DuckDB 空间数据管理指南

从 SQL 基础到高级地理空间分析

吴秋生

DuckDB 空间数据管理指南

从 SQL 基础到高级地理空间分析

吴秋生

2026

Contents

前言

引言

在这个日益数据驱动的世界中，有效管理和分析空间信息的能力变得至关重要。从城市规划和环境监测到物流和个性化位置服务，地理空间数据构成了众多影响我们日常生活的应用的基础。然而，处理空间数据通常被视为一个专业且复杂的领域，需要复杂的工具和陡峭的学习曲线。

DuckDB 的出现改变了这一切。

DuckDB 是一个创新的分析型数据库，专为高效性和易用性而设计。作为一个进程内 OLAP（联机分析处理）数据库，它直接在您的应用程序内运行，无需单独的服务器部署和繁琐的配置。这种嵌入式特性，加上其列式架构和向量化执行引擎，使 DuckDB 在处理大型数据集的分析查询时表现出色。DuckDB 最初因其通用数据处理能力而广受欢迎，其快速扩展的生态系统（尤其是空间数据扩展）代表了一次变革性的转变。

本书《***DuckDB* 空间数据管理指南：从 *SQL* 基础到高级地理空间分析**》旨在揭开地理空间数据的神秘面纱，展示 DuckDB 如何以前所未有的简便性和速度，让每个人（从数据分析师和科学家到开发人员和 GIS 专业人员）都能利用其强大功能。我们相信，强大的空间分析应该是人人可及的，而不应局限于昂贵的专业软件或复杂的编程语言。借助 DuckDB，这种可及性成为现实。

我们的旅程从空间数据的基本概念及其表示方式开始，为使用 SQL 处理点、线和多边形奠定坚实的基础。随着学习的深入，您将发现 DuckDB 的原生地理空间功能（通过其兼容 PostGIS 的扩展增强）如何通过优雅的 SQL 查询实现复杂的操作，如空间连接、缓冲区分析和最近邻搜索。我们将探索各种实际应用，演示如何加载、转换、分析和可视化空间数据集，使您能够从地理信息中提取有意义的洞察。

无论您是想将空间分析集成到数据管道中、执行快速的临时地理空间查询，还是开发交互式的位置感知应用程序，本书都将作为您的全面指南。我们将涵盖从设置 DuckDB 环境和导入各种空间文件格式（如 Shapefile、GeoJSON 和 GeoParquet）到执行复杂分析任务和与可视化工具集成等主题。

我们的目标不仅仅是教您语法，而是培养您对这些工具和技术为何强大的理解。在本书结束时，您将能够熟练地使用 DuckDB 作为空间数据管理和分析的首选引擎，为您的项目解锁新的可能性，并使您能够做出明智的、具有空间意识的决策。

请与我们一起深入探索 DuckDB 分析能力与丰富地理空间数据世界的精彩交汇点。可及的空间分析的未来已经到来，它运行在 DuckDB 上。

本书适合谁

本书专为那些正在应对现代空间数据分析复杂性的人设计。如果您曾经花费数小时等待空间连接完成、难以将大型地理数据集加载到内存中，或者希望有一种更简单的方式将 SQL 的强大功能与空间操作结合起来，那么这本书正适合您。

如果您是以下人员，将获得最大价值

GIS 专业人员：对桌面软件处理大型数据集的局限性感到沮丧。您熟悉 QGIS 或 ArcGIS，但需要分析数百万个要素、处理大量 GPS 轨迹，或将空间分析集成到自动化工作流程中。

数据科学家或分析师：经常遇到位置数据。您熟悉 Python 和 pandas，但空间数据常常让您感到困惑。您希望将地理维度纳入分析，而无需深入学习复杂的 GIS 软件。

软件开发人员：正在构建包含空间功能的应用程序。您需要快速的空间查询，希望避免繁重的数据库基础设施，并且更喜欢使用熟悉的 SQL 而不是专门的空间库。

研究人员或学者：从事地理学、环境科学或城市规划等领域的研究。您的研究涉及大型空间数据集，需要可重复、可扩展的分析方法，以适应不断增长的数据量。

商业智能专业人员：处理基于位置的业务数据。无论是门店位置、配送路线、客户区域还是房地产投资组合，您都需要将业务指标与空间洞察相结合。

基本前提条件

您应该熟悉以下内容：

- **Python 编程**：理解变量、函数以及如何导入库（不需要专家水平）
- **数据分析概念**：筛选记录、聚合数据和连接表
- **SQL 基础**：熟悉 SELECT、WHERE 和 GROUP BY 子句（我们将介绍空间方面的内容）
- **空间数据基础**：理解数据具有位置属性（经纬度、投影）

有帮助的背景知识（非必需）

- 使用 pandas、GeoPandas 或 Jupyter notebooks 的经验
- 之前接触过数据库或数据仓库
- 熟悉 GIS 软件（QGIS、ArcGIS、PostGIS）
- 了解空间文件格式（GeoJSON、Shapefiles、Parquet）

如果您是 Python 编程新手

如果您是地理空间 Python 编程的新手，以下书籍为 GIS 基础概念和 Python 编程提供了很好的入门介绍：

Wu, Q. (2025). *Introduction to GIS Programming: A Practical Python Guide to Open Source Geospatial Tools*. Independently published. ISBN 979-8286979455. https://www.amazon.com/dp/B0FFW34LL3

本书内容概览

本书提供了从 SQL 基础到高级地理空间分析的系统化学习路径，通过实际案例帮助您掌握实用技能。每章都从简单查询逐步过渡到复杂的空间分析，逐步提升您在现代地理空间数据管理方面的专业能力。

第一部分：DuckDB 基础 （第 *1-3* 章）

掌握所有后续内容的基本概念：

- **第 1 章：DuckDB 入门**：安装、初始查询，以及了解 DuckDB 如何革新空间分析。
- **第 2 章：空间分析的 SQL 基础**：针对空间数据的筛选、聚合、连接和查询优化的关键 SQL 模式。
- **第 3 章：DuckDB Python 集成**：将 SQL 的强大功能与 pandas 的灵活性相结合，打造无缝的空间分析工作流程。

完成第一部分后，您将能够自信地查询空间数据集，并将 DuckDB 集成到任何基于 Python 的分析管道中。

第二部分：空间数据操作 （第 *4-10* 章）

深入了解核心空间工具包，涵盖从数据加载到高级分析的所有内容：

- **第 4 章：加载空间数据格式**：导入各种格式，包括 CSV 坐标、来自 API 的 GeoJSON、大型 Shapefile 和云托管的 GeoParquet。
- **第 5 章：导出和转换空间数据**：将结果转换为利益相关者所需的任何格式。
- **第 6 章：几何操作和函数**：使用 SQL 函数创建、测量和转换空间要素。
- **第 7 章：空间查询和关系**：掌握表连接的空间等价物：包含、相交和邻近关系。
- **第 8 章：高级空间连接**：按位置而非 ID 组合数据集，这是空间分析的精髓。
- **第 9 章：交互式数据可视化**：生成引人注目的地图和图表，有效传达数据的空间叙事。
- **第 10 章：矢量切片与 PMTiles**：部署能够流畅处理数百万要素的交互式地图。

完成第二部分后，您将能够熟练管理任何空间数据格式，执行复杂操作，并创建专业级可视化。

第三部分：实际地理空间分析 （第 *11-14* 章）

探索四个使用大规模真实数据集的综合案例研究：

- **第 11 章：分析美国国家湿地清查数据**：对全美 50 个州进行环境分析，处理数百万个湿地多边形。
- **第 12 章：分析全球建筑物轮廓数据**：使用 Overture Maps 的全球建筑物轮廓数据分析城市数据。
- **第 13 章：分析纽约市出租车行程数据**：从数亿次出租车行程中发现时空模式，揭示城市出行的洞察。
- **第 14 章：使用 Voilà 和 Solara 开发交互式仪表板**：构建和部署 Web 应用程序，使您的分析成果可供利益相关者访问。

完成第三部分后，您将拥有展示高级空间分析能力的优质项目作品集。

贯穿全书的主题

- **大规模性能**：无论处理数千还是数百万空间要素都有效的技术。
- **云原生工作流程**：直接从 S3 处理数据，与现代数据栈无缝集成。
- **可重复分析**：可版本控制并部署到生产环境的可共享代码和方法。
- **真实数据挑战**：解决投影问题、缺失值和数据质量问题。
- **集成模式**：将 DuckDB 与更广泛的 Python 地理空间生态系统相结合，以增强功能。

本书的独特之处

与理论讨论或特定工具教程不同，本书强调_解决实际问题_。每种技术都植根于实际的分析挑战，使用真实数据集进行演示，并清晰解释何时以及为何使用它。

如何充分利用本书

为了最大化您使用本书的学习体验，请考虑以下建议：

建立适当的开发环境：按照第 1 章的说明安装 Python 和所需的库。配置良好的环境将在整个学习过程中为您节省时间和减少挫折。考虑使用 conda 或 uv 来管理您的 Python 包，因为这简化了地理空间库的安装。

跟随代码示例练习：本书设计为互动式的。不要只是阅读代码；亲自输入、运行并尝试修改。理解来自实践，每个示例都在培养您以后需要的技能。

完成练习：每章都包含旨在巩固所学概念的练习。这些不是可选的附加内容；它们是学习过程的重要组成部分。从指导练习开始，然后用自己的项目挑战自己。

使用真实数据：虽然本书为示例和练习提供了数据集，但请尝试将这些技术应用于您自己领域或兴趣的数据。这将帮助您理解概念如何应用于实际场景，并建立对自己能力的信心。

构建项目：随着您在本书中的进展，考虑开展一个您感兴趣的个人项目。这可以是分析您研究中的数据、为您的社区创建地图，或解决您在工作中遇到的问题。

对自己有耐心：编程可能令人沮丧，尤其是在学习时。预期会遇到错误、花时间调试，偶尔会感到困惑。这是正常的，也是学习过程的一部分。需要时休息一下，记住专业技能是通过持续练习逐渐发展的。如果遇到困难，不要犹豫在本书的 GitHub 仓库中寻求帮助。

持续练习：本书中的技能需要定期练习来保持和发展。定期安排时间进行地理空间编程项目，即使是小项目也可以。

本书使用的约定

本书使用了几种约定来帮助您浏览内容并理解代码示例：

代码格式：所有 Python 代码都以等宽字体显示在代码块中。当代码出现在正文中时，格式为 `这样`。文件和目录名也使用等宽字体格式。

代码示例：大多数代码示例都是完整且可运行的。它们包含解释所演示的关键概念和技术的注释。可能会包含行号以供正文参考。

```
# 这是一个代码块示例
import leafmap
m = leafmap.Map()
m.add_basemap("OpenTopoMap")  # 向地图添加底图
m
```

SQL 风格指南：为了一致性和可读性，SQL 示例遵循以下模式：

- **关键字大写**：`SELECT`、`FROM`、`WHERE`、`JOIN`
- **函数名保留大小写**：`ST_Area()`、`read_csv_auto()`
- **表名和列名小写**：`cities`、`population`
- **使用缩进提高可读性**：多行查询经过格式化以便清晰阅读

```
SELECT name, ST_Area(geometry) as area
FROM neighborhoods
WHERE borough = 'Manhattan'
ORDER BY area DESC;
```

命令行指令：在命令行或终端输入的命令以 `$` 提示符显示（不要输入 `$` 符号本身）：

```
$ pip install leafmap
$ python script.py
```

下载代码示例

本书的所有代码示例、数据集和补充材料都可在 GitHub 上免费获取：

https://github.com/giswqs/duckdb-spatial

要下载材料，您可以使用以下方法之一：

- **克隆仓库**（如果您已安装 Git）：

```
$ git clone https://github.com/giswqs/duckdb-spatial.git
```

- **下载 ZIP 文件**（如果您不想使用 Git）：
  - 访问 GitHub 仓库页面
  - 点击绿色的 **Code** 按钮
  - 选择 **Download ZIP**
  - 将文件解压到您喜欢的位置
- 如果您只需要特定示例，可以通过 GitHub 界面**在线浏览单个文件**

仓库会定期更新修正、改进和额外示例。请定期查看更新，或在 GitHub 上 **watch** 该仓库以获得变更通知。

如果您在代码中发现错误或有改进建议，请在 GitHub 上提交 issue 或 pull request。社区贡献有助于使这个资源对每个人都更好。

视频教程和补充资源

除了书面内容，本书还配有一系列全面的视频教程，讲解关键概念并提供额外示例：

https://tinyurl.com/duckdb-spatial-videos

视频旨在补充而非替代书面材料。它们对以下情况特别有帮助：

- 视觉型学习者，通过观看代码编写和执行获益
- 通过多种解释理解复杂概念
- 学习开发工作流程和最佳实践
- 了解如何处理问题和调试

播放列表按照本书的结构组织。您可以在阅读本书时按顺序观看，或根据需要跳转到特定主题。

这些视频是我在 2023 年秋季于田纳西大学教授**空间数据管理**[1] 课程时创建的。虽然课程已经结束，但视频仍然具有参考价值，可以作为本书的参考资料。未来还会添加更多视频。

[1] https://geog-414.gishub.org

社区和反馈

我欢迎读者的反馈、问题和建议。您的意见有助于改进本书，使其对地理空间编程社区更有用。

书籍相关问题和讨论：

- GitHub Issues：https://github.com/giswqs/duckdb-spatial/issues
- GitHub Discussions：https://github.com/giswqs/duckdb-spatial/discussions

特别有帮助的反馈类型：

- 文本或代码中的错误或不清楚的解释
- 对额外示例或用例的建议
- 新主题或章节的想法
- 不同操作系统或库版本的兼容性问题报告
- 您如何应用本书技术的成功案例

致谢

本书的完成要感谢许多人的贡献以及更广泛的开源地理空间社区。

首先，我要感谢 **DuckDB 开发团队** 创建了如此出色的数据库系统。他们对性能、简洁性和开源原则的承诺使空间分析对更广泛的受众变得可及。特别感谢为 DuckDB 空间功能做出贡献的团队成员。

我感谢田纳西大学的 **同事和学生们**，他们通过 **空间数据管理** 课程提供了反馈、测试了示例，并帮助完善了内容。他们的问题和见解使本书更加完善。

开源地理空间社区 值得特别认可。像 GDAL、GeoPandas、Shapely 等项目以及无数其他项目构成了使现代空间分析成为可能的基础。这个社区的协作精神继续激励着我的工作。

感谢 **早期读者** 对草稿章节提供的反馈，并帮助确定需要澄清或改进的地方。他们多样化的视角（从经验丰富的 GIS 专业人员到数据科学新手）帮助确保本书服务于其目标受众。

我要感谢 **我的家人** 在我花费许多晚上和周末写作时的耐心和支持。他们的理解和鼓励使这个项目成为可能。

最后，感谢 **您**，读者，对空间数据分析和开源工具的兴趣。正是像您这样的从业者推动了创新，使地理空间领域成为一个令人兴奋的工作场所。

如果本书能帮助您更有效地解决空间问题，那么所有的努力都是值得的。

关于作者

吴秋生博士是田纳西大学诺克斯维尔分校地理与可持续发展系副教授，同时也是亚马逊学者。吴博士的研究专注于通过云计算和 GeoAI 推进开源地理空间分析。他是多个广泛使用的开源 Python 包的创建者和维护者，包括 Geemap[2]、Leafmap[3]、SAMGeo[4] 和 GeoAI[5]，这些工具将基于云的地理空间平台与 AI 驱动的分析和可视化相集成。吴博士的工作连接了遥感、地球观测和人工智能，使大规

[2] https://geemap.org
[3] https://leafmap.org
[4] https://samgeo.gishub.org
[5] https://opengeoai.org

模地理空间数据对全球的研究人员、教育工作者和从业者更加可访问、可重复和智能化。他的开源项目可以在 GitHub 上找到：https://github.com/opengeos。

许可和版权