САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

**КАФЕДРА КОМПЬЮТЕРНОГО МОДЕЛИРОВАНИЯ И   
МНОГОПРОЦЕССОРНЫХ СИСТЕМ**

**МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ**

Тема: «Комплексная информационная система, предназначенная для верификации моделей конвективных облаков»

|  |  |
| --- | --- |
| Направление: | 02.04.02 – ФИиИТ |
| Магистерская программа: | ВМ.5502.2014 – Вычислительные технологии |
|  |  |
|  |  |
| Выполнил студент гр. 633 | Петров Дмитрий Алексеевич |
| Научный руководитель,  к. ф.-м. н., доцент | Е. Н. Станкова |

Санкт-Петербург – 2016

**Содержание**

[Введение 4](#_Toc451506250)

[Постановка задачи 8](#_Toc451506251)

[Обзор литературы 9](#_Toc451506252)

[Глава 1. Технология консолидации данных 14](#_Toc451506253)

[1.1. Понятие консолидации 15](#_Toc451506254)

[1.2. Основные задачи консолидации и источники данных 16](#_Toc451506255)

[1.3. Обобщенная схема процесса консолидации 17](#_Toc451506256)

[1.4. Понятие ETL-процесса 18](#_Toc451506257)

[1.5. Основные цели и задачи процесса ETL 18](#_Toc451506258)

[1.6. Извлечение данных в ETL 20](#_Toc451506259)

[1.7. Преобразование данных в ETL 21](#_Toc451506260)

[1.8. Загрузка данных в базу данных 22](#_Toc451506261)

[Глава 2. Нестационарная модель конвективного облака 24](#_Toc451506262)

[2.1. Классификация моделей конвективных облаков 25](#_Toc451506263)

[2.2. Описание полуторамерной нестационарной модели конвективного облака 27](#_Toc451506264)

[Глава 3. Проектирование реляционных баз данных 29](#_Toc451506265)

[3.1. Основы реляционной базы данных 29](#_Toc451506266)

[3.2. Этапы проектирования реляционной базы данных 31](#_Toc451506267)

[3.3. Организация реляционной базы данных 32](#_Toc451506268)

[для специализированной метеорологической информации 32](#_Toc451506269)

[Глава 4. Проектирование многомерных баз данных 36](#_Toc451506270)

[4.1. Понятие многомерной базы данных 38](#_Toc451506271)

[4.2. Основы многомерного представления данных 39](#_Toc451506272)

[4.3. Структура многомерного куба 40](#_Toc451506273)

[4.4. Основные операции над измерениями гиперкуба 41](#_Toc451506274)

[4.5. Архитектура OLAP-систем 43](#_Toc451506275)

[4.6. Технические аспекты многомерного хранения данных 46](#_Toc451506276)

[4.7. Загрузка данных в хранилище данных 47](#_Toc451506277)

[4.8. Организация многомерной базы данных для специализированной метеорологической информации 48](#_Toc451506278)

[Глава 5. Описание алгоритма работы системы 52](#_Toc451506279)

[5.1. Выбор источников данных 52](#_Toc451506280)

[5.2. Извлечение данных из гетерогенных источников 58](#_Toc451506281)

[5.3. Преобразование и очистка данных 60](#_Toc451506282)

[5.4. Загрузка данных из гетерогенных источников в БД 64](#_Toc451506283)

[5.5. Моделирование интегральных и спектральных характеристик конвективного облака 65](#_Toc451506284)

[5.6. Загрузка данных в хранилище данных 68](#_Toc451506285)

[5.7. Анализ многомерного куба данных с целью определения диапазонов параметров для конвективного явления 71](#_Toc451506286)

[Глава 6. Тестирование и демонстрация разработанной комплексной информационной системы 73](#_Toc451506287)

[6.1. Тестирование разработанной системы 73](#_Toc451506288)

[6.2. Демонстрация разработанной системы 74](#_Toc451506289)

[Выводы 82](#_Toc451506290)

[Заключение 85](#_Toc451506291)

[Список использованных источников 87](#_Toc451506292)

### Введение

В настоящее время как никогда актуальна проблема прогнозирования опасных конвективных явлений, таких как грозы, град, шквал и обильные осадки в связи с увеличением частоты этих явлений и масштабов разрушений, которые они производят. Информация о наступлении таких явлений нужна в первую очередь аэропортам, авиакомпаниям и службам МЧС.

Для прогнозирования и изучения конвективных явлений используются специально разработанные математические модели. Однако, проверка работоспособности и «настройка» таких моделей очень сложна из-за отсутствия в свободном доступе интегрированных данных о месте и времени наблюдаемых явлений в сочетании с метеорологическими данными, используемых в качестве начальных и граничных условий в этих моделях.

В настоящее время Всемирная метеорологическая организация поддерживает проведение натурных исследований, направленных на получение полной метеорологической информации, которая позволяет осуществлять сравнение результатов моделирования с данными наблюдений за реальными облаками и связанными с ними опасными явлениями погоды. При этом получение таких данных осуществляется в ходе проведения комплексных полевых экспериментов с использованием дорогостоящих приборов, радиолокаторов разных видов, самолетов и планеров. Соответственно, такие эксперименты являются крайне затратными и проводятся очень редко.

Однако, для получения статистически достоверных данных о пригодности модели для прогнозирования, необходимо проводить большие серии численных экспериментов. Для этого необходима программная среда, где такие эксперименты можно было бы проводить в автоматическом режиме с использованием интуитивно понятного интерфейса. Важными компонентами этой среды должны стать реляционная база данных, которая будет хранить консолидированные исходные данные и многомерная база данных, позволяющая анализировать результаты расчетов по модели.

Разработкой специализированных систем, предназначенных для сбора и обработки метеорологической информации, занимаются многие фирмы и научные институты. Например, Raytheon Company разработала систему ITWS (Integrated Terminal Weather System) [58], которая обеспечивает автоматизированный сбор метеорологической информации, предназначенной для использования авиадиспетчерами и руководителями полетов в районе аэродрома. ITWS предоставляет как оперативные данные, так и высокоточные прогнозы ожидаемых погодных условий. ITWS собирает информацию практически из всех доступных источников, включая различные датчики, радиолокаторы и прогностические модели.

ООО «Перспективные методы мониторинга» (Advanced Monitoring Methods) [55], используют SQL (*Structured Query Language –* язык структурированных запросов) для сбора метеорологических данных, их проверки, построения графиков, формирования оповещений и отчетов. Структура базы данных позволяет проводить сравнение и интеграцию данных из различных источников.

Институт радарной метеорологии (IRAM) [57] предоставляет специализированные системы для авиационных метеорологов, осуществляет сбор данных от метеорологических спутников, мировых центров прогнозирования погоды, сетей метеорологических станций и аэродромов.

Метеорологическая система усвоения данных (MADIS) [62] разработана в Национальном управлении океанических и атмосферных исследований (NOAA) и предназначена для сбора, интеграции, контроля качества и распространения данных метеонаблюдений. Данные MADIS можно получить через Интернет, используя специальное программное обеспечение Unidata's Local Data Manager (LDM). Пользователи могут оформить подписку на всю базу данных или запросить только отдельные наборы, которые представляют для них интерес.

Большая часть описанных выше систем предоставляет данные на коммерческой основе (ITWS, DAS, IRAM) и, следовательно, они не всегда могут быть доступны для научно-исследовательских организаций. Кроме того, данные хранятся в большинстве случаев как статические архивы на компьютерных дисках и других внешних носителях, что затрудняет выборку необходимой информации. И что самое главное, имеющиеся метеорологические данные не могут быть непосредственно использованы для верификации моделей конвективных облаков в связи с отсутствием локализованной комплексной информации о виде опасного конвективного явления, месте и времени его наблюдения и о сопровождавших явление атмосферных условиях. Таким образам, имеется возможность извлечь данные о состоянии атмосферы в определенном месте и в определенный день, но невозможно получить информацию из той же базы данных или архива о наличии или отсутствии в данном месте и в данное время какого-либо опасного явления. Для этого необходимо искать другой источник информации.

Кроме верификации модели ее необходимо настроить на определение типа конвективного явления, поскольку многие модели не имеют, например, «электрического» блока, позволяющего однозначно прогнозировать грозовые явления. Мы предлагаем такую настройку проводить с помощью многомерного анализа.

В настоящее время многомерный анализ широко используется в интересах бизнеса для повышения качества принятия решений, быстрого построения всевозможных отчетов по продажам различных товаров в различных торговых точках, а также для решения некоторых прикладных проблем [5, 49]. Мы пытаемся применить преимущества многомерного подхода для анализа специализированной метеорологической информации.

Разработанный нами программный продукт позволит: во-первых, проанализировать источники метеорологической информации и выяснить, в каком виде она хранится, во-вторых, – спроектировать специализированную базу для хранения интегрированных метеорологических данных, в-третьих, – разработать программную среду, предоставляющую интерфейс для доступа к базе интегрированных данных и позволяющую осуществлять численные эксперименты с помощью модели, а также хранить результаты расчетов и впоследствии иметь возможность их анализировать с помощью многомерной базы данных [4].

Система, рассмотренная в выпускной квалификационной работе уникальна по нескольким причинам. Во-первых, она может быть предоставлена на бесплатной основе, что делает ее доступной всем, желающим начать исследования в направлении конвективных облаков. Во-вторых, она может быть подстроена под любые первоначальные источники получения метеорологической информации. И, наконец, в-третьих, благодаря тому, что система построена на принципе OLAP-кубов ее оперативность позволяет обрабатывать большой объем данных за короткие сроки.

### Постановка задачи

Целью работы является разработка программного продукта, предназначенного для интеграции и анализа специализированной метеорологической информации.

Для достижения поставленной цели, необходимо решить следующие задачи:

1. провести анализ источников метеорологической информации, необходимой для верификации физико-математической модели конвективного облака;
2. провести анализ процесса интеграции данных и его типов;
3. осуществить проектирование реляционной базы данных для интегрированных метеорологических данных;
4. провести анализ существующих многомерных хранилищ данных, их функциональных возможностей и подходов к реализации;
5. разработать многомерную базу данных для хранения и анализа интегрированной метеорологической информации;
6. разработать программную среду и провести её тестирование;
7. провести тестовые численные эксперименты, направленные на верификацию модели конвективного облака и прогнозирование типов опасных конвективных явлений с использованием разработанного программного продукта.

### Обзор литературы

При написании данной выпускной квалификационной работы были проанализированы различные источники: научная и учебно-методическая литература, статьи в периодических изданиях, касающиеся технологий многомерных баз данных и консолидации. В результате анализа всех найденных как русскоязычных, так и англоязычных ресурсов по данной тематике был сделан вывод, что основными источниками, раскрывающими теоретические и практические основы технологии консолидации данных и многомерных хранилищ данных, явились работы таких авторов как: Паклин Н. Б., Орешкова В. И «Бизнес-аналитика от данных к знаниям»; Богданов А. В., Станкова Е. Н., Тхурейн К. Л. «Распределенные базы данных»; Барсегян А. А., Куприянов М. С., Степаненко В. В., Холод И. И. Методы и модели анализа данных: OLAP и Data Mining, а также Cameron Scott. Microsoft SQL Server 2008. Analysis Services. Step by Step. При этом лишь отдельные главы данных источников определяют практические аспекты выбранной темы. В основном вся информация носит сугубо теоретический характер. Тем не менее, исходя из знаний, полученных из названных выше источников, было четко сформулировано понимание вопроса.

В научном издании Паклина Н. Б., Орешковой В. И «Бизнес-аналитика от данных к знаниям» вторая глава целиком посвящена консолидации данных. В ней подробно описаны задачи этой технологии, дано ее четкое определение, указаны основные критерии оптимальности, объяснен принцип подбора источников данных. Также присутствует подробная структурная схема всего процесса консолидации данных, с помощью которой читателю не составит труда определить план создаваемой программной среды. В книге обозначены и подробно описаны процессы, которые лежат в основе процедуры консолидации. В пункте 9 главы 2 четко определены цели и задачи комплекса методов ETL, который лежит в основе процедуры консолидации. Что касается многомерного подхода, то рассматривая вторую и четвертую главы можно получить сведения о базовой структуре OLAP-кубов и манипуляций над ними, а также необходимые знания о специфике OLAP-анализа.

Книга Богданова А.В., Станковой Е.Н. и Тхурейна К.Л. «Распределенные базы данных» во многом дублирует предыдущий источник касательно технологии консолидации данных, но отчасти и дополняет имеющуюся информацию. В ней конкретно определены цели консолидации, указаны примеры источников данных, подробно описаны последовательные этапы этого процесса.

Представленный в пособиях Барсегяна А. А., Куприянова М. С., Степаненко В. В., Холода И. И. «Методы и модели анализа данных: OLAP» и Харинатха С., Куинна С. Data Mining и SQL Server «Analysis Services 2005 и MDX для профессионалов» материал содержит принципы разработки и исследования основных методов анализа данных, а также соответствующие подходы к проектированию многомерных хранилищ данных. Информация изложена четко и последовательно, применяемая терминология соответствует общепринятой, пособие в достаточной степени иллюстративно примерами и имеет множество готовых сценариев разработки, что позволяет получить наиболее детальное представление об OLAP-технологии. Каждая глава книги автономна настолько, насколько это возможно, что позволяет достаточно быстро изучить основные концепции, связанные с OLAP-кубами, являющимися одной из ключевых частей системы, представленной в данной работе. Тем не менее, уровень данного пособия не в полной мере соответствует уровню, необходимому для разработки такой комплексной системы, как представленная в этой работе.

Также необходимо отметить, что труд авторов Харинатха С., Куинна С. детально разъясняет практическую сторону вопроса разработки многомерных хранилищ данных и управления ими. Данное пособие в основном ориентировано на рассмотрение конкретных примеров, начав с которых, можно самостоятельно перейти к разработке крупномасштабных хранилищ данных, которые будут полезны для анализа и выявления закономерностей данных.

Таким образом, из перечисленных выше источников был получен весь необходимый объем теоретической информации. Однако, практическая сторона вопроса освещена в научной и учебной литературе более скудно или представленные практические примеры не отвечают масштабу, необходимому для серьезных проектов.

Были проанализированы не только основные, но и смежные отрасли знаний, так как они имеют непосредственное прикладное значение в данной выпускной квалификационной работе. Это такие отрасли как проектирование реляционных баз данных, необходимых для хранения специализированной метеорологической информации; моделирование атмосферных процессов, которое является особенно актуальным, поскольку математическое моделирование – основной метод научного познания закономерностей исследуемой системы, позволяющий совершенствовать объект изучения; проектирование и разработка сложных многоуровневых Web-приложений с помощью СУБД MySQL и языка программирования PHP.

Для изучения процесса проектирования реляционных баз данных были проанализированы издания следующих авторов: Бегга К., Дарвина Х., Дейта К. Дж., Евсеевой О. Н., Ершова Г. Н., Кодда Е. Ф., Коннолли Т., Скакун В. В., Шамшева А.Б. Говоря о степени изученности проблемы с уверенностью можно констатировать, что реляционные базы данных описаны в теоретической и практической литературе весьма широко. Практически в каждом труде, посвященном этой теме, большое внимание уделяется практическому аспекту. Базовыми работами на эту тему можно назвать Кодда Е. Ф. «Реляционная модель данных для больших совместно используемых банков данных» в пер. с англ. М. П. Когаловского и Дейта К. Дж. «Введение в системы баз данных».

Эдгар Кодд является основоположником реляционных баз данных. В 1970 году он опубликовал по данной тематике первую статью, и она по сей день актуальна. Вместе с ним работал над развитием реляционных СУБД Кристофер Дейт, автор классического учебника «Введение в системы баз данных», который как стандартный текст по системам баз данных используется во многих университетах мира.

Таким образом, отрасль проектирования реляционных баз данных уже давно изучена многими учеными и не смотря на свою актуальность утратила научную новизну.

Для анализа моделирования атмосферных процессов основной использованной литературой стали книги и статьи авторов: Ампиловой Н. Б., Веремеева Н. Е., Довгалюк Ю. А., Раба Н. О., Синькевич А. А., Станковой Е. Н. Khain A., Krugliak H., Ovtchinnikov M., Pinsky M., Pokrovsky A.

Анализируя данные источники была получена вся необходимая информация касательно начальных и граничных условий для модели конвективного облака. Это и стало основой для выбора данных-параметров, которые требуется консолидировать, а также определило структуру спроектированной базы данных. Поскольку нестационарные модели конвективных облаков весьма актуальны, они все еще имеют научную новизну и можно использовать современные источники, так как эта тема постоянно разрабатывается и дополняется новой информацией.

Изучая средства для Web-разработки упор, в основном, шел на труды следующих авторов: Бибо Б., Веллинга Л., Каца И., Костарева А., Котерова Д., Ташкова П. А., Томсона Л., Шкрыль А. А., Шлосснейгла Дж., DuBois P., Khalid M. I. Можно выделить самые значимые из них: Котеров Д., Костарев А. «PHP 5», Шлосснейгл Дж. «Профессиональное программирование на PHP», DuBois P. «MySQL», Ташков П. А. «Веб-мастеринг. HTML, CSS, JavaScript, PHP, CMS, AJAX, раскрутка».

Книга Котерова Д., Костарева А. «PHP 5» помогает освоить азы программирования на языке PHP 5. В ней содержатся основы изучения, начиная с базовых понятий и заканчивая разработкой малых проектов. Она послужит хорошей основой для начинающих программистов. Для более глубокого изучения данного языка поможет следующее издание – Шлосснейгл Дж. «Профессиональное программирование на PHP».

В книге Шлосснейгла Дж. очень подробно описан язык программирования PHP 5. Данное издание может помочь в полной мере освоить технически сложные приемы, тонкости и особенности этого языка. Книга, несмотря на указание «для профессионалов», доступна для понимания, обладает множеством примеров и не имеет лишней информации.

Издание DuBoisP. «MySQL» с первых страниц поражает количеством содержащейся в нем информации. Вся она строго структурирована, разбита на главы, в которых без проблем можно найти нужные данные. В книге описаны не только язык SQL, но и основные проблемы его взаимодействия с языками высокого уровня PHP API, Perl DBI, C API. Также можно найти и рекомендации по оптимизации запросов.

По работе Ташкова П.А. «Веб-мастеринг. HTML, CSS, JavaScript, PHP, CMS, AJAX, раскрутка» можно освоить лишь основные принципы языков, но она незаменима, если есть необходимость выбора конкретного языка для более детального изучения. Конечно, за 30 страниц, отведенных в этой книге, например, PHP и AJAX, невозможно досконально изучить их, но этого достаточно, чтобы понять спектр их применения и основные понятия.

### Глава 1. Технология консолидации данных

На основе данных, предоставляемых из гетерогенных источников стало ясно, что необходимо провести анализ технологий, позволяющих интеграцию данных (более подробно о выборе источников в главе 5). Необходимо, чтобы в результате работы этой технологии, данные извлекались из гетерогенных источников, становились надлежащего качества и размещались в базе данных должным образом.

Интеграция данных включает объединение данных, находящихся в различных источниках, и предоставление их пользователю в унифицированном виде. Существует три основных метода интеграции данных: федерализация, распространение и консолидация [1].

*Федерализация данных* – это обеспечение единой виртуальной картины одного или нескольких источников исходных данных. Федерализация позволяет извлекать данные из различных источников, объединять их и представлять пользователю в режиме реального времени. При этом физического перемещения данных не происходит.

Приложения, реализующие *распространение данных*, осуществляют их копирование из одного места в другое. При этом данные не подвергаются никакой обработке и не могут быть доведены до необходимого уровня качества.

Из сказанного выше были сделаны выводы о том, что эти технологии не позволяют добиться нужного результата, поскольку в федерализации нет возможности разместить данные в базу данных (БД), а технология распространения не позволяет улучшить качество данных.

*Консолидация* же позволяет извлекать данные из гетерогенных источников, обеспечивать необходимый уровень их информативности и качества, преобразовывать в единый формат и загружать их в БД [28].

## 1.1. Понятие консолидации

Ценность и достоверность знаний зависит не только от эффективности используемых аналитических методов и алгоритмов, но и от того, насколько правильно подобраны и подготовлены исходные данные для анализа.

В повседневной жизни мы сталкиваемся со следующими ситуациями: во-первых, данные расположены в различных источниках самых разнообразных форматов и типов (отдельные файлы офисных документов, учетные системы, базы данных); во-вторых, данные могут быть либо избыточными, либо недостаточными; и, в-третьих, данные являются «грязными», то есть содержат факторы, мешающие их правильной обработке и анализу (пропуски, шумы, аномальные значения, противоречия и дубликаты). Поэтому, прежде чем приступать к анализу данных, необходимо выполнить ряд следующих процедур, целью которых является доведение данных до приемлемого уровня качества и информативности, организовать их интегрированное хранение в структурах, обеспечивающих их целостность, непротиворечивость, высокую скорость и гибкость выполнения аналитических запросов [33].

*Консолидация* – комплекс методов и процедур, направленных на извлечение данных из различных источников, обеспечение необходимого уровня их информативности и качества, а также преобразование их в единый формат, в котором они могут быть загружены в базу данных (БД) или аналитическую систему.

Цели консолидации:

* доведение данных до соответствующего уровня качества и информативности;
* организация интегрированного хранения данных в структурах, обеспечивающих их целостность, непротиворечивость, высокую скорость и гибкость выполнения аналитических запросов [28].

Начальным этапом реализации любой аналитической задачи или проекта является консолидация данных. В её основе лежит процесс сбора и организации хранения данных в виде, оптимальном с точки зрения их обработки на конкретной аналитической платформе или решения конкретной аналитической задачи. Сопутствующими задачами консолидации является оценка качества данных и их обогащение.

Основные критерии оптимальности с точки зрения консолидации данных:

* обеспечение высокой скорости доступа к данным;
* компактность хранения;
* автоматическая поддержка целостности структуры данных;
* контроль непротиворечивости данных.

## 1.2. Основные задачи консолидации и источники данных

Основные задачи консолидации данных:

* выбор источников данных, определение типа и методики организации доступа к ним;
* разработка стратегии консолидации;
* оценка качества данных;
* обогащение;
* очистка;
* перенос в базу данных [28].

Первым шагом является осуществление выбора источников, содержащие данные, которые могли бы иметь отношение к решаемой задаче. Следующим шагом является определение типа источников и методика организации доступа к ним. В связи с этим можно выделить два основных подхода к организации хранения данных, доступных нам.

1. Данные, хранящиеся в локальных файлах. Источником может быть любой файл (например, TXT, CSV файлы).

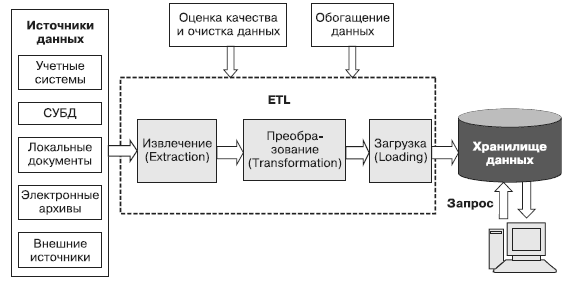
2. БД различных СУБД (систем управления базами данных), таких как Oracle MySQL, Microsoft SQL Server, Microsoft Access, Firebird и других. Поскольку тип и свойства полей жестко заданы при проектировании таблиц, файлы БД лучше поддерживают целостность структуры данных.

Другой важной задачей, которую требуется решить в рамках консолидации, является оценка качества данных с точки зрения их пригодности для обработки с помощью различных аналитических алгоритмов и методов. Если в процессе оценки качества будут выявлены факторы, которые не позволяют корректно применить к данным те или иные методы, необходимо выполнить соответствующую очистку данных.

Еще одной немаловажной операцией, которая может понадобиться при консолидации данных, является обогащение [33].

## 1.3. Обобщенная схема процесса консолидации

Место консолидации в общем процессе анализа данных может быть представлено в виде структурной схемы (рис. 1).

  
Рис. 1. Процесс консолидации данных

Обобщив все вышесказанное, получаем следующее: в основе процедуры консолидации лежит ETL-процесс (Extraction – извлечение, Transformation – преобразование, Loading – загрузка), решающий задачи:

* извлечения данных из гетерогенных источников;
* преобразования данных к виду, пригодному для хранения в определенной структуре;
* загрузки данных в соответствующую БД.

Консолидация позволяет осуществлять трансформацию значительных объемов данных в процессе их передачи от первичных систем к конечным местам хранения [45].

## 1.4. Понятие ETL-процесса

Извлечение данных из разнотипных источников и перенос их в БД с целью дальнейшей аналитической обработки связаны с рядом проблем:

* исходные данные расположены в источниках самых разнообразных типов и форматов, и, кроме того, могут использовать различную кодировку;
* данные в источниках обычно излишне детализированы;
* исходные данные, как правило, являются «грязными», то есть содержат различные факторы, которые мешают их корректному анализу.

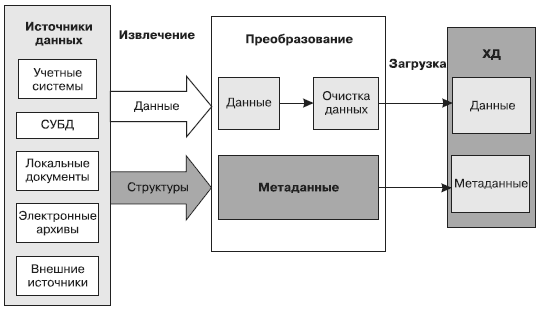
Поэтому для переноса исходных данных из различных источников в БД следует использовать специальный инструментарий, который извлекает данные из источников различного формата, преобразовывает их в единый формат, поддерживаемый БД, а при необходимости – производит очистку данных от факторов, мешающих корректно выполнять их аналитическую обработку, затем загружают преобразованные данные в БД. Такой комплекс программных средств получил обобщенное название ETL. Сам процесс переноса данных и связанные с ним действия называются ETL-процессом, а соответствующие программные средства – ETL-системами [45].

## 1.5. Основные цели и задачи процесса ETL

ETL-система должна обеспечивать выполнение трех основных этапов процесса переноса данных (ETL-процесса):

* извлечение данных. На этом этапе происходит излечение данных из гетерогенных источников с последующей подготовкой к преобразованию;
* преобразование данных. Производятся преобразование форматов и кодировки данных, а также их обобщение и очистка;
* загрузка данных – запись преобразованных данных в соответствующую систему хранения, например, базу данных.

Перемещение данных в процессе ETL можно разбить на последовательность процедур, представленных следующей функциональной схемой (рис. 2).

  
Рис. 2. Обобщенная структура процесса ETL

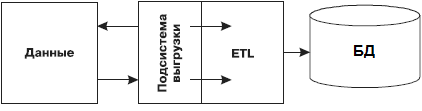
1. Извлечение. Данные извлекаются из источников и загружаются в промежуточную область.
2. Поиск ошибок. Производится проверка данных на соответствие спецификациям и возможность последующей загрузки в БД.
3. Преобразование. Данные группируются и преобразуются к виду, соответствующему структуре БД.
4. Распределение. Данные распределяются на несколько потоков в соответствии с особенностями организации процесса их загрузки в БД.
5. Вставка. Данные загружаются в БД-получатель.

Таким образом, ETL следует рассматривать не только как процесс переноса данных из одного приложения в другое, но и как инструмент их подготовки к анализу [33].

## 1.6. Извлечение данных в ETL

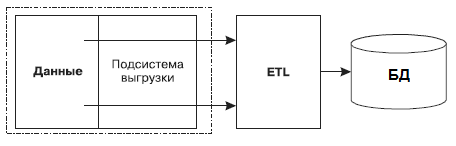
Процедура извлечения записей из источника данных и подготовка содержащейся в них информации к процессу преобразования является начальным этапом ETL. Процедуру извлечения можно реализовать двумя основными способами.

1. Извлечение данных с помощью специализированных программных средств (рис. 3).

  
Рис. 3. Первый способ извлечения данных в ETL

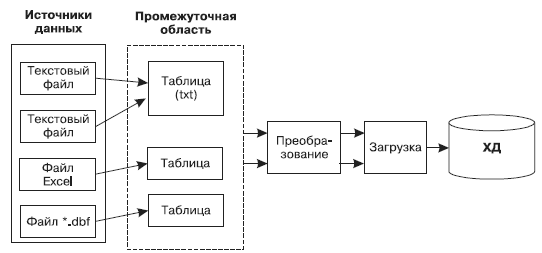
Преимущества данного способа заключаются в том, что использование специализированных программ позволяет, во-первых, избежать необходимости оснащать разработанные системы средствами выгрузки, во-вторых, учитывать особенности всего ETL-процесса уже в процессе выгрузки. В случае, когда данные извлекаются из локальных источников, альтернативы использованию специальных средств нет, поскольку такие виды источников данных не содержат средств выгрузки данных.

2. Извлечение данных средствами той системы, в которой они хранятся (рис. 4).

  
Рис. 4. Второй способ извлечения данных в ETL

Поскольку средства «самовыгрузки» системы разрабатываются с учетом особенностей структуры данных, это позволяет адаптировать процедуру извлечения к структуре извлекаемых данных, что в ряде случаев делает процесс более эффективным [33].

После извлечения данные помещаются в так называемую промежуточную область, где для каждого источника создается свой отдельный файл. В некоторых случаях, когда требуется выгрузить данные из нескольких источников одного типа, для них создается общая таблица; одно из ее полей указывает на источник, из которого были взяты данные (рис. 5).

  
Рис. 5. Схема организации выгрузки в ETL-процессе

## 1.7. Преобразование данных в ETL

Следующий этап ETL-процесса – преобразование данных. Его целью является подготовка данных к размещению в БД и приведение их к виду, наиболее удобному для последующего анализа. При этом должны учитываться некоторые требования, в частности, к уровню качества данных.

В процессе преобразования данных в рамках ETL выполняются следующие операции: преобразование структуры данных; агрегирование данных; перевод значений; создание новых данных; очистка данных.

Рассмотрим подробнее каждую из этих операций:

1. *Преобразование структуры данных*. Так как данные интегрируются из различных источников, то они могут отличаться своей структурной организацией (например, форматами, типами, кодировкой и так далее).

2. *Агрегирование данных*. Общим свойством всех этих источников является то, что они содержат данные с максимальной степенью детализации. Однако, для достоверного описания предметной области использование данных с максимальным уровнем детализации не всегда целесообразно. В результате агрегирования большое количество записей о каждом событии заменяется относительно небольшим количеством записей, содержащих агрегированные значения.

3. *Перевод значений*. Данные часто хранятся с использованием специальных кодировок, которые позволяют сократить избыточность данных и тем самым уменьшить объем памяти, требуемой для их хранения, то есть привести их в сокращенный вид. В этом случае перед загрузкой данных в БД требуется выполнить перевод таких сокращенных значений в более полные и, соответственно, понятные.

4. *Создание новых данных*. В процессе загрузки в БД может понадобиться вычисление некоторых новых данных на основе существующих. Создание новой информации на основе имеющихся данных тесно связано с процессом обогащения данных, который может производиться на этапе преобразования данных в ETL. Агрегирование также может рассматриваться как создание новых данных.

5. *Очистка данных*. Сбор данных производится из огромного числа источников и некоторые из них не содержат автоматических средств поддержки целостности, непротиворечивости и корректного представления данных. При переносе информации в БД можно столкнуться с потоками «грязных» данных, которые могут стать причиной неправильных результатов анализа. Поэтому в процессе ETL применяется очистка – процедура корректировки данных, которые не удовлетворяют определенным критериям качества (содержат пропуски, неправильные форматы, дубликаты и т. д.) [45].

## 1.8. Загрузка данных в базу данных

После того как данные извлечены из гетерогенных источников и выполнено преобразование, осуществляется последний этап ETL – загрузка данных в БД. Процесс загрузки заключается в переносе данных из промежуточных файлов в структуру БД.

Один из простейших вариантов загрузки данных – это формулирование интерактивных SQL-запросов к базе данных с использованием INSERT и UPDATE выражений.

Использование запросов SQL для загрузки в базу данных может быть использовано разными СУБД, так как синтаксис INSERT запросов, не использующих специфических функций, совпадает. Для последующего использования запросы удобно хранить в файлах, которые в свою очередь размещаются в системах контроля версий, так как все модификации сохраняются в виде истории и возможна одновременная работа нескольких пользователей над одним и тем же файлом.

Таким образом, загрузка данных может происходить, как в не заполненную базу данных, составляя первичное наполнение, так и дополняя данные в уже заполненной.

Загрузка огромного массива данных в БД является узким местом во всём процессе перемещения данных по сравнению с остальными этапами. Поэтому важно учитывать производительность процесса [16].

После завершения загрузки выполняются постзагрузочные операции, то есть операции над данными, только что загруженными в БД, перед тем как сделать их доступными для пользователя. К ним относится верификация данных. Прежде чем использовать новые данные для анализа, полезно убедиться в их надежности и достоверности. Для этих целей можно предусмотреть комплекс верификационных тестов.

Если тестирование показало, что несоответствия, позволяющие заподозрить потерю или недостоверность данных, отсутствуют, то можно считать загрузку данных в БД успешной и приступать к анализу новой информации [33].

### Глава 2. Нестационарная модель конвективного облака

Одним из наиболее эффективных инструментов изучения конвективных облаков является численное моделирование. Существует множество моделей развития конвективных облаков, отличающихся как размерностью, так и степенью детализации описания микрофизических процессов [7].

*Конвекция*– упорядоченный воздухообмен между верхними и нижними слоями тропосферы.

*Конвективное облако*– совокупность жидких капель и (или) ледяных частиц, образующаяся в системе вертикальных воздушных потоков (одна или несколько конвективных ячеек).

*Конвективная ячейка* – замкнутая система воздушных движений в вертикальной и горизонтальной плоскостях.

Причиной образования конвективного облака является конденсация водяного пара в поднимающемся и охлаждающемся воздухе. Выделение теплоты при конденсации является одним из механизмов, ответственных за развитие конвективного облака (чем теплее объем воздуха по отношению к окружающей среде, тем интенсивнее он всплывает вверх, что способствует усилению восходящего потока, продолжению конденсации и т.д.).

Часто верхняя граница конвективного облака пересекает изотерму 0°C и заходит в область отрицательных температур. При этом может происходить замерзание капель воды, а также сублимация (переход вещества из твёрдого состояния в газообразное без пребывания в жидком состоянии) водяного пара. Это способствует дополнительному выделению тепла и усилению восходящего потока.

По степени развития конвективные облака делятся на четыре категории: *плоские кучевые*, они же – кучевые облака хорошей погоды; *средние кучевые, мощные кучевые* и *кучево-дождевые*. С последними обычно бывают связаны наиболее интенсивные осадки и подавляющее большинство опасных атмосферных явлений (грозы, шквалы, грозы, смерчи). Этим они представляют большую опасность для летательных аппаратов. «Поражающими факторами», помимо вышеперечисленных опасных явлений, являются вертикальные воздушные потоки и обледенение летательных аппаратов (это уже относится не только к кучево-дождевым).

Четкие количественные различия между разновидностями конвективных облаков не определены до сих пор. Характерные значения вертикальной мощности: для плоских кучевых – менее 1 км, для средних кучевых – 1 - 2 км; для мощных кучевых – 2 - 4 км; для кучево-дождевых – свыше 4 км (рекордные значения, наблюдаемые исключительно в тропических широтах, превышают 20 км). Эти критерии условны. Для выявления кучево-дождевых существует еще и качественный критерий: наличие осадков (пусть даже не достигающих Земли). Конвективное облако, содержащее осадки, в любом случае считается кучево-дождевым.

## 2.1. Классификация моделей конвективных облаков

По размерности пространства различают одномерные, полуторамерные, двухмерные и трехмерные модели.

В одномерных моделях пространство (обычно в форме цилиндра) разбивается на слои по высоте. Значения параметров (таких как температура, скорость потока, влажность и др.) усредняются в каждом слое. Взаимодействие происходит только в вертикальном направлении, скорость потока имеет только вертикальную составляющую.

В отличие от одномерных в полуторамерных моделях дополнительно учитывается внешняя среда и радиальная составляющая скорости (в горизонтальном направлении, направлена внутрь цилиндра или во внешнюю среду). Помимо взаимодействия в вертикальном направлении, происходит взаимодействие и в горизонтальном направлении между цилиндром и окружающей средой. Кроме разбиения по высоте пространство может разбиваться на 2 концентрических цилиндра с разными диаметрами. В этом случае внутренний цилиндр соответствует области с восходящим потоком (в этом цилиндре и формируется облако), внешний – области с компенсирующим нисходящим движением. Значения параметров усредняются в каждом слое для каждого из цилиндров. Происходит взаимодействие в горизонтальном направлении между внутренним и внешним цилиндром.

В *двухмерных* моделях вертикальное сечение пространства делится на прямоугольные области. Взаимодействие по направлениям внутри этого сечения.

В *трехмерных* моделях пространство обычно разбивается на области в форме прямоугольного параллепипеда. Взаимодействие по всем направлениям.

В моделях с жидкой фазой предполагается, что вода существует в виде пара (газообразное состояние) или в виде капелек (жидкое состояние), а ледяные частицы (твердое состояние) не учитываются.

Модели со смешанной фазой учитывают воду в виде пара, капелек и ледяных частиц.

В моделях с *параметрической* микрофизикой спектры частиц не учитываются. Предполагается, что существуют облачные капли (маленькие), которые не движутся относительно потока, дождевые капли (крупные) и градины, которые падают относительно потока. Некоторые модели не учитывают градины, а иногда даже и дождевые капели, т.е. совсем не моделируют образование осадков. Переходы между паром, облачными каплями и дождевыми каплями (т.е. процессы конденсации, испарения, нуклеации, коагуляции и др.) задаются параметрически и никак не учитывают реальные размеры капель. Скорость падения дождевых капель рассчитывается исходя из их концентрации.

Модели с *подробной (спектральной)* микрофизикой учитывают спектры частиц. Обычно в таких моделях присутствуют водяные капли и ледяные частицы нескольких типов. Частицы разных размеров имеют разные устоявшиеся скорости падения относительно потока. Микрофизические процессы описываются более подробно и учитывают размеры, а иногда и форму частиц. Эти модели наиболее полно отражают природные процессы и позволяют проследить за эволюцией спектров.

## 2.2. Описание полуторамерной нестационарной модели конвективного облака

Для целей оперативного прогноза, например, в метеорологических центрах аэропортов, где для расчетов могут использоваться не суперкомпьютеры, а обычные персональные компьютеры, необходимы простые модели малой размерности. Они не требуют больших вычислительных ресурсов, однако способны воспроизводить те характеристики облака (значение высоты верхней и нижней границы, максимальные значения водности, перегрева, скорости восходящего потока и т.п.), которые могут стать предикторами для существующих методов прогноза опасных конвективных явлений, практически в режиме реального времени. Наилучшими, с этой точки зрения, являются полуторамерные модели, опыт применения которых для расчетов характеристик реальных облаков уже дает положительные результаты [30].

В реальных облаках присутствуют твердая фаза, а именно, различные ледяные кристаллы. Для большей точности нужна модель, учитывающая это – модель со смешанной фазой. Кроме капелек воды, в данной модели присутствуют ледяные частицы различных типов: 3 типа кристаллов, снежинки (сцепленные кристаллы), крупа, градины.

Информация о спектрах капель, ледяных частиц каждого типа, частиц аэрозоли хранится в каждом узле сетки. Частицы разных типов имеют разные устоявшиеся скорости падения относительно потока.

Нами была использована модель, разработанная Н. О. Раба и Е. Н. Станковой [3, 7, 10-13, 37].

Модель состоит из динамического и микрофизического блоков. Динамический блок модели представляет собой систему дифференциальных уравнений в частных производных, которые описывают эволюцию во времени и пространстве значений вертикальной скорости, температуры, отношения смеси водяного пара, капель воды и ледяных кристаллов различной формы.

Микрофизический блок состоит из системы кинетических уравнений, описывающих изменение спектров капель и кристаллов под действием процессов нуклеации (образование водяных капель на поверхности аэрозольных частиц), замерзание капелек, таяние ледяных частиц, конденсация на поверхности частиц, испарение с поверхности частиц, коагуляция. Тип частиц, образующихся при нуклеации, зависит от температуры. Тип частиц, образующихся при замерзании капелек, зависит от размеров капелек. При столкновении и слиянии двух частиц друг с другом образуется частица, тип которой зависит от типов столкнувшихся частиц, их масс, температуры.

### Глава 3. Проектирование реляционных баз данных

## 3.1. Основы реляционной базы данных

Базой данных (БД) называется организованная в соответствии с определенными правилами и находящаяся в памяти компьютера совокупность сведений об объектах, процессах, событиях или явлениях, относящихся к некоторой предметной области. Она организована таким образом, чтобы обеспечить информационные потребности пользователей, а также удобное хранение этой совокупности данных [31].

Реляционная база данных представляет собой множество взаимосвязанных таблиц, каждая из которых содержит информацию об объектах определенного вида. Каждая строка таблицы содержит данные об одном объекте – кортеж, а столбцы таблицы содержат различные характеристики этих объектов – атрибуты.

Реляционная база данных обладает рядом достоинств:

* отображает информацию в простой для пользователя форме;
* основана на развитом математическом аппарате, который позволяет достаточно лаконично описать основные операции над данными;
* позволяет создавать языки манипулирования данными непроцедурного типа;
* манипулирование данными на уровне выходной БД и возможность их изменения.

Реляционные системы базируются на формальных основах – теории, которая называется реляционной моделью данных. Такая система включает в себя три основных принципиальных аспекта данных [14]:

1. структурный аспект. Данные в базе воспринимаются как набор отношений. Отношения следует понимать как упорядоченную пару, состоящую из заголовка (атрибутов) и тела (кортежей);

2. аспект целостности. Обеспечение целостности заключается в контроле правильности, согласованности данных в базе (при добавлении, обновлении, удалении), что осуществляется с помощью ограничений целостности. Так называется логическое выражение, связанное с некоторой базой данных, принимающее значение «истина» (т. е. является истинным высказыванием о предметной области) [19];

3. аспект обработки. В распоряжении пользователя имеются операторы манипулирования «таблицами» (реляционная алгебра, реляционное исчисление), которые генерируют новые «таблицы» на основании уже имеющихся.

Для реляционной модели существует несколько формальных моделей манипулирования данными (на базе которых реализуются реляционные и близкие к ним языки запросов).

1. Реляционные алгебры. Неформально можно описать как (высокоуровневый) процедурный язык, с помощью которого можно сообщить СУБД, как построить новое отношение из одного или нескольких существующих отношений. Существует по меньшей мере две реляционные алгебры:
   * *реляционная алгебра Кодда* – первоначально предложена Э. Коддом в 1970-ые годы и включает 8 операций. В большей степени алгебра базируется на теории множеств. Базовыми операциями являются переименование атрибутов, объединение, пересечение, взятие разности, декартово произведение, проекция и ограничение. Операция соединения общего вида, хотя и включается в алгебру, является вторичной и явно представляется через другие операции [2];
   * *реляционная алгебра A* – предложена Кристофером Дейтом и Хью Дарвеном [14, 35]. В качестве базиса используются реляционные аналоги логических операций (реляционные отрицание, конъюнкция, дизъюнкция, удаление атрибута) на основе обычных теоретико-множественных операций и позволяют выражать напрямую операции пересечения, декартова произведения, естественного соединения, объединения отношений и т. д. Алгебра *A* позволяет лучше осознать логические основы реляционной модели, хотя является в меньшей степени ориентированной на практическое применение, чем алгебра Кодда, но зато красива и элегантна.
2. Реляционное исчисление. Неформально представляет собой непроцедурный язык, который можно использовать для определения того, каким будет некоторое отношение, созданное на основе одного или нескольких существующих отношений [14].

Взаимодействие с базой данных происходит при помощи Системы Управления Базой Данных (СУБД), которая расшифровывает запросы и производит операции с информацией в базе данных.

Для выполнения операций с данными применяется механизм запросов. Запросы к базе формируются на специально созданном для этого языке – SQL (*Structured Query Language –* язык структурированных запросов) [31].

Разработка структуры БД – важнейшая задача, решаемая при проектировании БД. Структура БД – это одно из основных проектных решений при создании приложений с использованием БД.

## 3.2. Этапы проектирования реляционной базы данных

Основная причина сложности проектирования базы данных заключается в том, что объекты реального мира и взаимосвязи между ними вовсе не обязаны иметь и, как правило, не имеют структуры, согласованной с реляционной моделью данных. При проектировании мы должны «придумать» представление для реальных объектов и их связей в терминах таблиц, полей, атрибутов, записей и т. п., то есть в терминах абстракций реляционной модели данных. Поэтому разработка эффективной базы данных состоит из нескольких этапов:

* концептуальное (инфологическое) проектирование – сбор, анализ и редактирование требований к данным;
* логическое (даталогическое) проектирование – преобразование требований к данным в структуры данных;
* физическое проектирование – преобразование логической структуры БД в физическую с учетом аспектов производительности СУБД [15].

Также для базы данных необходима процедура нормализации. Нормализацией схемы БД называется процедура, производимая над базой данных с целью удаления в ней избыточности.

*Первая нормальная форма (1НФ)* требует следующие правила: каждый столбец в строке должен содержать единственное значение; каждая строка в таблице обязана содержать одинаковое количество столбцов.

*Вторая нормальная форма (2НФ)* говорит о том, что: таблица обязана соответствовать первой нормальной форме; все столбцы, не входящие в полный первичный ключ, должны зависеть от полного первичного ключа.

*Третья нормальная форма (3НФ)* расширяет две предыдущие, неся в себе два правила: таблица должна соответствовать второй нормальной форме; все столбцы, не входящие в полный первичный ключ, должны зависеть от него и не должны зависеть друг от друга [14, 36].

## 3.3. Организация реляционной базы данных

## для специализированной метеорологической информации

Основой модели является реляционная база данных, которая содержит информацию о времени, местоположении и типе опасного конвективного явления в сочетании со всеми наборами метеорологических данных о состоянии атмосферы и поверхности Земли.

Концептуальный этап проектирования состоит в том, что сначала формируются сущности (объекты), а далее непосредственно к этим сущностям формируются атрибуты.

**Формулирование сущностей**

База данных состоит из двух частей:

1. Блок «данные»:

* «Города» – список городов для прогнозирования.
* «Временные периоды» – временной период для прогнозирования.
* «Метеорологические станции» – каталог метеорологических станций.
* «Погодные явления» – каталог погодных явлений.
* «Параметры» – метеоданные, используемые в качестве входных параметров для математических моделей конвективных облаков.
* «Метеорологические станции и параметры» (вспомогательная) – соответствие между метеорологическими станциями и параметрами.
* «Временные периоды и метеорологические станции» (вспомогательная) – соответствие между временными периодами и каталогом метеорологических станций.

2. Блок «управление данными»:

* «Пользователи» – каталог лиц, взаимодействующих с БД.
* «Пользовательские эксперименты» – каталог, в котором хранятся графики результатов численных экспериментов, проведенных пользователями.

Для удобства использования и хранения данных, БД была разделена на два больших блока: блок «данные» и блок «управление данными».

Каждая из этих сущностей обладает набором атрибутов (свойств), которые являются важными при разработке схемы и базы данных.

**Формулирование атрибутов сущностей**

Далее модель развивается путём определения атрибутов для каждого объекта. Они приводятся на логической схеме (рис. 7).

Перейдем к логическому проектированию. Общим способом представления логической модели базы данных является построение ER-диаграмм (*Entity-Relationship* – сущность-связь). В этой модели сущность определяется как дискретный объект, для которого сохраняются элементы данных, а связь описывает отношение между двумя объектами.

Реляционная модель характеризуется использованием ключей и отношений. *Первичный ключ* уникально идентифицирует строку в отношении, и каждое отношение может иметь только один первичный ключ, даже если больше чем один атрибут является уникальным. В некоторых случаях требуется более одного атрибута для идентификации строк в отношении. Совокупность этих атрибутов называется *составным ключом*. Другой тип ключа, называемый *внешним ключом*, существует только в терминах схемы данных между двумя отношениями. Внешний ключ в отношении – это атрибут, который является первичным ключом (или его частью) в другом отношении. Это – распределенный атрибут, который формирует схему данных между двумя отношениями в БД [31]. В нашей базе данных будут использоваться все три вида ключей.

На этапе физического моделирования БД описаны типы данных для каждого вида хранимой информации, а также способы и место их физического размещения. При этом для каждого поля таблицы определены типы данных, наиболее подходящие для хранения соответствующей информации. Такие поля не могут содержать пустые значения (NULL).

Схему реляционной базы данных изобразим в виде таблиц и связей между ними. При этом «таблицы» будут являться реализацией сущностей, а поля таблицы – свойствами сущностей. Следует отметить, что разработанная логическая модель БД находится в 3НФ, так как нет транзитивных зависимостей [18].

Разработанная схема данных содержит 9 отношений и может быть реализована при помощи SQL. Окончательная ER-модель данных представлена на (рис. 7).

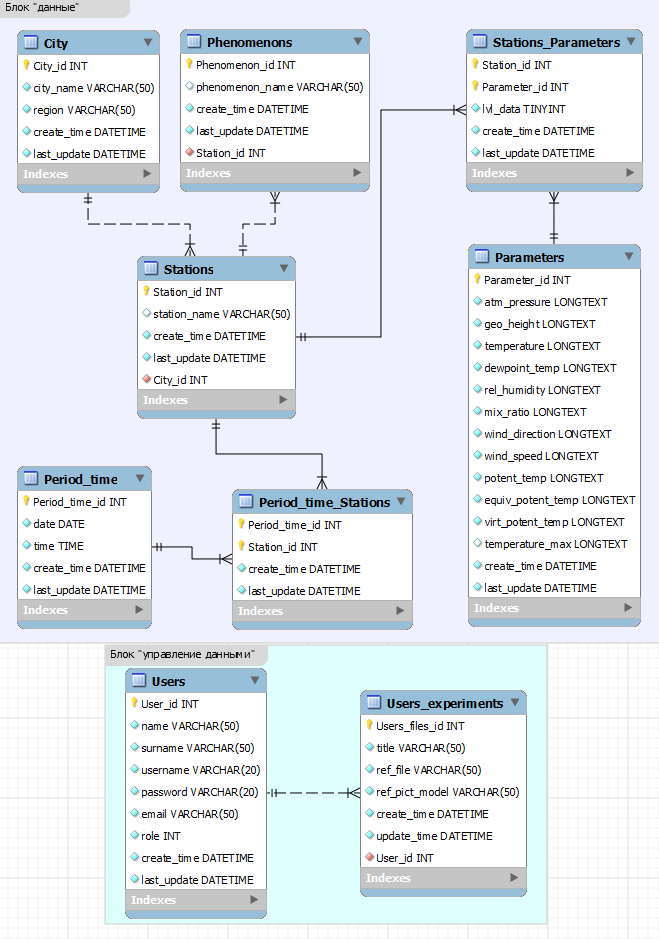


Рис. 7. ER-модель данных (Null обозначается белым ромбом, Not Null – синим)

### Глава 4. Проектирование многомерных баз данных

Реляционная модель данных, которая была предложена Э. Ф. Коддом в 1970 году служит основой современной многомиллиардной отрасли баз данных. За последнее время сложилась многомерная модель данных, которая используется, когда целью является именно сложный анализ данных, а не выполнение транзакций. Технология многомерных баз данных – ключевой фактор интерактивного анализа больших массивов данных с целью поддержки принятия решений. Подобные базы данных трактуют данные как многомерные кубы, что очень удобно именно для их анализа.

В процессе анализа данных, поиска решений часто возникает необходимость в построении зависимостей между различными параметрами. Кроме того, число таких параметров может варьироваться в широких пределах. Традиционные средства анализа, оперирующие данными, которые представлены в виде таблиц реляционной БД, не могут в полной мере удовлетворять таким требованиям. Э. Ф. Кодд рассмотрел ее недостатки, указав в первую очередь на невозможность «объединять, просматривать и анализировать данные с точки зрения множественности измерений, т.е. самым понятным для аналитиком способом» [26].

В 1993 году основоположник реляционного подхода Э. Ф. Кодд, стоявший у истоков технологии OLAP, опубликовал статью под названием «Providing OLAP to User-Analysts: An IT Mandate». В ней изложены основные концепции оперативной аналитической обработки и определены 18 требований, которым должны удовлетворять продукты, позволяющие выполнять оперативную аналитическую обработку [56].

Набор этих требований и послужил де-факто определением OLAP. Кроме того, Кодд разбил эти правила на следующие четыре группы, назвав их *особенностями*. Эти группы получили названия: основные особенности, специальные особенности, особенности представления отчетов и управление измерениями [26].

Более известен для определения OLAP тест FASMI (Fast Analysis of Shared Multidimensional Information), созданный в 1995 году Найджелом Пендсом и Ричардом Критом, который является альтернативой правилам Кодда. Акцент сделан на скорость обработки, многопользовательский доступ, релевантность информации, наличие средств статистического анализа и многомерность, т.е. представление анализируемых фактов как функций от большого числа их характеризующих параметров [40].

Многомерные модели данных имеют три важных области применения, связанных с проблематикой анализа данных:

* хранилища данных интегрируют для анализа информации из нескольких источников на предприятии;
* системы оперативной аналитической обработки (online analytical processing – OLAP) позволяют оперативно получить ответы на запросы, охватывающие большие объемы данных в поисках общих тенденций;
* приложения добычи данных служат для выявления знаний за счет полуавтоматического поиска ранее неизвестных шаблонов и связей в базах данных [52].

Перечислим основные преимущества многомерного подхода:

* представление данных в виде многомерных кубов более наглядно, чем совокупность нормализованных таблиц реляционной модели;
* возможности построения аналитических запросов к системе, использующей МХД, более широки;
* использование многомерной модели позволяет значительно уменьшить продолжительность поиска в МХД, обеспечивая выполнение аналитических запросов практически в режиме реального времени. Это связано с тем, что агрегированные данные вычисляются предварительно и хранятся в многомерных кубах вместе с детализированными [33].

Таким образом, основное назначение многомерных хранилищ данных (МДХ) – поддержка систем, ориентированных на аналитическую обработку данных, поскольку такие хранилища лучше справляются с выполнением сложных нерегламентированных запросов [26].

## 4.1. Понятие многомерной базы данных

Для того чтобы перейти к понятию многомерной базы данных необходимо ввести термин OLAP, который неразрывно связан с термином «хранилище данных».

Хранилище данных – это предметно-ориентированное, привязанное ко времени и неизменяемое собрание данных для поддержки процесса принятия управляющих решений.

Данные в хранилище попадают из оперативных систем (OLTP-систем), которые предназначены для автоматизации различных процессов. Кроме того, хранилище может пополняться за счет внешних источников, например, статистических отчетов.

Задача хранилища – предоставить «сырье» для анализа в одном месте и в простой, понятной структуре [40].

Многомерная модель данных, лежащая в основе построения многомерных хранилищ данных, опирается на концепцию многомерных кубов, или гиперкубов. Они представляют собой упорядоченные многомерные массивы, которые также часто называют OLAP-кубами (On-Line Analytical Processing – оперативная аналитическая обработка).

Технология комплексного многомерного анализа данных получила название OLAP. Она представляет собой методику оперативного извлечения нужной информации из больших массивов данных и формирования соответствующих отчетов [33].

Сегодня термин OLAP – это понятие для различных технологий, включая системы поддержки принятия решений, Business Intelligence и управленческие информационные системы.

Таким образом, многомерная база данных – это решение, которое предлагает не только высокую производительность и простоту использования, но и обеспечивает возможности, необходимые для разработки, расширения и быстрого развертывания приложений при сокращении ИТ-затрат [46].

## 4.2. Основы многомерного представления данных

В основе многомерного представления данных лежит их разделение на две группы – факты с соответствующими численными параметрами и использование текстовых измерений для предоставления максимально возможного контекста для фактов [17].

*Измерения* – это категориальные атрибуты, наименования и свойства объектов, участвующих в некотором процессе. Они могут быть числовыми, но в любом случае это данные дискретные, т.е. их значения принадлежат ограниченному набору. Измерения качественно описывают исследуемый процесс.

Следуя Коду, можно сказать, что многомерное концептуальное представление – это множественная перспектива, состоящая из нескольких независимых измерений, вдоль которых могут быть проанализированы определенные совокупности данных. Одномерный анализ по нескольким измерениям определяется как *многомерный анализ*.

Каждое измерение может быть представлено в виде иерархической структуры. Более того, некоторые измерения могут иметь несколько видов иерархического представления. На пересечениях осей измерений располагаются данные – факты.

*Факты (меры)* – это данные, количественно описывающие процесс, непрерывные по своему характеру, т.е. они могут принимать бесконечное множество значений [26, 33]. Каждый факт обладает некоторой гранулярностью, определенной уровнями, из которых создается их комбинация значений измерений.

Хранилища данных могут содержать следующие три типа фактов: события; мгновенные снимки; совокупные мгновенные.

В многомерной базе данных *параметры* представляют свойства факта, который пользователь хочет изучить. Параметры состоят из двух компонентов: численная характеристика факта и формула, обычно простая агрегативная функция. Свойство и формула выбираются таким образом, чтобы представлять осмысленную величину для всех комбинаций уровней агрегирования [52].

*Таблица фактов* – является основной таблицей хранилища данных. Как правило, она содержит сведения об объектах или событиях, совокупность которых будет в дальнейшем анализироваться [25].

Таблица фактов содержит уникальный составной ключ, объединяющий первичные ключи таблиц измерений. Помимо этого, таблица фактов содержит одно или несколько числовых полей, на основании которых в дальнейшем будут получены агрегатные данные.

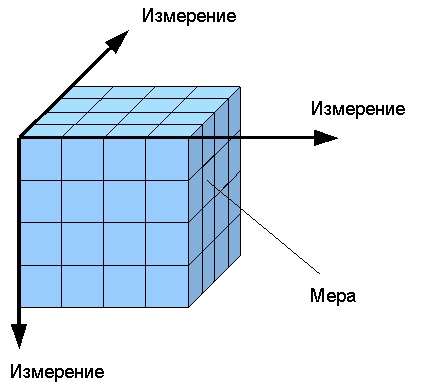
Для многомерного анализа пригодны таблицы фактов, содержащие как можно более подробные данные (т.е. соответствующие членам нижних уровней иерархии соответствующих измерений).

*Таблицы измерений* – это таблицы, содержащие неизменяемые либо редко изменяемые данные. Они также содержат как минимум одно описательное поле и целочисленное ключевое поле (суррогатный ключ) для однозначной идентификации члена измерения. Каждая таблица измерений должна находиться в отношении «один ко многим» с таблицей фактов [25].

Также необходимо отметить, что скорость роста таблиц измерений должна быть незначительной по сравнению со скоростью роста таблицы фактов [39].

## 4.3. Структура многомерного куба

Многомерную модель данных можно представить как гиперкуб (рис. 8) и рассматривать как систему координат, осями которой являются измерения. По осям будут откладываться значения измерений. В такой системе каждому набору значений измерений будет соответствовать ячейка, в которой можно разместить *меры* (числовые показатели), связанные с данным набором. Следовательно, между объектами процесса и их числовыми характеристиками будет установлена однозначная связь.

  
Рис. 8. Представление данных в виде гиперкуба

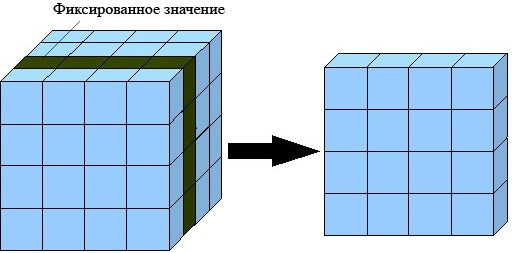
Таким образом, информация в многомерном хранилище данных является логически целостной.

Необходимо отметить, что использование многомерной модели данных сопряжено с определенными трудностями. Для ее реализации требуется большой объем памяти. Это связано с тем, что при реализации физической многомерности используется большое количество технической информации, поэтому объем данных, который может поддерживаться МХД, обычно не превышает нескольких десятков гигабайт, а также она труднее поддается модификации. На основании этого можно сделать вывод, что применение многомерного представления данных целесообразно только в тех случаях, когда объём используемых данных невелик, а сама многомерная модель имеет стабильный набор измерений [26, 33].

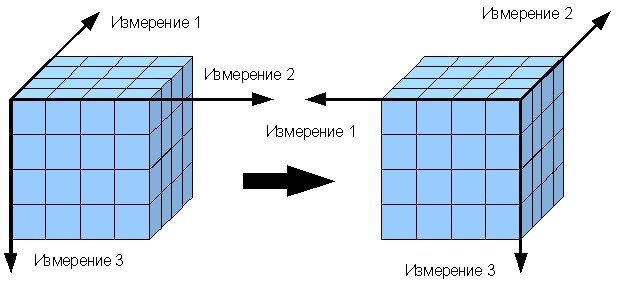
## 4.4. Основные операции над измерениями гиперкуба

В процессе поиска и извлечения из гиперкуба нужной информации над его измерениями производится ряд действий, наиболее типичными из которых являются: сечение; транспонирование; консолидация; детализация.

Операция *сечение (срез)* заключается в формировании подмножества многомерного массива данных, соответствующее единственному значению одного или нескольких элементов измерений, не входящих в это подмножество (рис. 9).

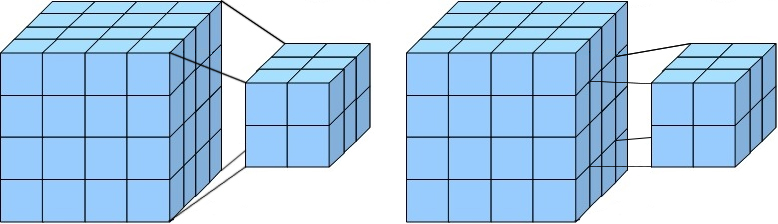
  
Рис. 9. Операция сечения

Операция *вращение (транспонирование)* – изменение расположения измерений, представленных в отчете или на отображаемой странице. Кроме того, вращением куба данных является перемещение внетабличных измерений на место измерений, представленных на отображаемой странице, и наоборот (рис. 10).

  
Рис. 10. Операция транспонирования

Операции *консолидация (свертка)* и *детализация (декомпозиция)* – это операции, которые определяют переход вверх по направлению от детального представления данных к агрегированному и наоборот, соответственно (рис. 11) [26, 33].

Рис. 11. Операции консолидации и детализации



## 4.5. Архитектура OLAP-систем

Полномасштабная OLAP-система должна выполнять сложные и разнообразные функции, включающие сбор данных из гетерогенных источников, их согласование, преобразование и загрузку в хранилище, хранение аналитической информации, регламентную отчетность, поддержку нерегламентируемых сложных запросов и многомерный анализ.

В общем виде архитектура корпоративной OLAP-системы описывается схемой с тремя выделенными уровнями (рис. 12): извлечение, преобразование и загрузка данных; хранение данных; анализ данных.

  
Рис. 12. Архитектура OLAP-системы

Данные поступают из различных транзакционных систем (OLTP-систем), от подчиненных структур, от внешних организаций в соответствии с установленным регламентом. Вся эта информация проверяется, согласуется, преобразуется и помещается в хранилище и витрины данных. После этого пользователи с помощью специализированных инструментальных средств получают необходимую им информацию для построения различных табличных и графических представлений, прогнозирования, моделирования и выполнения других аналитических задач.

Рассмотрим подробно каждый слой архитектуры OLAP-системы.

*Уровень извлечения, преобразования и загрузки данных*. Данный слой связан с понятием ETL-процесса, который был изложен в первой главе.

С организационной точки зрения, данный слой включает подразделения и структуры организации всех уровней, поддерживающие базы данных оперативного доступа. Он представляет собой низовой уровень генерации информации, уровень внутренних и внешних информационных источников, вырабатывающих «сырую» информацию.

Из источников данных информация перемещается на основе некоторого регламента в централизованное хранилище.

Кроме того, несмотря на различную функциональную направленность, исходные транзакционные системы часто «пересекаются» по данным, т.е. их локальные базы данных содержат однотипную по смыслу информацию. Перед загрузкой в хранилище она должна быть согласована, чтобы обеспечить целостность и непротиворечивость аналитических данных.

Таким образом, загрузка данных из источников в хранилище осуществляется специальными процедурами, позволяющими извлекать данные из разнородных БД, текстовых файлов, выполнять различные типы согласования и «очистки» данных, преобразовывать данные при перемещении их от источников к хранилищу, загружать согласованные и «очищенные» данные в структуры хранилища.

*Уровень хранения данных*. Рассматриваемый слой предназначен непосредственно для хранения значимой, проверенной, согласованной, непротиворечивой и хронологически целостной информации, которую с достаточно высокой степенью уверенности можно считать достоверной.

Хранилище данных не ориентировано на решение какой-либо определенной функциональной аналитической задачи. Его целью является обеспечение целостности и поддержки хронологии всевозможных данных, и с этой точки зрения оно нейтрально по отношению к приложениям.

Хранилище данных должно поддерживать эффективную работу с терабайтными объемами информации, иметь развитые средства ограничения доступа, обеспечивать повышенный уровень надежности и безопасности, соответствовать необходимым требованиям по восстановлению и архивации.

*Уровень анализа данных*. Для организации доступа аналитиков к данным ХД и ВД используются специализированные рабочие места, поддерживающие необходимые технологии как оперативного, так и долговременного анализа.

Аналитическая деятельность достаточно разнообразна и определяется характером решаемых задач, организационными особенностями соответствующих учреждений, уровнем и степенью подготовленности аналитиков.

В связи с этим современный подход к инструментальным средствам анализа не ограничивается использованием какой-то одной технологи. В настоящее время принято различать следующие основные виды аналитической деятельности: стандартная отчетность; нерегламентированные запросы; многомерный анализ (OLAP); интеллектуальный анализ данных (data mining).

Каждая из этих технологий имеет свои особенности, определенный набор типовых задач и должна поддерживаться специализированной инструментальной средой [39].

## 4.6. Технические аспекты многомерного хранения данных

OLAP-серверы скрывают от конечного пользователя способ реализации многомерной модели. Они формируют гиперкуб, с которым пользователи посредством OLAP-клиента выполняют необходимые манипуляции, анализируя данные. Однако способ реализации важен, поскольку от него зависят производительность решения и требуемые ресурсы.

Существует три основных способа реализации многомерной модели –MOLAP, ROLAP, HOLAP. Выбор был сделан в пользу MOLAP, поскольку он имеет необходимые в данной работе преимущества:

* общая простота системы, что позволяет осуществлять быстрое встраивание технологий многомерных СУБД в приложения;
* относительно низкая общая стоимость владения, а также быстрый возврат инвестиций;
* поиск и выборка данных осуществляется значительно быстрее, чем при многомерном концептуальном взгляде на реляционную БД, так как МБД денормализована и содержит заранее агрегированные показатели;
* многомерные БД легко справляются с задачами включения в информационную модель разнообразных встроенных функций [46].

MOLAP (Multidimensional OLAP) – для реализации многомерной модели используются многомерные БД. При этом данные хранятся в виде упорядоченных многомерных массивов. Такие массивы подразделяются на гиперкубы, в которых все хранимые в базе данных ячейки имеют одинаковую мерность. Физически данные хранятся в «плоских» файлах, при этом куб представляется в виде одной плоской таблицы, в которую построчно вписываются все комбинации элементов всех измерений с соответствующими им значениями мер.

На основании анализа многомерных БД можно выделить следующие условия, при которых их использование является эффективным:

* объем исходных данных для анализа не слишком велик (несколько гигабайт), т. е. уровень агрегации данных достаточно высок;
* набор информационных измерений стабилен;
* время ответа системы на нерегламентированные запросы является наиболее критичным параметром;
* требуется широкое использование сложных встроенных функций для выполнения кроссмерных вычислений над ячейками гиперкуба, в том числе возможность написания пользовательских функций [39].

## 4.7. Загрузка данных в хранилище данных

Первыми в процессе загрузки данных в ХД загружаются таблицы измерений, которые содержат суррогатные ключи и другую описательную информацию, необходимую для таблиц фактов.

При загрузке таблиц измерений требуется и добавлять новые записи, и изменять существующие.

При добавлении новых данных в таблицу измерений требуется определить, не существует ли в ней соответствующая записать. Если нет, то она добавляется в таблицу. В противным случае могут использоваться различные способы обработки изменений в зависимости от того, нужно ли поддерживать старую информацию в хранилище с целью ее последующего анализа.

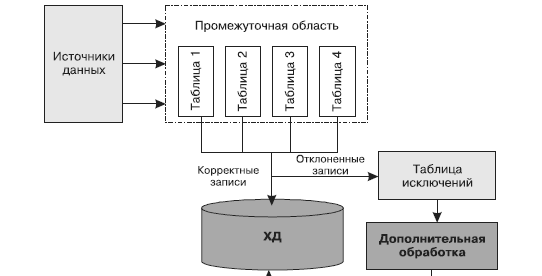
При загрузке таблицы фактов новая информация обычно добавляется в конец таблицы, чтобы не изменять существующие данные.

Для того чтобы загрузка данных прошла без каких-либо ошибок необходимо учесть ряд причин их возникновения:

* на этапе преобразования данных не удалось исправить все критические ошибки, которые блокируют загрузку записей в ХД;
* некорректный порядок загрузки данных. Например, предпринимается попытка загрузить факты для значения измерения, которое еще не было загружено;
* внутренние проблемы ХД, например, недостаток места в нем;
* прерывание процесса загрузки или остановка его пользователем.

При появлении данных, попытка загрузки которых потерпела неудачу, необходимо предусмотреть следующие действия (рис. 13):

* сохранить данные, не попавшие в ХД, в виде таблицы или файла того же формата, что и исходная таблица или файл (таблица исключений);
* провести анализ отклоненных данных для выявления причин, по которым они не были загружены;
* при необходимости произвести дополнительную обработку и очистку данных, после чего предпринять повторную попытку их загрузки.

  
Рис. 13. Последовательность действий при загрузке данных в ХД

Если и повторная попытка загрузки данных не увенчалась успехом, то в хранилище окажутся неполные данные, анализ которых может привести к неправильным выводам. Для решения этой проблемы можно:

* восстановить состояние хранилища, каким оно было до загрузки;
* полностью очистить в ХД таблицы с неполными данными [33].

## 4.8. Организация многомерной базы данных для специализированной метеорологической информации

Многомерные хранилища данных и многомерных анализ, используют, как правило, в интересах бизнеса для повышения качества принятия решений, быстрого построения всевозможных отчетов по продажам различных товаров в различных торговых точках. В настоящей работе делается попытка применить преимущества многомерного подхода для анализа специализированной метеорологической информации.

Нестационарная полуторамерная модель конвективного облака, используемая нами, позволяет получить в качестве результатов расчетов временной ряд динамических и микрофизических характеристик облака, таких как значения скорости восходящего потока, перегрев в облаке, водность и ледность облака, концентрации аэрозолей, капель, кристаллов льда различной формы, частиц ледяной крупы и града. Мы также можем рассчитать общее количество осадков на поверхности земли и радиолокационную отражаемость. Исследование эволюции всех этих характеристик в пространстве и времени представляют огромный интерес с научной точки зрения. Однако, сами по себе эти характеристике не могут однозначно указать на тип опасного конвективного явления, которое сопровождалось, или явилось следствием развития данного модельного облака.

Так, в модели не предусмотрен так называемый «электрический» блок, который позволяет рассчитывать напряженность электрического поля в облаке и фиксировать электрический разряд, наличие которого позволило бы однозначно указать на наличие грозы. Правда такое явление как град или сильный ливень можно определить по размеру ледяных частиц на поверхности земли и количеству выпавших на нее осадков. Однако, как правило, и сильный ливень и град могут быть спутниками грозы, поэтому однозначно определить какое именно опасное конвективное явление может развиться при данных начальных и граничных условиях, мы определить не можем.

Однако, для практического использования модели в целях оперативного прогноза именно определение типа конвективного явления является критически важным. Метеорологам в аэропортах важно знать: будет ли гроза в данном аэропорту и на пути следования самолета.

В связи с этим возникает проблема идентификации опасного конвективного явления по тем параметрам облака, которое позволяет рассчитывать численная модель. Мы предлагаем для решения этой проблемы использовать многомерный анализ.

Данная многомерная база данных будет представлять собой трехмерный куб (рис. 14), оси которого будут представлять собой следующие измерения: «Дата», «Явление», «Параметры облака (CF)». По оси «Дата» будут фиксироваться даты конкретных случаев, когда наблюдалось, или не наблюдалось, то или иное опасное конвективное явление. По оси «Явление» будут фиксироваться типы явлений (гроза, сильный ливень, без явлений, слабый дождь). По оси «Параметры облака» – вертикальная составляющая скорости, отклонение показателя температуры от температуры окружающей среды, относительная влажность (над водяной поверхностью), отношение смеси водяного пара, общее отношение водяных капель и вертикальная мощность облака, характеризующаяся как максимальная разность между верхней и нижней высотами облака.

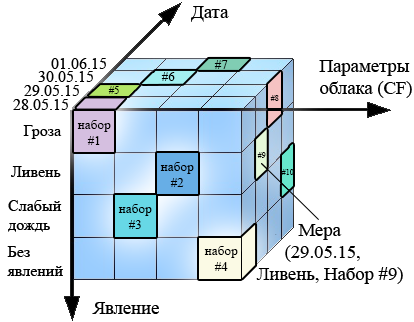


Рис. 14. Фрагмент многомерной базы данных, представленной в виде трехмерного куба

Заполнение многомерной базы будет проходить следующим образом: с помощью реляционной базы данных будут выбраны все случаи с наблюдавшимися в данном месте опасными конвективными явлениями, а также случаи без явлений, когда конвекция не развивалась. Соответственно, будут загружены данные о вертикальном распределении температуры и влажности в атмосфере, которые будут использоваться в качестве начальных и граничных условий в численной модели облака. Затем будут произведены расчеты, которые позволят определить выделенные выше параметры модельного облака. Значения рассчитанных параметров заполнят ячейки куба на пересечении соответствующих осей.

Когда ячейки будут заполнены, можно будет проводить соответствующий анализ. В частности, можно будет определить диапазон параметров облака, который будет соответствовать данному конвективному явлению.

Таким образом, в дальнейшем, при использовании модели для оперативного прогноза, проанализировав, в какой диапазон значений попали рассчитанные параметры облака, можно будет делать заключение о типе конвективного явления.

### Глава 5. Описание алгоритма работы системы

## 5.1. Выбор источников данных

Как было упомянуто выше, метеорологическую информацию, необходимую для верификации модели, можно найти в различных гетерогенных источниках, где она содержится в различных форматах. Выбор источников основывается на следующих принципах: источники должны предоставлять данные бесплатно, они должны содержать информацию о месте, дате и типе опасного явления и информацию о вертикальном распределении температуры и влажности в атмосфере окружающей среды на дату и место явления. Дополнительная информация о давлении и максимальной температуре на поверхности, типа облаков и синоптической ситуации желательна, но не обязательна. Из всего многообразия источников метеорологических данных, мы выбрали два, которые отвечают вышеуказанным критериям [8].

Интернет-сайты *http://weather.uwyo.edu/* и *http://meteocenter.net/* вместе позволяют получить всю необходимую для нас информацию. Сайт «Meteocenter» (http://meteocenter.net/) предоставляет архив данных, который можно использовать для извлечения информации о типе опасного явления, месте и дате, где оно произошло. Сайт Университета Вайоминга (UW) (http://weather.uwyo.edu/) позволяет получить вертикальные распределения температуры воздуха и влажности. Таким образом, используя информацию от первого сайта в качестве входных данных, можно извлечь необходимые данные из второго сайта.

Сайты различаются по своей структуре, языком пользовательского интерфейса и содержанием предоставляемой метеорологической информации. Поэтому у нас нет возможности создания универсального программного обеспечения, которое бы одинаковым образом извлекало информацию с обоих сайтов.

Например, данные за 01.05.2016 в СПб, взятые с сайта Университета Вайоминга, представлены на (рис. 15). Из этого источника нам потребуются все параметры [50].

-----------------------------------------------------------------------------

PRES HGHT TEMP DWPT RELH MIXR DRCT SKNT THTA THTE THTV

hPa m C C % g/kg deg knot K K K

-----------------------------------------------------------------------------

1018.0 78 3.4 2.8 96 4.62 280 4 275.1 287.8 275.9

1003.0 206 6.4 6.3 99 6.00 259 4 279.3 295.9 280.3

1000.0 232 6.4 6.0 97 5.90 255 4 279.6 295.8 280.6

954.0 619 6.7 3.2 78 5.07 190 10 283.7 298.1 284.6

946.0 688 6.8 2.7 75 4.94 182 10 284.4 298.5 285.3

929.0 837 6.2 3.7 84 5.39 165 12 285.2 300.6 286.2

925.0 872 6.0 3.9 86 5.50 165 12 285.4 301.1 286.4

911.0 997 4.6 2.8 88 5.16 170 18 285.2 300.0 286.1

897.0 1123 4.6 2.9 89 5.28 176 25 286.5 301.6 287.4

870.0 1372 4.4 0.8 77 4.68 187 38 288.8 302.4 289.6

850.0 1561 2.2 -3.8 65 3.41 195 49 288.4 298.5 289.0

841.0 1647 1.8 -4.2 64 3.35 184 36 288.9 298.8 289.5

830.0 1751 0.9 -4.7 66 3.26 170 19 289.0 298.7 289.6

738.0 2683 -7.3 -9.3 86 2.57 173 13 290.0 297.7 290.4

730.0 2768 -7.5 -9.3 87 2.60 174 12 290.6 298.5 291.1

716.0 2919 -7.3 -12.3 67 2.09 174 11 292.5 298.9 292.9

713.0 2952 -7.3 -18.3 41 1.28 174 11 292.8 296.9 293.1

700.0 3095 -7.9 -18.9 41 1.23 175 10 293.7 297.7 293.9

697.0 3128 -7.9 -18.9 41 1.24 175 11 294.1 298.0 294.3

693.0 3173 -8.2 -19.9 38 1.15 175 12 294.3 298.0 294.5

675.0 3375 -9.3 -24.4 28 0.79 210 4 295.2 297.8 295.3

656.0 3595 -10.6 -29.3 20 0.52 210 8 296.2 297.9 296.3  
Рис. 15. Таблица значений параметров с сайта Университета Вайоминга

Данные за 01.05.2016 в СПб, взятые с сайта «Meteocenter», представлены на (рис. 16). Из этого источника нам потребуются только некоторые параметры, такие как время, дата, явление и максимальная температура [38].

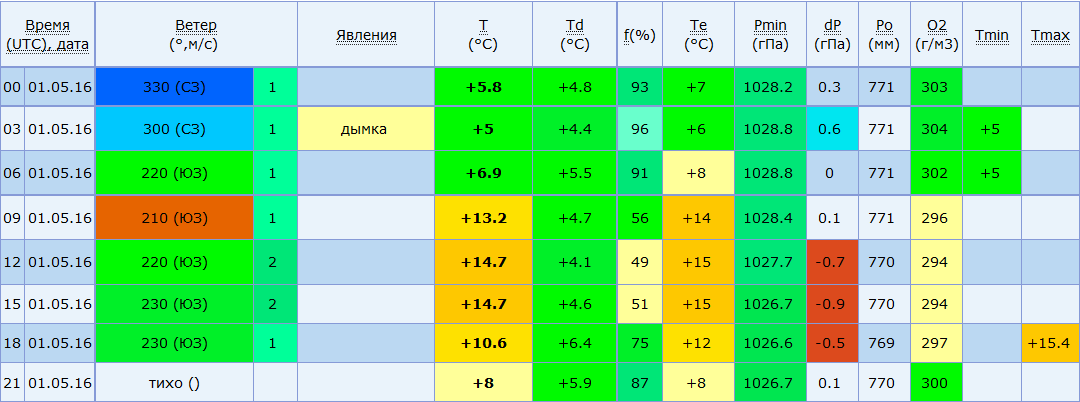


Рис. 16. Таблица значений параметров с сайта «Meteocenter»

Мы должны разработать специальное приложение (комплексную информационную систему), которое направлено на извлечение необходимых данных из гетерогенных источников и преобразование их в единую структуру. В результате, мы получим систему, которая позволит получать всю необходимую информацию и взаимодействовать с моделью конвективного облака (проводить моделирование опасных конвективных явлений), а также проводить многомерный статистический анализ с целью определения типа опасного явления. Для её реализации нам потребуется следующий инструментарий [20-23, 27, 29, 32]:

* серверная часть: PHP 5.6, MySQL 5.6 [51, 54];
* клиентская часть: HTML 5, CSS 3, JavaScript (ECMA Script 6).

Вся система реализована с помощью кроссплатформенной интегрированной среды разработки JetBrains PhpStorm 10.0.2 [61] и кроссплатформенной системой управления и анализа многомерных баз данных Jedox Palo BI Suite 6.0.2 [59].

Преимущества среды JetBrains PhpStorm:

* ускорение разработки. Достигается за счёт подсказок (тегов, функций, используемых переменных, классов и т.д.), удобного поиска (строки, переменной, метода) и другого;
* улучшение качества кода, т.е. качественный, быстрый и надежный рефакторинг;
* поддержка SQL и баз данных (рефакторинг схемы БД);
* интеграция с системами контроля версий (Git);
* имеет отличную интеграцию с инструментарием для тестирования PHPUnit.

Преимущества системы Jedox Palo BI Suite:

* OLAP-система с открытым исходным кодом, позволяющая стать клиентом для любой внешней программы или системы (например, ERP или CRM);
* быстрый доступ ко всем данным, поскольку Palo хранит их в памяти во время выполнения;
* множественное использование ХД несколькими пользователями;
* обработки сложных моделей данных для статистики и принятия решений;
* помимо многомерных запросов, данные могут быть записаны обратно в ХД и консолидированы в режиме реального времени.

Основными преимуществами языка серверного программирования PHP являются:

* PHP достаточно прост в изучении, но при этом обладает широким набором функций;
* не возникнет проблем с созданием с привязкой к определенной операционной системе или определенному Web-серверу, так как язык – кроссплатформенный;
* язык PHP очень гибок, т.е. не зависит от браузера, так как программный код выполняется на стороне сервера, а результат возвращается в виде HTML-документа;
* обладает высоким быстродействием;
* язык PHP – объектно-ориентированный, т.е. основными концепциями являются понятия объектов и классов;
* имеет огромный выбор готовых решений;
* язык предоставляет гибкие и эффективные средства безопасности;
* предоставляется бесплатно [21].

Так как для данного приложения был выбран язык серверного программирования PHP версии 5.6, в котором есть возможность использования библиотеки функций libcurl, то можно использовать технологию многопоточности (multi\_curl), а, следовательно, и многозадачность [24]. Данная библиотека позволяет взаимодействовать со множеством различных серверов по множеству протоколов, к тому же позволяет выполнять запросы параллельно, что значительно ускоряет получение данных с сайтов [44].

Наша система содержит несколько сложных компонент, обеспечивающих гетерогенное преобразование данных в единую структуру и систематизацию данных.

Сперва нам необходимо выяснить, какая именно информация интересует пользователя. Ему предлагается выбрать такие данные, какие он хотел бы использовать в дальнейшей работе с приложением. Если ранее он пользовался приложением, то сможет поработать и с выбранными в предыдущих сеансах данными. Можно выбрать интересующие параметры: даты, города, погодные явления и другие.

Далее, благодаря многопоточной передаче данных, происходит быстрое получение информации. Пользователь не видит самого процесса, так как он полностью автоматизирован, а видит лишь результат процесса ETL.

В результате извлечения, данные за короткое время попадают во временный файл, где преобразуются в необходимый формат. Затем они заполняют собой базу данных, с которой и будет работать пользователь с помощью Web-приложения. Весь алгоритм работы системы схематично представлен на (рис. 17-20). Рассмотрим подробнее ETL-этапы [9].

  
Рис. 17. Алгоритм работы системы

  
Рис. 18. Описание ETL этапов

  
Рис. 19. Использование БД для прогнозирования

Рис. 20. Использование многомерного статистического анализа

## 5.2. Извлечение данных из гетерогенных источников

*Этап 1*. После того, как пользователь успешно авторизовался, осуществляется автоматический переход на главную страницу приложения. Ему предлагается выбор: либо занести необходимую информацию в БД, что может быть выполнено вручную или автоматически, либо перейти к управлению данными, хранящимися в БД. Рассмотрим вариант автоматического заполнения БД посредством извлечения информации из гетерогенных источников, которыми являются два сайта, представленные выше.

Поскольку выбранные нами источники имеют свои системы выгрузки данных, то воспользуемся ими. Эти системы дают нам доступ ко всей информации, хранящейся на сайтах в свободном доступе. С одной стороны, это упрощает программную часть, поскольку нам не требуется создавать систему выгрузки самостоятельно, с другой стороны, у нас нет выбора, и мы можем только подстроиться под предложенную форму.

Извлечение начинается с русского сайта, так как в нем хранится необходимая информация, и она же является входной для американского сайта. Пользователь вводит интересующие его параметры: время, дата, город, регион. Так как мы не можем получить данные с русского сайта напрямую, мы воспользуемся библиотекой libcurl, которая позволит послать необходимый запрос. Сначала нужно инициализировать сеанс Curl: *$curl = curl\_init()*, после этого нужно указать опции для сеанса: *curl\_setopt($curl,$opt)*; адрес – *URL*, возвращать результат –*RETURNTRANSFER*, метод – *POST*, параметры, которые ввел пользователь –*POSTFIELDS*. Затем начинается извлечение информации с помощью выполнения следующего кода: *curl\_exec($curl)*. В итоге, извлеченные данные будут храниться в файле в формате CSV (*Comma-Separated Values –* значения, разделенные запятыми), так как выгрузка с русского сайта позволяет получить такой формат файла, который будет удобен для последующего анализа. Сохраним полученную информацию во временный файл *$out\_csv* и завершим сеанс с освобождением ресурсов *curl\_close($curl)*.

Для извлечения данных с американского сайта нам потребуется агрегированная информация с русского сайта (подробнее в этапе 2), а точнее параметр времени, то есть мы будем выбирать время по условию наличия параметра «явление». Для этого мы воспользуемся массивом *$new\_sel\_values[$i][$j]*, полученным на этапе агрегирования (этап 2) и выберем в нем только те временные интервалы, которые интересуют пользователя. После чего скопируем их в ассоциативный массив *$arg\_times[$k][$l]*, хранящий информацию о днях и соответствующие им периоды времени. После этого, мы создадим массив ссылок-запросов следующего вида: *$ref\_req=Array('http://weather.uwyo.edu/cgi-bin/sounding?region=europe&TYPE=TEXT%3ALIST&YEAR=2014&MONTH=03&FROM=0100&TO=0100&STNM=26063,…)*, где *region* – регион, *TYPE*– тип представления данных, *YEAR* – год, *MONTH* – месяц, *FROM* – с какого периода, *TO* – по какой период, *STNM* – индекса станции(города). Как раз в нем и используются агрегированные данные из *$arg\_times*. Далее мы проводим проверку на корректность запроса: разобьем запрос на компоненты, используя функцию *parse\_url($ref\_req[$i], PHP\_URL\_QUERY)* и применим к каждому компоненту свою проверку: проверку на целочисленность, отсутствие пробелов и специальных символов для *YEAR*, *MONTH*, *FROM*, *TO*, *STNM*. Для *region* осуществим проверку на отсутствие пробелов и специальных символов. Если они есть, то уберем их с помощью функции *trim($components[$j])* и проверим, чтобы переменная *region* состояла только из букв латинского алфавита. Затем нужно осуществить параллельное извлечение данных с сайта. Поскольку библиотека позволяет создавать множество экземпляров своего механизма и работать параллельно, создаем массив дескрипторов *$curls=array()* и массив результатов запросов *$result = array()*, дескриптор мультипотока *$mh = curl\_multi\_init()*, т.е. он отвечает за то, чтобы несколько запросов выполнялись параллельно. Далее для каждого *url* создаем отдельный curl механизм, который будет выполнять отдельный запрос *$curls[$id] = curl\_init()*. После этого добавляем массив опций, как в случае с сайтом «Meteocenter», *curl\_setopt($curls[$id], $opts)*, только с единственной разницей, что у нас не один curl механизм, а несколько. Далее, добавляем текущий механизм к числу работающих параллельно. Затем выполняем запросы функцией *curl\_multi\_exec($mh, $running)* пока они имеются в наличии, т.е. пока переменная *$running* больше нуля. В завершении, собираем из всех созданных механизмов результаты, а сами механизмы удаляем и освобождаем память от механизма мультипотоков *curl\_multi\_close($mh)*. В результате мы имеем данные с определенной структурой во временном файле *$out\_txt*, где каждая строка состоит строго из 77 символов [18].

## 5.3. Преобразование и очистка данных

*Этап 2*. Следующим этапом является преобразование данных. Нам нужно произвести подготовку данных к размещению в БД и приведение их к виду, наиболее удобному для хранения и анализа. Процесс преобразования состоит из пяти шагов. Рассмотрим их более подробно.

*Шаг 1. Преобразование структуры данных*. Так как данные могут различаться своей структурой и форматами, нам необходимо провести реструктуризацию, т.е. привести все данные к виду, который требуется для анализа. В нашем случае данные с сайтов различаются присутствием специальных символов, таких как: открывающаяся и закрывающаяся скобка ( { , } ); косая черта ( / ); звездочка ( \* ), от которых необходимо отказаться. Также недопустимы пропуски, т.е. в некоторых ячейках встречаются пустые значения, которые небезопасны для обработки, поэтому их нужно преобразовать в вид 9999.0, который сможет читать программный код модели. Так как пропуски, которые могут критически сказаться в прогнозе конвективного облака содержатся только на сайте Университета Вайоминга, мы применим следующее: находим все строчки в файле, полученном после 1-го этапа, с помощью функции *preg\_match\_all('/(.\*){77}/', $matches[1][0], $matches\_count\_str)* на вход, которой подаем шаблон поиска, место поиска и переменную, в которой будет храниться результат. После поиска заменяем пустые ячейки на нужное значение с помощью функции *ereg\_replace("[\t ]{7}", "9999.0", $temp[$z][$l])*, затем проводим проверку на числовые данные и получаем их с помощью функции *preg\_match\_all("(-?\d+(?:\.\d+)?)", $str, $matches\_values)*, т.е. мы образуем массив *$matches\_values*, хранящий параметры для прогнозирования, которые будут использоваться в дальнейшей обработке.

С сайтом «Meteocenter» данная процедура будет намного проще. Мы просто применяем анализатор (парсер) для формата CSV, после чего получаем все данные и заносим их в массив *$values*, затем выбираем только 1-й (отвечает за период времени), 2-й (за дату), 6-й (за явление) и 17-й (за Tmax) столбцы в этом массиве, в котором хранятся данные после анализа (парсинга). Но, как уже говорилось выше, 6-й и 17-й столбцы могут быть пустыми, так как могло не произойти никакого явления и не зафиксирована максимальная температура за прошедший день. Таким образом, выборка требуемых столбцов будет происходить по следующему правилу: совершим обход массива *$values[$i][$j]* по всем строкам *$i* и по столбцам *$j*, указанным выше и выберем в нем интересующие нас столбцы. В результате, получим требуемую выборку, которая будет храниться в массиве *$sel\_values[$k][$l]*. Дополнительно создадим ассоциативный массив явлений: *$assoc\_phen = array(0 => "без явления", 1 =>"сильный ливень", 2 => "гроза", 3 => "слабый дождь")* и воспользуемся им, чтобы преобразовать номер явления в его название и, наоборот. В массиве *$sel\_values* заменим все явления их численным эквивалентом. Для этого совершим обход массива, найдем явление и преобразуем его в число с помощью *$assoc\_phen*. В итоге, *$sel\_values* будет состоять только из числовых параметров.

*Шаг 2. Агрегирование данных*. Здесь нашей целью является сделать данные менее детализированными, тем самым уменьшив их количество и упростив работу с ними, но при этом уровень качества данных не должен упасть ниже допустимого минимума, установленного нами. На первом этапе при извлечении информации с сайта Университета Вайоминга было задействовано агрегирование данных. Так как данные с сайта Университета Вайоминга предоставляются за период 00:00 часов и 12:00 часов, а с «Meteocenter» вплоть до 2005 года сутки делились на 4 равных промежутка, начиная с 00:00. Вследствие этого, мы агрегируем данные по алгоритму: анализируем массив *$sel\_values[$i][$j]* построчно, если в строке есть явление, то находим время, в которое оно произошло, иначе переходим к следующей строке и продолжаем поиск. Процесс завершается, как только будут пройдены все строки массива. Найденное время проверяем на условие: если оно попало в промежуток с 18:00 до 06:00, то считаем это ночью, т.е. параметр времени равен 00:00 для сайта Университета Вайоминга, если же оно попало в промежуток с 06:00 до 18:00, то считаем это днем, т.е. параметр времени – 12:00. После этого, отправляем параметр времени в запрос на получение данных с сайта Университета Вайоминга (описано в 1-ом этапе). После 2005 года на сайте «Meteocenter» действует разделение суток на 8 равных промежутков времени начиная 00:00, что в любом случае не совпадает с делением на периоды времени на сайте Университета Вайоминга, следовательно, данные необходимо агрегировать. Проведем это по предыдущей схеме. В результате, у нас будет создан новый массив *$new\_sel\_values*, хранящий агрегированную информацию.

*Шаг 3. Перевод значений*. Данные в первоначальных источниках хранятся в кодировках, удовлетворяющих нужды владельцев сайтов, при этом данные могут иметь значительный объем. Для наших целей потребуется кодировка «*windows-1251*», так как приложение, рассчитывающее конвективное облако, работает именно с ней. Для определения кодировки символов используем *$ecn\_data = mb\_detect\_encoding($oldencdata)*, где *$oldencdata* – это массив параметров, полученных с «Meteocenter» или сайта Университета Вайоминга. После выполнения функция возвращает кодировку. Так как на сайте Университета Вайоминга используется кодировка «*ANSI*», а на «Meteocenter» – «*windows-1251*», то, используя функцию преобразования *iconv($enc\_american, "windows-1251", $oldencdata\_american)* для сайта Университета Вайоминга, получим все данные в кодировке «*windows-1251*». Также потребуется преобразовать дату и время в единый формат для удобного хранения в БД, но это не составит проблем, так как СУБД MySQL сделает это преобразование за нас на этапе загрузки наших параметров в БД.

*Шаг 4. Создание новых данных*. Порой нам надо больше данных, чем мы получили из первоисточника. Иногда мы можем получить их посредством агрегирования, т.е. создания новых данных с помощью манипуляций с уже имеющимися. Таким образом, можно добавить новые поля со значениями, вычисленными из существующих. В нашем случае, мы создали новые данные на этапе агрегирования, когда обобщали временной интервал.

*Шаг 5. Очистка данных*. Чаще всего к нам попадают данные нуждающиеся в очистке. Они могут нарушить результаты анализа и стать причиной невозможности применения некоторых методов и алгоритмов. Метод очистки данных зависит от загрязняющих факторов, которые чаще всего встречаются в этой области, например: неправильные форматы, нарушения структуры данных, пропуски, дубликаты и т.д. Нам же очистка данных не требуется, поскольку данные, взятые из первоисточников, в нашем случае уже очищены. Единственное, что нам нужно сделать, это рассмотреть возможность как появление дубликатов. Здесь нам поможет СУБД MySQL. С целью обезопасить себя от дубликатов мы установили для полей уникальные индексы: *CREATE UNIQUE INDEX(index\_name1,…) on TABLE (index\_col\_name1,…)*. Дубликаты, которые в результате этого все же не были удалены, будем удалять с помощью SQL-запросов. Пропусков быть не может, поскольку мы в таких случаях либо заменили их соответствующим значением, либо удалили пробельный символ [18].

## 5.4. Загрузка данных из гетерогенных источников в БД

*Этап 3*. После извлечения данных из гетерогенных источников и их преобразования, агрегации и очистки, осуществляется последний этап ETL-процесса – загрузка данных в БД. Процесс загрузки заключается в переносе данных из промежуточных областей (временных файлов) в структуру БД. Поскольку все необходимые параметры уже хранятся в соответствующих массивах, мы загрузим их с помощью SQL-запросов на вставку в БД. Также возможна загрузка из локальных источников с определенной структурой файла, например, *CSV*, как на сайте «Meteocenter» и *TXT*, как на сайте Университета Вайоминга. БД создана конкретно под приложение и удовлетворяет всем необходимым критериям. Она позволяет осуществлять конкретный поиск по необходимым параметрам, например, явлению, тем самым освобождая нас от поиска вручную.

После загрузки данных необходимо удостовериться, что данные, помещенные в БД, имеют необходимый уровень достоверности и информативности. Для этого воспользуемся комплексом верификационных тестов, состоящего из: целостности данных, т.е. отсутствие случайных значений, либо присутствие null-значений; бизнес-логики и емкостные или ресурсные ограничения [48].

Если в процессе выполнения верификационных тестов обнаруживаются какие-либо аномалии, то они фиксируются и после завершения тестов, мы уведомляем пользователя о наличии некорректных данных.

Приведенные выше тесты можно проводить как в автоматическом режиме, так и в ручном режиме с интуитивно понятным интерфейсом.

После выполнения постзагрузочных тестов, описанных выше, которые производились в автоматическом или ручном режиме, осуществляется передача управления пользователю. Ему предоставляется возможность выбора необходимой операции: просмотр, добавление, редактирование, обновление, удаление, создание специального файла для прогнозирования опасного конвективного облака [18].

## 5.5. Моделирование интегральных и спектральных характеристик конвективного облака

После того как загрузка данных успешно завершена и выполнены все верификационные тесты, подтверждающие соответствующий уровень качества данных, переходим в раздел «Верификация», заполняем интересующие нас поля и нажимаем кнопку «Получить». В результате чего генерируется специализированный файл с данными, содержащий начальные и граничные условия для полуторамерной модели конвективного облака. Данный файл загружается на локальную машину и передается в программный комплекс для расчета характеристик облака.

Перед тем как загрузить данные в модель из только что полученного файла, необходимо произвести настройку модели, т.е. задать параметры для численного эксперимента: время развития облака, радиус внутреннего и внешнего столбов, влияние окружающей среды, учет коагуляции, коэффициент вертикальной турбулентности, коэффициент бокового турбулентного перемешивания и другие.

После чего загружаем файл с параметрами в настроенную для эксперимента модель и проверяем полученные данные с помощью вспомогательного инструмента, встроенного в модель. Он визуализирует входящие данные в виде аэрологической диаграммы и позволяет редактировать входной файл в случае необходимости. Если на аэрологической диаграмме имеется некорректное отображение ветра, температуры воздуха или температуры точки росы, которая характеризует влажность в атмосфере, то мы вносим соответствующие изменения в файл самостоятельно.

Следующим шагом будет моделирование развития конвективного облака и его спектральных и интегральных характеристик. Для того чтобы начать данный процесс, необходимо зафиксировать время до которого будет проходить моделирование и нажать кнопку «Выполнить до». В результате численного моделирования характеристик конвективного облака, на которое затрачивается незначительное время, мы получаем картину эволюции облака и сгенерированные моделью выходные CSV-файлы. Все выходные данные в них отображаются в виде таблицы, столбцами которой являются время, высота над уровнем моря, название параметра и значения параметра (рис. 21). В ходе численных расчетов вычисляются следующие параметры облака: вертикальные и радиальные компоненты скорости, давления, плотность воздуха, температура окружающей среды, избыток в облаке, относительная влажность, вертикальная высота облака, соотношение смешивания водяного пара, пропорции смешивания аэрозолей, капли воды, частицы льда, крупы и града.

Также имеется возможность рассчитать интегральные и спектральные характеристики вручную, проводя моделирование опасного явления по шагам, т. е. зафиксировать шаг по времени и шаг по высоте, а далее, с помощью нажатия кнопки «След. шаг» производить расчет характеристик облака в интерактивном режиме. Для этого используются следующие значения параметров: шаг по времени ∆t = 20 сек, прирост в высоту ∆h= 150 м, коэффициент смешивания α2 = 0.08, коэффициент вертикального турбулентного перемешивания Kv = 100. Расчетная область H равна 10 км, радиусы внутренних и внешних цилиндры равны 3 и 10 км соответственно. Все вычисления выполняются до периода 1 часа, что является характерным временем эволюции конвективного облака.

|  |  |  |  |
| --- | --- | --- | --- |
| Время (с) | Высота (м) | Название параметра | Значение параметра |
| 1600.0 | 7200.0 | velocity | 11.185048 |
| 1600.0 | 7200.0 | velocityU | 0.48835342 |
| 1600.0 | 7200.0 | temperature | 251.09097 |
| 1600.0 | 7200.0 | deltaTemperature | 5.2771633 |
| 1600.0 | 7200.0 | relativeHumidity | 1.0433166 |
| 1600.0 | 7200.0 | vapor | 0.0016571974 |
| 1600.0 | 7200.0 | pressure | 40749.573 |
| 1600.0 | 7200.0 | density | 0.56537216 |
| 1600.0 | 7200.0 | aerosol | 0 |
| 1600.0 | 7200.0 | drop | 0.0011597369 |
| 1600.0 | 7200.0 | ice | 1.9362147E-0005 |
| 1600.0 | 7200.0 | hailAndGrits | 0.015275136 |
| 1600.0 | 7200.0 | velocity | 11.295842 |
| 1600.0 | 7200.0 | velocityU | 0.88229662 |
| 1600.0 | 7200.0 | temperature | 249.89693 |
| 1600.0 | 7200.0 | deltaTemperature | 5.1170796 |
| 1600.0 | 7200.0 | relativeHumidity | 1.0573496 |
| 1600.0 | 7200.0 | vapor | 0.0015417189 |
| 1600.0 | 7200.0 | pressure | 39917.628 |
| 1600.0 | 7200.0 | density | 0.55647577 |
| 1600.0 | 7200.0 | aerosol | 0 |
| 1600.0 | 7200.0 | drop | 0.0010368739 |
| 1600.0 | 7200.0 | ice | 1.9885503E-0005 |
| 1600.0 | 7200.0 | hailAndGrits | 0.015823842 |

Рис. 21. Фрагмент CSV-файла с параметрами

По завершении работы модели результаты расчетов интегральных и спектральных характеристик опасного явления сохраняются в CSV-файл, который загружается на сервер либо может быть сохранен на локальной машине пользователя.

## 5.6. Загрузка данных в хранилище данных

После того как проведено численное моделирование интегральных и спектральных характеристик конвективного опасного явления можно приступать к загрузке полученных данных в многомерное хранилище для последующего анализа.

Для того чтобы загрузить данные в ХД будет использовано PHP API (Application Program Interface написанный на языке PHP) [60], предоставляемый разработчиками Jedox Palo. Перед тем как загрузить выходные данные модели в многомерное хранилище, необходимо извлечь недостающие данные из реляционной БД. Для этого мы перейдем в раздел «Численные эксперименты» и в нем выберем интересующий нас эксперимент. Нажимая кнопку «Получить», выполняется SQL-запрос к реляционной БД на получение тех параметров, которые использовались в данном эксперименте, а именно, зафиксированная дата возникновения явления, его название и максимальная температура. После чего извлеченные параметры помещаются во временную область, в которой и будет происходить создание новых данных для заполнения ими многомерного куба данных.

Создание новых данных происходит следующим образом: поскольку SQL-запрос уже выполнен и возвращен результат в массив *$ValuesRDBforCube*, находящийся во временной области, то в столбцах этого массива будут храниться следующие данные: 1-й (дата), 2-й (название явления) и 3-й (максимальная температура). На этом этап создания данных завершится.

Следующим шагом будет загрузка выходных данных модели в промежуточную область для преобразования в соответствующие единицы данных, которые также будут загружены в многомерный куб.

Поскольку выходные данные модели хранятся в CSV-файле, который после проведения численного моделирования сохраняется на сервере, мы просто применяем анализатор (парсер) для формата CSV, как и в случае трансформации данных в ETL-процессе, после чего получаем все данные и заносим их в массив *$tmpValuesCSVforCube*. Столбцы массива будут содержать следующую информацию: 1-й (время), 2-й (высота), 3-й (параметр) и 4-й (значение параметра). Как уже упоминалось в пункте 4.9, нас будут интересовать только параметры: вертикальная составляющая скорости, отклонение показателя температуры от температуры окружающей среды, относительная влажность, отношение смеси водяного пара, общее отношение водяных капель и вертикальная мощность облака. Создадим массивы данных для интересующих нас значений параметров, условно говоря, сделаем фильтрацию массива *$tmpValuesCSVforCube*: *$arrayVelocity*, *$arrayTemperature, $arrayHumidity, $arrayVapor, $arrayDrops и $arrayThickness*. Чтобы заполнить вышеуказанные массивы параметров, необходимо сделать обход массива *$tmpValuesCSVforCube* и как только встречается какой-либо из интересующих параметров, добавляем его значение в соответствующий массив. В результате чего мы получим новые данные, т.е. массивы с соответствующими значениями конкретных параметров.

Следующим этапом после создания новых данных, а именно, массивов значений *$ValuesRDBforCube, $arrayVelocity*, *$arrayTemperature, $arrayHumidity, $arrayVapor, $arrayDrops и $arrayThickness* является операция по очистке данных. Поскольку данные, взятые из первоисточников (UW и Meteocenter) уже очищены, то нет необходимости в повторной очистке. Единственное, что необходимо учесть – ситуацию отсутствующих данных в массиве *$ValuesRDBforCube*. Данная ситуация имеет место, когда происходит сбой работы системы. Чтобы обезопаситься от подобной проблемы, мы сохраняем соответствующую информацию в специальных файлах (логах), которые используются для восстановления прежнего состояния системы при ее сбоях. Если проверка на отсутствующие значения завершилась успешно, то переходим к следующему шагу загрузки данных. В противном случае, уведомляем пользователя о факте возникновения ошибки.

Процесс загрузки заключается в переносе данных из промежуточных областей в структуру хранилища данных. Поскольку Jedox API, предоставляемый разработчиками, имеет возможность оперировать с многомерными массивами данных, то объединим все массивы с параметрами в один многомерный ассоциативный массив *$AllParamsforCube* со специализированной иерархической вложенностью, которая служит для упрощения загрузки данных в куб, т.к. совпадает с иерархией измерений спроектированного куба данных, рассмотренного в предыдущей главе.

Для того чтобы загрузить данные в многомерный куб, необходимо установить соединение с сервером OLAP PALO с помощью вызова функции *$connection = palo\_init(HOST, PORT, USER, PASS), где HOST* – имя хоста (IP или DNS), *PORT* – номер порта OLAP-сервера, *USER* и *PASS* – имя пользователя и пароль соответственно. Следует отметить, что функция *palo\_init* возвращает дескриптор соединения с OLAP-сервером, который хранится в течении всей сессии клиента, пока не будет вызвана функция *palo\_disconnect($connection)*, в которую и будет передан вышеуказанный дескриптор. Далее необходимо задать многомерный массив измерений с координатами, в который будут загружаться данные: *$arrayDim = array($Date, $Phenomenons, $IdParamaters[$i])*, где *$Date –* значение даты, когда зафиксировано опасное конвективное облако, т.е. *$ValuesRDBforCube[0]*; *$Phenomenons –* название явления, т.е. *$ValuesRDBforCube[1]*; *$IdParamaters – массив ссылок на $Paramaters[$i]*, связанных по следующему правилу: например, если *$IdParamaters[0]=5*, то берем *$Paramaters[5]*, а в свою очередь *$Paramaters[$i] –* многомерный массив параметров *$AllParamsforCube*, т.е. массив, состоящий из массивов конкретных параметров, которые определяются значением *$i*. Если необходимо загрузить конкретное явление, то будем использовать *$i* в качестве ссылки на массив данных, в противном случае будут загружены все явления [60].

Таким образом, в зависимости от явлений будут использоваться те или иные массивы с интегральными или спектральными характеристиками облака, которые были получены благодаря численному моделированию конвективного явления. Чтобы загрузить эти данные в OLAP-куб, в цикле вызываем функцию на вставку данных: *palo\_setdataa($Paramaters[$i][$j][$maxTemp], false, $connection, $db\_name, $cube\_name, $arrayDim[$k])*, где *$connection* – дескриптор соединения, *$db\_name* – *Forecasting, $cube\_name – PredictionCube, $Paramaters[$i][$j][$maxTemp]=$ValuesRDBforCube[2]*.

В результате выполнения вышеуказанной функции, вся метеорологическая информация, необходимая для анализа с целью определения типа конвективного явления, будет загружена в хранилище данных. В заключении этапа загрузки следует завершить соединение с OLAP-сервером, вызывая функцию *palo\_disconnect($connection)*, а также освободить все ресурсы, которые были использованы для выполнения данной задачи.

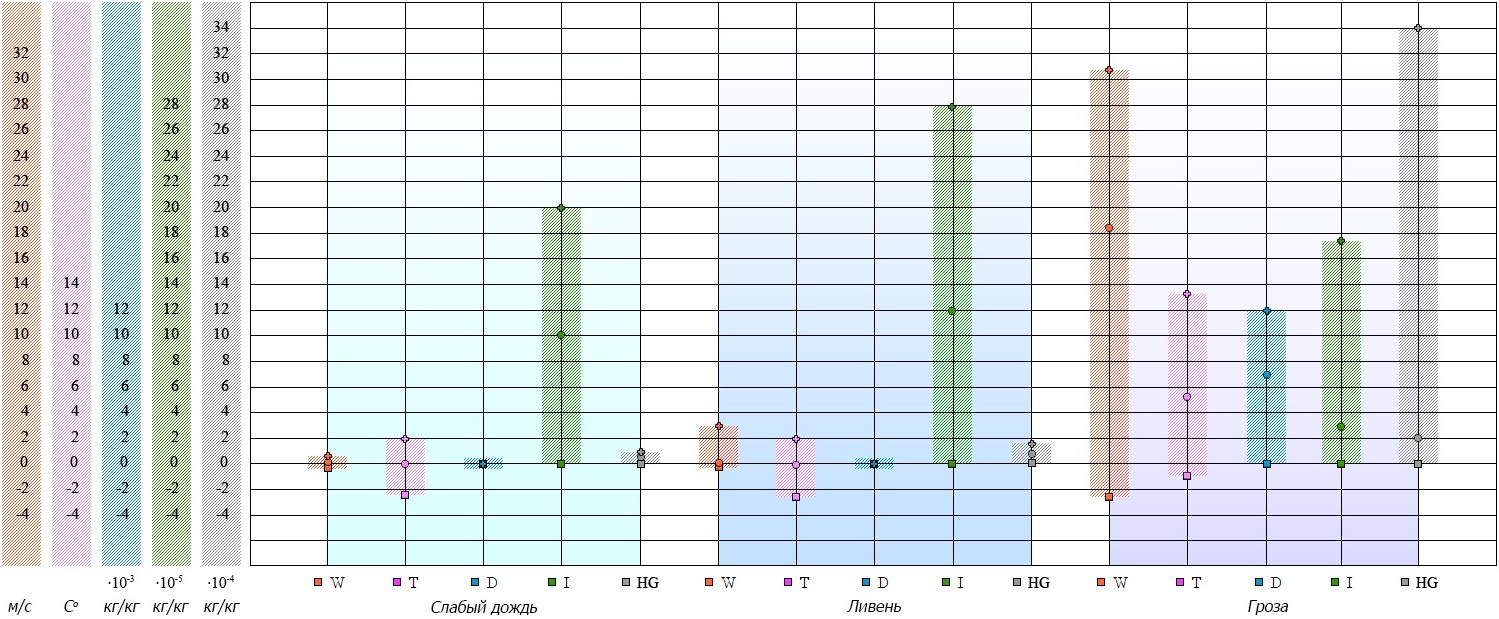
## 5.7. Анализ многомерного куба данных с целью определения диапазонов параметров для конвективного явления

После того как данные успешно загружены в хранилище данных можно переходить к их анализу. Была поставлена следующая задача: провести классификацию типов опасных конвективных явлений. Для этого будут исследованы изменения характерных параметров конвективного облака.

Используя разработанную комплексную информационную систему, был выявлен диапазон изменения характерных параметров облака. Анализ показал (рис. 22), что в случае грозы значения скорости восходящего потока, перегрева в облаке, а также отношение смеси водяных капель и ледяных частиц меняются в гораздо большем диапазоне значений, чем соответствующие параметры при слабом дожде и ливне. Средние значения этих же параметров в случае грозы также значительно превышают соответствующие средние значения в двух других случаях конвективных явлений. Однако полученные данные не позволили найти существенные отличия в значениях параметров дождя и ливня. Очевидно, этот вопрос требует дальнейшего исследования с привлечением большего количества исходных данных.

Таким образом, можно дать следующие рекомендации по прогнозированию грозы по данным расчетов:

1. зафиксировать момент максимального развития облака (максимальной мощности);
2. определить уровень максимальной водности облака;
3. на уровне из пункта 2 найти значения скорости восходящего потока, перегрева, отношения смеси водяных капель и ледяных частиц;
4. если значения скорости восходящего потока превышают 18 м/с, перегрева 5 градусов, отношение смеси водяных капель, ледяных частиц и града 7∙10-3 кг/кг, 3∙10-5 кг/кг и 2∙10-4 кг/кг соответственно, то можно прогнозировать грозу.

  
Рис. 22. Результаты многомерного статистического анализа.   
W – вертикальная составляющая скорости, T – перегрев, D – отношение смеси водяных капель, I – отношение смели ледяных частиц и HG – сумма отношений смеси града и крупы

### Глава 6. Тестирование и демонстрация разработанной комплексной информационной системы

## 6.1. Тестирование разработанной системы

В любой современной методологии разработки ПО тестирование является неотъемлемой частью процесса, а значимость его написания не ниже значимости написания кода. Поэтому главной задачей после разработки программного продукта является его тестирование, т.е. деятельность, направленная на выявление такого рода несоответствий между ожидаемым и действительным [47].

Для разработанного нами программного продукта был произведен ряд тестирований: Init-тесты инициализации и авторизации, функциональные CRUD-тесты для реляционное и многомерной баз данных, end-to-end сценарии, позитивные тесты, негативные тесты, которые были основаны на модульном тестировании (unit testing) [42].

В настоящее время PHPUnit [63] наиболее популярный фреймворк для юнит-тестирования в PHP. Кроме наличия таких возможностей, как «моки» («подделки») объектов, он также может анализировать покрытие кода, логировать, тестировать исключения, генерировать отчеты, а также предоставляет тысячи других возможностей. Среда разработки PhpStorm, которая была выбрана для реализации комплексной системы, включает в себя возможность использования PHPUnit тестирования [53]. Воспользовавшись этим богатым функционалом, были получены следующие результаты:

Init-тест (init / login / logout), CRUD-тест (Create, Read, Update, Delete), end-to-end сценарий (имитация последовательности действий пользователя), позитивный тест (корректность завершения разрешенных операций, входящих в CRUD-тесты) приложение прошло успешно и были получены положительные результаты. Негативный тест (корректность завершения разрешенных операций, входящих в CRUD-тесты) не был пройден продуктом. В результате теста была выявлена ошибка в работе системы, так как не было введено ограничение на количество вводимых символом в поля с ручным вводом. Основываясь на результатах теста, была проведена отладка программного кода.

Тест «end-to-end сценарий» не выдал ошибок, но мы обратили внимание на многократное обращение «виртуального пользователя» к разделу «база данных», поскольку её очистка была возможна только из этого раздела. В результате этого было принято решение о создании такой кнопки в разделе «Экстракция данных». После этого была произведена повторная проверка и количество обращение к разделу «База данных» с целью очистки базы данных значительно сократилось.

Также было проведено тестирование защищенности, в результате которого было выяснено, что можно легко подобрать пароль к профилю пользователя. В следствие этого был изменен алгоритм генерации пароля. Теперь он состоит из латиницы, цифр, специальных символов и учитывается регистр.

## 6.2. Демонстрация разработанной системы

Для создания программной среды были проанализированы основные типы программных продуктов для персональных компьютеров, такие как Web-приложения и настольные приложения [41]. Из рассмотренных типов было выбрано использование Web-приложения.

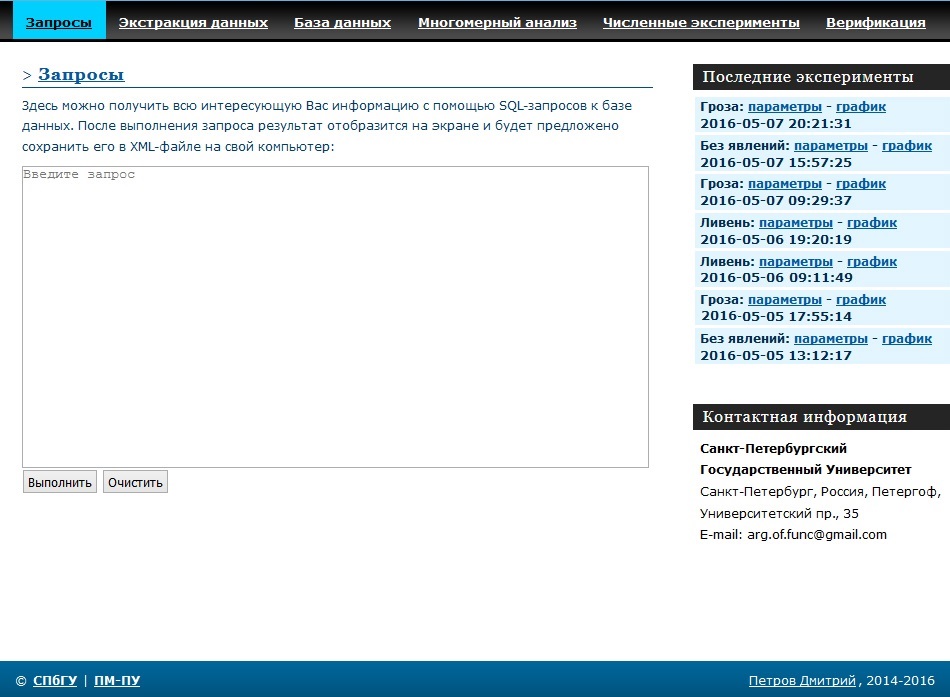
Причинами послужило следующее: такое приложение всегда мобильно, то есть, где бы не находился пользователь, если там есть выход в интернет и локальная машина, он сможет, не скачивая никаких дополнительных продуктов, начать работу. Это позволит сэкономить время, которое заняло бы скачивание и установка продукта, интернет-трафик и место на жестком диске пользователя. Для работы как настольного приложения, так и Web-приложения нужен постоянный выход в интернет, поскольку наши гетерогенные источники также расположены в сети интернет и подключение необходимо для получения из них новых, необработанных данных и использования уже консолидированных, хранящихся в БД. При этом, с помощью подключения к интернету мы можем сохранить их из БД на свой персональный компьютер и позднее работать с ними без подключения к сети интернет в любое время в любом месте.

Благодаря тому, что был выбран тип «Web-приложение» гораздо проще определить системные требования продукта. Они заключаются лишь в том, чтобы локальная машина поддерживала хотя бы один из современных браузеров, например, таких как: *Microsoft Internet Explorer, Google Chrome, Mozilla Firefox, Opera, Safari, Яндекс. Браузер* и т.п.

Итак, рассмотрим подробнее разделы разработанного Web-приложения.

В разделе «Запросы» пользователь может напрямую обращаться к данным, хранящимся в БД, с помощью написания SQL-запросов и сохранять их результат в XML-файле для дальнейшего анализа (рис. 23).

Рис. 23. Раздел «Запросы»



В разделе «Экстракция данных» нам предоставляется возможность указать параметры, на основе которых будут получены данные из гетерогенных источников. Необходимо выбрать регион, город (метеостанцию) и дату, после чего кликнуть по кнопке «Получить» (рис. 24).

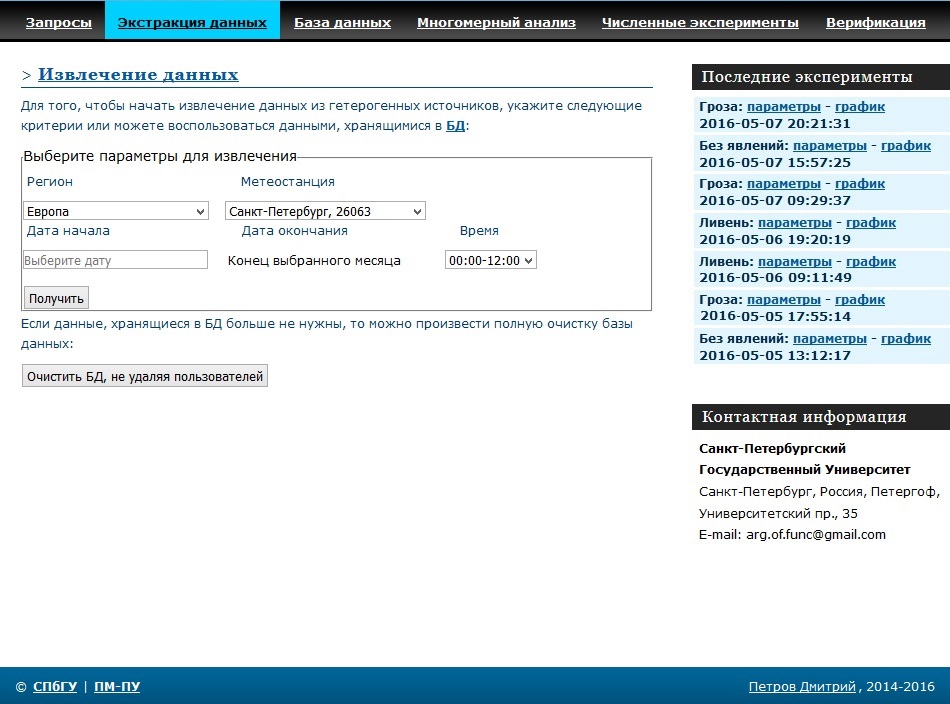


Рис. 24. Раздел «Экстракция данных»

На основе входных параметров система формирует таблицу извлеченных данных с сайта «Meteocenter». Переход на следующий этап будет произведен без участия пользователя.

На основе данных, полученных из источника «Meteocenter», и после их агрегирования, мы запрашиваем с сайта Университета Вайоминга параметры, которые будут использоваться для верификации полуторамерной нестационарной модели конвективного облака с дальнейшим преобразованием всех полученных данных, которые отображаются в виде таблицы. Осуществляется автоматический переход на последний этап, на котором происходит заполнение БД данными.

Раздел «База данных» состоит из 9 таблиц, в которых содержатся все ранее полученные данные. Мы можем просматривать их, а также осуществлять управление: добавление, редактирование и удаление. Имеется возможность полной очистки БД (рис. 25).

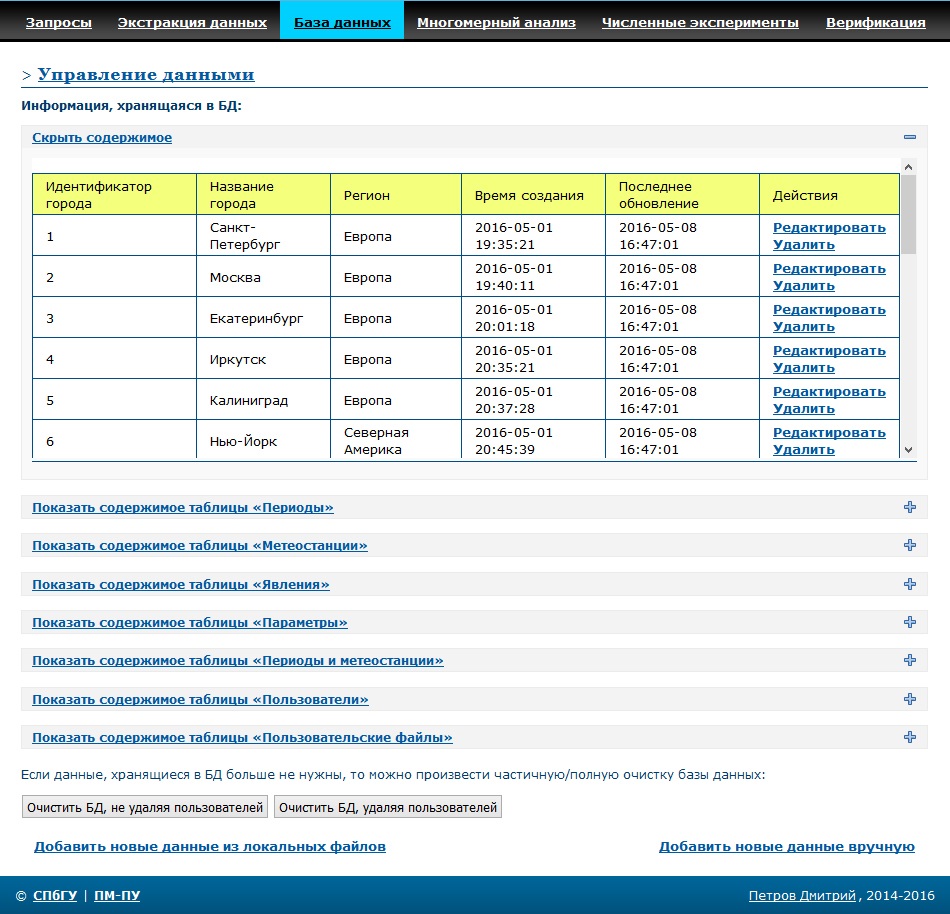


Рис. 25. Раздел «База данных»

Раздел «Многомерный анализ» позволяет проводить различные типы анализа информации, хранящейся в многомерной базе данных, с целью классификации типов опасных конвективных явлений. На выбор предоставляются стандартные встроенные алгоритмы системы Jedox Palo BI Suite и методы интеллектуального анализа данных – data mining, которыми может воспользоваться пользователь системы, чтобы принять решение о типе конвективного явления (рис. 26). Основу этих методов составляют всевозможные алгоритмы классификации, кластеризации, моделирования и прогнозирования. Также к этим методам относят статистические анализы, а именно: корреляционный и регрессионный, факторный, дисперсионный, дискриминантный и анализ временных рядов, что в совокупности является достаточно мощным инструментом для исследования, обнаружения в данных ранее неизвестных, нетривиальных знаний, необходимых для принятия решений и выявления скрытых структур зависимостей обработки как больших, так и сравнительно малых объемов данных [6].

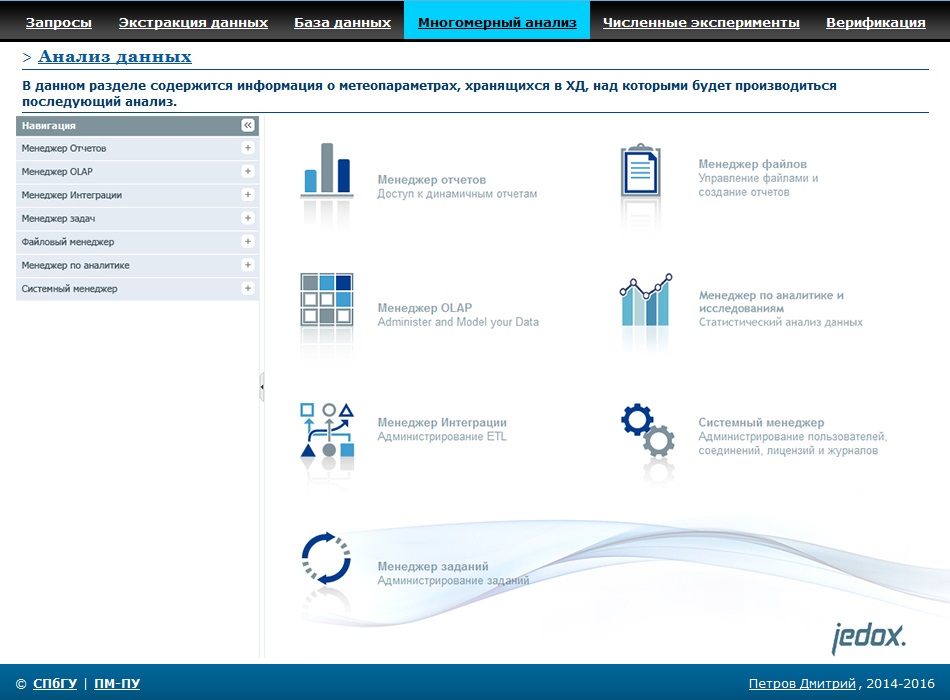


Рис. 26. Раздел «многомерный анализ»

Помимо этого, в данном разделе предусмотрена возможность манипуляций над данными, которые располагаются в хранилище данных, например, агрегация данных по измерениям, детализация: от обобщенных данным к более детальным; проекции и выборки; кросс-детализация и другие. Эти операции можно проводить как с использованием MDX (Multi-Dimensional Expressions) – языка запросов для простого и эффективного доступа к многомерным структурам данных, так и с заранее подготовленными шаблонами [25, 34, 43].

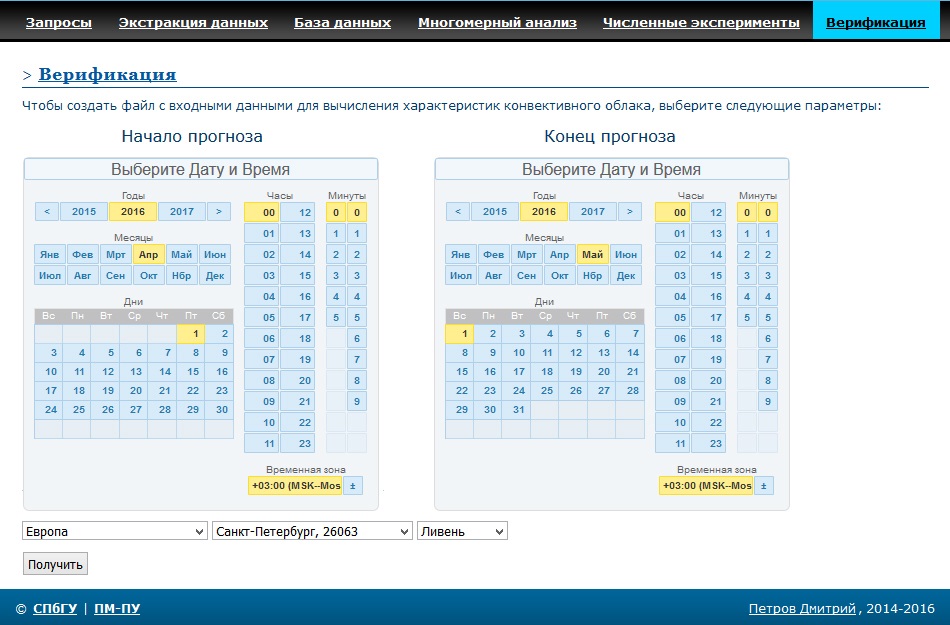
Также необходимо отметить, что в Jedox Palo BI Suite предусмотрена возможность генерации отчетов, поскольку эта система ориентирована на анализ бизнес-данных и поддержки принятия решений в бизнес-процессах. Используя данную возможность, можно представить информацию в удобочитаемом структурированном виде, который поможет пользователю в дальнейшей работе с данными.

Раздел «Численные эксперименты» демонстрирует различные случаи моделирования опасных конвективных явлений, проведенных пользователем, позволяет просматривать и редактировать содержимое файлов, в которых хранятся начальные и граничные условия для среды моделирования, а также дает возможность наглядно увидеть картину эволюции облака (рис. 27).

Рис. 27. Раздел «Численные эксперименты»



В разделе «Верификация» нам предоставляется выбор параметров, используемых в дальнейшем как входные для математической модели, которую мы верифицируем. Необходимо выбрать регион, город (метеостанцию), явление, начало и конец интересующего нас периода времени. После выбора параметров требуется кликнуть по кнопке «Получить», в результате чего система сформирует файл с параметрами и предложит сохранить его на локальную машину пользователя (рис. 28).

  
Рис. 28. Раздел «Верификация»

После этого следует загрузить сгенерированный файл в среду моделирования (рис. 29). В ней слева отображены динамические и микрофизические характеристики облака: скорость, температура, влажность и другие. В центральной области представлено поле развития облака за заданный промежуток времени. Справа расположена аэрологическая диаграмма (вертикальная стратификация атмосферы), на которой черная линия показывает температуру воздуха, красная – температуру точки росы, которая характеризует влажность в атмосфере.

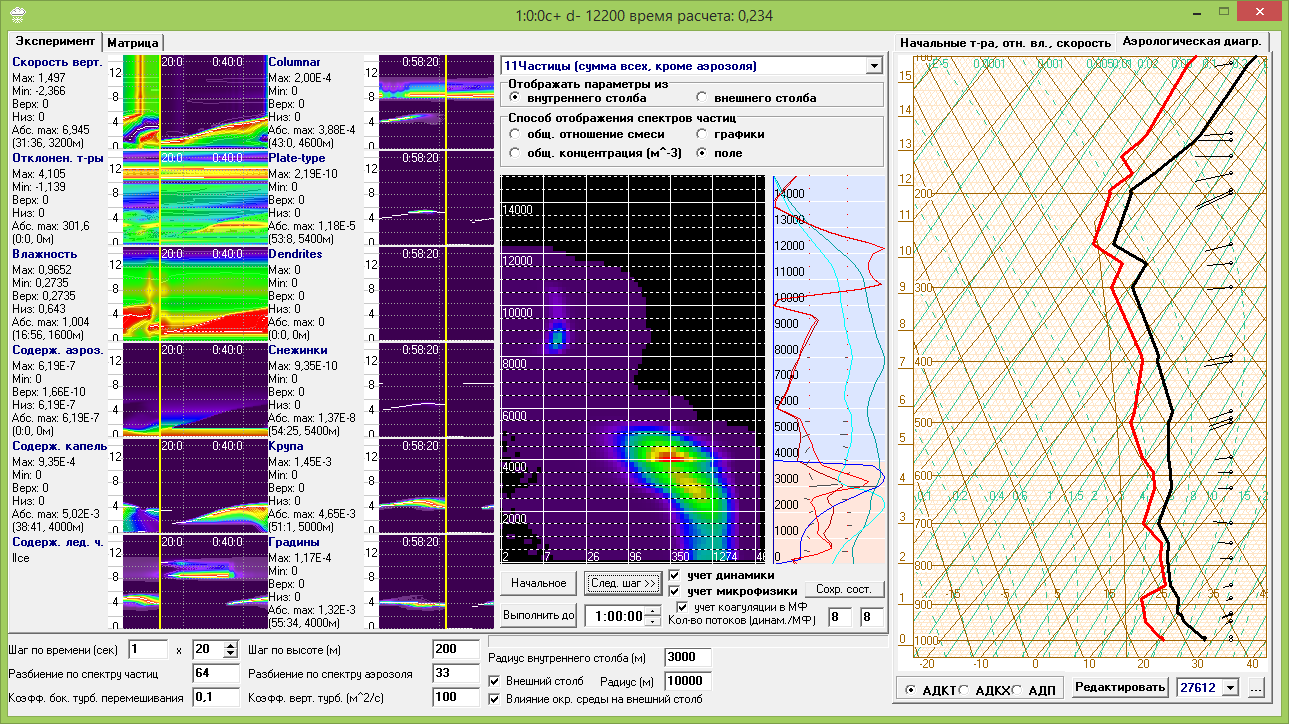


Рис. 29. Интерфейс программы расчета характеристик конвективного облака

Модель описывает процессы осадкообразования в смешанном конвективном облаке при различных вертикальных распределениях температуры и влажности в окружающей атмосфере. Она предсказывает максимальные и минимальные значения динамических и микрофизических характеристик, а также высоту верхней и нижней границы облака, интенсивность и общее количество выпавших осадков. Все эти характеристики имеют определяющее значение для прогнозирования опасных конвективных явлений, таких как, гроза, слабый дождь, шквал, град и сильный ливень.

### Выводы

На основании модели конвективного облака были сделаны выводы о начальных и граничных условиях, необходимых для ее проверки на адекватность. В соответствие с ними был проведен анализ источников метеорологической информации, из которого был сделан выбор двух из них. Это сайт «Meteocenter» и сайт Университета Вайоминга, так как они в сумме отвечают всем заявленным требованиям: присутствуют все параметры и информация предоставляется на бесплатной основе.

В результате анализа методов интеграции данных, был выбран метод консолидации. Он состоит из трех основных этапов, которые совместно именуются ETL – *Extract, Transform, Load*, что в переводе означает извлечение, преобразование, загрузка. На первом этапе данные извлекаются из гетерогенных источников и помещаются в промежуточную область, где, на втором этапе, происходит преобразование в необходимый формат, очистка от пропусков и ненужных знаков. Также данные агрегируются – становятся менее детализированными и переводятся в нужную кодировку. Если их недостаточно, некоторые из них можно восполнить путем манипуляций с уже имеющимися. И, наконец, очищаются методом, выбор которого зависит от загрязняющего фактора (например, противоречия, пропуски, дубликаты). Третий этап подразумевает загрузку данных из временных файлов в базу данных.

Нами была спроектирована и реализована реляционная база данных, соответствующая технологии консолидации и отвечающая всем ее требованиям и принципам. Она состоит из 9 отношений (таблиц), названиями для которых служат сущности, т.е. объекты реального мира. Формат информации, хранящийся в БД, не требует дальнейшего декодирования, что позволяет без дополнительных манипуляций использовать ее непосредственно для верификации и моделирования характеристик конвективного облака с помощью полуторамерной численной модели.

В результате анализа многомерных хранилищ данных было принято решение об использовании программного обеспечения Jedox Palo BI Suite, поскольку оно предоставляется на бесплатной основе и позволяет проводить обработку сложных моделей данных для статистики и принятия решений, а также способом реализации многомерного хранилища данных в нем является MOLAP, что является немало важной причиной его выбора.

Было спроектировано и реализовано многомерное хранилище данных, представленное в виде трехмерного куба, осями которого являются следующие измерения: «Дата», «Явление», «Параметры облака». Данные, содержащиеся в нем, находятся в агрегированном состоянии, это и позволяет осуществлять их оперативный анализ, быстрый поиск и выборку по заданным условиям и выполнение сложных нерегламентируемых запросов, а также использование языка MDX упрощает взаимодействие с данными, находящимися в многомерном хранилище.

Нами была разработана программная среда, которая отвечает целям, необходимым для верификации, моделирования характеристик конвективного облака с помощью полуторамерной численной модели и классификации типов опасных явлений. Данное Web-приложение включает в себя шесть разделов. Основные из них это: «Экстракция данных», «База данных», «Многомерный анализ» и «Верификация». Раздел «Экстракция данных» позволяет получить данные из гетерогенных источников, преобразовать и загрузить в БД. После этого их можно просмотреть в разделе «База данных» и осуществить манипуляции над ними. Раздел «Многомерный анализ» позволяет проводить многомерный статистический анализ с целью выявления диапазонов значений характерных параметров конвективных облаков для классификации типов конвективных явлений. В разделе «Верификация» можно подготовить и сохранить файл для дальнейшего использования его в среде моделирования с целью верификации физико-математической модели опасного облака.

Программная среда была протестирована с помощью программно-ориентированной системы тестирования «PHPUnit» и найденные ошибки и недочеты были исправлены.

На примере полуторамерной модели конвективного облака, разработанной Станковой Е. Н. и Раба Н. О., был произведен ряд тестовых численных экспериментов, направленных на верификацию этой модели. В результате она повела себя корректно, что означает адекватность модели.

Используя разработанную комплексную информационную систему, был выявлен диапазон изменения характерных параметров облака, а также их средние значения. На основании полученных результатов были даны рекомендации по прогнозу гроз с помощью численной модели.

Дальнейшее развитие прототипа системы предполагает:

* интеграцию метеоданных, передаваемых по защищённым каналам, что даст возможность создать высокоточную, качественную систему;
* добавление новых информационных источников;
* переход от полуторамерной модели к двухмерной и трехмерной;
* для уточнения прогнозов типов конвективных явлений можно будет расширить возможности использования многомерной базы данных за счет более широкого использования инструментов статистического анализа.

### Заключение

В данной выпускной квалификационной работе была поставлена задача разработать программный продукт, предназначенный для интеграции и анализа специализированной метеорологической информации.

Для решения поставленной задачи была выбрана интеграция данных по методу консолидации. Для извлечения, преобразования и загрузки данных применялся язык серверного программирования PHP. Быстрое извлечение необходимых данных было достигнуто использованием техники многопоточности, поддерживающейся функциями библиотеки libcurl.

Для хранения данных, используя среду Oracle MySQL Workbench, была разработана реляционная база данных. Она состоит из 9 отношений, в которых располагается вся интересующая нас специализированная метеорологическая информация. Манипуляции с данными осуществляются с помощью языка запросов SQL.

Для оперативного анализа специализированной метеорологической информации, используя продукт Jedox Palo BI Suite, было разработано многомерное хранилище данных, представленное в виде куба данных, содержащего информацию о параметрах конвективного облака, полученных с помощью серии численных экспериментов. Манипуляции над данными осуществляются с помощью языка запросов MDX.

Воспользоваться информацией из баз данных пользователь может с помощью разработанного нами Web-приложения. Оно реализовано в среде PhpStorm с помощью следующего инструментария: PHP, JavaScript, CSS, HTML. Приложение имеет интуитивно понятный интерфейс и состоит из шести разделов, основными из которых являются: «Экстракция данных», «База данных», «Многомерный анализ» и «Верификация».

Итак, основная задача была решена следующим образом. Был разработан прототип программного продукта, выполненный в виде Web-приложения, который позволяет интегрировать данные с помощью технологии консолидации, размещать их в реляционной базе данных, проводить серию численных экспериментов и классифицировать по типам опасные конвективные явления, такие как грозы, грады, шквалы и обильные осадки.

Используя разработанный прототип из гетерогенных источников были интегрированы данные, которые послужили входными начальными и граничными условиями для нестационарной полуторамерной модели конвективного облака. В результате верификации она повела себя корректно, что дает основания полагать, что модель адекватна, а это, в свою очередь, значит, что результаты экспериментов с моделью позволяют получить статистически обоснованные данные об эффективности использования облачной модели для прогнозирования опасных конвективных явлений.

Результаты работы были обобщены в докладах «Use of consolidation technology for meteorological data processing», «Integrated Information System for Verification of the Models of Convective Clouds», «Using Technologies of OLAP and Machine Learning for Validation of the Numerical Models of Convective Clouds», которые были после «слепого» рецензирования приняты к представлению на международных конференциях по компьютерным наукам и приложениям (ICCSA’14, ICCSA’15, ICCSA’16). Доклады в виде статей опубликованы в журнале Lecture Notes in Computer Science (издательство Springer). Статья «Комплексная информационная система, предназначенная для формирования входных данных моделей конвективных облаков» (после «слепого» рецензирования) была опубликована в журнале «Вестник Санкт-Петербургского университета» Серия 10, 2015.

### Список использованных источников

**Статья в журнале**

1. Белошицкий Д. А. Интеграция данных в информационных системах // Молодежный научно-технический вестник. Изд-во: ФГБОУ ВПО «МГТУ им. Н. Э. Баумана», эл. No. ФС77-51038, 2013. С. 1-17.
2. Кодд Е. Ф. Реляционная модель данных для больших совместно используемых банков данных // Системы Управления Базами Данных # 1/1995, Изд. дом «Открытые системы» / пер. с англ. Когаловский М. Р. / под ред. Сергея Кузнецова, 2009. С. 8-16.
3. Раба Н.О., Станкова Е.Н. Исследование влияния компенсирующего нисходящего потока на жизненный цикл конвективного облака с помощью численной полуторамерной модели с двумя цилиндрами // Труды Главной геофизической обсерватории им. А. И. Воейкова. – 2009. Вып. 559. С. 192-209.
4. Станкова Е. Н., Петров Д. А. Комплексная информационная система, предназначенная для формирования входных данных моделей конвективных облаков // «Вестник Санкт-Петербургского университета». Прикладная математика. Информатика. Процессы управления. Серия 10. 2015 Выпуск 3. Стр. 83-95.
5. Dr. Osama E. Sheta, Ahmed Nour Eldeen. The Technology of Using a Data Warehouse to Support Decision-Making in Health Care. International Journal of Database Management Systems (IJDMS). Vol. 5, No. 3, 2013. P. 75-86. DOI: 10.5121/ijdms.2013.5305
6. Han J. OLAP Mining: An Integration of OLAP with Data Mining. S. Spaccapietra et al. (Eds.): IFIP TC2 WG2.6 IFIP Seventh Conference on Data Semantics 1998, Data Mining and Reverse Engineering. Searching for semantics, 1998. P. 1-18. DOI 10.1007/978-0-387-35300-5\_1 http://link.springer.com/chapter/10.1007/978-0-387-35300-5\_1
7. Khain A., Ovtchinnikov M., Pinsky M., Pokrovsky A., Krugliak H. Notes on the state-of-the-art numerical modeling of cloud microphysics Review // Atmospheric Research. 2000. Vol. 55. P. 159-224.
8. Petrov D. A., Stankova E. N. "Use of Consolidation Technology for Meteorological Data Processing" B. Murgante et al. (Eds.): ICCSA 2014, Part I, LNCS 8579. P. 440-451. Springer International Publishing Switzerland (2014).
9. Petrov D. A., Stankova E. N. Integrated Information System for Verification of the Models of Convective Clouds. O. Gervasi et al. (Eds.): ICCSA 2015, Part IV, LNCS 9158, pp. 321–330, 2015. DOI: 10.1007/978-3-319-21410-8\_25. http://link.springer.com/chapter/10.1007/978-3-319-21410-8\_25
10. Raba N. O., Stankova E. N., Ampilova N. On Investigation of Parallelization Effectiveness with the Help of Multi-core Processors // Procedia Computer Science. 2010. Vol. 1, Issue 1. P. 2757-2762.
11. Raba N. O., Stankova E. N. On the Possibilities of Multi-Core Processor Use for Real-Time Forecast of Dangerous Convective Phenomena // Lecture Notes in Computer Science. Computational Science and Its Applications – ICCSA 2010. 2010. Vol. 6017. P. 130-138.
12. Raba N. O., Stankova E. N. On the Problem of Numerical Modeling of Dangerous Convective Phenomena: Possibilities of Real-Time Forecast with the Help of Multi-core Processors // Lecture Notes in Computer Science. Computational Science and Its Applications – ICCSA 2011. 2011. Vol. 6786. P. 633-642.
13. Raba N. O., Stankova E. N. On the Effectiveness of Using the GPU for Numerical Solution of Stochastic Collection Equation, 2013, 'On the Effectiveness of Using the GPU for Numerical Solution of Stochastic Collection Equation', ICCSA 2013, Lecture Notes in Computer Science, Part V, 7975. P. 248-258. DOI 10.1007/978-3-642-39640-3\_18 http://link.springer.com/chapter/10.1007/978-3-642-39640-3\_18

**Книга одного автора**

1. Дейт К. Дж. Введение в системы баз данных. – 8-е изд. / пер. с англ. Птицына К. А. М.: Вильямс, 2005. С. 99-100, 163-346, 349-359, 457-466.
2. Ершова Г. Н. Информационные технологии в книжном деле // Учебное пособие. М.: МГУП, 2002. С. 17-29.
3. Ковешников М. Г. Выпускная квалификационная работа бакалавра «Автоматизация загрузки больших массивов данных предметной области в промышленную». Новосибирск: НГУ, 2013. С. 9-10, 16-17.
4. Козлов А. Н. Интеллектуальные информационные системы // учебник / А. Н. Козлов; Мин-во с-х. РФ, ФГБОУ ВПО Пермская ГСХА. – Пермь: Изд-во ФГБОУ ВПО Пермская ГСХА, 2013. С. 137-161.
5. Петров Д. А. Выпускная квалификационная работа бакалавра «Использование технологии консолидации для разработки системы сбора и анализа специализированной метеорологической информации». СПб.: СПбГУ, 2014. С. 1-59.
6. Скакун В. В. Системы управления базами данных // Учебное пособие. Минск, 2006. С. 10-11.
7. Ташков П. А. Веб-мастеринг на 100%. HTML, CSS, JavaScript, PHP, CMS, AJAX, раскрутка. СПб.: Питер, 2010. С. 14-261.
8. Шкрыль А. А. PHP – это просто. Программирование для Web-сайта. СПб.: БХВ-Петербург, 2006. С. 16.
9. Шлосснейгл Дж. Профессиональное программирование на PHP / пер. с. англ. Швец В. М.: Вильямс, 2006. С. 49-82.
10. DuBoisP. MySQL. – 2-еизд. / пер. с. англ. Воронина Н.В. М.: Вильямс, 2004. С. 113-155, 264-270.
11. Khalid M. I. PHP/CURL Book. 2009. P. 10-17.
12. Scott C. Microsoft SQL Server 2008. Analysis Services Step by Step. – 1st ed., Microsoft Press. – 2009, P. 14-24, 150-155.

**Книга нескольких авторов**

1. Барсегян А. А., Куприянов М. С., Степаненко В. В., Холод И. И. Методы и модели анализа данных: OLAP и Data Mining. СПб.: БХВ-Петербург, 2004. С. 27-44, 49-66.
2. Бибо Б., Кац И. jQuery. Подробное руководство по продвинутому JavaScript. – 2-е изд. / пер. с. англ. Киселев А. СПб.: Символ-плюс, 2011. С. 499-512, 522-541.
3. Богданов А. В., Станкова Е. Н., Тхурейн Кива Лин. Распределенные базы данных. СПб.: СПбГЭТУ «ЛЭТИ», 2013. С. 39-43.
4. Веллинг Л., Томсон Л. Разработка Web-приложений с помощью PHP и MySQL. – 3-е изд. / пер. с англ. – М.: Изд. Вильямс, 2008. С. 283-300.
5. Довгалюк Ю. А., Веремей Н. Е., Синькевич А. А. Применение полуторамерной модели для решения фундаментальных и прикладных задач физики облаков. СПб, Гидрометеоиздат, 2007. С. 32-43.
6. Евсеева О. Н., Шамшев А. Б. Работа с базами данных на языке C#. Технология ADO.NET // Учебное пособие. УлГТУ, 2009. С. 7-15.
7. Котеров Д., Костарев А. PHP 5. – 2-е изд. СПб.: БХВ-Петербург, 2008. С. 354-361.
8. Паклин Н. Б., Орешков В. И. Бизнес-аналитика: от данных к знаниям // Учебное пособие. СПб.: Питер, 2013. С. 61-137.
9. Харинатх С., Куинн С. SQL Server. Analysis Services 2005 и MDX для профессионалов. М.: Вильямс, 2008. С. 84-113, 137-209.
10. Date C. J., Darwen H. Foundation for Future Database Systems: The Third Manifesto. – 2nd ed. – Addison Wesley Professional, 2000. P. 223-238.

**Книга под редакцией**

1. Коннолли Т., Бегг К. Базы данных: проектирование, реализация и сопровождение. – 3-е изд. / пер. с англ. Имамутдинова Р. Г., Птицына К. А. // под ред. Птицына К. А. М.: Вильямс, 2003. С. 460-472.

**Статья в сборнике**

1. Станкова Е. Н., Раба Н. О., Ампилова Н. Б. Численное моделирование микрофизических процессов в смешанных конвективных облаках. Сравнение результатов расчётов с данными натурных экспериментов // Научная конференция институтов Росгидромета «Теоретические и экспериментальные исследования конвективных облаков». – СПб., 2008. – С. 90-91.

**Ссылка на документ в интернете**

1. Архив погоды в РФ и мире. http://meteocenter.net/ussr\_fact.htm
2. Введение в основы OLAP-технологий и MS SQL Server Analysis Services 2008. http://www.intuit.ru/studies/courses/568/424/lecture/9641
3. Введение в OLAP-технологии и многомерные базы данных. http://www.olap.ru/basic/alpero2i.asp
4. Замараев В. Типы программных продуктов // Software&Startups. http://www.softwareandstartups.com/razrabotka-po/tipy-programmnyx-produktov/
5. Информационная безопасность и тестирование Web-приложений. http://forworktests.blogspot.ru/2013/03/web\_3.html?m=1
6. Калюжный О. Н. Язык запросов MDX. http://sqlmdx.narod.ru/
7. Клиентская библиотека работы с URL (cURL). http://www.php.net/manual/ru/book.curl.php
8. Консолидация данных – ключевые понятия // Корпоративный менеджмент. http://www.cfin.ru/itm/olap/cons.shtml
9. Многомерные базы данных (Multi-value Database). http://www.gazintech.ru/multivaluedb.php
10. Написание автоматических тестов и среда PHPUnit. http://phpclub.ru/detail/article/phpunit
11. Подходы к контролю качества данных. http://fsecrets.ru/2012/08/подходы-к-контролю-качества-данных
12. Принципы проектирования и использования многомерных баз данных. http://www.osp.ru/dbms/1996/03/13031490/
13. Сайт Университета Вайоминга. Кафедра атмосферных наук. http://weather.uwyo.edu/upperair/sounding.html
14. СУБД Oracle MySQL. https://www.mysql.com/
15. Технология многомерных баз данных. http://www.osp.ru/os/2002/01/180958/
16. Фреймворк PHPUnit для начинающих. http://phpprofi.ru/blogs/post/24
17. Язык программирования PHP. https://secure.php.net/
18. Advanced Monitoring Methods. High Quality Data Collection from any Location. http://advm2.com
19. Codd E. F., Codd S. B., Salley C. T. Providing OLAP to User-Analysts: An IT Mandate // White Paper, Codd & Date. 1993.
20. Institute for Radar Meteorology (IRAM). http://www.iram.ru/iram/index\_ru.php
21. Integrated Terminal Weather System (ITWS). https://www.ll.mit.edu/mission/aviation/faawxsystems/itws.html
22. Jedox AG Palo BI Suite. http://www.jedox.com/en/
23. Jedox AG Palo OLAP Server v4.0 // Manual, 2013. https://download.jedox.com/archive/files/manuals/4\_0/jedox\_olap\_server\_en\_40.pdf
24. JetBrains PhpStorm. https://www.jetbrains.com/phpstorm/
25. Meteorological Assimilation Data Ingest System (MADIS). https://madis.noaa.gov/
26. The Testing Framework PHPUnit. https://phpunit.de/