Narrative SQL

Crafting Data Analysis Queries That Tell Stories

Hamed Tabrizchi

Apress®

Narrative SQL

Crafting Data Analysis Queries That Tell Stories

Hamed Tabrizchi

Apress®

Narrative SQL: Crafting Data Analysis Queries That Tell Stories

Hamed Tabrizchi
Department of Computer Science, Faculty of Mathematics, Statistics, and Computer Science,
University of Tabriz
Tabriz, Iran

ISBN-13 (pbk): 979-8-8688-1559-1 ISBN-13 (electronic): 979-8-8688-1560-7
https://doi.org/10.1007/979-8-8688-1560-7

Managing Director, Apress Media LLC: Welmoed Spahr
Acquisitions Editor: Shaul Elson
Development Editor: Laura Berendson
Coordinating Editor: Gryffin Winkler
Copy Editor: Kezia Endsley

Cover image by Christian Horz from stock.adobe.com

Distributed to the book trade worldwide by Springer Science+Business Media New York, 1 New York Plaza, New York, NY 10004. Phone 1-800-SPRINGER, fax (201) 348-4505, e-mail orders-ny@springer-sbm.com, or visit www.springeronline.com. Apress Media, LLC is a Delaware LLC and the sole member (owner) is Springer Science + Business Media Finance Inc (SSBM Finance Inc). SSBM Finance Inc is a **Delaware** corporation.

For information on translations, please e-mail booktranslations@springernature.com; for reprint, paperback, or audio rights, please e-mail bookpermissions@springernature.com.

Apress titles may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Print and eBook Bulk Sales web page at http://www.apress.com/bulk-sales.

Any source code or other supplementary material referenced by the author in this book is available to readers on the Github repository. For more detailed information, please visit https://www.apress.com/gp/services/source-code.

If disposing of this product, please recycle the paper

To my adorable father, Hamid, my kind-hearted mother, Soheyla, and my wonderful brother, Mohammad—each of whom helped me understand the true value of life.

To Shaul and Gryffin, who believed in me and stood by me every step of the way.

Table of Contents

About the Author

Hamed Tabrizchi is an experienced data analyst and engaging storyteller who has more than five years of experience turning complex data into compelling narratives. Passionate about educating others, Hamed lectures at universities, leads workshops, and contributes to leading scientific journals. As a result of his observations in the professional community, he decided to write this book to fill a void he observed—the need for a resource that weaves SQL technicalities with narratives to empower analysts in delivering insights that resonate and drive action.

About the Technical Reviewer

Alexander Arvidsson is the chief technology officer at Analytics Masterminds, where he spends his days helping clients of all shapes and sizes take better care of—and make more sense of—their data.

He has spent the last 25 years poking around with data, databases, and related infrastructure services such as storage, networking, and virtualization, occasionally emerging from the technical darkness to attend a *Star Wars* convention somewhere in the world.

He is a long-time data platform MVP, frequent international speaker, podcaster, Pluralsight author, blogger, and a Microsoft Certified Trainer, focusing on the Microsoft data platform stack.

Acknowledgments

Writing a book is often seen as a solitary endeavor, but this one would not exist without the support, encouragement, and generosity of so many people.

My deepest thanks go to Apress for placing their trust in me as a first-time author and for accepting my proposal to write this book. I would like to thank Shaul Elson for his time, guidance, and belief in me. I'm also sincerely grateful to Gryffin Winkler for his dedication and support throughout this process. Special thanks to Alexander Arvidsson, whose valuable feedback, sharp eye, and honest critique—shared chapter by chapter—have shaped this work more than words can express.

This journey would not have been possible without the patience, love, and unwavering belief of my family.

Finally, to the readers—thank you for making space in your analytical and curious minds for these words.

Introduction

In the past decade, data analysis and SQL have played a central role in my professional career. My passion for query writing began during my bachelor's studies, where I completed a database course with full marks. As a result of this achievement, I was given the opportunity to become a teaching assistant the following semester, where I gained experience writing queries and explaining them to students. Despite speaking with a trembling voice at first, this role helped me gain confidence in technical communication and public speaking, and enhanced my advanced query-writing skills. These early experiences laid the foundation for my current expertise in SQL and data analysis, fundamental to my next career accomplishment.

After a year, I started working for a technology company, where I encountered more complex challenges. As a data analysis intern, two of the challenges I encountered included the lack of neatly organized data and the difficulty of collaborating. It was difficult and very different from teaching or solving textbook exercises to coordinate projects among programmers, query writers, UI/UX designers, and the other members of the team. Despite all the challenges, I was motivated day by day to gain experience and skills from my colleagues and improve my analytical skills to become a data analyst who has an insightful perspective.

Throughout the years, I have dealt with a number of projects and gained deeper insights and better analytical skills from the past. One day I decided that I had an insight that I could share with those who are interested in data analysis and query writing. So, I decided to write this book to teach what formed this perspective within me, in as much detail as I could. The core concept of this book is SQL query writing, which is at the core of my day-to-day activities, whether as a data analyst, a university lecturer, or a data team leader.

I believe that at the present time, people who are able to shape information into compelling stories hold an advantage in the ever-increasing world of data. Due to this belief, narrative SQL emerged, which is the idea that learning SQL should not be like learning a machine's language, but should instead feel like mastering a language of communication.

This book is for the curious analyst, the thoughtful developer, and the future storyteller of data. This book is designed to provide you with clear and creative guidance regardless of whether you are just beginning your journey into databases or want to improve your proficiency in SQL.

Using a narrative structure, this book begins with the basics—simple `SELECT` statements, filters, and `JOIN`s. In the subsequent chapters, this book explores queries that transform raw data into rich insights, including aggregations, subqueries, conditional logic, and more. With the SQL queries and stories provided, this book is not just a reference guide; it's a companion for your data journey, helping you think narratively, write clearly, and analyze clearly.

Each chapter explores stories that introduce concepts and skills toward mastery of powerful SQL tools, including window functions, subqueries for dynamic data manipulation, conditional logic with complex queries, and even optimization strategies based on indexes and views. Upon completion of this book, you should be able to tackle a wide range of data analysis challenges by writing SQL queries. The last few chapters of this book cover advanced topics such as tuning performance, optimizing scripts, and analytical storytelling with window functions, giving your narratives depth and precision. In this book, you will find both inspiration and practical skills—and when you close the last chapter, you will be prepared to tell your own powerful data stories.

Finally, it should be noted that all queries presented in this book have been developed and thoroughly tested on PostgreSQL 14.17, the enterprise-grade open-source relational database system known for its robustness, extensibility, and SQL compliance. Although the fundamental concepts of PostgreSQL should apply to all PostgreSQL versions, specific syntax, performance characteristics, or feature availability might be different.

The complete collection of queries, including stories and examples, can be accessed via the publisher's GitHub repository at `https://github.com/Apress/Narrative-SQL`. Throughout this repository, all queries are conveniently organized and categorized by chapter, allowing you to find and execute examples relevant to specific sections conveniently.

Chapter Overviews

Chapter 1: The Storyteller's Database

The purpose of this chapter is to provide a foundation for your journey into the world of data and narrative. This chapter introduces databases as storytelling tools, illustrating how narrative structures and relational models can aid in making data meaningful. As you go through this chapter, you are provided with all the information you need to get started on this journey by setting the foundation for the art and science of data storytelling. You will learn that SQL is not only capable of querying data, but it can also tell compelling stories. In the next step, you will begin to explore SQL's complexities in greater detail after setting up the storytelling environment.

Chapter 2: Starting with SELECT

In this chapter, you learn how to extract and explore basic data from tables using the `SELECT` statement. Using SQL's most commonly used command, you can retrieve and manipulate data effectively. This requires you to learn how to write precise `SELECT` statements in order to retrieve information that is needed for your narratives. This will set the stage for more advanced data manipulation and analysis techniques.

Chapter 3: Filtering Facts with WHERE

In this chapter, you discover another SQL command that is frequently used to refine data retrieval by using `WHERE` conditions, comparisons, and logical operators. This requires creating precise `WHERE` statements that filter data based on specific criteria. This will enable you to refine your datasets and extract even deeper insights, which in turn helps you tell richer stories with your data.

Chapter 4: Complex Characters with JOINs

The purpose of this chapter is to explore how to connect multiple tables using `JOIN`s, creating richer data narratives based on different sources of data. `JOIN` operations, which are fundamental to combining data from multiple tables, are discussed. As you become proficient in this operation, you will be able to create complex queries that provide deeper insights and more comprehensive analyses of your records.

Chapter 5: Aggregating Acts

The purpose of this chapter is to introduce aggregate functions such as `COUNT`, `SUM`, `AVG`, `MIN`, and `MAX`, which can be used to summarize and analyze grouped data. In SQL, an aggregate act is the application of aggregate functions to grouped data subsets. These actions enable SQL to extract useful summary and statistical information from data for analysis and decision-making.

Chapter 6: Ordering the Plot with ORDER BY and LIMIT

In this chapter, you learn how to sort query results and limit output in order to improve readability and performance. The focus is on sorting and filtering query results efficiently. Once you have mastered this operation, it will be possible to organize data meaningfully. To focus on the most relevant data points, you can prioritize key information and limit the results to the most relevant data points.

Chapter 7: Dynamic Dialogues with Subqueries

The purpose of this chapter is to present subqueries as powerful tools for nesting logic and constructing complex, layered data requests. This chapter explores the art of writing subqueries in order to add depth and dimension to data analysis. This chapter provides an overview of subqueries, their types, and narrative examples of their use in dynamic dialogues.

Chapter 8: Conditional Logic in Data Plotting

This chapter explains how SQL's conditional logic can be used to transform data analysis and visualization workflows to enable logic-based data visualization and transformation. You learn about conditional logic in SQL, categorize data, apply dynamic filtering to improve plot relevance for enhanced visualizations, create color-coded data for visualizations, aggregate data using conditional expressions, and handle missing data in visualizations. The focus of this chapter is not on how to visualize or plot data, but on the crucial process of preparing data. In this chapter, SQL is used to manipulate, clean, and structure data before it is visualized.

Chapter 9: Optimizing Your Script with Indexes and Views

This chapter sheds light on how indexes can be used to improve query performance and how views can be used to simplify logic. Optimizing SQL queries can significantly improve performance when dealing with large data volumes. Indexes and views are both powerful tools for achieving this type of optimization. An index enables the database engine to locate rows more efficiently, thereby reducing the need to scan entire tables in order to retrieve data. Alternatively, views simplify complex queries by storing reusable SQL logic, improving readability and maintenance.

Chapter 10: Analytics Alchemy: Turning Data into Gold

Turning data into gold with SQL requires mastering advanced analytical functions that help you extract deeper insights from raw data and transform them into compelling narratives. The SQL language provides powerful functions that can be used to transform raw data into compelling narratives. The chapter also discusses how raw data can be transformed into a compelling story and how recursive queries can be used to structure query logic effectively.

Chapter 11: The Grand Finale: Presenting Your Data Story

In this chapter, your journey is nearing its end. Through chapter-by-chapter learning, you learned how to extract deep insights and information from raw data to address complex and advanced analytical questions using raw data. This chapter summarizes the previous chapters and provides insight into presenting a narrative for data analysis.

Appendix A: SQL Syntax Reference Guide

This appendix provides a quick-access syntax guide for common SQL statements and clauses.

Appendix B: Glossary of Terms

This appendix defines key terms used throughout the book for quick reference and deeper understanding.

Appendix C: PostgreSQL Elements Reference

This appendix highlights PostgreSQL-specific features by providing a comprehensive alphabetical list of SQL statements, clauses, operations, and functions available in PostgreSQL.

CHAPTER 1

The Storyteller's Database

This chapter provides the basis of your journey into the world of data and narratives. Beginning with the basics of data, databases, and data analysis, this chapter explores Database Management Systems (DBMS) and SQL's pivotal role in navigating these repositories. Through an exploration of SQL commands and the use of data types, you can tell powerful stories. This chapter provides you with the essentials you need to get started on this journey. It covers the art and science of data storytelling.

Introduction to Data

In today's society, data is the foundation upon which everything is built, and it impacts every aspect of our lives. There are countless sources of data available to us, from weather patterns tracked by satellites to the number of steps you take each day. The term *data* refers to the qualitative or quantitative attributes of a variable. A great deal of data is collected, observed, or created for the purpose of analyzing it and making decisions based on it. It is possible to store structured or unstructured data, ranging from numbers, text, and multimedia to complex datasets used in computing and research.

In a nutshell, data is the raw material of information, the basis for understanding the world and making informed decisions. Data on its own is like a pile of unrefined oil. In spite of the fact that data has value, it's useless until it's processed and analyzed. There is a great deal of value in data because it is capable of revealing hidden patterns, trends, and insights. The ability to analyze data allows people to make better decisions, solve complex problems, and drive innovation in business. It is widely accepted that data has become a part of every aspect of our lives in the digital age, and that it is the basis of all decision-making across sectors such as the healthcare, finance, and technology industries. Table 1-1 provides five fundamental questions and answers when exploring data.

H. Tabrizchi, *Narrative SQL*, https://doi.org/10.1007/979-8-8688-1560-7_1

Table 1-1. *Five Fundamental Questions and Answers for Exploring Data*

#	Question	Answer
1	Why is data so valuable?	The use of data leads to decisions, innovation, and progress. Data contains the fundamental insights and evidence required to make informed decisions, resolve complex problems, and predict future trends in an era in which information is power.
2	Who relies on data?	All of us. Data is used in a wide range of things, from businesses using customer data to modify their products, to governments using data to plan policies, to scientists using research data to make discoveries. Many sectors and societies rely on data.
3	Where does data come from?	The world over. This includes countless devices and sensors in the vast expanse of the Internet and the billions of devices and sensors making up the Internet of Things (IoT). From the depths of the oceans using climate monitoring equipment, to the far reaches of space with satellites collecting data about our universe. Each source of data provides unique insights that contribute to our collective understanding.
4	When is data used?	Every day, continuously. Data is used constantly throughout our lives and is not limited to specific moments. Data is used to guide emergency responses and to facilitate financial transactions, and it is used to influence long-term policy decisions and scientific research. Data has relevance that extends across timescales, from the immediate to the generational.
5	What does data do?	In data, transformation occurs. Data has the power to change the world in many ways. These include informing policy, driving economic growth, advancing science and healthcare, improving education, and enriching cultural understanding. By analyzing data, we can uncover patterns, predict outcomes, personalize experiences, and foster innovation.

Data Analysis

Similar to oil in the 20th century—which powered economies, revolutionized transportation, and played a fundamental role in industrial advancement—data is the fundamental resource driving societal progress, economic growth, and innovation in

the 21st century. In the same way that oil must be extracted, refined, and distributed to use its energy, data must be collected, organized, and analyzed to unlock its full potential. Thus, the importance of controlling the extraction, refinement, and analysis of data in this era cannot be understated. The purpose of data analysis is to find useful information, provide insight into conclusions, and support decision-making through inspection, cleansing, transforming, and modeling the data.

Databases

Databases are structured collections of data that are stored electronically and accessed by a computer system. This system facilitates the efficient organization, management, and retrieval of data. A database is designed to handle large amounts of data, allowing users to add, modify, and query data quickly and securely. Databases support a variety of data types, including text, numbers, multimedia files, and more, organized in a manner that facilitates the analysis of business operations, the management of transactions, and decision making. Thousands of applications rely on databases, from the websites people visit daily to the financial systems that operate on a global scale.

There is more to databases than just storing data; they are intricately designed to organize information in a way that makes it easily accessible and useful. Organization is of considerable importance, as the value of data lies not only in its existence, but in the ability to retrieve and interpret it. In every sector of society—be it healthcare, education, business, or technology—databases help manage patient records, student information, financial transactions, and more.

In general, there are several types of databases, including structured, semi-structured, and unstructured. As a result of the need to optimize storage, retrieval, and analysis of data based on its nature, there is a relationship between data types and database types. Structured, semi-structured, and unstructured data all require different database features. For analysis, it is crucial to be aware of the distinctions between each, since each requires different tools and approaches in order to extract its insights.

In structured data, each piece of information is organized and formatted in a way that is easily searchable in databases and spreadsheets, and it is stored in predefined models or schemas. In this way, computers can perform efficient processing and analysis. Structured data can be easily accessed and analyzed through relational databases.

In the real world, data is not always stored in a structured form and may be unstructured or semi-structured. Unstructured data consists of everything from emails to videos to social media posts, often stored in non-relational (NoSQL) databases that handle diverse and dynamic datasets. Semi-structured data straddles the line, combining elements of both. Examples of semi-structured data are JSON and XML documents, which, although not fitting into traditional table schemas, have inherent structure that can be queried and analyzed.

Relational Databases vs Non-Relational Databases

There is a major distinction between relational and non-relational databases, each with its own characteristics, advantages, and applications. A relational database is based on the relational model of data. A table (relation), which consists of rows and columns, is used to organize data in this model. A row represents a unique record, and a column represents a field. The power of relational databases lies in their use of SQL (Structured Query Language) for data manipulation and retrieval, which allows high levels of flexibility and precision in querying the data. The most popular relational database management systems (RDBMS) are PostgreSQL, MySQL, Oracle Database, and Microsoft SQL Server. Alternatively, non-relational databases, known as *NoSQL* databases, emerged as a response to the limitations of relational models, especially in handling large volumes of unstructured or semi-structured data. As opposed to having fixed schemas, these databases are often capable of storing a variety of data types, such as documents, key-values, wide columns, and graphs. NoSQL databases are particularly well suited to applications that require rapid development, scalability, and the ability to handle a variety of data types. Among the most popular are MongoDB (document-based), Redis (key-value store), Cassandra (wide-column database), and Neo4j (graph database).

Transferring between structured, semi-structured, and unstructured data types is often driven by a variety of needs and challenges in data management and analysis. Each has its unique characteristics and optimal use cases. Thus, a variety of databases and database types exist, ranging from relational databases, which can handle structured data effectively through well-defined schemas and relationships, to NoSQL databases such as document, key-value, wide-column, and graph databases. They are each tailored to meet the specific requirements of unstructured or semi-structured data. As the title of the book indicates, the following chapters focus on structured data, exploring the expansive world of relational databases and the use of the SQL language to interact with them.

Exploring Relational Database Management Systems (RDBMS)

Relational databases are crucial throughout the entire process. As relational databases are structured, they ensure the integrity and consistency of data, which is crucial for data analysis. The powerful querying capability of SQL allows analysts to retrieve specific subsets of data quickly and efficiently from large databases. SQL's querying ability enables analysts to perform complex aggregations, joins, and filtering operations with ease, which are essential tasks in the data preprocessing and exploration phases. The purpose of this book is to teach you how to extract and analyze data stored in databases using SQL. Throughout the remainder of this chapter, you learn more and more about data analysis, databases, and other concepts, but you first need a better understanding of the data in order to be able to analyze it.

In most relational databases, SQL is used to query and manage the data. Due to SQL's powerful and flexible capabilities, it has become the standard language for relational database management systems (RDBMS). There are a number of popular relational database systems that use SQL, including MySQL, PostgreSQL, Oracle Database, Microsoft SQL Server, and SQLite. Due to the standard way in which SQL interacts with structured data, these systems can perform complex queries, update data, create and modify schemas, and manage database access more easily.

This book uses PostgreSQL as the basis for its SQL examples, which offers numerous advantages for readers who are eager to gain a deeper understanding of data analysis through the lens of relational databases. The PostgreSQL database system is well known for its robustness, open-source nature, and compliance with SQL standards, making it one of the most advanced and reliable relational database systems. Through its open-source model, users are not only able to access a high-quality database system without licensing fees, but also benefit from a vibrant developer community that is constantly enhancing its capabilities. In addition to complex SQL queries, foreign keys, triggers, views, and stored procedures, PostgreSQL supports a wide range of SQL functionality.

Databases in Data Analysis and Storytelling

Databases provide more than just a means of storing data; they are also instrumental in analyzing data and telling stories based on that data. In data analysis, data is examined, cleaned, transformed, and modeled in order to find useful information,

draw conclusions, and support decision-making. By structuring and organizing data, databases facilitate this process; they enable analysts to query and manipulate data effectively. On the other hand, storytelling involves using narratives to communicate information in an engaging and understandable manner. Data storytelling involves crafting narratives around insights in data in order to make complex information accessible and understandable. The definition of databases in the context of data storytelling could be expressed as a source of truth from which narratives are constructed in data storytelling. As a result of enabling the extraction of meaningful patterns and trends, databases make it easy for storytellers to tell narratives that are in tune with their audiences. In a nutshell, databases, by providing the raw material for these stories, lie at the heart of this process.

Diving into SQL

In the 1970s, the creation of SQL marked a pivotal moment in the evolution of data storage and retrieval. Over the decades, SQL evolved from a simple query language to a tool for professionals working with data. The journey of SQL began in the early 1970s at IBM, where researchers Donald D. Chamberlin and Raymond F. Boyce developed a prototype called SEQUEL (Structured English Query Language). This prototype was designed to manipulate and retrieve data stored in IBM's early relational database management system. The language was late renamed SQL to avoid brand-name issues. By the late 1970s and early 1980s, SQL had been adopted as the standard language for RDBMSs. Since then, SQL has undergone several revisions to include updated features and capabilities. These features include support for XML data, window functions, and expanding its utility and efficiency in managing diverse data types and complex queries. SQL allows users to interact with databases to perform operations such as querying, updating, inserting, and deleting data.

SQL Command Types: The Five Principles of Database Interaction

There are five distinct types of commands in SQL, each of which performs a specific function when it comes to managing and manipulating data. These categories are Data Definition Language (DDL), Data Manipulation Language (DML), Data Control Language (DCL), Transaction Control Language (TCL), and Data Query Language (DQL).

- **Data Definition Language (DDL)**: DDL commands are used to define, alter, and manage the schema and structure of database objects like tables, indexes, and views. These commands do not manipulate the data itself but instead shape the "containers" that hold the data, allowing for the creation and modification of database structures.
- **Data Manipulation Language (DML)**: DML commands are likely to be the most frequently used, as they deal directly with data manipulation within existing database structures. They enable users to insert, update, delete, and manage the database data.
- **Data Control Language (DCL)**: DCL commands are focused on permissions and access control for database objects. For security and confidentiality, these commands are crucial in multi-user databases.
- **Transaction Control Language (TCL)**: TCL commands manage the changes made by DML operations as transactions, which are either completely processed or not processed at all, ensuring data consistency and integrity. These commands allow users to commit or roll back changes to the database.
- **Data Query Language (DQL)**: DQL deals with retrieving data and is primarily represented by the `SELECT` command, which queries data from tables within a database. DQL allows users to specify exactly which data should be returned from the query, making it a powerful tool for extracting and analyzing information stored in the database.

Figure 1-1 illustrates an overview of the SQL command types, indicating each category along with its respective commands. This illustration serves as a guide, mapping out the distinct SQL command types. This visualization not only aids in understanding the functional divisions within SQL but also highlights the specific operations that can be performed within each category.

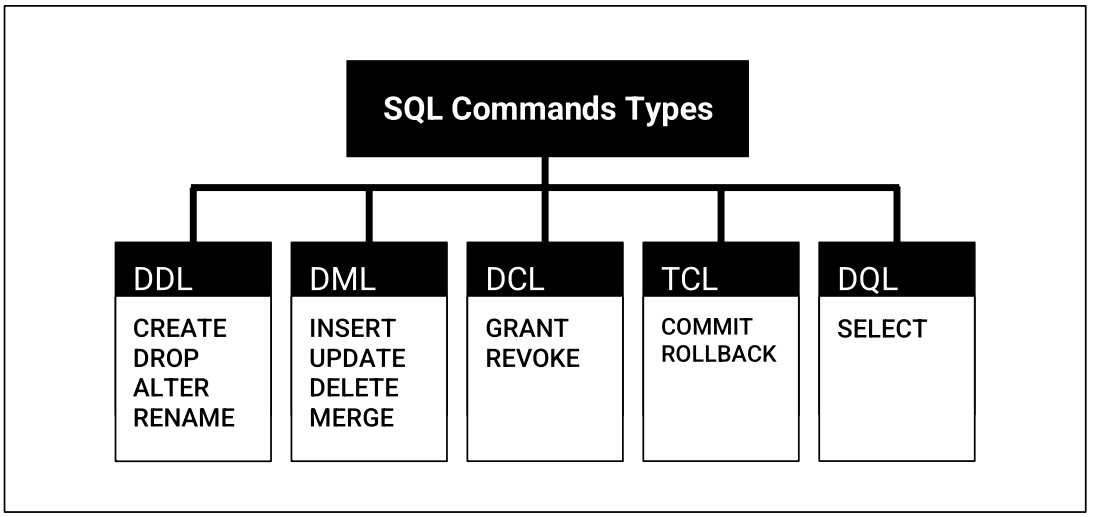

Figure 1-1. *SQL command types*

When it comes to data analysis, you will mainly be working with Data Manipulation Language (DML) and Data Query Language (DQL) commands. These two types of SQL commands are especially useful:

- Data Manipulation Language (DML):
    1. **Insight extraction**: DML commands are used to insert, update, delete, and manage data within database tables. The primary goal of data analysis is to extract insights from data rather than modify it.
    2. **Data preparation**: Before analyzing data, it often needs to be cleaned and preprocessed. DML commands like `UPDATE` can correct data errors, and `DELETE` can remove irrelevant or duplicate records. Data preparation is crucial for accurate analysis.

3. **Inserting data**: The `INSERT` command is useful for adding new data to the database, which might be needed for analysis. This could include new data points, calculated metrics, or results from previous analyses that you want to store for future use.

- Data Query Language (DQL):

1. **Data retrieval**: The essence of data analysis in SQL environments is to query the database for specific datasets. `SELECT` allows you to specify exactly which data to retrieve, including which tables to source from and under what conditions.

2. **Data aggregation and filtering**: `SELECT` queries can be augmented with clauses like `WHERE`, `GROUP BY`, and `HAVING`. They filter data, aggregate it (e.g., finding averages, sums, counts), and select data that meets certain conditions. These operations are fundamental to data analysis, enabling analysts to explore trends, patterns, and outliers in the data.

3. **Joining tables**: Data analysis often requires combining data from multiple tables to get a complete picture. The `SELECT` command can join tables based on specific criteria, enabling comprehensive analysis across diverse datasets.

- Both DML and DQL:

1. **Flexibility in data handling**: DML provides the flexibility to manipulate data as needed for analysis, ensuring the dataset is accurate and relevant. DQL offers the tools to dig into that data, pulling out the insights and information critical to informed decision-making.

2. **Basis for advanced analysis**: While other SQL commands focus on database structure and access control, DML and DQL are directly concerned with the data itself. Mastery of these commands allows analysts to extract and manipulate data.

3. **Data integrity**: While DML helps maintain the quality and relevance of the dataset, DQL ensures that the integrity of the data is preserved during analysis. By using DQL, analysts can perform read-only operations that don't risk altering or damaging the underlying data.