**R.A. Rodriges Zalipynis**
Р.А. Родригес Залепинос

# Spatial Data Management

## Управление пространственными данными

**Course Syllabus**
**(Curriculum)**
**in English**

Учебная программа
на английском языке

**ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ**
**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ**
**ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ**
**ВЫСШЕГО ОБРАЗОВАНИЯ**
**НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ**
**«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

Факультет компьютерных наук
Департамент программной инженерии

Р.А. Родригес Залепинос

# Spatial Data Management
# Управление пространственными данными

Учебная программа
на английском языке

Электронное издание
локального распространения

Санкт-Петербург
Наукоемкие технологии
2023

ISBN 978-5-907804-14-2

УДК 004.65
ББК 32.973
Р60

Рецензенты:

**Миркин Борис Григорьевич**, доктор технических наук, профессор департамента анализа данных и искусственного интеллекта, факультет компьютерных наук, НИУ ВШЭ;

**Шилов Валерий Владимирович**, кандидат технических наук, профессор департамента программной инженерии, факультет компьютерных наук, НИУ ВШЭ.

Р60 **Родригес Залепинос Р.А.** Spatial Data Management [Электронный ресурс] = Управление пространственными данными: учебная программа на английском языке. – Электрон. текстовые дан. (3,3 Мб). – СПб.: Наукоемкие технологии, 2023. – 30 с. – 1 электрон. опт. диск (CD-ROM).

ISBN 978-5-907804-14-2

По оценкам экспертов, около 80% всех данных в мире содержат локационные (пространственные) компоненты. Пространственные данные имеют решающее значение для управления городскими, лесными и водными ресурсами, быстрого реагирования на чрезвычайные ситуации, контроля качества воздуха, продовольственной безопасности, мониторинга окружающей среды и огромного множества других практически важных задач.

Учебная программа предлагает получить фундаментальные знания и навыки управления пространственными данными, изучив теоретический стек (определения, алгоритмы, подходы) и технологический стек (инструменты, системы, библиотеки) предметной области. Широко охватываются аспекты программного управления пространственными данными, включая их хранение, обработку и визуализацию. Используются реальные пространственные данные, например, векторные карты городов и данные дистанционного зондирования Земли. Данная учебная программа может помочь обучающемуся глубоко понять особенности пространственных данных, освоить теоретическую базу и пройти интенсивный практикум применения ключевых методов и средств управления пространственными данными.

Учебная программа предназначена для преподавателей ВУЗов, студентов программ бакалавриата (старшие курсы) и магистратуры (младшие курсы) в области компьютерных наук, а также лиц, заинтересованных в управлении пространственными данными.

Текстовое электронное издание

Минимальные системные требования:
- процессор: Intel x86, x64, AMD x86, x64 не менее 1 ГГц;
- оперативная память RAM ОЗУ: не менее 512 МБайт;
- свободное место на жестком диске (HDD): не менее 120 МБайт;
- операционная система: любая, при наличии Adobe Acrobat Reader;
- дисковод CD-ROM.

УДК 004.65
ББК 32.973

ISBN 978-5-907804-14-2

Учебное издание

**Родригес Залепинос** Рамон Антонио

# Spatial Data Management
# Управление пространственными данными

Учебная программа
на английском языке

Электронное издание
локального распространения

Подписано к использованию 25.12.2023
Объем издания – 3,3 Мб.
Комплектация издания – 1 CD.
Тираж 100 CD.

Издательство «Наукоемкие технологии»
ООО «Корпорация «Интел Групп»
https://publishing.intelgr.com
E-mail: publishing@intelgr.com
Тел.: +7 (812) 945-50-63

ISBN 978-5-907804-14-2

9 785907 804142

GOVERNMENT OF THE RUSSIAN FEDERATION
FEDERAL STATE AUTONOMOUS
EDUCATIONAL INSTITUTION OF HIGHER EDUCATION
NATIONAL RESEARCH UNIVERSITY
"HIGHER SCHOOL OF ECONOMICS"
(HSE UNIVERSITY)

Faculty of Computer Science
School of Software Engineering

Ramon Antonio Rodriges Zalipynis

# Spatial Data Management

Course Syllabus (Curriculum)

Electronic Edition

Saint Petersburg
Naukoemkie Technologii
2023

ISBN 978-5-907804-14-2

Reviewers:

**Boris G. Mirkin**, Doctor of Sciences, Professor, School of Data Analysis and Artificial Intelligence, Faculty of Computer Science, HSE University.
**Valery V. Shilov**, PhD, Professor, School of Software Engineering, Faculty of Computer Science, HSE University.

**R.A. Rodriges Zalipynis.** Spatial Data Management [Electronic Resource] = Spatial Data Management: Course Syllabus (Curriculum). – Text electronic edition (3.3 MB). – St. Petersburg: Naukoemkie Technologii, 2023. – 30 P. – 1 electronic optical disk (CD-ROM).

ISBN 978-5-907804-14-2

Experts estimate that location (spatial) components are part of about 80% of all data in the world. Spatial Data are crucial for Urban, Forest, and Water Management, Rapid Response to Emergencies, Air Quality Control, Food Security, Environmental Monitoring, and myriads of other practical tasks.

The Course Syllabus (Curriculum) proposes to acquire fundamental knowledge and skills in Spatial Data Management by studying the Theoretical Stack (includes definitions, algorithms, and approaches) and the Technological Stack (includes instruments, systems, and libraries) of the subject area. The course broadly covers programmatic aspects of spatial data management including spatial data storage, processing, and visualization. We use real-world spatial data, for example, vector city maps and Earth remote sensing data. The Course Syllabus (Curriculum) can help students deeply understand spatial data peculiarities, master the theoretical basis, and undergo intensive practical training in applying key methods and tools for managing spatial data.

University instructors, senior undergraduate and junior graduate computer science students, as well as anyone interested in Spatial Data Management are the target audience for this Course Syllabus (Curriculum).

Text electronic edition

Minimal system requirements:
– CPU: Intel x86, x64, AMD x86, x64 at least 1 GHz;
– RAM: at least 512 MB;
– Free disk space: at least 120 MB;
– Adobe Acrobat Reader or other PDF reader;
– CD-ROM.

ISBN 978-5-907804-14-2

Educational publication

**Rodriges Zalipynis** Ramon Antonio

# Spatial Data Management

Course Syllabus (Curriculum)
in English

Electronic Edition

Signed up for use 25.12.2023
Publication volume: 3.3 MB.
Edition contents: 1 CD.
Edition circulation: 100 CD.

Naukoemkie Technologii, Publishing House
Intel Group Corporation, Ltd.
https://publishing.intelgr.com
E-mail: publishing@intelgr.com
Tel.: +7 (812) 945-50-63

ISBN 978-5-907804-14-2
9 785907 804142

## TABLE OF CONTENTS

# 1. Course Description

## 1.1. ECTS

The Course Syllabus (Curriculum) is designed to meet the following ECTS parameters:

- Total ECTS credits: 5

With a total of 190 hours, of which

- Contact work: 60 hours (30 hours of lectures, 30 hours of seminars/practice)
- Independent work: 130 hours

## 1.2. Motivation

Experts estimate that location (spatial) components are part of about 80% of all data in the world [1, 2, 3]. Spatial Data are crucial for Urban, Forest, and Water Management, Rapid Response to Emergencies, Air Quality Control, Food Security, Environmental Monitoring, and myriads of other practical tasks [4].

[1] Forbes, https://www.forbes.com/sites/forbespr/2015/03/12/location-intelligence-leads-to-business-insights-and-creates-competitive-advantage/
[2] Forbes Insights, The Power of Place: How Location Intelligence Reveals Opportunity in Big Data, https://i.forbesimg.com/forbesinsights/pitney_bowes_power_of_place/PowerOfPlace.pdf
[3] Carto, https://carto.com/blog/four-analytic-types-shaping-location-data-today
[4] ArcGIS interactive book, https://learn.arcgis.com/ru/arcgis-imagery-book/

## 1.3. Pre-requisites

In general, this course has no mandatory pre-requisites. However, some topics in this course require the following skills: SQL, Java, or another programming language.

The course is given completely in English, so a good knowledge of English (at least listening and reading, intermediate speaking) is also required.

## 1.4. Courses that used this Syllabus (Curriculum)

The course "Spatial Data Management" was taught to students of the HSE University several times. Information about each course appears on the official HSE University Portal. The following course metadata is copied from the official HSE University Portal.

Spatial Data Management course in the year of study 2022/2023:

- Course name: **Spatial Data Management**
- Year of study: 2022/2023
- Type: Elective course (Software Engineering)
- Area of studies: Software Engineering
- Degree: Bachelor
- Delivered by: School of Software Engineering
- Where: Faculty of Computer Science
- When: 3 year, 1, 2 module (1 semester)
- Mode of studies: offline
- Open to: students of all HSE University campuses

- Instructors: Ramon Antonio Rodriges Zalipynis
- Language: English
- ECTS credits: 5
- Contact hours: 60

Link (official HSE University Portal):
https://www.hse.ru/en/edu/courses/646523060

Spatial Data Management course in the year of study 2023/2024:

- Course name: **Spatial Data Management**
- Year of study: 2023/2024
- Type: Elective course (Software Engineering)
- Area of studies: Software Engineering
- Degree: Bachelor
- Delivered by: School of Software Engineering
- Where: Faculty of Computer Science
- When: 3 year, 1, 2 module (1 semester)
- Mode of studies: offline
- Open to: students of all HSE University campuses
- Instructors: Ramon Antonio Rodriges Zalipynis
- Language: English
- ECTS credits: 5
- Contact hours: 56

Link (official HSE University Portal):
https://www.hse.ru/en/edu/courses/836731568

## 2. Learning Outcomes

### Study Objectives

- Know fundamental spatial data management theory.
- Know how an end-to-end spatial data management technological stack can be composed.
- Be able to apply spatial data management theory and tools to real-world data.

### Planned Results

- The student can list real-world spatial data management applications.
- The student can define and use spatial reference systems.
- The student can describe spatial Point, Line, and Polygon storage techniques.
- The student can apply spatial relationship logic to query spatial data.
- The student can describe spatial Multidimensional Array storage techniques.
- The student can apply spatial vector data management systems and operators to real-world spatial data.
- The student can apply spatial vector data management systems, libraries, and operators to real-world spatial data.
- The student can apply spatial array data management systems, libraries, and operators to real-world spatial data.
- The student can implement a basic client-server system for spatial data exchange.
- The student can build a basic spatial web mapping service.
- The student can describe the principles and techniques of spatial navigation systems.
- The student can narrate about state-of-the-art and foreseen spatial data management.

## 3. Assessment

The 10-point scale is basic for all assessment types. The contribution of each activity to the final grade is given in the table below.

**Activity Contribution Table**

| **Code** | **Assessment Type** | **Description** | **Grading Period** | **Contribution, %** |
|---|---|---|---|---|
| CW | In-class assignment | Control Work | 1st module | 20 |
| HW1 | Homework | Home Assignment 1 | 1st module | 10 |
| HW2 | Homework | Home Assignment 2 | 1st module | 10 |
| HW3 | Homework | Home Assignment 3 | 2nd module | 10 |
| HW4 | Homework | Home Assignment 4 | 2nd module | 10 |
| HW5 | Homework | Home Assignment 5 | 2nd module | 20 |
| EX | Written assignment | Exam | 2nd module | 20 |
| **TT** | | **Total contribution** | | 100 |

Examples of practical tasks and a home assignment are included in this syllabus.

The final grade is calculated according to TT (Total contribution).

**Final Grade Calculation Table**

| **Total Contribution, %** | **Final Grade** |
|---|---|
| <= 10 | 0 |
| <=15 | 1 |
| 20 | 2 |
| 30 | 3 |
| 40 | 4 |
| 50 | 5 |
| 60 | 6 |
| 70 | 7 |
| 80 | 8 |
| 90 | 9 |
| over 95 | 10 |

The Final Grade conversion rules from the 10-point scale to the 5-point scale are given in the table below.

**Conversion between the 10-point and 5-point grading systems**

| **10-point scale** | **5-point scale** |
|---|---|
| 1 – unsatisfactory<br>2 – very poor<br>3 – poor | unsatisfactory – 2 |
| 4 – satisfactory<br>5 – quite satisfactory | satisfactory – 3 |
| 6 – good<br>7 – very good | good – 4 |
| 8 – almost excellent<br>9 – excellent<br>10 – brilliant | excellent – 5 |

## 4. Competencies

The following competencies are formulated in accordance with “The Concept of the Educational Program “Software Engineering”, the direction of training № 03.09.04 “Software Engineering”. This program was ABET-accredited.

УК-1 Able to learn, acquire new knowledge, skills, including in a field other than professional

*All practical tasks and homework are directed towards acquiring new knowledge and skills. In addition, we use spatial data that is outside the software development knowledge domain like Earth remote sensing data. These data are used in a broad range of domains, including urban planning, forestry, agriculture, emergencies, and others. As these data are constantly increasing in volume and complexity, they require significant efforts from software engineering specialists.*

УК-2 Able to identify the scientific essence of problems in the professional field

*We use problems that have a solid underlying scientific basis. To solve a problem, a student must identify the scientific techniques that can be used to solve the problem.*

УК-3 Able to solve problems in professional activities based on analysis and synthesis

*All problems, home tasks, and practical lessons provide problems that require analysis and synthesis. To solve a problem, a student must decompose it and identify core building blocks, which must be used further to construct a complete solution.*

УК-4 Able to assess the need for resources and plan their use in solving problems in professional activities

*A student, when solving a task (a problem), must think thoroughly about which technological and time resources must be used in order to successfully solve the problem in time.*

УК-5 Ability to work with information: find, evaluate, and use information from various sources necessary to solve scientific and professional problems (including based on a systematic approach)

*The tasks in the course are posed in such a way that they provide required links to information that must be used for self-study. Typically, the information from various sources must be combined and applied to problem solving.*

УК-9 Be able to critically evaluate and rethink the accumulated experience (their own and others) to reflect on professional and social activities

*Without evaluating and rethinking the accumulated experience that a student acquires during the seminars (practical lessons), it will be significantly harder to solve a problem.*

ПК-1 Able to apply basic concepts, principles, theories, and facts related to information in solving scientific and research tasks

*The course gives fundamental knowledge and theory for managing spatial data. Hence, during the course, we require a student to apply these concepts and principles to problem solving.*

ПК-2 Capable of formalization in its subject area, taking into account the limitations of the research methods used

*Before solving a problem at a practical lesson or a home task, the problem must be formalized, taking into consideration the knowledge the student has gained so far during the course.*

ПК-9 Able to create software for computers and systems of various architectures

*Almost all practical work and home tasks are targeted at creating software.*

ПК-10 Capable of designing, constructing, and testing software products

*It is required that the solutions to home tasks and other stated problems (that is, software) be quality-assured: tested on a set of input data and within different conditions.*

ПК-11 Can read, understand, and isolate the main idea, read source code, documentation

*To solve a task, a student must acquire knowledge of working with particular software tools or libraries. Hence, a student must read and understand the documentation (and sometimes the source code) of the tools they use to solve the problem.*

ПК-13 Capable of evaluating time and capacity complexity software

*In several problems, a student is expected to solve a problem in several different ways. In addition, a student must evaluate the complexity of the solution and justify why one is more efficient than the other.*

ПК-15 Able to use operating systems, network technologies, software interface development tools, apply languages and formal specification methods, database management systems

*We actively use operating systems (Linux, Windows), software development tools (IntelliJ IDEA, others), programming languages (Java, SQL), libraries (netcdf-java, jts), and database management systems (PostgreSQL/PostGIS, h2).*

ПК-30 Capable of conscious goal setting, professional, and personal development

*All tasks are targeted to stimulate thinking, so this also stimulates professional and personal development.*

ПК-33 Able to understand and analyze worldviews, socially and personally significant problems and processes happening in society

*Many challenges are inspired by real-world and well-known problems, for example, related to air quality or urban management. These problems and use-cases have resulted in world-wide impacts. By analyzing real-world data related to these problems, students can form objective worldviews on significant problems and processes that have happened or will happen in society.*

# 5. Course Plan

## 5.1. Course Outline

The following table defines the course plan. Please note that for some topics two lectures and seminars / practical lessons are assigned.

| № | Topic name | Course hours, total | Contact hours | | Self-study |
|---|---|---|---|---|---|
| | | | Lectures | Practical lessons | |
| **Module 1** | | | | | |
| 1 | Introduction to Spatial Data Management | 4 | 2 | 2 | 0 |
| 2 | Spatial Reference Systems | 8 | 4 | 4 | 20 |
| 3 | Spatial Vector Data | 4 | 2 | 2 | 10 |
| 4 | Spatial Relationship Logic (Topology) | 4 | 2 | 2 | 10 |
| 5 | Spatial Vector Data Management | 8 | 4 | 4 | 20 |
| **Subtotal:** | | **28** | **14** | **14** | **60** |
| **Module 2** | | | | | |
| 6 | Spatial Multidimensional Arrays | 4 | 2 | 2 | 10 |
| 7 | Spatial Array Data Management | 8 | 4 | 4 | 20 |
| 8 | Spatial Data Communication | 8 | 4 | 4 | 20 |
| 9 | Spatial Web Mapping | 4 | 2 | 2 | 10 |
| 10 | Spatial Navigation Systems | 4 | 2 | 2 | 10 |
| 11 | The Future of Spatial Data Management | 4 | 2 | 2 | 0 |
| **Subtotal:** | | **32** | **16** | **16** | **70** |
| **Total:** | | **60** | **30** | **30** | **130** |

## 5.2. Software Support

Below is the list of core software required for the course.
The software is open source and freely available.

| № | Name | Availability |
|---|---|---|
| 1 | Java Development Kit (JDK), at least version 21 LTS | Free software |
| 2 | Panoply (data visualization) | Free software |
| 3 | Linux | Free software |
| 4 | GDAL (Geospatial Data Abstraction Library) | Free software |
| 5 | IntelliJ IDEA Community, Eclipse, or other free IDE | Free software |
| 6 | Netty framework | Free software |
| 7 | QGIS (Quantum GIS), a Geographic Information System | Free software |
| 8 | Python | Free software |
| 9 | PostgreSQL / PostGIS | Free software |
| 10 | PgAdmin | Free software |
| 11 | Netcdf-java | Free software |
| 12 | GeoTools | Free software |
| 13 | h2/h2gis | Free software |
| 14 | NCO (NetCDF Operators) | Free software |
| 15 | OTB (Orfeo ToolBox) | Free software |

## 5.3. Topic-Wise Course Content

### Topic 1
### Introduction to Spatial Data Management

Spatial Data Management Concepts, Challenges, Input and Output Data. Spatial database management systems and spatial data platforms. Big Spatial Data (6Vs): volume, velocity, variety, veracity, value, and variability. Examples of spatial data sources: unmanned aerial vehicles, Earth remote sensing satellites, mobile devices, IoT (Internet of Things), and generated data. Examples of problems that can be tackled with spatial data: insurance companies (forestry, hurricanes, etc.), modeling of traffic movement (optimizing traffic lights, what-if questions of road network modifications), monitoring of infrastructure development, and other research and business applications.

**Seminar/Practice.**

It is recommended to visually explore samples of spatial-temporal data to get a sense of what they look like:

- Satellites, LIDARs, UAVs
- Numerical Simulation (weather, climate, hydrology, …)
- Diverse Sensors
- Mobile Devices
- Any Georeferenced Rasters

Many spatial-temporal data samples can be explored using QGIS (Quantum GIS), a freely available Geographic Information System. Of course, the visual exploration should be accompanied by introductory descriptions of spatial-temporal data samples.

Please discuss the assertion that 80% of all data have a location component.
For example, refer to Forbes [1].

It is also possible to note that:

- Major conferences allocate separate sessions for Spatial Data, e.g.
  VLDB (Very Large Data Bases): https://vldb2020.org/program.html
- SIGSPATIAL – the conference is entirely devoted to spatial data.
- Space Research Institute – an institute that heavily focuses on a specific spatial data subtype: Earth remote sensing data.

[1] Forbes: Location Intelligence Leads to Business Insights and Creates Competitive Advantage https://www.forbes.com/sites/forbespr/2015/03/12/location-intelligence-leads-to-business-insights-and-creates-competitive-advantage

## Topic 2
## Spatial Reference Systems

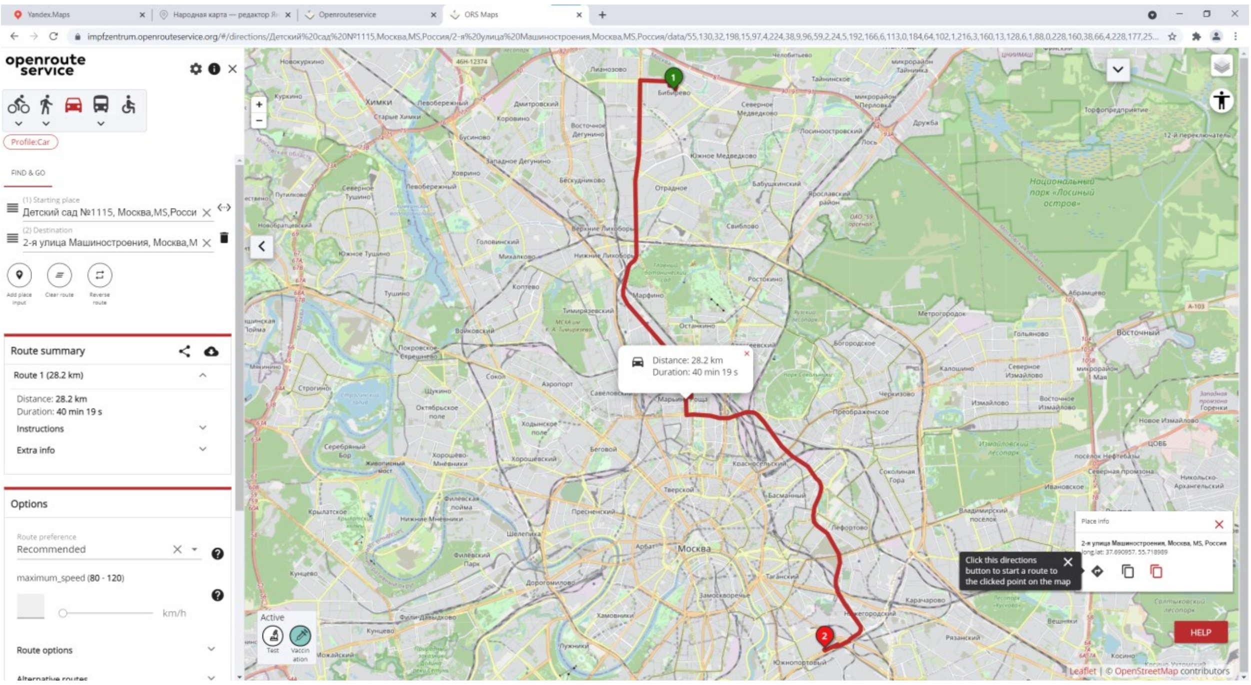
Calculating areas, distances, angles, and perceiving shapes on a 2-d map. Spatial Reference Systems (SRS). Types of SRS Text Formats (WKT: WKT1, WKT2, proj4, XML, etc.). Calculating distances (Euclidean and Haversine formulae). SRS support by data management systems. Predefined and commonly used Spatial Reference Systems. Illustrations of area and distance calculations with proper and improper spatial reference systems: real-world World Bank data and OpenRouteService tracks.

### Seminar/Practice.

Compute the length of a real-world track in different spatial reference systems.

Download a real-world track from **OpenRouteService** (this feature is not available on Google Maps, Bing Maps, Yandex Maps, and similar map services): https://openrouteservice.org/

Please, find an example in Fig. 1.



impfzentrum.openrouteservice.org/#/directions/Детский%20сад%20№1115,Москва,MS,Россия/2-я%20улица%20Машиностроения,Москва,MS,Россия/data/55,130,32,198,15,97,4,224,38,9,96,59,2,24,5,192,166,6,113,0,184,64,102,1,216,3,160,13,128,6,1,88,0,228,160,38,66,4,228,177,25...

(1) Starting place: Детский сад №1115, Москва,MS,Росси

(2) Destination: 2-я улица Машиностроения, Москва,M

Route 1 (28.2 km)

Distance: 28.2 km
Duration: 40 min 19 s

Route preference: Recommended

maximum_speed (80 - 120) km/h

2-я улица Машиностроения, Москва, MS, Россия

Click this directions button to start a route to the clicked point on the map

**Fig. 1.** A real-world track from OpenRouteService

The format of the track is GeoJSON, a fragment of the file is shown in Fig. 2. The track is annotated with diverse routing information.

```
{
  "type": "FeatureCollection",
  "features": [
    {
      "type": "Feature",
      "properties": {
        "ascent": 103.1,
        "descent": 120.1,
        "segments": [
          {
            "distance": 28.233,
            "duration": 2419.3,
            "steps": [
              {
                "distance": 0.121,
                "duration": 27.6,
                "type": 11,
                "instruction": "Head north",
                "name": "-",
                "way_points": [
                  0,
                  3
                ]
              },
```

**Fig. 2.** A track fragment from OpenRouteService (GeoJSON)

Next, we can try calculating the distance using EPSG:3857 – a coordinate reference system typically used for a Web Map: Google Maps, and others, see http://epsg.io/3857

Please, try:
**length(transform($geometry, 'EPSG:4326', 'EPSG:3857'))**

In the result, we get a distance equal to 50 km. However, the real distance is about 28 km (it can be found in the file itself, check for `"distance": 28.233` in the fragment above, Fig. 2). The difference is about 22 km, which is 78%.

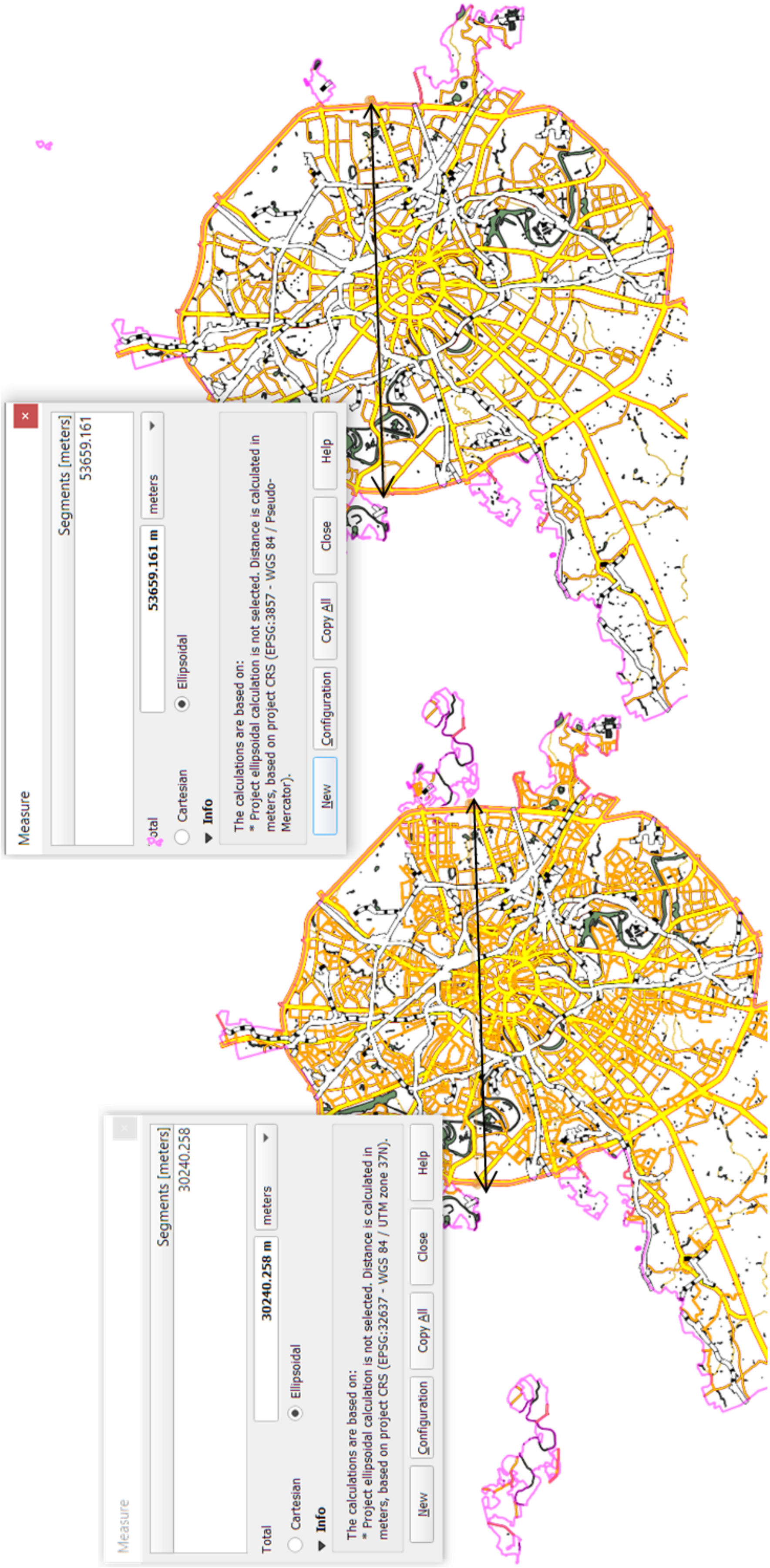
To get the correct result, we must use a proper spatial reference system. For Moscow, it can be UTM 37N, which is based on the conformal projection and the precision is sufficient for our purposes.

Let's try:
**length(transform($geometry, 'EPSG:4326', 'EPSG:32637'))**

In the result, we get a distance equal to 28 km, which is what we want (and what is correct).

Please, note that the shape of Moscow is almost identical for both **'EPSG:32637'** and **'EPSG:3857'**, but the distance estimates made for the identical routes differ greatly.

Please, compare the figures below (Fig. 3). Which map of Moscow is displayed in **'EPSG:32637'** and which is in **'EPSG:3857'**?



Measure

Segments [meters]

53659.161

Total 53659.161 m meters

○ Cartesian ● Ellipsoidal

▼ Info

The calculations are based on:
* Project ellipsoidal calculation is not selected. Distance is calculated in meters, based on project CRS (EPSG:3857 - WGS 84 / Pseudo-Mercator).

Measure

Segments [meters]

30240.258

Total 30240.258 m meters

○ Cartesian ● Ellipsoidal

▼ Info

The calculations are based on:
* Project ellipsoidal calculation is not selected. Distance is calculated in meters, based on project CRS (EPSG:32637 - WGS 84 / UTM zone 37N).

**Fig. 3.** A visualization of Moscow in 'EPSG:32637' and 'EPSG:3857' (OpenStreetMap data, screenshot using QGIS)

## Topic 3
## Spatial Vector Data

Point, Line, Polygon, Multi types, Geometry Collection, Empty data**.** OGC Simple Features. Package and companion files. Objects and attribute tables. PostgreSQL/PostGIS. Encoding. Spatial and attribute indexing: goals and techniques. Creating and using spatial indexes. Spatial Extensions. Spatial data types (geometry, geography, box3d, box2d, spheroid, raster, gidx, topology.TopoGeometry). Spatial objects. CREATE (DATABASE, TABLE, INDEX), SELECT, INSERT, UPDATE. GiST (Generalized Search Tree). shp2pgsql, pgsql2shp, raster2pgsql. JDBC.

**Seminar/Practice**.

### First steps with a Spatial DBMS.

A Spatial Database Management System (SDBMS) provides spatial data types, storage engines, functions to process/query spatial data (usually with optimizations).

When choosing a SDBMS, several important aspects must be considered:

- Is it compliant with OGC Simple Features?
- Does it support both raster and vector data?
- What additional dimensions does it support, except x (lon), y (lat), and z (height)?
- Distributed? License? Hadoop or PostgreSQL interaction?

In our course, the PostgreSQL/PostGIS bundle is recommended for use. PostgreSQL is open-source, popular, powerful, and stable. Its initial release was in 1989 and it is in the top 5 database management systems by popularity according to the DB Engines ranking portal. Let’s try the following (using a graphical user interface called PgAdmin):

1. Create a database in PostgreSQL
2. Enable PostGIS in the newly created database
3. Create tables (possibly with columns of spatial data types)
4. Create indices (to accelerate SELECT statements)
5. Execute SQL queries: SELECT, INSERT, UPDATE, DELETE
6. Use spatial functions in SQL queries
7. Use command line tools shipped with PostGIS to import/export data to/from the database
8. Use JDBC to programmatically connect to PostgreSQL and execute SQL queries, import/export data

We can enable PostGIS by using the commands in Fig. 4.

```
-- Enable PostGIS (includes raster)
CREATE EXTENSION postgis;
-- Enable Topology
CREATE EXTENSION postgis_topology;
```

**Fig. 4.** Enable PostGIS (basic extensions)

This is because PostGIS is a set of data types and functions designed for PostgreSQL.
Next, we must create a table that can store our spatial data types (e.g., geometries), Fig. 5.

```
-- Create a table with a spatial column
CREATE TABLE geotable (
 id SERIAL PRIMARY KEY,
 geom GEOMETRY(Point, 26910),
 name VARCHAR(128)
);
```

**Fig. 5.** Create a table with a spatial column

In Fig. 5 we create a table with three columns. The first column is an integer field that will be automatically incremented each time we insert a new row. It will serve as a primary key. The next column will store our geometries. The type of geometries is restricted together with its spatial reference system. This column will accept only points in EPSG:26910. Insertion of other geometry types or points with coordinates in other spatial reference systems will result in errors.

After we create a table, the best practice is to add an index for some of its columns, Fig. 6. Database indexes are data structures that can significantly accelerate query execution.

```
-- Add a spatial index
CREATE INDEX geotable_gix
 ON geotable
 USING GIST (geom);
```

**Fig. 6.** Add a spatial index

We can populate our table by using INSERT INTO queries which are part of the standard SQL syntax. Let’s insert a point, Fig. 7.

```
-- Add a point
INSERT INTO geotable (geom) VALUES (
 ST_GeomFromText('POINT(0 0)', 26910)
);
```

**Fig. 7.** Add a point

In the query above, ST_GeomFromText is a PostGIS function that creates a binary representation of the geometry specified in human-readable text that can be stored inside a PostgreSQL table. Please note that 26910 is a spatial reference system identifier. You can insert multiple geometries using a single INSERT INTO statement.

Finally, you can query your geometries. For example, let’s find all points that are within 1000 meters of some given point, in this case a point with coordinates (0, 0), Fig. 8. We use the same function for creating a gemetry as for the previous query and a PostGIS function ST_DWithin which returns TRUE if two geometries (which are both function arguments) are within the given distance from each other.

```
SELECT id, name
FROM geotable
WHERE ST_DWithin(
 geom,
 ST_GeomFromText('POINT(0 0)', 26910),
 1000
);
```

**Fig. 8.** Query for nearby points

## Topic 4
## Spatial Relationship Logic (Topology)

Distance relationship: test if two geometries are within or entirely within a given distance; test if a point geometry is inside a circle defined by a center and radius. Given geometries X and Y, test relationships between X and Y: no points of X are outside Y; X and Y have only interior points in common; X and Y have some but not all points in common; X and Y have no points in common; X and Y are the same geometry with points in the same directional order. Generating and using DE-9IM. Practical tasks on real-world geospatial data.

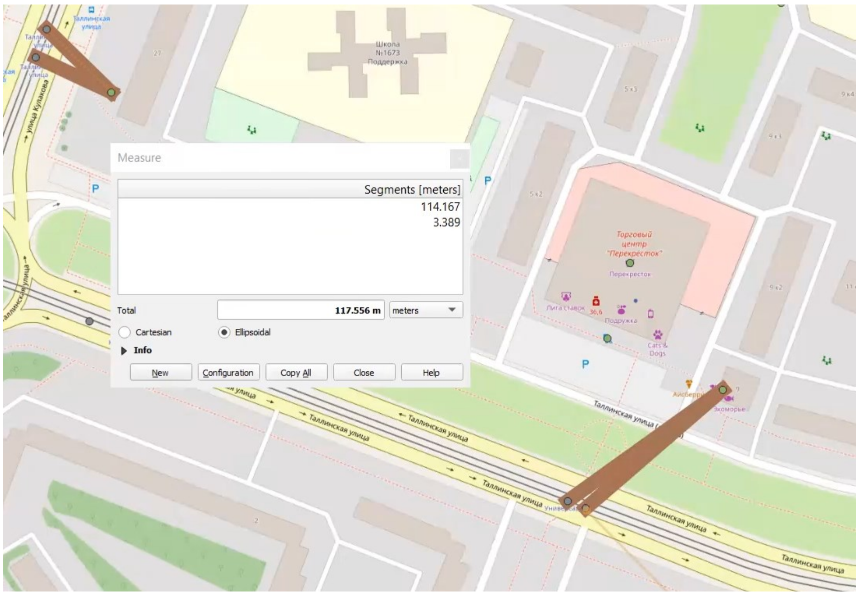
### Seminar/Practice.

**Insurance Agent / Representative.**
**Please, solve the following task.**
Bob works as an insurance representative. In addition, he dislikes taking long walks. His insurance company provides him with an unlimited tram ticket. Bob uses the tram ticket to visit shops and negotiate contracts with their managers. Unfortunately, the trams do not go directly to the shops, so Bob visits a shop only if the shop if it is no more than 200 meters from the closest tram stop.

Please assist Bob in creating a map of the shops he can get to under the given constraints. In other words, design a query such that, based on the objects from the files "poi-point.*" (the "shop" field is not null) and "railway-station-point.*" (railway = 'tram_stop'), will find such shops where the distance between the shop and the nearest tram stop is not more than 200 meters.

In addition, the query should return not just points (shops); it is inconvenient to visualize those on the map, and it is not clear to Bob which stop he needs to get to in order to visit a particular shop. The query must return a set of LineString objects, each of which (1) consists of two points: a shop and a tram stop, (2) has length of no more than 200 meters.

It is forbidden to use distance calculation operations (only topological operators to determine the distance). The result should look similar to the one shown in Fig. 9.



Segments [meters]
114.167
3.389

Total 117.556 m meters

○ Cartesian ● Ellipsoidal

▶ Info

**Fig. 9.** A Portion of the Solution to the Insurance Agent Task (Screenshot using QGIS)

## Topic 5
## Spatial Vector Data Management

Spatial vector management tools. The ogrinfo tool. Data sources. Layers, features, schema, extent, projection. Attribute queries in restricted form. SQL queries. Area of interest. Metadata domains. The PostGIS SDBMS. Geometry Constructors: ST_Collect, ST_Make(Envelope, Line, Point, Polygon). Geometry Accessors. Geometry Editors (ST_AddPoint, FlipCoordinates, Multi, Normalize, RemoveRepeatedPoints, SnapToGrid, etc.). Validation (ST_IsValid, MakeValid). Geometry input: ST_GeomFromText and others. Bounding Box Operators. Distance Operators.

**Seminar/Practice.**

**Infrastructure Planning.**
To optimize traffic flow, the city officials intend to construct a new road (street, highway, etc. – it doesn't matter). You are instructed to choose a place for the new road. Suppose you need to connect several existing roads or build a parallel road to an existing one. In any case, choose a new road located close to existing facilities (buildings and transport stops). Your new road is 20 meters wide and at least 1 km long. Be sure to build a road that is less than 20 meters away from some buildings or stops (QGIS has a distance measurement tool).

Create a new road as a single LineString or MultiLineString object.
To complete the task, use QGIS:
1. Create a new vector layer (geometry type LineString or MultiLineString)
2. Switch to edit mode
3. Draw an object
4. Save the layer
5. Highlight your object with a special color and thickness

Find transport stops and buildings (geometries) to be demolished to build the road (objects from railway-station-point.shp and building-polygon.shp OSM files). It is forbidden to use distance calculation operations (use other spatial vector management operations). The new road (its user-friendly textual definition) can be specified directly in the query.

Next, connect to PostgreSQL from QGIS, construct and run the respective query in the DB Manager, and display the resulting geometry features as a separate layer. Color objects such that they are clearly visible, Fig. 10.

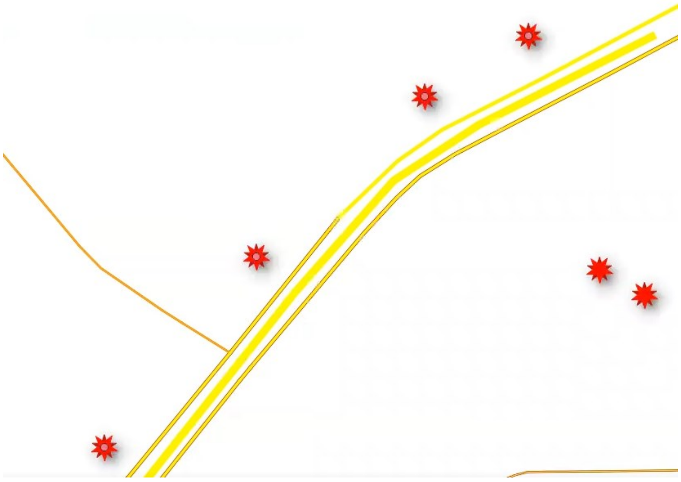


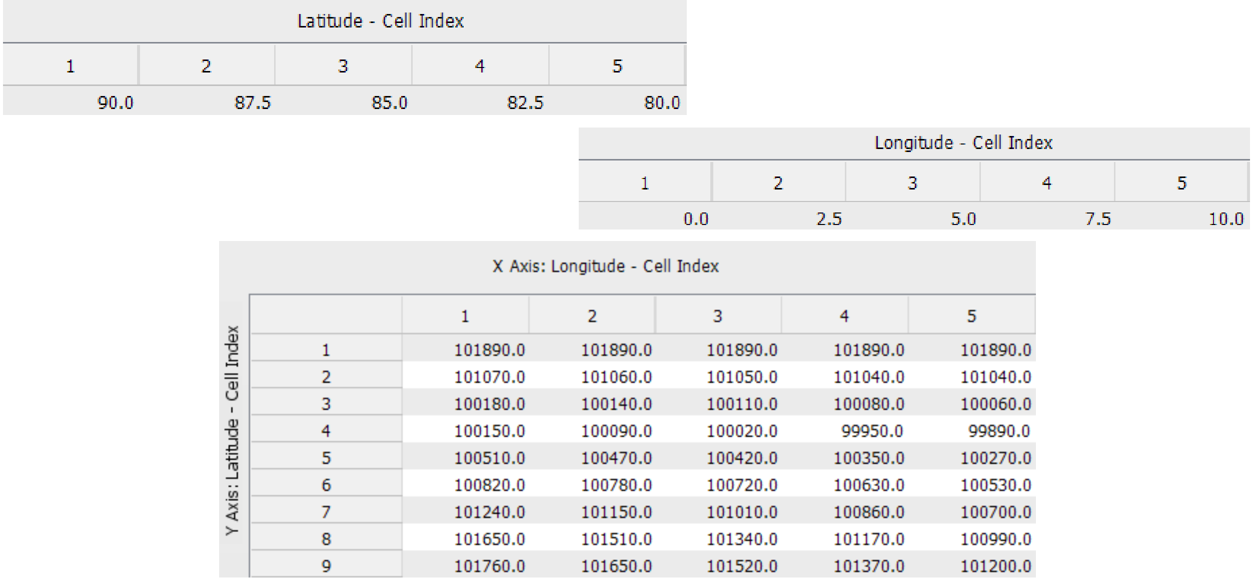
**Fig. 10.** A Portion of the Solution to the Infrastructure Planning Task (Screenshot using QGIS)

## Topic 6
## Spatial Multidimensional Arrays

Band, array, variable, dataset. Metadata. Data types (unsigned and signed integers, floats, and complex types). Palette-based images. Chunking. Color models. Tags. Worldfiles and georeferencing. Virtual file formats. Ground control points. Bit number. Block size. Overviews. NoData representation. Sparse files. Interleave types. Compression modes (JPEG, LZW, PACKBITS, etc.). Big files (over 4 GB in volume). Spatial Joins.

### Seminar/Practice.
### Explore Multidimensional Arrays

First, the notion of coordinate arrays should be clearly explained, Fig. 11.



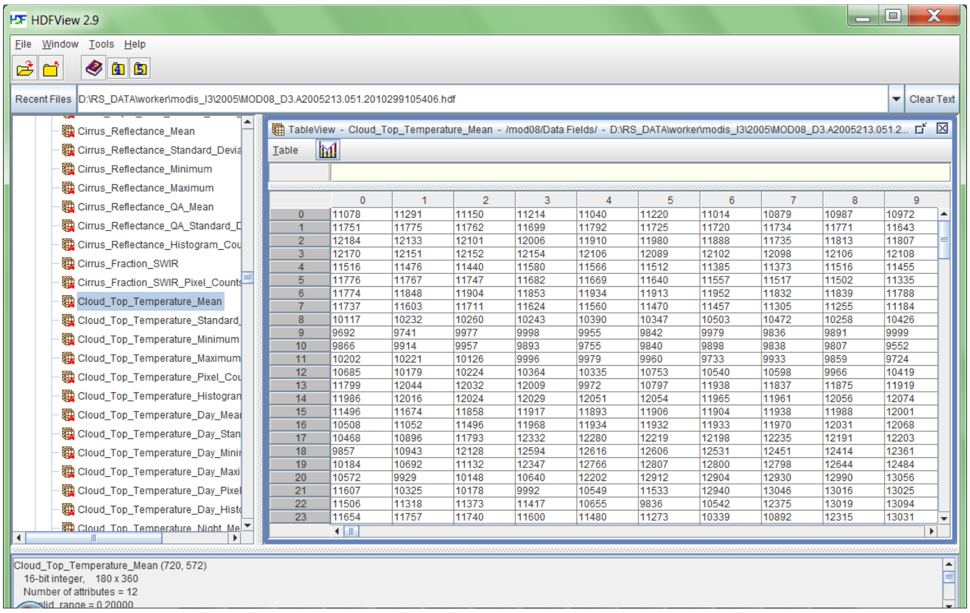
| Latitude - Cell Index | | | | |
|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 |
| 90.0 | 87.5 | 85.0 | 82.5 | 80.0 |

| Longitude - Cell Index | | | | |
|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 |
| 0.0 | 2.5 | 5.0 | 7.5 | 10.0 |

| Y Axis: Latitude - Cell Index | X Axis: Longitude - Cell Index | | | | |
|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 |
| 1 | 101890.0 | 101890.0 | 101890.0 | 101890.0 | 101890.0 |
| 2 | 101070.0 | 101060.0 | 101050.0 | 101040.0 | 101040.0 |
| 3 | 100180.0 | 100140.0 | 100110.0 | 100080.0 | 100060.0 |
| 4 | 100150.0 | 100090.0 | 100020.0 | 99950.0 | 99890.0 |
| 5 | 100510.0 | 100470.0 | 100420.0 | 100350.0 | 100270.0 |
| 6 | 100820.0 | 100780.0 | 100720.0 | 100630.0 | 100530.0 |
| 7 | 101240.0 | 101150.0 | 101010.0 | 100860.0 | 100700.0 |
| 8 | 101650.0 | 101510.0 | 101340.0 | 101170.0 | 100990.0 |
| 9 | 101760.0 | 101650.0 | 101520.0 | 101370.0 | 101200.0 |

**Fig. 11.** Data Array and Coordinate Arrays

Next, *N*-d arrays can be explored in a graphical tool like HDFView or Panoply, Fig. 12.



D:\RS_DATA\worker\modis_l3\2005\MOD08_D3.A2005213.051.2010299105406.hdf

- Cirrus_Reflectance_Mean
- Cirrus_Reflectance_Standard_Devia
- Cirrus_Reflectance_Minimum
- Cirrus_Reflectance_Maximum
- Cirrus_Reflectance_QA_Mean
- Cirrus_Reflectance_QA_Standard_D
- Cirrus_Reflectance_Histogram_Cou
- Cirrus_Fraction_SWIR
- Cirrus_Fraction_SWIR_Pixel_Counts
- Cloud_Top_Temperature_Mean
- Cloud_Top_Temperature_Standard_
- Cloud_Top_Temperature_Minimum
- Cloud_Top_Temperature_Maximum
- Cloud_Top_Temperature_Pixel_Cou
- Cloud_Top_Temperature_Histogran
- Cloud_Top_Temperature_Day_Mear
- Cloud_Top_Temperature_Day_Stan
- Cloud_Top_Temperature_Day_Minir
- Cloud_Top_Temperature_Day_Maxi
- Cloud_Top_Temperature_Day_Pixel
- Cloud_Top_Temperature_Day_Histo
- Cloud_Top_Temperature_Night_Me

TableView - Cloud_Top_Temperature_Mean - /mod08/Data Fields/ - D:\RS_DATA\worker\modis_l3\2005\MOD08_D3.A2005213.051.2...

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 11078 | 11291 | 11150 | 11214 | 11040 | 11220 | 11014 | 10879 | 10987 | 10972 |
| 1 | 11751 | 11775 | 11762 | 11699 | 11792 | 11725 | 11720 | 11734 | 11771 | 11643 |
| 2 | 12184 | 12133 | 12101 | 12006 | 11910 | 11980 | 11888 | 11735 | 11813 | 11807 |
| 3 | 12170 | 12151 | 12152 | 12154 | 12106 | 12089 | 12102 | 12098 | 12106 | 12108 |
| 4 | 11516 | 11476 | 11440 | 11580 | 11566 | 11512 | 11385 | 11373 | 11516 | 11455 |
| 5 | 11776 | 11767 | 11747 | 11682 | 11669 | 11640 | 11557 | 11517 | 11502 | 11335 |
| 6 | 11774 | 11848 | 11904 | 11853 | 11934 | 11913 | 11952 | 11832 | 11839 | 11788 |
| 7 | 11737 | 11603 | 11711 | 11624 | 11560 | 11470 | 11457 | 11305 | 11255 | 11184 |
| 8 | 10117 | 10232 | 10260 | 10243 | 10390 | 10347 | 10503 | 10472 | 10258 | 10426 |
| 9 | 9692 | 9741 | 9977 | 9998 | 9955 | 9842 | 9979 | 9836 | 9891 | 9999 |
| 10 | 9866 | 9914 | 9957 | 9893 | 9755 | 9840 | 9898 | 9838 | 9807 | 9552 |
| 11 | 10202 | 10221 | 10126 | 9996 | 9979 | 9960 | 9733 | 9933 | 9859 | 9724 |
| 12 | 10685 | 10179 | 10224 | 10364 | 10335 | 10753 | 10540 | 10598 | 9966 | 10419 |
| 13 | 11799 | 12044 | 12032 | 12009 | 9972 | 10797 | 11938 | 11837 | 11875 | 11919 |
| 14 | 11986 | 12016 | 12024 | 12029 | 12051 | 12054 | 11965 | 11961 | 12056 | 12074 |
| 15 | 11496 | 11674 | 11858 | 11917 | 11893 | 11906 | 11904 | 11938 | 11988 | 12001 |
| 16 | 10508 | 11052 | 11496 | 11968 | 11934 | 11932 | 11933 | 11970 | 12031 | 12068 |
| 17 | 10468 | 10896 | 11793 | 12332 | 12280 | 12219 | 12198 | 12235 | 12191 | 12203 |
| 18 | 9857 | 10943 | 12128 | 12594 | 12616 | 12606 | 12531 | 12451 | 12414 | 12361 |
| 19 | 10184 | 10692 | 11132 | 12347 | 12766 | 12807 | 12800 | 12798 | 12644 | 12484 |
| 20 | 10572 | 9929 | 10148 | 10640 | 12202 | 12912 | 12904 | 12930 | 12990 | 13056 |
| 21 | 11607 | 10325 | 10178 | 9992 | 10549 | 11533 | 12940 | 13046 | 13016 | 13025 |
| 22 | 11506 | 11318 | 11373 | 11417 | 10655 | 9836 | 10542 | 12375 | 13019 | 13094 |
| 23 | 11654 | 11757 | 11740 | 11600 | 11480 | 11273 | 10339 | 10892 | 12315 | 13031 |

```
Cloud_Top_Temperature_Mean (720, 572)
  16-bit integer,    180 x 360
  Number of attributes = 12
  valid_range = 0 20000
```

**Fig. 12.** Array Exploration in HDFView (screenshot)

## Topic 7
## Spatial Array Data Management

Array DBMSs and Array-Oriented Systems. Array Data Models and Query Languages. Array Storage: files, “database approaches”, array compression. Array Operations (Workload Types). Physical simulations and array data models. Convergence of discrete simulations and array (tensor) database management systems. Array Join Techniques. Array Tiling & Chunking Strategies. In-situ Array Processing. Array Indexing. Array Visualization and Machine Learning. Array Database Benchmarks. SciDB, ChronosDB, and RasDaMan Array DBMSs. Numerous promising Research and Development opportunities for future work.

**Required material:**

- Ramon Antonio Rodriges Zalipynis. Array DBMS: Past, Present, and (Near) Future. PVLDB, 14(12): 3186–3189, 2021. PDF: http://www.vldb.org/pvldb/vol14/p3186-zalipynis.pdf

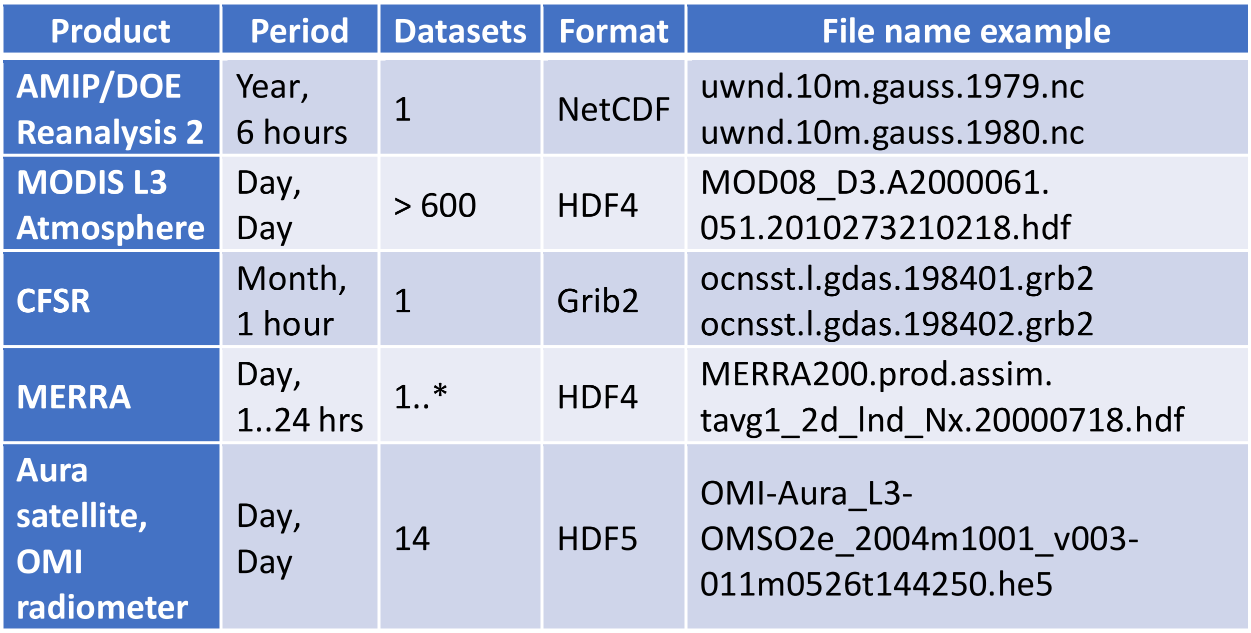
**Required watch:**

- Video-lectures: http://vldb2021.gis.gg/

**Seminar/Practice.**
Arrays are traditionally stored in files that have complex naming, diverse coordinate systems, formats, and data types. The same dataset may have very different storage layouts in different file formats, Fig. 13.

A part of the Seminar/Practice can be also devoted to exploring diverse raster data types. For example, http://www.wikience.org/data/air-quality-satellite/

Students can also utilize diverse software libraries to read and manipulate the array datasets. In addition, some datasets can be ingested into SciDB, ChronosDB, RasDaMan, or other Array DBMS in order to manage the arrays via a specialized array query language.



| Product | Period | Datasets | Format | File name example |
|---|---|---|---|---|
| AMIP/DOE Reanalysis 2 | Year,<br>6 hours | 1 | NetCDF | uwnd.10m.gauss.1979.nc<br>uwnd.10m.gauss.1980.nc |
| MODIS L3 Atmosphere | Day,<br>Day | > 600 | HDF4 | MOD08_D3.A2000061.<br>051.2010273210218.hdf |
| CFSR | Month,<br>1 hour | 1 | Grib2 | ocnsst.l.gdas.198401.grb2<br>ocnsst.l.gdas.198402.grb2 |
| MERRA | Day,<br>1..24 hrs | 1..* | HDF4 | MERRA200.prod.assim.<br>tavg1_2d_lnd_Nx.20000718.hdf |
| Aura satellite, OMI radiometer | Day,<br>Day | 14 | HDF5 | OMI-Aura_L3-<br>OMSO2e_2004m1001_v003-<br>011m0526t144250.he5 |

**Fig. 13.** Examples of Array Datasets and Their Storage Containers

## Topic 8
## Spatial Data Communication

High-performance spatial data dissemination. Spatial data communication protocols. Software engineering and spatial data communication. Business logic. Query-Response Spatial Data Serving. Typical patterns for query processing. Decoding queries, unpacking messages, processing, and encoding results. Accepting incoming connections and registering channels. Channel lifetimes. Persisting information from query to query. Input and output pipelines. Multithreaded query processing. Interoperability of spatial data servers and spatial standardization. Spatial data infrastructure (SDI).

**Seminar/Practice.**
Practice is required to consolidate the knowledge of the Spatial Data Communication software facilities. First, revise the patterns of high-performance spatial data dissemination, see Fig. 14. Next, develop a server based on Netty (Java), a JavaScript client, and a business protocol on top of WebSocket. The client must have a GUI with a (1) text box containing a geometry name, (2) text area for response status, (3) button "show", and (4) an area to display the geometry.

Functionality requirements:

- A user must be able to type a geometry definition in the text box
- When a user clicks the button, the client submits the request to the server
- The client → server request must contain the defined geometry
- The server receives the request
- If the geometry syntax is correct and if the geometry specification is valid, the server renders the geometry and sends the rendered image to the client (transfer mode is binary)
- Otherwise, the server responds with an error message to the client
- When the client receives a successful response from the server, the client displays the picture within the display area; the client displays the text "Showing " + the geometry definition in the text area for the response status
- When the client receives an error response, the client must display the received error message (text) in the text area for the response status

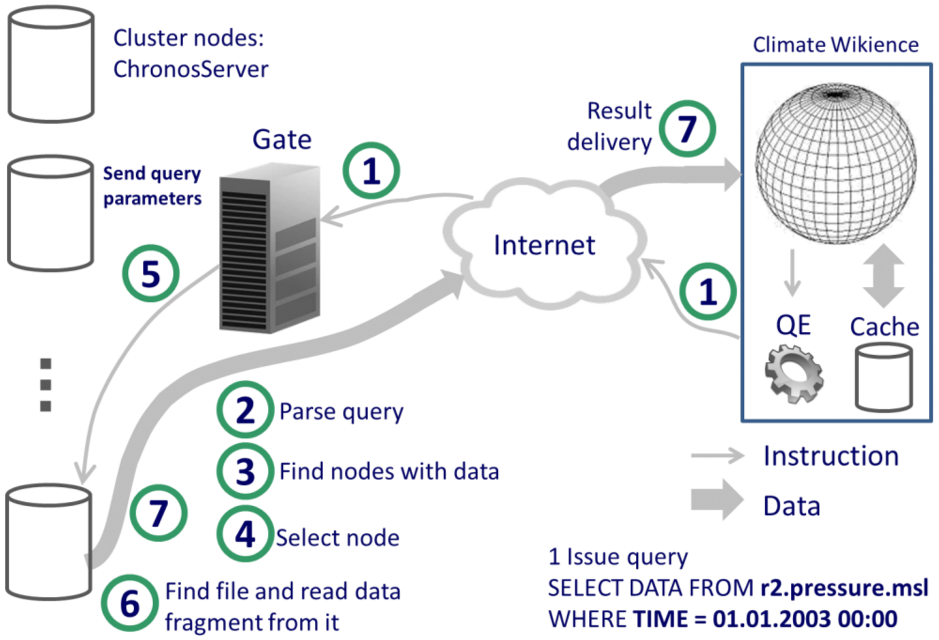


**Fig. 14.** ChronosServer Query Execution (shown for 1 client).
Author: R.A. Rodriges Zalipynis, Source: http://www.wikience.org/chronosserver/

## Topic 9
## Spatial Web Mapping

Challenges of presenting spatial data in Web browsers. Web maps. Tiles and images. Approaches to delivering spatial data to Web browsers. High-performance tile rendering. Canvas, HTML5, WebGL and their roles in Web Mapping. Map controls. Layers and styles. Tile data providers. Calculations inside a Web browser. Modern techniques for managing spatial data on the Web. Web Array DBMSs and Web GISs.

### Required material:

- R.A. Rodriges Zalipynis, N. Terlych. WebArrayDB: A Geospatial Array DBMS in Your Web Browser. PVLDB, 15(12): 3622–3625, 2022. https://wikience.github.io/webdb2022

### Seminar/Practice.
### Computing and Mapping Raster Data in a Web browser.

WebArrayDB and ArrayGIS are innovative systems. WebArrayDB is the first geospatial array DBMS that can run completely inside a Web browser. ArrayGIS is a new Web GIS based on WebArrayDB. A Web-based array DBMS should reduce server load, enable offline work, decrease network I/O, and improve user experience.

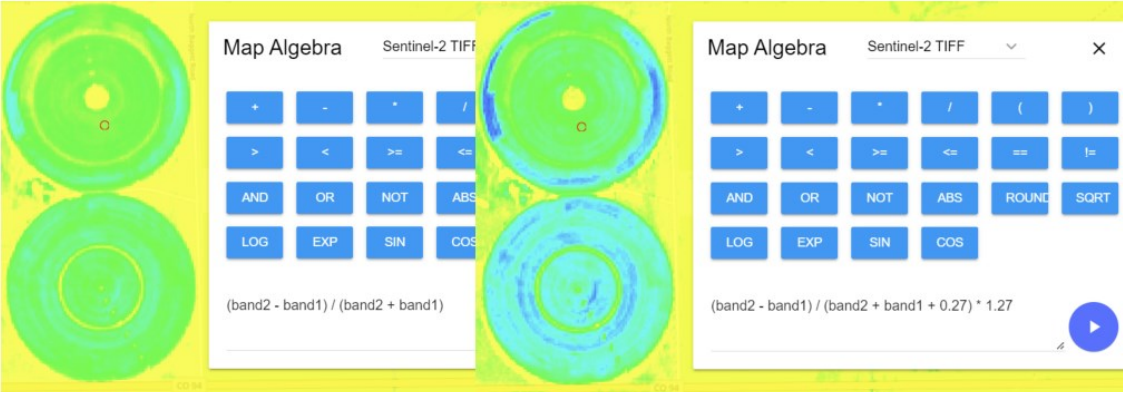
These systems deliver raw spatial array data to Web browsers. You can use SQL-like queries to perform computations directly in a Web browser. For example, the following query will compute a popular vegetation index:

```
SELECT (band8 - band4)/(band8 + band4)
INTO NDVI
FROM WMTS (https://services.sentinel-hub.com/ogc/wmts/
<personal_api_key>?REQUEST=GetCapabilities)
```

ArrayGIS is freely available at http://webdb.gis.gg

In addition, you can check out the paper to go step-by-step through interactive lessons related to real-world tasks on multispectral satellite imagery.

For example, you can go step-by-step through the food security lesson, Fig. 15.



(band2 - band1) / (band2 + band1)

(band2 - band1) / (band2 + band1 + 0.27) * 1.27

**Fig. 15.** Illustrations for the Food Security Lesson.
Author: R.A. Rodriges Zalipynis, Source: https://www.vldb.org/pvldb/vol15/p3622-zalipynis.pdf

## Topic 10
## Spatial Navigation Systems

State-of-the-art spatial navigation. Managing spatial data produced by navigation systems. What kind of spatial data do navigation systems produce? National spatial navigation systems. Devices that facilitate spatial navigation.

## Topic 11
## The Future of Spatial Data Management

Big spatial data. Systems for big spatial data. Earth remote sensing data and HPC simulation data. Hardware accelerators for spatial data management. New gadgets that generate spatial data. Edge and fog spatial data management. Research and development challenges in spatial data management. Promising research and development opportunities.

**Seminar/Practice**.
Discuss Big Spatial Data Types and Criteria:

| VOLUME | VARIETY | VELOCITY | VERACITY | VALUE | VARIABILITY |
|---|---|---|---|---|---|
| The volume of data and the number of distinct data sources | The types of data: structured, semi-structured, unstructured | The speed at which big data are generated | The degree to which big data can be trusted | The business value of the data collected | The ways in which the big data can be used and formatted |

The most exciting example of Big Spatial Data is Earth remote sensing data. For example, Maxar (former DigitalGlobe), collects about 80 terabytes of data on an average day and has accumulated over 100 petabytes of data in the Amazon Cloud [1] – **volume** and **velocity**.

Array data have multiple dimensions, e.g., 1D (a time series), 2D (a single band), 3D (a climate reanalysis for some variable), 4D (a weather forecast), and others. Over 140 formats exist to store array (raster) data. They also have diverse structure (e.g., DEM – Digital Elevation Models, Vertical Profiles, *N*-d arrays) – **variety.**

With array data, we can answer the following questions:

- **Emergencies:** which areas of the city will be flooded when the water level rises by 1 meter (use DEM and simulation)?
- **Precision Agriculture:** how to fertilize a field to increase crop yield?
- **Urban planning:** where exactly and what hygiene measures should be undertaken to improve air quality?

These and other applications are described in [2] – **value** and **variability.**

Earth remote sensing data are physical measurements. Recall the Federal Law 2017 (Forest Codex): The lessee of the forest area is obliged to attach to the forest use report materials of Earth remote sensing data, confirming the observance of the boundaries of the allocated area for felling – **veracity**.

[1] Maxar Talk, https://youtu.be/mkKkSRIxU8M
[2] ArcGIS interactive book, https://learn.arcgis.com/ru/arcgis-imagery-book/

## 6. Spatial Data

We use a rich collection of spatial data. All data are freely available. However, the instructor will provide data samples to students.

- OSM (Open Street Map), format: ESRI Shapefile
- Landsat, Sentinel, Terra/MODIS Earth remote sensing data
- Climate Reanalysis data, see http://www.wikience.org/data/climate-reanalysis/
- Cyclone tracks, see http://www.wikience.org/tropical-cyclones/tracks/
- World Bank data, see https://data.worldbank.org/
- OpenRouteService tracks, see https://openrouteservice.org/

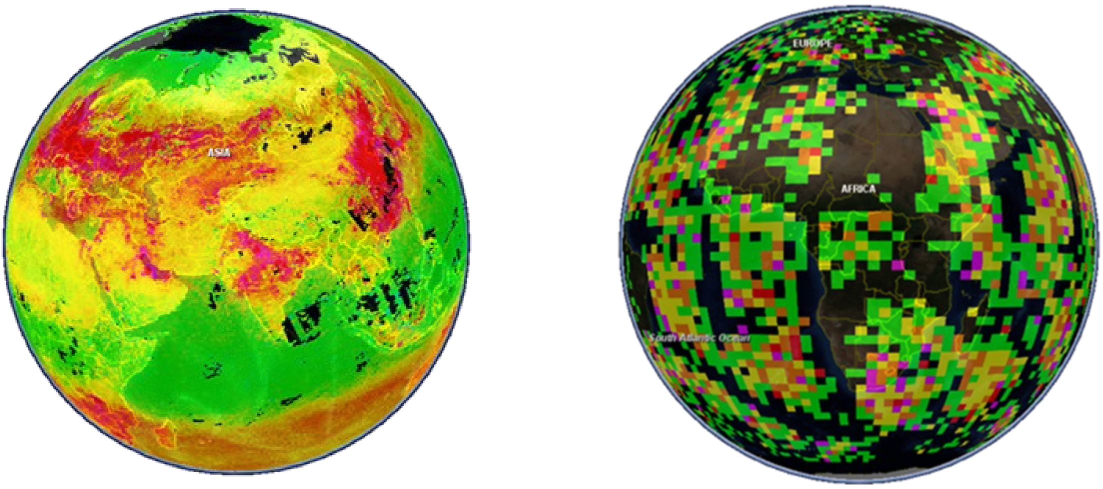
The formats of data are ESRI Shapefiles, GeoJSON, CSV, VRT, GeoTIFF, SAFE, HDF4, HDF5, NetCDF4, NetCDF5, and others.

- Landsat data are accessible from https://earthexplorer.usgs.gov/
- To access Sentinel data, it is recommended to use https://www.sentinel-hub.com/

Sentinel-Hub currently provides data using a set of OGC (Open Geospatial Consortium) protocols: WMS, WCS, WMTS, WFS. It is possible to use these protocols for the home assignment within the "Spatial Data Communication" topic.

We also use the following air quality data collection (please, also see Fig, 16):

| Component | Bias and units | | Satellite (instrument) | Spatial resolution | Starting date |
|---|---|---|---|---|---|
| Sulfur dixide ($SO_2$) | 5 | DU | Aura<br>(OMI) | 0,25° × 0,25°<br>27,5 × 18 km | 01.10.2004 |
| Nitrogen dioxide ($NO_2$) | – | | | | |
| Ozone ($O_3$) | 6 | | | | |
| Aerosol optical thickness, AOT (for 0,55 μm) | 15% | – | Terra<br>(MODIS) | 1,0° × 1,0°<br>110 × 72 km | 02.03.2000 |
| Ozone ($O_3$) | – | DU | | | |
| Carbon oxide (CO) | 10% | DU | Terra<br>(MOPITT) | 1,0° × 1,0°<br>110 × 72 km | 03.03.2000 |
| Carbon dioxide ($CO_2$) | 2 | ppm | Aqua<br>(AIRS) | 2,0° × 2,5°<br>220 × 144 km | 01.01.2004 |



Nitrogen dioxide ($NO_2$)

Carbon dioxide ($CO_2$)

**Fig. 16.** Examples of Air Quality Data Samples Visualized in 3D using Climate Wikience
Author: R.A. Rodriges Zalipynis, Source: http://www.wikience.org/data/air-quality-satellite/

## 7. Bibliography

### Recommended Core Bibliography

- Hsu, Leo S., and Regina Obe. *PostGIS in action*. Simon and Schuster, 2021.
- Schildt, Herbert. "The complete reference Java." (2020).
- Schönig, Hans-Jürgen. *Mastering PostgreSQL 13: Build, administer, and maintain database applications efficiently with PostgreSQL 13*. Packt Publishing Ltd, 2020.
- Wegmann, Martin, Benjamin Leutner, and Stefan Dech, eds. Remote sensing and GIS for ecologists: using open source software. Pelagic Publishing Ltd, 2016.
- Duncan, Scott. "Practical SQL." *Software Quality Professional* 20.4 (2018): 64-64.
- Ramon Antonio Rodriges Zalipynis. Array DBMS: Past, Present, and (Near) Future. PVLDB, 14(12): 3186–3189, 2021.
- Ramon Antonio Rodriges Zalipynis. ChronosServer: real-time access to "native" multi-terabyte retrospective data warehouse by thousands of concurrent clients, Informatics, cybernetics and computer engineering, Vol. 14 (188): 151–161, 2011.
- Ramon Antonio Rodriges Zalipynis, Terlych N. WebArrayDB: A Geospatial Array DBMS in Your Web Browser. PVLDB, 15(12): 3622–3625, 2022.

### Recommended Additional Bibliography

- Zurbarán, Mayra, et al. *PostGIS Cookbook: Store, organize, manipulate, and analyze spatial data*. Packt Publishing Ltd, 2018.
- Bush, Josephine. *Learn SQL Database Programming: Query and manipulate databases from popular relational database servers using SQL*. Packt Publishing Ltd, 2020.
- Chamberlin, Don. "SQL++ for SQL Users: A Tutorial." *Couchbase Inc.: Santa Clara, CA, USA* (2018).
- Badia, Antonio. *SQL for data science: data cleaning, wrangling and analytics with relational databases*. Springer Nature, 2020.
- Date, Chris J. *Database design and relational theory: normal forms and all that jazz*. Apress, 2019.
- Liang, Shunlin, and Jindi Wang, eds. *Advanced remote sensing: terrestrial information extraction and applications*. Academic Press, 2019.
- Ganapathy, Apoorva, et al. "HTML Content and Cascading Tree Sheets: Overview of Improving Web Content Visualization." *Turkish Online Journal of Qualitative Inquiry* 12.3 (2021): 2428-2438.
- Haverbeke, Marijn. *Eloquent JavaScript: a modern introduction to programming*. No Starch Press, 2018.
- Dombrovskaya, Henrietta, Boris Novikov, and Anna Bailliekova. *PostgreSQL Query Optimization*. Apress, 2021.
- Siahaan, Vivian, and Rismon Hasiholan Sianipar. *POSTGRESQL FOR JAVA GUI: Database, Cryptography, and Image Processing*. SPARTA PUBLISHING, 2019.
- Shaik, Baji, and Avinash Vallarapu. *Beginning PostgreSQL on the Cloud: Simplifying Database as a Service on Cloud Platforms*. Apress, 2018.

# HOME ASSIGNMENT EXAMPLE

**(one of the course assessment elements)**

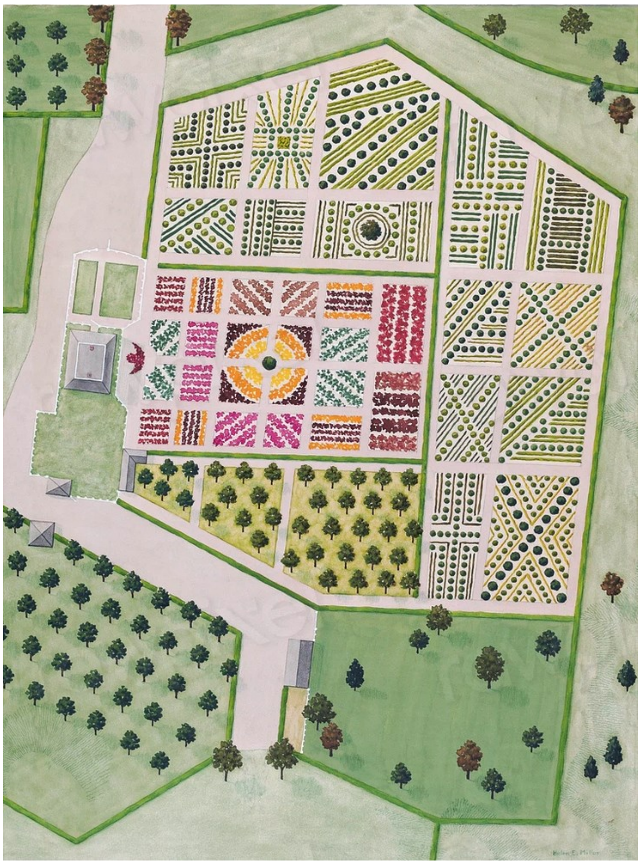
**The task "Map of the Area"**

**Motivation**. You need to consolidate your knowledge of the international standard for defining geometries (vector data): WKT (Well-Known Text). WKT is used in many spatial databases and libraries. With this standard, you can work with spatial data in almost any database management system or software library.

You will need to upload the solution to Smart LMS.
**Project in Smart LMS**: <NAME> (see below which files need to be uploaded to Smart LMS)
**Deadline**: <DEADLINE DATE> 23:59 UTC

**Part 1**. Defining Geometries

Please, use the image below or find your own. The image must contain objects in the area (for example, buildings/objects in a park that can be defined by WKT).



Nicholas Bayard Estate (1936) by Helen Miller. Original from The National Gallery of Art. Digitally enhanced by rawpixel.

Source:
Free to use and download under the CC0 license:
https://www.rawpixel.com/image/3243697/free-illustration-image-map-vintage-public-domain-maps

Use a blank sheet of paper. Draw a Cartesian coordinate system on the sheet and objects from the selected area on it. You need to represent a part of the area image using geometries in the WKT standard, while you must have at least:

- 3 points (POINT)
- 3 objects with several points (MULTIPOINT)
- 3 polygons with holes inside (POLYGON)
- 2 polygons with polygons inside (MULTIPOLYGON)

- 2 lines with multiple segments (LINESTRING)
- 1 MULTILINESTRING
- 1 geometry collection with at least 3 different geometry types inside

Please:

- Use UTM 37N as your spatial reference system for defining geometries (units – meters). This will simplify your work compared to EPSG 4326 (units – degrees).
- Do not get coordinates from epsg.io or other service.
- Near each point, put its coordinates, obtained approximately and manually (use only integer coordinates for points to simplify your work).

The drawing must be in graphic format; it is better to draw it by hand. You need to use the geometries in the drawing that were discussed in the lecture: Point, Polygon, LineString, and others (see above). You can imagine a house to be a Polygon, a statue to be a Point, etc.

**Part 2**. Spatial SQL

Ingesting geometries into PostgreSQL/PostGIS. Create one (1) file **task_01.sql** in which:

- Database Task01 is created in PostGIS
    - CREATE DATABASE
- A table is created (CREATE TABLE) Objects with fields:
    - identifier (primary key)
    - geometry
    - name
- The table is populated (INSERT INTO)
    - by the geometries created in Part 1with their names

Create a table such that only geometries in UTM 37N can be inserted into the table.

**Part 3**. PostGIS Access

Use the Java programming language (or another language of your choice) to access the created database and table in Part 2. After connecting to the database, run the following queries:

- The user enters 1
    - Print all geometries to the console, the coordinates must be in the human-readable WKT format
- The user enters 2 and number1, number2, number3
    - Print all geometries to the console that are within a distance from the point POINT(number1 number2) no further than number3 meters
- The user enters 3 and a geometry in the WKT format
    - Insert (add) geometry to an existing table, the name can be left blank

**Uploading the Solution to Smart LMS**

One file needs to be uploaded: an archive (e.g., ZIP archive) containing 3 files:

- Solution for Part 1: photo or scan of the sheet of paper with Cartesian coordinates and objects (geometries) in UTM 37N coordinates
    - note: one file (photo or scan) in JPEG or PNG format
- Solution for Part 2: one file **task_01.sql**
- Solution for Part 3: one file **task_01.java** (or with another extension depending on the selected programming language)