

**Частное учреждение профессионального образования**

**«Высшая школа предпринимательства»**

**(ЧУПО «ВШП»)**

КУРСОВОЙ ПРОЕКТ

«Разработка базы данных для метеорологической службы »

Выполнил:

студент 3-го курса специальности

09.02.07 «Информационные системы и программирование»  
Щербакова Маргарита Евгеньевна

подпись: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Проверил:

проректор по информатизации,  
зав. каф. ИСиТ при ЧУВО «ВШП»,  
к.ф.н. Ткачев П.С.

оценка: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

подпись: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Содержание

[Введение. 3](#__RefHeading___Toc1807_1528580707)

[Глава 1. Теоретическая часть. 5](#__RefHeading___Toc1809_1528580707)

[Параграф 1. Метеослужбы и анализ их баз данных. 5](#__RefHeading___Toc1915_1528580707)

[Параграф 2. Анализ существующих способов разработки баз данных. 7](#__RefHeading___Toc1917_1528580707)

[Параграф 3. Цели, задачи и основные характеристики разрабатываемой базы данных. 10](#__RefHeading___Toc1919_1528580707)

[Глава 2. Практическая часть. 12](#__RefHeading___Toc1921_1528580707)

[Параграф 1. База данных и роли. 12](#__RefHeading___Toc1923_1528580707)

[Параграф 2. Структура и таблица базы данных. 12](#__RefHeading___Toc1925_1528580707)

[Параграф 3. Типовые запросы, представления и процедуры. 15](#__RefHeading___Toc1927_1528580707)

[Заключение. 19](#__RefHeading___Toc1929_1528580707)

[Литература. 21](#__RefHeading___Toc1931_1528580707)

[Приложение 22](#__RefHeading___Toc1933_1528580707)

## Введение.

Моя работа направлена на анализ существующих баз данных, направленных на сбор и систематизацию метеорологических показаний на метеостанциях, анализ их функционала и возможностей применения, а также на разработку собственной базы данных.

Актуальность этой темы связана с тем, что зафиксирован ряд значительных изменений климата по всей территории нашей планеты, а также важным курсом для всего человечества стало сохранение окружающей среды. Сейчас люди обеспокоены вопросом происходящих климатический изменений, так как это неизменно влияет на многие сферы жизни людей. Погода играет важную роль в сельскохозяйственной промышленности. Прогнозирование погоды важно для борьбы с лесными пожарами в засушливый сезон, так как своевременная подготовка и профилактические меры могут заметно снизить ущерб в случаях возникновения чрезвычайных ситуаций. Состояние погоды в курортных городах в сезон влияет на туристический потенциал стран. Изменение климата может привести к исчезновению ряда экосистем и вымиранию отдельных видов животных и растений.

Наблюдение за погодой в течение значительных промежутков времени позволяет выявить тенденции в изменении климата, проанализировать их и предложить решение для уменьшения негативных влияний климатических изменений и адаптации к ним. Именно потому столь важно отслеживать изменения погоды, это необходимо для выживания человечества. Использование баз данных для систематизации информации о погоде позволяет оперативно анализировать актуальные данные и действовать в соответствии со сделанными выводами. Именно поэтому мы выбрали данную тему.

Разработанность. На данный момент существует ряд компаний, которые занимаются сбором данных по погоде. Наиболее известные из них GISMETEO, Яндекс Погода и другие. А также существует несколько организаций, которым принадлежат свои метеостанции, такие как ЦЕНТРАЛЬНОЕ УГМС, ВЕРХНЕВОЛЖСКОЕ УГМС, СЕВЕРО-ЗАПАДНОЕ УГМС. У каждой из этих компаний есть своя база данных. Большая часть из них являются закрытыми, по этой причине нет возможности провести полный и подробный их анализ, однако, мы рассмотрим ту их часть, которая есть в открытом доступе. Базы данных этих организаций разнообразны, и каждая имеет свои задачи и функционал, так как невозможно создать базу данных одинаково хорошо подходящую под все запросы.

Объектом исследования моей курсовой работы выступают базы данных метеорологических станций и метеорологических компаний.

Целью моей курсовой работы было проанализировать, как устроены существующие базы данных метеорологических служб, и на их основе разработать свою базу данных для метеослужбы.

Задачи моей работы:

1. Проанализировать существующие базы данных в области метеорологии и определить конкретные запросы для базы данных метеорологической службы, которые могут стать наиболее актуальными в ближайшие годы.
2. Проанализировать существующие способы разработки баз данных.
3. Разработать свою СУБД на основе существующих, а также добавить разные роли пользователей для этой базы данных.

При создании своей курсовой работе я пользовалась следующими методами исследования:

1. метод сравнительного анализа,
2. метод классификации,
3. метод изучения и анализа литературы,
4. метод синтеза.

## Глава 1. Теоретическая часть.

## Параграф 1. Метеослужбы и анализ их баз данных.

Согласно определению **Большой российской энциклопедии,** метеорологическая служба - это общее название специальных государственных организаций, в состав которых входят сети метеорологических, аэрологических и других специализированных станций, оперативные и научные метеорологические учреждения. Задачами метеорологических станций являются развитие научных исследований атмосферы, обеспечение народного хозяйства и населения информацией о климате, погоде, загрязнениях атмосферы, составление и распространение прогнозов погоды и её опасных явлений.

Метеослужбы бывают разные и, как правило, они связаны с государством из-за чего их базы данных являются приватными. По этой причине их анализ ограничен только той информацией, которая в открытом доступе. Например, на сайте2 можно получить информацию о существующих метеорологических станциях и их специальных задачах. На основе этих данных можно предположить функционал и особенности баз данных, использующихся в работе этих метеорологических служб.

Тем не менее, из-за отсутствия возможности изучения непосредственно существующих баз данных метеорологических служб, моя база данных может отличаться от них по структуре.

Проанализировав открытые источники, мы можем заключить следующее:

1. У каждой метеостанции своя база данных, которая соединена с подобными базами данных для получения максимально широкого круга данных, собранных по категориям оборудования с показателями. Таким образом, создается огромная архитектура баз данных с огромным массивом данных, которые с помощью определенных запросов выдают собранную информацию, которую позднее анализируют специалисты.
2. Для создания таких сложных архитектур необходим многолетний труд специалистов и ввиду закономерного появления новых станций и более точных технологий для сбора информации, требуется постоянный надзор за работой этой структуры, а также работа над ее развитием. Мы предполагаем, что над этой структурой ведется постоянный надзор, анализ и обслуживание в виде прописывания функций для более легкого анализа данных.
3. Метеостанции могут иметь свой идентификатор локации, который используется в полноценных базах данных как составной ключ, который связывает их с другими таблицами.

При разработке собственной базы данных, я решила ее сделать куда более локальной, чем другие подобные структуры из-за того, что так будет легче отслеживать изменения погоды в конкретном регионе и это упростит аналитику данных. В будущем при развитии моей базы данных и локализацией на конкретный легион большая общественность сможет самостоятельно отслеживать изменение климата и погоды и самостоятельно делать определенные выводы из полученных данных.

При разработке собственной базы данных я основываюсь на поиске наиболее практичного решения задачи сбора информации о погоде. Именно поэтому при разработке я создала пять конкретных таблиц, о которых в дальнейшем будет рассказано подробно. Все пять таблиц имеют между собой связь, создавая архитектуру. В моем случае архитектура весьма простая, но также она является лаконичной, а главное – работающей, что дает ей преимущество. Данная простота обусловлена тем, что такое решение имеет практическое применение, суть it направления заключается в том, чтобы облегчить жизнь людям и стремится к максимальному функционалу, при этом обладая простым управлением. В данном случае простота нужна для более простого управления, так как при развитии моя база данных нацелена на открытый доступ общественности, чтобы все могли не только увидеть, но и вносить данные по изменениям погоды и климата.

## Параграф 2. Анализ существующих способов разработки баз данных.

Прежде чем начать разрабатывать базу данных нужно разобраться, какие существуют базы данных и какие больше из них подходят для решения поставленной задачи.

Существуют разные нормальные формы баз данных. Всего их шесть, но также существуют дополнительные нормальные формы. Прежде чем рассматривать каждую из них, нужно дать определение нормальной форме баз данных.

Нормальная форма баз данных - это свойство отношения в [реляционной](https://ru.wikipedia.org/wiki/Реляционная_модель_данных) модели, определяющей её с точки зрения избыточности, потенциально приводящей к логически ошибочным результатам выборки или изменения данных. Нормальная форма определяется как совокупность требований, которым должно удовлетворять отношение.

Существует шесть нормальных форм баз данных и три дополнительных формы:

1. Нулевая форма по факту не является базой данных в соответствии с вышеприведенным определением. По большому счету, это двухмерный массив данных.
2. Первая нормальная форма базы данных соответствует этой нормальной форме, только когда она находится в нулевой нормальной форме и когда соответствует реляционной модели данных.
3. Вторая нормальная форма - это база данных, которая соответствует этой нормальной форме, когда она была в первой нормальной форме и таблицы имеют ключ.
4. Третья нормальная форма - это база данных, которая находится во второй нормальной форме при отсутствии транзитной зависимости в таблицах.
5. Нормальная форма Бойса Кодда - это переходная форма между третьей и четвертой нормальной формами. Требования для базы данных в этой нормальной форме заключаются в том, чтобы она была в третьей нормальной форме, и чтобы ключевые атрибуты составного ключа не зависели от не ключевых атрибутов.
6. Четвертая нормальная форма база данных предполагает, чтобы она соответствовала третьей нормальной форме, и чтобы в таблицах не было нетривиальных многозадачных зависимостей.
7. Пятая нормальная форма база данных соответствует этой нормальной форме, когда она находится в четвертой нормальной форме, и когда все нетривиальные зависимости соединений определяются ключом этого отношения. Также стоит отметить, что, как правило, эта нормальная форма является последней в разбиении таблиц на проекции и их соединения.
8. Доменно - это ключевая нормальная форма. Про нее следует отметить только то, что эта форма уже является специфичной, и поэтому она достаточно редко фигурирует в реальных проектах без особого уклона.
9. Шестая нормальная форма - эта форма также весьма специфичная, она используется в работе с хронологическими базами данных.

Я выбрала третью нормальную форму для разработки своей базы данных, так как она является оптимальной для моих задач, также она является самой популярной нормальной формой при создании проектов, в то время, когда все предыдущие нормальные формы являются не совсем полноценными базами данных, а все последующие являются более специфическими и нужны для особых задач.

Создавать базу данных можно на разных платформах. Среди них можно выделить такие, как Podgres и MySQL. У этих платформ есть свои плюсы и минусы. PostgreSQL — это реляционная база данных с открытым кодом, которая поддерживается в течение 30 лет разработки и является одной из наиболее известных среди всех существующих реляционных баз данных.

Популярностью у разработчиков и администраторов база данных PostgreSQL обязана своей исключительной гибкости и целостности. Например, база данных PostgreSQL поддерживает как реляционные, так и нереляционные запросы.

MySQL является решением для малых и средних приложений. Обычно MySQL используется в качестве сервера, к которому обращаются локальные или удалённые клиенты, однако в дистрибутив входит библиотека внутреннего сервера, позволяющая включать MySQL в автономные программы.

Гибкость СУБД MySQL обеспечивается поддержкой большого количества типов таблиц: пользователи могут выбрать как таблицы типа [MyISAM](https://ru.wikipedia.org/wiki/MyISAM), поддерживающие полнотекстовый поиск, так и таблицы [InnoDB](https://ru.wikipedia.org/wiki/InnoDB), поддерживающие транзакции на уровне отдельных записей. Более того, СУБД MySQL поставляется со специальным типом таблиц EXAMPLE, демонстрирующим принципы создания новых типов таблиц. Благодаря открытой архитектуре и GPL-лицензированию, в СУБД MySQL постоянно появляются новые типы таблиц.

Для разработки своей базы данных мы будем использовать MySQL. Эта платформа отличается простотой графического интерфейса платформы и тем, что не обязательно прописывать запросы создания баз данных таблиц и заполнения данных, что хорошо подходит под наши задачи. Тем не менее, при создании базы данных метеорологической службы мы будем все же прописывать эти запросы для сохранения знания синтаксиса MySQL.

## Параграф 3. Цели, задачи и основные характеристики разрабатываемой базы данных.

Итак, мы рассмотрели, что такое метеослужбы и какие бывают базы данных этих служб. На основе этого мы можем сформулировать, какой должна быть разрабатываемая нами база данных для метеослужбы.

Для начала следует оговорить, какие задачи стоят перед нашей базой данных. Основной задачей выступает сбор и анализ данных с метеостанций, а также выявление тенденций изменения климата, отслеживание этих изменений.

Так как в открытом доступе не так много информации по существующим базам данных метеослужб, для разработки нашей мы руководствовались соображениями удобства использования базы данных, а также ее целями и задачами. По результатам рассмотрения разных вариантов нормальных форм баз данных мы решили создавать базу данных третьей нормальной формы, потому что для решения поставленной задачи она является оптимальной.

В качестве платформы для разработки мы выбрали MySQL, так как она является оптимальной платформой для этих задач в виду наиболее рационального сочетания достаточной функциональности и простого интерфейса, упрощенного варианта создания базы данных ерд диаграммы с таблицами и упрощенным вариантом заполнения таблиц данными.

Таким образом, мы определили, что наша база данных предназначена для сбора данных с определенных метеостанций. Собранные данные будут группироваться по датам. Это позволит широкому кругу пользователей самостоятельно следить за климатическими изменениями в выбранном регионе, так как при дальнейшей разработке база данных будет выкладываться в открытый доступ.

Цель создания базы данных для метеорологических служб была проанализирована и разбита на конкретные задачи.

Задачами выступают:

1. проектирование базы данных,
2. создание таблиц,
3. заполнение таблиц подходящими данными,
4. создание хранимых процедур,
5. создание представления,
6. прописывание типовых запросов для структурирования данных в таблицах,
7. создание особых ролей для базы данных.

## Глава 2. Практическая часть.

## Параграф 1. База данных и роли.

В самом начале нужно было создать базу данных.

create database `cyrs`;  
use cyrs;

Тут показано, что я прописала создание базы данных `cyrs` и сделала так, чтобы все запросы отходили к ней.

Далее были использованы роли:

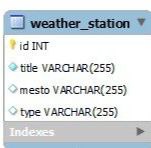
Администратор - у него были все права на базу данных, и он мог создавать, обращаться и удалять как целиком базу данных, так и конкретные таблицы.

Ученый - он был уже ограничен в правах, он не мог создавать таблицы и не мог удалять их, однако конкретные строки в таблицах он мог редактировать.

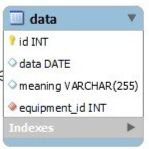
## Параграф 2. Структура и таблица базы данных.

Затем я прописала создание таблиц. В моей базе данных есть пять таблиц weather\_station, data, equipment, dimension, type\_equipment, каждая из них выполняют свои функции.

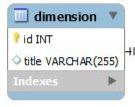
Weather\_station хранит в себе столбцы title с типом данных varchar(255) с названием метеостанции mesto с типом данных varchar(255) с расположением и type с типом данных varchar(255) с названием управляющей компании.



Data хранит в себе data с типом данных data, с датой получения данных meaning, со значением показателей определенного инструмента и equipment\_id, который является внешним ключом и соединяет с таблицей equipment.



Dimension является таблицей с типом оборудования метеостанций, обладает столбцом title c информацией.



Equipment хранит в себе внешние ключи из таблиц weather\_station и equipment, показывая какое конкретно оборудование есть на конкретных метеостанциях.



type\_equipment хранит в себе title, в котором дается четкое название оборудованию, и dimension\_id, который также является внешним ключом и связывает с таблицей dimension.



Код того, как были прописаны таблицы: см. приложение.

Выбор типов данных не случаен. Многие названия и типы данных повторяются.

Структура показана на ерд диаграмме, увидеть ее можно в приложении Рисунок 1. Там показаны связи между таблицами, и четко видно, что везде используются связи один ко многим, но из-за не самой стандартной структуры таблица ----- может восприниматься как переходная между --- и --- при связи «многие ко многим». Данное решение по проектированию я приняла, потому что мне посчитала это наиболее выигрышным вариантом.

Существуют связи между weather\_station и equipment, также таблица equipment связана с type\_equipment. Это сделано для того, чтобы показать, какое конкретное оборудование есть на конкретных метеостанциях. Существует связь между data и equipment. Это сделано для того, чтобы в таблице data было показано, в какой день были сняты показатели с конкретного оборудования на конкретной метеостанции. Есть связь между dimension и type\_equipment. Она сделана так, как таблица dimension - показывает конкретный тип оборудования по типу, измеряет он скорость, температуру и т.д. И таким образом, из таблицы dimension данные типа «оборудование» перетекают в таблицу type\_equipment, в которой хранятся данные о конкретном оборудовании.

Заполнение таблиц происходило через оператор INSERT INTO.

Пример:

INSERT INTO `project`.`weather\_station` (`title`, `mesto`, `type`) VALUES ('ТОРЖОК', ' Р.ТВЕРЦА', 'ЦЕНТРАЛЬНОЕ УГМС');  
INSERT INTO `project`.`weather\_station` (`title`, `mesto`, `type`) VALUES ('СТАРИЦА', ' P.BOЛГA', 'ЦЕНТРАЛЬНОЕ УГМС');

## Параграф 3. Типовые запросы, представления и процедуры.

Мной было создано пять типовых запросов.

SELECT   
avg(meaning)  
FROM cyrs.data;

Этот запрос показывает среднее значение столбца meaning из таблицы data.

SELECT count(\*) FROM cyrs.weather\_station  
where mesto like '%ОЗ.%';

Этот запрос показывает количество метеостанций, которые базируются на озере.

SELECT   
data.meaning,  
equipment.type\_equipment\_id,  
type\_equipment.title  
FROM cyrs.data  
join equipment  
on equipment.id = equipment\_id  
join type\_equipment  
on type\_equipment.id = type\_equipment\_id;

Этот запрос отображает показатели столбцов meaning из таблицы data, type\_equipment\_id из таблицы equipment и title из таблицы type\_equipmen. Такой запрос работает благодаря связям между таблицами и с помощью join.

SELECT \* FROM cyrs.data  
where data like '%2021%'  
;

Этот запрос показывает данные, полученные за 2021 год.

SELECT   
type\_equipment.title,  
type\_equipment.dimension\_id,  
dimension.title  
FROM cyrs.type\_equipment  
 join dimension  
 on dimension.id = dimension\_id

where dimension.title = 'гидрологический';

Этот запрос также построен на связях между таблицами. Тут показан запрос, который связывает таблицу type\_equipment и dimension, отображая столбцы title dimension\_id и title из таблицы dimension с условием, что значение dimension.title должно быть равно «гидрологический».

Далее идет представление.

create view cyrs.station\_equpment AS   
select  
 e.weather\_station\_id  
 ,we.title AS station\_title  
 ,we.mesto  
 ,we.type  
 ,e.type\_equipment\_id  
 ,d.title AS type\_equipment\_title  
FROM  
cyrs.equipment AS e  
 JOIN cyrs.weather\_station AS we ON (e.weather\_station\_id=we.id)   
 JOIN cyrs.type\_equipment AS d ON (e.type\_equipment\_id=d.id)  
 ;

Суть представлений в том, чтобы создавать так называемые виртуальные таблицы. Данное представление объединяет таблицы weather\_station и type\_equipment, чтобы конкретно показать, какое именно оборудование есть на определенных метеостанциях.

Процедур я сделала две. Одна из них по добавлению новой метеостанции в таблицу.

use cyrs;  
DELIMITER $$  
create DEFINER=`root`@`localhost` PROCEDURE weather\_station\_add(IN \_title varchar(255), in \_mesto varchar(255), in \_type varchar(255))  
begin   
 INSERT INTO cyrs.weather\_station (title,mesto,type) VALUES (\_title,\_mesto,\_type);  
END$$  
DELIMITER ;

В начале прописывается, что запрос идет к конкретной базе данных.

Потом используется delimiter. Эта часть помогает обработчику не обращать внимание на знаки препинания в процедуре. Далее прописывается создание процедуры, в котором прописываются переменные, которые обозначают каждый столбец и прописывается, что при обращении к данной процедуре и при заполнении переменных определенными данными, эти самые определенные данные заполняют строки в таблице при помощи оператора INSERT.

Вторая процедура по удалению неактуальных метеостанций из таблицы. cyrs.

use cyrs;  
delimiter //  
create procedure cyrs.weather\_station\_delete( in \_id int)  
begin   
 DELETE FROM cyrs.weather\_station WHERE id =\_id;  
END   
//

В этой процедуре используется такой же принцип переменных но здесь используется другой оператор, если в первой процедуре был оператор INSERT то в данной процедуре это оператор DELETE

## Заключение.

В заключение хочется сказать про то, какие цели были достигнуты.

При разработке ставилась цель создать базу данных на тему «». Эта задача была разбита на задачи: проектирование базы данных, создание таблиц, заполнение таблиц подходящими данными, создание хранимых процедур, создание представления, прописывание типовых запросов для структурирования данных в таблицах, создание особых ролей для базы данных.

Чтобы достичь цели, были решены следующие задачи.

Первая задача, проектировка базы данных, была достигнута созданием таблиц и связей между ними.

Вторая задача, создание таблиц, было реализована также на этапе создания структуры базы данных.

Третья задача, заполнить таблицу из базы данных, было выполнено благодаря открытым источником и оператору INSERT INTO.

Четвертая задача, создание хранимых процедур, было выполнено с учетом того, какие именно хранимые процедуры нужны научным разработчикам, по принципу написания процедур, которые у меня есть, можно создать подобные процедуры для каждой таблицы.

Пятая задача, проектирование представления, было достаточно легка из-за наличия таблиц weather\_station и type\_equipment. Было логичным решением расписать представление, в котором будет четко показано, какое конкретное оборудование имеется на конкретных метеостанциях.

Шестая задача, создание пяти типовых запросов, было так же реализовано. Задачи создавались с упором на то, чтобы было легче проводить анализ данных, что можно проследить в этих пяти запросах.

Седьмая задача заключалась в создании ролей для базы данных. Было создано две роли: администратор, у которого были все права на базу данных и ученый, который был ограничен в правах и не мог удалять таблицы и базу данных целиком.

Также стоит отметить, что у моей базы данных есть огромный потенциал для развития, и она может быть доработана, чтобы в дальнейшем выложить ее в открытый доступ для использования широкими массами населения, чтобы любой желающий мог проследить изменения погоды и климата.

В ходе работы над данной курсовой работой были достигнуты все поставленные цели, что привело к выводу о том, что такая база данных функциональна относительно поставленных целей и действительно актуальна, что может иметь развитие в дальнейшем.

## Литература.

<https://old.bigenc.ru/geography/text/2208703>

http://esimo.ru/dataview/viewresource?resourceId=RU\_RIHMI-WDC\_2667

https://ru.wikipedia.org/wiki/Нормальная\_форма

https://azure.microsoft.com/ru-ru/resources/cloud-computing-dictionary/what-is-postgresql/

https://ru.wikipedia.org/wiki/MySQL

## Приложение

Рисунок 1

Приложение 1