На правах рукописи

БРЕШЕНКОВ АЛЕКСАНДР ВЛАДИМИРОВИЧ

МЕТОДОЛОГИЯ ПРОЕКТИРОВАНИЯ РЕЛЯЦИОННЫХ БАЗ ДАННЫХ С ИСПОЛЬЗОВАНИЕМ ДАННЫХ ТАБЛИЧНОГО ВИДА

Специальность 05 25 05 – Информационные системы и процессы, правовые аспекты информатики

АВТОРЕФЕРАТ

диссертации на соискание ученой степени
доктора технических наук

003065382

Москва – 2007

Работа выполнена в Московском государственном техническом университете им Н Э Баумана

Официальные оппоненты доктор технических наук, профессор Селетков Сергей Николаевич,

доктор технических наук, ст н с Романов Виктор Петрович,

доктор технических наук, профессор Горячев Александр Васильевич

Ведущая организация Московский автомобильно-дорожный институт (государственный технический университет)

Защита состоится 10 октября 2007 года на заседании диссертационного совета Д 502 006 17 в Российской академии государственной службы при Президенте Российской Федерации по адресу 119606, Москва, проспект Вернадского, д 84 в 16^{00}

С диссертацией можно ознакомиться в библиотеке Российской академии государственной службы при Президенте Российской Федерации (119606, Москва, пр-т Вернадского, д 84)

Автореферат разослан « _____» _____2007 г

Ученый секретарь
диссертационного совета,
доктор педагогических наук,
кандидат физ -мат наук,
доцент Митин А И

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Актуальность проблемы. Сегодня трудно переоценить значение информации и информационных систем Особое место в составе информационных систем принадлежит базам данных (БД) В настоящее время во многих областях человеческой деятельности используются БД При этом потребность в них и в системах управления базами данных (СУБД) постоянно растет

К числу наиболее распространенных моделей построения БД относятся реляционные модели данных (РМД) Достоинства реляционной модели обусловили проведение теоретических и практических разработок в области теории проектирования реляционных БД (РБД), в области разработки инструментальных средств, ориентированных на их создание Среди них можно назвать работы Е Ф Кодда, К Дж Дейта, Мейера Д , Г Хансена, Дж Хансена, Ульмана Дж , Чена Р Р , Райана Стивенса, Рональда Плю, Дэйва Энсора, Харитоновой И А., Михеевой В Д , Тихомирова Ю В , Григорьева Ю А , Ревункова Г И , Дрибаса В П , Карповой Т С и других

РБД проектируются в соответствии с предложенными теоретиками реляционного подхода этапами проектирования, модели РБД строятся в соответствии с требованиями к реляционным моделям данных, реляционные таблицы проектируются с учетом требований нормализации Существующая методология проектирования РБД позволяют разработчику обоснованно назначить ключевые и индексные поля, сформировать связи между таблицами, обеспечить безопасность данных и выполнить много полезных мероприятий, обеспечивающих разработку высококачественных программных систем

Однако, даже основоположники РМД, в частности Дейт К Дж , признают, что традиционная теория проектирования РБД пока далека от совершенства, а “проектирование БД – это скорее искусство, чем наука” Это связано с тем, что проектные решения принимаются исходя из анализа предполагаемых схем отношений без учета реальных данных

Нередко данные табличного вида (ДТВ), которые нуждаются в автоматизированной обработке, находятся вне БД и даже вне ЭВМ Применение методов и автоматизированных средств проектирования РБД с использованием существующих ДТВ, с одной стороны, позволит свести к минимуму недостатки современной теории, которая вынуждена отталкиваться

от гипотетических данных, а, с другой стороны, в случае необходимости позволит выполнить эффективное преобразование ДТВ в РБД

Проблема заключается в отсутствии методологии и инструментальных средств проектирования РБД на основе существующих ДТВ

Предметом исследования являются модели, методы, алгоритмы и методики принятия решений в ходе выполнения проектных процедур при разработке РБД на основе ДТВ

Цель и основные задачи исследования. Цель исследования – повышение *оперативности и эффективности* проектирования целостных, непротиворечивых, неизбыточных РБД путем разработки методологии проектирования РБД с использованием ДТВ

В соответствии с поставленной целью в диссертации решаются следующие задачи

1 Системный анализ традиционных методов и моделей, используемых при проектировании РБД в условиях использования ДТВ

2 Разработка моделей ДТВ и РБД, моделей таблиц ДТВ и реляционных таблиц

3 Определение состава методов для обеспечения эффективного решения проблем проектирования РБД на основе ДТВ

4 Разработка операторной и сетевой моделей преобразования ДТВ в РБД

5 Исследование методологии преобразования ДТВ в РБД на основе использования сетевой модели

6 Разработка метода преобразования таблиц ДТВ в реляционные таблицы

7 Разработка метода назначения ключевых полей в заполненных таблицах ДТВ.

8 Разработка метода приведения заполненных реляционных таблиц к четырем нормальным формам

9 Разработка метода выявления и формирования всех типов связей между заполненными реляционными таблицами

10 Программная реализация компонентов системы и ее исследование

Методы исследования.

При разработке формальных моделей, методов и методик в диссертации использовались реляционная алгебра, реляционное исчисление, ис-

числение предикатов, теория множеств, алгебра логики, теория операторов, теория алгоритмов, аппарат сетей Петри

Научную новизну работы определяет концепция и теоретические основы проектирования РБД на основе ДТВ

Научные результаты, выносимые на защиту:

- Впервые исследована проблема проектирования реляционных БД с использованием данных табличного вида
- Построены модели данных табличного вида и реляционных таблиц
- Впервые предложен метод преобразования данных табличного вида в реляционные таблицы
- Предложен оригинальный метод приведения заполненных таблиц ДТВ к нормальным формам
- Впервые разработан метод назначения первичных ключей в заполненных таблицах ДТВ
- Предложен оригинальный метод выявления и формирования всех видов связей между заполненными реляционными таблицами
- Предложена новая методология проектирования РБД с использованием ДТВ на основе комплексного применения разработанных моделей, методов и методик

Достоверность научных положений, рекомендаций и выводов

Обоснованность научных положений, рекомендаций и выводов, изложенных в работе, определена корректным использованием современного математического аппарата, проверкой адекватности аналитических моделей Достоверность положений и выводов диссертации подтверждена положительными результатами внедрения на предприятиях, организациях и в учебном процессе

Практическая ценность и реализация результатов работы

Научные результаты, полученные в диссертации, доведены до практического использования Они представляют непосредственный интерес в области проектирования РБД на базе ДТВ Методы, алгоритмы и программные средства могут быть использованы при решении задач проектировании РБД государственными и коммерческими предприятиями и организациями

Кроме того, каждый из разработанных методов может быть использован для решения частных задач при проектировании БД на основе ДТВ, а также для улучшения характеристик существующих БД К этим задачам относятся назначение ключевых полей, исключение сложных атрибутов, избыточности и противоречий в таблицах, приведение значений атрибутов

к одному типу, исключение транзитивных и многозначных зависимостей, формирование связей между таблицами

Разработанные методы и алгоритмы прошли апробацию и внедрены для практического применения на предприятиях ОАО МосОТИС при преобразовании электронных таблиц в форматы неизбыточных, непротиворечивых и целостных реляционных БД, ФГУП НИИ Автоэлектроника при преобразовании текстовых таблиц каталогов в форматы неизбыточных, непротиворечивых и целостных реляционных БД, ООО "Центрстрой" при автоматизированной обработке текущей информации, Союзе кинологических организаций России при сборе и обработке информации из регионов, МГТУ им Н Э Баумана в учебном процессе

Содержание отдельных разделов и диссертации в целом было доложено и получило одобрение

- на Российских, межрегиональных и международных научно-технических конференциях и семинарах (1985 – 2006 г г),
- на заседании кафедры "Компьютерные системы и сети" МГТУ им Н Э Баумана,
- на научном семинаре кафедры "Информатизация структур государственной службы" Российской академии государственной службы при Президенте Российской Федерации

Совокупность научных положений, идей и практических результатов исследований в области информатизации составляет новое направление в области теоретических и практических методов проектирования РБД

Структура работы соответствует списку перечисленных задач, содержит описание разработанных методов, моделей, алгоритмов и методик

Диссертационная работа состоит из введения, пяти глав и заключения, опубликованных на 382 страницах машинописного текста, содержит 188 рисунков, 65 таблиц, список литературы из 214 наименований и 3 приложения

ОСНОВНОЕ СОДЕРЖАНИЕ РАБОТЫ

Во введении показана актуальность решаемой проблемы, сформулирована цель и задачи исследования, приводится краткое описание содержания глав диссертации

В первой главе «Анализ проблем разработки методологии проектирования реляционных баз данных с использованием данных табличного вида» выполнен аналитический обзор современной методологии проектирования реляционных баз данных, сформулированы ее достоинства и не-

достатки В результате исследований по разработке методологии проектирования РБД с использованием существующих ДТВ

- сформулированы мотивы разработки методологии,
- выполнен анализ проблем разработки методов в рамках методологии,
- выполнен анализ применимости современных теоретических и практических разработок для решения этих проблем,
- выполнена постановка задачи разработки методологии,
- определен состав методов, разрабатываемых в рамках методологии

Проведенный анализ современной методологии проектирования РБД позволил выделить четыре этапа проектирования формулировка и анализ требований, концептуальное проектирование, датологическое проектирование, физическое проектирование

Формулировка и анализ требований связаны с определением сферы применения БД, сбором информации об использовании данных, выделением информационных потоков, определением семантики, экспертированием, определением целей разработки Этот этап заканчивается техническим заданием (ТЗ) на БД При наличии ДТВ ТЗ в значительной степени уже сформулировано Действительно, определен состав данных, их семантика, имеются сведения об информационных потоках, имеются сведения об использовании данных

Концептуальное (инфологическое) проектирование посвящено построению модели предметной области Оно строится на основе применения теории РБД с использованием РМД Именно на этом этапе реляционный подход к проектированию БД проявляет себя в полной мере Он позволяет выявлять концептуальные ошибки в проекте БД, закладывать эффективные решения на ранних этапах ее разработки И именно на этом этапе в наибольшей мере проявляется специфика подхода проектирования РБД на основе ДТВ.

Датологическое проектирование связано с построением модели данных на основе инфологической модели Причем в качестве модели данных используются сами данные На этом этапе, как и на предыдущем, проверяется адекватность модели, ее непротиворечивость и расширяемость. В диссертации показано, что при проектировании РБД на основе ДТВ достигается слияние инфологического и датологического этапов проектирования

Физическое проектирование позволяет привязать датологическую модель к среде хранения На этом этапе осуществляется выбор носителя

данных, выбор внутренних форматов их хранения, выбор методов доступа к данным, методов сжатия данных Этот этап мало связан с моделью данных, и практически не зависит от того, использовались ли ДТВ при проектировании РБД

В первой главе рассмотрена реляционная модель данных (РМД). Основным понятием РМД является отношение, которое представляет собой подмножество декартового произведения доменов $D_1, D_2, \quad ,D_k$ вида $D=D_1 \times D_2 \times \quad \times D_k$, где $D_1=(d_{11},d_{12}, \quad ,d_{1i}, \quad ,d_{1m1})$, $D_2=(d_{21},d_{22}, \quad ,d_{2i}, \quad ,d_{2m2})$, $D_k=(d_{k1},d_{k2}, \quad ,d_{ki}, \quad ,d_{kmn})$

Домен – множество элементов, типы которых могут не совпадать

Отношение R: $R \subseteq D=D_1 \times D_2 \times \quad \times D_k$.

Ближайший аналог отношения – таблица Атрибутам отношения соответствуют заголовки столбцов, а телу – содержимое таблицы Заголовки называют схемой отношения Основные свойства отношения следующие

- каждая строка представляет собой кортеж из k значений,
- каждый кортеж содержит точно одно значение (соответствующего типа) для каждого атрибута,
- порядок столбцов фиксирован (1, 2, , k),
- порядок строк произволен,
- любые две строки различаются хотя бы одним элементом

На основе этих требований можно судить о некоторых проблемах представления данных в виде реляционных таблиц в процессе традиционного инфологического проектирования БД В частности

- Из-за отсутствия реальных данных неочевиден выбор атрибута или атрибутов, которые обеспечили бы уникальность записей Этот выбор субъективен и далеко не всегда лучший Использование ДТВ для назначения соответствующих атрибутов позволяет этот процесс формализовать и добиться наилучшего решения.
- Назначение типа для каждого атрибута также субъективно и впоследствии при заполнении таблиц данными может оказаться неверным Назначение типа атрибутов в ДТВ формализуется, так как основывается не на опыте и интуиции разработчика, а на анализе реальных данных

Рассмотрена концепция *функциональной зависимости (ФЗ)* Пусть R является отношением, а X и Y – произвольными подмножествами множества атрибутов отношения R Тогда Y функционально зависимо от X, $X \rightarrow Y$, когда для любого допустимого значения переменной отношения R каждое значение множества X отношения R связано точно с одним значением

множества Y отношения R Анализ *ФЗ* и мнений экспертов позволил сделать следующие выводы

• Единственный способ определения функциональных зависимостей для схемы отношения заключается в том, чтобы внимательно проанализировать семантику атрибутов В этом смысле зависимости являются фактически высказываниями о реальном мире Они не могут быть доказаны

• Таким образом, при традиционном проектировании РБД выявление функциональных зависимостей – процесс трудоемкий и субъективно зависимый Вероятность необнаружения всех функциональных зависимостей при анализе "вручную" велика А невыявленные функциональные зависимости могут сказаться на качественных характеристиках проектируемой РБД, на ее целостности, непротиворечивости, неизбыточности

• Процесс выявления функциональных зависимостей на основе имеющихся данных подлежит формализации, что позволит свести к минимуму не выявленные функциональные зависимости, существенно ускорить процесс проектирования РБД

Рассмотрены принципы *нормализации отношений* – аппарата ограничений на формирование отношений Он позволяет устранить дублирование, обеспечивает непротиворечивость хранимых данных, уменьшает затраты на ведение БД Анализ принципов нормализации и мнений экспертов позволил сделать следующие выводы

• Принципы нормализации при традиционном проектировании РБД являются не более и не менее чем соображениями здравого смысла, записанными в формальном виде

• Попытка формализации процесса выявления и исключения транзитивных и множественных зависимостей на базе анализа только схемы отношения вряд ли может быть успешной Это связано с тем, что даже профессиональный разработчик далеко не всегда может сформулировать критерии зависимостей, даже для конкретного отношения, опираясь только на схему отношения. В случае наличия реальных данных в таблицах ДТВ, выявление и исключение зависимостей можно автоматизировать, и выполнить нормализацию данных в полном объеме

Рассмотрены вопросы *семантического моделирования данных.* В процессе семантического моделирование используются понятия *сущностей и связей* между ними. При отсутствии реальных данных не очевидны участники связей и их типы Из контекста ДТВ можно автоматически выявить не только реальные связи, но и определить их типы

Таким образом, в результате анализа положений традиционной теории проектирования РБД сделаны выводы о том, что, несмотря на ее несомненные достоинства, большинство ее проектных процедур неформализуются, проектные решения субъективны и не гарантируют выбора лучшего варианта При наличии ДТВ проектные решения базируются на анализе реальных данных, что обеспечивает возможность автоматизированного выбора наилучшего варианта

Введено неформальное понятие ДТВ ДТВ обладает следующими свойствами. в них могут отсутствовать разделители строк и столбцов, элементы данных могут размещаться в нескольких строках, типы данных одноименного столбца могут различаться; заголовки ДТВ могут включать в себя подзаголовки и размещаются внутри таблицы

Выполнен анализ мотивов разработки методологии проектирования В результате анализа сделаны следующие выводы

- с точки зрения потребителей ДТВ мотивами преобразования ДТВ в БД являются потребность использования преимуществ БД, необходимость внедрения в существующие БД данных из ДТВ,

- с точки зрения разработчиков методов преобразования ДТВ в БД мотивами их разработки являются сформулированные в работе отличия ДТВ от РБД

Введены формализованные понятия РБД и ДТВ

Схематичное представление модели РБД приведено на рис 1

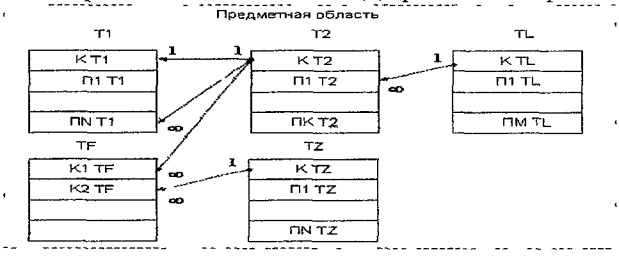


Рис 1. Схематичное представление РБД

На рисунке отображены структуры таблиц и все возможные связи между ними

Префикс *"К"* означает ключ (ключевое поле или поля)

На рисунке приведены возможные связи "1" означает связь одного значения поля, "∞" означает связь с несколькими значениями поля Таким

образом отображены все возможные типы связей – “1 1”, “1 ∞”, “∞ 1”, “∞ ∞” Связь “∞ ∞” реализуется посредством двух связей “1 ∞”

Таким образом, укрупненную модель БД можно представить в следующем виде $T = (T_1, T_2, \ldots, T_i, \ldots, T_K)$, где

T_i – i-я реляционная нормализованная таблица БД,

K – число таблиц в БД,

$T_i = (K_i, П_{i1}, П_{i2}, \ldots, П_{ij}, \ldots, П_{in})$, где

K_i – ключевое поле i-й таблицы, $П_{ij}$ – j-е поле i-й таблицы,

n – число неключевых полей

$S=(S_1, S_2, \ldots, S_j, \ldots, S_m)$, где

S_j – связь между таблицами, m – число связей

$T(S_j)=(T_{j1}, T_{j2})$ – пара таблиц, инцидентных связи S_j,

$N(S_j)=$ “1 1” $\vee$ “1 ∞” $\vee$ “∞ 1” $\vee$ “∞ ∞” – тип связи

Схематичное представление модели ДТВ приведено на рис 2

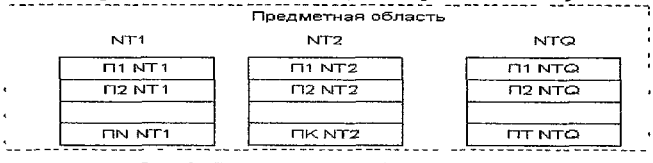


Рис. 2. Схема данных табличного вида

В данном случае известны, и соответственно отображены, только структуры таблиц Следует подчеркнуть, что в отличие от случая, представленного на рис 2 таблицы всегда заполнены данными

Укрупненную модель данных табличного вида можно представить следующим образом

$NT=(NT_1, NT_2, \ldots, NT_i, \ldots, NT_q)$, где q – число таблиц в наборе,

NT_i – i-я нереляционная, ненормализованная таблица набора

$NT_i = (П_{i1}, П_{i2}, \ldots, П_{ij}, \ldots, П_{in})$, где $П_{ij}$ – j-е поле i-й таблицы,

n – количество полей в таблице

По сути, проблема проектирования заключается в преобразовании модели ДТВ к модели РБД Несмотря на сходство представленной укрупненной модели и укрупненной модели БД, очевидны их существенные различия В частности

- NT_i, в отличие от T_i – нереляционные, ненормализованные таблицы,
- NT_i, в отличие от T_i не содержит ключевые поля,
- $NT(S_j) = \varnothing$ – множества связанных таблиц пустые

Этих отличий, даже выявленных на базе укрупненных моделей, достаточно для того, чтобы сформулировать проблемы преобразования

Выполнен анализ проблем разработки методологии проектирования

Проблема приведения таблиц ДТВ к реляционному виду

1 $R=(A_1, \quad ,A_i, \quad ,A_k)$, $i=\overline{1,k}$, где k – степень отношения R, A_i – атрибут отношения $A_i=\{e_{i_1}, \quad ,e_{i_j}, \quad ,e_{i_n}\}$, $j=\overline{1,n}$, где n – мощность отношения, e_{i_j} – j-й элемент атрибута A_i

Необходимо обеспечить выполнение условия $(|e_{i_j}|=0)\vee(|e_{i_j}|=1)$

2 Для всех атрибутов необходимо обеспечить выполнение условия $T(e_{i_1})= \quad =T(e_{i_j})= \quad =T(e_{i_n})\}$, $j=\overline{1,n}$, где n – мощность отношения, $T(e_{i_j})$ – тип j-го элемента атрибута A_i.

3 Каждый столбец должен иметь уникальное имя

$A_1\neq, \quad ,\neq A_i\neq, \quad ,\neq A_k$, $i=\overline{1,k}$, где k – степень отношения,

4 Необходимо исключить дублирование записей

$S_1\neq \quad \neq S_i\neq \quad \neq S_n$, $i=\overline{1,n}$, где S_i – i-я запись, n – мощность отношения

$S_i=(e_{i_1}, \quad ,e_{i_j}, \quad ,e_{i_k})$, $j=\overline{1,k}$, где k – степень отношения

Проблема нормализации заполненных таблиц ДТВ

$R=(K_1, \quad ,K_m, \quad ,K_n,A_1, \quad ,A_i, \quad ,A_j, \quad ,A_k)$, $m=\overline{1,n}$, $i=\overline{1,k}$, где

K_m – атрибут, входящий в сложный ключ, A_i – неключевой атрибут

Необходимо обеспечить.

1 $(|A_{i_{j_t}}|=\varnothing)\wedge(|A_{i_j}|=\varnothing)$ 2 $\neg((K_m\to A_i)\vee \quad \vee(K_m\to A_j))$

3 $\neg((A_i\to A_j)\vee \quad \vee(A_n\to A_m))$ 4 $\neg((A_i\to A_j)\wedge(A_n\to A_m))$

Проблема назначения ключевых полей в таблицах ДТВ

$R=(A_1, \quad ,A_i, \quad ,A_k)$, $i=\overline{1,k}$, где k – степень отношения, A_i – множество атрибутов отношения $A_i=\{e_{i_1}, \quad ,e_{i_j}, \quad ,e_{i_n}\}$, $j=\overline{1,n}$, где n – мощность отношения, e_{i_j} – j-й элемент атрибута A_i

Необходимо найти такой атрибут A_i, чтобы обеспечилась истинность выражения $e_{i_1}\neq \quad \neq e_{i_j}\neq \quad \neq e_{i_n}$

Проблема выявления связей между таблицами ДТВ Необходимое условие наличия связи между отношениями A и B

$(\forall b_{pj})(b_{pj} \in Z(B_j))(Ea_{ri})(a_{ri} \in Z(A_i)(b_{pj} = a_{ri}),\ p = \overline{1,q},\ j = \overline{1,k},\ r = \overline{1,m},\ i = \overline{1,n}$, где q – степень отношения B, k – мощность отношения B, m – степень отношения A, n – мощность отношения A, $Z(B_j)$ – значения атрибута B_j, $Z(A_i)$ – значения атрибута A_i

В результате анализа применимости современных разработок для решения сформулированных проблем сделаны выводы

– отдельные положения современной методологии построения реляционных БД могут быть использованы при разработке методов и средств преобразования ДТВ в таблицы реляционных БД Однако необходима разработка новых методов и методик их использования, учитывающих преимущества наличия реальных данных, а также их специфику,

– существующие практические разработки могут быть использованы для решения отдельных задач преобразования небольших по объему ДТВ в РБД Однако необходимы программные средства, позволяющие решать задачи проектирования РБД на основе ДТВ любых объемов

В соответствии с определением методология – это учение о структуре, логической организации, методах и средствах деятельности В качестве структуры объекта исследования необходимо разработать структурные модели РБД и ДТВ, логическую организацию необходимо отразить посредством формализации процесса проектирования, в качестве методов преобразования объекта необходимо разработать и реализовать методы соответствующие сформулированным проблемам.

С учетом вышесказанного построена схема иерархии методов, рис 3

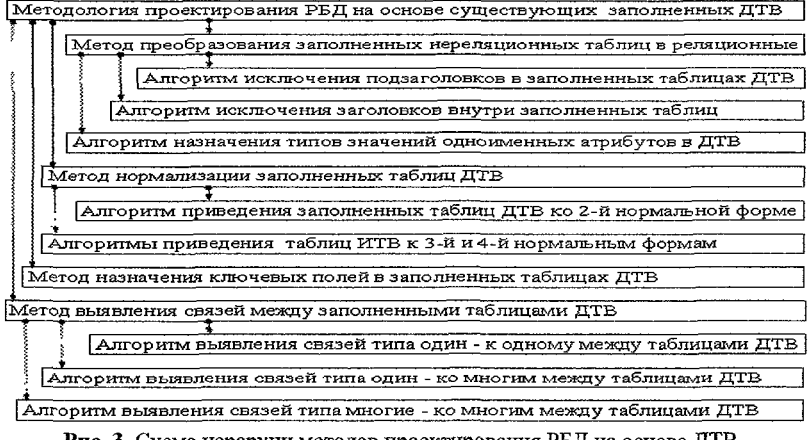


Рис. 3. Схема иерархии методов проектирования РБД на основе ДТВ

Следует обратить внимание на то, что необходимо разработать нормативную методологию, в которой методологические знания выступают в форме предписаний и норм, фиксирующих содержание и последовательность определенных видов деятельности

Во второй главе «Формализация процесса проектирования реляционных баз данных с использованием данных табличного вида» решена задача формализации процесса проектирования РБД на основе ДТВ Разработаны операторная и сетевая модели процесса проектирования Выполнено исследование процесса проектирования на основе использования сетевой модели На основе анализа сетевой модели процесса проектирования скорректирована операторная модель

Постановка задачи формализации процесса проектирования Необходимо разработать приемы, способы и практические операции, подчиненные решению задач преобразования ДТВ в РБД Методы решения частных задач преобразования ДТВ в РБД взаимосвязаны между собой и выполняются в логической последовательности А взаимосвязанные методы, выполняемые в логической последовательности, составляют методологию

Неочевидно, в какой последовательности решать проблемы работы сначала разработать методы, а затем предложить методологию их использования или наоборот – разработать модель процесса проектирования, абстрагируясь от детализации методов, а затем разработать методы в рамках формализации процесса проектирования

В работе выполнен анализ обоих подходов и обоснован выбор компромиссного решения – разработка и исследование модели процесса проектирования для исключения принципиальных ошибок на ранних этапах построения системы После разработки методов – построение методики их использования

В первом случае речь идет лишь о математической модели процесса проектирования Причем модель должна отражать свойства объекта исследования, позволять выявлять концептуальные ошибки в процессе Модель должна быть адекватна начальному представлению процесса проектирования и отражать его заданные свойства с приемлемой точностью В связи с этим выполнены следующие мероприятия

- определен состав компонентов методологии (методов, методик, алгоритмов, специализированных средств),
- сформированы взаимосвязи между этими компонентами,

- выбран математический аппарат, позволяющий описать процесс проектирования и исследовать его характеристики,
- выполнено построение модели процесса проектирования на основе использования выбранного математического аппарата,
- выполнено исследование свойств методики на основе использования ее модели,
- предложенная модель использована в качестве основы для разработки необходимых методов, алгоритмов и специализированных средств, а также в качестве основы для разработки частных и общей методик проектирования РБД на основе ДТВ

Несмотря на то, что большинство объектов методологии не предполагается детализировать на ранних этапах разработки, такие объекты, как модель таблиц ДТВ и модель таблиц РБД необходимо рассмотреть более детально Они являются основными компонентами моделей ДТВ и РБД, и именно отличия этих моделей в основном определяет состав необходимых методов решения проектных задач

Модель данных табличного вида Данные табличного вида представляются множеством $DT = \{Z,D\}$,

где Z – множество заголовков, D – множество данных

$Z=\{Z_1, \quad ,Z_2, \quad ,Z_n\}, \ i=\overline{1,n}, \ n>=1,$

где n – степень множества заголовков

Допустима ситуация, когда $Z_i = Z_m, \ i=\overline{1,n}, \ m=\overline{1,n}, \ i \neq m$,

где n – степень множества заголовков, то есть возможно полное совпадение заголовков

В данных табличного вида возможны подзаголовки 1-го уровня, что формально выглядит следующим образом

$$Z_i = \{PZ_{i1}, \quad , PZ_{ij}, \quad , PZ_{ik}\}, \ j=\overline{1,k}, \ k>=2, \tag{1}$$

где k – степень множества подзаголовков i-го заголовка

$$Z_p = \{PZ_{p1}, \quad , PZ_{pt}, \quad , PZ_{pm}\}, \ t=\overline{1,m}, \ m>=2, \tag{2}$$

где m – степень множества подзаголовков p-го заголовка

Допустима ситуация, когда $PZ_{ij} = PZ_{pt}$ (3)

В данных табличного вида возможны подзаголовки 2-го уровня, что формально выглядит следующим образом

$$PZ_{ij} = \{PPZ_{ij1}, \quad ,PPZ_{ijm}, \quad ,PPZ_{ijf}\}, \ m=\overline{1,f}, \ f>=2, \tag{4}$$

где f – степень множества подзаголовков 2-го уровня ij-го подзаголовка 1-го уровня

$$PZ_{pt} = \{PPZ_{pt1}, \quad , PPZ_{ptr}, \quad , PPZ_{ptg}\}, \ r=\overline{1,q}, \ q>=2, \tag{5}$$

где q – степень множества подзаголовков 2-го уровня pt-го подзаголовка 1-го уровня

Допустима ситуация, когда $PPZ_{ytm} = PPZ_{ptm}$. (6)

$D = \{SD, Z\}$, где SD – множество строк данных (7)

Такого рода представление D допускает наличие нескольких заголовков и подзаголовков 1-го и 2-го уровней, расположенных в области данных В том числе допускается наличие заголовков и подзаголовков, расположенных до, после и между записями $SD=\{SD_1, \ ,SD_i, \ ,SD_n\}$, $i=\overline{1,n}$, $n >> 1$, где n – мощность множества строк данных

$SD_i = \{ED_{i1}, \ ,ED_{ij}, \ ,ED_{ik}\}$, $j=\overline{1,k}$, $k >= 1$,

где k – степень множества элементов данных i-й строки данных, ED_{ij} – элемент данных

Для данных табличного вида должно выполняться следующее правило

$(\forall ED)(ED \in SD)(\exists z(z \in Z)(z \leftrightarrow ED))V(\exists (PZ))(z \in PZ)(PZ \leftrightarrow ED)V(\exists (PPZ))(PPZ \in PZ)(PPZ \leftrightarrow ED)$, где "$\leftrightarrow$" означает взаимооднозначное соответствие

То есть каждому элементу данных соответствует заголовок или подзаголовок 1-го или 2-го уровней и наоборот

$(\forall ED)(ED \in SD)(\exists TED\ (TED \in T(ED)))$,

где $TED=stringV\ integerV\ datetime$, $T(ED)$ – тип ED

То есть каждому элементу данных соответствует определенный тип данных В общем случае

$TED_{11} \neq \quad \neq TED_{i1} \neq \quad \neq TED_{n1}$,

$TED_{1j} \neq \quad \neq TED_{ij} \neq \quad \neq TED_{nj}$,

$TED_{1k} \neq \quad \neq TED_{ik} \neq \quad \neq TED_{nk}$, $i=\overline{1,n}$, $n >> 1$, $j=\overline{1,k}$, $k >= 1$,

где n – мощность множества строк данных, k – степень множества элементов i-й строки данных Другими словами, значения типов данных одного столбца могут не совпадать

Допустима ситуация, когда $SD_i = SD_j$, $i=\overline{1,n}$, $j=\overline{1,n}$, $i \neq j$,

где n – мощность множества данных, то есть возможно полное совпадение строк данных

Модель реляционной таблицы Реляционная таблица (RT) представляется множеством $RT = \{Z, D\}$,

где z – множество заголовков, D – множество данных

$Z = \{z_1, \ ,z_i, \ ,z_n\}$, $i=\overline{1,n}$, $n >= 1$,

где n – степень множества заголовков

Должно быть обеспечено условие $z_i \neq z_m$, $i=\overline{1,n}$, $m=\overline{1,n}$, $i \neq m$, где n – степень множества заголовков, то есть недопустимо совпадение заголовков

$D=\{SD\}$, где SD – множество строк данных (8)

$SD=\{SD_1, \quad ,SD_i, \quad ,SD_n\}$, $i=\overline{1,n}$, $n >> 1$, (9)

где n – мощность множества строк данных

$SD_i = \{ED_{i1}, \quad , ED_{ij}, \quad , ED_{ik}\}$, $j=\overline{1,k}$, $k >= 1$, где k – степень множества i-й строки данных, ED_{ij} – элемент данных

Недопустима ситуация, когда внутри таблицы данных могут встретиться заголовки, то есть должно выполнятся условие

$SD_i \neq z_j$, $i=\overline{1,n}$, $n >> 1$, $j=\overline{1,k}$, $k >= 1$, (10)

где n – мощность множества строк данных, k – степень множества заголовков

Для реляционных таблиц выполняется правило

$(\forall ED)(ED \in SD)(\exists z(z \in Z)(z \leftrightarrow ED))$

То есть каждому элементу данных соответствует только один заголовок

$(\forall ED)(ED \in SD)(\exists TED(ED \leftrightarrow TED))$,

где $TED=string \vee integer \vee datetime \vee real \vee logical$

То есть каждому элементу данных соответствует определенный тип данных В реляционных таблицах обязательно выполнение следующего требования

$TED_{11}= \quad =TED_{i1}= \quad =TED_{n1}$,

$TED_{1j}= \quad =TED_{ij}= \quad =TED_{nj}$,

$TED_{1k}= \quad =TED_{ik}= \quad =TED_{nk}$, $i=\overline{1,n}$, $n>>1$, $k>=1$,

где n – мощность множества строк данных, k – степень множества i-й строки данных, ED_{ij} – элемент данных Другими словами, значения типов данных одного столбца должны совпадать

Недопустима ситуация, когда $SD_i=SD_j$, $i=\overline{1,n}$, $j=\overline{1,n}$, $i \neq j$,

где n – мощность множества данных, то есть невозможно полное совпадение строк данных

Несмотря на некоторое сходство модели данных табличного вида и модели реляционной таблицы, в них имеются существенные различия Сравнивая условия (1–7) в модели данных табличного вида и условия (8 – 10) в модели реляционной таблицы, нетрудно заметить, что эти условия не совпадают В связи с этим для преобразования данных табличного вида в

реляционные таблицы необходимо как минимум добиться выполнения условий (8 – 10)

После детализации моделей таблиц ДТВ и РБД они стали адекватными задачам разработки модели методики

Реализована следующая процедура построения процесса проектирования РБД на основе ДТВ

• На первом этапе (по аналогии с описанием процесса взаимодействия решающих систем), используя отличия моделей ДТВ и РБД, в операторной форме описываются шаги преобразования ДТВ в РБД, формируются связи между ними, определяются правила и порядок их использования Такое описание разработано с целью выявления основных компонент разрабатываемой человеко-машинной системы выявления основных связей между ними, построения модели процесса Под оператором согласно его определению понимается отображение $OP \ X \rightarrow Y$, в котором множества X и Y являются множествами функций с элементами $x(t)$ и $y(t)$ Формально факт преобразования функции $x(t)$ в функцию $y(t)$ посредством выполнения оператора OP отмечается следующим образом $y(t) = OP(x\ (t))$

• На втором этапе операторная модель используется в качестве исходной формализации для разработки модели процесса проектирования, построенной на основе аппарата сетей Петри, и формируется соответствующая сеть

• На третьем этапе с помощью аппарата сетей Петри выявляются и исключаются дефекты модели, а следовательно исключаются дефекты объекта моделирования В конечном итоге строится сетевая модель процесса проектирования, свободная от концептуальных ошибок

• На четвертом этапе с помощью деревьев достижимости анализируются динамические свойства процесса проектирования

Операторная модель преобразования ДТВ в РБД На начальном уровне абстрагирования от большинства компонент человеко-машинной системы схема процесса преобразования ДТВ в РБД может быть проиллюстрирована рис 4

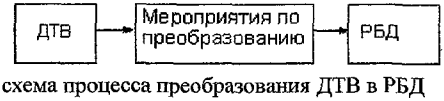


Рис 4 Исходная схема процесса преобразования ДТВ в РБД

Для последовательного развертывания операторной модели в работе рассмотрены все возможные сочетания пунктов несоответствия модели ДТВ и модели РБД ДТВ – нереляционные таблицы, ДТВ – ненормализо-

ванные таблицы, в ДТВ отсутствуют ключевые поля, таблицы ДТВ не связаны между собой Соответствующие постфиксы в операторах модели “р”, “н”, “к”, “с” Индекс “и” используется для оператора импортирования, который задействован при любом сочетании

Таким образом, отслежено 16 вариантов возможных сочетаний несоответствия моделей ДТВ и РБД. Как известно, существуют четыре нормальные формы и четыре типа связей между таблицами, два требования к ключевым полям и несколько требований к реляционным таблицам Поэтому придется отследить 1048576 сочетаний Это нереально, и в первом приближении считается, что процессам нормализации, назначению ключевых полей и назначению связей между таблицами соответствует по одному оператору В результате серии итераций построена операторная модель (рис 5)

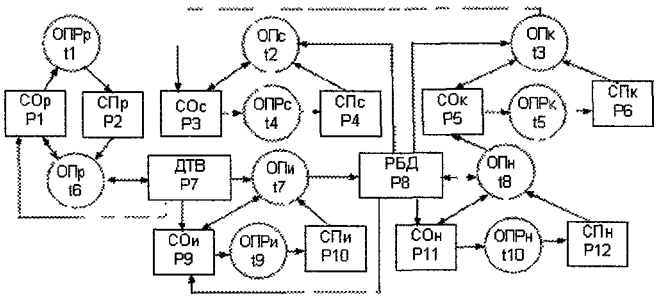


Рис. 5 Окончательная операторная модель преобразования ДТВ в РБД

РБД=ОПи(ОПр(ДТВ,СОр,ОПР(СОр)),СОи,ОПРи(СОи),ОПн(РБД,СОн,ОПРн(СОн)),ОПк(РБДр,СОк,ОПРк(СОк)),ОПс(РБДр,СОс,ОПРс(СОс))

Эта модель отражает процесс проектирования РБД на основе ДТВ С другой стороны, она позволяет сделать заключение о необходимом составе операторов преобразования, реализуемых программными средствами *(ОП)*, операторов преобразования, реализуемых разработчиком *(ОПР)*, систем оценки текущего состояния дел в процессе преобразования *(СО)*, систем преобразований, назначаемых разработчиком *(СП)* Из рис.5 можно сделать вывод о минимальном составе операторов, которые необходимо реализовать в виде средств, алгоритмов и методов для обеспечения процесса проектирования РБД на основе ДТВ К ним относятся операторы, перечисленные ниже

ОП={ОПр,ОПи,ОПн,ОПк,ОПс}, ОПР={ОПРр,ОПРи,ОПРн,ОПРк,ОПРс}, СО={СОр,СОи,СОн,СОк,СОс}, СП={СПр,СПи,СПн,СПк,СПс}

Как следует из сказанного об этапах нормализации, назначения ключевых полей и формировании связей между таблицами необходимо реализовать еще ряд следующих операторов

$ОПн=\{ОПн_1,ОПн_2,ОПн_3,ОПн_4\}$, $ОПРн=\{ОПРн_1,ОПРн_2,ОПРн_3,ОПРн_4\}$,
$СОн=\{СОн_1,СОн_2,СОн_3,СОн_4\}$, $СПн=\{СПн_1,СПн_2,СПн_3,СПн_4\}$,
$ОПс=\{ОПс_1,ОПс_2,ОПс_3,ОПс_4\}$, $ОПРс=\{ОПРс_1,ОПРс_2,ОПРс_3,ОПРс_4\}$,
$СОс=\{СОс_1,СОс_2,СОс_3,СОс_4\}$, $СПс=\{СПс_1,СПс_2,СПс_3,СПс_4\}$

Операторная модель, построенная на основе анализа отличий модели ДТВ и РБД, позволила выделить основные компоненты человеко-машинной системы проектирования, определить порядок их использования, сформировать связи между ними Таким образом, она может быть принята в качестве исходной формализации для разработки методов проектирования РБД на основе ДТВ и методики их использования Однако нет полной уверенности в том, что эта модель свободна от концептуальных ошибок Во избежание ошибок на ранних этапах разработки системы необходимо исследовать следующие ее характеристики

- выполнение баланса входных и выходных информационных потоков для всех систем принятия решения (*устойчивость*),
- достижимость всех состояний человеко-машинной системы (*живость*),
- отсутствие ситуаций, когда система приходит в нестационарное состояние и число информационных потоков превышает критическую отметку (*безопасность*)

Исследование процесса преобразования ДТВ в РБД Несмотря на достоинства операторной модели, она не позволяет выполнить исследование перечисленных характеристик системы С точки зрения представления объектов на начальных этапах разработки модели проектирования в ее рамках используются такие понятия, как ДТВ, РБД, разработчик, методы, системы оценок Причем, эти объекты, как правило, не конкретизируются, а лишь фиксируется факт их наличия, их взаимосвязи, место в системе проектирования, последовательность и правила их использования Такое представление наиболее близко к системному уровню, а на данном уровне преимущественно применяют модели систем массового обслуживания и сети Петри Кроме того, аппарат сетей Петри позволяет исследовать моделируемые процессы на *устойчивость*, *живость* В свою очередь операторная модель служит прототипом для разработки сетевой модели На этом основании и с учетом специфики особенностей объекта моделирова-

ния в качестве основного математического аппарата для моделирования и предварительного исследования процесса преобразования ДТВ в РБД выбран аппарат сетей Петри

В работе выполнен формальный переход от операторной модели к сетевой модели

Сеть Петри – набор $N = (S, T, F)$, где

1 $S=\{s_1, \quad ,s_n\}$ – множество мест,

2 $T=\{t_1, \quad ,t_n\}$ – множество переходов таких, что $S \cap T = \varnothing$

3 $F \subseteq \mu S \times T \times \mu S$ – отношение инцидентности такое, что

$$\forall \langle Q_1^{'},t_1,Q_1^{''}\rangle,\langle Q_2^{'},t_2,Q_2^{''}\rangle \in F\ \langle Q_1^{'},t_1,Q_1^{''}\rangle \neq \langle Q_2^{'},t_2,Q_2^{''}\rangle \Rightarrow t_1 \neq t_2,\ \{t \mid \langle Q,t,Q^{''}\rangle \in F\} = T$$

Операторная модель – набор $M=(C, O, L)$, где $C=CO \cup CП \cup MO$

$CO=\{CO_1, \quad ,CO_q, \quad ,CO_k\}$ – множество систем оценок проектных решений,

$CП=\{CП_1, \quad ,CП_f, \quad ,CП_z\}$ – множество систем принятия решения,

$MO=\{MO_1, \quad ,MO_t, \quad ,MO_p\}$ – множество моделей $CO \cap CП \cap MO = \varnothing$

$O=\{ОПРр, ОПс, ОПк, ОПРс, ОПРк, ОПр, ОПи, ОПн, ОПРи, ОПРн\}$

$L=\subseteq \mu C \times O \times \mu C$ – отношение инцидентности такое, что

$$\forall \langle G_1^{'},o_1,G_1^{''}\rangle,\langle G_2^{'},o_2,G_2^{''}\rangle \in L\ \langle G_1^{'},o_1,G_2^{''}\rangle \Rightarrow o_1 \neq o_2,\ \{o \mid \langle G,o,G\rangle \in L\} = O$$

Соответствие между моделями $C \leftrightarrow S$, $O \leftrightarrow T$, $L \leftrightarrow F$

$|C|=|S|$, $|O|=|T|$, $|L|=|F|$

Места в сети Петри S называют состояниями или положениями P

Сеть Петри является *устойчивой*, если она имеет потоковое назначение $\varphi_t > 0$ для каждого $t_i \in T$, где потоковое назначение – это функция, которая приписывает каждому переходу $t_i \in T$ положительное рациональное число, называемое ее потоком Потоковое назначение для сети Петри должно удовлетворять требованиям. каждая дуга переносит поток, равный потоку перехода, к которому присоединена эта дуга, для каждого положения сумма потоков входных дуг должна равняться сумме потоков выходных дуг

Начальный уровень представления интерактивного взаимодействия в процессе преобразования ДТВ в РБД в форме сети Петри разработан на основе операторной модели (рис 5) и приведен на рис 6

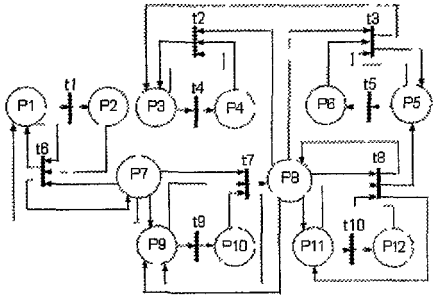


Рис. 6. Сеть Петри для операторной модели преобразования ДТВ в РБД

Пусть для каждого перехода сети рис 5 назначен поток φ_i Для каждого положения P_i записаны уравнения потоков Уравнения сведены в таблицу 1

Таблица 1. Уравнения потоков

	t1	t2	t3	t4	t5	t6	t7	t8	t9	t10	
P1	-φ1					-φ6 +φ6					=0
P2	+φ1					-φ6					=0
P3		+φ2 -φ2	+φ3	-φ4							=0
P4		-φ2		+φ4							=0
P5			+φ3 -φ3		-φ5			+φ8			=0
P6			-φ3		+φ5						=0
P7						-φ6 +φ6	-φ7				=0
P8		-φ2	-φ3				+φ7	-φ8 +φ8			=0
P9							-φ7 +φ7		-φ9		=0
P10							-φ7		+φ9		=0
P11								-φ8 +φ8		-φ10	=0
P12								-φ8		+φ10	=0

Полученные уравнения переписаны в более компактной форме

$-\varphi 1-\varphi 6+\varphi 6=0$, $+\varphi 1-\varphi 6=0$, $-\varphi 2+\varphi 3-\varphi 4=0$, $-\varphi 2+\varphi 4=0$,
$+\varphi 3-\varphi 3+\varphi 8-\varphi 5=0$, $-\varphi 3+\varphi 5=0$, $-\varphi 7-\varphi 6+\varphi 6=0$, $-\varphi 3+\varphi 7+\varphi 8-\varphi 8=0$,
$-\varphi 7+\varphi 7-\varphi 9=-\varphi 7+\varphi 9=0$, $\varphi 8+\varphi 8-\varphi 10=0$, $-\varphi 8+\varphi 10=0$

После преобразования уравнения, и выполнения соответствующих подстановок, получено $\varphi 1=\varphi 2=\varphi 3=\varphi 4=\varphi 5=\varphi 6=\varphi 7=\varphi 8=\varphi 9=\varphi 10=0$

То есть все потоковые значения нулевые Это противоречит условию устойчивости системы В связи с этим рассмотренная сеть не является устойчивой, а, следовательно, процессы, ею описываемые, также являются неустойчивыми

В диссертации выполнена серия итераций исключения противоречий в моделях Конечный результат отражен на рис 7

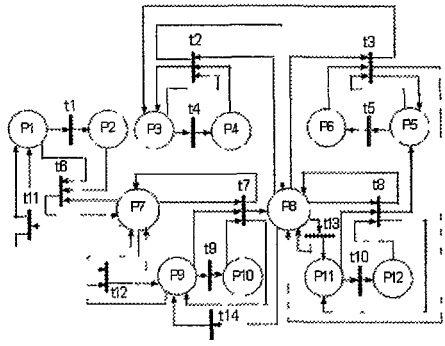


Рис. 7 Окончательная модификация сети Петри

В результате выполненных преобразований исходной сети Петри получена устойчивая сеть, функционирование которой не приведет к коллизиям перемещения информационных потоков А так как полученная сеть взаимно однозначно соответствует модифицированной в соответствии с сетевой моделью операторной модели (рис 8), то модель преобразования ДТВ в РБД является устойчивой, что исключает принципиальные ошибки в процессе проектирования на ранних этапах его разработки и реализации

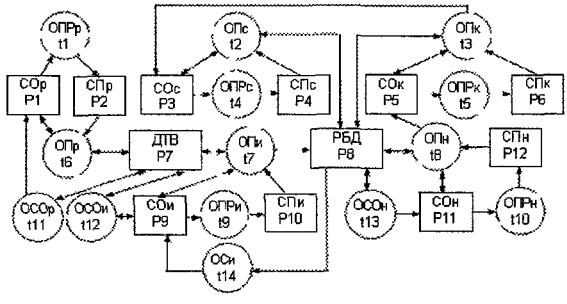


Рис. 8. Окончательная операторная модель преобразования ДТВ в РБД

В результате выполненных преобразований, с одной стороны, получена операторная модель преобразования ДТВ в РБД, свободная от принципиальных ошибок, а с другой стороны, получено формальное описание этой модели, которое может быть использовано для ее дальнейшего исследования и улучшения В связи с тем, что в формальном описании модели выявлены и исключены принципиальные ошибки, то она соответствует реальному моделируемому процессу

Полученная операторная модель позволяет сделать вывод

• о необходимом составе операторов (методов, алгоритмов и средств), которые необходимо разработать для реализации полнофункциональной системы,

• о необходимом составе системы оценок или функций (методов, алгоритмов и средств), которые необходимо разработать для реализации полнофункциональной системы проектирования,

• о связях между методами и средствами в разрабатываемой системе

Кроме того, операторная модель иллюстрирует порядок и правила использования методов и средств, задействованных в процессе решения задач преобразования ДТВ в РБД в разрабатываемой системе

Для более глубокого анализа динамических свойств модели и соответствующего объекта исследования оправданно использование специальных средств – маркерных сетей Петри, которые рассмотрены ниже

Исследование динамических свойств системы Динамику функционирования системы можно моделировать перемещением маркеров в сети в соответствии с правилами перехода

$M'(P)=M(P)-P(t_i)+H(t_i)$, где $M(P)=(M(P1), \quad ,M(PN))$ – разметка сети

Инициация всего процесса преобразования начинается с ДТВ и осуществляется разработчиком Собственно процесс моделирования функционирования системы посредством маркеров состоит в следующем Маркеры перемещаются по сети из положения в положение При этом маркеры могут быть перемещены лишь в том случае, если во всех положениях, связанных с переходом по входу, имеются маркеры

Оператор выполняется лишь тогда, когда все данные, необходимые для его выполнения, имеются в наличии Перемещение маркеров по сети позволит сделать заключения о том, является ли сеть *живой*, то есть, есть ли принципиальная возможность срабатывания всех переходов, является ли сеть *достижимой*, то есть достижимы ли все положения сети

Для визуального отображения перемещения маркеров по сети с демонстрацией всех возможных состояний сети потребовалось 54 рисунка, подобных рис. 6, только с различной разметкой положений В связи со значительным объемом информации, представленной в виде рисунков и плохой обозримости этой информации, использована разметка сети в виде кортежей. В этом случае порядковый номер в кортеже соответствует количеству маркеров в соответствующем положении Начальной маркировке сети соответствует следующая разметка сети

$M1=\{000000100000\}$

Срабатывание перехода *t11* приведет к очередной разметке сети $M2=\{100000100000\}$

Для удобства записи разметок сети и упрощения последующего их анализа последовательность разметок представлена в виде таблицы, фрагмент которой приведен в таблице 2

Таблица 2. Последовательность разметок сети

N	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11	P12	T
1	0	0	0	0	0	0	1	0	0	0	0	0	t0
2	1	0	0	0	0	0	1	0	0	0	0	0	t11
3	0	1	0	0	0	0	1	0	0	0	0	0	t1
4	1	1	0	0	0	1	0	0	0	0	0	0	t11
5	1	0	0	0	0	0	0	0	0	0	0	0	t6
6	0	1	0	0	0	0	0	0	0	0	0	0	t1
7	0	1	0	0	0	0	1	0	0	0	0	0	t0

После просмотра последней 54-й строки таблицы сделан вывод о том, что все переходы сработали, причем многие из них неоднократно Это свидетельствует о том, что полученная сеть – живая, а значит и процесс соответствующей сети – живой То есть имеется принципиальная возможность выполнения всех операторов динамической модели

Таким образом, анализируемая сеть является достижимой, а значит и все состояния соответствующего объекта исследования достижимы

Из анализа таблицы следует, что анализируемая сеть является *безопасной*, то есть выполняется условие

$\forall M(P)\subset R(P)\{\forall P_i/M(P_i)\leq 1\}$, где $i=\overline{1,n}$, $n=|P|$, $R(P)$ – разметки сети

Из этого следует, что моделируемая система функционирует в стационарном режиме

Резюмируя сказанное выше, можно сделать вывод о том, что разработанная сетевая модель обладает следующими свойствами Она *устойчивая, живая, достижимая, безопасная,* в ней отсутствуют тупиковые положения Эта сеть отражает реальный процесс преобразования ДТВ в РБД, и таким образом соответствующая операторная модель обладает теми же свойствами Таким образом, построена операторная модель процесса преобразования, свободная от принципиальных ошибок Она может быть использована в качестве исходной формализации для разработки и реализации этого процесса, а также в качестве основы для разработки соответствующих методов, алгоритмов и средств

В третьей главе «Методы преобразования заполненных нереляционных таблиц в реляционные таблицы и их нормализации» разработан метод преобразования таблиц ДТВ к реляционному виду В рамках метода предложены алгоритмы избавления от сложных атрибутов, приведения значений одноименных атрибутов заполненных таблиц к единому типу,

исключения дублирования записей в текстовых формах представления ДТВ Сформулированы проблемы нормализации заполненных таблиц ДТВ Разработан метод преобразования таблиц ДТВ к четырем нормальным формам

Метод преобразования таблиц ДТВ к реляционному виду

Приведение значений атрибутов заполненных таблиц к одному типу В реляционных таблицах все значения одноименных атрибутов одного типа В таблицах ДТВ данное требование часто не выполняется

Постановка задачи приведения значений одноименных атрибутов отношения к одному типу

Дано отношение $R=(A_1, \ ,A_i, \ ,A_j, \ ,A_k)$, $i=\overline{1,k}$, где k – число атрибутов отношения R, A_i – множество значений атрибута

$A_1=\{a_{1_1}, \ ,a_{1_j}, \ ,a_{1_n}\}$, $j=\overline{1,n}$, где n – число записей отношения R

$A_k=\{a_{k_1}, \ ,a_{k_j}, \ ,a_{k_n}\}$, где a_{i_j} – значение i-го атрибута, расположенного в j-й записи отношения R

Для приведения значений одноименных атрибутов отношения к одному типу необходимо обеспечить

$$(T(a_{1_1})= \ =T(a_{1_j})= \ =T(a_{1_n}))\wedge \ \wedge((T(a_{i_1})= \ =T(a_{i_j})= \ =T(a_{i_n}))\wedge \ \wedge((T(a_{k_1})= \ =T(a_{k_j})= \ =T(a_{k_n})),$$

где $T(a_{i_j})$ – тип значения a_{i_j}

$T(a_{i_j})$ = "логический" $\vee$ "числовой" $\vee$ "символьный" $\vee$ "дата"

В реальных случаях необходимо обеспечить минимальное число типов значений каждого атрибута с тем, чтобы выявить превалирующий тип и с помощью автоматизированных средств привести значения одноименных атрибутов к одному типу Обозначим $КТ_1$ – число типов, которые принимают значения 1-го столбца таблицы, $КТ_i$ – число типов, которые принимают значения i-го столбца таблицы, $КТ_k$ – число типов, которые принимают значения k-го столбца таблицы,

В этом случае целевая функция выглядит следующим образом

$min(КТ_1, \ ,КТ_i, \ , КТ_k)$

В диссертации разработан алгоритм приведения значений одноименных атрибутов отношения к одному типу и доказана его корректность

Суть предложенного алгоритма – выявление наиболее вероятного типа атрибута на основе контекстного анализа его значений и их преобразование в соответствии с выявленным типом В результате преобразований

формируются таблица преобразованных значений и таблица значений, которые не удалось преобразовать Разработчику предоставляется возможность анализа таблицы непреобразованных значений и выбора из предлагаемого списка типа значения

Исключение дублирования записей В реляционных таблицах недопустимо дублирование записей В электронных таблицах, текстовых файлах в отличие от БД данные могут быть выделены цветом или шрифтом, что придает различный смысл разным группам данных, даже если они совпадают

Постановка задачи исключения повторяющихся записей.

Дано отношение $R=(A_1, \ ,A_i, \ ,A_k)$, $i=\overline{1,k}$, где k – число атрибутов отношения

$S_1=(a_{1_1}, \ ,a_{1_i}, \ ,A_{1_k})$ – кортеж значений 1-й записи

$S_j=(a_{j_1}, \ ,a_{j_i}, \ ,A_{j_k})$ – кортеж значений j-й записи

$S_n=(a_{n_1}, \ ,a_{n_i}, \ ,A_{n_k})$ – кортеж значений n-й записи

a_{j_i} – i-е значение j-й записи

Для исключения повторяющихся записей необходимо обеспечить истинность выражения

$S_i \neq \ \neq S_j \neq \ \neq S_n$

В работе предложен алгоритм исключения повторяющихся записей и доказана его корректность

Суть метода приведения нереляционных таблиц к реляционному виду состоит в преобразовании модели ДТВ в модель РБД в соответствии с формализованной методикой, постановками задач и предложенным комплектом алгоритмов

Другими словами, метод состоит в контекстном анализе значений таблиц ДТВ, выявлении несоответствий требованиям к реляционным таблицам и исключению этих несоответствий на основе применения предложенной формализации

Метод преобразования таблиц ДТВ к нормальным формам

Проблемы нормализации ДТВ В таблице 3 представлен общий вид заполненной реляционной таблицы.

Таблица 3 Общий вид реляционной таблицы

A_1	A_2		A_i		A_j		A_k
a_{11}	a_{12}		a_{1i}		a_{1j}		a_{1k}
a_{21}	a_{22}		a_{2i}		a_{2j}		a_{2k}
a_{n1}	a_{n2}		a_{ni}		a_{nj}		a_{nk}
a_{m1}	a_{m2}		a_{mi}		a_{mj}		a_{mk}

Здесь $A=(A_1,A_2, \quad ,A_i, \quad ,A_j, \quad ,A_k)$ – множество атрибутов (заголовков) таблицы (отношения)

$a=((a_{11},a_{12}, \quad ,a_{1i}, \quad ,a_{1j}, \quad ,a_{1k}), \quad ,(a_{21},a_{22}, \quad ,a_{2i}, \quad ,a_{2j}, \quad ,a_{2k}), \quad ,(a_{n1},a_{n2}, \quad , a_{ni}, \quad ,a_{nj}, \quad ,a_{nk}), \quad ,(a_{m1},a_{m2}, \quad ,a_{mi}, \quad ,a_{mj}, \quad ,a_{mk}))$ – множество кортежей значений атрибутов

К данной таблице предъявляются следующие требования

1 $\forall A_i\ (A_i \subset A) \neg \exists (\ A'_i \subset A_i), \quad i=\overline{1,k}, \quad A=(A_1,A_2, \quad ,A_i, \quad ,A_j, \quad ,A_k)$

Другими словами все атрибуты отношения должны быть неделимы

2 $(a_{11} \neq a_{21} \neq \quad \neq a_{n1} \neq \quad \neq a_{m1}) \vee \quad \vee (a_{1i} \neq a_{2i} \neq \quad \neq a_{ni} \neq \quad \neq a_{mi}) \vee \quad \vee (a_{1k} \neq a_{2k} \neq \quad \neq a_{nk} \neq \quad \neq a_{mk}))$

То есть все значения элементов хотя бы одного из столбцов таблицы должны отличаться друг от друга Это требование обеспечивает наличие первичного ключа Поскольку ключ может включать в себя более одного атрибута, то к предыдущему выражению со связкой "ИЛИ" можно приписать следующее выражение (для двух атрибутов)

$concat(a_{1i},a_{1j}) \neq concat(a_{2i}, a_{2j}) \neq \quad \neq concat(a_{ni},a_{nj}) \neq \quad \neq concat(a_{mi},a_{mj}),$
$i=\overline{1,k}, \ j=\overline{1,k}, \ i \neq j$

Здесь $concat(a_{1i},a_{1j})$ – конкатенация (сцепление) значений атрибутов A_i и A_j

3 $\forall A_i\ (A_i \subset A)(T(a_{1i})=T(a_{2i})= \quad =T(a_{ni})= \quad =T(a_{mi})), \ n=\overline{1,m}$

Здесь $T(a_{2i})$ – тип значения атрибута a_{2i} Другими словами тип значений атрибутов должен совпадать.

4 $concat(a_{11},a_{12}, \quad ,a_{1i}, \quad ,a_{1j}, \quad ,a_{1k}) \neq \quad \neq concat(a_{n1},a_{n2}, \quad ,a_{ni}, \quad ,a_{nj}, \quad ,a_{nk}) \neq \quad \neq concat\ (a_{m1},a_{m2}, \quad ,a_{mi}, \quad ,a_{mj}, \quad ,a_{mk})$

Другими словами, все записи таблицы должны быть уникальны Выше рассмотрен метод приведения данных табличного типа к реляционному виду, в связи с этим для дальнейших выкладок используется таблица 3, удовлетворяющая требованиям к реляционным таблицам

Структура реляционной таблицы не всегда оптимальна Под оптимальностью в данном случае понимается непротиворечивость, отсутствие избыточности, отсутствие сложных зависимостей внутри таблицы Это достигается посредством нормальных форм

Следует обратить внимание на то, что в рассматриваемом случае речь идет не о проектировании отношений, когда разработчик БД может создавать отношения, основываясь на сформулированных условиях нормализации Речь идет о заполненных реляционных таблицах, для которых в общем случае, эти условия не выполнены

Избавление от сложных атрибутов Простые атрибуты – это первое условие нормализации реляционных таблиц При проектировании таблиц БД это условие закладывается изначально В ДТВ оно, как правило, не обеспечивается

Постановка задачи приведения отношений к первой нормальной форме

Дано отношение

$R=(A_1, \quad ,A_i, \quad ,A_k)$, $i=\overline{1,k}$, где A_i – множество атрибутов отношения

$A_i=(A_{i_1}, \quad ,A_{i_j}, \quad ,A_{i_n})$, $j=\overline{1,n}$, где A_{i_j} – j-й подзаголовок первого уровня i–го атрибута

$A_{i_j}=(A_{i_{j_1}}, \quad ,A_{i_{j_t}}, \quad ,A_{i_{j_p}})$, $t=\overline{1,p}$, где $A_{i_{j_t}}$ – t-й подзаголовок второго уровня j-го подзаголовка первого уровня i-го атрибута

Для приведения отношений к первой нормальной форме необходимо обеспечить требование $|A_i|=1$ и $|A_{i_j}|=1$ В работе доказано это утверждение

В работе предложен алгоритм избавления от сложных атрибутов и доказана его корректность В соответствии с предложенным алгоритмом для исключения подзаголовков без потери смысла атрибутов необходимо выполнить конкатенации заголовков и подзаголовков всех уровней, и значениям подзаголовков нижнего уровня приписать значения конкатенаций После этого необходимо удалить строки с заголовками верхнего уровня

Исключение подзаголовков, расположенных внутри таблицы Данные таблицы чередуются со значениями атрибутов Это недопустимо в таблицах, используемых в РБД

Особенность таблицы такого рода состоит в том, что в некоторых ее строках значение имеет только один атрибут Принимается, что такой атрибут является внутренним подзаголовком таблицы Формально условие отсутствия внутренних подзаголовков в таблице выглядит следующим образом $\neg\exists s_j(s_j \in s)\ \forall(a_{jt}=NULL)(a_{jt}\in s_j)$, $j=\overline{1,m}$, $t=\overline{1,k-1}$

Здесь s – множество строк таблицы, a_{jt} – значение t-го атрибута в j-й строке, m – мощность таблицы, k – степень таблицы

Постановка задачи исключения подзаголовков внутри ДТВ Дано отношение $R=(A_1, \ ,A_i, \ ,A_k)$, $i=\overline{1,k}$, где k – число атрибутов отношения

$S_1=(a_{1_1}, \ ,a_{1_i}, \ ,A_{1_k})$ – кортеж значений 1-й записи

$S_j=(a_{j_1}, \ ,a_{j_i}, \ ,A_{j_k})$ – кортеж значений j-й записи

$S_n=(a_{n_1}, \ ,a_{n_i}, \ ,A_{n_k})$ – кортеж значений n-й записи

a_{j_i} – i-е значение j-й записи

Для исключения подзаголовков внутри данных табличного вида необходимо обеспечить истинность выражения

$(|S_1|>1)\wedge \ \wedge(|S_j|>1)\wedge \ \wedge(|S_n|>1)$

В работе предложен алгоритм избавления от подзаголовков внутри таблиц ДТВ и доказана его корректность В соответствии с алгоритмом для исключения внутренних подзаголовков в таблицах ДТВ выявляются записи со всеми пустыми, кроме одного, значениями полей, строится таблица на основе этих значений и формируются связи между таблицами

Нормализация заполненных таблиц с подзаголовками в первом столбце Нередко в ДТВ первый столбец используется для хранения подзаголовков групп Использование такого рода таблиц в составе БД недопустимо

Постановка задачи исключения подзаголовков в первом столбце ДТВ Дано отношение $R=(A_1, \ ,A_i, \ ,A_k)$, $i=\overline{1,k}$, где A_i – множество значений i-го атрибута

$A_1=\{a_{1_1}, \ ,a_{1_n}, \ ,a_{1_m}, \ ,a_{1_f}\}$, $n=\overline{1,f}$, где a_{1_n} – значение 1-го атрибута n-й записи.

$A_1 \supseteq A_1'=\{a_{1_n}, \ ,a_{1_m}\}$, $(a_{1_n}=NULL)\wedge \ \wedge(a_{1_m}=NULL)$

$|A_1'|\approx|A_1|$

Для исключения подзаголовков, расположенных в первом столбце ДТВ необходимо обеспечить истинность выражения:

$(a_{1_n} \neq NULL)\wedge \ \wedge(a_{1_m} \neq NULL)$

В работе предложен алгоритм избавления от подзаголовков в первом столбце таблиц ДТВ и доказана его корректность В соответствии с алгоритмом для исключения подзаголовков в первом столбце в таблицах ДТВ выявляются записи со всеми пустыми, кроме одного, значениями полей,

строится таблица на основе этих значений и формируются связи между таблицами

Преобразование заполненных таблиц ко второй нормальной форме Отношение считается приведенной ко второй нормальной форме, если никакие неключевые атрибуты не являются функционально зависимыми лишь от части ключа

Постановка задачи приведения отношений ко второй нормальной форме Дано отношение $R=(K_1, \ ,K_m, \ ,K_n,A_1, \ ,A_i, \ ,A_j, \ ,A_k)$, $m=\overline{1,n}$, $i=\overline{1,k}$, где $(K_1, \ ,K_m, \ ,K_n)$ – атрибуты, составляющие сложный ключ

$S_1=(K_{1_1}, \ ,K_{m_1}, \ ,K_{n_1},A_{1_1}, \ ,A_{i_1}, \ ,A_{j_1}, \ ,A_{k_1})$, где S_1 – 1-я запись отношения R

$S_t=(K_{1_t}, \ ,K_{r_t}, \ ,K_{n_t},A_{1_t}, \ ,A_{i_t}, \ ,A_{j_t}, \ ,A_{k_t})$, где S_t – t-я запись отношения R

$S_p=(K_{1_p}, \ ,K_{m_p}, \ ,K_{n_p},A_{1_p}, \ ,A_{i_p}, \ ,A_{j_p}, \ ,A_{k_p})$, где S_p – p-я запись отношения R

Для приведения отношений ко второй нормальной форме необходимо обеспечить

$\neg((K_{m_1},A_{i_1}, \ ,A_{j_1}) = \ = (K_{m_t},A_{i_t}, \ ,A_{j_t}) = \ = (K_{m_p},A_{i_p}, \ ,A_{j_p}))$, $m=\overline{1,n}$, $i=\overline{1,k}$, $j=\overline{1,k}$; $i \neq j$

В работе доказана теорема о справедливости данного утверждения

В диссертации предложен алгоритм преобразования заполненных таблиц ко второй нормальной форме и доказана его корректность

В соответствии с алгоритмом для каждого атрибута, входящего в первичный ключ выполняется поиск неключевых атрибутов, находящихся в полной функциональной зависимости Для этого осуществляется проверка равенства значений неключевых атрибутов в случае одинаковых значений атрибутов, входящих в первичный ключ Если такие атрибуты найдутся, то исходная таблица разбивается на две таблицы Для этого осуществляется операция проекции для исходной таблицы по всем столбцам, кроме функционально зависимого столбца Далее осуществляется операция проекции для исходной таблицы по столбцу с функционально зависимым атрибутом и столбцу, который является частью первичного ключа и для которого выявлена функциональная зависимость. Таблица, в которую не включен функционально зависимый атрибут, повторно проверяется на соответствие второй нормальной форме

Преобразование заполненных таблиц к третьей нормальной форме Реляционная таблица считается приведенной к третьей нормальной форме в том случае, если в ней отсутствуют транзитивные зависимости между неключевыми атрибутами

Постановка задачи приведения отношений к третьей нормальной форме

Дано отношение $R=(K_1, \ ,K_m, \ ,K_n,A_1, \ ,A_i, \ ,A_j, \ ,A_k)$, $m=\overline{1,n}$, $i=\overline{1\ k}$,

где $(K_1, \ ,K_m, \ ,K_n)$ – атрибуты, составляющие сложный ключ

$S_1=\{K_{1_1}, \ ,K_{m_1}, \ ,K_{n_1},A_{1_1}, \ ,A_{i_1}, \ ,A_{j_1}, \ ,A_{k_1}\}$, где S_1 – 1-я запись отношения R

$S_t=\{K_{1_t}, \ ,K_{r_t}, \ ,K_{n_t},A_{1_t}, \ ,A_{i_t}, \ ,A_{j_t}, \ ,A_{k_t}\}$, где S_t – t-я запись отношения R

$S_p=\{K_{1_p}, \ ,K_{m_p}, \ ,K_{n_p},A_{1_p}, \ ,A_{i_p}, \ ,A_{j_p}, \ ,A_{k_p}\}$, где S_p – p-я запись отношения R

Для приведения отношений к третьей нормальной форме необходимо обеспечить

$$\neg((A_{i_1}, \ ,A_{j_1})= \ =(A_{i_t}, \ ,A_{j_t})= \ =(A_{i_p}, \ ,A_{j_p})), \ i=\overline{1,k}, \ j=\overline{1,k}, \ i\neq j$$

В работе доказана теорема о справедливости данного утверждения

В диссертации предложен алгоритм преобразования заполненных таблиц к третьей нормальной форме и доказана его корректность

Суть алгоритма сводится к выявлению зависимых атрибутов на основе анализа их значений, формированию из этих атрибутов отдельной таблицы, формированию таблицы из оставшихся атрибутов и формированию связей между ключевыми атрибутами полученных таблиц

Преобразование заполненных таблиц к четвертой нормальной форме Реляционные таблицы приведены к четвертой нормальной форме, если в них отсутствуют многозначные зависимости между неключевыми атрибутами

Постановка задачи приведения отношений к четвертой нормальной форме

Дано отношение R $R=(A_1, \ ,A_i, \ ,A_j, \ ,A_k, \ ,A_m, \ ,A_p)$

$S_1=a_{1_1}, \ ,a_{i_1}, \ ,a_{j_1}, \ ,a_{k_1}, \ ,a_{m_1}, \ ,a_{p_1}$ – кортеж значений 1-й записи

$S_f=a_{1_f}, \ ,a_{i_f}, \ ,a_{j_f}, \ ,a_{k_f}, \ ,a_{m_f}, \ ,a_{p_f}$ – кортеж значений f-й записи

$S_z=a_{1_z}, \ ,a_{i_z}, \ ,a_{j_z}, \ ,a_{k_z}, \ ,a_{m_z}, \ ,a_{p_z}$ – кортеж значений z-й записи

$S_q=a_{1_q}, \ ,a_{i_q}, \ ,a_{j_q}, \ ,a_{k_q}, \ ,a_{m_q}, \ ,a_{p_q}$ – кортеж значений q-й записи

$M1=\{(a_{i_1}, \ ,a_{j_1}), \ ,(a_{i_f}, \ ,a_{j_f}), \ ,(a_{i_z}, \ ,a_{j_z}), \ ,(a_{i_q}, \ ,a_{j_q})\}$ – множество сочетаний значений одноименных кортежей

$M2=\{(a_{k_1}, ,a_{m_1}), ,(a_{k_f}, ,a_{m_f}), ,(a_{k_z}, ,a_{m_z}), ,(a_{k_q}, ,a_{m_q})\}$ – другое множество сочетаний значений одноименных кортежей

$M1 \neq M2$ $M1' \subseteq M1$, $M1'=\{(a_{i_f}, ,a_{j_f}), ,(a_{i_z}, ,a_{j_z})\}$

$M2' \subseteq M2$, $M2'=\{(a_{k_f}, ,a_{m_f}), ,(a_{k_z}, ,a_{m_z})\}$

Для приведения отношений к четвертой нормальной форме необходимо обеспечить истинность выражения

$$\neg(((a_{i_f}, ,a_{j_f})= =(a_{k_f}, ,a_{m_f}))\wedge((a_{i_z}, ,a_{j_z})= =(a_{k_z}, ,a_{m_z})))$$

В диссертации предложен алгоритм преобразования заполненных таблиц к четвертой нормальной форме и доказана его корректность

В соответствии с алгоритмом в таблице ищутся кортежи атрибутов, конкатенации значений которых встречаются неоднократно Если таких групп кортежей находится больше двух, то имеет место многозначная зависимость Для избавления от этой зависимости формируются таблицы с найденными кортежами атрибутов В них исключаются повторяющиеся записи Затем для каждой пары полученных таблиц строится таблица, которая обеспечивает связи записей этой пары между собой – таблица связей Таблица связей строится на основе анализа исходной таблицы

Суть метода нормализации таблиц ДТВ состоит в приведении модели ДТВ к модели РБД, которая отвечает условиям нормализации в соответствии с формализованной методикой, постановками задач и предложенным комплектом алгоритмов

Другими словами метод состоит в контекстном анализе значений таблиц ДТВ, выявлении всех видов функциональных зависимостей, исключении этих зависимостей, формировании нормализованных таблиц и связей между ними на основе применения предложенной формализации

В четвертой главе «Методы назначения ключевых полей в заполненных таблицах и формирования связей между ними» разработан метод назначения ключевых полей в заполненных таблицах ДТВ Разработан метод выявления связей между таблицами ДТВ В рамках метода предложены алгоритмы выявления связей “один к одному”, “один ко многим”, “многие ко многим”

Метод назначения ключевых полей в заполненных таблицах В реляционных таблицах практически всегда необходим первичный ключ Он используется для обеспечения уникальности записей, а также для органи-

зации связей между таблицами Первичный ключ должен удовлетворять требованиям уникальности и минимальности

Постановка задачи назначения ключевых полей в таблицах ДТВ

Дано отношение R

$R=(A_1, \ ,A_i, \ ,A_j, \ ,A_k)$, $i=\overline{1,k}$, где k – число атрибутов отношения,

A_i – i-й атрибут отношения

$S_1=(a_{1_1}, \ ,a_{1_i}, \ ,a_{1_j}, \ ,a_{1_k})$, где S_1 – 1-я запись отношения R

$S_n=(a_{n_1}, \ ,a_{n_i}, \ ,a_{n_j}, \ ,a_{n_k})$, где S_n – n-я запись отношения R

$S_m=(a_{m_1}, \ ,a_{m_i}, \ ,a_{m_j}, \ ,a_{m_k})$, где S_m – m-я запись отношения R

M – число записей, a_{n_j} – значение j-го атрибута n-й записи

Для назначения ключевых полей в таблицах ДТВ необходимо обеспечить выявление набора атрибутов, при которых выполняется условие

$$((a_{1_i}, \ ,a_{1_j}) \neq \ \neq (a_{n_i}, \ ,a_{n_j}) \neq \ \neq (a_{m_i}, \ ,a_{m_j})) \wedge \min \left|(a_{n_i}, \ ,a_{n_j})\right|$$

Пусть отношение R_1, на котором построена таблица базируется на p – атрибутах $A=\{A_1,A_2, \ ,A_p\}$ Атрибут A_i имеет в таблице множество значений $K_i=\{K_{i_1},K_{i_2}, \ ,K_{i_n}\}$ Атрибут A_j имеет в таблице множество значений $K_j=\{K_{j_1},K_{j_2}, \ ,K_{j_m}\}$ Так как при этом анализируемая таблица имеет фиксированное количество записей, то n и m имеют конкретные значения и в рассматриваемом случае $n = m$ и равно числу записей таблицы Необходимо найти кортеж атрибутов $KA = \{KA_1, KA_2, \ , KA_r\}$, $A \supset KA$ такой, что все кортежи их значений были уникальны В таком случае мы имеем r – множеств значений по числу атрибутов в кортеже KA, мощность каждого множества n – по числу строк

$KK_1=\{KK_{1_1},KK_{1_2}, \ ,KK_{1_n}\}$, $KK_i=\{KK_{i_1},KK_{i_2}, \ ,KK_{i_n}\}$, $KK_r=\{KK_{r_1},KK_{r_2}, \ ,KK_{r_n}\}$

Тогда должно быть выполнено условие

$$\{KK_{1_1}, \ ,KK_{i_1}, \ ,KK_{r_1}\} \neq \ \neq \{KK_{1_2}, \ ,KK_{i_2}, \ ,KK_{r_2}\} \neq \ \neq \{KK_{1_n}, \ ,KK_{i_n}, \ ,KK_{r_n}\} \quad (11)$$

Таким образом, необходимо выбрать кортеж атрибутов KA такой, чтобы выполнялось условие (11), которое обеспечивает уникальность сложного ключа Также важно обеспечить другое требование к первичному ключу – минимальность Для реализации этого требования нужно стремиться к минимизации числа полей, входящих в первичный ключ, то есть к минимизации размера кортежа KA Кроме того, необходимо учитывать

длину полей и в качестве претендентов на поля, включаемые в первичный ключ, выбирать поля с минимальной длиной

В диссертации предложены алгоритмы назначения первичных ключей в заполненных таблицах ДТВ, учитывающие различные структуры таблиц, и доказана их корректность

Суть алгоритмов состоит в том, что, отталкиваясь от требования минимальности, исключая недопустимые поля (например,"логический", "числовой", "символьный", "дата"), ищется совокупность атрибутов таблицы, суммарные значения которых бы не совпадали

Суть метода назначения ключевых полей в заполненных таблицах состоит в приведении модели ДТВ к модели РБД, которая отвечает условиям целостности в соответствии с формализованной методикой, постановкой задачи и предложенным комплектом алгоритмов

Другими словами, метод состоит в контекстном анализе значений таблиц ДТВ и выявлении кортежей атрибутов, все значения которых удовлетворяют требованиям уникальности и минимальности, а также имеют допустимый для ключей тип

Метод выявления связей между таблицами ДТВ

Выявление связей типа "один к одному" Рассматриваются два отношения $R_1 = (A_1 \quad A_n)$ и $R_2 = (B_1 \quad B_k)$,

где $A_1 \quad A_n$ – множество атрибутов R_1, $B_1 \quad B_k$ – множество атрибутов R_2

В общем виде отношения R_1 и R_2 представлены в таблицах 4, 5

Таблица 4 Отношение R_1

A_1	A_2		A_n
a_{11}	a_{12}		a_{1n}
a_{21}	a_{22}		a_{2n}
a_{m1}	a_{m2}		a_{mn}

Таблица 5. Отношение R_2

B_1	B_2		B_k
b_{11}	b_{12}		b_{1k}
b_{21}	b_{22}		b_{2k}
b_{q1}	b_{q2}		b_{qk}

A_i – первичные ключи отношения R_1; B_j – первичные ключи отношения R_2

$Z(A_i) = (a_{1i}, \ a_{2i} \quad a_{mi})$ – множество значений A.

Формализованное условие наличия связи "один к одному" выглядит следующим образом

Пусть $m \geq q$ и для каждого b_{pj} найдется такое a_{ri}, причем только одно, что $b_{pj} = a_{ri}$, $p = \overline{1,q}$, $j = \overline{1,k}$, $r = \overline{1,m}$, $i = \overline{1,n}$

Тогда между отношениями A и B имеется связь 1 1

И наоборот, пусть $q \geq m$ и для каждого a_{ri} найдется такое b_{pj}, причем только одно, что $a_{ri} = b_{pj}$, $p = \overline{1,q}$, $j = \overline{1,k}$, $r = \overline{1,m}$, $i = \overline{1,n}$, $i = \overline{1,k}$

Тогда между отношениями A и B имеется связь 1 1

Формальные условия наличия связи “один к одному” выглядят следующим образом

$(\forall b_{pj})(b_{pj} \in Z(B_j))(\exists a_{ri})(a_{ri} \in Z(A_i))(b_{pj} = a_{ri}) \wedge (\neg \exists a_{ti})(a_{ti}) \in Z(A_i)(b_{pj} = a_{ti})$,
$p = \overline{1,q}$, $j = \overline{1,k}$, $r = \overline{1,m}$, $i = \overline{1,n}$, $t = \overline{1,m}$, $q \geq m$

$(\forall a_{ri})(a_{ri} \in Z(A_i))(\exists b_{pj})(b_{pj} \in Z(B_j))(a_{ri} = b_{pj}) \wedge (\neg \exists b_{tj})(b_{tj} \in Z(B_j))(a_{ri} = b_{tj})$,
$p = \overline{1,q}$, $j = \overline{1,k}$, $r = \overline{1,m}$, $i = \overline{1,n}$, $t = \overline{1,q}$, $m \geq q$

Постановка задачи выявления связей типа “один к одному”

Даны отношения R_1 и R_2

$R_1 = (A_1, \ ,A_i, \ ,A_k)$, $i = \overline{1,k}$, где A_i – множество значений i-го атрибута

$R_2 = (B_1, \ ,B_j, \ ,B_n)$, $j = \overline{1,n}$, где B_j – множество значений j-го атрибута

$A_i = \{a_{i_1}, \ ,a_{i_n}, \ ,a_{i_t}\}$, $n = \overline{1,t}$, где n – число записей отношения R_1 В записи множества не могут быть использованы повторения элементов

$B_j = \{b_{j_1}, \ ,b_{j_n}, \ ,b_{j_t}\}$, $n = \overline{1,t}$, где n – число записей отношения R_2 В записи множества не могут быть использованы повторения элементов

Для выявления связей типа ”один к одному” необходимо найти такие значения i и j, при которых будет обеспечена справедливость утверждения $A_i = B_j$

В диссертации предложен алгоритм, обеспечивающий выявление связей типа “один к одному” между заполненными таблицами и реализацию мероприятий по отражению этих связей на схемах данных, доказана его корректность

Выявление связей типа “один ко многим” Рассматриваются два отношения $R_1 = (A_1 \quad A_n)$ и $R_2 = (B_1 \quad B_k)$,

где A_i – первичный ключ R_1, B_j – простой атрибут R_2

Условие наличия связей “один ко многим” выглядит следующим образом

$\forall b_{pj}(b_{pj} \in Z(B_j)) \exists a_{ri}(a_r \in Z(A_i)), (b_{pj} = a_{ri})$, где $Z(B_j)$ – значение атрибута B_j, $p = \overline{1,q}$, $r = \overline{1,m}$, $j = \overline{1,k}$, q – мощность отношения R_2, m – мощность отношения R_1, j – степень отношения R_2

Постановка задачи выявления связей типа “один ко многим”

$R_1 = (A_1, \ ,A_i, \ ,A_k)$, $i = \overline{1,k}$, где A_i – множество значений i-го атрибута

$R_2 = (B_1, \ ,B_j, \ ,B_n)$, $j = \overline{1,n}$, где B_j – множество значений j-го атрибута

$A_i = \{a_{i_1}, \ ,a_{i_f}, \ ,a_{i_t}\}$, $f = \overline{1,t}$, где f – число записей отношения R_1 В записи множества не могут быть использованы повторения элементов

$B_j=\{b_{j_1}, \ ,b_{j_f}, \ ,b_{j_p}\}$, $f=\overline{1,p}$, где p – число записей отношения R_2

В записи множества возможны повторения элементов, поэтому, $p \geq t$

Пусть существует подмножество $A_l^{'} \subseteq A_l$

Для выявления связей типа "один ко многим" необходимо найти такие значения i и j, при которых будет обеспечена справедливость утверждения $A_i^{'}=B_j$.

В диссертации предложен алгоритм, обеспечивающий выявление связей типа "один ко многим" между заполненными таблицами и реализацию мероприятий по отражению этих связей на схемах данных, доказана его корректность

Выявление связей типа "многие ко многим"

Пусть A_i – неключевой атрибут отношения R_1,

A_m – другой неключевой атрибут отношения R_1,

B_j – неключевой атрибут отношения R_2,

B_s – другой неключевой атрибут отношения R_2

Тогда между отношениями R_1 и R_2 имеется множественная зависимость, если выполняется следующее условие

$(\exists A_i)(A_i \in R_1)\exists A_m\ (A_m \in R_1)(\exists B_j)(B_j \in R_2)(\exists B_s)(B_s \in R_2)$

$(Z(A_{it})=Z(B_{jp}))(Z(A_{mt})=Z(B_{sp}))$,

где $i=\overline{1,n-1}$, $m=\overline{1,n-1}$, $j=\overline{1,k-1}$, $s=\overline{1,k-1}$

Здесь n – степень отношения R_1, k – степень отношения R_2,

t – номер записи в отношении R_1, p – номер записи в отношении R_2,

$R_1=(A_1, \ ,A_i, \ ,A_n)$, $R_2=(B_1, \ ,B_j, \ ,B_k)$

Следует обратить внимание, что $i=\overline{1,n-1}$, $m=\overline{1,n-1}$, $j=\overline{1,k-1}$, $s=\overline{1,k-1}$ в связи с тем, что ключевые атрибуты не рассматриваются (в каждом отношении предполагается, что для формирования первичного ключа используется по одному атрибуту).

Постановка задачи выявления связей типа "многие ко многим"

Даны два отношения R_1 и R_2

$R_1=(A_1, \ ,A_{i1}, \ ,A_{i2}, \ ,A_k)$, $i1=\overline{1,k}$, где A_{i1} – значения $i1$-го атрибута

$R_2=(B_1, \ ,B_{j1}, \ ,B_{j2}, \ ,B_n)$, $j1=\overline{1,n}$, где B_{j1} – начения $j1$-го атрибута

$A_{i1}=\{a_{i1_1}, \ ,a_{i1_f}, \ ,a_{i1_q}, \ ,a_{i1_t}\}$, $f=\overline{1,t}$, где t – число записей R_1

$A_{i2}=\{a_{i2_1}, \ ,a_{i2_f}, \ ,a_{i2_q}, \ ,a_{i2_t}\}$, $f=\overline{1,t}$, где t – число записей R_1

$B_{j1}=\{b_{j1_1}, \ ,b_{j1_v}, \ ,b_{j1_w}, \ ,b_{j1_p}\}$, $v=\overline{1,p}$, где p – число записей R_2

$B_{j2}=\{b_{j2_1}, \ ,b_{j2_v}, \ ,b_{j2_w}, \ ,b_{j2_p}\}$, $v=\overline{1,p}$, где p – число записей R_2

$A'=\{(a_{i1_f}, \ ,a_{i2_f}), \ ,(a_{i1_q}, \ ,a_{i2_q})\}$, где $(a_{i1_f}, \ ,a_{i2_f})$ – кортежи значений неодноименных атрибутов

$B'=\{(b_{j1_v}, \ ,b_{j2_v}), \ ,(b_{j1_w}, \ ,b_{j2_w})\}$, где $(b_{j1_v}, \ ,b_{j2_v})$ – кортежи значений неодноименных атрибутов

Для выявления связей типа "многие ко многим" необходимо найти такие A' и B', при которых будет обеспечена справедливость утверждения $A' = B'$

В диссертации предложены алгоритмы выявления связей типа "многие ко многим" между заполненными таблицами ДТВ, исключения лишних атрибутов, формирования третьей таблицы для формирования связей между исходными таблицами

Суть метода состоит в автоматизированном выявлении и формировании всех типов связей в модели ДТВ в соответствии с формализованной методикой, постановками задач и предложенным комплектом алгоритмов

Другими словами, метод состоит в контекстном анализе значений всех пар таблиц ДТВ, выявлении в них связанных атрибутов, *определении* типов связи на основе сформулированных условий, отражении выявленных связей в схемах данных

В пятой главе «Программная реализация процесса проектирования РБД с использованием данных табличного вида» выполнена программная реализация процесса проектирования РБД с использованием ДТВ Выполнена разработка информационного, лингвистического, методического, программного и организационного обеспечения системы проектирования Выполнено исследование временных показателей основных модулей системы На основе использования аппарата временных сетей Петри проведено комплексное исследование временных характеристик методики проектирования РБД на базе ДТВ, предложены мероприятия по улучшению этих характеристик

Общие принципы разработки системы С целью сокращения сроков разработки ПО системы интерактивного проектирования РБД на основе ДТВ выполнены следующие мероприятия

• разработка и исследование системы и ее отдельных компонентов, позволяющих выявить принципиальные ошибки в ней на ранних этапах ее создания,

• использование опыта предыдущих разработок автоматизированных систем, а также систем интерактивного взаимодействия,

• анализ требований к разрабатываемой системе в рамках всех этапов ее разработки с целью исключения возможности распространения ошибок проектирования от предыдущих этапов к последующим,

• детальное определение спецификаций системы, унификация лингвистического и информационного обеспечений с целью обеспечения совместимости программных модулей

Исходными данными для разработки программного обеспечения подсистемы являются

• математическое обеспечение подсистемы (глава 3, глава 4),

• сетевые модели интерактивного взаимодействия разработчика и подсистемы в процессе проектирования (глава 2),

• множество информационных и управляющих кадров терминала

При разработке ПО учитывалось то, что характер требований к показателям эффективности программы определяется ее уровнем Чем ниже уровень разрабатываемой программы в иерархии программного обеспечения, тем большую долю в общих затратах машинного времени занимает время ее выполнения

Поэтому наиболее часто используемые программы низкого уровня, в частности программы управления экраном терминала, реализованы на Visual Basic for Applications Программы высокого уровня реализованы посредством макросов и языка SQL

Разработка структуры системы В соответствии со схемой иерархии реализуемых методов (рис 3) сформирована схема иерархии алгоритмов, структурная схема системы

Функциональная схема построена на основе операторной и сетевой модели процесса проектирования Функциональная схема или схема данных – это схема взаимодействия компонентов программного обеспечения с описанием информационных потоков, состава данных в потоках и указанием используемых файлов и устройств (рис 9)

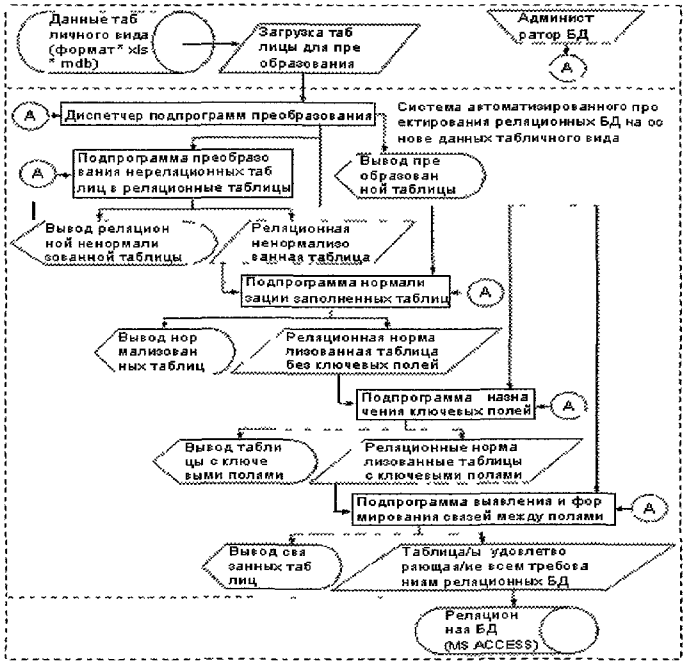


Рис 9. Функциональная схема системы

Реализация проектных процедур системы Для реализации системы выбран язык объектно-ориентированного программирования VBA, интегрированный в среду MS Access Это обусловлено хорошей доступностью данной системы и ее совместимостью с большинством СУБД

Для манипуляций с данными использованы библиотеки объектов данных ADO (ActiveX Data Objects – объектная модель, содержащая ряд специализированных объектов извлечения данных)

Подобно другим объектным моделям, библиотека ADO – это иерархическая система объектов Эта модель состоит из коллекций специфических типов объектов Ядром объектной модели ADO является объект Connection (Подключение), который предоставляет одно подключение к источнику данных OLE DB

Для работы с таблицами используется объект Recordset (Набор данных) Таблица в данном случае извлекается в объект Recordset и представляется в виде набора записей Recordset, с которыми можно проводить раз-

личные манипуляции (удаление, добавление, поиск по различным критериям, обновление и т д), последовательно перемещаясь по всему набору записей

Библиотека объектов ADO позволяет работать с уже имеющимся набором данных, хранимым в БД Однако, когда в системе требуется создавать объекты непосредственно из программы, данной библиотекой воспользоваться нельзя В таких случаях задействована библиотека расширений ADOX (ADO Extensions for DDL and Security) С помощью объектов данной библиотеки можно создавать таблицы, назначать и добавлять ключевые поля, формировать связи между таблицами, что является важным моментом в реализации основных алгоритмов системы

Методика взаимодействия пользователя с системой Для описания технологии взаимодействия пользователя с системой использована функциональная модель В основе системы автоматизированного проектирования РБД с использованием ДТВ лежит преобразование табличной информации в файлы реляционной базы данных Исходными данными являются нереляционные таблицы (формат * xls, * mdb), реляционные ненормализованные таблицы (формат * mdb), а также нормализованные реляционные таблицы без ключевых полей (формат * mdb) Результатом процесса преобразования является таблица, удовлетворяющая требованиям РБД Диаграмма преобразования представлена на рис 10

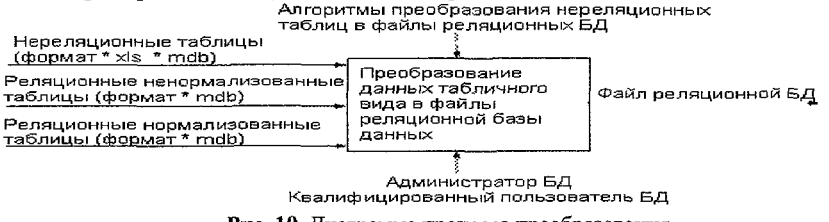


Рис 10 Диаграмма процесса преобразования

Кроме описанной методики в диссертации разработана *технология сопровождения и администрирования* системы.

Экспериментальные исследования компонент системы На основе реальных данных выполнены экспериментальные исследования временных характеристик программных реализаций предложенных методов Результаты части исследований представлены на рис 11, 12

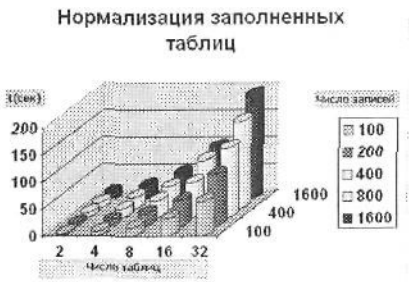


Рис. 11. Диаграмма нормализации

Рис. 12. Диаграмма формирования связей

Как показали исследования, зависимости носят полиномиальный характер.

Следует отметить, что представлена оценка только проектных процедур, реализованных в виде программ. Характеристики процедур, выполняемых разработчиком, значительно уступают характеристикам программных средств и существенно зависят от его опыта и квалификации.

Тем не менее, опыт разработок показывает, что на проектирование баз данных средней сложности в рамках традиционной методологии уходят месяцы работы, тогда как при наличии ДТВ и применении предложенных методов сроки проектирования измеряются неделями.

Комплексное исследование временных характеристик системы. Для реализации возможности исследования временных свойств системы использован аппарат временных сетей Петри, разработанных для моделирования асинхронных конкурирующих систем.

Временная сеть Петри – это пара $<N, \Omega>$, где N – сеть Петри $<S, T, F>$, а Ω – функция, которая каждому переходу t_i в сети приписывает действительное неотрицательное число τ_i. Число $\tau_i = \Omega(t_i)$ соответствует времени срабатывания перехода. Срабатывание перехода возможно, если во входном положении содержится знак. При инициировании перехода из входных положений удаляется знак. Фаза выполнения продолжается τ_1 единиц времени, в конце этого промежутка времени переход срабатывает и знак перемещается в выходное положение перехода. При этом должно выполняться следующее условие, называемое балансом знаков:

$$F(0,P)+T(\tau,t_{i1})+\ldots+T(\tau,t_{in})=F(\tau,P)+I(\tau,t_{j1})+\ldots+T(\tau,t_{jm}),$$

где $T(\tau,t)$ – число терминаций перехода t,

$I(\tau,t)$ – число инициаций перехода t, включая время τ;

$F(0, P)$ – число знаков в положении в начальный момент времени,

$F(\tau, P)$ – число знаков в положении в момент времени τ.

Временные оценки, используемые в моделях, получены на основании анализа сложности решаемых системой задач с использованием экспертных методов, анализа временных оценок проектных процедур

Для отражения динамики функционирования сети и исследования ее временных характеристик оправданно использовать графа переходов

В графе переходов последовательно отображаются переходы и только те положения, в которые перемещаются маркеры после срабатывания этих переходов На рис 13 изображен граф переходов, соответствующий динамике функционирования сети

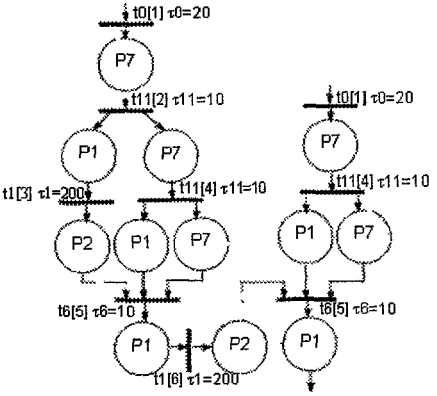


Рис 13. Граф переходов сети

Для переходов $t1$, $t6$ и $t11$ в соответствии с изложенной методикой выполнены предварительные оценки времен их срабатывания в секундах, соответственно $\tau 1=200$, $\tau 6=10$, $\tau 11=10$, $\tau 0=20$ Оценки выполнялись для одной таблицы с числом атрибутов 20 и числом записей 1000, то есть ДТВ представлены всего одной таблицей среднего объема Данные задержки приписаны соответствующим переходам на дереве переходов Исходя из анализа дерева переходов, сформирована таблица осуществимого расписания срабатывания переходов сети (таблица 6)

Таблица 6 Осуществимое расписание срабатывания переходов

Номер срабатывания перехода Идентификатор перехода	1	2	3	4	5	6	7
t0	0,20	20,40	40,60	60,80	80,100		
t11	20 30	30,40	40 50	60,70	70,80	80,90	100,110
t1	30,230	240,440	440,640	640,840			
t6	230,240	440,450	640,650	840,850			

Из таблицы видно, что при определенной последовательности “зажиганий” инициации переходов происходят через равные интервалы време-

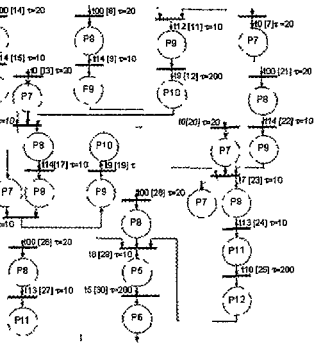
ни Например, в установившемся режиме все “зажигания” перехода $t0$ осуществляются через 20 секунд, ”зажигания” 2 и 3 перехода $t1$ осуществляются через 200 секунд, ”зажигания” 2 и 3 перехода $t6$ осуществляются через 200 секунд. Особый период ”зажигания” зависит от времени срабатывания перехода $t0$, поэтому он составляет то 0, то 20 секунд Из-за особенностей сети переход $t11$, как видно из таблицы, может срабатывать 7 раз, в то время как переходы $t1$ и $t6$ успевают сработать 4 раза. Хотя это несвойственно для системы в целом, для установившейся последовательности зажиганий можно определить скорость срабатывания переходов в единицу времени Из таблицы 6 получены оценки для переходов

Возникает естественный вопрос можно ли увеличить продолжительность отдельной компоненты интерактивной системы, а следовательно, и производительность системы в целом? Оценка скорости выполнения $к$-го перехода определяется по формуле $U_k=n_k/П_к$, где n_k – знаковое содержание контура, включающего $к$-й переход, $П_к$ – период “зажигания” $к$-го перехода

Реализуемой границей скорости вычисления для временной сети Петри в целом является значение $min(n_1 / П_1,\ n_2/П_2,\ \ ,\ n_m / П_m)$, где m – число переходов в сети

Таким образом, для повышения производительности временной сети Петри необходимо увеличить знаковое содержание контура или уменьшить периоды “зажигания” переходов Знаковое содержание контуров в сети соответствует в системе числу процессов, протекающих одновременно в данном режиме Периоду “зажигания” соответствует интервал времени от окончания какого-либо процесса до его повторной инициации, который зависит от времени реакции разработчика и используемых средств преобразования ДТВ в РБД в различных состояниях этого процесса В процессе анализа остальных контуров сети сделан вывод о том, что можно увеличить знаковое содержание сети или распараллелить отдельные процессы проектирования

Для представления динамики функционирования оставшихся компонентов сети, демонстрации живости и достижимости, оценки временных характеристик и выявления возможности повышения ее производительности построен граф переходов (рис 14, 15)



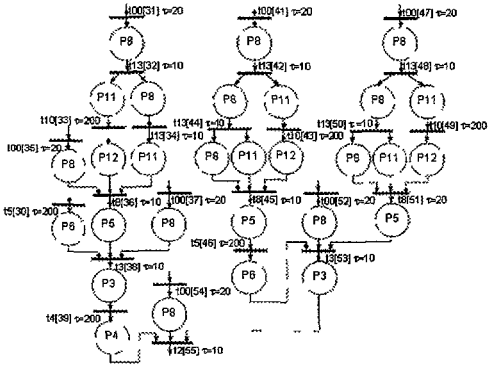


Рис.14 Граф переходов (часть 1) **Рис. 15.** Граф переходов (часть 2)

На основании анализа построенного графа можно оценить минимальное время срабатывания всех переходов и достижения всех положений Это позволило выполнить оценку времени, необходимого для проектирования БД на основе ДТВ, представленной одной таблицей средней сложности С учетом задержки срабатывания переходов и правила их срабатывания, сформирована таблица “зажиганий” переходов (таблица 7)

Таблица 7 Таблица зажиганий переходов

N	Переход	Время срабатывания	N	Переход	Время срабатывания	N	Переход	Время срабатывания
1	t00	20	18	t10	710	35	t13	170
2	t14	30	19	t00	80	36	t13	180
3	t0	20	20	t00	100	37	t10	370
4	t00	40	21	t13	110	38	t8	380
5	t14	50	22	t8	720	39	t5	580
6	t0	40	23	t5	920	40	t00	160
7	t12	60	24	t00	120	41	t00	180
8	t9	260	25	t00	140	42	t13	190
9	t7	270	26	t13	150	43	t13	200
10	t14	280	27	t10	1120	44	t10	390
11	t12	290	28	t13	930	45	t8	400
12	t9	490	29	t8	1130	46	t00	180
13	t0	60	30	t5	200	47	t3	410
14	t00	60	31	t00	140	48	t00	200
15	t14	70	32	t3	1140	49	t2	1350
16	t7	500	33	t4	1340			
17	t13	510	34	t00	160			

Принцип формирования таблицы “зажиганий” переходов поясним на примере перехода *t7* В него входят три поддерева

$\{t00, P8, t14, P9\}$, $\{t0, P7\}$ и $\{\{t00, P8, t10\}, \{t0, P7\}, t12, P9, t9, P10\}$

Время срабатывания перехода *t7* вычисляется следующим образом

$$max((\tau 00+\tau 14), \tau 0, (max((\tau 00+\tau 14), \tau 0)+\tau 12+\tau 9))+\tau 7 =$$

$$max((20+10),20,(max((40+10),20)+10+200))+10=270$$

В соответствии с этим правилом рассчитаны времена срабатывания всех переходов *t* итерации=240+1350=1590 сек

Выполнен анализ возможностей повышения производительности сети Один из способов производительности сети – это повышение ее маркерной нагрузки, что соответствует распараллеливанию моделируемых процессов Процесс нормализации реляционных таблиц тесно связан с процессом формирования связей между таблицами В связи с этим эти процессы часто протекают практически одновременно. Рассмотрена возможность совмещения этих процессов и сделан вывод о том, что такая возможность реальна В результате построены модифицированная сетевая и соответствующая ей операторная модели (рис 16 и рис 17)

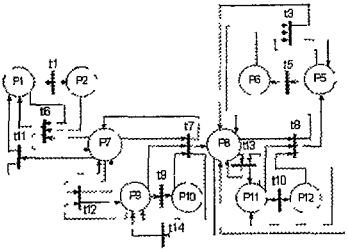


Рис 16. Сетевая модель

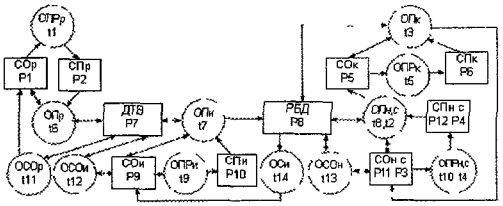


Рис. 17 Операторная модель

Полученные модели исследованы на устойчивость, живость, достижимость Результатом исследований явились мероприятия направленные на исключение концептуальных ошибок в системе

В заключении подведены итоги выполненных исследований и перечислены основные научные и практические результаты, полученные автором:

1 Разработаны и формально описаны модели данных табличного вида и реляционных таблиц, которые использованы в качестве компонентов методологии проектирования реляционных баз данных

2 Выполнена формализация процесса проектирования реляционных баз данных на основе данных табличного вида

3 Формально поставлена задача и предложен метод преобразования данных табличного вида к реляционным таблицам

4 Предложен метод преобразования ненормализованных заполненных таблиц к нормализованному виду

5 Формально поставлена задача и предложен метод назначения ключевых полей в заполненных таблицах

6 Предложен метод выявления связей между заполненными таблицами

7 Разработана система автоматизированного проектирования реляционных баз данных на основе данных табличного вида

8. Разработана методология проектирования реляционных баз данных с использованием данных табличного вида, в основу которой легли предложенные информационные модели, формализация процесса проектирования, методы решения проблем проектирования, методики решения проблем проектирования, программные средства проектирования

9. Разработанная методология успешно внедрена на ряде предприятий различного профиля

ПУБЛИКАЦИИ ПО ТЕМЕ ДИССЕРТАЦИОННОЙ РАБОТЫ

Монографии

1 Брешенков А В Методы решения задач проектирования реляционных баз данных на основе использования существующей информации табличного вида – М Изд-во МГТУ им Н Э Баумана, 2007 – 154 с

2 Балдин А В , Брешенков А В Анализ проблемы проектирования реляционных баз данных на основе использования информации табличного вида и разработка модели методики проектирования – М Изд-во МГТУ им Н Э Баумана, 2007 – 150 с

Научные статьи, опубликованные в центральных журналах

3 Брешенков А В Разработка модели методики проектирования реляционных баз данных на основе использования информации табличного вида//Вестник Московского государственного технического университета – М , 2007 – №2 – С 40–56

4 Брешенков А В. Преобразование заполненных таблиц к первой нормальной форме // Инженерное образование, 2007 – №2 – 14 с (Наука и образование Эл науч издание Номер гос регистрации 0420700025/0005)

5 Брешенков А В Приведение заполненных таблиц к третьей нормальной форме // Инженерное образование, 2007 – №4. – 15с (Наука и образование Эл науч издание Номер гос регистрации 0420700025/0016.)

6 Брешенков А В Приведение заполненных таблиц к четвертой нормальной форме // Инженерное образование, 2007 – №4 – 15с (Наука и образование Эл науч издание Номер гос регистрации 0420700025/0017)

7 Брешенков А В. Исследование методики проектирования реляционных баз данных на основе сетевой модели//Вестник Московского государственного технического университета – М , 2007 – №3 – С. 55–70

8.Брешенков А В Интегрированная среда Access 2002 Построение баз данных с помощью шаблонов//Инженерное образование – 2004 – №9 – 15 с (Наука и образование Эл науч издание Номер гос регистрации 0420400011\0053)

9 Брешенков А В Проектирование таблиц в Access 2002// Инженерное образование, 2004 – №9 – 20 с (Наука и образование Эл науч издание Номер гос регистрации 0420400011\0022)

10 Брешенков А В , Голубкин В Н , Тимофеев В В Подсистема функционально-логического проектирования ЭВА//Вестник Московского государственного технического университета – М , 1990 – №1 – С 43–48

11 Брешенков А В , Голубкин В Н , Проскурина Е А Вопросы методики подготовки специалистов в области вычислительной техники//Вестник Московского государственного технического университета – М , 1995 – №2 – С 54–59

12 Брешенков А В , Голубкин В Н , Гончар А И Способы представления качественных состояний объектов в САПР//Вестник Московского государственного технического университета – М , 1996 – №2 – С 56–53

13 Балдин А В , Брешенков А В Исследование временных свойств системы проектирования реляционных баз данных на основе использования информации табличного вида//Вестник Московского государственного технического университета – М , 2007 – №3 – С 9–23

14. Балдин А В , Брешенков А В Анализ проблемы проектирования реляционных баз данных на основе использования существующей информации табличного вида//Вестник Московского государственного технического университета – М , 2007 – №2 – С 66–80

15. Брешенков А В , Белоус В В Преобразование заполненных таблиц ко второй нормальной форме// Инженерное образование, 2007 – №2 – 16 с (Наука и образование Эл науч издание Номер гос регистрации 0420700025/0005)

Научные статьи и тезисы докладов

16 Брешенков А В Информационное обеспечение интерактивного анализа регистровых структур ЭВМ//Тез доклада Всесоюзной научно-технической конференции "Актуальные проблемы современного приборостроения" – М , 1986 – С 60–61

17 Брешенков А В, Морозов А В Реализация информационной службы разработчика в интерактивной САПР регистровых структур//Тез доклада II Всесоюзной научно-технической конференции "Актуальные

проблемы современного приборостроения" – М , 1988 – С 56

18 Брешенков А В Подготовка документов на ПЭВМ М Машиностроение, 1992 – 100 с

19 Брешенков А В , Черников А С Информационные ресурсы компьютерной сети факультета//Тез доклада Всероссийской научно-методической конференции "Телематика 95" – СПб , 1995 – С 134.

20 Брешенков А В , Бессолицын А А Перспективы и направления развития автоматизированной информационной системы факультета ИУ// Тез доклада научно-технической конференции «165 лет МГТУ им Н Э Баумана» – М , 1995 – С 9–12

21 Брешенков А В , Михайлов С В Перспективы развития автоматизированной системы факультета ИУ//Тез доклада научно-технической конференции «165 лет МГТУ им Н Э Баумана» – М , 1995 – С 77

22 Брешенков А В О перспективах развития информационных технологий при комплектации машин изделиями автотракторного электрооборудования //Автоэлектрооборудование, №1 – М , 2001 – С 7–22

23 Брешенков А В , Бараков Д Д Неформальная постановка проблемы преобразования информации табличного вида в файлы баз данных//Тез доклада Международной научно-технической конференции "Гражданская авиация на современном этапе развития науки, техники, общества" – М , 2003 – С 9

24 Брешенков А В Неформальная постановка проблемы преобразования информации табличного вида в файлы баз данных// Сб трудов АУ МВД России "Актуальные вопросы технологий в деятельности органов внутренних дел" – М , 2004 – С 55–70

25 Брешенков А В , Бараков Д Д Вопросы преобразования электронных таблиц в таблицы реляционных баз данных//Современные информационные технологии Сб трудов каф ИУ–6, посвященный 175-летию МГТУ им Н Э Баумана – М Эликс +, 2004 – С 44–50

26 Брешенков А В , Бондарь М Л Построение системы репликации баз данных на основе протоколирования изменений (алгоритмы синхронизации и односторонней репликации)//Сб. трудов каф ИУ–6, посвященный 175-летию МГТУ им Н Э Баумана – М Эликс +, 2004 – С 31–37

27 Брешенков А В , Бараков Д Д Методика назначения ключевых полей в заполненных реляционных таблицах//Современные информационные технологии Сб трудов каф ИУ–6, посвященный 175-летию МГТУ им Н Э Баумана – М Эликс +, 2005 – С 102–106

28 Брешенков А.В , Устинов Н В Векторное квантование и кластеризация больших объемов данных//Современные информационные технологии Сб трудов каф ИУ– 6, посвященный 175-летию МГТУ им Н Э Баумана. – М · Эликс +, 2005 – С 71–75

29 Брешенков А В , Чехарин Д А Система протоколирования действий пользователя//Современные информационные технологии Сб трудов каф ИУ– 6, посвященный 175-летию МГТУ им Н.Э Баумана – М Эликс +, 2005 – С 55–58

30. Брешенков А В , Бондарь М Л Поддержка целостности данных в алгоритмах оптимистической репликации//Современные информационные технологии Сб трудов каф ИУ–6, посвященный 175-летию МГТУ им Н Э Баумана – М Эликс +, 2005 – С 44–49

31 Брешенков А В Избавление от сложных атрибутов в заполненных нереляционных таблицах// Сб трудов каф ИУ–6 – М Эликс +, 2006 – С 10–15

Учебно-методические труды

32. Брешенков А В , Рубанов А Н Вопросы построения баз данных для организации учебного процесса в физико-математической школе// Современные информационные технологии Сб трудов каф ИУ–6, посвященный 175-летию МГТУ им Н Э Баумана – М Эликс+, 2004 – С 27–30

33 Брешенков А В , Коняхин А Е Оперативная разработка баз данных средствами системы Clarion – М Машиностроение, 1995 – 100 с

34 Брешенков А В Коммерческая база данных Опыт создания средствами системы Clarion – М Машиностроение, 1996 – 100 с

35 Брешенков А В , Губарь А М Проектирование баз данных в среде Access Учебное пособие для вузов – М Изд-во МГТУ им Н Э Баумана, 2006 – 160 с

36 Брешенков А В Проектирование баз данных на основе информации табличного вида Учебное пособие для вузов – М Изд-во МГТУ им Н Э Баумана, 2007 – 200 с.

Автореферат

Диссертация на соискание ученой степени доктора технических наук

Брешенков Александр Владимирович

Тема диссертационного исследования

Методология проектирования реляционных баз данных
с использованием данных табличного вида

Научный руководитель

Изготовление оригинал-макета
Брешенков Александр Владимирович

Подписано в печать 28.06. Тираж 100 экз

Усл п л 2,4

Российская академия государственной службы
при Президенте Российской Федерации

Отпечатано ОПМТ РАГС Заказ № 347

119606 Москва, пр-т Вернадского, 84