An introductory tutorial of labeled property graphs for psychological data

Juan C. Correa[1,2*]

[1*]Departmento de Estudios Empresariales, Universidad Iberoamericana, Mexico City, Mexico, 1219.
[2*]Research and Development Unit, Critical Centrality Institute.

Corresponding author(s). E-mail(s): j.correa.n@gmail.com;

Abstract

Labeled property graphs offer a flexible framework for representing complex psychological data structures, such as multilevel designs, longitudinal assessments, and dynamic networks. This tutorial introduces graph databases and their labeled property graphs as an alternative to relational databases, illustrating how they enable efficient modeling of interconnected psychological phenomena. We provide a step-by-step guide for implementing property graphs using Neo4j, including schema design, query patterns, and integration with R/Python for analysis. Practical examples drawn from open psychological datasets demonstrate performance and flexibility gains over traditional relational approaches. Reproducible materials, including code and data, are available via OSF to facilitate adoption and replication.

Keywords: Graph database, Network modeling, complex behavior

1 Introduction

In a recent article, Soto (2025) introduced relational databases and illustrated their use by behavioral scientists through real-world examples. Although relational databases with their standardized "structured query language" (SQL) remain the dominant paradigm in research and industry, alternative approaches are gaining traction. The so-called "Not Only SQL" (NoSQL) category includes database systems that employ non-tabular data models, such as key-value pairs, documents, wide columns, matrices, and graphs. Whereas relational databases store structured data in tables (i.e.,

columns as variables, rows as cases, and cells containing specific values), NoSQL systems accommodate flexible formats that better support evolving and highly connected data.

In this article, I examine graphs as a database paradigm that deviates from the traditional lenses of relational databases, where nodes and edges (instead of tables and joins) represent the basic elements of any behavior that can be represented as a network or a complex system. Networks have a long history in mathematics as "*graph theory*" (Estrada, 2011). In sociology and social sciences, graph theory is known as "social network analysis" (Wasserman & Faust, 1994). In this context, the term "social network" should not be confused with online platforms such as Instagram or Facebook, as they are technological implementations that do not necessarily represent all aspects of social networks as an object of study. Psychologists have leveraged this framework to analyze the structure of psychopathology (Borsboom & Cramer, 2013), conduct bibliometric analysis of cyberbehavior (Serafin, Garcia-Vargas, García-Chivita, Caicedo, & Correra, 2019), estimate the correct number of dimensions in psychological and educational instruments (Golino & Epskamp, 2017), or understand the measurement of organizational climate (Menezes, Menezes, Moraes, & Pires, 2021). Even though these examples show the versatility of social network analysis for psychological research, they do not necessarily rely on graph database management systems and this aspect is what differentiates this article. Here, I intend to illustrate how the network lenses, when applied as a database management system, unveils novel ways for modeling and analyzing behavioral data.

2 Network as a collection nodes and edges

Estrada (2011) defines a network as a collection of points (called nodes) joined together in pairs by lines (called arcs or edges) like those depicted in Figure 1. Despite this simplistic definition, networks provide a powerful framework to model any type of system from planets in a galaxy to neurons in the nervous system (Vazza & Feletti, 2020).
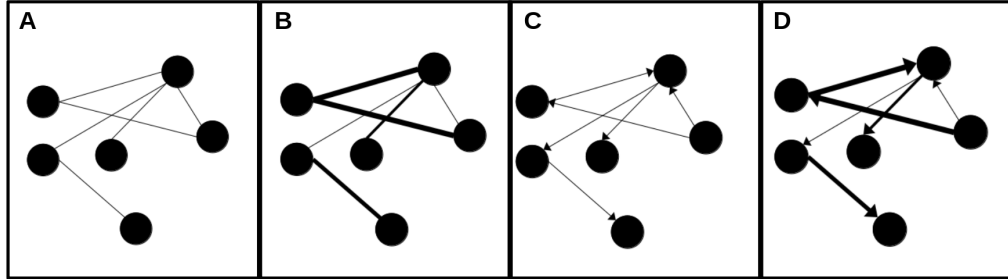

Fig. 1 A visual representation of four types of networks: A) non-directed unweighted network, B) non-directed weighted network, C) directed unweighted network, and D) directed weighted network.

In behavioral sciences, networks have been used to understand the mechanisms of team assembly and how these mechanisms determine collaboration structure and

team performance (Guimerà, Uzzi, Spiro, & Amaral, 2005). Graphs offer fundamental concepts for understanding how entities (nodes) and their relationships (edges) form interconnected structures. From a data management viewpoint, the analysis of these networks requires tools that go beyond the rigid tabular constraints of relational databases. As the concepts of graphs are thoroughly covered in introductory texts (Newman, 2010), these will not be revisited here. Instead, this article aims to illustrate how graph databases can enrich the methodological toolbox of psychological researchers and behavioral scientists, enabling analyses that embrace complexity, dynamic relationships, and multi-level contingencies (Robinson, Webber, & Eifrem, 2015).

3 Graph databases: A gentle introduction

Robinson et al. (2015) define a graph database as a system that implements **C**reate, **R**ead, **U**pdate, and **D**elete (CRUD) operations on a graph data model, where entities are represented as nodes and relationships as edges like the one depicted in Figure 2.
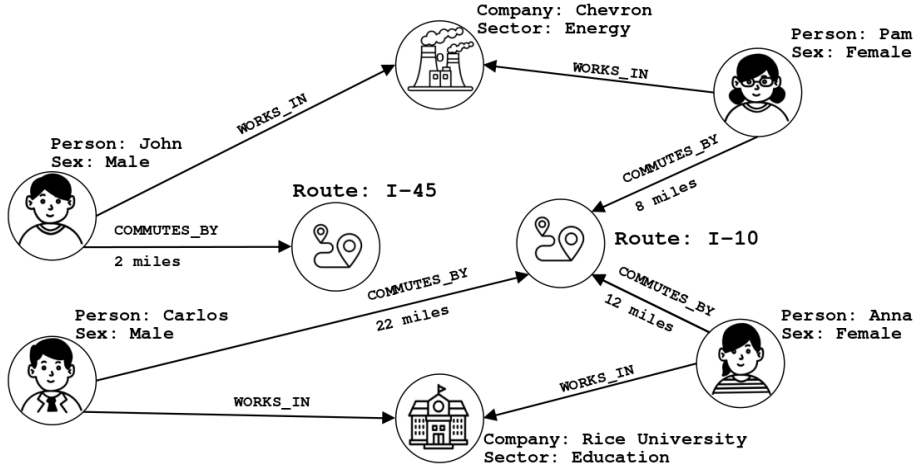

Fig. 2 A visual representation of a graph database that combines persons, companies, and routes

Unlike relational databases, which organize data in tables, graph databases treat relationships as first-class elements rather than secondary links between tables. This design enables efficient traversal and pattern matching across highly connected data, making it ideal for modeling complex networks such as behavioral contingencies or social interactions. Nodes in Figure 2 refers to real-world entities such as persons (i.e., Anna, Pam, Carlos, and John) companies (i.e., Chevron and Rice University), and routes (i.e., I-10 and I-45). Edges represent the relationship between pairs of nodes. Thus, John and Pam work in Chevron but they commute distinct distances through different routes. Carlos and Anna work in Rice University, they both commute by the same route but they have to drive different distances. Interestingly, the information

of nodes and edges can be enriched with attributes. These attributes also represent real-world characteristics like the distance each person has to commute, their sex, or the sector of the company they work for. This graph can become even more complex by adding more properties on nodes. For example, the series of edges labeled as `"COMMUTES_BY"` that already have the property of distance in miles can be enriched with other properties representing the specific time-frame and the current travel time each person experiences when using the road, allowing the integration of contextual contingencies into individual commuting behavior. As nodes and edges can represent distinct behaviors, contingencies, and interactions, graph database management systems represent a relevant methodological tool for behavioral sciences. I will elaborate upon these implications immediately.

4 Methodological implications of graph databases

A graph database model like the one depicted in Figure 2 leverages the so-called "labeled property graph." As a methodological tool it paves the way for incorporating complex systems thinking in any discipline (Krakauer, 2019). In psychology and behavioral sciences, complex systems thinking is not novel at all (Guastello, Koopmans, & Pincus, 2009; Luce, 1999). However, a labeled property graph and its elements have been largely overlooked. This gap coincides with recent reviews that promote other methodologies as potential solutions to increase intra and interdisciplinary interactions across subfields of behavior analysis in both basic and applied settings (Elcoro, Diller, & Correa, 2023).

An important methodolgical implication embedded in graph database models is their performance when handling connected data compared to relational databases that rely on tabular structures (Robinson et al., 2015). In relational databases, join-intensive queries slow down as datasets grow, whereas graph databases maintain relatively stable performance —even with millions of nodes and edges. From a traditional perspective, the idea of working with huge datasets might seem implausible for most psychological researchers. Nonetheless, big data research is no longer outside professional considerations (Vezzoli & Zogmaister, 2023). Such considerations, however, have been tackled by network scientists in charge of analyzing urban traffic at an individual level by tracking the movements of hundreds of thousands of drivers every two hours (Gonzalez, Hidalgo, & Barabasi, 2008). In cases like this, graph databases excel because queries operate on localized portions of the graph rather than scanning the entire dataset. If edges store information such as commuting distances, queries can be designed to retrieve specific conditions (e.g., individuals who commute less than 10 miles versus persons who commute more than 20 miles). Consequently, execution time depends only on the size of the subgraph traversed, not the overall graph size.

Another advantage of graph databases has to do with their flexibility. Behavioral scientists aim to connect data in ways that reflect their knowledge and expertise. This is why graph databases serve as the underlying infrastructure for constructing "knowledge graphs" (Barrasa & Webber, 2023). With these knowledge graphs, researchers allow the database to evolve alongside their understanding of the behavioral phenomenon rather than being rigidly defined upfront, when knowledge is most

limited. This is particularly important when a behavioral phenomenon lacks theoretical background or lacks replication (Burgos, 2025). Graph databases fulfill this need by providing a flexible model that adapts to changing requirements and evidence. Because graphs are inherently additive, new nodes, relationships, labels, and subgraphs can be introduced. The modifications introduced to the original graph do not imply a threat to existing queries or application functionality. This flexibility minimizes the need for exhaustive upfront modeling and lowers the frequency of costly migrations (particularly for companies that hire behavior data scientists in charge of analyzing customers observed behavior), thereby reducing maintenance overhead in both academic and business settings.

A third benefit of graph databases is the agility that they offer. Modern graph databases enable smooth development and easy system maintenance. Their schema-free design, combined with testable APIs and query languages, allows controlled evolution of applications. While the absence of rigid schemas means traditional governance mechanisms are missing, this is not a drawback. On the contrary, it encourages more transparent and actionable data governance. Typically, governance is enforced programmatically through tests that validate data models, queries, and business rules. This approach aligns well with agile and test-driven development practices, making graph database applications adaptable to changing business needs.

It is worth mentioning that relational databases acknowledge relationships, but only during modeling, where they serve as join mechanisms between tables. In graph databases, we often need to clarify the meaning of relationships and even qualify their strength. These are aspects that relational database management systems do not address explicitly (Robinson et al., 2015). From this viewpoint, graph databases demand ontological considerations such as those recently described for psychology and behavioral sciences (Burgos, 2025). As datasets grow more complex and less uniform, relational data management systems like Microsoft Access, PostgreSQL, or SQLite become burdened with large join tables, sparsely populated rows, and extensive null-checking logic. Greater interconnectedness in relational databases leads to more joins, which degrade performance and complicate adaptation to evolving requirements. Soto (2025) has highlighted some of these limitations as "challenges to adoption." From the lenses of graph databases, however, these challenges are not even necessary. I will elaborate upon this particular aspect below.

5 Applying graph database in behavioral research

To illustrate the application of graph databases for behavioral scientists, I revisit data from customers who used an online food delivery platform in the city of Bogotá. The database is available in a public data repository (Segura & Correa, 2019) which refers to an associated research article (Correa et al., 2019). Behavioral scientists who are familiar with structured datasets will find that this database is in a coma-separated values (.csv) file, just like most tables used in relational database management systems like Microsoft Access or SQLite. Given the wide variety of graph database management systems in the market (e.g., Neo4j, Microsoft Azure Cosmos, Aerospike, Amazon Web Services Neptune, NebulaGraph, Memgraph, TigerGraph, Giraph), the rest of this

work relies on Neo4j (Barrasa & Webber, 2023). Neo4j ranks as the leading freemium software in the segment of graph database management systems and has been used in several industry sectors including retail, finance, pharmaceutics, hospitality, and electronic commerce, among others. Neo4j offers free and enterprise editions, which can be installed on local or server environments following official documentation. As the official documentation provides helpful material for newcomers, downloading and installation details are not necessary here, and the reader can consult specific details elsewhere (Van Bruggen, 2014).

5.1 Instance creation and data import

Instance creation is the first step in using graph databases with Neo4j. This is a top-level operation for setting up a new database environment, and involves allocating resources (memory, CPU, storage), defining a database version, and setting up initial user credentials by clicking one button. When creating a new instance, neo4j asks the user to provide a name as part of the instance details. The user is asked to provide a password of eight characters as a security check for further interactions. The instance created is called "OFD," an acronym for "Online Food Delivery," and computational details can be found in our public GitHub repository. By default any instance created in neo4j is "stopped" mode and the user should start it by clicking one button. The data import is straightforward using the import option depicted with a red circle in the left panel of neo4j (see Figure 4).
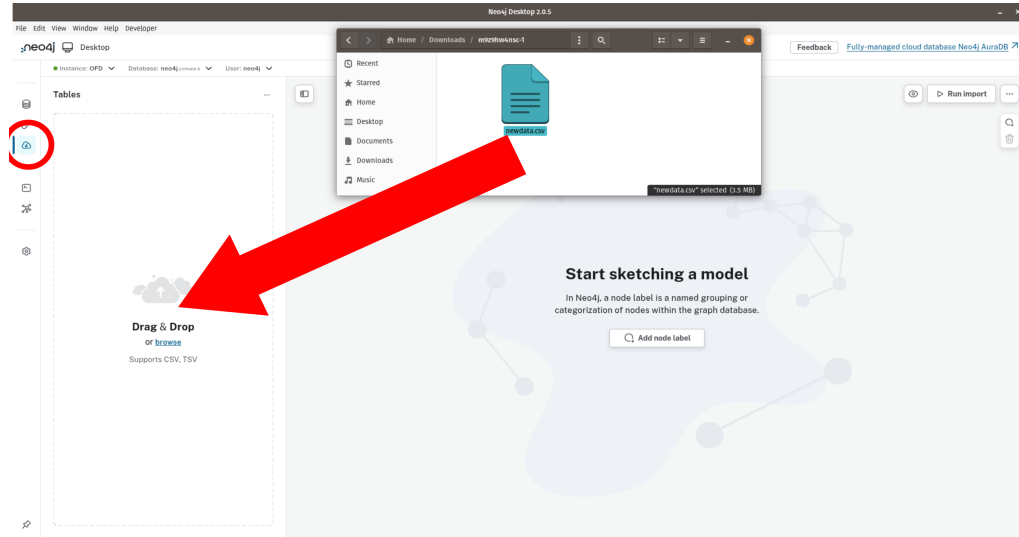

Fig. 3 The drag-and-drop import files option in the left panel of neo4j

As described above, the database is a csv file (newdata.csv) with a total of 19 variables. According to Segura and Correa (2019), this database was developed with the goal of evaluating the impact of traffic conditions on key performance indicators

of a sample of restaurants with food delivery services in Bogotá City. It includes the physical location of both restaurants and customers, as well as key performance indicators and traffic descriptions captured by the Google Maps API at three time points during Saturday rush hours.

5.2 Understanding the context

Traffic conditions in cities like Bogotá promote food delivery services as an alternative to cooking or driving to get food (Correa et al., 2019). As Bogotá has long suffered traffic congestion problems, local governmental authorities have decided to impose a restriction program called "Pico y Placa" (Montero, Sepúlveda, & Basso, 2025). This program was introduced in August 1998 and has been modified several times looking to extend its scope. For example, since July 2012, Pico y Placa affects the half of residential and commercial vehicles every other day of the week (excluding weekends) from 6:00 to 8:30 a.m. and then from 3:00 to 7:30 p.m, and buses, police cars, ambulances, fire trucks, government and diplomatic vehicles, school buses and vans, and electric and hybrid vehicles are exempt. To decide which half of the fleet is restricted in any given day, the Pico y Placa follows an odd–even schedule based on the last digit of the vehicle's license plate (Montero et al., 2025).

In this context, Correa et al. (2019) reported that the data collected from Google Maps API indicated that rush hours occur three times during Saturdays: in the morning (between 8:00 and 10:00 a.m); around midday (between 12:00 and 2:00 p.m); and in the evenings (between 6:00 and 8:00 p.m). Based on this information, Correa et al. (2019) classified the typical traffic in three categories: "free" or "green traffic" (G), "average" or "orange traffic" (O), and "heavy" or "red traffic" (R). By using letter triads they characterized the typical daily traffic with sequences like "G-G-G" or "R-O-R." Thus, for example, the sequence "R-O-G" means that the typical traffic changes from "red" in the morning to "orange" at noon and "green" in the afternoon, describing a place where traffic conditions improve as time passes.

5.3 Sketching a graph database model

Sketching a graph database model is the next step after a successful data import in Neo4j. Given the benefits of the labeled property graph, an initial model can be modified multiple times. These modifications are important because they reflect the analyst's expertise and highlight critical nodes and relationships to be mapped. An initial graph database model can consider the most elementary schema focused on the relationship between restaurants and customers (Figure 4).

Fig. 4 A visual representation of the initial schema about restaurants-customers relationship

In this graph model, the node "`Restaurant`" has four hidden properties (i.e., web, name, latitude, and longitude), the node "`Customer`" has two hidden properties (i.e., ClientLatitude, ClientLongitude), and the relationship "`SERVES`" has one property (i.e., distance). Despite its simple visual representation, with this model we can examine several statistical aspects. For example, by using cypher, which is Neo4j's declarative, SQL-like query language for property graphs, the following syntax provides the minimum, the mean, and the maximum distance coverage from restaurants to customers' location.

```
MATCH (:Restaurant)-[s:SERVES]->(:Customer)
RETURN round(avg(toFloat(s.Distance)), 1) AS avgDistanceMts,
       min(toFloat(s.Distance))           AS minDistanceMts,
       max(toFloat(s.Distance))           AS maxDistanceMts;
```

References

Barrasa, J., & Webber, J. (2023). *Building knowledge graphs: A practitioner's guide.* Sebastopol, CA: O'Reilly.

Borsboom, D., & Cramer, A. (2013, 03). Network analysis: An integrative approach to the structure of psychopathology. *Annual Review of Clinical Psychology*, *9*(Volume 9, 2013), 91–121, https://doi.org/10.1146/annurev-clinpsy-050212-185608

Burgos, J. (2025). Getting ontologically serious about the replication crisis in psychology. *Journal of Theoretical and Philosophical Psychology*, *45*(2), 79–100, https://doi.org/10.1037/teo0000281

Correa, J.C., Garzón, W., Brooker, P., Sakarkar, G., Carranza, S.A., Yunado, L., Rincón, A. (2019). Evaluation of collaborative consumption of food delivery services through web mining techniques. *Journal of Retailing and Consumer Services*, *46*, 45–50, https://doi.org/10.1016/j.jretconser.2018.05.002

Elcoro, M., Diller, J.W., Correa, J.C. (2023). Promoting reciprocal relations across subfields of behavior analysis via collaborations. *Perspective on Behavior Science*, *46*, 431-446, https://doi.org/10.1007/s40614-023-00386-x

Estrada, E. (2011). *The structure of complex networks: Theory and applications.* Oxford University Press.

Golino, H.F., & Epskamp, S. (2017). Exploratory graph analysis: A new approach for estimating the number of dimensions in psychological research. *PLOS ONE*, *12*(6), e0174035, https://doi.org/10.1371/journal.pone.0174035

Gonzalez, M.C., Hidalgo, C.A., Barabasi, A.-L. (2008). Understanding individual human mobility patterns. *Nature*, *453*(7196), 779–782, https://doi.org/10.1038/nature06958

Guastello, S.J., Koopmans, M., Pincus, D. (2009). *Chaos and complexity in psychology: The theory of nonlinear dynamical systems.* Cambridge University Press.

Guimerà, R., Uzzi, B., Spiro, J., Amaral, L. (2005, 04). Team assembly mechanisms determine collaboration network structure and team performance. *Science*, *308*(5722), 697–702, https://doi.org/10.1126/science.1106340

Krakauer, D. (2019). *Worlds hidden in plain sight: The evolving idea of complexity at the santa fe institute.* New Mexico: The Santa Fe Institute Press.

Luce, R.D. (1999). Where is mathematical modeling in psychology headed? *Theory & Psychology*, *9*(6), 723–737,

Menezes, I., Menezes, A.C., Moraes, E., Pires, P.P. (2021). Measuring organizational climate via psychological networks analysis. *International Journal of Organization Theory & Behavior*, *24*(3), 229–250, https://doi.org/10.1108/IJOTB-08-2020-0142

Montero, J.-P., Sepúlveda, F., Basso, L.J. (2025). Pricing congestion to increase traffic: The case of bogotá. *Journal of the European Economic Association*, jvaf039, https://doi.org/10.1093/jeea/jvaf039

Newman, M.E.J. (2010). *Networks: An introduction.* Oxford University Press.

Robinson, I., Webber, J., Eifrem, E. (2015). *Graph databases: New opportunities for connected data.* Sebastopol, California: O'Reilly.

Segura, M.A., & Correa, J.C. (2019). Data of collaborative consumption in online food delivery services. *Data in Brief*, *25*, 104007, https://doi.org/10.1016/j.dib.2019.104007

Serafin, M.J., Garcia-Vargas, G.R., García-Chivita, M., Caicedo, M.I., Correra, J.C. (2019). Cyberbehavior: A bibliometric analysis. *Annual Review of CyberTherapy and Telemedicine*, *17*, 17–24,

Soto, P.L. (2025, 12). Relational databases for behavior science. *Perspectives on Behavior Science*, , https://doi.org/10.1007/s40614-025-00486-w

Van Bruggen, R. (2014). *Learning neo4j.* Birmingham, UK: Packt Publishing.

Vazza, F., & Feletti, A. (2020). The quantitative comparison between the neuronal network and the cosmic web. *Frontiers in Physics*, *8*, 525731, https://doi.org/10.3389/fphy.2020.525731

Vezzoli, M., & Zogmaister, C. (2023). An introductory guide for conducting psychological research with big data. *Psychological Methods*, *28*(3), 580–599, https://doi.org/10.1037/met0000513

Wasserman, S., & Faust, K. (1994). *Social network analysis: Methods and applications.* Cambridge University Press.