1 Introduction

2 OLAP and Graph Databases

This chapter introduces the main concepts that forms the theoretical foundation necessary to better understand the work presented in this dissertation. Initially, we will explore the definition of Data Warehouse, Multidimensional Model and OLAP tools. Then, we will dive into concepts related to Graph and Graph Databases.

2.1 Data Warehouse

According to (INMON, 2005), a data warehouse (DW) is "a subject-oriented, integrated, nonvolatile, and time-variant collection of data in support of management‘s decisions". The first important aspect of a data warehouse is that it is subject-oriented, which means that the information stored in the DW is related to the company‘s subject. Consider a retail company for example: the main subjects can be product, sale, vendor and customer, therefore the data in the warehouse will be related to these entities.

The second characteristic of a data warehouse is that it is integrated, which means it contains data coming from multiple different sources. During the process of loading the warehouse, the data is converted, formatted, normalised and go through any other process to make the final information stored in the warehouse consistent. Another important aspect of a data warehouse is that it is nonvolatile, which means that the data in the warehouse does not get updated as operational environment. The warehouse is loaded in batches and it stores a snapshot, creating a history of the data. The final important aspect of a DW is that every record of data contains some sort of a time attribute to mark the moment in which the record is accurate.

Figure 2.1 shows the basic elements of a data warehouse (KIMBALL; ROSS, 2011). The Operational Source Systems capture and store the transactions of the business and they are considered elements outside of the data warehouse, since there is no control on the content or the format of the data that they store. The Data Staging Area is where the process of Extract-Transformation-Load (ETL) is made: the data is extracted from the operations source system, then they are transformed in order to integrate all the information in a consistent format, and finally the data is loaded into the presentation area.

The Data Presentation Area is where the data is organised, stored and accessed by analytical applications. According to (KIMBALL; ROSS, 2011), this area should be composed by a series of integrated data marts, which are repositories that present the data for a single process of the organisational business. A data mart stores atomic data and

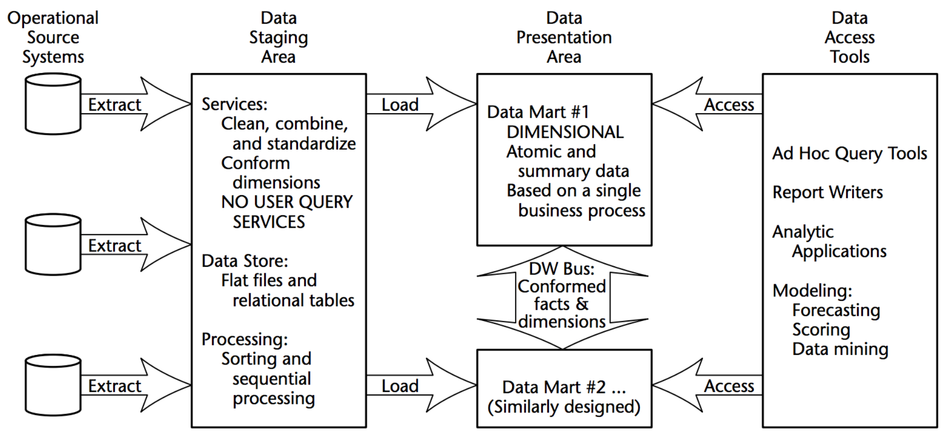

Operational Source Systems
Data Staging Area
Data Presentation Area
Data Access Tools
Extract
Extract
Extract
Services: Clean, combine, and standardize Conform dimensions NO USER QUERY SERVICES
Data Store: Flat files and relational tables
Processing: Sorting and sequential processing
Load
Load
Data Mart #1 DIMENSIONAL Atomic and summary data Based on a single business process
DW Bus: Conformed facts & dimensions
Data Mart #2 ... (Similarly designed)
Access
Access
Ad Hoc Query Tools
Report Writers
Analytic Applications
Modeling: Forecasting Scoring Data mining

Figure 2.1 – Elements of a Data Warehouse (KIMBALL; ROSS, 2011)

organizes them in a model that is more legible to humans. The most popular technology used to implement data marts adopted by the industry is On-Line Analytical Processing (OLAP), that will be covered in more detail along this chapter.

Finally, the Data Access Tools query the data in the presentation area. This element is formed by a set of different applications, since simple ad hoc queries until complex data mining application.

2.2 Multidimensional Model

The main functionality of a data warehouse is to facilitate multidimensional analysis (OLAP COUNCIL, 1997). This type of analysis reduces the number of misinterpretations by aligning the data with the analyst's mental model of the business. The multidimensional analysis provides an easy navigation through the database, showing specific subsets of data in different orientations and performing analytical calculations.

In order to perform multidimensional analysis, the data must be organised in a multidimensional model, where information is stored in a multidimensional array called *hypercube* or *cube* (VASSILIADIS, 1998). A cube is composed by cells that store *measures* aggregated by the data *dimensions*. A *dimension* is a cube's structural attribute formed by a list of properties that are similar to each other according to the user's perception of the data. Measures are the values being analysed by the user. Figure 2.2 shows an example of the dimensions and measures extracted from a sales application.

As shown in Figure 2.2 , each dimension (Geography, Time and Item) can be associated with an hierarchy of data aggregation levels. This feature allows user to view the data from different levels of details, for example: the measure *Sales* aggregated by *Region* can be detailed by *Country* or even further detailed by *City*.

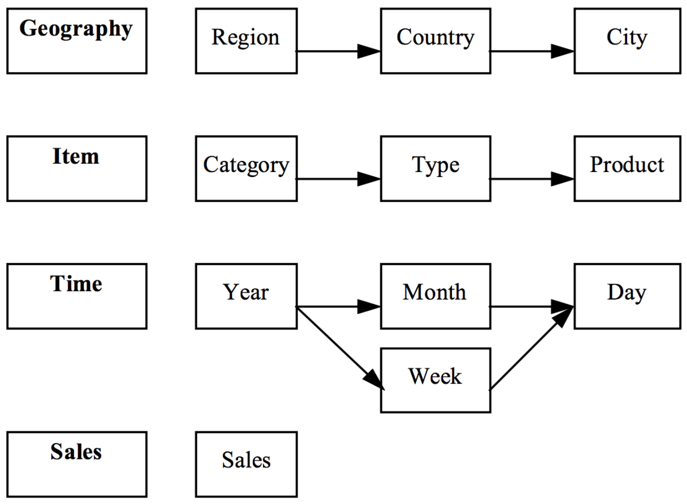


Figure 2.2 – Example of dimensions and measure for a sales application (VASSILIADIS, 1998)

2.3 OLAP

On-Line Analytical Processing (OLAP) is a category of technological tools used by analysts and managers to extract new knowledge from consolidated enterprise data (OLAP COUNCIL, 1997). In order to achieve this goal, data are organised in cubes and stored following multidimensional models. The access to this data should be fast, consistent and interactive and it should provide various views of information, reflecting the different dimensions of an enterprise as perceived by the user.

Usually, the data loaded into an OLAP system comes from a data warehouse. According to (INMON, 2005), the relationship between OLAP systems and data warehouse is complimentary: while OLAP offers control and flexible ways to explore data in different dimensions and hierarchy levels, data warehouse provides a robust data source for the OLAP system, where up-to-date data is available already extracted and properly integrated.

2.3.1 Types of OLAP Systems

There are two approaches for the physical model of an OLAP system (VASSILIADIS; SELLIS, 1999): Multidimensional On-Line Analytical Processing (MOLAP) and Relational On-Line Analytical Processing (ROLAP) Architectures.The MOLAP architecture provides a direct multidimensional view of the data. This approach stores the data in a Multidimensional Database Management System (MDBMS), which use n-dimensional arrays that contains the measures of the cube. This type of DBMS has a better performance than traditional Relational Databases, but it is more difficult to manage updates.

The ROLAP architecture is a multidimensional interface to relational data. This approach uses a traditional Relational Database Management System (RDBMS) to store the data organised in a star or snowflake schema. A star schema is formed by one or more dimension tables and one centralised fact table, which stores the measures of interest for the OLAP system. Figure 2.3 shows an example of a star schema, where the tables TIME, GEOGRAPHY, ACCOUNT and PRODUCT are dimension tables and SALES is the fact table.

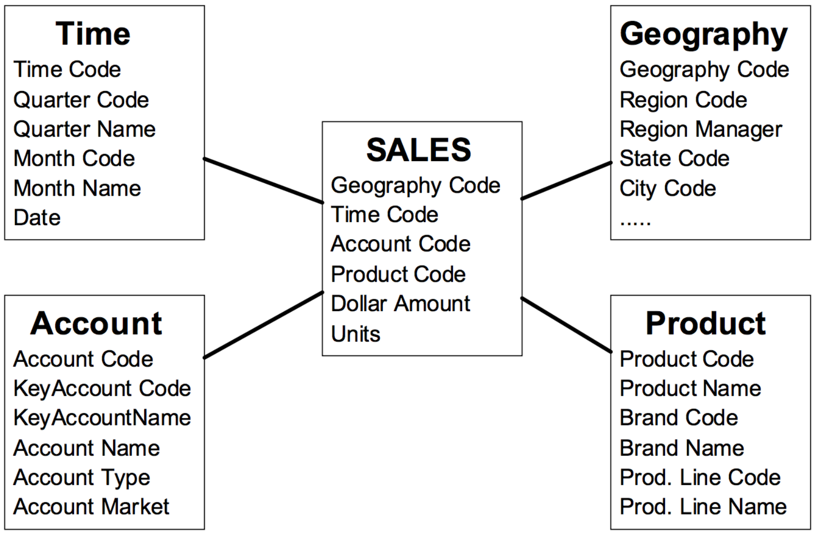


Figure 2.3 – Example of a Star Schema (VASSILIADIS; SELLIS, 1999)

Despite MOLAP architecture has the advantage of relying on a multidimensional native storage mechanism, the ROLAP approach allows easy integration with existing relational information systems and has the advantage of relational data being more efficiently stored than multidimensional data.

2.3.2 OLAP Operators

As mentioned earlier in this chapter, an OLAP system should provide a fast and interactive way for the user to explore the data stored in the cube. These are the main OLAP operations that can be used to navigate and manipulate different dimensions of the data (VASSILIADIS, 1998):

Roll-up Aggregates information along one dimension, summarising data on a higher level in its hierarchy. Consider the dimensions shown in Figure 2.2 and the measure of total dollar amount of sales per city, we can perform a roll-up operation to obtain the total dollar amount of sale per state.

Drill-down Detail information along one dimension, navigating the data from a higher to a lower hierarchy. Consider the measure of total dollar amount of sales per year, we can drill-down this query to obtain the total dollar amount of sales per month.

Slice Selects a slice of the cube according to user-specified dimension values. For instance, the user can select to view the total dollar amount of sales for the year of 2016.

Dice Selects a subcube from the original data according to user-specified conditions referred to more than one dimension. For instance, the user can select to view to total dollar amount of sales for the year of 2016 in the city of Recife.

Pivot Changes the orientation of dimensions in the cube, i.e. swap a row dimension to a column dimension.

There are other operations that can be performed in a OLAP system, but the ones shown above are considered the basic set of operations to support dynamic multidimensional analysis (INMON, 2005).

2.4 Graph

Graph is a data structure formed by a set of vertices (or nodes) $V = \{v_1, \ldots, v_n\}$ and a set of pair of vertices called edges (or relationships) $E = \{v_1v_2, \ldots, v_nv_m\}$. It can be represented graphically (where the vertices are shown as circles and edges are shown as lines, as shown in Figure 2.4) or mathematically in the form $G = (V, E)$ (TOBERGTE; CURTIS, 2013).

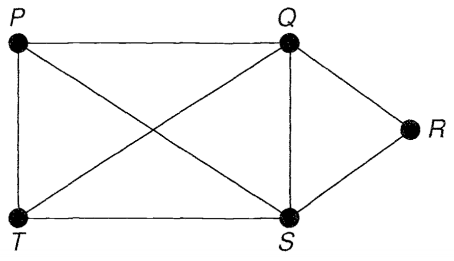


Figure 2.4 – Example of graph (TOBERGTE; CURTIS, 2013)

There are several real world applications that can take advantage of a graph model, specially the ones where the relationship between entities is an important information to be represented (MILLER, 2013). Consider, for example, a social network application similar to Twitter, where a user can follow another user. In this scenario, we can represent a user as a vertex and the relationship between users as an edge, as shown in Figure 2.5

2.4.1 Graph Theory

Graph Theory is a branch of Mathematics dedicated to the study of graph structures (TOBERGTE; CURTIS, 2013). Given the definition of a general graph explained above, there are several classifications and concepts associated with graph structures.

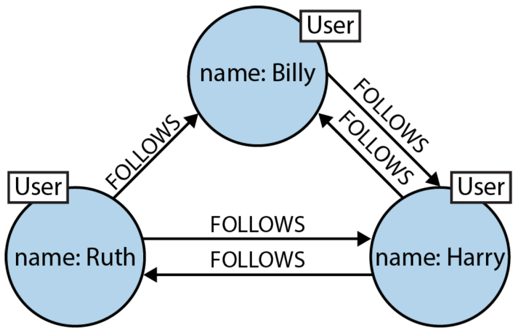


Figure 2.5 – Example of graph representation of a social network (MILLER, 2013)

Loop When an edge starts and finishes in the same vertex, as the one shown in Figure 2.6 starting at vertex T and finishing at vertex T.

Multigraph When a graph allows multiple edges connecting the same pair of vertices, as illustrated in Figure 2.6 with two edges connecting vertices Q and R.

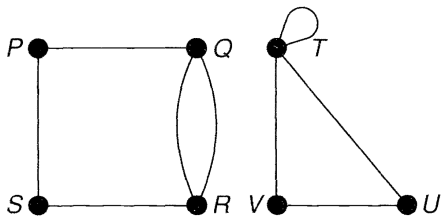


Figure 2.6 – Example of a multigraph with a loop (TOBERGTE; CURTIS, 2013)

Simple Graph When a graph does not have loops or multiple edges, as the one depicted in Figure 2.4.

Complete Graph When each pair of distinct vertices are connected to each other, as shown in Figure 2.7.

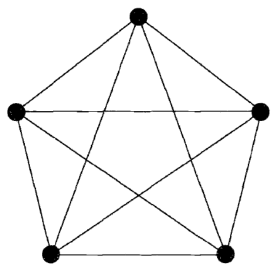


Figure 2.7 – Example of a complete graph (TOBERGTE; CURTIS, 2013)

Directed Graph When the edges have directions expliciting the start and end of the connection, as illustrated in Figure 2.8.

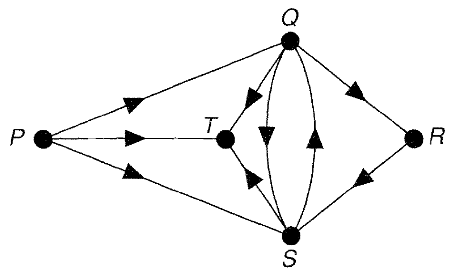
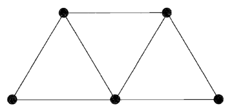


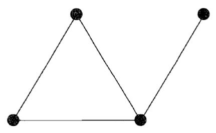
Figure 2.8 – Example of a directed graph (TOBERGTE; CURTIS, 2013)

Subgraph It is a graph obtained from another graph G, where its vertices are a subset of the vertices of G and its edges are a subset of the edges of G. Figure 2.9 shows in 2.9b a subgraph of the graph in 2.9a.

Path It is a subgraph obtained by a sequence of adjacent and distinct vertices.



(a) Original graph



(b) Subgraph

Figure 2.9 – Example of a subgraph (TOBERGTE; CURTIS, 2013)

There are other types of classifications of a graph and other concepts associated to it (TOBERGTE; CURTIS, 2013). These are the main definitions for the purposes of understanding the work presented here.

2.4.2 Network Analysis

There are different kind of analysis that can be done depending on the nature of the graph data. Due to the work presented here, we will explore the aspects of social network analysis, which focus on the structure of relationships (edges) between entities (vertices) in the graph. Several types of analysis can be performed in a graph, but the measure of a node‘s centrality is historically one of the most studied cases of analysis (FREEMAN, 1978). A common application for a centrality measure is to find out what are the focal points in a social network, i.e. who are the people that are the most connected to other people? There are three types of centrality measures (FREEMAN, 1978):

Degree Centrality This measure is given by the number of adjacent vertices. Considering a social network, a person (vertex) that has the greatest number of connections can be considered one of the focal points of the network, since it can directly read the most nodes in the graph. Given a graph with n vertices, the maximum value the degree centrality of a vertex in the graph can be is $n - 1$. The degree centrality of a vertex is related to its potential communication activity.

Betweenness Centrality This measure is given by the frequency in which a vertex is present in the shortest path between other vertices. Considering a social network, a person with high betweenness centrality is capable of influencing others by intercepting connections between several vertices in the graph. The betweenness centrality of a vertex is related to its potential control of communication.

Closeness Centrality This measure is given by the inverse of the average distance of the shortest path of a vertex between other vertices in the graph. Considering a social network, a person with high closeness centrality do not depend as much on others to communicate with other people in the network. The closeness centrality of a vertex is related to its potential independency and efficiency to control communication.

2.5 Graph Databases

Graph Databases are an alternative to Relational Database Management Systems (RDBMS), which are commonly used in the industry since the early 1980‘s. Despite the popularity of RDBMSs, GDBs allow the storage of data in graph model, which is a more natural way to represent information for some applications, such as social networks, semantic web pages and recommendation systems (MILLER, 2013).

The most popular form of graph model is the labeled property graph model (ROBINSON; WEBBER; EIFREM, 2015). The main characteristics of a labeled property graph are:

- Nodes (vertices) and Relationships (edges) can contain properties, i.e. key-value pairs
- Nodes can be labeled with one or more labels
- Relationships are named and directed, i.e. always have start and end nodes

An example of a labeled property graph was shown in Figure 2.5, where the nodes have the label “User” and the property “name” and the relationships are named “Follows” and have arrows indicating the direction of the edge.

2.5.1 Historical Overview

Scientific research related to graph data models were continuously published between 1980‘s and the first half of 1990‘s. Then, the database community attention turned to semistructured data model, due to the emergence of XML and the growth of hypertext document applications (ANGLES; GUTIERREZ, 2008). From this period, the main focus of the graph databases proposed was to establish a better way to represent data and methods to retrieve and manipulate data modeled as graph.

Recently this area has gained again attention from the community, due to the emergence of trendy projects (chemistry, biology, social network, semantic web, among others) where the importance of information relies not only in the entities but also in the relationship between them (ANGLES, 2012). Most recent implementations of graph databases are concerned with handling increasing amount of data and improving performance in retrieving and manipulating data.

The most popular graph database according to DBEngines[1] website is Neo4J[2]. It is an open source native graph storage database implemented in Java. Neo4J has its own query language called Cypher, which can be used to create, update and retrieve nodes and relationships. This graph database also provides high availability and scalability for big volume of graph data.

Besides Neo4J, there are other popular graph databases according to DB Engine site. OrientDB[3] is a NoSQL Multi-Model database, which means it stores data in the form of documents, key-value stores, objects, graph and others. Like Neo4J, OrientDB is also implemented in Java, but it uses an extended version of SQL that includes special operators to manipulate graph. This database allows the creation of a pre-defined data schema and the definition of complex data type, like dates, decimals and binary objects (BLOB).

TitanDB[4] is a native graph database implemented in Java. In order to establish connection with the hard disk, Titan needs to be linked to a data storage system - Cassandra[5], HBase[6] or BerkeleyDB[7] - that is suitable for the application. The creation of nodes, edges and the submission of queries can be done through a Java API or through a Gremlin server.

2.5.2 Comparing Graph Databases Performance

One of the basic tasks when manipulating a database is to retrieve an object given its id. The experiment described in (JOUILI; VANSTEENBERGHE, 2013) records the average time to retrieve a node by its id, given an 500000 nodes graph and 4 clients

[1] https://db-engines.com/en/system/Neo4j
[2] https://neo4j.com/
[3] http://orientdb.com/orientdb/
[4] http://titan.thinkaurelius.com/
[5] http://cassandra.apache.org/
[6] https://hbase.apache.org/
[7] http://www.oracle.com/technetwork/database/database-technologies/berkeleydb/overview/index.html

ht]

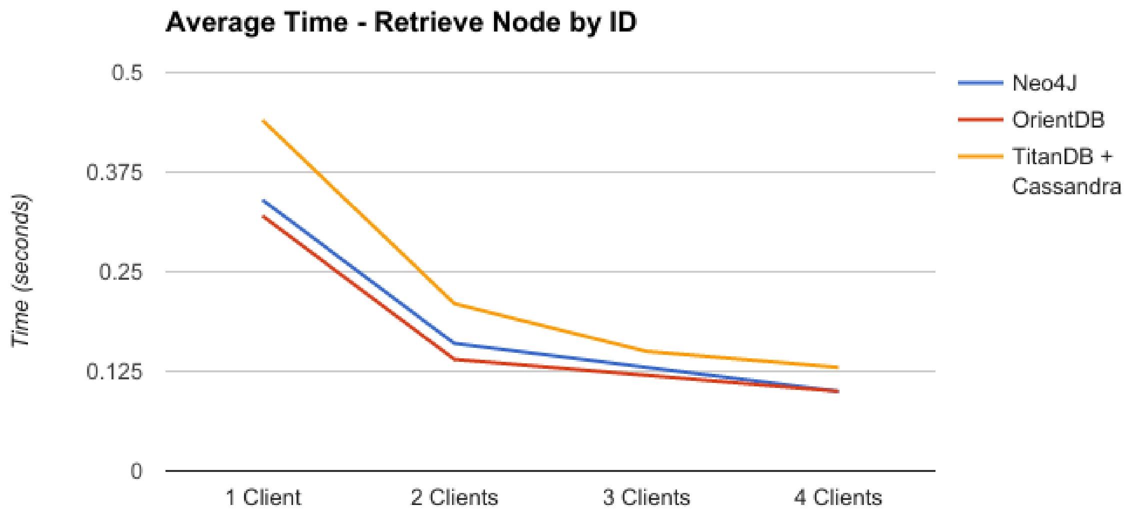


Figure 2.10 – Chart with mean time that each DB took to retrieve a node by its id

executing this action 200 times. This process was executed using Neo4J, OrientDB and TitanDB using Cassandra as backend.

According to Figure 2.10, Neo4J and OrientDB have similar performance, being OrientDB slightly faster to retrieve a node given its id. On the other hand, TitanDB using Cassandra as backend has a bad performance compared with the other DBs. However, TitanDB can be connected with other backend services, which can improve its performance for this task.

Another common concern among developers is the amount of memory taken by the database. In the experiment shown in (MCCOLL et al., 2014), a graph containing 32,000 nodes and 256,000 edges was stored in Neo4J, OrientDB and TitanDB and the amount of memory required by each database was measured.

From the Figure 2.11, we can notice that OrientDB is the one that consumes the most, reaching up to 10.156 MB of memory. On the other hand, TitanDB requires less memory, only needing 388 MB to store the graph.

2.6 Final Considerations

In this chapter, we discussed the main concepts related to the work presented in this dissertation. Initially, the definition of Data Warehouse and Multidimensional Model were introduced, followed by a detailed view on different types and operators an OLAP system can have. Then, we covered the main components of a graph and how it can be

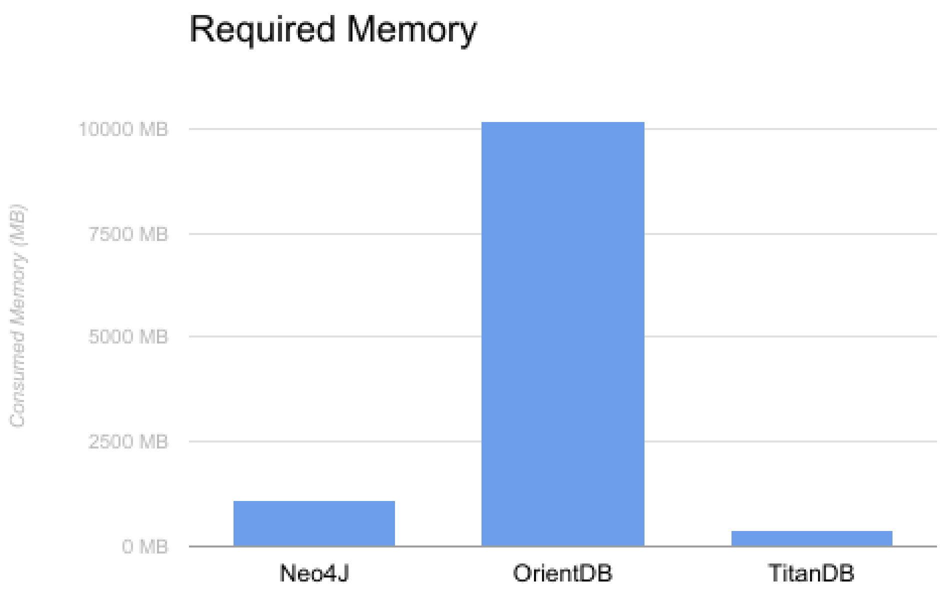


Figure 2.11 – Chart with the amount of memory required for each DB

classified according to different characteristics. Finally, we explained the definition of a Graph Database and went through a historical overview of such systems, as well as a comparison of the most popular Graph DBs nowadays. In the next chapter, we will review the most important research published that implements a OLAP system using Graph Databases.

3 Graph Cubes: State of the Art

This chapter presents the most important research works published in the area of OLAP systems implemented using graph databases. The first work presented here dates back to 2011 and introduces several concepts - such as graph cube, aggregated graphs and cuboid queries - that serve as basis for other papers published since then.

3.1 Graph Cube: On Warehousing and OLAP Multidimensional Networks

One of the main works in graph analysis using OLAP methods is described in (ZHAO et al., 2011), and it introduces several concepts that will be used by other authors to describe their work in this area. Graph Cube is one of those concepts and it is defined as a multidimensional model that extends multidimensional networks to provide decision support features. A multidimensional network is a graph $N = (V, E, A)$, where V is a set of vertices, E is a set of edges and A is a set of vertex-specific attributes. Each vertex in the graph is a multidimensional tuple and the attributes of the vertex define the graph cube dimensions.

A graph cube is formed by all possible aggregate networks calculated from the original multidimensional network. An aggregate network (often called cuboid) is a summarisation of the original graph with respect to one or more dimensions, which is calculated by applying an aggregate function (e.g. COUNT, SUM, AVERAGE, among others) on the vertices attributes. Consider the multidimensional network illustrated in Figure 2.11.

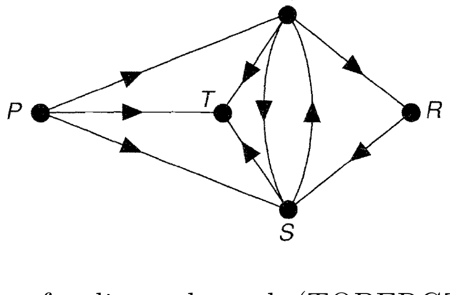


Figure 3.1 – Example of a directed graph (TOBERGTE; CURTIS, 2013)

Bibliography

ANGLES, R. A comparison of current graph database models. *Proceedings - 2012 IEEE 28th International Conference on Data Engineering Workshops, ICDEW 2012*, n. February, p. 171–177, 2012. Cited in page 11.

ANGLES, R.; GUTIERREZ, C. Survey of graph database models. *ACM Computing Surveys*, v. 40, n. 1, p. 1–39, 2008. ISSN 03600300. Disponível em: <http://portal.acm.org/citation.cfm?doid=1322432.1322433>. Cited in page 11.

FREEMAN, L. C. Centrality in social networks conceptual clarification. *Social Networks*, v. 1, n. 3, p. 215–239, 1978. ISSN 03788733. Cited in page 9.

INMON, W. *Building the data warehouse.* [S.l.: s.n.], 2005. 428 p. ISBN 0471081302. Cited 3 times in pages 3, 5, and 7.

JOUILI, S.; VANSTEENBERGHE, V. An empirical comparison of graph databases. In: IEEE. *Social Computing (SocialCom), 2013 International Conference on.* [S.l.], 2013. p. 708–715. Cited in page 11.

KIMBALL, R.; ROSS, M. *The Data Warehouse Toolkit: the complete guide to dimensional modeling.* [S.l.]: John Wiley & Sons, 2011. Cited 2 times in pages 3 and 4.

MCCOLL, R. C. et al. A performance evaluation of open source graph databases. In: ACM. *Proceedings of the first workshop on Parallel programming for analytics applications.* [S.l.], 2014. p. 11–18. Cited in page 12.

MILLER, J. Graph Database Applications and Concepts with Neo4J. *Proceedings of the Southern Association for Information Systems Conference*, p. 141–147, 2013. Disponível em: <http://sais.aisnet.org/2013/MillerJ.pdf>. Cited 3 times in pages 7, 8, and 10.

OLAP COUNCIL. *OLAP AND OLAP Server Definitions.* [S.l.], 1997. Disponível em: <http://www.olapcouncil.org/research/glossaryly.htm>. Cited 2 times in pages 4 and 5.

ROBINSON, I.; WEBBER, J.; EIFREM, E. *Graph Databases.* 2nd editio. ed. [S.l.]: O'Reilly Media, Inc., 2015. 238 p. ISBN 9781491930892. Cited in page 10.

TOBERGTE, D. R.; CURTIS, S. *Introduction to GraphTheory.* [S.l.: s.n.], 2013. v. 53. 1689–1699 p. ISSN 1098-6596. ISBN 9788578110796. Cited 4 times in pages 7, 8, 9, and 15.

VASSILIADIS, P. Modelling multidimensional database, cube and cube operations. *Scientific and Statistical Database Management, 1998. Proceedings. Tenth International Conference on*, n. May 2000, 1998. Cited 3 times in pages 4, 5, and 6.

VASSILIADIS, P.; SELLIS, T. A survey of logical models for OLAP databases. *ACM SIGMOD Record*, v. 28, n. 4, p. 64–69, 1999. ISSN 01635808. Disponível em: <http://portal.acm.org/citation.cfm?doid=344816.344869>. Cited 2 times in pages 5 and 6.

ZHAO, P. et al. Graph Cube: On Warehousing and OLAP Multidimensional Networks. *Sigmod '11*, p. 853–864, 2011. ISSN 07308078. Disponível em: <http://www.cs.uiuc.edu/{~}hanj/pdf/sigmod11{_}pzha>. Cited in page 15.