалий, которые они вызывают при модификации базы данных. При этом под процессом модификации базы данных мы понимаем внесение новых данных в БД или удаление некоторых данных из БД, а также обновление значений некоторых атрибутов.

**Цель:** построение логической модели данных на основе концептуальной модели данных, отражающей представление отдельного пользователя о предметной области приложения и проверка полученной модели с помощью методов нормализации и контроля выполнения транзакции.

На данном этапе необходимо проверить связи, проверить модели с помощью правил нормализации и построить окончательную диаграмму сущность-связь.

**2.1 Проверка связей**

Цель: построение логической модели данных на основе концептуальной модели данных с целью удаления из них нежелательных элементов и преобразование полученных моделей в локальные логические модели данных.

Первым пунктом является удаление связи " многие ко многим". Удалить эту связь можно путем ввода промежуточной сущности. Такая связь х в моей курсовой работе отсутствует, так как была сразу выделена в качестве отдельной сущность "марка".

При введении промежуточной сущности атрибутами этой сущности будут являться первичные ключи связываемых сущностей.

Вторым пунктом является удаление сложных связей - связей существующих между тремя и более типами сущности. Такие связи в моей модели отсутствуют.

Теперь необходимо удалить рекурсивные связи - это связи, в которых одни и те же сущности участвуют несколько раз и в разных полях. Таких связей в мною рассматриваемой модели нет.

Четвертым пунктом является удаление связи с атрибутами, то есть связи, имеющей дополнительные характеристики, таких связей в данной работе также не наблюдается.

Далее необходимо удалить множественные атрибуты, если они имеются. В данном случае их нет.

Также следует проверить связи типа один-к-одному. В разрабатываемой мною модели такие связи вообще отсутствуют.

Заключительным пунктом является удаление избыточности связи. Связь считается избыточной, если одна и та же информация может быть получена не только через нее, но и при помощи другой связи. Сделав анализ данных можно сказать, что таких связей в данной работе также не наблюдается.

Пример выборки по связям на рисунке 6:

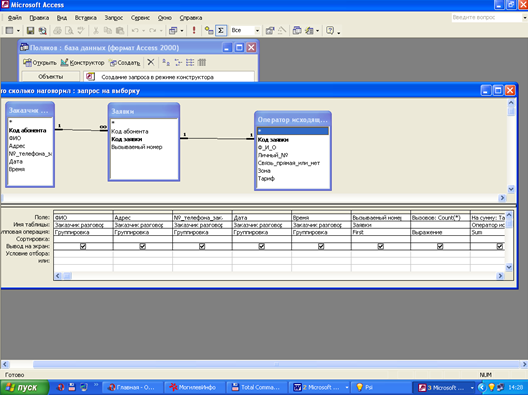


Рис.№6.

**2.2 Проверка моделей с помощью правил нормализации**

**Цель:** проверка локальной логической модели данных с использованием технологии нормализации. Технология проектирования реляционных баз данных связано с теорией нормализации, основанной на анализе функциональных зависимостей между атрибутами отношений. Понятие функциональной зависимости является фундаментальным в теории нормализации реляционных баз данных. Функциональные зависимости определяют устойчивые отношения между объектами и их свойствами в рассматриваемой предметной области. Именно поэтому процесс поддержки функциональных зависимостей, характерных для данной предметной области, является базовым для процесса проектирования. Нормализация - это метод создания набора отношений с заданными свойствами на основе требуемых данных, установленных некоторой организацией. Это формальный метод анализа отношений на основе первичного ключа и существующих функциональных зависимостей. Он включает ряд правил, которые могут использоваться для проверки отдельных отношений таким образом, чтобы вся БД была нормализована до желаемой степени нормализации. В теории реляционных баз данных обычно выделяется следующая последовательность нормальных форм:

1. 1 нормальная форма

2. 2 нормальная форма

3. 3 нормальная форма.

1) Отношение находится в первой нормальной форме тогда и только тогда, когда на пересечении каждого столбца и каждой строки находятся только элементарные значения атрибутов;

Существует два подхода исключения повторяющихся групп из ненормализованных таблиц:

- в первом случае повторяющиеся группы удаляются путем ввода соответствующих данных в пустые столбцы строк с повторяющимися данными. Этот подход называется выравниванием таблицы.

- во втором случае один атрибут или группа атрибутов назначаются ключом ненормализованной таблицы, а затем повторяющиеся группы изымаются и помещаются в отдельные отношения с копиями ключа исходной таблицы.

2) Отношение находится во второй нормальной форме тогда и только тогда, когда оно находится в первой нормальной форме и не содержит неполных функциональных зависимостей не первичных атрибутов от атрибутов первичного ключа.

" Функциональная зависимость - описывает связь между атрибутами отношения. Если в отношении R, содержащем атрибуты А и В, атрибут В функционально зависит от атрибута А (А—>В), то каждое значение атрибута А связано только с одним значением атрибута В.

В некотором отношении R атрибут В называют полностью функционально зависимым от атрибута А, если атрибут В функционально зависит от полного значения атрибута А и не зависит ни от какого подмножества полного значения атрибута А. Вторая нормальная форма применяется только для отношений с составными первичными ключами, а в моей БД не имеется отношений с составными первичными ключами.

3) Отношение находится в третьей нормальной форме тогда и только тогда, когда оно находится во второй нормальной форме и не содержит транзитивных зависимостей.

Если для атрибутов А, В, С некоторого отношения R существует зависимость вида А—>В, В—"С, то говорят, что атрибут С транзитивно зависит от атрибута А через атрибут В.

Все мои отношения находятся в первой, второй и третьей нормальной форме. В моей БД таких зависимостей не обнаружено.

**2.3 Построение окончательной диаграммы "Сущность- связь"**

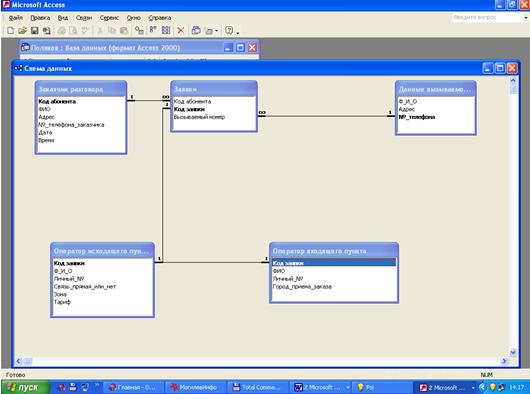


Рис.№7.

**Заключение**

СУБД обеспечивает полный контроль над процессом определения данных их обработкой совместных использований. Все эти функциональные возможности в полной мере реализуются в программе MS Access, в ней предусмотрены все необходимые средства для определения и обработки данных, а также для управления ими при работе с большими объемами информации.

В данной курсовой работе разработана база данных для автоматизации \* учета в автомобильном салоне, для учета поступления автомобилей с завода, для учета покупки и продажи марок автомобилей покупателю. Также в ней содержится информация о покупателях, дилерах, заводе - изготовителе.

Разработка базы данных началась с определения объектов, которые представляют наибольший интерес для пользователей, то есть с выделения типов сущностей. Далее дается характеристика типов связей, которые присутствуют между данными сущностями. Третьим пунктом было определение атрибутов, входящих в ту или иную сущность и связывание их с типами сущностей и связей. Было представлено подробное описание доменов атрибутов. Отдельным пунктом выделили определение атрибутов, являющихся потенциальными, первичными и внешними ключами. На основании этих данных построена диаграмма "сущность - связь".

На втором этапе была сделана проверка связей, и проверка моделей с помощью правил нормализации. И, наконец, заключительным пунктом данной курсовой работы было построение окончательной диаграммы "сущность - связь" с указанием всех имеющихся атрибутов и промежуточных сущностей.

В заключении хочу добавить, что современные базы данных являются основой многочисленных информационных систем. Информация, накопленная в них, является чрезвычайно ценным материалом, и в настоящий момент широко распространяются методы обработки баз данных с точки зрения извлечения из них дополнительных знаний, методов, которые связаны с обобщением и различными дополнительными способами обработки данных. Базы данных в данной концепции выступают как хранилища информации, при необходимости к которым всегда можно обратиться

**Список использованной литературы**

1. Базы данных. Учебник под ред. А.Д, Хомоненко - СПб.: " Корона принт", 2000

2. Вейскос Дж. Эффективная работа с MS Access 2000 - СПб.: Питер, 2001

3. Глушаков СВ., Ломотько Д, В. Базы данных. Уч. Курс- Харьков: "Фо-пио";2000

4. Дейт К. Дж. Введение в систему баз данных - 7-е издание-М.: Виль-ям,2001

5. Диго СМ. Проектирование и использование баз данных. Учебник - М.: Финансы и статистика, 1995

6. Карпова Т.С: Базы данных; Модели, разработка, реализация. - СПб.: Питер- 2000