Государственный комитет Российской Федерации

 по Высшему Образованию

Московский государственный институт радиотехники,

электроники и автоматики

(Технический университет)

**факультет**: Кибернетики

**кафедра**: ИТС

**Курсовой проект**

**по дисциплине**: Инженерия Знаний

**на тему**: Представление и использование знаний об объектах.

Представление и использование знаний об отношениях между объектами. Организация обобщенной информации на основе использования мультииерархической таксономической структуры.

**группа разработчиков:** Частухин Виталий

Рыбина Марина

Юданова Ирина

**группа**: АИ-1-93

**руководитель**: Потресова Галина Дмитриевна

Москва 1996

Содержание

Задание.............................................................................................................3

Введение...........................................................................................................4

Формулировка цели КПР..............................................................................4

1. Теоретический раздел.................................................................................5

2. Практический раздел................................................................................24

2.1 Ограничения на использование

 разработанного программного обеспечения........................................24

2.2 Необходимые знания для работы в MS ACCESS2.0.

 Краткий экскурс в Access..........................................................................24

2.3 Планирование БД...................................................................................27

2.4 Создание таблиц для ваших данных....................................................29

2.5 Ввод данных в ваши таблицы..............................................................33

2.6 Связи между таблицами.........................................................................33

3. Варианты заданий по л.р.........................................................................35

4. Содержание отчета о л.р..........................................................................36

5. Контрольные вопросы.............................................................................36

6. Выполнение л.р. для одного из вариантов задания.............................36

Заключение.....................................................................................................39

Библиографический список..........................................................................40

Задание на КПР

по дисциплине: Инженерия Знаний

студентам: Частухину Виталию, Рыбиной Марине,

 Юдановой Ирине, группа: АИ-1-93

на тему: Представление и использование знаний об объектах.

Представление и использование знаний об отношениях между объектами. Организация обобщенной информации на основе использования мультииерархической таксономической структуры.

1. Перечень разрабатываемых вопросов

1.1 Формулировка цели КПР

1.2. Составление вариантов заданий и определение порядка выполнения л.р.

1.3 Составление библиографического списка по тематике КПР

1.4 Подготовка теоретического материала по тематике КПР

1.5 Составление контрольных вопросов

1.6 Разработка методики выполнения практической части работы

1.7 Определение содержания отчетов о работе

1.8 Разработка необходимого программного и информационного

 обеспечения,необходимого для выполнения л.р.

1.9 Выполнение и оформление л.р. для одного ищз вариантов задания

2. Календарный план - график

2.1 Пункты 1.1, 1.2, 1.3 к 20.02

2.2 Пункты 1.4, 1.5 к 5.03

2.3 Пункты 1.6, 1.7 19.03

2.4 Пункт 1.8 к 9.04

2.5 Пункт 1.9 к 16.04

3. Задание выдано 13.02.96 г.

Введение

За последние тридцать лет теория баз данных (БД) стала одним из основных разделов информатики, оказывающим сильное воздействие на развитие архитектуры вычислительных систем, современную информационную технологию и методологию проектирования автоматизированных информационных систем (АИС), особенно банков данных (БнД). В последние годы в теории БД сложилось новое направление научных исследований, получившее название семантического, или концептуального, моделирования в БД. Его основная цель - интеллектуализация банков данных и организация интерфейса конечного пользователя с информационной системой на уровне представлений о предметной области, а не на уровне структур данных.

Первоначально в теории БД основное внимание уделялось средствам эффективной организации данных и манипулирования ими. Так возникли получившие всеобщее распространение три "великих", по выражению Дж. Ульмана, модели данных: иерархическая, реляционная и сетевая. Одновременно стало ясным, что эффективность новых информационных систем - банков данных - зависит не только от характеристик технических средств и программного обеспечения, но и от эффективности взаимодействия человек - машина. Перенос акцента на конечного пользователя и привел в конечном счете к формированию нового направления исследований в теории БД, непосредственно связанного с задачей о представлении знаний.

В настоящее время определился основной подход к решению задач семантического моделирования в базах данных. Он заключается в выделении двух уровней моделирования: уровня концептуального моделирования ПО (предметной области) и уровня моделирования собственно базы данных. На верхнем уровне осуществляется переход от неформализованного описания ПО и информационных потребностей конечного пользователя к их формальному выражению с помощью специальных языковых средств. Обсуждению возможных конструкций концептуальной модели ПО посвящен теоретический раздел данного курсового проекта.

Формулировка цели КП

Целью данного курсового проекта является разработка теоретического и практического материала для обеспечения доступного изучения соответствующих разделов по дисциплине Инженерия Знаний, а именно: представления знаний об объектах, представлении и использовании знаний об отношении между объектами, а также организации обобщенной информации на основе использования мультииерархической таксономической структуры. Практические разработки данного КП (структуры баз данных в СУБД Microsoft Access 2.0, а также приведенная методика создания этих структур в Access 2.0) могут являться материалом для выполнения лабораторных работ по дисциплине Инженерия Знаний при условии наличия необходимых технических средств.

1. Теоретический раздел

Инженерия знаний - область информатики, в которой исследуются методы представления знаний, манипулирования ими и поддержания их в актуальном состоянии.

Информационная система - организованная совокупность технических и обеспечивающих средств, технологических процессов и кадров, реализующих следующие основные функции:

-сбор информации;

-хранение информации;

-поиск и обработка информации;

-передача информации;

 Модель ПО

 (понятийное знание)

 Система управления

 базами знаний

 Пакеты База данных

прикладных

 программ

(процедурные

 знания) Предметное знание

 рис.1

Краеугольным камнем исследований последнего времени является понятие "предметная область банка данных" (ПО) (enterprise, universe of discourse).

Совокупность объектов, о которых можно задавать вопросы, образует объектное ядро предметной области. Здесь слово "объект" употреблено как синоним слова "реалия"; не предполагается, что объекты обязательно имеют материальную, "вещную" природу. Термин "объект" также является первичным, неопределяемым понятием. (См. рис.2а) Синонимами, помимо употребленного выше слова "реалия", служат слова "сущность" (entity), "вещь" (thing). Примеры объектов - отдельный студент, группа студентов, аудитория, время занятий, слова, числа и т.д. Обычно считается, что быть объектом - значит быть дискретным и различимым. По - видимому, примеры "необъектов" - это мир, время, материя, смысл и т.п., хотя можно представить ИС, в которой хранятся сведения об этих категориях.

 **Объект Предмет**

 первичное

 неопределяемое модель объекта

 понятие

 в качестве объектов

 могут быть

 абстракция

 не только физические,

 но и  **объекты**

 **мышления**

 рис. 2а рис.2б

Часто слово "объект" считается близким по смыслу слову "предмет". Однако представляется удобным слово "предмет" использовать в том случае, когда объект, существующий вне нас, становится носителем определенной совокупности свойств и входит в различные взаимоотношения, которые представляют интерес для потребителей информации, хранящейся в АИС. Другими словами, предмет - это объект, ставший объектом рассмотрения, наблюдения, ставший носителем определенных свойств. Один и тот же объект воспринимается разными системами как разные предметы. Предмет является результатом абстракции реального объекта, результатом огрубления действительности, при котором игнорируется бесконечное многообразие свойств и взаимодействий объекта. Таким образом, предмет - это модель реального объекта: если объект имеет онтологический статус, то предмет - гносеологический. (См. рис.2б)

Объекты могут быть простыми и сложными. Простыми называются объекты, которые для данной предметной области считаются бесструктурными. Простые объекты представляются совокупностью характеризующих их свойств. Внутренняя структура простого объекта не раскрывается.

Сложные объекты имеют определенный состав и структуру. Например:

 Студент Иванов учится в группе АИ-2-93

*сложный объект*

*Способы идентификации объектов:*

1. Прямая идентификация. Основана на использовании *имен объектов.*

2. Косвенная идентификация. Основана на указании *свойств объекта*. В БД косвенная идентификация основывается на использовании *ключей*.

*Ключ -* набор полей записи, значения которых однозначно определяют запись файла.

В конкретный момент времени состояние предметной области определяется совокупностью объектов и ситуаций, имеющих место в данный момент времени.

Пространство состояний предметной области - это класс всех возможных конечных множеств объектной и системной предметной области.

Для описания динамики предметной области нужно рассмотреть траекторию предметной области в пространстве состояний.

Семантика предметной области определяется как общее свойство всех возможных траекторий в пространстве состояний.

На рис.3 предложена классификация ситуаций. Эта классификация рассматривает свойства как специальный случай ситуаций. Одновременно она вводит еще одно важное понятие - понятие времени, которое явно или неявно присутствует при выделении подклассов указанной иерархии. Время выступает как множество отдельных моментов и как множество интервалов (временное состояние, негомогенный процесс). В явном виде структуры этих множеств не фиксированы, хотя отношение следования между моментами времени считается известным.

Следует различать реальное время внешнего мира и его отражение в ИС и в источниках информации. Внутри системы зависящие от времени взаимосвязи фокусируются только после их регистрации источниками и передачи сообщений информационной системе.

Другое понятие, также неявно участвующее в указанной классификации, - это понятие пространства. "Постоянными являются все свойства и соотношения, вытекающие из принадлежности объектов к той или иной таксономической категории..., свойства типов объектов, а также отношения кровного родства, неотъемлемой принадлежности, относительное расположение неподвижных предметов ..., свойства и отношения абстрактных объектов, не локализованные в пространстве". В приведенном высказывании хочется обратить внимание еще на два обстоятельства: для лингвиста несомненно наличие таксономических категорий объектов, которые обладают собственными свойствами, и наличие общих свойств у объектов одной таксономической категории. Эти само собой разумеющиеся обстоятельства широко используются в семантических моделях баз данных.

 ситуации

 статические динамические

 состояния постоянные процессы события

 свойства и

 отношения

устойчивые временные результаты происшествия

 рис.3

Ситуации сами являются объектами, поэтому могут обладать различными свойствами.

Введя пространство состояний, можно рассматривать в нем определенные *траектории* или последовательности состояний γ0, γ1, ..., γt, ... , в которых находится ПО в моменты времени 0, 1, ..., t, ... Члены такой последовательности не могут быть совершенно произвольными, поскольку состояние γt обычно как - то связано с предшествующими состояниями γ0, γ1, ..., γt-1. Поэтому предметную область можно определить как класс всех действительно возможных последовательностей состояний. Такие последовательности называются *траекториями ПО*. Совокупность всех общих свойств траекторий называется *семантикой предметной области*.

Итог проведенных рассмотрений подведем следующей диаграммой, классифицирующей объекты в предметной области и дополняющей рис.3.

 Объекты ПО

материальные нематериальные

 имена понятия ситуации

 рис.4

*Концептуальные средства описания предметной области*

Эти средства обладают абстрактностью и универсальностью. В силу своей абстрактности средства описания стали называться *концептуальными*. Универсальность определяется общностью систем базовых понятий и средствами порождения новых понятий.

В исследованиях по искусственному интеллекту проблема описания предметной области трактуется как проблема представления знаний, а соответствующие концептуальные средства называются *схемой представления*.

В теории БД принято говорить о концептуальном, или информационно - логическом, моделировании ПО. Центральным понятием является понятие концептуальной схемы (модели) ПО.

В силу свойства универсальности, описать средства построения концептуальной схемы - это значит описать возможные состояния абстрактной предметной области, которая предназначена для моделирования структуры и динамики произвольных ПО.

Понятия:

1. *Тип объекта* - должен иметь уникальное имя и все объекты предметной области могут быть отнесены к одному или нескольким типам.

τ0 - тип объекта;

τ0t - множество объектов типа τ в момент времени t;

Эти множества могут пересекаться, если объект имеет несколько типов.

Тип объекта может сообщаться системе при первом упоминании этого объекта.

Тип объекта может определяться системой на основании свойств объекта с помощью классифицирующих процедур.

Тип объекта может быть автоматически изменен при изменении состояния объекта.

2. *Концептуальная схема* предметной области - множество типов, снабженное некоторой структурой. Структура на множестве типов задается множеством отношений и операций, определенными на множестве типов.

Обозначим ϕ конечные множества объектов предметной области.

Для некоторого момента времени t можно выделить множество объектов типа (τ1)obt и (τ2)obt, имеющих определенные соотношения, например:

 (τ1)obt ⊂ (τ2)obt

 (τ1)obt ∩ (τ2)obt = 0

Если такие соотношения являются устойчивыми (не меняются во времени), то эти свойства состояния семантической предметной области.

Соотношения:

1. Родовидовое отношение SUP

τ1 sup τ2 - τ1 является подтипом τ2 ((вид) является подтипом (род))

 Это соотношение задается между типами объектов. Оно верно, если

 ∀ t (τ1)obt ≤ (τ2)obt

Эти отношения принято задавать графически. Типы объектов представляются вершинами, а отношения между ними - ребрами.

Например:

 sup Помещение sup

 Учебное Служебное

 помещение помещение

sup sup sup sup

аудитория лаборатория помещение помещение

 кафедры деканата

 рис.5

Следующая диаграмма описывает набор базовых типов объектов для любой концептуальной модели (См. рис. 6).

 Объект

 sup sup

Неопределенный Определенный

 объект sup объект

 sup

Материальный Нематериальный

 объект объект

 sup sup sup sup sup

 имя время понятие ситуация пространство

 sup sup

 статическая динамическая

 sup sup

 пост. сит. сост.

 sup sup

 рис.6 уст. сост. врем. сост.

Тип объекта является общим для всех предметных областей.

2. Отношение частичного порядка PART OF. *Модель времени* строится на основании конечного множества типов временных интервалов, на которых определено отношение PART OF.

 τi PART OF τj

 *явл. частью*

*Каждый интервал типа τi является частью одназначно определенного интервала τj.*

 Время

 sup sup

год неделя

 PART OF PART OF

месяц

 PART OF

сутки

 PART OF

час

 PART OF

минута

PART OF

секунда

рис. 7

Модель пространства строится на конкретном множестве, так же, как и времени, но по отношению к части окрестности.

(См. рис.8)

 пространство

 sup sup

страна континент

 PART OF PART OF

область регион

 PART OF PART OF

населенный

 пункт

 PART OF PART OF

город деревня

 рис.8

3. Отношение принадлежности ISA

 Иванов А.И. ISA студент

 *связывает имя объекта с его типом*

4. Отношение INSTANCE OF (INS)

 Иванов INS АИ-1-93

 *отношение может быть задано как на*

 *множестве имен, так и типов*

 name (O1) INS name (O2)

 Если τ1 INS τ,

 τ2  INS τ,

 ..........

 τn INS τ, то объект типа τ может состоять из объектов

 типов τ1,τ2...τn, причем объект типа τ может включать

 несколько объектов данных типов.

Это отношение (INS) применяется для множества дискретных объектов,

PART OF - непрерывных.

*студент* INS *группа студентов*

 лаборатория

 INS INS

 INS

 сотрудники помещение оборудование

лаборатории лаборатории лаборатории

 рис.9

*Свойства этих отношений*

1. PART OF и SUP обладают свойством транзитивности:

 τ1 SUP τ2 ; τ2 SUP τ3

 τ1 SUP τ3

 доцент SUP SUP сотрудник

 преподаватель

 рис.10

 доцент SUP преподаватель ; преподаватель SUP сотрудник

 доцент SUP сотрудник

Для отношения INS это свойство не выполняется:

2. τ1 INS τ2 ; τ2 INS τ3 ; τn-1 INS τn  *В графе отношения INS нет*

 *неверно, что τn INS τ1 циклов*

3. τ1 SUP τ2 ; τ2 INS τ3

 τ1 INS τ3

лаборант SUP сотрудник лаборатории ; сотрудник лаборатории INS лаборатория

 лаборант INS лаборатория

 лаборатория

 INS

 сотрудник INS

 SUP

 лаборант

 рис.11

4. name (O) ISA τ1 ; τ1 SUP τ2

 name (O) ISA τ2

Москва ISA город ; город SUP населенный пункт

 Москва ISA населенный пункт

*Представление типов объектов*

Конечное множество троек вида (τ, ni, τi);

 где τ, τi - типы объектов,

 ni - имя атрибута,

 (*Атрибуты* - характеристики объектов или роли, которые они играют в определенных ситуациях) i=1,..., к,

 ni≠nj, если i≠j, называется представлением типа τ, если в каждый момент времени ni: τобt → (τi)обt,

(ni - отображ. мн-во объектов типа τ на мн-во объектов типа τ1)

причем разным объектам О1 и О2 типа τ соответствуют разные картежи значений функций ni

 {n1t(O1), n2t (O1,..., nkt(O1)}

 {n1t(O2), n2t(O2,...,nkt(O2)},

 последнее условие называется *условием различимости* объектов.

В качестве примера рассмотрим объект типа автомобиль:

 τ ni τi

 (автомобиль, марка, марка автомобиля)

 (автомобиль, номер, номер автомобиля)

 (автомобиль, цвет, цвет автомобиля)

 автомобиль(москвич, МКА-21-17, белый)

 автомобиль(жигули, МВА-11-25, красный)

*Композиционный тип объектов*

(ni , τi) COMPONENT OF τ

COMPONENT OF означает, что объект типа τi является компонентой объекта типа τ.

 дата

 COMPONENT OF

 COMPONENT OF COMPONENT OF

год n1 число n3 месяц n2

год τ1  число τ3  месяц τ2

 рис.12

Пример композиционного объекта:

Ситуация "находится" (нах. человек в опред. помещении опред. время)

|  |
| --- |
| τ находится |
| n1 участник ситуации | n2 место ситуации | n3 время ситуации |
| τ1  человек | τ2  помещение | τ3 время |
| sup sup | sup sup |  дата ситуац. | момент сит. |
| учащиеся | сотрудники | аудитория | лаборатор. |  дата |  момент |
|  |  ins ins | числосит. | месяцсит. | годсит. | час ситуации | минутаситуации |
|  |  оборудование | число | месяц | год | час | минута |

*Шаблонные высказывания*

человек . . . находится в помещении . . . во время . . .

сотрудник . . . находится в аудитории . . . в . . . часов . . . минут . . .

. . . число . . . месяц . . . год

В шаблонные высказывания могут быть поставлены имена конкретных объектов.

*Минимальное представление*

Представление {(τ, ni, τi)| i=1...k} называют минимальным, если при удалении ∀ из этих троек, оставшееся множество троек перестает быть представлением данного типа объектов.

Пример:

τ - аренда

τ1 - объект аренды

τ2 - арендатор

τ3 - арендодатель

τ4 - срок аренды

τ5 - стоимость аренды

*Функциональная зависимость*

Предположим, задано представление

{(τ, ni, τi)| i=1...k}

 X={τi1, ..., τip}

 Y={τj1, ..., τjs}

Говорят, что компоненты Y функционально зависят от компонент X, если в любой момент времени t для любых двух объектов О1, О2 ∈ τобt из равенств: ni1t(O1) = ni1t(O2)

 . . .

 nipt(O1) = nipt(O2),

 следуют равенства:

 nj1t(O1) = nj1t(O2)

 . . .

 njst(O1) = njst(O2)

Пример: X={τ4, τ5, τ6} Y={τ1, τ2,τ3}

|  |
| --- |
| τ Лекция |
| лектор | слушатель | предмет лекции | место | день недели | время начала |
| преподаватель | группа | предмет | аудитория | дата |   |

τ1 τ2 τ3 τ4 τ5 τ6

 min представление

*Операции над типами объектов*

Позволяют получать из существующих типов новые типы.

1. Могут быть заданы операции объединения (τ1∪τ2), пересечения

 (τ1∩τ2), разбиения и др.

Если { τi INS τ | i=1,...,k}, то тип ∪ τi состоит из тех и только объектов, которые могут быть элементами объектов типа τ.

 Если {(τi SUP τ)| i=1,...,k}, то равенство τ=∪τi означает, что список подтипов τi является исчерпывающим для объектов типа τ.

 Примера :

 объект = человек ∪ помещение ∪ оборудование

 человек∩помещение = человек∩оборудование=

 = помещение∩оборудование = 0

2. Операция разбиения типов объектов на классы эквивалентных объектов.

(человек, пол человека, пол)

 τ n1 τ1

пол = {муж, жен}

 τ2 = муж

 τ3 = жен

Типы τ2 и τ3 являются результатом разбиения типа человек по типу пол (τ/τ1)

|  |
| --- |
| τ Знание иностранного языка |
| n1 знающий | n2 предмет знания |
| τ1 человек | τ2 иностранный язык |

Новые объекты - группы людей, знающих некоторый язык определяются в результате разбиения τ1/(τ2,τ)

*Представление знаний об отношениях между типами*

ER - модели

ER - диаграммы

(Entity Relationship Diagram)

П. Чен - 1983 год.

1. сущности (типы объектов)

2. связи (типы отношений)

3. свойства сущностей и связей

 (атрибут (характеристики, роли))

1) Е1, Е2, ..., Еn

 преподаватель

2) R1, R2, ...,Rn

 занятие

кафедра работает преподаватель нагрузка

группа занятие предмет

обучение факультет изучение

курс специальность

 рис.13

3. Свойства (атрибуты)

 Ei, Ri - соответствующие области знаний

 имя - имя области знаний

На дуге подписываются соответствующие свойства.

 имя

 *название*

 кафедра

имя имя имя

 фамилия имя отчество

 преподаватель

 рис.14

 код

имя группа

 группы

 рис.15

 название имя

предмет

 объем

 натуральное число

 рис. 16

 натуральное

 лекция число

нагрузка

лабораторная натуральное

 число

 рис. 17

 шифр специальность

код

 рис.18

Часто в виде связи выступает прямое дополнение.

В виде свойств обычно представляются время и место ситуации.

 занятие

 место начало

помещение день недели время

 рис.19

Часто свойства вводятся с помощью слова "иметь"

 код

группа код

 группы

 рис.20

 преподаватель

обучение учебн. студент

 процесс

 рис. 21

 учебный

 процесс

 контроль

 административный

 персонал

 рис. 22

"Учебный процесс" - является и сущностью и связью.

 преподаватель

курс обучение учебный студент

 процесс

факультет специальность контроль

 административный

 персонал

 рис. 23

Часть предложения, которая являются придаточным, рассматривается как сущность более высокого уровня абстракции.

 распределение

заведующ. решает преподаватель

 кафедрой

 группа занятие

 предмет

 рис. 24

Концептуальная схема позволяет представить различные иерархические связи между объектами. Классический пример - классификация животных.

(См. ниже рис. 25)

 животные

простейшие . . . губки . . . хордовые

. . . . . .

 оболочники . . . бесчерепные . . . позвоночные

 . . . . . . . . . . . .

 рис. 25. *Иерархическая таксономическая структура*

*Таксон* - группа дискретных объектов, связанных некоторой степенью общности свойств. Набор свойств должен быть необходимым и достаточным для того, чтобы таксон занимал единственное место в структуре и не пересекался с другими таксонами.

При декларации таксона указывается таксон верхнего уровня и свойства, которые отличают данный таксон от других таксонов.

*Таксономические категории* -

 {тип - подтип - класс - отряд - род - вид}

 Обычно с типом объекта нижнего уровня связывают конкретные факты.

 кошки род

 sup sup

 домашняя лесная вид

 isa

барсик

 рис. 26

Свойства, характерные для некоторого типа объекта, наследуются всеми типами объекта нижнего уровня. В БД могут быть описаны только один раз.

2. Практический раздел

Как уже отмечалось ранее, целью данного КПР является подготовка теоретического и практического материала для обеспечения доступного изучения соответствующих разделов по дисциплине Инженерия Знаний. Подготовленный теоретический материал можно найти в "Теоретическом разделе" данного КПР. Он содержит в себе все понятия и определения, а также примеры, необходимые для успешного использования (применения) этих знаний и изучения соответствующих тем.

Для наиболее лучшего освоения и закрепления полученных знаний мы предлагаем проводить практические занятия по данному разделу. В качестве материалов для проведения практических занятий предлагается использовать разработки данного КПР, приведенные ниже.

2.1 Ограничения на использование разработанного программного обеспечения

Для того, чтобы в дальнейшем не возникало непредвиденных проблем с применением разработанного программного обеспечения, ниже приводятся необходимые аппаратные требования для успешного функционирования программ:

 ***HARD*** *(минимальные требования):*

ю : персональный компьютер фирмы IBM или совместимый с ним;

ю : 486 SX-33Mhz, RAM 4Mb, VideoRAM 512Mb, 45 Mb свобод.

 места на HDD, VGA, манипулятор типа "мышь";

 ***SOFT****:*

ю = операционная система MS-DOS 5.0 и выше,

 либо я MS Windows - 95;

ю = графическая оболочка я MS Windows 3.xx

 (данный пункт опускается, если используется я MS Windows -95)

ю = полностью установленный на жесткий диск MS ACCESS2.0,

 если используется я MS Windows 3.xx,

 либо MS ACCESS7.0 for я MS Windows -95;

2.2 Необходимые знания для работы в MS ACCESS2.0

Краткий экскурс в Access

MS ACCESS2.0 - это очередная разработка фирмы Microsoft®. По сути дела - это СУБД с очень дружественным и наглядным интерфейсом, как собственно и все программы, разработанные фирмой Microsoft® для ее же графической оболочки я MS Windows 3.xx или для операционной системы я MS Windows -95.

В данном КПР не приводится описание работы я MS Windows, так как считается, что пользователь уже знаком с ней. В противном случае необходимо обратиться к специальной литературе.

Если Access инсталлирован на компьютере, его запуск происходит следующим образом:

1. Запустить я MS Windows;

2. Открыть окно Access-группы Диспетчера Программ (Program Manager) и дважды щелкнуть мышью на пиктограмме, соответствующей Access (на ней изображен ключ и имеется надпись "Microsoft Access";

*Чтение экрана Access*

После загрузки Access в верхней строке главного окна (строка заголовка) появляется имя программы - Access. Ниже расположена строка меню. Под ней расположена строка инструментов. Различные кнопки этой панели появляются в зависимости от того, в какой части программы вы находитесь. В нижней части экрана расположена строка состояния, которая содержит полезную информацию о ходе работы с Access.

*Открытие меню*

Почти всякая работа в Access начинается с открытия БД с помощью меню File (Файл). Для этого щелкните на элементе File (Файл) в строке меню, и на экране раскроется меню, с помощью которого Access позволяет создавать новые или открывать существующие БД, а также выполнять другие операции.

*Получение помощи (Меню подсказки в Access)*

Справочная система Access схожа с другими аналогичными системами, которые имеются в приложениях Windows.

Для этого щелкните на элементе Help (Справка) строки меню и выберите соответствующий раздел данного меню.

Другое важное средство, имеющееся в окне справочной системы - Cue Cards (Карточки подсказки) - специальный вид экранной подсказки, который проведет вас шаг за шагом при выполнении сложных процедур в Access.

 Wizards (Мастера) также проведут вас шаг за шагом, но, в отличие от Cue Cards сделают за вас большую часть работы. В окне справочной системы нет никакой информации о Wizards, но как только вы начнете что-нибудь создавать, Access сразу же оживет и спросит, хотите ли вы, чтобы мастер сделал это за вас.

Выход из справочной системы - ALT+F4 (или дважды щелкнув на кнопке системного меню в его левом верхнем углу).

*Выход из Access*

Выйти из Access можно с помощью одного из следующих способов:

- При открытом окне Access одновременно нажать ALT+F4.

- Дважды щелкнуть на кнопке системного меню в верхнем левом углу окна Access.

- Щелкнуть на кнопке системного меню окна и затем на директиве Close (Закрыть).

*Таблицы для хранения ваших данных*

Таблица - основной структурный объект во внутреннем строении БД. Она содержит все записи определенного вида. Каждая запись таблицы содержит всю необходимую информацию о каком-то отдельном элементе БД.

Прежде, чем спроектировать БД, необходимо определить, какие таблицы нам потребуются (см. ниже). Нерационально вводить все данные в одну таблицу, поэтому Access предлагает вводить различные виды данных в разные таблицы. При необходимости нетрудно объединить данные из различных таблиц.

*Строки и столбцы*

Объект, выведенный в табличном виде, называется в Access табличным (datasheet).

Названия столбцов (колонок) появляются в верхней части главного окна Access. Колонки называются полями. Каждое поле содержит определенную информацию. Строки называются записями. Каждая запись имеет одинаковые поля, которые имеют одинаковый размер в каждой записи.

*Запросы*

После ввода данных в БД вам может понадобиться найти какую-то их часть. Это можно сделать несколькими способами, но лучшим способом является использование простого запроса для поиска необходимой записи. Запрос - это вопрос к БД.

После формулировки запроса сообщите Access, какое поле вы хотели бы просмотреть и что хотели бы в нем найти. Затем просто щелкните на соответствующей кнопке, и Access тут же выведет на экран необходимую запись.

*Формы*

Другим видом представления таблицы является так называемая форма. Формы - еще одно важное средство Access. При проектировании формы учитывают, что ею будет пользоваться человек, который ничего не знает ни о данных, ни о БД вообще. Используя Access, можно включить в форму любые текстовые комментарии, которые объясняют, какие данные и в какие поля вводить.

*Отчеты*

С помощью Access можно формировать отчеты для распечатки ваших данных в различных видах.

2.3 Планирование БД

После того, как стало известно, какие средства может предложить Access, необходимо спроектировать БД в соответствии с тематикой данного КПР.

Таким образом, нам необходимы следующие базы данных:

1. Классы. Поля данной БД: *Код класса*,

 *Название класса*,

 *Описание класса*.

2. Атрибуты. Поля данной БД: *Код атрибута*,

 *Название атрибута*,

 *Описание атрибута*.

3. Атрибуты класса. Поля данной БД: *Код класса*,

 *Код атрибута*,

 *Значение атрибута.*

4. Род-вид. Поля данной БД: *Код класса,*

 *Код подкласса.*

5. Структура. Поля данной БД: *Код класса,*

 *Код компоненты*.

6. Связи классов. Поля данной БД: *Код связи,*

 *Код класса,*

 *Роль класса (причина, следствие).*

7. Типы связей. Поля данной БД: *Код типа связей,*

 *Название типа связей,*

 *Описание типа связей.*

8. Коды связей. Поля данной БД: *Код типа связи,*

 *Код связи.*

Предложенные базы данных после их создания необходимо связать, тем самым обеспечив иерархическую структуру. Ниже будет описано, как это сделать.

*Создание БД*

После загрузки Access активизируйте меню ФАЙЛ в левой верхней части Access и в появившемся меню щелчком выберите директиву New Database (Создать). Для создания первой БД в поле FileName введите имя файла, в котором будет храниться первая БД, например, "Klasses" и щелкните на командной кнопке ОК в правой верхней части появившегося окна. Access создаст для вас новую БД. Обратите внимание, что в левой части окна БД Klasses имеется ряд так называемых корешков, каждому из которых соответствует определенный объект Access: таблицы (Table), запросы (Query), формы (Forms), отчеты (Report), а также макрос (Macro), модуль (Module). Как использовать их, описано ниже. Необходимо создать БД для каждого из 8 пунктов, описанных выше, в разделе Планирование БД. То есть указанные действия выполняются восемь раз для наших БД. Предлагаем использовать следующие имена для этих баз:

1. Klasses (ее создание описано в данном разделе)

2. Atribute - атрибуты

3. Atr\_klass - атрибуты класса

4. Rod\_vid - род-вид

5. Structur - структура

6. Sv\_klass - связи классов

7. Type\_sv - типы связей

8. Code\_sv - коды связей

Необходимость описания БД именно этими именами связана с тем, что в данной пояснительной записке используются именно эти имена при описании методики проведения практической работы.

2.4 Создание таблиц для ваших данных

*Планирование структуры таблицы*

Речь идет о перечне полей, из которых она состоит. При создании таблицы на первом этапе необходимо определить, какие поля вам нужны, какого они должны быть типа и размера.

Поле - структурный элемент таблицы, содержащий сведения о ком-то или о чем-то в БД.

Для БД Klasses необходимы поля, указанные выше, в разделе "Планирование БД". Рассмотрим пример создания структуры таблицы для БД Klasses.

Необходимые поля для данной таблицы, это:

 *Код класса*,

 *Название класса*,

 *Описание класса*.

*Создание таблицы*

После определения необходимых полей, приступим к созданию собственно таблицы в Access.

1. Определить, какие поля необходимы (см. выше)

2. Открыть окно конструктора таблиц. Каждое поле таблицы - это одна строка верхней части этого окна. В упрощенном варианте достаточно использовать только верхнюю часть окна.

3. В верхнюю часть окна конструктора таблицы ввести имя каждого поля, при необходимости можно использовать и нижнюю часть окна для задания длины поля, его формата и т.д.

4. При желании можно создать поле, которое будет выступать в качестве уникального идентификатора каждой записи. Это так называемый ключ таблицы, а соответствующее ему поле называется ключевым.

5. Сохранить созданную таблицу и выйти из окна конструктора таблиц.

*Загрузка БД*

Чтобы загрузить БД, созданные в разделе "Создание БД", необходимо:

1. Загрузить Access.

2. В Access активизировать меню ФАЙЛ. Перед вами откроется список имен файлов. Выберите нужный и щелкните на нем (klasses.mbd).

*Функции кнопок*

Окно БД имеет две группы кнопок.

- Слева в окне расположены кнопки режима (корешки) с именами различных структурных объектов БД - таблицы (Table), запросы (Query), формы (Forms), отчеты (Report). Каждый раз щелчком можно выбрать только один корешок из этого ряда.

- В верхней части окна расположены командные кнопки New (Создать), Open (Открыть), Design (Конструктор). Щелкнув на кнопке New, вы сообщаете Access, сто хотите создать новую таблицу. (Кнопка Table должна быть при этом нажата, о чем сигнализирует более яркая надпись).

Щелкнув на кнопке New, - на экране появится диалоговое окно. Оно содержит кнопки для выполнения различных функций. В рассматриваемом примере следует щелкнуть на кнопке New Table (Новая таблица). Щелкнув на этой кнопке, активизируется режим TABLE-DESIGN (Режим конструктора таблиц).

*Формирование полей таблицы*

Каждому полю таблицы следует присвоить уникальное имя, определить тип данных, которые будут в него вводиться, и описать содержимое поля, т.е. ввести текстовые комментарии, в которых указано назначение данного поля. В отличие от имени БД, которое не может содержать более 8 символов, имя поле может иметь до 64 символов.

Окно конструктора таблиц делится на две основные части. В верхней части вы можете указать имя. тип и описание каждого поля. В нижней части окна указываются различные параметры каждого поля. Для того, чтобы определить первое поле, введите с клавиатуры в первом столбце первой строки слово "Код класса" и нажмите Tab. Access создаст поле "Код класса". После нажатия клавиши табуляции "оживает" второй столбец Data Type (Тип данных) первой строки. В ней появляется слово Text (Текстовый), справа от которого будет находится маленькая кнопка - манипулятор. Созданное поле должно быть счетчиком порядковых номеров записей. В окне-списке типов данных щелкните на слове "Counter" (Счетчик).

После этого нажмите Tab еще раз. Курсор переместится в столбец описания поля. Наберите на клавиатуре описание поля: Код класса.

*Выбор ключа таблиц*

Последнее, что осталось сделать с полем "Код класса", - это дать ему статус ключа таблицы. Поле с таким статусом помогает Access упорядочивать данные в таблице, быстрее их находить и связывать таблицу с другими таблицами.

1. Убедитесь, что курсор мыши находится в одной строке с названием "Код класса".

2. Щелкните на маленькой кнопке, обозначенной ключом и расположенной в центре экрана на панели инструментов конструктора таблиц.

*Выбор правильного типа данных*

В Access можно выбрать один из следующих типов данных:

TEXT (Текстовый). Используется для текстовой информации.

MEMO (Комментарии). Используется для хранения произвольного текста. До 32 000 символов.

NUMBER (Числовой). Используется для математических операций.

CURRENCY (Валютный). Для операций с валютой.

DATE/TIME (Дата/Время). Информация для хранения даты и времени.

COUNTER (Счетчик). Порядковый номер.

YES/NO. (Логическое).

OLE object (Поле объекта OLE). рисунки, графики и таблицы в формате Windows.

*Оформление оставшихся полей*

Снова нажмите клавишу Tab. Курсор переместится в первый столбец следующей строки. Access готов к вводу информации о следующем поле. Специфицируйте остальные поля вашей БД следующим образом:

|  |  |  |  |
| --- | --- | --- | --- |
| **Имя поля** | **Тип данных**  | **Размер** | **Описание** |
| Название класса | текстовый | 15 | название класса |
| Описание класса | текстовый  | 15 | описание класса |

*Сохранение описания таблицы*

После того, как вы оформили поля, осталось сделать только одну, но очень важную процедуру - сохранить их.

Для этого щелкните на элементе File в строке меню и выберите директиву Save. Появится диалоговое окно Save As (Сохранение).

Access предложит свой вариант - Table1 (Таблица1). Мы советуем для удобства дальнейшего использования изменить имя на "Классы".

Необходимо отметить, что создать таблицу можно еще одним способом, - используя мастер по разработке таблиц. Здесь не приводится описание работы этого мастера, так как при его использовании не должно возникать трудностей, потому что он работает в пошагово-диалоговом режиме.

Таким образом, мы создали первую таблицу. Но необходимо то же самое проделать с остальными. Ниже приводим рекомендации по спецификации остальных таблиц для оставшихся БД.

- БД Atribute

|  |  |  |  |
| --- | --- | --- | --- |
| **Имя поля** | **Тип данных** | **Размер** | **Описание** |
| Код атрибута | Counter | по умолчанию | Код атрибута |
| Название атрибута | текстовый | 15 | Название атрибута |
| Описание атрибута | текстовый | 15 | Описание атрибута |

- БД Atr\_klass

|  |  |  |  |
| --- | --- | --- | --- |
| **Имя поля** | **Тип данных** | **Размер** | **Описание** |
| Код класса | Number | по умолчанию | Код класса |
| Код атрибута | Number | по умолчанию | Код атрибута |
| Значение атрибута | текстовый | 15 | Значение атрибута |

- БД Rod\_vid

|  |  |  |  |
| --- | --- | --- | --- |
| **Имя поля** | **Тип данных** | **Размер** | **Описание** |
| Код класса | Number | по умолчанию | Код класса |
| Код подкласса | Number | по умолчанию | Код подкласса |

- БД Structur

|  |  |  |  |
| --- | --- | --- | --- |
| **Имя поля** | **Тип данных** | **Размер** | **Описание** |
| Код класса | Number | по умолчанию | Код класса |
| Код компоненты | Number | по умолчанию | Код компоненты |

- БД Sv\_klass

|  |  |  |  |
| --- | --- | --- | --- |
| **Имя поля** | **Тип данных** | **Размер** | **Описание** |
| Код связи | Counter | по умолчанию | Код атрибута |
| Код класса | Number | по умолчанию | Код класса |
| Роль класса | текстовый | 20 | Роль класса |

- БД Type\_sv

|  |  |  |  |
| --- | --- | --- | --- |
| **Имя поля** | **Тип данных** | **Размер** | **Описание** |
| Код типа связей | Counter | по умолчанию | Код типа связей |
| Название типа связей | текстовый | 15 | Название типа связей |
| Описание типа связей | текстовый | 20 | Описание типа связей |

- БД Code\_sv

|  |  |  |  |
| --- | --- | --- | --- |
| **Имя поля** | **Тип данных** | **Размер** | **Описание** |
| Код типа связи | Number | по умолчанию | Код типа связи |
| Код связи | Number | по умолчанию | Код связи |

2.5 Ввод данных в ваши таблицы

*Открытие окна таблицы*

Загрузите Access и откройте файл "Klasses.mdb". В окне БД должна быть представлена таблица "Классы". (Если ее нет, то щелкните на корешке Table). Дважды щелкните на названии таблицы и окно откроется.

Вдоль верхнего края расположены имена полей таблицы. Каждое поле соответствует определенному столбцу в таблице. В левой части первой строки вы видите маленький треугольник, указывающий на то, что Access готов принимать любую информацию в первую строку.

Теперь можно заполнить БД в соответствии с заданием преподавателя (используя свою предметную область).

Нет необходимости вводить с клавиатуры данные в поле типа "Counter", так как оно заполняется Access автоматически.

Перемещаться по таблице можно используя клавиши управления курсором либо с помощью мыши.

После ввода каждой строки Access автоматически сохраняет запись.

Access позволяет редактировать введенные данные.

2.6 Связи между таблицами

В Access можно установить два вида связи между таблицами: одна-с-одной и одна-со-многими.

Для объединения таблиц необходимо общее поле. Это поле обязательно должно быть одного типа. Исключение составляет поле счетчик. Если в связанной таблице общее поле будет такого же типа, то связь один-со-многими будет невозможна. Поэтому, если общее поле является счетчиком в таблице, то в связанной таблице оно должно быть числовым, а его размер должен быть установлен как Длинное целое.

*Установление связей*

Откройте меню Правка (Edit) и выберите в нем директиву Relationships (Схема данных).

Рассмотрим пример для связи двух таблиц - "атрибуты" и "атрибуты класса":

1. В списке полей таблицы "атрибуты" щелкните на поле "Код атрибута" для того, чтобы выделить его.

2. Установив курсор мыши на этом поле, нажмите ее левую кнопку.

3. Переместите это поле с помощью мыши на поле "Код атрибута" в таблице "атрибуты класса".

4. Отпустите левую кнопку мыши.

 При этом появится диалоговое окно установления связей, если в нем все выглядит как надо, щелкните на "Создать".

5. Подобные связи необходимо создать для всех таблиц в соответствии со схемой, представленной на рис. 27.

6. Закройте окно Relationships. Когда Access спросит, желаете ли вы сохранить изменения в схеме данных, ответьте "Да".

*Для удаления ошибочной связи* выделите черную линию, соединяющую два поля и нажмите Del (в окне Relationships).

Итак, мы создали связи таблиц - т.е. иерархию. Под эту иерархию можно подвести множество предметных областей. Но необходимо соблюдать точность и внимательность при заполнении самих таблиц реальными данными. Вводить данные во все созданные БД необходимо с учетом этих связей.

Описав ПО с помощью данной структуры, пользователь может производить различные действия. Суть манипуляций заключается в изучении иерархии, наследования признаков и т.д.

Access позволяет быстро находить нужные данные с использованием директивы Find (Найти) или Query (Запросов), создавать формы для вывода нужных данных, удобные для восприятия либо в соответствии с какими-либо требованиями (например - ГОСТ).

Все эти возможности можно реализовывать при проведении практических занятий. Для подробного ознакомления с такими возможностями, как Запросы, Формы пользователю будет необходима либо специальная литература по Access, либо полностью установленная версия Access на жесткий диск компьютера, о чем уже упоминалось ранее, чтобы была возможность использовать справочную систему Access.

Упомянутые возможности Access не входят в темы КПР, поэтому мы допускаем ссылки на специальную литературу.

3. Варианты заданий по л.р.

1. Представление знаний об объектах предметной области:

 а) Выделить основные типы объектов

 б) Выделить основные атрибуты

 в) Задать представление знаний о конкретных объектах в виде БД

2. Представление знаний об отношениях между объектами:

 а) Построить иерархическую структуру на основе отношения SUP

 б) С типами объектов нижнего уровня связать имена конкретных

 объектов

3. Установление связей между типами объектов:

 а) Выделить основные типы связей

 б) Выделить коды связей

 в) Установить связи между типами

 Перечень предметных областей:

1. Швейная промышленность

2. Текстильная промышленность

3. Обувная промышленность

4. Автомобильная промышленность

5. Ликеро-водочная промышленность

6. Кораблестроительная промышленность

7. Парфюмерная промышленность

8. Хлебо-пекарное производство

9. Кондитерское производство

10. Производство телевизоров

11. Станкостроительная промышленность

12. Военная промышленность

13. Производство канцтоваров

14. Производство меховых изделий

15. Производство колбасных изделий

16. Производство молочных изделий

4. Содержание отчета

Отчет о л.р. должен содержать:

 1. Тему л.р.

 2. Цель л.р.

 3. Теоретическое вступление (кратко)

 4. Анализ предметной области - концептуальная схема

 5. Вывод о проделанной работе

 6. Заключение

 5. Контрольные вопросы

1. Предмет Инженерии Знаний

2. Понятие предметной области (ПО)

3. Способы идентификации объектов

4. Концептуальные средства описания ПО

5. Базовые типы объектов

6. Представление знаний о пространственных отношениях между объектами

7. Отношения между типами объектов

8. Композиционный тип объектов

9. Операции над типами объектов

10. Операция разбиения типов объектов на классы эквивалентных объектов

11. Понятия "предмет", "объект" в рамках ИЗ

12. Шаблонные описания состояний предметной области

13. Представление знаний между типами

14. Таксон и таксономические категории

6. Выполнение л.р. для одного из вариантов заданий

ПО - автомобили

*тема:* Представление знаний об объектах и отношениях между ними.

*цель:* Описать предметную область на основе знаний о представлении знаний об отношении меду объектами и о самих объектах. Использовать электронную модель в ACCESS2.0.

*теоретическое вступление* (См. Теоретический раздел данного КПР)

*анализ ПО: концептуальная схема (сокращена):*

 автомобиль

 SUP SUP SUP

автобус легковой\_автомобиль грузовой\_автомобиль

 SUP SUP SUP SUP SUP

микроавтобус рейсовый междугородный "седан" "пикап"

 ISA ISA ISA ISA

"раф" "тойота" "львов" "лиаз"

Для удобства восприятия ниже приводим предсталение ПО на языке предикатов в PROLOGе:

sup("автобус","автомобиль")

 sup("легковой\_автомобиль","автомобиль")

 sup("грузовой\_автомобиль","автомобиль")

 sup("микроавтобус","автобус")

 sup("рейсовый\_автобус","автобус")

 sup("междугородный\_автобус","автобус")

 sup("седан\_автомобиль","легковой\_автомобиль")

 sup("пикап\_автомобиль","легковой\_автомобиль")

 sup("фургон","грузовой\_автомобиль")

 sup("бортовой\_автомобиль","грузовой\_автомобиль")

 sup("самосвал","грузовой\_автомобиль")

 isa("раф","микроавтобус")

 isa("тойота","микроавтобус")

 isa("львов","рейсовый\_автобус")

 isa("лиаз","рейсовый\_автобус")

 isa("икарус","междугородный\_автобус")

 isa("мерседес","междугородный\_автобус")

 isa("ваз","седан\_автомобиль")

 isa("газ\_волга","седан\_автомобиль")

 isa("москвич\_иж","пикап\_автомобиль")

 isa("москвич\_азлк","пикап\_автомобиль")

 isa("газ","фургон")

 isa("сканиа","фургон")

 isa("зил","бортовой\_автомобиль")

 isa("каз","бортовой\_автомобиль")

 isa("камаз","самосвал")

 isa("татра","самосвал")

 isa("белаз","самосвал")

 isa("маз","самосвал")

*Пример заполнения БД в ACCESS:*

1. Классы

|  |  |  |
| --- | --- | --- |
| **Код класса** | **Название класса** | **Описание класса** |
| 1 | автомобиль | \*Комментарии\* |
| 2 | автобус | \*Комментарии\* |
| 3 | легковой автомобиль | \*Комментарии\* |
| 4 | седан-автомобиль | \*Комментарии\* |
| 5 | газ-волга | \*Комментарии\* |

2. Атрибуты

|  |  |  |
| --- | --- | --- |
| **Код атрибута** | **название атрибута** | **Описание атрибута** |
| 1 | марка | \*Комментарии\* |
| 2 | цвет | \*Комментарии\* |
| 3 | номер кузова | \*Комментарии\* |

3. Атрибуты класса

|  |  |  |
| --- | --- | --- |
| **Код класса** | **Код атрибута** | **Значение атрибута** |
| 1 | 1 | москвич |
| 1 | 2 | синий |
| 1 | 3 |  |

4. Род-Вид (SUP)

|  |  |
| --- | --- |
| **Код класса** | **Код подкласса** |
| 1 | 2 |
| 1 | 3 |

5. Структура (ISA)

|  |  |
| --- | --- |
| **Код класса** | **Код компонеты** |
| 4 | 5 |

6. Связи классов

|  |  |  |
| --- | --- | --- |
| **код связи** | **код класса** | **роль класса** |
| 1 | 2 | \*Комментарии\* |
| 2 | 3 | \*Комментарии\* |
| 3 | 4 | \*Комментарии\* |
| 4 | 5 | \*Комментарии\* |

7. Типы связей

|  |  |  |
| --- | --- | --- |
| **Код типа связи** | **Название типа связи** | **описание типа связи** |
| 1 | SUP | \*Комментарии\* |
| 2 | ISA | \*Комментарии\* |

8. Коды связей

|  |  |
| --- | --- |
| **Код типа связи** | **Код связи** |
| 1 | 1 |
| 1 | 2 |
| 1 | 3 |
| 2 | 4 |

Заключение

Результатом данного КПР по теме "представление и использование знаний об объектах; представление и использование знаний об отношениях между объектами; организация обобщенной информации на основе использования мультииерархической таксономической структуры" является подборка теоретического и практического материала для проведения практических работ по соответствующим разделам дисциплины Инженерия Знаний. В результате группе разработчиков удалось создать электронную модель, по сути являющейся универсальной структурой, пригодной для описания практически любой предметной области, что дает возможность использовать ее большим количеством пользователей. Данная модель создана в СУБД MS ACCESS 2.0.

 В работе также приводится методика создания структуры в MS ACCESS, что в свою очередь может являться заданием на практическую работу.

 Текст пояснительной записки набран в среде MS WORD 6.0 for Windows. Отпечатано на принтере Epson LX-100.

 Все упомянутые в тексте зарегистрированные товарные знаки и торговые марки являются собственностью компаний, которым они принадлежат.

Библиографический список

1. Цаленко М.Ш. Моделирование семантики в базах данных., Москва, Наука, 1989г., - 286 с.

2. Человеко - машинные системы и анализ данных, РАН, Институт проблем передачи информации, Сборник научных трудов, Москва, Наука, 1992г., - 174с.

3. Палмер Скотт, ACCESS 2 for DUMMIES, Киев, 1996г., - 249 с.

4. Конспект лекций по дисциплине Инженерия знаний за V семестр.