



Data Warehousing in the Cloud

PEDRO JOEL FERNANDES FERREIRA

Outubro de 2017

Data Warehousing in the Cloud

Pedro Joel Fernandes Ferreira

**Dissertação para obtenção do Grau de Mestre em
Engenharia Informática, Área de Especialização em
Sistemas Computacionais**

Orientador: Prof.^a Ana Maria Neves de Almeida Baptista Figueiredo

Co-orientador: Prof. Jorge Fernandes Rodrigues Bernardino

Júri:

Presidente:

[Nome do Presidente, Categoria, Escola]

Vogais:

[Nome do Vogal1, Categoria, Escola]

[Nome do Vogal2, Categoria, Escola] (até 4 vogais)

Porto, Outubro 2017

Dedicatória

Para os meus filhos, Dinis e Duarte.

Resumo

Um data warehouse, mais que um conceito, é um sistema concebido para armazenar a informação relacionada com as atividades de uma organização de forma consolidada e que sirva de ponto único para toda e qualquer relatório ou análise que possa ser efetuada. Este sistema possibilita a análise de grandes volumes de informação que tipicamente têm origem nos sistemas transacionais de uma organização (OLTP – Online Transaction Processing). Este conceito surgiu da necessidade de integrar dados corporativos espalhados pelos vários servidores aplicativos que uma organização possa ter, para que fosse possível tornar os dados acessíveis a todos os utilizadores que necessitam de consumir informação e tomar decisões com base nela.

Com o surgimento de cada vez mais dados, surgiu também a necessidade de os analisar. No entanto os sistemas de data warehouse atuais não têm a capacidade suficiente para o tratamento da quantidade enorme de dados que atualmente é produzida e que necessita de ser tratada e analisada.

Surge então o conceito de cloud computing. Cloud computing é um modelo que permite o acesso ubíquo e a pedido, através da Internet, a um conjunto de recursos de computação partilhados ou não (tais como redes, servidores ou armazenamento) que podem ser rapidamente provisionados ou libertados apenas com um simples pedido e sem intervenção humana para disponibilizar/libertar. Neste modelo, os recursos são praticamente ilimitados e em funcionamento conjunto debitam um poder de computação muito elevado que pode e deve ser utilizado para os mais variados fins.

Da conjugação de ambos estes conceitos, surge o cloud data warehouse que eleva a forma como os sistemas tradicionais de data warehouse são definidos ao permitir que as suas fontes possam estar localizada em qualquer lugar desde que acessível pela Internet, tirando também partido do grande poder computacional de uma infraestrutura na nuvem. Apesar das vantagens reconhecidas, há ainda alguns desafios sendo dois dos mais sonantes a segurança e a forma como os dados são transferidos para a nuvem.

Nesta dissertação foi feito um estudo comparativo entre variadas soluções de data warehouse na cloud com o objectivo de recomendar a melhor solução de entre as estudadas e alvo de testes. Foi feita uma avaliação com base em critérios da Gartner e num inquérito sobre o tema. Desta primeira avaliação surgiram as duas

soluções que foram alvo de uma comparação mais fina e sobre as quais foram feitos os testes cuja avaliação ditou a recomendação.

Palavras-chave: cloud computing, data warehouse

Abstract

A data warehouse, rather than a concept, is a system designed to store the information related to the activities of an organization in a consolidated way and that serves as a single point of truth for any report or analysis that can be carried out. It enables the analysis of large amounts of information that typically comes from the organization's transactional systems (OLTP). This concept arose from the need to integrate corporate data across multiple application servers that an organization might have, so that it would be possible to make data accessible to all users who need to consume information and make decisions based on it.

With the appearance of more and more data, there has also been a need to analyze it. However, today's data warehouse systems do not have the capacity to handle the huge amount of data that is currently produced and needs to be handled or analyzed.

Then comes the concept of cloud computing. Cloud computing is a model that enables ubiquitous and on-demand access to a set of shared or non-shared computing resources (such as networks, servers, or storage) that can be quickly provisioned or released only with a simple request and without human intervention to get it done. In this model, the features are almost unlimited and in working together they bring a very high computing power that can and should be used for the most varied purposes.

From the combination of both these concepts, emerges the cloud data warehouse. It elevates the way traditional data warehouse systems are defined by allowing their sources to be located anywhere as long as it is accessible through the Internet, also taking advantage of the great computational power of an infrastructure in the cloud. Despite the recognized advantages, there are still some challenges. Two of the most important are the security and the way data is transferred to the cloud.

In this dissertation a comparative study between several data warehouse solutions in the cloud was carried out with the aim of recommending the best solution among the studied solutions. An assessment was made based on Gartner criteria and a survey on the subject. From this first evaluation came the two solutions that were the target of a finer comparison and on which the tests whose assessment dictated the recommendation were made.

Keywords: cloud computing, data warehouse

Agradecimentos

Agradeço aos meus orientadores, Professora Doutora Ana Almeida e Professor Doutor Jorge Bernardino, o conhecimento, a partilha e disponibilidade.

Um agradecimento especial à Rita pelo apoio e ajuda no decorrer deste trabalho.

A todos, muito obrigado.

Index

1	Introduction	1
1.1	Problem description.....	1
1.2	Value Analysis	2
1.3	Objective	3
1.4	Achieved results	3
1.4.1	Paper Publication.....	3
1.5	Document structure	3
2	Cloud Data Warehousing	6
2.1	What is a data warehouse?.....	7
2.2	What is cloud computing?	9
2.2.1	Cloud environment actors.....	12
2.2.2	Cloud implementation models.....	12
2.3	What is a cloud data warehouse?	14
3	Cloud Data Warehousing market solutions	18
3.1	Amazon Redshift	19
3.2	Microsoft Azure SQL Data Warehouse	21
3.3	Snowflake	23
3.4	Pivotal GreenPlum.....	25
3.5	Solutions Comparison	27
4	Methodology	29
4.1	How to evaluate and compare.....	29
4.2	Hypotheses and evaluation methodologies	30
4.2.1	Architecture	30
4.2.2	Experimental setup	31
4.2.3	Test cases.....	35
5	Experimental Evaluation	37
5.1	Cloud solutions comparison	37
5.1.1	Redshift vs Azure SQL Data Warehouse System properties comparison.....	42
5.1.2	Redshift Configuration and Test	45
5.1.3	Azure SQL Data Warehouse Configuration and Test.....	49
5.1.4	Redshift and Azure SQL Data Warehouse Unit Test	52
5.1.5	Redshift and Azure SQL Data Warehouse side by side	54
5.1.6	Redshift and Azure SQL Data Warehouse load test.....	55
5.2	Recommendation	55

6	Conclusions and future work	57
---	-----------------------------------	----

Figures List

Figure 1 – Inmon data warehouse default architecture (source: (“1 Data Warehousing Concepts,” n.d.)).....	8
Figure 2 – Kimball data warehouse default architecture (source: (“Inmon vs. Kimball: Which approach is suitable for your data warehouse?,” n.d.)).....	9
Figure 3 – Cloud computing architecture (source: (Zhang et al., 2010)).....	11
Figure 5 – Microsoft Azure SQL Data Warehouse system architecture (source: (“SQL Data Warehouse Microsoft Azure,” n.d.))	22
Figure 6 – Snowflake system architecture (source: (“Key Concepts & Architecture – Snowflake Documentation,” n.d.))	23
Figure 7 – Pivotal Greenplum system architecture (source: (“About the Greenplum Architecture Pivotal Greenplum Database Docs,” n.d.))	26
Figure 8 – Test system configuration	31
Figure 10 – Source and destination database data model of sales subject area	33
Figure 11 – Public cloud adoption survey results.....	39
Figure 12 – Public cloud adoption survey results – 2017 vs 2016.....	39
Figure 13 – Enterprise Public cloud adoption survey results – 2017 vs 2016	40
Figure 14 – Enterprise Public cloud adoption survey results	41
Figure 15 – SMB Public cloud adoption survey results – 2017 vs 2016.....	41
Figure 16 – SMB Public cloud adoption survey results.....	42
Figure 17 – Cluster configuration resume	45
Figure 18 – Rule configuration	46
Figure 19 – Oracle SQL Developer connection configuration	46
Figure 20 – Azure SQL Data Warehouse overview	49
Figure 21 – Resources resume	49
Figure 22 – Azure Portal Query Editor Interface	50
Figure 23 – Azure Data Factory Activity Window	51
Figure 24 – Redshift loading table.....	53
Figure 25 – SQL Data Warehouse loading table	54
Figure 26 – Load Test results.....	55

Tables List

Table 1 — Consumer responsibilities comparison	11
Table 2 — Database comparison characteristics	27
Table 3 — Account dimension table	34
Table 4 — Survey response fact table	34
Table 5 — File size and number of records	34
Table 6 — FactResellerSales file with larger volume.....	35
Table 7 — Solutions evaluation by Gartner Criteria (Studied Cases)	37
Table 8 — Other solutions evaluation by Gartner Criteria.....	38
Table 9 — Redshift and Azure SQL system properties	43
Table 10 — Pros and Cons of Redshift and Azure SQL Data Warehouse	44
Table 11 — Upload elapsed time to S3 Storage	47
Table 12 — Elapsed time of Loading data into RedShift table	48
Table 13 — Upload elapsed time to Azure Blob Storage	50
Table 14 — Elapsed time of Loading data into Azure SQL DW table	52
Table 15 — Elapsed time of Unit Test	53
Table 16 — Redshift and Azure SQL Data Warehouse side by side	54
Table 17 — Currency dimension table	62
Table 18 — Customer dimension table	62
Table 19 — Date dimension table	62
Table 20 — Department dimension table	63
Table 21 — Employee dimension table.....	63
Table 22 — Geography dimension table	64
Table 23 — Organization dimension table	64
Table 24 — Product dimension table	64
Table 25 — Product category dimension table	65
Table 26 — Product subcategory dimension table	65
Table 27 — Promotion dimension table	65
Table 28 — Reseller dimension table.....	66
Table 29 — Sales reason dimension table.....	66
Table 30 — Sales territory dimension table	66
Table 31 — Scenario dimension table.....	66
Table 32 — Call center fact table	68
Table 33 — Currency rate fact table	68
Table 34 — Finance fact table	68
Table 35 — Internet sales fact table	68
Table 36 — Internet sales reason fact table.....	69
Table 37 — Reseller sales fact table.....	69
Table 38 — Sales quota fact table.....	70

Acronyms

AWS	Amazon Web Services
BI	Business Intelligence
BPM	Business Process Management
CPU	Central Processing Unit
DBAAS	Database as a Service
DHCP	Dynamic Host Configuration Protocol
DMS	Data Movement Service
DNS	Domain Name System
DW	Data Warehouse
DWU	Data Warehouse Units
ETL	Extract, Transform, Load
FTP	File Transfer Protocol
GB	Gigabyte
GPDB	Greenplum database system
HTTP	Hypertext Transfer Protocol
HTTPS	Hyper Text Transfer Protocol Secure
IAAS	Infrastructure as a Service
IMAP	Internet Message Access Protocol
IT	Information Technology
JDBC	Java Database Connectivity

MPP	Massively Parallel Processing
ODBC	Open Database Connectivity
OLTP	Online Transaction Processing
PAAS	Platform as a Service
PB	Petabyte
POP3	Post Office Protocol version 3
RDBMS	Relational Database Management System
SAAS	Software as a Service
SMB	Small and Midsize Business
SMTP	Simple Mail Transfer Protocol
SOA	Service Oriented Architecture
SQL	Structured Query Language
TB	Terabyte

1 Introduction

Data warehouses are defined as customized data storage that aggregate data from multiple sources and store it in a common location to be able to run reports and queries over it. Many companies use data warehouses to compile regular financial reports or business metric analyses.

The success of the implementation depends on the existence of a service-oriented strategy at the organization level, which would provide the necessary infrastructure for the Cloud implementation. Without SOA and BPM, integration of a Data Warehousing solution based on Cloud Computing serves no financial purpose, involving high costs for the present systems reengineering. Also, in order to be successful, Cloud strategy has to be led according to the business strategy of the organization.

1.1 Problem description

One of the biggest challenges with data warehousing in the Cloud is how the data is transferred up into the cloud. Pumping gigabytes, terabytes, or even petabytes of data up into Cloud over the public Internet can not only come with security concerns, but also performance challenges.

When selecting a Data Warehousing solution, we have to take into account the newest trends on the DW and Cloud Computing market, the present and future needs and the opportunity of integration. In order to be successful, the selection of a Cloud DW solution has to be achieved objectively based on good criteria that have been analysed and weighted according to the present and future needs of the organization.

Cloud Data Warehousing is a potential cost saver for big companies, and removing a cost barrier that have held data warehousing back from small and mid-sized businesses. Why a potential cost saver? Acquisition costs are lower because equipment is rented, companies only pay for what they use, maintenance and software/hardware upgrade is no longer customer responsibility, easy scale up or down without the need to allocate company human resources to do this job and so on.

A Cloud Data Warehouse must be designed to take away the undifferentiated heavy lifting of running infrastructure at heavy scale, allowing the customers to focus on their core competencies – its business.

The growing interest in cloud-based data warehousing is driven by the high return on investment. Nonetheless, the adoption of cloud computing for data warehousing faces security challenges given the proprietary nature of the enclosed data.

Moving to the cloud is a hard decision not only because the data owners like to have their data near to them (on-premises) but also due to security and data confidentiality issues, because of this last two the decision makers tend to delay the decision of moving to the cloud.

1.2 Value Analysis

“Value analysis is an examination of the function of parts and materials in an effort to reduce cost and/or improve product performance. The primary objective of value analysis is assess how to increase the value of an item or service at the lowest cost without sacrificing quality.” (Susana Nicola, 2016).

“The creation of value is key to any business, and any business activity is about exchanging some tangible and/or intangible good or service and having its value accepted and rewarded by customers or clients, either inside the enterprise or collaborative network or outside.” (Susana Nicola, Eduarda Pinto Ferreira, J. J. Pinto Ferreira, International Journal of Information Technology & Decision Making 2012).

The product “developed” in this project is an analysis and proposition of a cloud data warehousing solution to implement in an organization, whatever it is.

It is not a developed product, but a recommendation of an existing product in the market that accomplishes the goal of having a data warehouse in the cloud.

1.3 Objective

In this project we intend to analyze all the aspects of Cloud Data Warehousing, putting the stress on the integration of a Cloud DW solution within organizations. More importantly, the opportunity of using a Cloud Data Warehouse solution is analyzed, in contrast of using a traditional DW solution. An important point is, to evaluate the various players in the market with special focus on security and performance issues.

1.4 Achieved results

As result of this project, some cloud data warehousing solutions were analysed, evaluated and a recommendation is given of what is the best cloud data warehouse solution in the market based on the defined criteria for the comparison.

1.4.1 Paper Publication

In alignment with the investigation done in this thesis, a paper was submitted and accepted for publication at the KDIR. The published article titled, “Data Warehousing in the Cloud: Amazon Redshift vs Microsoft Azure SQL”, will be presented at the 9th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management, held in Funchal in November 2017.

1.5 Document structure

This document has six chapters. In chapter 2 the concepts related with this work are presented and described.

In chapter 3 are presented four market solutions related with cloud data warehousing.

In chapter 4 is presented the methodology used in the project.

In chapter 5 the solution evaluation is presented in detail with the criteria that supported the decision of what it the best cloud data warehouse solution.

Finally, chapter 6 concludes the document and presents future work.

Annex A presents the definition of the dimension tables of the data warehouse model used in the evaluation.

Annex B presents the definition of the fact tables of the data warehouse model used in the evaluation.

Annex C shows the database model and all its relations.

Annex D presents a value analyses along with a business model, Canvas.

Annex E presents the article accepted at 2017 KDIR 9th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management.

Annex F shows the month pricing for the identified cloud solutions that provide data warehousing services.

Annex G presents the script for creating the database tables used on tests.

Annex H has the Python script used for uploading files up into Amazon S3 Storage.

Annex I has the script used to load files into database tables in Amazon RedShift.

Annex J has the Python script used for uploading files up into Microsoft Blob Storage.

Annex K has the source code of the web service used is the load test.

Annex L has the SQL statements used in the tests.

2 Cloud Data Warehousing

In this chapter we give a briefly definition of cloud computing and data warehouse. A data warehouse is a collection of data from multiple data sources which are nonvolatile, integrated, time variant and subject-oriented (Inmon, 2002). “Cloud computing is a type of Internet-based computing that provides shared computer processing resources and data to computers and other devices on demand. It is a model for enabling ubiquitous, on-demand access to a shared pool of configurable computing resources (e.g., computer networks, servers, storage, applications and services), which can be rapidly provisioned and released with minimal management effort.” (Nirmala et al., 2016).

Nowadays, companies have a greater collection of data than ever before. This includes a huge variety of sources, including cloud-based applications or even company datamarts. In order to make good decisions, get insights and achieve a competitive advantage, companies need to have their data properly analyzed, on time.

The conventional data warehouse architecture is widespread in a large number of companies working with large and diverse data sets but is a very closed and complex model to response with the agility that companies currently need. People who make analysis need to wait a number of hours or even days for data to flow into the data warehouse before it becomes available for analysis. In most cases, the storage and compute resources required to process that data are insufficient (or the same) and this leads to hanging or crashing systems (Goutas et al., 2016).

To increase the level of efficiency and competitiveness, organizations must be able to process and analyze their data in the lowest period of time possible and now this is a reality (Dageville et al., 2016).

One of the big concerns on moving to the cloud is the time to “live” with both on-premises and cloud data warehouse system because it is not a good idea to move at once the whole data warehouse. To ease this concern, a data virtualization solution can be used to help out the migration and coexistence of the both data warehouse systems while migration to cloud is on going.

2.1 What is a data warehouse?

A data warehouse is a computer system designed with the purpose of storing corporate data to be analyzed and get insights to decision makers (Sheta and Eldeen, 2013). Typically, organizations use a data warehouse to integrate internal information from its different data sources and have a unique and complete vision of all organization corporate data.

On-premises data warehouses follow one of the following architecture models: Inmon (Inmon, 2002) or Kimball (Kimball and Ross, 2013).

While Bill Inmon defends a top-down approach to the data warehouse, Kimball defends a bottom-up approach. The Inmon approach is the design approach where the DW, is the centralized container for the entire company. It is built by defining a normalized data model. The top-down approach defines the whole business model for all the data to be used in the company at its first step, the tables, data, and the relationships among them (“Data Warehousing Implementations,” n.d.). Kimball defends that after the most important business indicators, data marts should be created first because they will provide a refined view of organizational data.

To Bill Inmon, a data warehouse system is a centralized repository for an entire enterprise and it stores the atomic data at its lowest level of detail and after the completion of the data warehouse provisioning the dimensional data marts can be created (Inmon, 2002). Figure 1 illustrates Inmon default data warehouse architecture.

A data warehouse, to Inmon, is defined in the following terms (Inmon, 2002):

- Subject-oriented;
- Time-variant;
- Non-volatile;
- Integrated.

It is subject-oriented because a data warehouse can be used to analyse a particular subject area. Time-variant because historical data is kept in the data warehouse. Non-volatile because once the data is in the data warehouse, it will not change. So, historical data in a data warehouse should never be altered. Integrated because a data warehouse integrates data from multiple data sources. For example, source A and source B may have different ways of identifying a customer, but in a data warehouse, there will be only a single way of identifying the same customer.

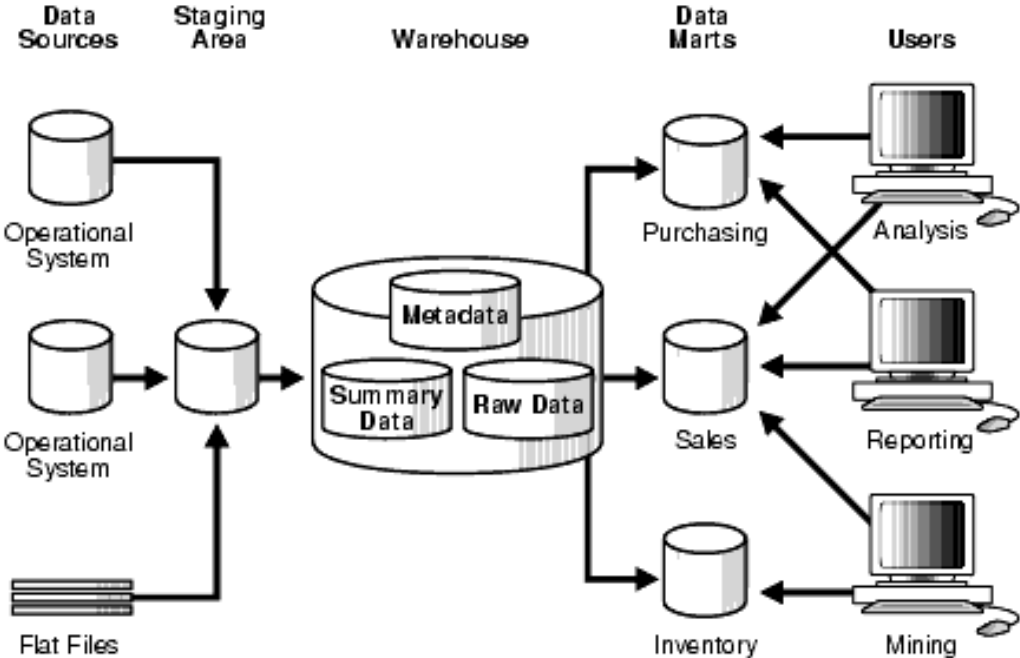


Figure 1 – Inmon data warehouse default architecture (source: (“1 Data Warehousing Concepts,” n.d.))

Figure 1 illustrates:

- Data sources (operational systems and flat files);
- Staging area, where data sources go before the data warehouse;
- Data Warehouse;
- Data marts;
- Users that consume the information of data marts.

The Ralph Kimball approach of building a data warehouse starts with the identification of the key business processes and questions that the data warehouse needs to answer. So, Kimball’s approach suggests the creation of multiple star-schemas that form the data warehouse but each one is subject-oriented – the

complete collection of the star-schemas form a dimensional model to be explored by the users, see Figure 2.

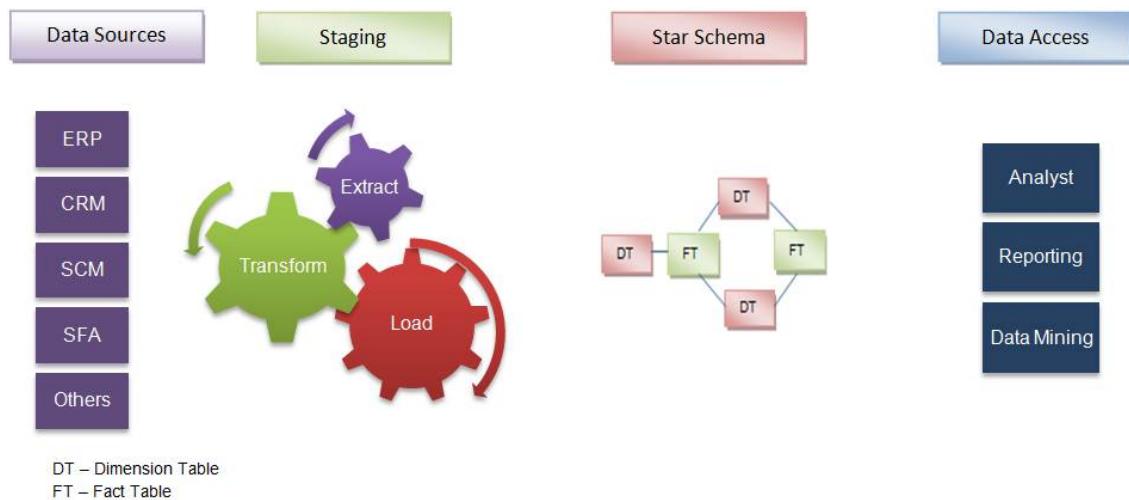


Figure 2 – Kimball data warehouse default architecture (source: (“Inmon vs. Kimball: Which approach is suitable for your data warehouse?,” n.d.))

These two approaches have given proves of its capabilities but now with the amount of data generated by organization or collected from Internet, this models don’t respond with the quickness or agility that nowadays is acceptable and the move to a cloud data warehouse solution may be the answer.

2.2 What is cloud computing?

There is no definitive definition of cloud computing, but Kimball and Ross (2013) propose that “Cloud computing is a self-provisioning and on-demand delivery of computing resources and applications over the Internet with a pay-as-you-go pricing model”.

Cloud, by definition, is a self-service system that allows end users to setup applications and services in a cloud computing environment without the intervention of an IT service provider (Armbrust et al., 2010).

Cloud computing addresses large amounts of computing power by aggregating computing resources and offering a single view of the system.

There are some common characteristics to the most of cloud computing environments (Mell and Grance, 2011):

- *On-demand usage*: the capacity of unilaterally provision computing capabilities as needed without requiring human interaction;
- *Broad network access*: resources available over the network, supporting heterogeneous client platforms;
- *Resource pooling*: multiple customers from the same physical resources, by securely separating the resources logically;
- *Rapid elasticity*: computing resources scale as required in response to runtime conditions or demanded by the consumer;
- *Metered usage*: resource usage is monitored, measured and reported transparently based on utilization.

A cloud computing solution addresses three main areas of operation, see Figure 3 (“Above the Clouds: A Berkeley View of Cloud Computing | EECS at UC Berkeley,” n.d.) (Mell and Grance, 2011):

- *Infrastructure as a service (IaaS)*: it is a layer where virtualization is present and has a pool of resources ready to be used. Storage, networking and computer nodes are present here and can be dynamically assigned on demand – this is only possible due to virtualization technologies. These resources are available through virtual machines provided by the provider where the consumer can install an operating system and any other applications. The consumer does not control the infrastructure that provides the service but has the power to manage all its resources;
- *Platform as a service (PaaS)*: it is built on top of infrastructure layer and is made by the operating system and other application frameworks such as Java or .Net. All maintenance of the platform and its native components are in the responsibility of the provider but the consumer has full control of the applications that he implements on the platform. The main idea behind this model is to enable consumers to streamline and simplify applications, freeing them from the complexity of installing and configuring infrastructure and programming environments;
- *Software as a service (SaaS)*: provides the users with a complete software application or the user interface to the application itself. The cloud service provider manages the underlying cloud infrastructure, including servers, network, operating systems, storage, and application software. The user is unaware of the underlying architecture of the cloud.

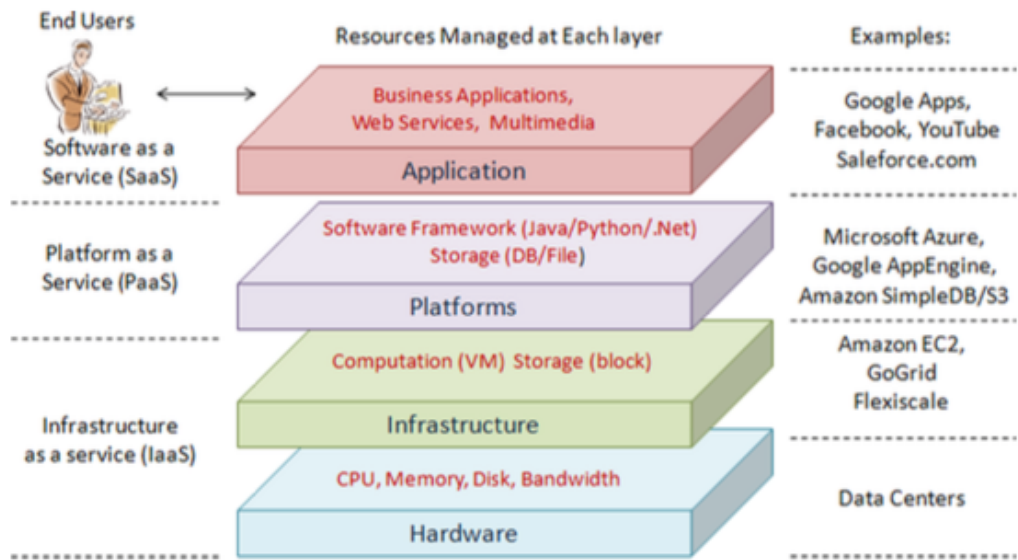


Figure 3 – Cloud computing architecture (source: (Zhang et al., 2010))

From the consumer point of view, Table 1 clearly illustrates its responsibilities comparing the traditional approach with the three cloud computing areas where the green blocks illustrate the areas managed and in responsibility of the cloud provider and the white ones illustrates the consumer responsibilities area.

Table 1 – Consumer responsibilities comparison

Traditional Approach	IaaS	PaaS	SaaS
Applications	Applications	Applications	Applications
Data	Data	Data	Data
Runtime	Runtime	Runtime	Runtime
Middleware	Middleware	Middleware	Middleware
Operating System	Operating System	Operating System	Operating System
Virtualization	Virtualization	Virtualization	Virtualization
Processing	Processing	Processing	Processing
Storage	Storage	Storage	Storage
Network	Network	Network	Network

There are also four types of cloud (Zhang et al., 2010) or four different models of implementation of the cloud services (detailed in section 3.2.2):

- *Private cloud*, for personal use of an organization. It is also known as internal cloud;
- *Public cloud*, where service providers offer their resources as services to the general public;
- *Hybrid cloud*, is a combination of private and public cloud services;
- *Community cloud*, for exclusive use by a specific community of consumers that have shared concerns.

2.2.1 Cloud environment actors

Cloud brought new features and possibilities to improve its users life but also originated a new market segment for IT with new business opportunities. Many organizations build their business around the cloud not only to use its services but also to offer business solutions in this environment. Like any other service, we can identify at least two actors that are directly connected with it:

- User or consumer, is the one who uses the functionalities or resources provided by the cloud. It is the entity or organization that uses the cloud computing services, whether they are related with software, platform or infrastructure;
- Provider, is the entity or organization responsible for making cloud services available to consumers. The provider is also responsible for managing the necessary infrastructure which supports its services.

Although these actors are the main stakeholders of a cloud environment, they don't represent the total number of actors. As this concept grown, several entities have tried to enter in this business by offering a number of activities related to the provisioning process between consumers and providers. From entities auditing cloud service implementations, to intermediaries and resellers that add functionality to existing services, or even organizations that provide tools for service implementation.

2.2.2 Cloud implementation models

The cloud implementation models are the different means by which consumers access cloud services. These models are different and are characterized by the target audience they serve. If this assumption is taken into account, there are only two models: private and public. Regarding the definition and characterization of

community and hybrid clouds, the NIST proposal was considered confusing, inconsistent and sometimes ambiguous (Chou, 2011).

2.2.2.1 Private Cloud

Private clouds are typically used in the distribution of services in an internal organization environment. The infrastructure that supports the private cloud can be delegated to a service provider and stays private at all, it has not to be acquired and managed by the private cloud users themselves.

When managed internally by users themselves, they have full control over the processes, data or applications used in the cloud, but they lose some of the general principles or benefits of cloud computing, such as access to infrastructure at reduced prices, elasticity, resources availability or rapid deployment times.

2.2.2.2 Public Cloud

Unlike private clouds, this model is used for the distribution of services to the general public, typically via the Internet. The service provider is responsible for the entire support infrastructure. This infrastructure is fully shared by the various users of this cloud. Thanks to this sharing and optimization of resource management processes, providers are able to maximize their use and enable them to be supplied at reduced prices to consumers. The separation of resources by users is merely logical and is done for example through the use of access credentials.

2.2.2.3 Other NIST proposed models

NIST says that a community cloud is targeted by a specific set of users from various organizations that have some kind of common interest. It also mentions that the infrastructure that supports a community cloud can belong to one or more of the involved organizations, by a service provider outside the community or by any combination of these.

Regarding the hybrid cloud, NIST describes it as a composition of different types of cloud mentioned earlier (private, public and community). The various clouds are interconnected through standard or proprietary means in order to allow the portability of data or applications between them.

2.3 What is a cloud data warehouse?

Cloud data warehousing was born from the convergence of three trends – huge changes in data sources, volume and complexity; the need for data access and analytics; and better technology that increased the efficiency of data access, analytics and storage. Traditional data warehouse systems were not designed to handle the volume, variety and complexity of today's data.

A data warehouse in the cloud is a database which information is consumed over the Internet, a typical database as a service (DBaaS). In Wikipedia Cloud Database is defined as: "With a database as a service model, application owners do not have to install and maintain the database themselves. Instead, the database service provider takes responsibility for installing and maintaining the database, and application owners are charged according to their usage of the service." ("Cloud database - Wikipedia," n.d.)

Cloud data warehousing is a cost-effective way for organizations to use and take advantage high technology without high upfront costs to purchase, install and configure the required hardware, software and infrastructure (Talia, 2013).

The various cloud data warehousing options are generally grouped into the following three categories (Snowflake, 2016):

- *Traditional data warehouse software deployed on cloud infrastructure:* this option is very similar to a conventional data warehouse, as it reuses the original code base. The IT expertise is still needed to build and manage the data warehouse. While you don't have to purchase and install the hardware and software, you may still have to do significant configuration and tuning, and perform operations such as regular backups;
- *Traditional data warehouse hosted and managed in the cloud by a third party as a managed service:* with this option, the third party provider supplies the IT expertise, but you're still likely to experience many of the same limitations of a traditional data warehouse. The DW is hosted on hardware installed in a data center managed by the vendor. The customer still has to specify in advance how much disk space and compute resources they expect to use;
- *A true SaaS data warehouse:* with this option, the vendor delivers a complete cloud data warehouse solution that includes all infrastructure, database and IT expertise required. Clients pay only for the storage and computing resources

they use, when they use them. This option should scale up and down on demand.

This means that organizations have to take into account some aspects when thinking about moving an on-premises data warehouse to the cloud or starting a new data warehouse project:

- Addresses current and future needs;
- The ability to integrate structured and unstructured data;
- Human resources knowledge of the new technology;
- The scalability of the solution;
- Security issues;
- Analytics support;
- Provisioning.

In terms of scalability the data warehouse should be able to scale both compute and storage independently so you are not forced to add more storage when you really just need more compute, and vice versa.

It should be possible to integrate structured and unstructured data without the need to install or configure additional software and using only one repository so the data can be all in one place avoiding the need to create data silos.

The human resources knowledge of the new technology trends is also an important point to focus on, not only inside organization but also on outsourcing company in order to see if there are people with good knowledge in the chosen technology.

About security issues, one of the concerns of who is moving to the cloud, the data confidentiality and privacy are de major ones (Guermazi et al., 2015). Another one is the relation between customer and cloud provider, mainly when the relation is about to end or when problems occur or even because the customer does not know where its data really is (“Cloud computing complicates customer-vendor relationships,” n.d.).

The provisioning of a data warehouse based in the cloud is also a problem when the source data is outside the cloud because of the duration of the transfer process.

3 Cloud Data Warehousing market solutions

In this chapter, the architecture of four cloud data warehousing solutions is described and listed other existing solutions in the market that were not analysed. The solutions identified and analysed are:

- Amazon Redshift;
- Microsoft Azure SQL Data Warehouse;
- Snowflake;
- Pivotal GreenPlum.

These four solutions were selected because the first two are the most popular and the other two because are the newest when compared with solutions not analyzed.

There are also other solutions that have not been analysed in detail, due to lack of time and lack of response from vendors to asked questions, but are identified in the following list:

- HPE Vertica;
- IBM dashDB;
- Teradata Cloud Data Warehousing and Analytics;
- Oracle Cloud;

- Google BigQuery.

3.1 Amazon Redshift

Gartner says “AWS is often considered the leading cloud data warehouse platform-as-a-service provider” (Gartner, 2016a).

Recognized by Gartner as leader, Amazon Redshift is a fast, fully managed, petabyte-scale data warehouse that it makes simple and cost-effective to analyze all our data using existing business intelligence tools.

Amazon Redshift engine is a SQL-compliant, massively parallel processing (MPP), query processing and database management system designed to support analytics workload. The storage and compute is distributed across on or more compute nodes (Gupta et al., 2015). The core infrastructure of Amazon Redshift data warehouse is a cluster and it is composed by:

- A leader node;
- One or more compute nodes.

The leader node accepts connections from the client applications and dispatch the work to the compute node: it parses and develops execution plans to carry out database operations and based on the execution plan it compiles code, distributes the compiled code to the compute nodes and assigns a portion of the data to each node (“Data Warehouse System Architecture - Amazon Redshift,” n.d.).

The leader node distributes SQL statements to the compute nodes only when a query references tables that are stored on the compute nodes, otherwise they run exclusively on the leader node (“Data Warehouse System Architecture - Amazon Redshift,” n.d.). In Figure 4 is presented Amazon Redshift system architecture.

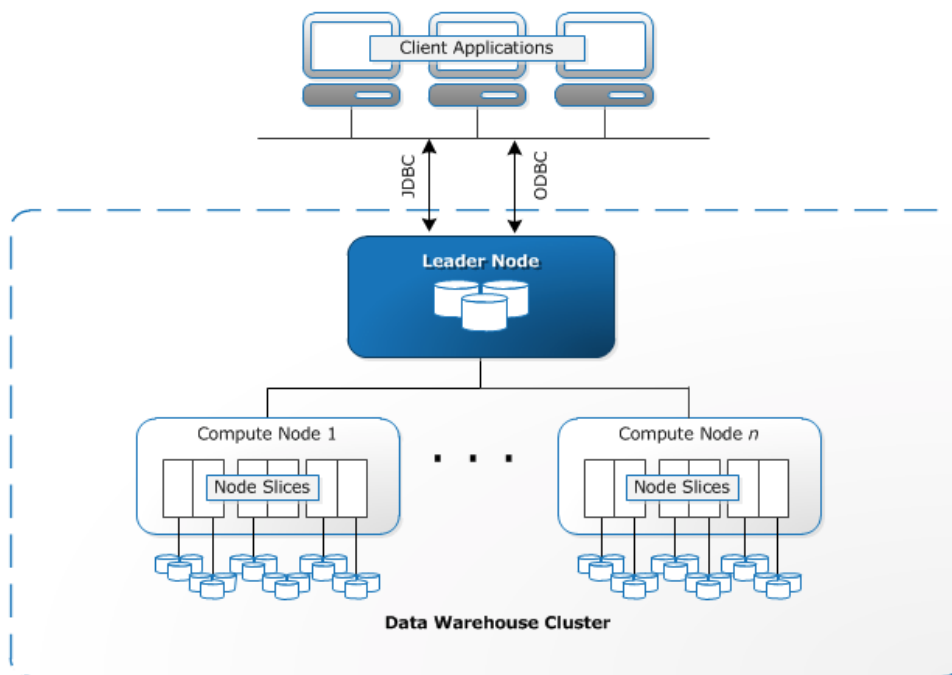


Figure 4 – Amazon Redshift system architecture (source: (“Data Warehouse System Architecture - Amazon Redshift,” n.d.))

The compute nodes execute the compiled code sent by the leader and send the results back for final aggregation. Each compute node has its own dedicated CPU, memory and storage – it is easy to scale the cluster by upgrading the compute nodes or adding new ones. The minimum storage for each compute node is 160GB and scale up to 16TB to support a petabyte of data or more (“Data Warehouse System Architecture - Amazon Redshift,” n.d.).

The compute node is partitioned into slices and each of it is allocated a portion of the node’s memory and disk space, where it processes a portion of the workload assigned to the node – the leader node manages the distribution of data of the workload to the slices and then they work in parallel to complete the operation (“Data Warehouse System Architecture - Amazon Redshift,” n.d.).

The cluster contains one or more databases. Amazon Redshift is a relational database management system and provides the same functionality as a typical RDBMS including all related with OLTP but it is optimized for high-speed performance analysis and reporting of very large datasets (“Data Warehouse System Architecture - Amazon Redshift,” n.d.).

The database engine is based on PostgreSQL. Another interesting characteristic of Amazon Redshift is that it is a columnar database, which means that each record is not saved as a unique block of data but it is stored in independent columns. The query

performance can greatly improve by selecting a limited subset of columns rather than the full record.

3.2 Microsoft Azure SQL Data Warehouse

“Microsoft Azure SQL data warehouse is a cloud-based, scale-out database capable of processing massive volumes of data, both relational and non-relational. It is a massively processing (MPP) distributed database system.” (“SQL Data Warehouse | Microsoft Azure,” n.d.)

It provides SaaS, PaaS and IaaS services and supports many different programming languages, tools and frameworks, including non-Microsoft software.

SQL Data Warehouse is based on the SQL Server relational database engine and integrates with the tool that its users may be familiar with, Management Studio to connect to the database and other Microsoft Tools. This includes (“SQL Data Warehouse | Microsoft Azure,” n.d.):

- Analysis Services;
- Integration Services;
- Reporting Services;
- Cloud-based tools.

The Microsoft Azure SQL Data Warehouse is composed by a Control Node, Compute nodes and Storage. It also has a service called Data Movement Service that is responsible for the data movement between the nodes (“SQL Data Warehouse | Microsoft Azure,” n.d.). In Figure 5 is presented Microsoft Azure SQL Data Warehouse system architecture.

Similar to Amazon Redshift leader node, the control node manages and optimizes queries and is responsible for the coordination of all the data movement and computation required to run parallel queries. When a request is made to SQL Data Warehouse, the control node transforms it into separate queries that run on each compute node in parallel.

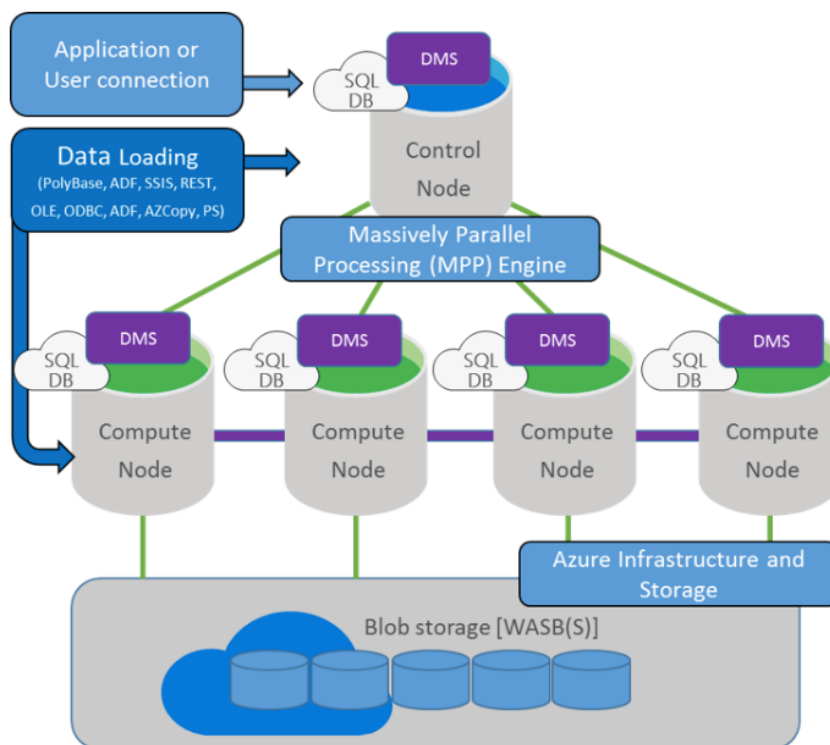


Figure 5 – Microsoft Azure SQL Data Warehouse system architecture (source: (“SQL Data Warehouse | Microsoft Azure,” n.d.))

The compute nodes are SQL databases that store the data and process queries. When data is added, it is distributed to the compute nodes and when the data is requested these nodes are the workers that run queries in parallel. After processing, they pass the results back to the control node so it can aggregate the results and return the final result (“SQL Data Warehouse | Microsoft Azure,” n.d.).

All the data stored in Azure SQL Data Warehouse is stored in Azure Blob Storage – it is a service that stores unstructured data in the cloud as objects/blobs. Blob Storage can store any type of text or binary data, such as a document, media file or application installer. When compute nodes interact with data, they write and read directly to and from blob storage. Compute and Storage are independent (“SQL Data Warehouse | Microsoft Azure,” n.d.).

As said before, Data Movement Service (DMS) is responsible for all the data movements between the nodes. It gives the compute nodes access to data they need for joins and aggregations. It is not an Azure service but a Windows service that runs alongside SQL Database on all the nodes and it is only visible on query plans because they include some DMS operations since data movement is necessary to run a query in parallel (“SQL Data Warehouse | Microsoft Azure,” n.d.).

3.3 Snowflake

Snowflake is a fully relational massively parallel processing database that can take advantage of a cloud infrastructure (“Snowflake Reinvents the Data Warehouse for the Cloud,” 2014). It is an analytic data warehouse provided as software-as-a-service and runs completely on cloud infrastructure. All components of snowflake run in a public cloud infrastructure, currently exclusively in the Amazon Web Services (AWS). Snowflake cannot be run on private cloud infrastructures (on-premises or hosted).

Snowflake is a mixture of traditional shared-disk and shared-nothing database architectures. Like a shared-disk architecture, it uses a central data repository for persisted data that is accessible from all compute nodes in the data warehouse. But similar to a shared-nothing architecture it processes queries using MPP compute clusters where each node of the cluster stores a portion of the dataset locally. This feature offers the best of the two architectures, simplicity of the shared-disk architecture with data store and management and the performance and scale-out of the shared-nothing architecture (“Key Concepts & Architecture — Snowflake Documentation,” n.d.). In Figure 6 is presented the Snowflake system architecture.

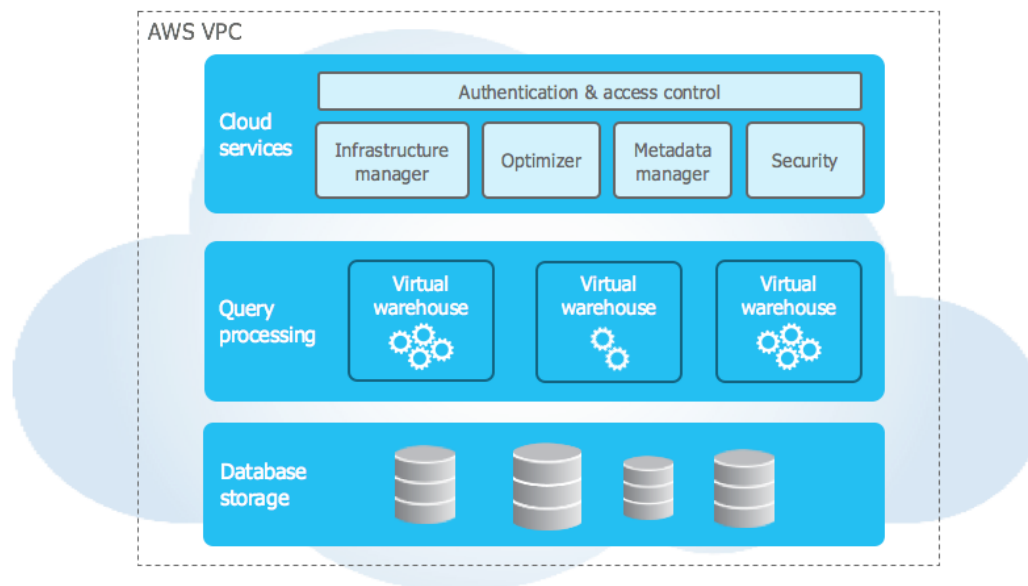


Figure 6 – Snowflake system architecture (source: (“Key Concepts & Architecture — Snowflake Documentation,” n.d.))

This hybrid architecture consists of three layers:

- Database storage;
- Query Processing;

- Cloud Services.

As referred before, Snowflake uses AWS to operate. The database storage is also in Amazon Web Services S3 (Simple Storage Service). When data is loaded into Snowflake, it reorganizes that data and stores it in the cloud storage. All aspects related with how data is stored in S3 are managed by Snowflake – “the organization, file size, structure, compression, metadata, statistics, and other aspects of data storage”. An important feature is that the data stored by Snowflake in S3 is not visible or accessible by customers unless they are running SQL query operations using Snowflake (“Key Concepts & Architecture — Snowflake Documentation,” n.d.).

Query execution is performed in the query processing layer and uses “virtual warehouses”. This virtual warehouse is a “massively parallel processing compute cluster composed of multiple compute nodes allocated by Snowflake from Amazon EC2”. Each virtual warehouse does not share compute resources with other virtual warehouses and as result they have no impact on the performance of the other compute nodes (“Key Concepts & Architecture — Snowflake Documentation,” n.d.).

The cloud services layer is a collection of services used to coordinate activities across Snowflake (Dageville et al., 2016). These services work together in order to process user requests from login to query dispatch. Cloud Services also runs on compute instances provisioned by Snowflake from Amazon EC2. The available services are (“Key Concepts & Architecture — Snowflake Documentation,” n.d.):

- Authentication and access control;
- Infrastructure management;
- Metadata management;
- Query parsing and optimization.

“Snowflake is designed to protect user data against attacks on all levels of the architecture, including the cloud platform. To this end, Snowflake implements two-factor authentication, (client-side) encrypted data import and export, secure data transfer and storage, and role-based access control for database objects” (Dageville et al., 2016).

Snowflake keeps metadata for every individual table file. The metadata not only covers plain relational columns, but also a selection of auto detected columns inside of semi-structured data.

All queries issued by users pass through the Cloud Services layer. Here, all the early stages of the query life cycle are handled: parsing, object resolution, access control, and plan optimization (Dageville et al., 2016).

Connecting to Snowflake is simple and has multiple ways to do it:

- Web-based user interface;
- Command line clients;
- ODBC and JDBC drivers;
- Native connectors;
- Third-party connectors used by ETL or BI tools.

3.4 Pivotal GreenPlum

Greenplum database is a massively parallel processing (MPP) database server with an architecture designed to manage large-scale analytic data warehouses and business intelligence workloads and it is based on a shared nothing architecture where each processor with its own memory, operating system and disks cooperate to carry out an operation, see Figure 7. Greenplum uses MPP high-performance system architecture to distribute the load of a data warehouse or process a query in parallel. It is based on PostgreSQL open-source technology (“About the Greenplum Architecture | Pivotal Greenplum Database Docs,” n.d.).

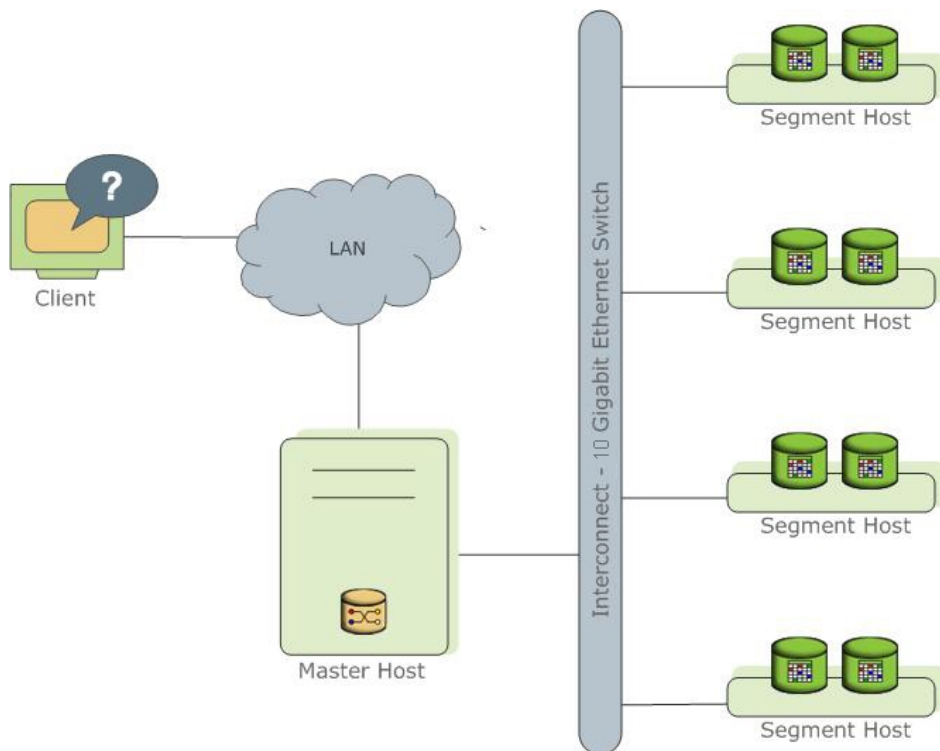


Figure 7 – Pivotal Greenplum system architecture (source: (“About the Greenplum Architecture | Pivotal Greenplum Database Docs,” n.d.))

The Greenplum architecture is composed by a Master, segments and interconnect components that work together to make up a Greenplum database system (GPDB).

The Greenplum database master is the entry to the GPDB, accepting client connections and queries, and distributing work to the segment hosts. The connection to the database can be made by using client programs or application programming interfaces (APIs) such as JDBC or ODBC (“About the Greenplum Architecture | Pivotal Greenplum Database Docs,” n.d.).

The master host does not contain user data. It contains a set of system tables that contain metadata about the GPDB itself. The user data only resides on the segment hosts (“About the Greenplum Architecture | Pivotal Greenplum Database Docs,” n.d.).

The work of the master host is to authenticate the client connections, process incoming SQL commands, distribute workloads among segment hosts, coordinate the results returned by each segment host, and finally present the final results to the client program (“About the Greenplum Architecture | Pivotal Greenplum Database Docs,” n.d.).

The segment hosts are independent PostgreSQL databases that each store a set of data and perform the majority of query processing.

To run in the cloud, Greenplum database can be deployed on Microsoft Azure or Amazon Web Services. It is possible to create clusters of Greenplum database and the use the database as a service.

3.5 Solutions Comparison

In this section is compared side by side a set of characteristics defined in the following list as shown in Table 2:

- Developer: who developed the database;
- MPP support: the ability to support massively parallel processing;
- SQL standard support: the support of the SQL standard;
- Scale up to: the scale up to the database can grow;
- In-memory capabilities: the ability to put data in-memory and use it;
- Availability in the cloud: the database is only available for cloud usage or not;
- License: the type of licensing;

The chosen characteristics were selected based on the knowledge base of relational and NoSQL database management systems and from the previous analysis of each of the solutions (“Cloud DW solutions Comparison,” 2017).

Table 2 — Database comparison characteristics

Characteristic/Database	Amazon Redshift	Microsoft Azure SQL Data Warehouse	Snowflake	Pivotal Greenplum
Developer	Amazon, based on PostgreSQL	Microsoft	Snowflake Computing Inc.	Pivotal Software Inc., based on PostgreSQL
MPP support	Yes	Yes	Yes	Yes
SQL standard support	Does not fully support	Yes	Yes	Yes
Scale up to	PetaByte	PetaByte	PetaByte	PetaByte
Can pause resources	No	Yes	Yes	Yes

In-memory capabilities	Yes	No	No	No
Only available in the cloud	Yes	Yes	Yes	No
License	Commercial	Commercial	Commercial	Open-source

There are some common characteristics in the four compared solutions. One is the fact that RedShift and Greenplum, both are based on PostgreSQL database engine. All of them support MPP and scale up to Petabyte. The SQL standard is support in all these solutions but RedShift not fully supports it, more details about RedShift SQL standard support are presented in section 5.1.1.

In order to save costs, the ability of pausing resources is a must have but this is not present in RedShift. In the other hand, Redshift offers in-memory capabilities that are not present in the other three solutions. Finally, all of the solutions are available in the cloud but Greenplum is also available to install on-premises.

4 Methodology

This chapter presents in detail the criteria that will support the decision of what solution will be target of the described test.

4.1 How to evaluate and compare

To evaluate and compare the selected solutions will be used the Gartner criteria defined on the “Gartner 2016 Magic Quadrant for Data Warehouse and Database Management Solutions for Analytics” and “Gartner 2016 Magic Quadrant for Cloud Infrastructure as a Service, worldwide”. It will be also considered the 2017 Rightscale survey about “The State of Cloud” which covers a huge broad of criteria.

From the Gartner’s review criteria, the following have been selected due to be the ones that can be measured:

- *Product/Service*: This criterion relates to the product capability to support a data warehouse and its workloads. For a cloud-based solution, the existing capacity and the possibility of increasing the infrastructure is evaluated in a self-service perspective. (Gartner, 2016a) (Gartner, 2016b);
- *Pricing*: This criterion examines the price and pricing models of the DBMS. (Gartner, 2016a);

Other Gartner’s review criteria like product viability, customer experience, sales execution, operations and innovation were not select due to be difficult to obtain information to do an evaluation. All these criteria are fully described in Annex D at question 6.

From the Rightscale survey results, the following have been selected, due to be the ones that are related with this work theme, to be analysed and used as input for the previous criteria evaluation:

- Enterprise public cloud adoption;
- SMB public cloud adoption.

After the comparison of the previous evaluation, the best two solutions will be evaluated using the following metrics:

- Elapsed time of the transfer and store process in the cloud;
- Elapsed time of querying the database in the cloud.

The previous metrics were selected because one of the main concerns of having data in the cloud is the time to put there the data and in the other hand the time to consume that data. Evaluating this metrics can lead to prove that the elapsed time to do the transfer and consumption of data is or not an issue.

It was used a public version of Microsoft Adventure Works DW database as our data source with data provided by Microsoft, explained in detail in section 4.2.2.1.

4.2 Hypotheses and evaluation methodologies

In this section are presented the test and environment conditions and also the test cases that will support the result of the work.

4.2.1 Architecture

The planned architecture for the project is composed by a source database and a cloud infrastructure with storage and database engine.

The source database was supported by Microsoft SQL Server 2012 database engine and was installed on commodity hardware.

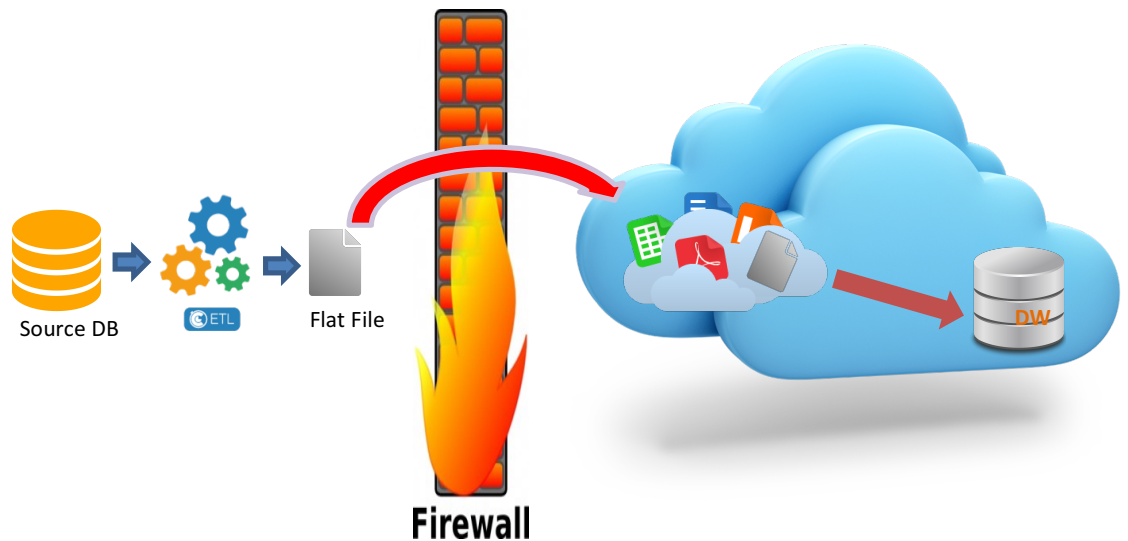


Figure 8 – Test system configuration

To get the data to transfer up into the cloud storage was used the export utility provided by SQL Server and then a Python script to transfer the data up into the cloud, to see the size of the files created and its number of records see section 4.2.1.1.

Once the file is transferred and stored in the cloud storage, the next step is to copy its data into the database. The source database is Microsoft Adventure Works DW model which is of free usage. Further information about this database is presented in section 4.2.2.1.

4.2.2 Experimental setup

The environment is composed by a computer where the source database is hosted. This computer has a double core CPU, 4 GB of RAM and a storage drive with 320 GB capacity. The operating system installed is Microsoft Windows 10 Professional and the database is a Microsoft SQL Server 2012 Standard Edition.

Figure 9 shows the test environment configuration, where this computer is not connected directly to the Internet, it is connected to an intermediate router and this router is connected to another router that is connected to the Internet. The connection between the two routers is made by an Ethernet connection at 100 Mbps

and the connection between the computer and the router is done using a wireless connection at 150 Mbps.

The router connected to the Internet has a firewall configured with maximum security settings, as follows:

- Inbound policy set to deny;
- Outbound policy set to deny;
- Only the services in the list have granted access in the outbound policy: DHCP, DNS, IMAP, SMTP, POP3, HTTP, HTTPS, FTP and Telnet.

The other router has the firewall completely disabled and has other devices connected to it only for internal usage.

Finally, the cloud infrastructure will be supplied by the solution where the tests will be done.

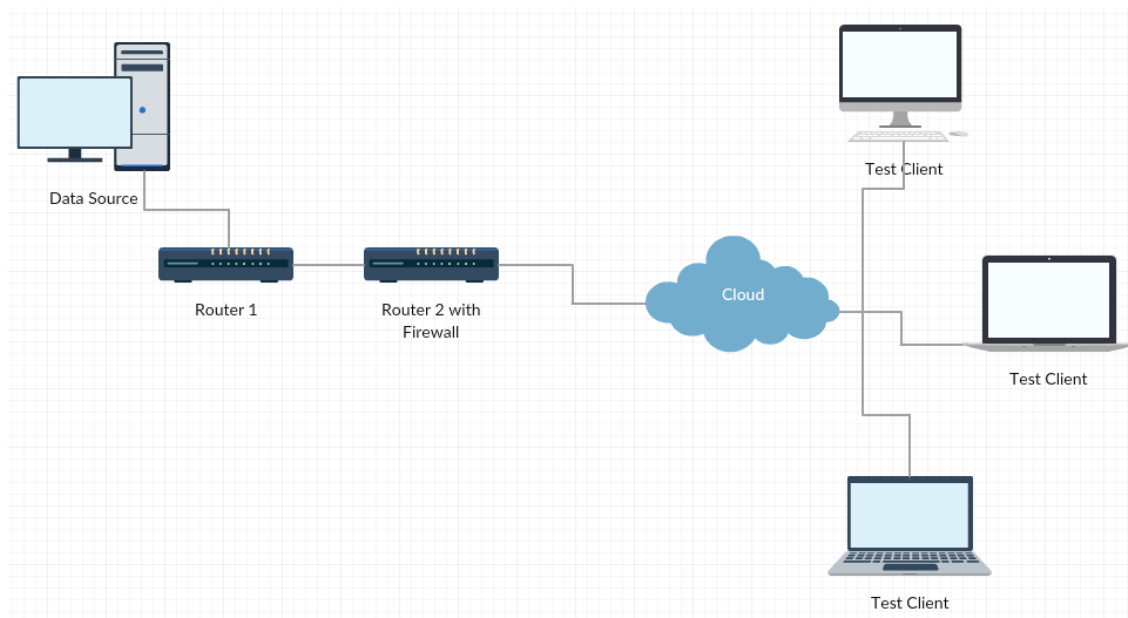


Figure 9 – Test environment configuration

The Internet connection that will support the data transfer to the cloud is not a dedicated line and it has a maximum download speed of 130 Mbps and 30 Mbps of maximum upload speed.

There will be clients accessing the cloud resources to consume database information or other relevant information stored in the cloud.

4.2.2.1 Adventure Works DW database model

“Adventure Works DW contains a subset of the tables from the OLTP database, in addition to financial information that is pulled from a separate data source. Adventure Works DW contains two subject areas, finance and sales.” (“Adventure Works Sample Data Warehouse,” n.d.)

The database model is composed by 25 tables: 16 dimension tables, 8 fact tables and a table with prospective buyer information.

Figure 10 shows the view of one of the Adventure Works DW subject areas.

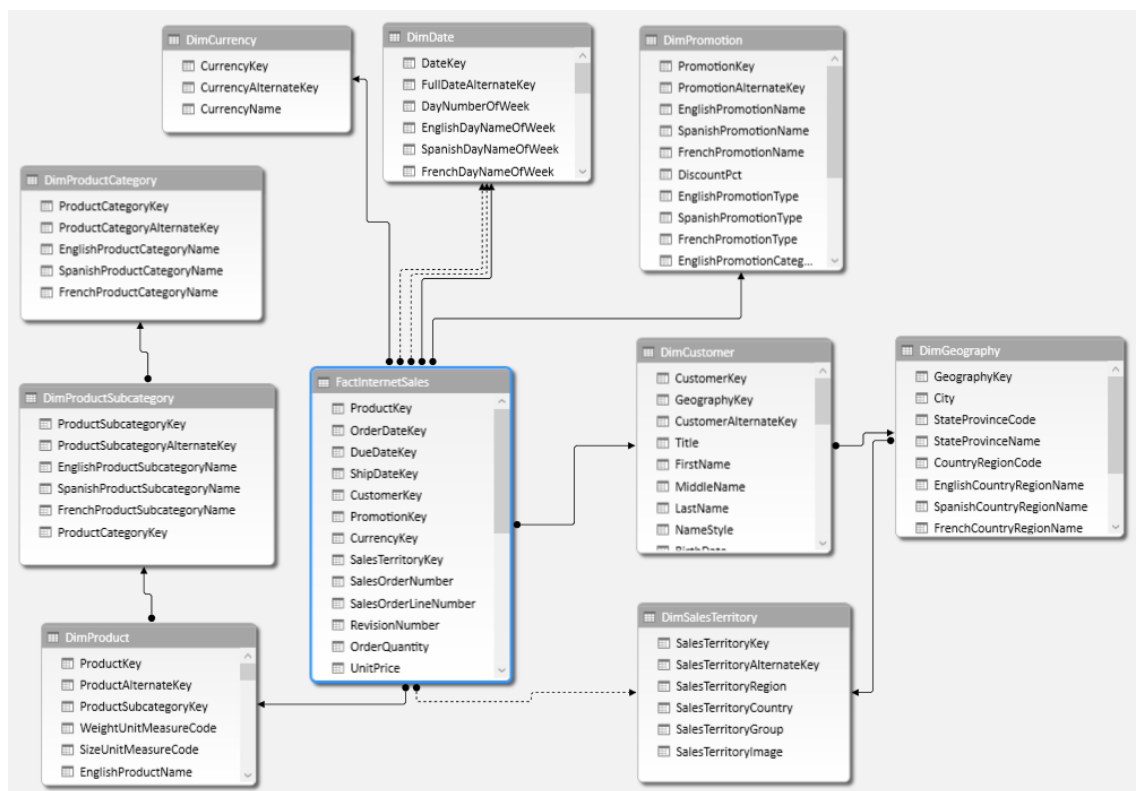


Figure 10 – Source and destination database data model of sales subject area

In the Annex C is the complete data model with all the database tables.

4.2.2.2 Database model tables

In this section are presented the database dimension and fact tables.

4.2.2.2.1 Dimension tables

The structure of the dimension table “Account” is listed below. In the Annex A are the other dimension tables.

Table 3 — Account dimension table

Column	Data type	Nullability
AccountKey	int	NOT NULL
ParentAccountKey	int	NULL
AccountCodeAlternateKey	int	NULL
ParentAccountCodeAlternateKey	int	NULL
AccountDescription	nvarchar(50)	NULL
AccountType	nvarchar(50)	NULL
Operator	nvarchar(50)	NULL
CustomMembers	nvarchar(300)	NULL
ValueType	nvarchar(50)	NULL
CustomMemberOptions	nvarchar(200)	NULL

4.2.2.2.2 Fact tables

The structure of the fact table “Survey response” is listed below. In Annex B are the other fact tables.

Table 4 — Survey response fact table

Column Name	Data type	Nullability
SurveyResponseKey	int	NOT NULL
DateKey	int	NOT NULL
CustomerKey	int	NOT NULL
ProductCategoryKey	int	NOT NULL
EnglishProductCategoryName	nvarchar(50)	NOT NULL
ProductSubcategoryKey	int	NOT NULL
EnglishProductSubcategoryName	nvarchar(50)	NOT NULL

4.2.2.2.3 Flat files size and number of records

The size of the data exported to flat file and the number of records in each file is presented in Table 5.

Table 5 — File size and number of records

Filename	Size (KB)	Number of records
DimAccount.txt	11	99
DimCurrency.txt	4	105
DimCustomer.txt	8927	18484
DimDate.txt	231	1188
DimDepartmentGroup.txt	1	7
DimEmployee.txt	140	296

DimGeography.txt	96	655
DimOrganization.txt	1	14
DimProductCategory.txt	1	4
DimProductSubcategory.txt	3	37
DimPromotion.txt	8	16
DimProduct.txt	988	1017
DimReseller.txt	240	701
DimSalesReason.txt	1	10
DimSalesTerritory.txt	1	11
DimScenario.txt	1	3
FactCallCenter.txt	16	120
FactCurrencyRate.txt	1415	14264
FactFinance.txt	2719	39409
FactInternetSales.txt	14877	60398
FactInternetSalesReason.txt	1685	64515
FactResellerSales.txt	19265	60855
FactSalesQuota.txt	12	163
FactSurveyResponse.txt	246	2727
ProspectiveBuyer	380	2059

It was also created a file with larger volume and number of records to use in the tests, see Table 6.

Table 6 — FactResellerSales file with larger volume

Filename	Size (KB)	Number of records
FactResellerSales_GB.txt	924705	5842080

4.2.3 Test cases

In this section we present two scenarios that will be used to evaluate the selected solutions.

4.2.3.1 First scenario

The first test case scenario is to measure the elapsed time of the transfer process of the data files up into the cloud. The time will be measured using programming techniques to be the most precise as possible. It was used a Python script to do the files upload.

After the files upload into the cloud, another metric to be taken is the elapsed time of the process that stores the data in the data warehouse database that is also in the cloud.

4.2.3.2 Second scenario

The second scenario has to be tested after the conclusion of the previous one and corresponds to a load test to evaluate the performance of querying the data warehouse in the cloud and compare the elapsed time of the query with the ones done in the source database, see Annex L to look at the queries. For the consumption of the data in the cloud database, depending on the chosen solution, will be used a web service.

5 Experimental Evaluation

In this section is presented the evaluation and comparison of solutions.

5.1 Cloud solutions comparison

Using the Gartner selected criteria, the identified solutions were evaluated and the result is presented in Table 7 and Table 8. The detailed information is in Annex F. Four variables were analysed:

- Data warehouse capacity, in order to evaluate if the solution is oriented for data warehouse;
- Full-cloud based solution, in order to evaluate if the solution have all components provided by a single entity and not shared between two or more entities;
- Self-Service capacity, in order to evaluate if the customer have the capacity to request a service and it is automatically provided;
- Price, in order to evaluate the cost of the solution. In this field it is important to say that all simulations were based on equivalent infrastructure properties.

Table 7 — Solutions evaluation by Gartner Criteria (Studied Cases)

	Data Warehouse Capacity	Full-Cloud based solution	Self-Service Capacity	Price € (mo)
Redshift	Yes	Yes	Yes	645,25
Azure SQL Data Warehouse	Yes	Yes	Yes	1 251,45
Snowflake	Yes	No	Yes	1 301,44
Pivotal Greenplum	Yes	No	Yes	not available

Table 8 — Other solutions evaluation by Gartner Criteria

	Data Warehouse Capacity	Full-Cloud based solution	Self-Service Capacity	Price € (mo)
HP Vertica	Yes	No	Yes	not available
IBM dashDB	Yes	Yes	Yes	not available
Teradata	Yes	No	Yes	13 204,46
Oracle Cloud	Yes	Yes	Yes	3 473
Google BigQuery	Yes	Yes	Yes	1 371,90

All the solutions identified in Table 7 were analysed in detail in chapter 3 and conclusions are:

- All the solutions have capabilities in providing a data warehouse environment with self-service capacity;
- Only RedShift and Azure have a full-cloud based solution;
- Only RedShift and Azure fully satisfy the defined criteria.

After this comparison, the two solutions that fully satisfies the defined criteria were also compared by the result of the survey about “The State of the Cloud” done by RighthScale.

As referred before, the input for the comparison using the results of this survey are the solutions select by Gartner criteria:

- Amazon RedShift
- Microsoft Azure SQL Data Warehouse

These two solutions were compared in terms of adoption in two different perspectives:

- Enterprise public cloud adoption;
- SMB public cloud adoption.

Respondents were asked to tell which clouds they were using and whether they were running applications in cloud, experimenting with cloud, planning to use cloud, or had no plans to use cloud. The respondents are mainly from North America and from industries like Software, Tech Services, Telecom and Financial Services corresponding to 1002 respondents.

The aggregated results of public cloud adoption of enterprise and small-medium businesses are shown in Figure 11.

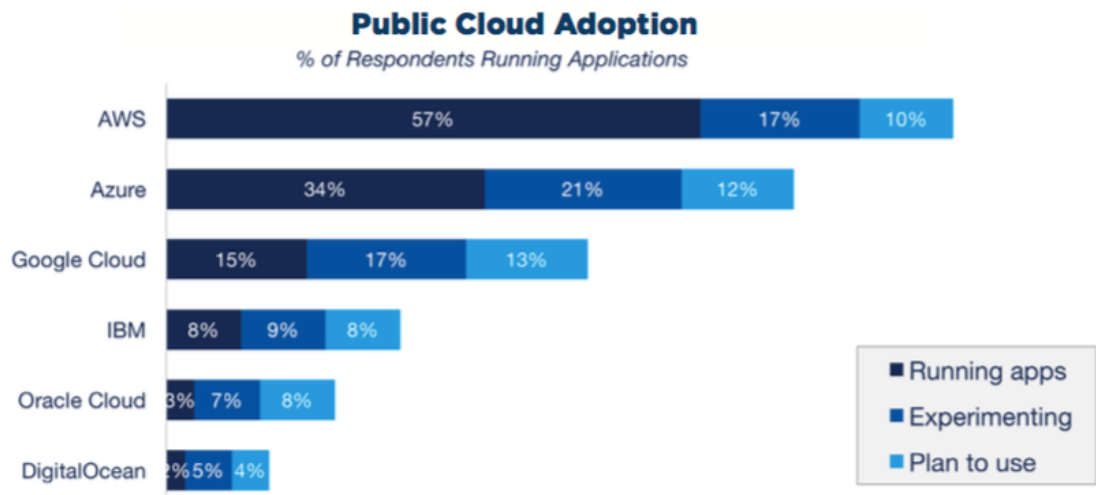


Figure 11 – Public cloud adoption survey results

Comparing the adoption of public clouds, in particular AWS and Azure, we can see that AWS is the number one adopted cloud solution for the most respondent users of the 2017 Rightscale survey, as shown in Figure 11.

Although AWS continues to lead in public cloud adoption (57 percent of respondents currently run applications in AWS, see Figure 12), this number has stayed the same as in 2016.

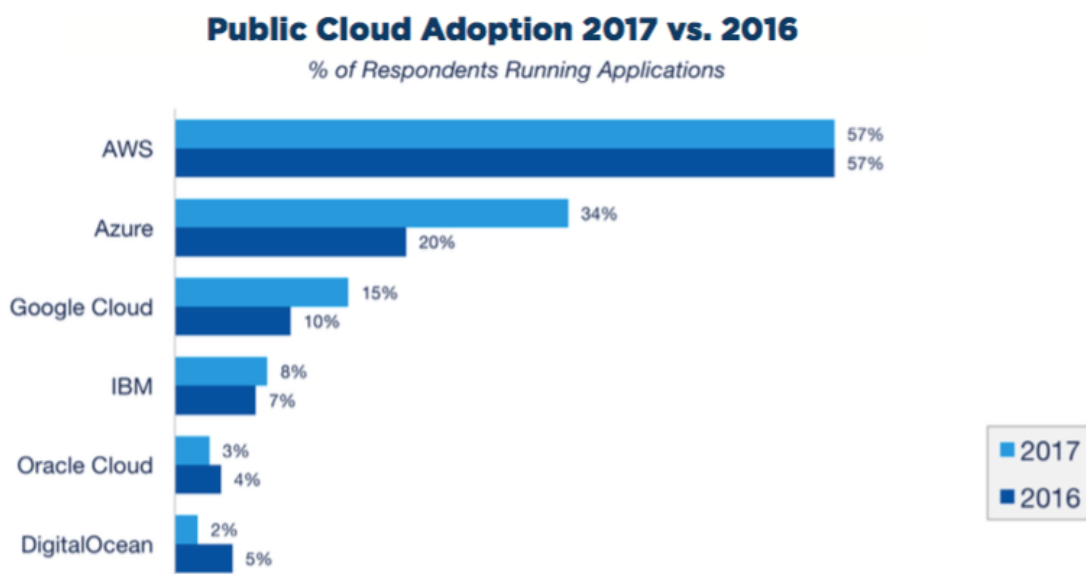


Figure 12 – Public cloud adoption survey results – 2017 vs 2016

In contrast, over the last year, happened a significant growth in the percentage of respondents running applications in Azure and Google, the second and third public cloud providers (see Figure 12). Overall Azure adoption grew from 20 to 34 percent of respondents, reducing the AWS lead. Google also increased from 10 to 15 percent.

Among enterprises, AWS grew slightly over last year, now with adoption at 59 percent compared to 56 percent last year, as shown in Figure 13. Microsoft Azure surged from 26 to 43 percent and Google from 9 to 15 percent.

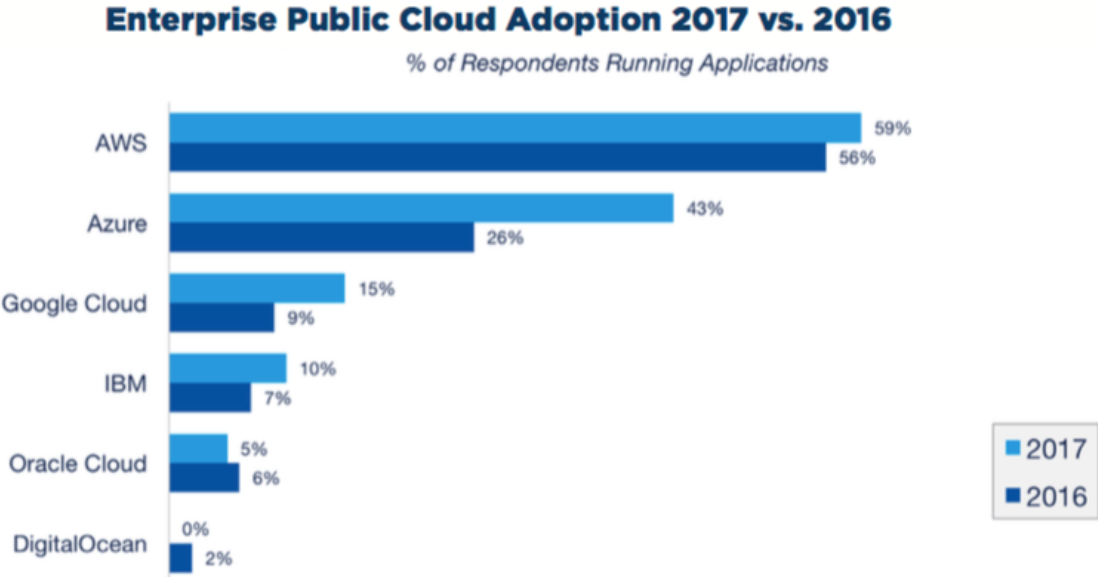


Figure 13 – Enterprise Public cloud adoption survey results – 2017 vs 2016

Respondents with future projects show the most interest in Azure with 32 percent (combination of experimenting and planning to use), see Figure 14.

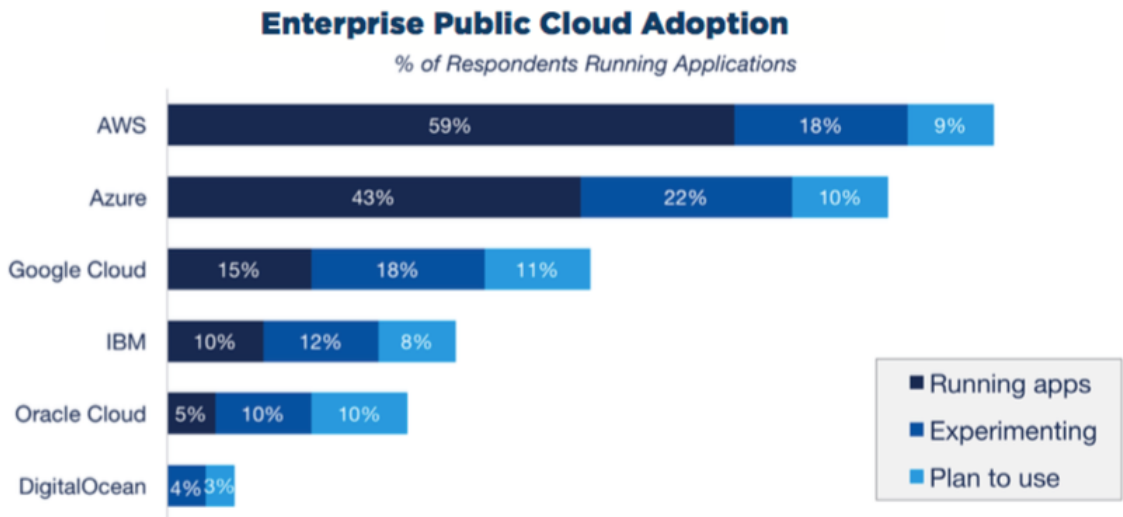


Figure 14 – Enterprise Public cloud adoption survey results

Among smaller organizations, AWS is still in first position despite a slight decline in adoption (from 58 to 55 percent). Azure increased adoption significantly (from 15 to 25 percent) as shown in Figure 15.

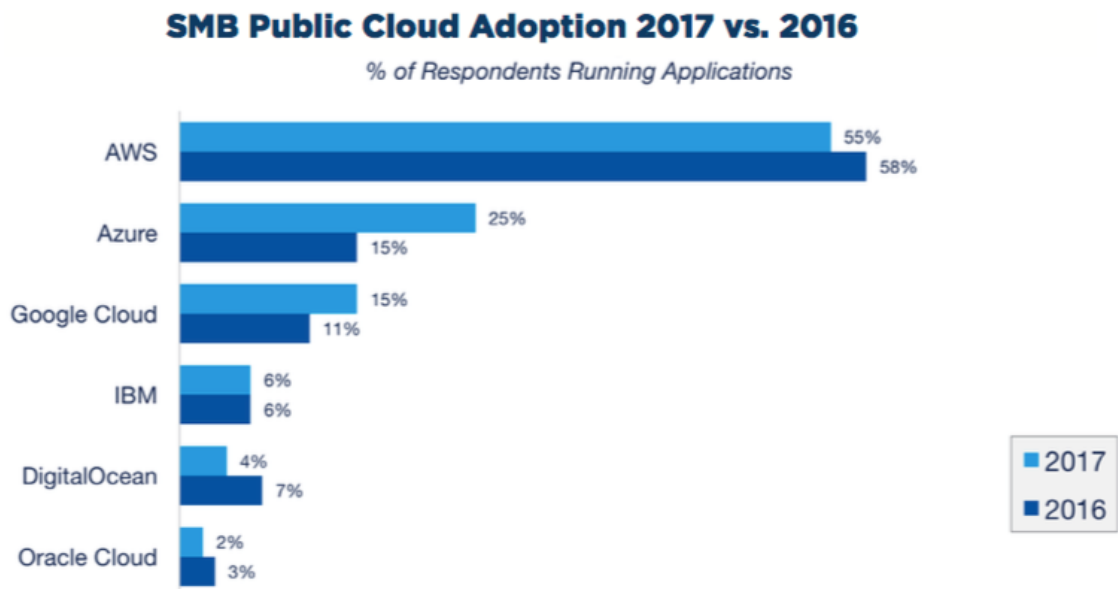


Figure 15 – SMB Public cloud adoption survey results – 2017 vs 2016

Azure also has the highest number of respondents that are experimenting or planning to use, see Figure 16.

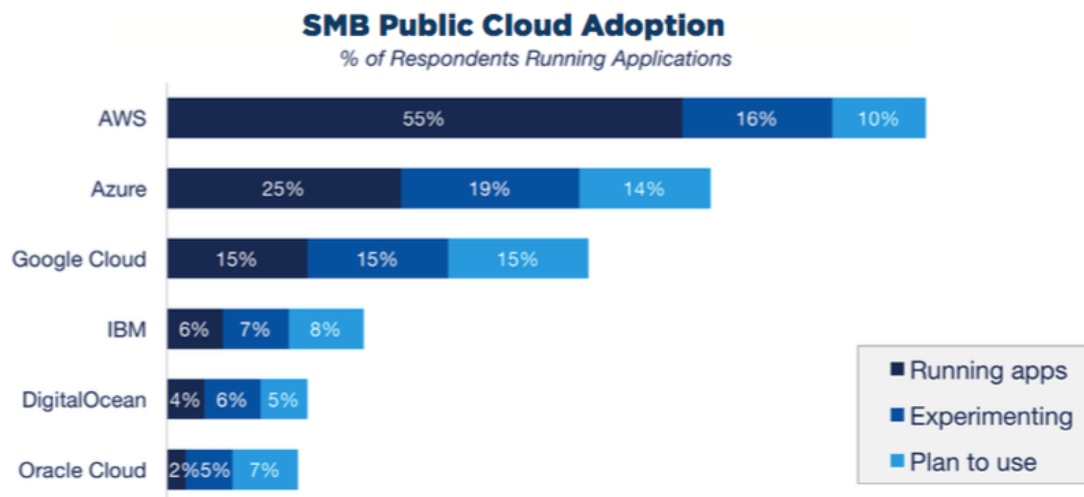


Figure 16 – SMB Public cloud adoption survey results

From this survey, the main conclusion is that Microsoft Azure is increasing adoption among small-medium businesses and bigger enterprises.

5.1.1 Redshift vs Azure SQL Data Warehouse System properties comparison

RedShift and Azure SQL have a database model based on relational database management system (RDBMS) which supports the relational data model.

Amazon Redshift is built around industry-standard SQL, with added functionality to manage very large datasets and support high-performance analysis that data. Although it is based on PostgreSQL, there are some unsupported features, data types and functions. Some SQL features are also implemented differently, for example:

- CREATE TABLE
- ALTER TABLE
- INSERT, UPDATE and DELETE

Amazon Redshift does not support tablespaces, table partitioning, inheritance, and certain constraints. The Amazon Redshift implementation of CREATE TABLE enables users to define the sort and distribution algorithms for tables to optimize parallel processing. ALTER COLUMN actions are not supported. ADD COLUMN supports adding only one column in each ALTER TABLE statement.

Using INSERT, UPDATE, and DELETE the WITH is not supported. For the complete list of unsupported features, data types and functions, our suggestion is to check on AWS documentation, in particular the one that concerns about Amazon Redshift and PostgreSQL.

Table 9 — Redshift and Azure SQL system properties

<i>System Properties</i>	Amazon Redshift	Microsoft Azure SQL Data Warehouse
Database model	Relational DBMS	Relational DBMS
Developer	Amazon (based on PostgreSQL)	Microsoft
License	Commercial	Commercial
Cloud based	Yes	Yes
Implementation language	C	C++
XML support	No	Yes
SQL standard support	Does not fully support	Yes
Supported programming languages	All languages supporting JDBC/ODBC	.Net, Java, JavaScript PHP, Python Ruby
Server-side scripts	User defined Python functions	Transact SQL
Support for concurrent data manipulation	Yes	Yes
MapReduce API support	No	No
In-memory support	Yes	No
Control over node configuration	Yes	No

In-Memory OLTP is a technology for optimizing performance of transaction processing, data ingestion, data load, and transient data scenarios. In-Memory support is available on RedShift but not in Azure SQL Data Warehouse because it is only available for OLTP workloads in SQL Server since version 2014 and Azure SQL Database.

In MapReduce support property, both databases do not offer an API for user-defined Map/Reduce methods. In case of RedShift, it is possible to combine MapReduce with RedShift by processing input data with MapReduce and import results to RedShift. In case of Azure SQL Data Warehouse, Polybase (barbkess, 2017) unifies data in relational data stores with non-relational ones, combining data from both RDBMS and Hadoop so that users don't need to understand HDFS or MapReduce.

Redshift and Azure SQL Data Warehouse offer many similar capabilities, so it is not necessarily a matter of one provider being better or worse than the other. It all depends on what our business needs, but each solution has pros and cons. See Table 5 to find some pros and cons of Amazon RedShift and Microsoft Azure SQL Data Warehouse cloud solutions.

Table 10 — Pros and Cons of Redshift and Azure SQL Data Warehouse

	Pros	Cons
Amazon Redshift	Performance through use of local storage	Compute cannot be scaled independent of storage (and vice versa)
	Loading data from S3 is very fast	Can't pause resources
	Beyond Petabyte	Queries that require joins against multiple columns can suffer in performance
	Columnar data store allows high performance queries on large volumes of data	
	Familiarity with PostgreSQL makes adopting Redshift easier	
Microsoft Azure SQL Data Warehouse	Resources can be paused during idle time in workload	Can only run 32 concurrent queries (maximum)
	Scale separate compute and storage resources and pay only for what is used	Not fully supports T-SQL
	Excellent integration with Azure Services	

Some conclusions concerning the two cloud data warehouse solutions have been taken:

- Using Redshift to scale our data warehouse we must increase both the compute and storage units. With Azure SQL DW, compute and storage is decoupled so we can scale them individually. This is a very different economic model that can save customers a lot of money as they don't have to purchase additional storage when they just need more compute power, or vice-versa;
- Azure SQL DW has the ability to pause compute when not in use so we only pay for storage, as opposed to Redshift in which we are billed 24/7 for all the virtual machines that make up the nodes in our cluster;
- RedShift is easier to configure than Azure SQL Data Warehouse and takes less time to be online and available after its setup.

5.1.2 Redshift Configuration and Test

It was configured a standard cluster to support the Amazon Redshift database with the following configuration:

- 1 Computer node with 2 virtual cores
- 15 GB of RAM
- 160 GB of SSD storage

The resume of the cluster configuration is shown in Figure 17.

The screenshot displays the Amazon Redshift console interface for a cluster named 'pedro-mestrdo'. The cluster status is 'available', the database health is 'healthy', and it is not in maintenance mode. The endpoint is 'pedro-mestrdo.cbtwsdsybaam.us-east-2.redshift.amazonaws.com:5439 (authorized)'. The console is divided into several sections:

- Cluster Properties:**
 - Cluster Name: pedro-mestrdo
 - Node Type: dc1.large
 - Nodes: 1
 - Zone: us-east-2a
 - Cluster Parameter Group: default.redshift-1.0 (in-sync)
 - Enhanced VPC Routing: No
- Cluster Status:**
 - Cluster Status: available
 - Database Health: healthy
 - In Maintenance Mode: no
 - Parameter Group Apply Status: in-sync
 - Pending Modified Values: None
- Cluster Database Properties:**
 - Port: 5439
 - Database Name: dw
 - Master Username: pedro
 - Encrypted: No
- Backup, Audit Logging, and Maintenance:**
 - Automated Snapshot Retention Period: 1
 - Cross-Region Snapshots Enabled: No
 - Audit Logging Enabled: No
 - Maintenance Window: wed:07:30-wed:08:00
 - Allow Version Upgrade: Yes
- Tags:**
 - You have not created any tags. Please add tags using the Manage Tags button above.

Figure 17 – Cluster configuration resume

The ability to create a Redshift cluster is fast, the creation of the cluster used in this project was done in a minute. The cluster took 12 minutes to be ready after the order of creation.

To connect to the database from the Internet, a security rule was configured in order to allow connections from the outside. For this test was created a rule for a specific IP address, see Figure 18.

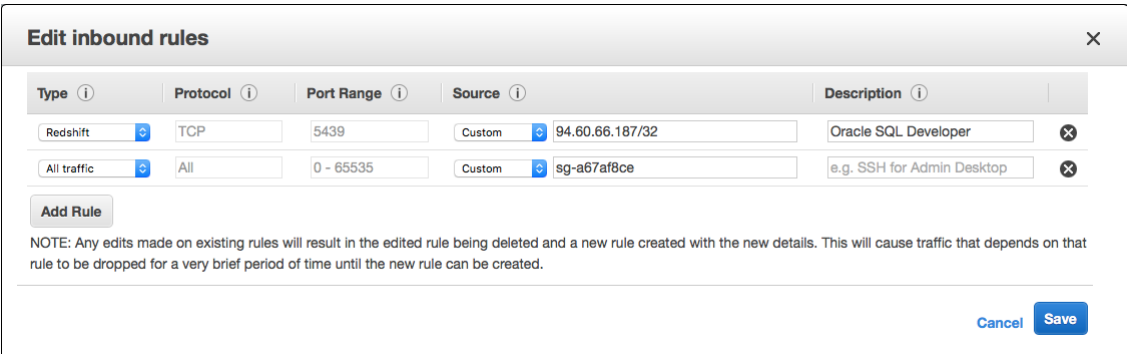


Figure 18 – Rule configuration

This is a mandatory step to be able to connect to the database from the outside.

The connection to the database was made by Oracle SQL Developer, see Figure 19, configured to support connection to a PostgreSQL database. The connection to the database is made using JDBC connector with the JDBC connection string provided by AWS for the configured database.

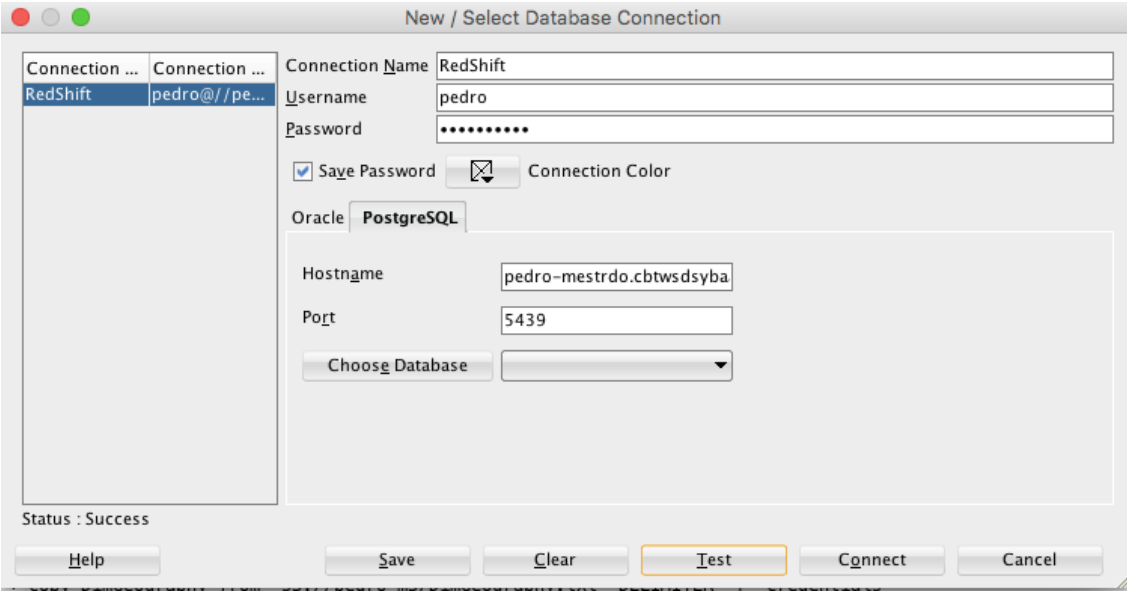


Figure 19 – Oracle SQL Developer connection configuration

After the connection to the database, the tables that support the test were created using the scripts provided by Microsoft and adapted the RedShift database engine. The scripts used are in Annex G.

As referred before, the recommendation of Amazon to load data into RedShift is to use S3 storage. All the files used to load the created tables were uploaded to S3. The upload was made using the AWS framework and a program written in Python, code in Annex H. The results of the upload time are shown in Table 11.

Table 11 — Upload elapsed time to S3 Storage

Filename	Time elapsed (ss.ms)
DimAccount.txt	0.331897
DimCurrency.txt	0.293321
DimCustomer.txt	1.816410
DimDate.txt	0.348714
DimDepartmentGroup.txt	0.293161
DimEmployee.txt	0.331219
DimGeography.txt	0.289604
DimOrganization.txt	0.285157
DimProduct.txt	0.388092
DimProductCategory.txt	0.289657
DimProductSubcategory.txt	0.340574
DimPromotion.txt	0.279039
DimReseller.txt	0.334619
DimSalesReason.txt	0.290870
DimSalesTerritory.txt	0.292220
DimScenario.txt	0.288650
FactCallCenter.txt	0.287555
FactCurrencyRate.txt	0.389258
FactFinance.txt	0.601055
FactInternetSales.txt	1.395725
FactInternetSalesReason.txt	0.379334
FactResellerSales.txt	4.421550
FactSalesQuota.txt	0.349143
FactSurveyResponse.txt	0.329648
ProspectiveBuyer.txt	0.382752
Total	15.029224

At this time the tables are created in RedShift database and the files with data are stored in S3 storage. The next step is loading the data into the database using the

COPY command (“COPY - Amazon Redshift,” n.d.). An example of the COPY command is presented next:

```
copy DimEmployee from 's3://pedro-m3/DimEmployee.txt' DELIMITER '|' credentials
'aws_access_key_id=AKIAJUOL3JIJO3MZZLCA;aws_secret_access_key=VZeCODxBBtpK
A9ACo3Y8GbQyryX9IGcwOleQo0c7';
```

The script with all loads is in Annex I. The Table 12 shows the time of the loading process.

Table 12 — Elapsed time of Loading data into RedShift table

Table Name	Start	End	Elapsed (sec)
DimAccount	2017-08-11 00:30:48.450859	2017-08-11 00:30:50.814802	2.363943
DimCurrency	2017-08-11 00:33:06.978451	2017-08-11 00:33:08.993621	2.01517
DimCustomer	2017-08-11 00:34:29.641646	2017-08-11 00:34:33.33977	3.698124
DimDate	2017-08-11 00:35:25.25155	2017-08-11 00:35:27.968862	2.717312
DimDepartmentGroup	2017-08-11 00:35:37.272087	2017-08-11 00:35:39.614611	2.342524
DimEmployee	2017-08-11 00:37:51.787847	2017-08-11 00:37:55.492763	3.704916
DimGeography	2017-08-11 00:38:13.51268	2017-08-11 00:38:15.936346	2.423666
DimOrganization	2017-08-11 00:38:29.146635	2017-08-11 00:38:31.49024	2.343605
DimProductCategory	2017-08-11 00:38:46.990817	2017-08-11 00:38:49.175663	2.184846
DimProductSubcategory	2017-08-11 00:39:03.197947	2017-08-11 00:39:05.352683	2.154736
DimPromotion	2017-08-11 00:39:35.94235	2017-08-11 00:39:38.683624	2.741274
DimProduct	2017-08-11 00:46:51.138273	2017-08-11 00:46:55.686192	4.547919
DimReseller	2017-08-11 00:50:50.480233	2017-08-11 00:50:53.412731	2.932498
DimSalesReason	2017-08-11 00:51:05.584227	2017-08-11 00:51:07.714588	2.130361
DimSalesTerritory	2017-08-11 00:51:15.328763	2017-08-11 00:51:17.691267	2.362504
DimScenario	2017-08-11 00:51:25.993011	2017-08-11 00:51:28.283244	2.290233
FactCallCenter	2017-08-11 00:51:54.532987	2017-08-11 00:51:56.86828	2.335293
FactCurrencyRate	2017-08-11 00:52:05.923877	2017-08-11 00:52:07.977028	2.053151
FactFinance	2017-08-11 00:52:23.58125	2017-08-11 00:52:25.831364	2.250114
FactInternetSales	2017-08-11 00:53:07.658283	2017-08-11 00:53:10.758176	3.099893
FactInternetSalesReason	2017-08-11 00:53:22.197651	2017-08-11 00:53:24.394749	2.197098
FactResellerSales	2017-08-11 00:54:08.018795	2017-08-11 00:54:11.45517	3.436375
FactSalesQuota	2017-08-11 00:54:25.609758	2017-08-11 00:54:27.669916	2.060158
FactSurveyResponse	2017-08-11 00:54:36.339623	2017-08-11 00:54:38.562756	2.223133
ProspectiveBuyer	2017-08-11 00:55:20.887202	2017-08-11 00:55:24.318819	3.431617
Total			66.040463

5.1.3 Azure SQL Data Warehouse Configuration and Test

For this test it was configured a 100 DWU capacity Azure SQL Data Warehouse with default settings, see Figure 20.

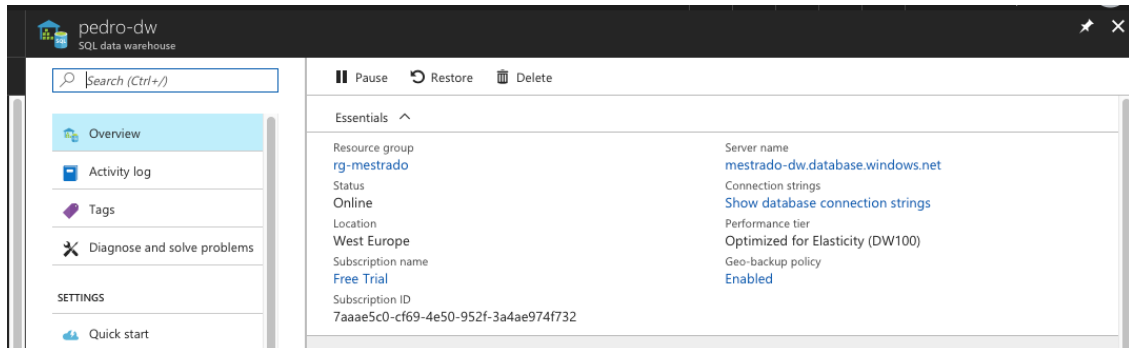


Figure 20 – Azure SQL Data Warehouse overview

It was also necessary to configure a Blob Storage account to be the file repository and a Data Factory to be able to do the data loading capacity of Azure. The resume of the necessary components is shown in Figure 21.

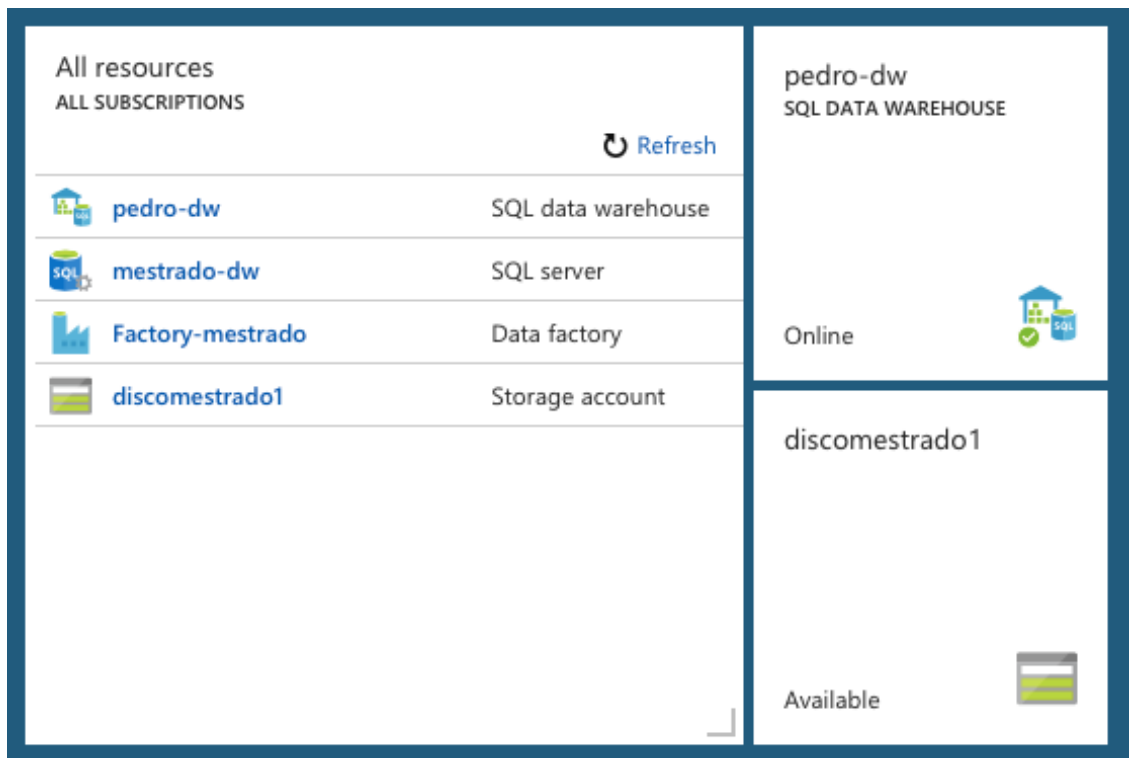


Figure 21 – Resources resume

The setup of this resources took at least 35 minutes to be ready to use.

The connection to the database can be done using SQL Server Management Studio (on premises) or the query editor application from Azure Portal. For this test it was used the query editor from Azure Portal. It is necessary to login into the database and the use the query editor code T-SQL or write SQL queries, see Figure 22 to look at the query editor interface.

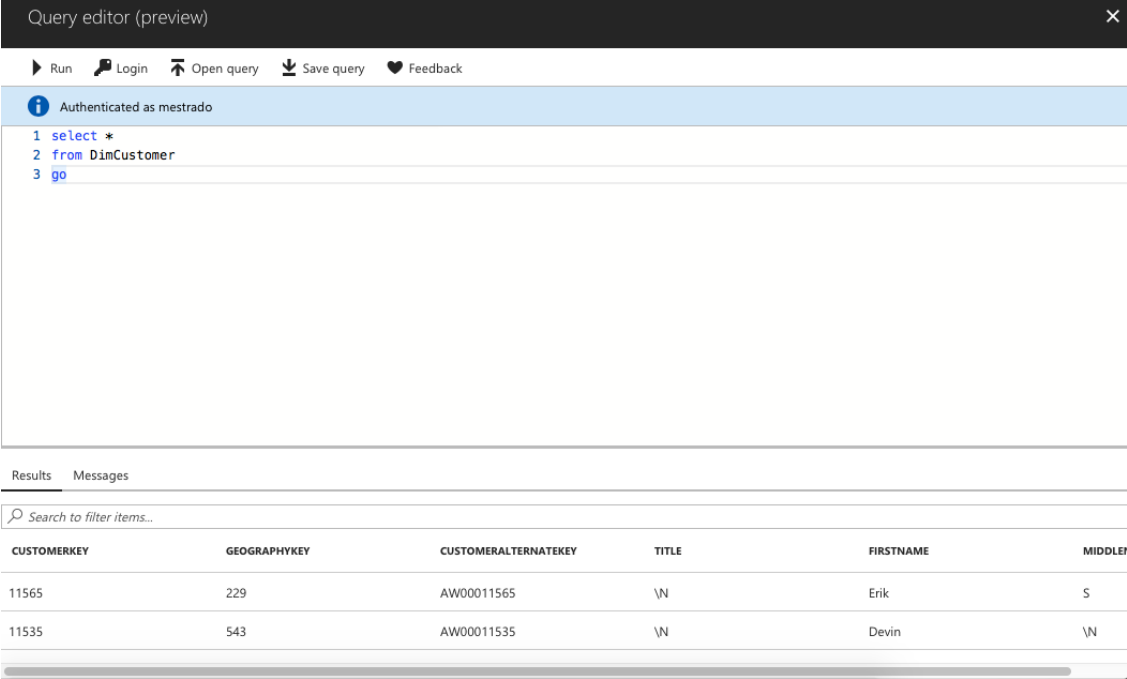


Figure 22 – Azure Portal Query Editor Interface

The database model used in the test was created using the script provided by Microsoft and was opened directly in the query editor to create the tables. The script is in Annex G.

Similar to Amazon, Microsoft has its own online storage service caller Blob Storage. All the files with data to load the created table were uploaded to this created resource using a Python script, see Annex J for details. The results of the upload time are shown in Table 13.

Table 13 — Upload elapsed time to Azure Blob Storage

Filename	Time elapsed (ss.ms)
DimAccount.txt	0.313650
DimCurrency.txt	0.225940
DimCustomer.txt	3.739673
DimDate.txt	0.514568
DimDepartmentGroup.txt	0.234943
DimEmployee.txt	0.496598
DimGeography.txt	0.424199
DimOrganization.txt	0.244209

DimProduct.txt	0.779377
DimProductCategory.txt	0.236475
DimProductSubcategory.txt	0.227863
DimPromotion.txt	0.232548
DimReseller.txt	0.507333
DimSalesReason.txt	0.242104
DimSalesTerritory.txt	0.223790
DimScenario.txt	0.227170
FactCallCenter.txt	0.271814
FactCurrencyRate.txt	1.730939
FactFinance.txt	2.064854
FactInternetSales.txt	2.526516
FactInternetSalesReason.txt	0.865134
FactResellerSales.txt	5.550673
FactSalesQuota.txt	0.270135
FactSurveyResponse.txt	0.487803
ProspectiveBuyer.txt	0.689207
Total	23.327515

At time of this write there is no functionality to get the elapsed time of the insert process of the data into a database table. So the evaluation was done using Data Factory wizard for each of the files that need to be stored in database tables and the elapsed time was consulted in data factory activity window and in the detailed information of each load process, see Figure 23.

Pipeline	Activity	Window...	Window...	Status	Type	Last Att...	Last Att...	Duration	Retry At...
DWLoad...	Activity-...	10/05/2...	10/06/2...	Ready	Copy	10/08/2...	10/08/2...	00:01:05	1
CopyPip...	Activity-...	10/04/2...	10/05/2...	Failed	Copy	10/07/2...	10/07/2...	00:01:05	3
DWLoad...	Activity-...	10/04/2...	10/05/2...	Ready	Copy	10/07/2...	10/07/2...	00:01:05	1
DWLoad...	Activity-...	10/04/2...	10/05/2...	Ready	Copy	10/07/2...	10/07/2...	00:01:05	1
DWLoad...	Activity-...	10/04/2...	10/05/2...	Failed	Copy	10/07/2...	10/07/2...	00:01:05	3
DWLoad...	Activity-...	10/04/2...	10/05/2...	Ready	Copy	10/07/2...	10/07/2...	00:01:05	1
DWLoad...	Activity-...	10/04/2...	10/05/2...	Failed	Copy	10/08/2...	10/08/2...	00:02:38	3

Figure 23 – Azure Data Factory Activity Window

It is important to say that none of the loaded tables were loaded without a retry. All the files were correctly formatted but Azure was not able to process the files without at least one retry.

The results of the load process are present in the next table, see Table 14.

Table 14 — Elapsed time of Loading data into Azure SQL DW table

Table Name	Elapsed (sec)
DimAccount	9
DimCurrency	4
DimCustomer	65
DimDate	13
DimDepartmentGroup	2
DimEmployee	11
DimGeography	4
DimOrganization	3
DimProductCategory	11
DimProductSubcategory	6
DimPromotion	8
DimProduct	21
DimReseller	10
DimSalesReason	2
DimSalesTerritory	2
DimScenario	1
FactCallCenter	9
FactCurrencyRate	14
FactFinance	22
FactInternetSales	47
FactInternetSalesReason	13
FactResellerSales	62
FactSalesQuota	6
FactSurveyResponse	6
ProspectiveBuyer	2
Total	353

5.1.4 Redshift and Azure SQL Data Warehouse Unit Test

For testing purposes, the versions of RedShift and Azure SQL Data Warehouse were free trials with some restrictions in time, available disk space, performance and

support. Due to available disk space, the model was populated with no representative data for data warehouse purpose.

In this unit test, a file with 900 MB of data was uploaded to each of the online storage and then all the data was inserted in a table.

The results of the test are in Table 15.

Table 15 — Elapsed time of Unit Test

Filename	Amazon S3 / Redshift		Azure Blob / SQL Data Warehouse	
	Upload elapsed time (hh:mm:ss.ms)	Table loading (sec)	Upload elapsed time (hh:mm:ss.ms)	Table loading (sec)
FactResellerSales_GB.txt	0:01:49.268183	39	0:06:02.862765	65

In the next two figures are presented the total run time of the table loading process of RedShift and Azure SQL databases, see Figure 24 and Figure 25.

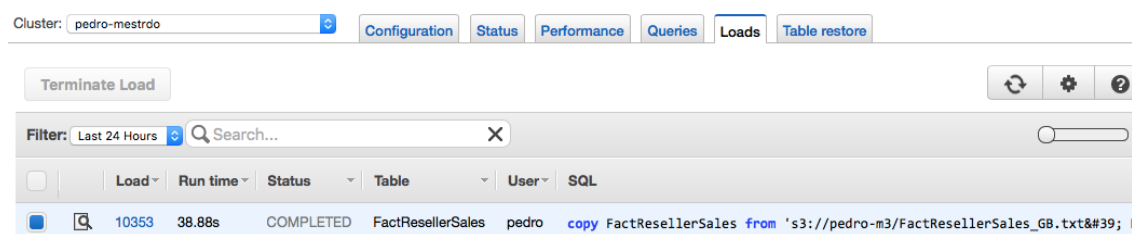


Figure 24 – Redshift loading table

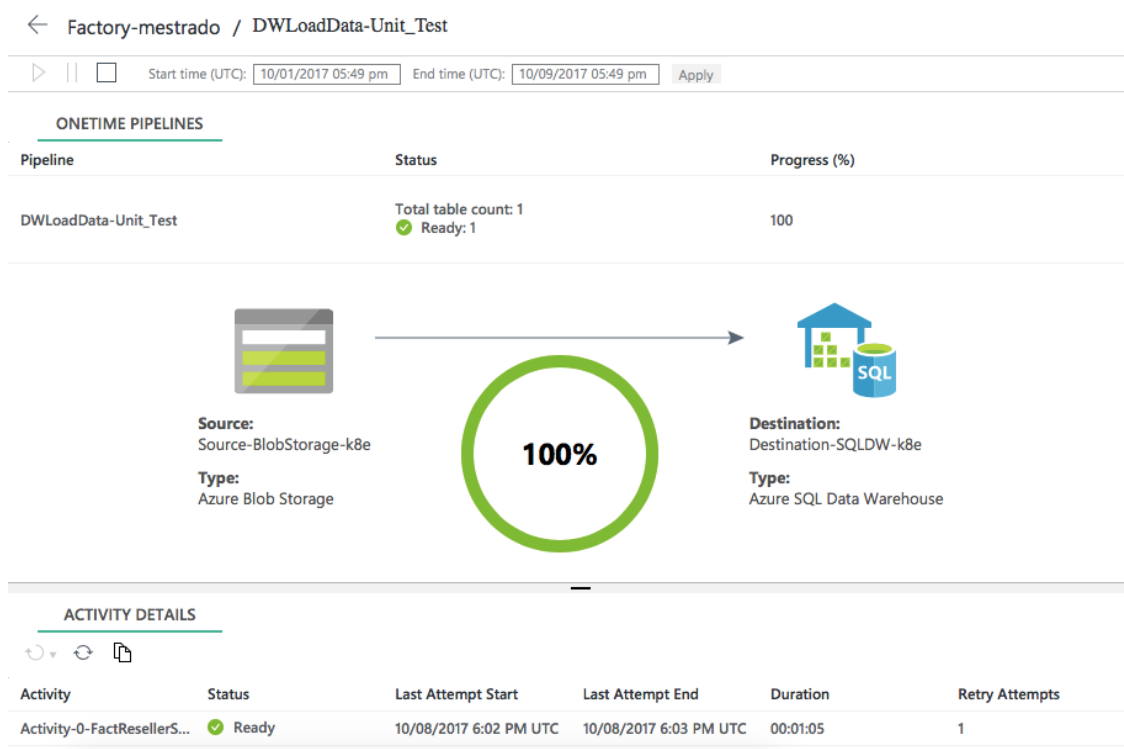


Figure 25 – SQL Data Warehouse loading table

5.1.5 Redshift and Azure SQL Data Warehouse side by side

The information presented in this section is based on the collection of the test results, see Table 16.

Table 16 — Redshift and Azure SQL Data Warehouse side by side

Filename/Table	Amazon S3 / Redshift		Azure Blob / SQL Data Warehouse	
	Upload elapsed time (mm:ss.ms)	Table loading (sec)	Upload elapsed time (mm:ss.ms)	Table loading (sec)
FactResellerSales_GB.txt	01:49.268183	38.88	06:02.862765	65
DimAccount.txt	0.331897	2.363943	0.313650	9
DimCurrency.txt	0.293321	2.01517	0.225940	4
DimCustomer.txt	1.816410	3.698124	3.739673	65
DimDate.txt	0.348714	2.717312	0.514568	13
DimDepartmentGroup.txt	0.293161	2.342524	0.234943	2
DimEmployee.txt	0.331219	3.704916	0.496598	11
DimGeography.txt	0.289604	2.423666	0.424199	4
DimOrganization.txt	0.285157	2.343605	0.244209	3
DimProductCategory.txt	0.388092	2.184846	0.779377	11
DimProductSubcategory.txt	0.289657	2.154736	0.236475	6
DimPromotion.txt	0.340574	2.741274	0.227863	8
DimProduct.txt	0.279039	4.547919	0.232548	21
DimReseller.txt	0.334619	2.932498	0.507333	10
DimSalesReason.txt	0.290870	2.130361	0.242104	2
DimSalesTerritory.txt	0.292220	2.362504	0.223790	2
DimScenario.txt	0.288650	2.290233	0.227170	1
FactCallCenter.txt	0.287555	2.335293	0.271814	9
FactCurrencyRate.txt	0.389258	2.053151	1.730939	14
FactFinance.txt	0.601055	2.250114	2.064854	22
FactInternetSales.txt	1.395725	3.099893	2.526516	47
FactInternetSalesReason.txt	0.379334	2.197098	0.865134	13
FactResellerSales.txt	4.421550	3.436375	5.550673	62
FactSalesQuota.txt	0.349143	2.060158	0.270135	6
FactSurveyResponse.txt	0.329648	2.223133	0.487803	6
ProspectiveBuyer.txt	0.382752	3.431617	0.689207	2

5.1.6 Redshift and Azure SQL Data Warehouse load test

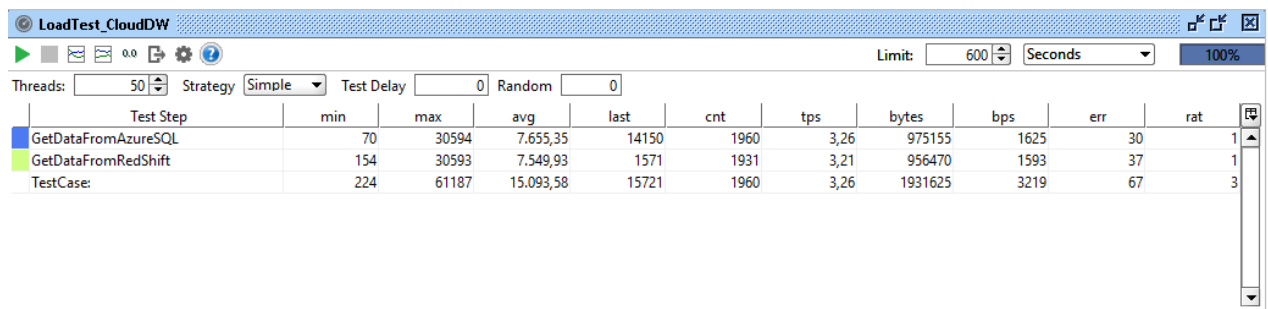
The second test scenario is to perform a load test. The load test was based on the consumption of a web service with two methods, one to call RedShift and another to call Azure SQL.

The web service was coded in C# and installed on a web server. The source code is available on Annex K. The queries to the database were the same for the two methods.

To perform the load test was used a tool called SOAPUI and the characteristics of the test are:

- Maximum of 50 simultaneous requests;
- No delay between requests;
- 10 minutes of duration.

The results of the test are presented in Figure 26.



The screenshot shows the SOAPUI interface for a load test named 'LoadTest_CloudDW'. The test configuration includes 50 threads, a simple strategy, a 600-second limit, and 100% concurrency. The results table shows three test steps: 'GetDataFromAzureSQL', 'GetDataFromRedShift', and 'TestCase:'. The 'TestStep' column is highlighted in blue for the first two rows. The 'min', 'max', 'avg', 'last', 'cnt', 'tps', 'bytes', 'bps', 'err', and 'rat' columns provide performance metrics for each step.

Test Step	min	max	avg	last	cnt	tps	bytes	bps	err	rat
GetDataFromAzureSQL	70	30594	7.655,35	14150	1960	3,26	975155	1625	30	1
GetDataFromRedShift	154	30593	7.549,93	1571	1931	3,21	956470	1593	37	1
TestCase:	224	61187	15.093,58	15721	1960	3,26	1931625	3219	67	3

Figure 26 – Load Test results

The result of the test concludes that the difference between the two databases in terms of performance in this test are not significant. One relevant thing is that in both cases there were some connection errors (rejected connection errors) on the execution thrown by the database engine.

Compared to the same queries, see Annex L, executed on the on-premises database, the results on the cloud are 35% higher.

5.2 Recommendation

With the results of the tests done and with the experience of all processes needed to accomplish the objective of having a data warehouse in the cloud, the recommended solution is Amazon RedShift despite of all integration and familiarity with products of Microsoft Azure because:

- Ease of use;
- Ease of configure;
- Faster on the availability of resources;

- More robust than Microsoft Azure in terms of cloud services and with larger experience;
- Comparing the basic and free support, Amazon is very faster to respond to a request. Compared to Microsoft, Amazon answered the request in the same day and that did not happen with Microsoft which had a 2 day delay on the response;
- The upload of files is easier in both solutions but quicker in Amazon S3;
- The load process of data into databases tables is more clear in RedShift than in Azure SQL Data Warehouse.

6 Conclusions and future work

Data warehouses are defined as customized data storage that aggregate data from multiple sources and store it in a common location to be able to run reports and queries over it. Many companies use data warehouses to compile regular financial reports or business metric analyses.

Cloud data warehousing is the convergence of three trends – huge changes in data sources, volume and complexity; the need for data access and analytics; and better technology that increased the efficiency of data access, analytics and storage. Traditional data warehouse systems were not designed to handle the volume, variety and complexity of today's data.

The integration of a Cloud DW solution needs a very well defined strategy that would involve Cloud Computing capabilities.

The success of the implementation depends on the existence of a service-oriented strategy at the organization level, which would provide the necessary infrastructure for the Cloud implementation.

There are several challenges when deploying data warehouses into the cloud:

- Importing data for the data warehouse into the cloud for storage can be a challenge, because when using the cloud, a customer is dependent on the Internet connection and the infrastructure of the cloud provider. It can be necessary to use a dedicated communication line to mitigate the connection problems but it as a cost;

- Getting large amounts of data from cloud storage to compute nodes provided by the cloud solution for computing can lead to a performance issue;
- Loss of control can lead to issues involving security and trust.

In this project were evaluated some cloud data warehouse solutions in order to recommend what is the best solution according to the defined criteria.

With this work was proved that the two best solutions for data warehousing in the cloud are also the cloud leaders defined by Gartner. The solutions are Amazon RedShift and Microsoft Azure SQL Data Warehouse. Comparative tests were done in terms of database response to queries and data loading and also to the time to transfer data up into the cloud storage. With the results obtained on the tests was proved that the time to obtain data from the database is higher compared to the same data obtained in an on-premises database and the transfer period of time can be a problem when we have a huge amount of data.

Considering the results of the tests, the recommendation is to choose Amazon Redshift cloud database solution for data warehousing. It is only a recommendation based on the defined criteria in this project scope.

As time goes by, cloud services are becoming more and more robust and reliable which means that a data warehouse in the cloud can be a solution to take into account for organizations that are evolving its on-premises DW or starting a new one.

As future work, tests with real data need to be done in order to prove our recommendation, some data with dimension rounding TB need to be used and also the usage of real business needs to query the cloud database. Some benchmarks have to be taken.

Further research in combining data warehousing and cloud computing is needed and more tools for exploring a database in the cloud are also needed.

References

- 1 Data Warehousing Concepts [WWW Document], n.d. URL
https://docs.oracle.com/cd/B12037_01/server.101/b10736/concept.htm
(accessed 2.24.17).
- About the Greenplum Architecture | Pivotal Greenplum Database Docs [WWW Document], n.d. URL
http://gpdb.docs.pivotal.io/4350/admin_guide/intro/arch_overview.html
(accessed 2.16.17).
- Above the Clouds: A Berkeley View of Cloud Computing | EECS at UC Berkeley [WWW Document], n.d. URL
<https://www2.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-28.html>
(accessed 2.24.17).
- AdventureWorks Sample Data Warehouse [WWW Document], n.d. URL
[https://technet.microsoft.com/en-us/library/ms124623\(v=sql.100\).aspx](https://technet.microsoft.com/en-us/library/ms124623(v=sql.100).aspx)
(accessed 2.24.17).
- Armbrust, M., Fox, A., Griffith, R., Joseph, A.D., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A., Stoica, I., Zaharia, M., 2010. A View of Cloud Computing. *Commun ACM* 53, 50–58. doi:10.1145/1721654.1721672
- barbkess, 2017. PolyBase, what is [WWW Document]. URL
<https://docs.microsoft.com/en-us/sql/relational-databases/polybase/polybase-guide> (accessed 10.19.17).
- Cloud computing complicates customer-vendor relationships [WWW Document], n.d. URL <http://searchcloudcomputing.techtarget.com/podcast/Cloud-computing-complicates-customer-vendor-relationships> (accessed 2.24.17).
- Cloud database - Wikipedia [WWW Document], n.d. URL
https://en.wikipedia.org/wiki/Cloud_database (accessed 2.23.17).
- Cloud DW solutions Comparison [WWW Document], 2017. URL <https://db-engines.com/en/system/Amazon+Redshift%3BGreenplum%3BMicrosoft+Azure+SQL+Database%3BSnowflake> (accessed 10.16.17).
- COPY - Amazon Redshift [WWW Document], n.d. URL
http://docs.aws.amazon.com/redshift/latest/dg/r_COPY.html (accessed 10.7.17).
- Dageville, B., Cruanes, T., Zukowski, M., Antonov, V., Avanes, A., Bock, J., Claybaugh, J., Engovatov, D., Hentschel, M., Huang, J., Lee, A.W., Motivala, A., Munir, A.Q., Pelley, S., Povinec, P., Rahn, G., Triantafyllis, S., Unterbrunner, P., 2016. The Snowflake Elastic Data Warehouse, in: *Proceedings of the 2016 International Conference on Management of Data, SIGMOD '16*. ACM, New York, NY, USA, pp. 215–226. doi:10.1145/2882903.2903741
- Data Warehouse System Architecture - Amazon Redshift [WWW Document], n.d. URL
https://docs.aws.amazon.com/redshift/latest/dg/c_high_level_system_architecture.html (accessed 1.1.17).
- Data Warehousing Implementations: A Review [WWW Document], n.d. URL
<http://studylib.net/doc/8838941/data-warehousing-implementations--a-review>
(accessed 10.16.17).

- Gartner, 2016a. Magic Quadrant for Data Warehouse and Database Management Solutions for Analytics [WWW Document]. Magic Quadr. Data Wareh. Database Manag. Solut. Anal. URL <https://www.gartner.com/doc/reprints?id=1-2ZFVZ5B&ct=160225&st=sb> (accessed 1.2.17).
- Gartner, 2016b. Magic Quadrant for Cloud Infrastructure as a Service, Worldwide [WWW Document]. Magic Quadr. Cloud Infrastruct. Serv. Worldw. URL <https://www.gartner.com/doc/reprints?id=1-2G2O5FC&ct=150519> (accessed 12.21.16).
- Goutas, L., Sutanto, J., Aldarbesti, H., 2016. The Building Blocks of a Cloud Strategy: Evidence from Three SaaS Providers. *Commun. ACM* 59, 90–97. doi:10.1145/2756545
- Guermazi, E., Ben-Abdallah, H., Ayed, M.B., 2015. Modeling a secure cloud data warehouse with SoaML, in: 2015 11th International Conference on Information Assurance and Security (IAS). Presented at the 2015 11th International Conference on Information Assurance and Security (IAS), pp. 55–60. doi:10.1109/ISIAS.2015.7492745
- Gupta, A., Agarwal, D., Tan, D., Kulesza, J., Pathak, R., Stefani, S., Srinivasan, V., 2015. Amazon Redshift and the Case for Simpler Data Warehouses, in: Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data, SIGMOD '15. ACM, New York, NY, USA, pp. 1917–1923. doi:10.1145/2723372.2742795
- Inmon vs. Kimball: Which approach is suitable for your data warehouse? [WWW Document], n.d. URL <http://www.computerweekly.com/tip/Inmon-vs-Kimball-Which-approach-is-suitable-for-your-data-warehouse> (accessed 12.28.16).
- Inmon, W.H., 2002. Building the Data Warehouse, 3rd Edition. ed. Wiley.
- Install and Configure the Amazon Redshift ODBC Driver on Microsoft Windows Operating Systems - Amazon Redshift [WWW Document], n.d. URL <http://docs.aws.amazon.com/redshift/latest/mgmt/install-odbc-driver-windows.html> (accessed 10.20.17).
- Key Concepts & Architecture — Snowflake Documentation [WWW Document], n.d. URL <https://docs.snowflake.net/manuals/user-guide/intro-key-concepts.html> (accessed 2.16.17).
- Kimball, R., Ross, M., 2013. The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling, Third Edition. ed. John Wiley & Sons.
- Mell, P., Grance, T., 2011. The NIST definition of Cloud Computing.
- Nirmala, M., Bharathi, R., Srujana, M., 2016. Survey on Cloud Computing and its Security. Presented at the International Journal of Advance Research in Computer Science and Management Studies.
- Sheta, D.O.E., Eldeen, A.N., 2013. The technology of using a data warehouse to support decision-making in health care. *Int. J. Database Manag. Syst.* 5, 75–86. doi:10.5121/ijdms.2013.5305
- Snowflake, 2016. Cloud Data Warehousing For Dummies. Snowflake.
- Snowflake Reinvents the Data Warehouse for the Cloud, 2014. . Snowflake.
- SQL Data Warehouse | Microsoft Azure [WWW Document], n.d. URL <https://azure.microsoft.com/en-us/services/sql-data-warehouse/> (accessed 11.13.16).
- Talia, D., 2013. Clouds for Scalable Big Data Analytics. *Computer* 46, 98–101. doi:10.1109/MC.2013.162

Zhang, Q., Cheng, L., Boutaba, R., 2010. Cloud computing: state-of-the-art and research challenges. *J. Internet Serv. Appl.* 1, 7–18. doi:10.1007/s13174-010-0007-6

Annex A

Table 17 – Currency dimension table

Column Name	Data type	Nullability
CurrencyKey	int	NOT NULL
CurrencyAlternateKey	nchar(3)	NOT NULL
CurrencyName	nvarchar(50)	NOT NULL

Table 18 – Customer dimension table

Column Name	Data type	Nullability
CustomerKey	int	NOT NULL
GeographyKey	int	NULL
CustomerAlternateKey	nvarchar(15)	NOT NULL
Title	nvarchar(8)	NULL
FirstName	nvarchar(50)	NULL
MiddleName	nvarchar(50)	NULL
LastName	nvarchar(50)	NULL
NameStyle	bit	NULL
BirthDate	date	NULL
MaritalStatus	nchar(1)	NULL
Suffix	nvarchar(10)	NULL
Gender	nvarchar(1)	NULL
EmailAddress	nvarchar(50)	NULL
YearlyIncome	money	NULL
TotalChildren	tinyint	NULL
NumberChildrenAtHome	tinyint	NULL
EnglishEducation	nvarchar(40)	NULL
SpanishEducation	nvarchar(40)	NULL
FrenchEducation	nvarchar(40)	NULL
EnglishOccupation	nvarchar(100)	NULL
SpanishOccupation	nvarchar(100)	NULL
FrenchOccupation	nvarchar(100)	NULL
HouseOwnerFlag	nchar(1)	NULL
NumberCarsOwned	tinyint	NULL
AddressLine1	nvarchar(120)	NULL
AddressLine2	nvarchar(120)	NULL
Phone	nvarchar(20)	NULL
DateFirstPurchase	date	NULL
CommuteDistance	nvarchar(15)	NULL

Table 19 – Date dimension table

Column Name	Data type	Nullability
DateKey	int	NOT NULL
FullDateAlternateKey	date	NOT NULL
DayNumberOfWeek	tinyint	NOT NULL
EnglishDayNameOfWeek	nvarchar(10)	NOT NULL

SpanishDayNameOfWeek	nvarchar(10)	NOT NULL
FrenchDayNameOfWeek	nvarchar(10)	NOT NULL
DayNumberOfMonth	tinyint	NOT NULL
DayNumberOfYear	smallint	NOT NULL
WeekNumberOfYear	tinyint	NOT NULL
EnglishMonthName	nvarchar(10)	NOT NULL
SpanishMonthName	nvarchar(10)	NOT NULL
FrenchMonthName	nvarchar(10)	NOT NULL
MonthNumberOfYear	tinyint	NOT NULL
CalendarQuarter	tinyint	NOT NULL
CalendarYear	smallint	NOT NULL
CalendarSemester	tinyint	NOT NULL
FiscalQuarter	tinyint	NOT NULL
FiscalYear	smallint	NOT NULL
FiscalSemester	tinyint	NOT NULL

Table 20 — Department dimension table

Column Name	Data type	Nullability
DepartmentGroupKey	Int	NOT NULL
ParentDepartmentGroupKey	Int	NULL
DepartmentGroupName	nvarchar(50)	NULL

Table 21 — Employee dimension table

Column Name	Data type	Nullability
EmployeeKey	int	NOT NULL
ParentEmployeeKey	int	NULL
EmployeeNationalIDAlternateKey	nvarchar(15)	NULL
ParentEmployeeNationalIDAlternateKey	nvarchar(15)	NULL
SalesTerritoryKey	int	NULL
FirstName	nvarchar(50)	NOT NULL
LastName	nvarchar(50)	NOT NULL
MiddleName	nvarchar(50)	NULL
NameStyle	bit	NOT NULL
Title	nvarchar(50)	NULL
HireDate	date	NULL
BirthDate	date	NULL
LoginID	nvarchar(256)	NULL
EmailAddress	nvarchar(50)	NULL
Phone	nvarchar(25)	NULL
MaritalStatus	nchar(1)	NULL
EmergencyContactName	nvarchar(50)	NULL
EmergencyContactPhone	nvarchar(25)	NULL
SalariedFlag	bit	NULL
Gender	nchar(1)	NULL
PayFrequency	tinyint	NULL
BaseRate	money	NULL
VacationHours	smallint	NULL
SickLeaveHours	smallint	NULL
CurrentFlag	bit	NOT NULL

SalesPersonFlag	bit	NOT NULL
DepartmentName	nvarchar(50)	NULL
StartDate	date	NULL
EndDate	date	NULL
Status	nvarchar(50)	NULL

Table 22 — Geography dimension table

Column Name	Data type	Nullability
GeographyKey	int	NOT NULL
City	nvarchar(30)	NULL
StateProvinceCode	nvarchar(3)	NULL
StateProvinceName	nvarchar(50)	NULL
CountryRegionCode	nvarchar(3)	NULL
EnglishCountryRegionName	nvarchar(50)	NULL
SpanishCountryRegionName	nvarchar(50)	NULL
FrenchCountryRegionName	nvarchar(50)	NULL
PostalCode	nvarchar(15)	NULL
SalesTerritoryKey	int	NULL

Table 23 — Organization dimension table

Column Name	Data type	Nullability
OrganizationKey	Int	NOT NULL
ParentOrganizationKey	Int	NULL
PercentageOfOwnership	nvarchar(16)	NULL
OrganizationName	nvarchar(50)	NULL
CurrencyKey	Int	NULL

Table 24 — Product dimension table

Column Name	Data type	Nullability
ProductKey	int	NOT NULL
ProductAlternateKey	nvarchar(25)	NULL
ProductSubcategoryKey	int	NULL
WeightUnitMeasureCode	nchar(3)	NULL
SizeUnitMeasureCode	nchar(3)	NULL
EnglishProductName	nvarchar(50)	NOT NULL
SpanishProductName	nvarchar(50)	NOT NULL
FrenchProductName	nvarchar(50)	NOT NULL
StandardCost	money	NULL
FinishedGoodsFlag	bit	NOT NULL
Color	nvarchar(15)	NOT NULL
SafetyStockLevel	smallint	NULL
ReorderPoint	smallint	NULL
ListPrice	money	NULL
Size	nvarchar(50)	NULL
SizeRange	nvarchar(50)	NULL
Weight	float	NULL
DaysToManufacture	int	NULL

ProductLine	nchar(2)	NULL
DealerPrice	money	NULL
Class	nchar(2)	NULL
Style	nchar(2)	NULL
ModelName	nvarchar(50)	NULL
LargePhoto	varbinary	NULL
EnglishDescription	nvarchar(400)	NULL
FrenchDescription	nvarchar(400)	NULL
ChineseDescription	nvarchar(400)	NULL
ArabicDescription	nvarchar(400)	NULL
HebrewDescription	nvarchar(400)	NULL
ThaiDescription	nvarchar(400)	NULL
GermanDescription	nvarchar(400)	NULL
JapaneseDescription	nvarchar(400)	NULL
TurkishDescription	nvarchar(400)	NULL
StartDate	datetime	NULL
EndDate	datetime	NULL
Status	nvarchar(7)	NULL

Table 25 — Product category dimension table

Column Name	Data type	Nullability
ProductCategoryKey	int	NOT NULL
ProductCategoryAlternateKey	int	NULL
EnglishProductCategoryName	nvarchar(50)	NOT NULL
SpanishProductCategoryName	nvarchar(50)	NOT NULL
FrenchProductCategoryName	nvarchar(50)	NOT NULL

Table 26 — Product subcategory dimension table

Column Name	Data type	Nullability
ProductSubcategoryKey	int	NOT NULL
ProductSubcategoryAlternateKey	int	NULL
EnglishProductSubcategoryName	nvarchar(50)	NOT NULL
SpanishProductSubcategoryName	nvarchar(50)	NOT NULL
FrenchProductSubcategoryName	nvarchar(50)	NOT NULL
ProductCategoryKey	int	NULL

Table 27 — Promotion dimension table

Column Name	Data type	Nullability
PromotionKey	int	NOT NULL
PromotionAlternateKey	int	NULL
EnglishPromotionName	nvarchar(255)	NULL
SpanishPromotionName	nvarchar(255)	NULL
FrenchPromotionName	nvarchar(255)	NULL
DiscountPct	float	NULL
EnglishPromotionType	nvarchar(50)	NULL
SpanishPromotionType	nvarchar(50)	NULL
FrenchPromotionType	nvarchar(50)	NULL
EnglishPromotionCategory	nvarchar(50)	NULL

SpanishPromotionCategory	nvarchar(50)	NULL
FrenchPromotionCategory	nvarchar(50)	NULL
StartDate	datetime	NOT NULL
EndDate	datetime	NULL
MinQty	int	NULL
MaxQty	int	NULL

Table 28 — Reseller dimension table

Column Name	Data type	Nullability
ResellerKey	int	NOT NULL
GeographyKey	int	NULL
ResellerAlternateKey	nvarchar(15)	NULL
Phone	nvarchar(25)	NULL
BusinessType	varchar(10)	NOT NULL
ResellerName	nvarchar(50)	NOT NULL
NumberEmployees	int	NULL
OrderFrequency	char(1)	NULL
OrderMonth	tinyint	NULL
FirstOrderYear	int	NULL
LastOrderYear	int	NULL
ProductLine	nvarchar(50)	NULL
AddressLine1	nvarchar(60)	NULL
AddressLine2	nvarchar(60)	NULL
AnnualSales	money	NULL
BankName	nvarchar(50)	NULL
MinPaymentType	tinyint	NULL
MinPaymentAmount	money	NULL
AnnualRevenue	money	NULL
YearOpened	int	NULL

Table 29 — Sales reason dimension table

Column Name	Data type	Nullability
SalesReasonKey	int	NOT NULL
SalesReasonAlternateKey	int	NOT NULL
SalesReasonName	nvarchar(50)	NOT NULL
SalesReasonReasonType	nvarchar(50)	NOT NULL

Table 30 — Sales territory dimension table

Column Name	Data type	Nullability
SalesTerritoryKey	int	NOT NULL
SalesTerritoryAlternateKey	int	NULL
SalesTerritoryRegion	nvarchar(50)	NOT NULL
SalesTerritoryCountry	nvarchar(50)	NOT NULL
SalesTerritoryGroup	nvarchar(50)	NULL

Table 31 — Scenario dimension table

Column Name	Data type	Nullability
ScenarioKey	int	NOT NULL

ScenarioName nvarchar(50) NULL

Annex B

Table 32 — Call center fact table

Column Name	Data type	Nullability
FactCallCenterID	int	NOT NULL
DateKey	int	NOT NULL
WageType	nvarchar(15)	NOT NULL
Shift	nvarchar(20)	NOT NULL
LevelOneOperators	smallint	NOT NULL
LevelTwoOperators	smallint	NOT NULL
TotalOperators	smallint	NOT NULL
Calls	int	NOT NULL
AutomaticResponses	int	NOT NULL
Orders	int	NOT NULL
IssuesRaised	smallint	NOT NULL
AverageTimePerIssue	smallint	NOT NULL
ServiceGrade	float	NOT NULL

Table 33 — Currency rate fact table

Column Name	Data type	Nullability
CurrencyKey	int	NOT NULL
DateKey	int	NOT NULL
AverageRate	float	NOT NULL
EndOfDayRate	float	NOT NULL

Table 34 — Finance fact table

Column Name	Data type	Nullability
FinanceKey	int	NOT NULL
DateKey	int	NOT NULL
OrganizationKey	int	NOT NULL
DepartmentGroupKey	int	NOT NULL
ScenarioKey	int	NOT NULL
AccountKey	int	NOT NULL
Amount	float	NOT NULL

Table 35 — Internet sales fact table

Column Name	Data type	Nullability
ProductKey	int	NOT NULL
OrderDateKey	int	NOT NULL
DueDateKey	int	NOT NULL
ShipDateKey	int	NOT NULL
CustomerKey	int	NOT NULL
PromotionKey	int	NOT NULL
CurrencyKey	int	NOT NULL
SalesTerritoryKey	int	NOT NULL
SalesOrderNumber	nvarchar(20)	NOT NULL

SalesOrderLineNumber	tinyint	NOT NULL
RevisionNumber	tinyint	NOT NULL
OrderQuantity	smallint	NOT NULL
UnitPrice	money	NOT NULL
ExtendedAmount	money	NOT NULL
UnitPriceDiscountPct	float	NOT NULL
DiscountAmount	float	NOT NULL
ProductStandardCost	money	NOT NULL
TotalProductCost	money	NOT NULL
SalesAmount	money	NOT NULL
TaxAmt	money	NOT NULL
Freight	money	NOT NULL
CarrierTrackingNumber	nvarchar(25)	NULL
CustomerPONumber	nvarchar(25)	NULL

Table 36 — Internet sales reason fact table

Column Name	Data type	Nullability
SalesOrderNumber	nvarchar(20)	NOT NULL
SalesOrderLineNumber	tinyint	NOT NULL
SalesReasonKey	int	NOT NULL

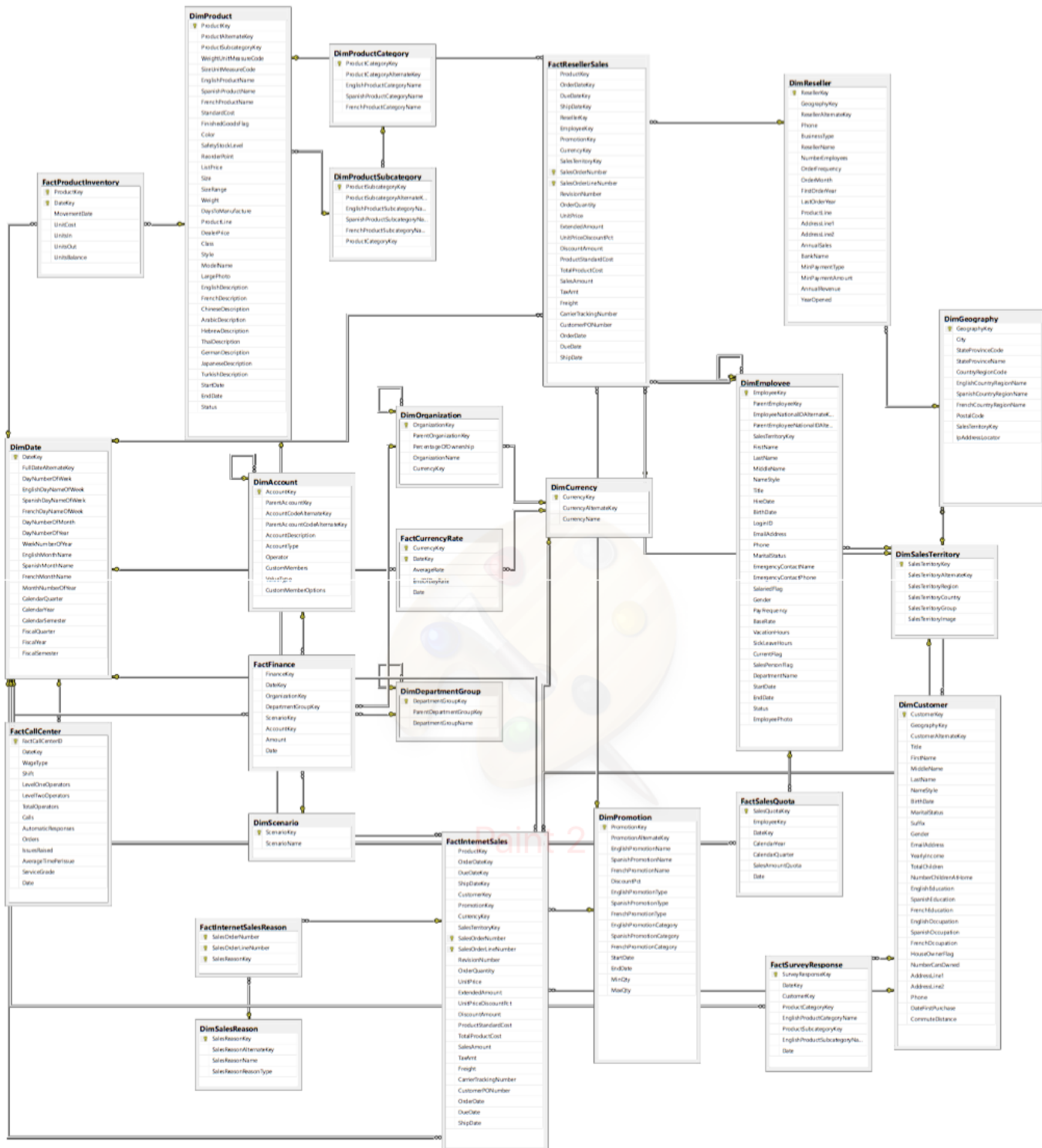
Table 37 — Reseller sales fact table

Column Name	Data type	Nullability
ProductKey	int	NOT NULL
OrderDateKey	int	NOT NULL
DueDateKey	int	NOT NULL
ShipDateKey	int	NOT NULL
ResellerKey	int	NOT NULL
EmployeeKey	int	NOT NULL
PromotionKey	int	NOT NULL
CurrencyKey	int	NOT NULL
SalesTerritoryKey	int	NOT NULL
SalesOrderNumber	nvarchar(20)	NOT NULL
SalesOrderLineNumber	tinyint	NOT NULL
RevisionNumber	tinyint	NULL
OrderQuantity	smallint	NULL
UnitPrice	money	NULL
ExtendedAmount	money	NULL
UnitPriceDiscountPct	float	NULL
DiscountAmount	float	NULL
ProductStandardCost	money	NULL
TotalProductCost	money	NULL
SalesAmount	money	NULL
TaxAmt	money	NULL
Freight	money	NULL
CarrierTrackingNumber	nvarchar(50)	NULL
CustomerPONumber	nvarchar(50)	NULL

Table 38 — Sales quota fact table

Column Name	Data type	Nullability
SalesQuotaKey	int	NOT NULL
EmployeeKey	int	NOT NULL
DateKey	int	NOT NULL
CalendarYear	smallint	NOT NULL
CalendarQuarter	tinyint	NOT NULL
SalesAmountQuota	money	NOT NULL

Annex C



Annex D

Value Analysis

“Value analysis is an examination of the function of parts and materials in an effort to reduce cost and/or improve product performance. The primary objective of value analysis is assess how to increase the value of an item or service at the lowest cost without sacrificing quality.” (Susana Nicola, 2016).

In this chapter is described the design purpose, the value purpose and finally the market opportunities.

Design purpose

The purpose of this project is to study the available cloud data warehouse solutions in the market, analyse them and recommend the best one according to our evaluation criteria.

Value purpose

Currently, organizations are facing some issues regarding its on premise data warehouse system like lack of performance or storage.

A cloud data warehouse brings some benefits to organizations, as example:

- Cost reduction;
- Flexibility;
- On-demand scalability;
- Automatic software upgrade;
- Centralized infrastructure management;
- Availability;
- Reliability.

Other organizations are thinking about an implementation of a cloud data warehouse in order to take advantage of the benefits brought by cloud computing.

Product

The product purposed in the work is the selection of the best data warehousing solution in the cloud.

Potential users

The solution recommended by this project will have as potential customers all the medium and large organizations that currently have an on-premises data warehouse

and already have a service oriented architecture implementation within its company or organizations that are thinking about building a data warehouse.

Business model Canvas

Table 1 — Business model Canvas

			Project: Data Warehouse in the cloud	Date: 25/02/2017
Key Partners - Telecommunication Organizations - Cloud providers	Key Activities - Advisory for implementation of a cloud data warehouse solution	Value Propositions - Cost reduction - Flexibility - On-demand scalability - Automatic software update - Centralized infrastructure management - Availability - Reliability	Customer Relationships - User experience	Customer Segments - Medium organizations - Large organizations
	Key Resources - Information provided by cloud dw solution providers		Channels - Internet	
Cost Structure - Dedicated communication line - Development		Revenue Streams - Time to focus on clients		

Market opportunities

In digital era, cloud computing and big data are the top objectives of the most organizations. Helping some organizations moving to the cloud can be an opportunity to apply the obtained knowledge.

Quiz answers

This section has the answers to all the six questions of the value analysis quiz.

Question 1

“Numa fase inicial de um processo de negócio e de inovação, baseado no modelo de Peter Koen:

- Identifique e explicita, de acordo com o seu tema de projeto, os 5 elementos chave do modelo “the new concept development model” (NCD).
- Identifique métodos/técnicas e/ou ferramentas para analisar cada elemento chave”

The five key elements of the new concept development model (NCD) are:

- Opportunity identification
 - Organizations of medium and large scale that want to evolve its data warehouse to a cloud basis data warehouse or are trying to build one from scratch;
 - These organizations need to focus on the client and let the management of its IT outside their scope;
 - Simplify, speed-up the response to internal clients and reduce the cost of operations.
- Opportunity Analysis
 - It takes place when an organization already has a service oriented architecture and want to move to the cloud;

- Identify the best solution to implement within the organization;
- Questions need to be answered:
 - a) How much would the customer be willing to give up on data outside the organization?;
 - b) What are the regulatory issues?
- iii. Idea Generation and Enrichment
 - Implement the data warehouse in the cloud migrating the on premise warehouse or beginning a complete new implementation;
 - Help in the coexistence of two data warehouses using data virtualization techniques.
- iv. Idea Selection
 - Definition of the financial cost reduction;
 - Definition of the technological risks;
 - Definition of the advantages.
- v. Concept definition
 - Develop a business case based on estimates of cost reduction, investment requirements and level of technology knowledge.
 - Methods or techniques to analyze each element:
 - Opportunity identification
 - i. Brainstorming with organization senior elements.
 - Opportunity Analysis
 - I. What-if analysis.
 - Idea Generation and Enrichment
 - I. Bring internal customers to a brainstorm conversation in order to get more ideas.
 - iv. Idea Selection
 - I. Read scientific papers;
 - II. Get insights from others in the market;
 - III. Proof of concept.
- v. Concept definition
 1. Meetings with customers with the objective to build a business case and get the needed information from the organization different areas.

Question 2

“Baseado nos conceitos “value”, “value for the customer” e “perceived value”, e de acordo com o tema da sua tese, qual o valor (benefícios/sacrifícios) para o cliente? Justifique convenientemente a sua resposta enquadrando os vários benefícios /sacrifícios numa perspetiva longitudinal de valor.”

Currently, organizations are facing some issues regarding its on premise data warehouse system like lack of performance or storage.

A cloud data warehouse brings some benefits to organizations, as example:

- Cost reduction;

- Flexibility;
- On-demand scalability;
- Automatic software upgrade;
- Centralized infrastructure management;
- Availability;
- Reliability.

Other organizations are thinking about an implementation of a cloud data warehouse in order to take advantage of the benefits brought by cloud computing.

Question 3

“Enuncie a proposta de valor do seu Produto/Serviço”

The product purposed in the work is the selection of the best data warehousing solution in the cloud. It will have as potential customers all the medium and large organizations that currently have an on-premises data warehouse and already have a service oriented architecture implementation within its company or organizations that are thinking about building a data warehouse.

Question 4

“Apresente o modelo de negócio de Canvas para descrever a sua ideia de negócio”

			Project: Data Warehouse in the cloud	Date: 25/02/2017
Key Partners - Telecommunication Organizations - Cloud providers	Key Activities - Advisory for implementation of a cloud data warehouse solution	Value Propositions - Cost reduction - Flexibility - On-demand scalability - Automatic software update - Centralized infrastructure management - Availability - Reliability	Customer Relationships - User experience	Customer Segments - Medium organizations - Large organizations
	Key Resources - Information provided by cloud dw solution providers		Channels - Internet	
Cost Structure - Dedicated communication line - Development		Revenue Streams - Time to focus on clients		

Question 5

““People naturally network as they work so why not model itself as network” (V.Allee).

Baseado nesta afirmação, de que forma podemos contruir e analisar o valor? Explique de que forma poderia utilizar o modelo de Verna Allee ou a cadeia de valor de Porter para analisar o valor de negócio.”

Primary Activities

Select a cloud data warehousing solution and then develop the necessary procedure to implement the solution in an organization.

Inbound logistics

Data sources data that need to be stored in the data warehouse.

Operations

The necessary transformation processes to format the data to move to the cloud.

Outbound Logistics

A data warehouse in the cloud ready to be explored by business intelligence tools or other analytical tools.

Marketing and sales

Only marketing inside the organization to show the solution and how to work with it. It is also important to show some differences between the past and the new cloud scenario and enhance the capabilities.

Service

Once the cloud data warehouse is implemented, it has to be agile and continue to bring value to the customers more quickly and faster than ever before. To do that, the data warehouse owner need to have the business team on sight and be always one step forward.

Question 6

“De uma forma geral, um problema que envolva a necessidade de optar por uma decisão que envolva critérios e alternativas com graus de importância diferentes ou pesos variáveis para o decisor é necessário o uso de métodos multicritério. A variação desses pesos para cada critério pode ter diferentes motivos, podendo por exemplo, numa análise de valor de negócio depender de valor para cliente, da percepção do cliente, dos processos existentes, ou mesmo de outras opções com carácter subjetivo. Através de um exemplo real ou ilustrativo do seu trabalho, indique de que forma utilizaria o método AHP. Apresente os cálculos necessários à elaboração do método.”

To answer the question “What cloud data warehouse solution should be used?”, here is some criteria based on “Gartner 2016 Magic Quadrant for Data Warehouse and Database Management Solutions for Analytics” and “Gartner 2016 Magic Quadrant for Cloud Infrastructure as a Service, worldwide”:

- Product/Service: “This criterion relates to increasingly divergent market demands — for traditional logical data warehousing, operational data warehousing and context-independent data management for analytics (for definitions, see "Critical Capabilities for Data Warehouse and Data Management Solutions for Analytics"). The largest and most traditional portion of the analytics and data warehouse market is still dominated by the demand to support relational analytical queries over normalized and dimensional models (including simple trend lines through complex dimensional models). Data management for analytics solutions are increasingly expected to include repositories, semantic data access (such as federation/virtualization) and distributed processing in combination — what is referred to in the market as LDWs. All the

traditional demands of the data warehouse remain. Operational data warehouse use cases also exhibit traditional requirements, plus loading of streaming data, real-time data loading and real-time analytics support. Users expect solutions to become self-tuning, to reduce the number of staff required to optimize the data warehouse, especially as mixed workloads increase. Context-independent warehouses (CIWs) do not necessarily support mixed workloads, nor do they require the same level of mission-critical support. CIWs serve more as data discovery support or sandboxes.”(Gartner, 2016).

“Service providers were evaluated on the capabilities of their cloud IaaS offering to support all use cases being evaluated. We evaluated the breadth and depth of the feature set, self-service capabilities, automated system management and suitability to run a broad range of workload types. This criterion is important to buyers who want to purchase the most capable, feature-rich service.”(Gartner, 2016b).

- Overall Viability: “Providers were evaluated on the success of their cloud IaaS business, as demonstrated by current revenue and revenue growth since the launch of their service; their financial wherewithal to continue investing in the business and to execute successfully on their roadmaps; commitment to their current offerings, with no plans to execute disruptive platform transitions or migrations in the next two years; and their organizational commitment to this business, and its importance to the company's overall strategy. This criterion is important to buyers who prefer to purchase services from large vendors with ample financial resources, or from vendors that have a position of market leadership and are continuing to invest aggressively in the business. It is also important to buyers who are concerned about their long-term strategic investment in a particular vendor, or who want to avoid potentially disruptive service changes.” (Gartner, 2016).

- Customer experience: “Our assessment for this criterion was based on surveys of reference customers and discussions with users of Gartner's inquiry service during the previous six quarters. Also considered are a vendor's track record on proofs of concept, customers' perceptions of its product(s), and customers' loyalty to the vendor (this reflects their tolerance of its practices and can indicate their level of satisfaction). This criterion is sensitive to year-over-year fluctuations, based on customer experience surveys. Additionally, customer input regarding the application of products to limited use cases can be significant, depending on the success or failure of a vendor's approach in the market.” (Gartner, 2016).

- Sales Execution/Pricing: “This criterion examines the price/performance and pricing models of the DBMS, and the ability of the vendor's sales force to manage accounts (judged by the feedback from our clients and feedback collected from a survey of reference customers). It also considers the market share of DBMS software. Also included is the diversity and innovative nature of packaging and pricing models, including the ability to promote, sell and support the product within target markets and around the world. Aspects such as vertical-market sales teams and specific vertical-market data models are considered.” (Gartner, 2016).

- Operations: “This criterion relates to the alignment of a vendor's operations, and how this enhances its ability to deliver. Aspects considered include field delivery of appliances, manufacturing (including identification of diverse geographic cost

advantages), internationalization of product(s) (in light of both technical and legal requirements) and adequate staffing. This criterion also considers a vendor's ability to support clients throughout the world, around the clock, and in many languages. Anticipation of regional and global economic conditions is also evaluated.” (Gartner, 2016).

- Innovation: “Vendors demonstrate innovation by developing new functionality, allocating funds to R&D and leading the market in new directions. This criterion also covers a vendor's ability to innovate and develop new functionality for accomplishing data management for analytics. Also addressed is the maturation of alternative delivery methods, such as infrastructure as a service (IaaS) and cloud infrastructures, as well as solutions for hybrid on-premises-and-cloud and cloud-to-cloud data management support. A vendor's awareness of new methodologies and delivery trends is also considered. Organizations are increasingly demanding data storage strategies that balance cost with performance optimization, so solutions that address the aging and temperature of data will become increasingly important.” (Gartner, 2016).

- Offering (Product) Strategy: “When viewed from a vision perspective, this criterion is clearly distinguished from product execution. We evaluate the roadmap for enhancing traditional data warehouse capabilities (including plans to address currently missing execution components). Also considered are expected functionality and any timetable for meeting new market demands. We looked for, among other things, roadmaps and development plans for the following: a semantic design tier; system and solution auditing and health management to ensure use case SLA compliance; static and dynamic cost-based optimization, with the potential to span processing environments and data structures; management and orchestration of multiple processing engines; and elastic workload management and process distribution. Our assessment also bore in mind that end-user organizations are taking a "best fit" engineering approach that requires vendors to allow their technology to be easily combined with that of other vendors.” (Gartner, 2016).

Annex E

In this appendix is presented the publication, Data Warehousing in the Cloud: Amazon Redshift vs Microsoft Azure SQL, which demonstrates the work developed in this thesis. This article was submitted and accepted for publication at the KDIR and will be presented at the 9th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management, held in Funchal in November 2017.

Data Warehousing in the Cloud: Amazon Redshift vs Microsoft Azure SQL

Pedro Joel Ferreira¹, Ana Almeida¹ and Jorge Bernardino²

¹ Instituto Superior de Engenharia do Porto, Rua Bernardino de Almeida, Porto, Portugal

² Instituto Superior de Engenharia de Coimbra, Rua Pedro Nunes, Coimbra, Portugal
1030537@isep.ipp.pt, amn@isep.ipp.pt, jorge@isec.pt

Keywords: Cloud computing, Data warehousing, Cloud data warehousing.

Abstract: A data warehouse enables the analysis of large amounts of information that typically comes from the organization's transactional systems (OLTP). However, today's data warehouse systems do not have the capacity to handle the massive amount of data that is currently produced, then comes the concept of cloud computing. Cloud computing is a model that enables ubiquitous and on-demand access to a set of shared or non-shared computing resources (such as networks, servers, or storage) that can be quickly provisioned or released only with a simple request and without human intervention. In this model, the features are almost unlimited and in working together they bring a very high computing power that can and should be used for the most varied purposes. From the combination of both these concepts, emerges the cloud data warehouse. It advances the way traditional data warehouse systems are defined by allowing their sources to be located anywhere as long as it is accessible through the Internet, also taking advantage of the great computational power of an infrastructure in the cloud. In this paper, we study two of the most popular cloud data warehousing market solutions: Amazon Redshift and Microsoft Azure SQL Data Warehouse.

1 INTRODUCTION

Data warehouses (DW) are defined as customized data storage that aggregate data from multiple sources and store it in a common location to be able to run reports and queries over it. This concept arose from the requisite to integrate corporate data across multiple application servers that an organization might have, so that it would be possible to make data accessible to all users who need to consume information and make decisions based on it. Many companies use data warehouses to compile regular financial reports or business metric analyses.

Integration of a Cloud Data Warehouse solution demands a very well defined strategy that would involve Cloud Computing capabilities. The success of the implementation depends on the existence of a service-oriented strategy at the organization level, which would provide the necessary infrastructure for the Cloud implementation. Without SOA and BPM, integration of a Data Warehousing solution based on Cloud Computing serves no financial purpose, involving high costs for the present systems reengineering. Also, in order to be successful, Cloud

strategy has to be led according to the business strategy of the organization.

One of the biggest challenges with data warehousing in the Cloud is how the data is transferred up into it. Pumping gigabytes, terabytes, or even petabytes of data up into Cloud over the public Internet can not only come with security concerns, but also performance challenges.

When selecting a Data Warehousing solution, we have to take into account the newest trends on the DW and Cloud Computing market, the present and future needs and the opportunity of integration. Therefore, in order to be successful, the selection of a Cloud DW solution has to be achieved objectively based on good criteria that have been analyzed and weighted according to the present and future needs of the organization. Cloud Data Warehousing is a potential cost saver for big companies, and removing a cost barrier that have held data warehousing back from small and mid-sized businesses.

Cloud Data Warehouse must be designed to take away the undifferentiated heavy lifting of running infrastructure at heavy scale, allowing the customers to focus on their core competencies – its business.

The growing interest in cloud-based data warehousing is driven by the high return on investment. Nonetheless, the adoption of cloud computing for data warehousing faces security challenges given the proprietary nature of the enclosed data. Moving to the cloud is a hard decision not only because the data owners like to have their data near to them (on-premises) but also due to security and data confidentiality issues, because of this last two the decision makers tend to delay the decision of moving to the cloud.

There are many benefits to implement a DW in the cloud, such as cost, efficiency, elasticity, flexibility in the choice of provider, more competitiveness and less time involved in installation and maintenance. It is assumed that cloud system efficiency is even more efficient with the use of parallel computing and the cloud is motivation for small and medium enterprises by the possibility of expansion at an affordable cost.

In this paper we analyze the characteristics of two of most popular cloud data warehousing platforms: Amazon Redshift and Microsoft Azure SQL Data Warehouse. The remainder of this paper is structured as follows. Section 2 describes the related work. Section 3 presents the cloud computing and Section 4 presents a summary of cloud data warehousing area. Section 5 presents two cloud data warehousing market solutions: Amazon Redshift and Microsoft Azure SQL Data Warehouse. Section 6 presents a brief comparison of the cloud data warehouse solutions. Finally, conclusions and future work are summarized in Section 7.

2 RELATED WORK

Multiple research works have been done to compare and evaluate existing Big Data platforms. However, most of the research focus only on a specific capability, technology or purpose.

Almeida and Bernardino (2015a; 2015b) focus on the capability of mining data, and in a mix of technical parameters and features that are suitable for Small and Medium Enterprise environments. On the other hand, Morshed et al. (2016) focused their work on platforms addressing distributed real-time data analytics, and concluded that the platforms present on their research do not cover all the features that are required for distributed computation in real-time.

Kaur et al. (2012) describe some solutions from multiple vendors in the cloud to support a Data Warehouse. They state that each service provider

implements different levels of use, some provide ETL mechanisms or Business Intelligence (BI) solutions while others provide the storage infrastructure and the client then chooses the supplier for their needs and the services required. In the paper it is considered that the market is still immature and services vary in price and performance.

Hemlata Verna (2013) is concerned about how to manage the data through recycle, reuse, reduce and recover information in cloud environment. Popeangă (2014) concentrate her work on studying what architecture is suited for a data warehouse in the cloud.

Mathur et al., (2011), focused their work on distributed databases in the cloud. They propose IaaS cloud servers that can be used to store these databases at low initial cost.

Therefore, there are few related works which evaluate solutions based on specific capabilities, technology or purpose. Our work will be of a broader scope in functionalities and applications in order to be used by SMEs.

3 CLOUD COMPUTING

There is no definitive definition of cloud computing, but Kimball and Ross (2013) propose that "Cloud computing is a self-provisioning and on-demand delivery of computing resources and applications over the Internet with a pay-as-you-go pricing model".

Cloud, by definition, is a self-service system that allows end users to setup applications and services in a cloud computing environment without the intervention of an IT service provider (Armbrust et al., 2010). Cloud computing addresses large amounts of computing power by aggregating computing resources and offering a single view of the system.

Cloud brought new features and possibilities to improve its user's life but also originated a new market segment for IT with new business opportunities. Many organizations build their business around the cloud not only to use its services but also to offer business solutions in this environment. Like any other service, we can identify at least two actors that are directly connected with it:

- User or consumer: is the one who uses the functionalities or resources provided by the cloud. It is the entity or organization that uses the cloud computing services, whether they are related with software, platform or infrastructure;

- **Provider:** is the entity or organization responsible for making cloud services available to consumers. The provider is also responsible for managing the necessary infrastructure which supports its services.

There are different cloud implementation models and they are the different means by which consumers access cloud services. These models are different and are characterized by the target audience they serve. If this assumption is taken into account, there are only two models: private clouds and public clouds.

Private clouds are typically used in the distribution of services in an internal organization environment. The infrastructure that supports the private cloud can be delegated to a service provided and stays private at all, it has not to be acquired and managed by the private cloud users themselves.

When managed internally by users themselves, they have full control over the processes, data or applications used in the cloud. However, they lose some of the general principals or benefits of cloud computing, such as access to infrastructure at reduced prices, elasticity, resources availability or rapid deployment times.

Unlike private clouds, public clouds are used for the distribution of services to the general public, typically via the Internet. The service provider is responsible for the entire support infrastructure. This infrastructure is fully shared by the various users of this cloud. Thanks to this sharing and optimization of resource management processes, providers are able to maximize their use and enable them to be supplied at reduced prices to consumers.

4 CLOUD DATA WAREHOUSING

Nowadays, companies have a greater collection of data than ever before. This includes a huge variety of sources, including cloud-based applications or even company datamarts. In order to make good decisions, get insights and achieve a competitive advantage, companies need to have their data properly analyzed, on time.

The conventional data warehouse architecture is widespread in a large number of companies working with massive and diverse data sets but is a very closed and complex model to respond with the agility that companies currently need (Tereso and Bernardino, 2011). People who make analysis need to wait a number of hours or even days for data to flow into the data warehouse before it becomes available for analysis. In most cases, the storage and

compute resources required to process that data are insufficient (or the same) and this leads to hanging or crashing systems (Goutas et al., 2016).

One of the major concerns on moving to the cloud is the time to “live” with both on-premises and cloud data warehouse system because it is not a good idea to move at once the whole data warehouse. To ease this concern, a data virtualization solution can be used to help out the migration and coexistence of the both data warehouse systems while migration to cloud is ongoing.

Cloud data warehousing was born from the convergence of three trends – huge changes in data sources, volume and complexity; the need for data access and analytics; and better technology that increased the efficiency of data access, analytics and storage. Traditional data warehouse systems were not designed to handle the volume, variety and complexity of today’s data (Almeida et al., 2008).

A data warehouse in the cloud is a database which information is consumed over the Internet, a typical database as a service (DBaaS). Cloud data warehousing is a cost-effective way for organizations to use and take advantage of high technology without high upfront costs to purchase, install and configure the required hardware, software and infrastructure (Talia, 2013).

In the next section we will analyse two of the most popular cloud data warehousing market solutions.

5 CLOUD DATA WAREHOUSING MARKET SOLUTIONS

In this section, the architecture of two cloud data warehousing solutions is described.

The following sections describe the characteristics of most popular platforms: Amazon Redshift and Microsoft Azure SQL Data Warehouse.

5.1 Amazon Redshift

Gartner reports Amazon Web Services (AWS) is often considered the leading cloud data warehouse platform-as-a-service provider (Gartner, 2016a).

Recognized by Gartner as a leader, Amazon Redshift is a fast, fully managed, petabyte-scale data warehouse that makes simple and cost-effective to analyze all our data using existing business intelligence tools.

Amazon Redshift engine is a SQL-compliant, MPP, query processing and database management system designed to support analytics workload. The storage and compute is distributed across on or more compute nodes (Gupta et al., 2015).

The core infrastructure of Amazon Redshift data warehouse is a cluster and it is composed by:

- A leader node;
- One or more compute nodes.

The leader node accepts connections from the client applications and dispatch the work to the compute node: it parses and develops execution plans to carry out database operations and based on the execution plan it compiles code, distributes the compiled code to the compute nodes and assigns a portion of the data to each node (see Figure 1).

The leader node distributes SQL statements to the compute nodes only when a query references tables that are stored on the compute nodes, otherwise they run exclusively on the leader node (“Data Warehouse System Architecture - Amazon Redshift,” n.d.).

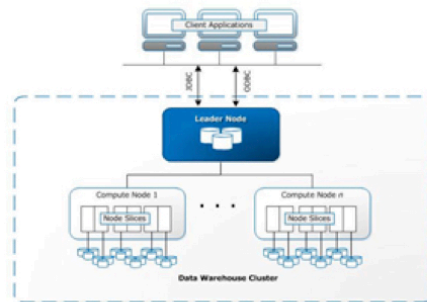


Figure 1 – Amazon Redshift system architecture (source: aws.amazon.com/redshift)

The compute nodes execute the compiled code sent by the leader and send the results back for final aggregation. Each compute node has its own dedicated CPU, memory and storage – it is easy to scale the cluster by upgrading the compute nodes or adding new ones. The minimum storage for each compute is 160GB and scale up to 16TB to support a petabyte of data or more. The compute node is partitioned into slices and each of it is allocated a portion of the node’s memory and disk space, where it processes a portion of the workload assigned to the node – the leader node manages the distribution of data of the workload to the slices and then they work in parallel to complete the operation.

The cluster contains one or more databases. Amazon Redshift is a relational database management system and provides the same functionality as a typical RDBMS including all related with OLTP but it is optimized for high-speed performance analysis and reporting of very large datasets (“Data Warehouse System Architecture - Amazon Redshift,” n.d.).

The database engine is based on PostgreSQL. Another interesting characteristic of Amazon Redshift is that it is a columnar database, which means that each record is not saved as a unique block of data but it is stored in independent columns. The query performance can greatly be improved by selecting a limited subset of columns rather than the full record.

The data warehouse functionality is comparable to the high level databases. The ease of use and scalability of Redshift is definitely a huge advantage of this solution.

5.2 Microsoft Azure SQL Data Warehouse

Microsoft Azure SQL data warehouse is a cloud-based, scale-out database capable of processing massive volumes of data, both relational and non-relational. It is a massively processing (MPP) distributed database system. “It provides SaaS, PaaS and IaaS services and supports many different programming languages, tools and frameworks, including non-Microsoft software (“SQL Data Warehouse | Microsoft Azure,” n.d.).

SQL Data Warehouse is based on the SQL Server relational database engine and integrates with the tool that its users may be familiar with. This includes (“SQL Data Warehouse | Microsoft Azure,” n.d.):

- Analysis Services;
- Integration Services;
- Reporting Services;
- Cloud-based tools.

The Microsoft Azure SQL Data Warehouse is composed by a Control Node, Compute nodes and Storage. It also has a service called Data Movement Service that is responsible for the data movement between the nodes (“SQL Data Warehouse | Microsoft Azure,” n.d.).

Like the Leader Node of Amazon Redshift, the Azure Control Node manages and optimizes queries and is responsible for the coordination of all the data movement and computation required to run parallel queries (see Figure 2). When a request is made to SQL Data Warehouse, the control node transforms it

into separate queries that run on each compute node in parallel.

The compute nodes are SQL databases that store the data and process queries. When data is added, it is distributed to the compute nodes and when the data is requested these nodes are the workers that run queries in parallel. After processing, they pass the results back to the control node so it can aggregate the results and return the final result to the user.

All the data stored in Azure SQL Data Warehouse is stored in Azure Blob Storage – it is a service that stores unstructured data in the cloud as objects/blobs. Blob Storage can store any type of text or binary data, such as a document, media file or application installer. When compute nodes interact with data, they write and read directly to and from blob storage. Compute and Storage are independent.

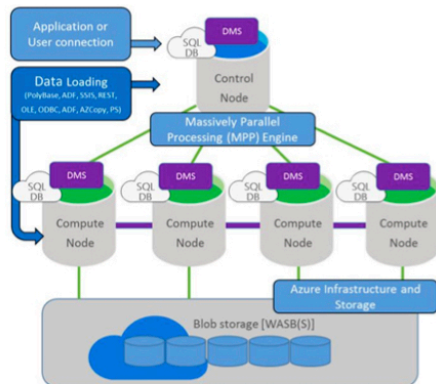


Figure 2 – Microsoft Azure SQL Data Warehouse system architecture (source: “SQL Data Warehouse | Microsoft Azure,” n.d.)

As previously referred, Data Movement Service (DMS) is responsible for all the data movements between the nodes. It gives the compute nodes access to data they need for joins and aggregations. It is not an Azure service but a Windows service that runs alongside SQL Database on all the nodes and it is only visible on query plans because they include some DMS operations since data movement is necessary to run a query in parallel.

Azure is an enterprise-level SQL data warehouse that extends the SQL Server family of products and services into the cloud. Azure can also scale storage and computing so the customers only have to pay for what they require.

6 COMPARING CLOUD DATA WAREHOUSES

Data warehouses in the cloud are gaining popularity because cloud vendors offer DW services at lower costs. While Amazon Redshift is the number one in the market, according to Gartner (2016), Azure offers a competitive platform to be considered.

Redshift and Azure SQL Data Warehouse both support petabyte scale systems. Both of them have leader or control nodes and compute nodes. The biggest difference between Azure SQL Data Warehouse and Redshift is the decoupling of storage and compute resources.

About scalability, in Redshift, when the cluster is modified, the changes are immediately applied. While the new clusters are being provisioned, the current cluster is available in read only mode, so during this process the data is available for read operations. After the new clusters are provisioned, the data is copied.

In Azure SQL Data Warehouse, the scaling of the clusters can happen in few minutes. The scale out can be done for compute and storage units independently. Azure SQL Data Warehouse also supports pausing a compute operation. There is no cost applied when the compute nodes are in pause state; only a storage cost is charged.

From Data sources, data can be integrated with Redshift from Amazon S3 storage. If there is an on-premises database to be integrated with Redshift, the data needs to be extracted from the database to a file and then imported up into S3.

Azure SQL Data Warehouse is integrated with Azure Blob storage. It uses a similar approach as Redshift to import the data from SQL Server. The SQL Server data is exported to a text file and then copied across to Azure Blob storage.

Comparing the adoption of public clouds, in particular AWS and Azure, we can see that AWS is the number one adopted cloud solution for the most respondent users of the 2017 Rightscale survey, as shown in Figure 3.

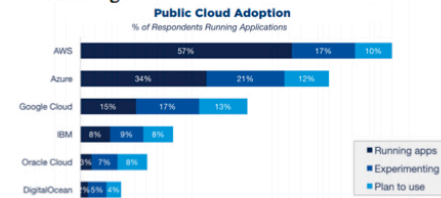


Figure 3 – Adoption of public cloud in 2017 (source: “RightScale 2017 - State of the cloud report,” 2017)

Although AWS continues to lead in public cloud adoption (57 percent of respondents currently run applications in AWS), this number has stayed the same as in 2016.

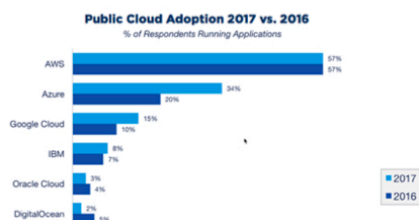


Figure 4 – Adoption of public cloud in 2017 vs 2016 (source: “RightScale 2017 - State of the cloud report,” 2017)

In contrast, over the last year, happened a significant growth in the percentage of respondents running applications in Azure and Google, the second and third public cloud providers (see Figure 4). Overall Azure adoption grew from 20 to 34 percent of respondents, reducing the AWS lead. Google also increased from 10 to 15 percent.

A comparison of system properties is shown in Table 1.

RedShift and Azure SQL have a database model based on relational database management system (RDBMS) which supports the relational data model.

Amazon Redshift is built around industry-standard SQL, with added functionality to manage very large datasets and support high-performance analysis that data. Although it is based on PostgreSQL, there are some unsupported features, data types and functions. Some SQL features are also implemented differently, for example:

- CREATE TABLE
- ALTER TABLE
- INSERT, UPDATE and DELETE

Amazon Redshift does not support tablespaces, table partitioning, inheritance, and certain constraints. The Amazon Redshift implementation of CREATE TABLE enables users to define the sort and distribution algorithms for tables to optimize parallel processing. ALTER COLUMN actions are not supported. ADD COLUMN supports adding only one column in each ALTER TABLE statement. Using INSERT, UPDATE, and DELETE the WITH is not supported. For the complete list of unsupported features, data types and functions, our suggestion is to check on AWS documentation, in

particular the one that concerns about Amazon Redshift and PostgreSQL (“Amazon Redshift and PostgreSQL - Amazon Redshift,” n.d.).

Table 1 – Comparison Redshift Vs Azure SQL Data Warehouse

<i>System Properties</i>	Amazon Redshift	Microsoft Azure SQL Data Warehouse
Database model	Relational DBMS	Relational DBMS
Developer	Amazon (based on PostgreSQL)	Microsoft
Licence	Commercial	Commercial
Cloud based	Yes	Yes
Implementation language	C	C++
XML support	No	Yes
SQL standard support	Does not fully support	Yes
Supported programming languages	All languages supporting JDBC/ODBC	.Net, Java, JavaScript, PHP, Python, Ruby
Server-side scripts	User defined Python functions	Transact SQL
Support for concurrent data manipulation	Yes	Yes
MapReduce API support	No	No
In-memory support	Yes	No
Control over node configuration	Yes	No

In-Memory OLTP is a technology for optimizing performance of transaction processing, data ingestion, data load, and transient data scenarios. In-Memory support is available on RedShift but not in Azure SQL Data Warehouse because it is only available for OLTP workloads in SQL Server since version 2014 and Azure SQL Database.

In MapReduce support property, both databases do not offer an API for user-defined Map/Reduce methods. In case of RedShift, it is possible to combine MapReduce with RedShift by processing input data with MapReduce and import results to RedShift. In case of Azure SQL Data Warehouse,

Polybase unifies data in relational data stores with non-relational ones, combining data from both RDBMS and Hadoop so that users don't need to understand HDFS or MapReduce.

Redshift and Azure SQL Data Warehouse offer many similar capabilities, so it is not necessarily a matter of one provider being better or worse than the other. It all depends on what our business needs, but each solution has pros and cons. See Table 2 to find some pros and cons of Amazon RedShift and Microsoft Azure SQL Data Warehouse cloud solutions.

Table 2 – Pros and Cons of Redshift and Azure SQL Data Warehouse

	Pros	Cons
<i>Amazon Redshift</i>	Performance through use of local storage	Compute cannot be scaled independent of storage (and vice versa)
	Loading data from S3 is very fast	Can't pause resources
	Beyond Petabyte	Queries that require joins against multiple columns can suffer in performance
	Columnar data store allows high performance queries on large volumes of data	
	Familiarity with PostgreSQL makes adopting Redshift easier	
<i>Microsoft Azure SQL Data Warehouse</i>	Resources can be paused during idle time in workload	Can only run 32 concurrent queries (maximum)
	Scale separate compute and storage resources and pay only for what is used	Not fully supports T-SQL
	Excellent integration with Azure Services	

7 CONCLUSIONS AND FUTURE WORK

Data warehouses are defined as customized data storage that aggregate data from multiple sources and store it in a common location to be able to run reports and queries over it. Many companies use data warehouses to compile regular financial reports or business metric analyses.

In this paper we analyse cloud data warehousing area that is the convergence of three trends – huge changes in data sources, volume and complexity; the need for data access and analytics; and better technology that increased the efficiency of data access, analytics and storage. Traditional data warehouse systems were not designed to handle the volume, variety and complexity of today's data.

The integration of a Cloud DW solution needs a very well defined strategy that would involve Cloud Computing capabilities. The success of the implementation depends on the existence of a service-oriented strategy at the organization level, which would provide the necessary infrastructure for the Cloud implementation.

In this study we conclude that there are several challenges when deploying data warehouses into the cloud:

- Importing data for the data warehouse into the cloud for storage can be a challenge, because when using the cloud, a customer is dependent on the Internet connection and the infrastructure of the cloud provider. It can be necessary to use a dedicated communication line to mitigate the connection problems but it as a cost;
- Getting large amounts of data from cloud storage to compute nodes provided by the cloud solution for computing can lead to a performance issue;
- Loss of control can lead to issues involving security and trust.

In this work we analyse and evaluate two of the most popular cloud data warehousing solutions: Amazon Redshift and Microsoft Azure SQL Data Warehouse.

Some conclusions concerning the two cloud data warehouse solutions have been taken:

- Using Redshift to scale our data warehouse we must increase both the compute and storage units. With Azure SQL DW, compute and storage is decoupled so we can scale them individually. This is a very different economic model that can save customers a lot of money as they don't

have to purchase additional storage when they just need more compute power, or vice-versa.

- Azure SQL DW has the ability to pause compute when not in use so we only pay for storage, as opposed to Redshift in which we are billed 24/7 for all the virtual machines that make up the nodes in our cluster.
- RedShift is easier to configure than Azure SQL Data Warehouse and takes less time to be online and available after its setup.

As future work we intend to analyze these two platforms with data from a company, and a recommendation will be given of what is the best cloud data warehouse solution in the market based on a set of criteria.

REFERENCES

- Almeida, R., Vieira, J. Vieira, M. Madeira, H. and Bernardino, J. "Efficient Data Distribution for DWS". In International Conference on Data Warehousing and Knowledge Discovery - DaWaK, pages 75–86, 2008.
- Almeida, P., and Bernardino, J. "Big Data Open Source Platforms". BigData Congress 2015: 268-275
- Almeida, P., and Bernardino, J. "A comprehensive overview of open source big data platforms and frameworks", International Journal of Big Data (IJBD), 2(3), 2015, pp. 15-33.
- Amazon Redshift and PostgreSQL - Amazon Redshift [WWW Document], n.d. URL http://docs.aws.amazon.com/redshift/latest/dg/c_redshift-and-postgres-sql.html (accessed 9.2.17).
- Amazon Redshift vs. Microsoft Azure SQL Data Warehouse vs. Microsoft Azure SQL Database Comparison [WWW Document], n.d. URL <https://db-engines.com/en/system/Amazon+Redshift%3BMicrosoft+Azure+SQL+Data+Warehouse%3BMicrosoft+Azure+SQL+Database> (accessed 9.2.17).
- Armbrust, M., Fox, A., Griffith, R., Joseph, A.D., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A., Stoica, I., Zaharia, M., 2010. A View of Cloud Computing. Communications of ACM 53, 50–58.
- Combining Hadoop/Elastic Mapreduce with AWS Redshift Data Warehouse [WWW Document], n.d. URL <http://atbros.com/2013/02/25/combining-hadoopelastic-mapreduce-with-aws-redshift-data-warehouse/> (accessed 9.3.17).
- Data Warehouse System Architecture - Amazon Redshift [WWW Document], n.d. URL https://docs.aws.amazon.com/redshift/latest/dg/c_high_level_system_architecture.html (accessed 1.1.17).
- Database Manag. Solut. Anal. URL <https://www.gartner.com/doc/reprints?id=1-2ZFVZ5B&ct=160225&st=sb> (accessed 1.2.17).
- Gartner, 2016. Magic Quadrant for Data Warehouse and Database Management Solutions for Analytics [WWW Document]. Magic Quadr. Data Wareh.
- Goutas, L., Sutanto, J., Aldarbesti, H., 2016. The Building Blocks of a Cloud Strategy: Evidence from Three SaaS Providers. Communications of ACM 59, 90–97.
- Gupta, A., Agarwal, D., Tan, D., Kulesza, J., Pathak, R., Stefani, S., Srinivasan, V., 2015. Amazon Redshift and the Case for Simpler Data Warehouses, in: Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data, SIGMOD '15. ACM, New York, NY, USA, pp. 1917–1923.
- Hemlata Verna, 2013. Data-warehousing on Cloud Computing, in: International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) Volume 2, Issue 2, February 2013.
- Kaur, H., Agrawal P., and Dhiman, A., "Visualizing Clouds on Different Stages of DWH - An Introduction to Data Warehouse as a Service," 2012 Int. Conf. on Computing Sciences, Phagwara, 2012, pp. 356-359.
- Key Concepts & Architecture — Snowflake Documentation [WWW Document], n.d. URL <https://docs.snowflake.net/manuals/user-guide/intro-key-concepts.html> (accessed 2.16.17).
- Mathur, A., Mathur, M. & Upadhyay, P., 2011. Cloud Based Distributed Databases: The Future Ahead. In : International Journal on Computer Science and Engineering (IJCSE) , 3(6), pp.2477-81.
- Miller, J.A., Bowman, C., Harish, V.G., Quinn, S., 2016. Open Source Big Data Analytics Frameworks Written in Scala, in: 2016 IEEE International Congress on Big Data (BigData Congress), pp. 389–393.
- Morshed, S.J., Rana, J., Milrad, M., 2016. Open Source Initiatives and Frameworks Addressing Distributed Real-Time Data Analytics, in: 2016 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW). Presented at the 2016 IEEE Int. Parallel and Distributed Processing Symposium Workshops (IPDPSW), pp. 1481–1484.
- Popeangã, J, 2014. Shared-Nothing Cloud Data Warehouse Architecture, in: Database Systems Journal vol. V, no. 4/2014.
- RightScale 2017 - State of the cloud report [WWW Document], 2017. URL: <https://assets.rightscale.com/uploads/pdfs/RightScale-2017-State-of-the-Cloud-Report.pdf> (accessed 8.11.17).
- SQL Data Warehouse | Microsoft Azure [WWW Document], n.d. URL <https://azure.microsoft.com/en-us/services/sql-data-warehouse/> (accessed 11.13.16).
- Talia, D., 2013. Clouds for Scalable Big Data Analytics. Computer 46, 98–101. doi:10.1109/MC.2013.162
- Tereso, M., and Bernardino, J. "Open source business intelligence tools for SMEs". Information Systems and Technologies (CISTI), 6th Iberian Conference on, IEEE (2011) 1–4.

Annex F

Amazon RedShift Pricing

Services		Estimate of your Monthly Bill (\$ 786.79)
Estimate of Your Monthly Bill		
<input checked="" type="checkbox"/> Show First Month's Bill (include all one-time fees, if any)		
Below you will see an estimate of your monthly bill. Expand each line item to see cost breakout of each service. To save this bill and input values, click on 'Save and Share' button. To remove the service from the estimate, jump back to the service and clear the specific service's form.		
Export to CSV		Save and Share
<input type="checkbox"/> Amazon S3 Service (US-East)		\$ 47.11
<input type="checkbox"/> Amazon Redshift Service (US-East)		\$ 622.20
<input type="checkbox"/> Amazon VPC Service (US-East)		\$ 36.60
<input type="checkbox"/> AWS Data Transfer In		\$ 0.00
<input type="checkbox"/> AWS Data Transfer Out		\$ 82.26
<input type="checkbox"/> AWS Support (Basic)		\$ 0.00
Free Tier Discount:		\$ -1.38
Total Monthly Payment:		\$ 786.79

Microsoft Azure SQL Data Warehouse Pricing

Mestrado [↗](#) [↶](#) [🗑](#)

SQL Data Warehouse Compute: 1 x 744 Hours, Storage: 2 TB €1,176.69

Bandwidth Zone 1: North America, Europe, 1 TB €74.76

Support 📘

SUPPORT:

Included €0.00

Programs and Offers

LICENSING PROGRAM:

Microsoft Online Services Program (MOSP) 📘

SHOW DEV/TEST PRICING 📘

Estimated monthly cost €1,251.45

[📄 Export](#) Euro (€) ⌵

Snowflake Pricing

	Price (€)	Space (TB)	Hours	Total (€)
Storage - US West (Oregon)/US East (N. Virginia)	33,2	2		66,4
Compute - US West (Oregon)/US East (N. Virginia)	1,66		744	1235,04
				1301,44

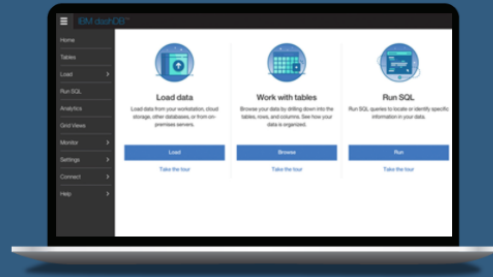
IBM dashDB Pricing

IBM Db2 Warehouse

IBM Db2 Warehouse (formerly IBM dashDB Local) is a client-managed, private cloud data warehouse for Docker container supported infrastructures.


Start your free trial


Launch interactive demo



Please contact us for pricing

We can help you find the right edition and pricing for your business needs.

 Email an expert

 Call IBM 1-855-300-7287

Priority code: WDP

A contact with IBM has been done but not answered.

Asterisks (*) indicate fields required to complete this transaction.

Business Contact Information

Country or Region *
Portugal

Salutation
(e.g. Mr., Mrs.)
Mr

First name *
Pedro

Last name *
Ferreira

Work e-mail address *
1030537@isep.ipp.pt

Phone *
[Empty field]

State or Province *
Porto

Company *
DEI-ISEP

Please let us know how we can be of help *
Hi, I'm currently writing my master thesis under the DW in Cloud theme and in order to study and evaluate dashDB database for cloud data warehouse usage, can you

Please keep me informed of products, services and offerings from IBM companies worldwide.
I accept [IBM's Privacy statement](#).

Pivotal Greenplum

Let's Get Started

Press/Analyst ▾

Pedro Ferreira

1030537@isep.ipp.pt +351917167287

DEI-ISEP Portugal ▾

Porto Other ▾

Hi.
I'm currently writing my master thesis under the DW in Cloud theme and in order to study and evaluate Pivotal Greenplum database for data warehouse usage, can you please give me more details about the database and pricing model? Thank you in advance. Regards.

Submit [Looking for support? →](#)

HPE Vertica

The contact with HP was done by online chat requesting information about the database and pricing models. The answer was not positive because they can only give pricing information to clients.

Teradata

IntelliCloud (Teradata Database)

Infrastructure	Instance Type	Database Tier	Monthly Commit		1-Year Commit		3-Year Commit	
			Monthly	Hourly	Monthly	Hourly	Monthly	Hourly
IntelliBase	IntelliBase	Base	\$10,985	\$14.76	\$6,990	\$9.40	\$4,993	\$6.71
		Advanced	\$16,103	\$21.64	\$10,249	\$13.78	\$7,319	\$9.84

- Hourly prices are effective rates for comparison purposes only based on a 31-day month; services not sold on hourly basis.
- IntelliCloud (Teradata Database) services are billed annually in advance (except for the Monthly Commit option).
- Minimum subscription is for 2 nodes; pricing for 1 node is shown above.
- Currently available from Teradata data centers in the US only.

Oracle Cloud

Metered Services

Buy Now

Product (per OCPU)	General Purpose Compute		High-Memory Compute	
	Per Month	Per Hour	Per Month	Per Hour
Standard Package	€521.00	€0.88	€608.00	€1.02
Enterprise Package	€2,605.00	€4.38	€2,691.00	€4.52
High Performance Package	€3,473.00	€6.00	€3,560.00	€6.00
Extreme Performance Package	€4,341.00	€7.00	€4,428.00	€7.00

- All packages include Oracle Database Transparent Data Encryption. Standard package includes the Oracle Database Standard Edition. Enterprise package includes the Oracle Database Enterprise Edition. High Performance extends the Enterprise package with the following options: Multitenant, Partitioning, Real Application Testing, Advanced Compression, Advanced Security, Label Security, Database Vault, OLAP, Advanced Analytics, Spatial & Graph, Diagnostics Pack, Tuning Pack, Database Lifecycle Management Pack, Data Masking and Subsetting Pack and Cloud Management Pack for Oracle Database. Extreme Performance package extends the High Performance package with the following options: RAC (Real Application Clusters), In-Memory Database, Active Data Guard.
- This bundled metered service allows you to set and activate the offerings currently available for Oracle Database Cloud Service. Note that the purchase of the 'Pay as you go' service does not require any upfront payment; monthly invoice will be generated based on usage. Discounted pre-paid option is available, learn more [here](#).
- For Compute shapes, see [here](#).
- For more information on Database versions, review the [documentation](#).
- Pricing is based on each individual service. For Database Backup Service and Storage Service charges, review [Database Backup Service pricing](#) and [Storage Cloud Service pricing](#).

Google BigQuery

1 x	n1-standard-4 Sustained Usage Discount Monthly Breakdown: • 1st ¼ - 182.5 hrs @ 0.0% off: \$34.67 • 2nd ¼ - 182.5 hrs @ 20.0% off: \$27.74 (\$6.94 saved) • 3rd ¼ - 182.5 hrs @ 40.0% off: \$20.80 (\$13.87 saved) • 4th ¼ - 182.5 hrs @ 60.0% off: \$13.87 (\$20.81 saved)	730 total hours per month	\$520.41
Cloud Storage	Multi-Regional storage	2048 GB	\$83.95
Datastore Storage	BigQuery Datastore Storage	2 GB / 2048 GB	\$91.06 / \$393.57
Dataproc		5952	\$69.52
Cloud Dataflow	1 x n1-standard-4 workers in Batch Mode	32	\$8.43
db-pg-g1-small	2048 GB	730 total hours per month	\$393.71
Network Bandwidth	Egress - Americas and EMEA	2048 GB	0
Network Bandwidth	Interconnect Europe	2048 GB	\$112.40
Total Estimated Monthly Cost			\$1673.05

Annex G

Database Creation Scripts

```
CREATE TABLE DimAccount(  
    AccountKey int NOT NULL,  
    ParentAccountKey int,  
    AccountCodeAlternateKey int,  
    ParentAccountCodeAlternateKey int,  
    AccountDescription nvarchar(50),  
    AccountType nvarchar(50),  
    Operator nvarchar(50),  
    CustomMembers nvarchar(300),  
    ValueType nvarchar(50),  
    CustomMemberOptions nvarchar(200))  
;  
  
CREATE TABLE DimCurrency(  
    CurrencyKey int NOT NULL,  
    CurrencyAlternateKey nchar(3) NOT NULL,  
    CurrencyName nvarchar(50) NOT NULL);  
  
CREATE TABLE DimCustomer(  
    CustomerKey int NOT NULL,  
    GeographyKey int,  
    CustomerAlternateKey nvarchar(15) NOT NULL,  
    Title nvarchar(8),  
    FirstName nvarchar(50),  
    MiddleName nvarchar(50),  
    LastName nvarchar(50),  
    NameStyle boolean,  
    BirthDate date,  
    MaritalStatus nchar(1),  
    Suffix nvarchar(10),  
    Gender nvarchar(1),  
    EmailAddress nvarchar(50),  
    YearlyIncome float,  
    TotalChildren int,  
    NumberChildrenAtHome int,  
    EnglishEducation nvarchar(40),  
    SpanishEducation nvarchar(40),  
    FrenchEducation nvarchar(40),  
    EnglishOccupation nvarchar(100),  
    SpanishOccupation nvarchar(100),
```

```
FrenchOccupation nvarchar(100),  
HouseOwnerFlag nchar(1),  
NumberCarsOwned int,  
AddressLine1 nvarchar(120),  
AddressLine2 nvarchar(120),  
Phone nvarchar(20),  
DateFirstPurchase date,  
CommuteDistance nvarchar(15));
```

```
CREATE TABLE DimDate(  
    DateKey int NOT NULL,  
    FullDateAlternateKey date NOT NULL,  
    DayNumberOfWeek int NOT NULL,  
    EnglishDayNameOfWeek nvarchar(10) NOT NULL,  
    SpanishDayNameOfWeek nvarchar(10) NOT NULL,  
    FrenchDayNameOfWeek nvarchar(10) NOT NULL,  
    DayNumberOfMonth int NOT NULL,  
    DayNumberOfYear smallint NOT NULL,  
    WeekNumberOfYear int NOT NULL,  
    EnglishMonthName nvarchar(10) NOT NULL,  
    SpanishMonthName nvarchar(10) NOT NULL,  
    FrenchMonthName nvarchar(10) NOT NULL,  
    MonthNumberOfYear int NOT NULL,  
    CalendarQuarter int NOT NULL,  
    CalendarYear smallint NOT NULL,  
    CalendarSemester int NOT NULL,  
    FiscalQuarter int NOT NULL,  
    FiscalYear smallint NOT NULL,  
    FiscalSemester int NOT NULL);
```

```
CREATE TABLE DimDepartmentGroup(  
    DepartmentGroupKey int NOT NULL,  
    ParentDepartmentGroupKey int,  
    DepartmentGroupName nvarchar(50));
```

```
CREATE TABLE DimEmployee(  
    EmployeeKey int NOT NULL,  
    ParentEmployeeKey int,  
    EmployeeNationalIDAlternateKey nvarchar(15),  
    ParentEmployeeNationalIDAlternateKey nvarchar(15),  
    SalesTerritoryKey int,  
    FirstName nvarchar(50) NOT NULL,  
    LastName nvarchar(50) NOT NULL,  
    MiddleName nvarchar(50),  
    NameStyle boolean NOT NULL,
```

```

Title nvarchar(50),
HireDate date,
BirthDate date,
LoginID nvarchar(256),
EmailAddress nvarchar(50),
Phone nvarchar(25),
MaritalStatus nchar(1),
EmergencyContactName nvarchar(50),
EmergencyContactPhone nvarchar(25),
SalariedFlag boolean,
Gender nchar(1),
PayFrequency int,
BaseRate float,
VacationHours smallint,
SickLeaveHours smallint,
CurrentFlag boolean NOT NULL,
SalesPersonFlag boolean NOT NULL,
DepartmentName nvarchar(50),
StartDate date,
EndDate date,
Status nvarchar(50));

```

```

CREATE TABLE DimGeography(
    GeographyKey int NOT NULL,
    City nvarchar(30),
    StateProvinceCode nvarchar(3),
    StateProvinceName nvarchar(50),
    CountryRegionCode nvarchar(3),
    EnglishCountryRegionName nvarchar(50),
    SpanishCountryRegionName nvarchar(50),
    FrenchCountryRegionName nvarchar(50),
    PostalCode nvarchar(15),
    SalesTerritoryKey int);

```

```

CREATE TABLE DimOrganization(
    OrganizationKey int NOT NULL,
    ParentOrganizationKey int,
    PercentageOfOwnership nvarchar(16),
    OrganizationName nvarchar(50),
    CurrencyKey int);

```

```

CREATE TABLE DimProduct(
    ProductKey int NOT NULL,
    ProductAlternateKey nvarchar(25),
    ProductSubcategoryKey int,

```

```

WeightUnitMeasureCode nchar(3),
SizeUnitMeasureCode nchar(3),
EnglishProductName nvarchar(50) NOT NULL,
SpanishProductName nvarchar(50),
FrenchProductName nvarchar(50),
StandardCost float,
FinishedGoodsFlag boolean NOT NULL,
Color nvarchar(15) NOT NULL,
SafetyStockLevel smallint,
ReorderPoint smallint,
ListPrice float,
[Size] nvarchar(50),
SizeRange nvarchar(50),
Weight float,
DaysToManufacture integer,
ProductLine nchar(2),
DealerPrice float,
Class nchar(2),
Style nchar(2),
ModelName nvarchar(50),
EnglishDescription nvarchar(400),
FrenchDescription nvarchar(400),
ChineseDescription nvarchar(400),
ArabicDescription nvarchar(400),
HebrewDescription nvarchar(400),
ThaiDescription nvarchar(400),
GermanDescription nvarchar(400),
JapaneseDescription nvarchar(400),
TurkishDescription nvarchar(400),
StartDate datetime,
EndDate datetime,
Status nvarchar(7));

```

```

CREATE TABLE DimProductCategory(
    ProductCategoryKey int NOT NULL,
    ProductCategoryAlternateKey int,
    EnglishProductCategoryName nvarchar(50) NOT NULL,
    SpanishProductCategoryName nvarchar(50) NOT NULL,
    FrenchProductCategoryName nvarchar(50) NOT NULL);

```

```

CREATE TABLE DimProductSubcategory(
    ProductSubcategoryKey int NOT NULL,
    ProductSubcategoryAlternateKey int,
    EnglishProductSubcategoryName nvarchar(50) NOT NULL,
    SpanishProductSubcategoryName nvarchar(50) NOT NULL,

```

```
FrenchProductSubcategoryName nvarchar(50) NOT NULL,  
ProductCategoryKey int);
```

```
CREATE TABLE DimPromotion(  
    PromotionKey int NOT NULL,  
    PromotionAlternateKey int,  
    EnglishPromotionName nvarchar(255),  
    SpanishPromotionName nvarchar(255),  
    FrenchPromotionName nvarchar(255),  
    DiscountPct float,  
    EnglishPromotionType nvarchar(50),  
    SpanishPromotionType nvarchar(50),  
    FrenchPromotionType nvarchar(50),  
    EnglishPromotionCategory nvarchar(50),  
    SpanishPromotionCategory nvarchar(50),  
    FrenchPromotionCategory nvarchar(50),  
    StartDate datetime NOT NULL,  
    EndDate datetime,  
    MinQty int,  
    MaxQty int);
```

```
CREATE TABLE DimReseller(  
    ResellerKey int NOT NULL,  
    GeographyKey int,  
    ResellerAlternateKey nvarchar(15),  
    Phone nvarchar(25),  
    BusinessType varchar(20) NOT NULL,  
    ResellerName nvarchar(50) NOT NULL,  
    NumberEmployees int,  
    OrderFrequency char(1),  
    OrderMonth int,  
    FirstOrderYear int,  
    LastOrderYear int,  
    ProductLine nvarchar(50),  
    AddressLine1 nvarchar(60),  
    AddressLine2 nvarchar(60),  
    AnnualSales float,  
    BankName nvarchar(50),  
    MinPaymentType int,  
    MinPaymentAmount float,  
    AnnualRevenue float,  
    YearOpened int);
```

```
CREATE TABLE DimSalesReason(  
    SalesReasonKey int NOT NULL,  
    SalesReasonAlternateKey int,  
    EnglishSalesReasonName nvarchar(255),  
    SpanishSalesReasonName nvarchar(255),  
    FrenchSalesReasonName nvarchar(255),  
    SalesReasonCategory nvarchar(50),  
    SalesReasonSubcategory nvarchar(50),  
    SalesReasonType nvarchar(50),  
    SalesReasonSubcategoryKey int);
```

```
SalesReasonKey int NOT NULL,  
SalesReasonAlternateKey int NOT NULL,  
SalesReasonName nvarchar(50) NOT NULL,  
SalesReasonReasonType nvarchar(50) NOT NULL);
```

```
CREATE TABLE DimSalesTerritory(  
SalesTerritoryKey int NOT NULL,  
SalesTerritoryAlternateKey int,  
SalesTerritoryRegion nvarchar(50) NOT NULL,  
SalesTerritoryCountry nvarchar(50) NOT NULL,  
SalesTerritoryGroup nvarchar(50));
```

```
CREATE TABLE DimScenario(  
ScenarioKey int NOT NULL,  
ScenarioName nvarchar(50));
```

```
CREATE TABLE FactCallCenter(  
FactCallCenterID int NOT NULL,  
DateKey int NOT NULL,  
WageType nvarchar(15) NOT NULL,  
Shift nvarchar(20) NOT NULL,  
LevelOneOperators smallint NOT NULL,  
LevelTwoOperators smallint NOT NULL,  
TotalOperators smallint NOT NULL,  
Calls int NOT NULL,  
AutomaticResponses int NOT NULL,  
Orders int NOT NULL,  
IssuesRaised smallint NOT NULL,  
AverageTimePerIssue smallint NOT NULL,  
ServiceGrade float NOT NULL);
```

```
CREATE TABLE FactCurrencyRate(  
CurrencyKey int NOT NULL,  
DateKey int NOT NULL,  
AverageRate float NOT NULL,  
EndOfDayRate float NOT NULL);
```

```
CREATE TABLE FactFinance(  
FinanceKey int NOT NULL,  
DateKey int NOT NULL,  
OrganizationKey int NOT NULL,  
DepartmentGroupKey int NOT NULL,  
ScenarioKey int NOT NULL,  
AccountKey int NOT NULL,  
Amount float NOT NULL);
```



```

CREATE TABLE FactInternetSales(
    ProductKey int NOT NULL,
    OrderDateKey int NOT NULL,
    DueDateKey int NOT NULL,
    ShipDateKey int NOT NULL,
    CustomerKey int NOT NULL,
    PromotionKey int NOT NULL,
    CurrencyKey int NOT NULL,
    SalesTerritoryKey int NOT NULL,
    SalesOrderNumber nvarchar(20) NOT NULL,
    SalesOrderLineNumber int NOT NULL,
    RevisionNumber int NOT NULL,
    OrderQuantity smallint NOT NULL,
    UnitPrice float NOT NULL,
    ExtendedAmount float NOT NULL,
    UnitPriceDiscountPct float NOT NULL,
    DiscountAmount float NOT NULL,
    ProductStandardCost float NOT NULL,
    TotalProductCost float NOT NULL,
    SalesAmount float NOT NULL,
    TaxAmt float NOT NULL,
    Freight float NOT NULL,
    CarrierTrackingNumber nvarchar(25),
    CustomerPONumber nvarchar(25));

```

```

CREATE TABLE FactInternetSalesReason(
    SalesOrderNumber nvarchar(20) NOT NULL,
    SalesOrderLineNumber int NOT NULL,
    SalesReasonKey int NOT NULL);

```

```

CREATE TABLE ProspectiveBuyer(
    ProspectiveBuyerKey int NOT NULL,
    ProspectAlternateKey nvarchar(15),
    FirstName nvarchar(50),
    MiddleName nvarchar(50),
    LastName nvarchar(50),
    BirthDate datetime,
    MaritalStatus nchar(1),
    Gender nvarchar(1),
    EmailAddress nvarchar(50),
    YearlyIncome float,
    TotalChildren int,
    NumberChildrenAtHome int,

```

```
Education nvarchar(40),  
Occupation nvarchar(100),  
HouseOwnerFlag nchar(1),  
NumberCarsOwned int,  
AddressLine1 nvarchar(120),  
AddressLine2 nvarchar(120),  
City nvarchar(30),  
StateProvinceCode nvarchar(3),  
PostalCode nvarchar(15),  
Phone nvarchar(20),  
Salutation nvarchar(8),  
Unknown int);
```

```
CREATE TABLE FactResellerSales(  
    ProductKey int NOT NULL,  
    OrderDateKey int NOT NULL,  
    DueDateKey int NOT NULL,  
    ShipDateKey int NOT NULL,  
    ResellerKey int NOT NULL,  
    EmployeeKey int NOT NULL,  
    PromotionKey int NOT NULL,  
    CurrencyKey int NOT NULL,  
    SalesTerritoryKey int NOT NULL,  
    SalesOrderNumber nvarchar(20) NOT NULL,  
    SalesOrderLineNumber int NOT NULL,  
    RevisionNumber int,  
    OrderQuantity smallint,  
    UnitPrice float,  
    ExtendedAmount float,  
    UnitPriceDiscountPct float,  
    DiscountAmount float,  
    ProductStandardCost float,  
    TotalProductCost float,  
    SalesAmount float,  
    TaxAmt float,  
    Freight float,  
    CarrierTrackingNumber nvarchar(25),  
    CustomerPONumber nvarchar(25));
```

```
CREATE TABLE FactSalesQuota(  
    SalesQuotaKey int NOT NULL,  
    EmployeeKey int NOT NULL,  
    DateKey int NOT NULL,  
    CalendarYear smallint NOT NULL,  
    CalendarQuarter int NOT NULL,
```

SalesAmountQuota float NOT NULL);

```
CREATE TABLE FactSurveyResponse(  
    SurveyResponseKey int NOT NULL,  
    DateKey int NOT NULL,  
    CustomerKey int NOT NULL,  
    ProductCategoryKey int NOT NULL,  
    EnglishProductCategoryName nvarchar(50) NOT NULL,  
    ProductSubcategoryKey int NOT NULL,  
    EnglishProductSubcategoryName nvarchar(50) NOT NULL);
```

Annex H

Program written in Python to upload files into S3 Storage

```
from datetime import datetime
import glob
import os
from boto3.s3.transfer import S3Transfer
import boto3

client = boto3.client('s3',
                      aws_access_key_id="AKIAJUOL3JIJ03MZZLCA",
                      aws_secret_access_key="VZeC0DxBBtpKA9ACo3Y8GbQyryX9IGcw0IeQo0c7")

transfer = S3Transfer(client)

for filename in glob.iglob('/Users/pedroferreira/Desktop/RS/*.txt'):
    startTime= datetime.now()
    transfer.upload_file(filename, "pedro-m3", os.path.basename(filename))
    timeElapsed=datetime.now()-startTime
    print(filename)
    print('Time elapsed (hh:mm:ss.ms) {}'.format(timeElapsed))
```

Annex I

COPY command script to load data into RedShift

```
copy DimAccount from 's3://pedro-m3/DimAccount.txt' DELIMITER '|' credentials
'aws_access_key_id=AKIAJUOL3JIJO3MZZLCA;aws_secret_access_key=VZeCODxBBtpK
A9ACo3Y8GbQyryX9IGcwOleQo0c7';
copy DimCurrency from 's3://pedro-m3/DimCurrency.txt' DELIMITER '|' credentials
'aws_access_key_id=AKIAJUOL3JIJO3MZZLCA;aws_secret_access_key=VZeCODxBBtpK
A9ACo3Y8GbQyryX9IGcwOleQo0c7';
copy DimCustomer from 's3://pedro-m3/DimCustomer.txt' DELIMITER '|' credentials
'aws_access_key_id=AKIAJUOL3JIJO3MZZLCA;aws_secret_access_key=VZeCODxBBtpK
A9ACo3Y8GbQyryX9IGcwOleQo0c7';
copy DimDate from 's3://pedro-m3/DimDate.txt' DELIMITER '|' credentials
'aws_access_key_id=AKIAJUOL3JIJO3MZZLCA;aws_secret_access_key=VZeCODxBBtpK
A9ACo3Y8GbQyryX9IGcwOleQo0c7';
copy DimDepartmentGroup from 's3://pedro-m3/DimDepartmentGroup.txt'
DELIMITER '|' credentials
'aws_access_key_id=AKIAJUOL3JIJO3MZZLCA;aws_secret_access_key=VZeCODxBBtpK
A9ACo3Y8GbQyryX9IGcwOleQo0c7';
copy DimEmployee from 's3://pedro-m3/DimEmployee.txt' DELIMITER '|' credentials
'aws_access_key_id=AKIAJUOL3JIJO3MZZLCA;aws_secret_access_key=VZeCODxBBtpK
A9ACo3Y8GbQyryX9IGcwOleQo0c7';
copy DimGeography from 's3://pedro-m3/DimGeography.txt' DELIMITER '|'
credentials
'aws_access_key_id=AKIAJUOL3JIJO3MZZLCA;aws_secret_access_key=VZeCODxBBtpK
A9ACo3Y8GbQyryX9IGcwOleQo0c7';
copy DimOrganization from 's3://pedro-m3/DimOrganization.txt' DELIMITER '|'
credentials
'aws_access_key_id=AKIAJUOL3JIJO3MZZLCA;aws_secret_access_key=VZeCODxBBtpK
A9ACo3Y8GbQyryX9IGcwOleQo0c7';
copy DimProduct from 's3://pedro-m3/DimProduct.txt' DELIMITER '|' credentials
'aws_access_key_id=AKIAJUOL3JIJO3MZZLCA;aws_secret_access_key=VZeCODxBBtpK
A9ACo3Y8GbQyryX9IGcwOleQo0c7';
copy DimProductCategory from 's3://pedro-m3/DimProductCategory.txt' DELIMITER
'|' credentials
'aws_access_key_id=AKIAJUOL3JIJO3MZZLCA;aws_secret_access_key=VZeCODxBBtpK
A9ACo3Y8GbQyryX9IGcwOleQo0c7';
copy DimProductSubcategory from 's3://pedro-m3/DimProductSubcategory.txt'
DELIMITER '|' credentials
'aws_access_key_id=AKIAJUOL3JIJO3MZZLCA;aws_secret_access_key=VZeCODxBBtpK
A9ACo3Y8GbQyryX9IGcwOleQo0c7';
```

```

copy DimPromotion from 's3://pedro-m3/DimPromotion.txt' DELIMITER '|' credentials
'aws_access_key_id=AKIAJUOL3JIJO3MZZLCA;aws_secret_access_key=VZeC0DxBBtpK
A9ACo3Y8GbQyryX9IGcwOleQo0c7';
copy DimReseller from 's3://pedro-m3/DimReseller.txt' DELIMITER '|' credentials
'aws_access_key_id=AKIAJUOL3JIJO3MZZLCA;aws_secret_access_key=VZeC0DxBBtpK
A9ACo3Y8GbQyryX9IGcwOleQo0c7';
copy DimSalesReason from 's3://pedro-m3/DimSalesReason.txt' DELIMITER '|'
credentials
'aws_access_key_id=AKIAJUOL3JIJO3MZZLCA;aws_secret_access_key=VZeC0DxBBtpK
A9ACo3Y8GbQyryX9IGcwOleQo0c7';
copy DimSalesTerritory from 's3://pedro-m3/DimSalesTerritory.txt' DELIMITER '|'
credentials
'aws_access_key_id=AKIAJUOL3JIJO3MZZLCA;aws_secret_access_key=VZeC0DxBBtpK
A9ACo3Y8GbQyryX9IGcwOleQo0c7';
copy DimScenario from 's3://pedro-m3/DimScenario.txt' DELIMITER '|' credentials
'aws_access_key_id=AKIAJUOL3JIJO3MZZLCA;aws_secret_access_key=VZeC0DxBBtpK
A9ACo3Y8GbQyryX9IGcwOleQo0c7';
copy FactCallCenter from 's3://pedro-m3/FactCallCenter.txt' DELIMITER '|' credentials
'aws_access_key_id=AKIAJUOL3JIJO3MZZLCA;aws_secret_access_key=VZeC0DxBBtpK
A9ACo3Y8GbQyryX9IGcwOleQo0c7';
copy FactCurrencyRate from 's3://pedro-m3/FactCurrencyRate.txt' DELIMITER '|'
credentials
'aws_access_key_id=AKIAJUOL3JIJO3MZZLCA;aws_secret_access_key=VZeC0DxBBtpK
A9ACo3Y8GbQyryX9IGcwOleQo0c7';
copy FactFinance from 's3://pedro-m3/FactFinance.txt' DELIMITER '|' credentials
'aws_access_key_id=AKIAJUOL3JIJO3MZZLCA;aws_secret_access_key=VZeC0DxBBtpK
A9ACo3Y8GbQyryX9IGcwOleQo0c7';
copy FactInternetSales from 's3://pedro-m3/FactInternetSales.txt' DELIMITER '|'
credentials
'aws_access_key_id=AKIAJUOL3JIJO3MZZLCA;aws_secret_access_key=VZeC0DxBBtpK
A9ACo3Y8GbQyryX9IGcwOleQo0c7';
copy FactInternetSalesReason from 's3://pedro-m3/FactInternetSalesReason.txt'
DELIMITER '|' credentials
'aws_access_key_id=AKIAJUOL3JIJO3MZZLCA;aws_secret_access_key=VZeC0DxBBtpK
A9ACo3Y8GbQyryX9IGcwOleQo0c7';
copy FactResellerSales from 's3://pedro-m3/FactResellerSales.txt' DELIMITER '|'
credentials
'aws_access_key_id=AKIAJUOL3JIJO3MZZLCA;aws_secret_access_key=VZeC0DxBBtpK
A9ACo3Y8GbQyryX9IGcwOleQo0c7';
copy FactSalesQuota from 's3://pedro-m3/FactSalesQuota.txt' DELIMITER '|'
credentials
'aws_access_key_id=AKIAJUOL3JIJO3MZZLCA;aws_secret_access_key=VZeC0DxBBtpK
A9ACo3Y8GbQyryX9IGcwOleQo0c7';
copy FactSurveyResponse from 's3://pedro-m3/FactSurveyResponse.txt' DELIMITER '|'
credentials

```

```
'aws_access_key_id=AKIAJUOL3JIJO3MZZLCA;aws_secret_access_key=VZeCODxBBtpK
A9ACo3Y8GbQyryX9IGcwOleQo0c7';
copy ProspectiveBuyer from 's3://pedro-m3/ProspectiveBuyer.txt' DELIMITER '|'
credentials
'aws_access_key_id=AKIAJUOL3JIJO3MZZLCA;aws_secret_access_key=VZeCODxBBtpK
A9ACo3Y8GbQyryX9IGcwOleQo0c7';
```

Annex J

Program written in Python to upload files into Azure Blob Storage

```
from datetime import datetime
import glob
import os
from azure.storage.blob import ContentSettings
from azure.storage.blob import BlockBlobService

for filename in glob.iglob('/Users/pedroferreira/Desktop/RS/*.txt'):
    startTime= datetime.now()
    block_blob_service = BlockBlobService(account_name='discomestrado1',
account_key='bVG924bY/CineMZKyBw1IWBt4ZxGNKHimetqd+yu6bzNIUcEJBFNWUYdfHQ3A37
Zp6IuHLxpLLUF7+8ST6c/KA==')

    block_blob_service.create_blob_from_path(
        'pedro-m3',
        filename,
        filename,
        content_settings=ContentSettings(content_type='text')
    )
    timeElapsed=datetime.now()-startTime
    print(filename)
    print('Time elapsed (hh:mm:ss.ms) {}'.format(timeElapsed))
```


Annex K

Web Service source code

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Runtime.Serialization;
using System.ServiceModel;
using System.ServiceModel.Web;
using System.Text;

namespace WCFCloud
{
    [ServiceContract]
    public interface IService1
    {
        [OperationContract]
        string GetDataFromRedShift();

        [OperationContract]
        string GetDataFromAzureSQL();
    }
}
```

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Runtime.Serialization;
using System.ServiceModel;
using System.ServiceModel.Web;
using System.Text;
using System.Data;
using System.Data.Odbc;
using System.Data.SqlClient;

namespace WCFCloud
{
    public class Service1 : IService1
    {
        private string query = "select distinct City, StateProvinceName,
DimProductCategory.EnglishProductCategoryName from FactSurveyResponse,
DimCustomer, DimProductCategory, DimDate, DimGeography where
FactSurveyResponse.CustomerKey = DimCustomer.CustomerKey and
FactSurveyResponse.ProductCategoryKey = DimProductCategory.ProductCategoryKey
and FactSurveyResponse.DateKey = DimDate.DateKey and DimCustomer.GeographyKey =
DimGeography.GeographyKey and CountryRegionCode = 'DE' and CalendarYear = 2002
and CalendarQuarter = 4 order by StateProvinceName, City,
EnglishProductCategoryName";
        private string query1 = "select count(*) from FactFinance where Amount >
(select avg(amount) from FactFinance)";
        private string query2 = "select p.PromotionKey, p.EnglishPromotionName,
SalesAmount from DimPromotion p, (select s.PromotionKey, sum(SalesAmount)";
    }
}
```

```
SalesAmount from FactInternetSales s where PromotionKey != 1 group by
s.PromotionKey) s where p.PromotionKey = s.PromotionKey order by
p.PromotionKey";
```

```
public string GetDataFromRedShift()
{
    DataSet ds = new DataSet();
    DataTable dt = new DataTable();
    DataSet ds1 = new DataSet();
    DataTable dt1 = new DataTable();
    DataSet ds2 = new DataSet();
    DataTable dt2 = new DataTable();

    string server = "pedro-mestrdo.cbtwsdsybaam.us-east-
2.redshift.amazonaws.com";
    string port = "5439";

    string userName = "pedro";
    string userPassword = "*****";

    // Database name
    string DBName = "dw";

    try
    {
        string connString = "Driver={Amazon Redshift (x64)};" +
String.Format("Server={0};Database={1};" +
"UID={2};PWD={3};Port={4};SSL=true;Sslmode=Require",
server, DBName, userName,
userPassword, port);

        OdbcConnection conn = new OdbcConnection(connString);
        conn.Open();

        string sql = query;
        OdbcDataAdapter da = new OdbcDataAdapter(sql, conn);
        da.Fill(ds);
        dt = ds.Tables[0];
        foreach (DataRow row in dt.Rows)
        {

        }

        sql = query1;
        OdbcDataAdapter da1 = new OdbcDataAdapter(sql, conn);
        da1.Fill(ds1);
        dt1 = ds1.Tables[0];
        foreach (DataRow row in dt1.Rows)
        {

        }
        conn.Close();
    }
    catch (Exception ex)
    {
        Console.Error.WriteLine(ex.Message);
    }

    return "OK";
}
```

```

public string GetDataFromAzureSQL()
{
    try
    {
        SqlConnectionStringBuilder builder
= new SqlConnectionStringBuilder();
        builder.DataSource = "pedro-dw.database.windows.net";
        builder.UserID = "mestrado";
        builder.Password = "*****";
        builder.InitialCatalog = "mestrado";

        using (SqlConnection connection
= new SqlConnection(builder.ConnectionString))
        {
            connection.Open();

            String sql = query;

            using (SqlCommand command = new SqlCommand(sql, connection))
            {
                using (SqlDataReader reader = command.ExecuteReader())
                {
                    while (reader.Read())
                    {

                    }
                }
            }

            sql = query1;

            using (SqlCommand command = new SqlCommand(sql, connection))
            {
                using (SqlDataReader reader = command.ExecuteReader())
                {
                    while (reader.Read())
                    {

                    }
                }
            }

            sql = query2;

            using (SqlCommand command = new SqlCommand(sql, connection))
            {
                using (SqlDataReader reader = command.ExecuteReader())
                {
                    while (reader.Read())
                    {

                    }
                }
            }
        }
    }
    catch (SqlException e)
    {

```

```
        throw e;
    }
    return "OK";
}
}
```

Additional configuration described in (“Install and Configure the Amazon Redshift ODBC Driver on Microsoft Windows Operating Systems - Amazon Redshift,” n.d.)

Annex L

SQL queries

-- Total sales per promotion

```
select
  p.PromotionKey, p.EnglishPromotionName, SalesAmount
from
  DimPromotion p,
  (select s.PromotionKey, sum(SalesAmount) SalesAmount from FactInternetSales s
  where PromotionKey != 1 group by s.PromotionKey) s
where
  p.PromotionKey = s.PromotionKey
order
  by p.PromotionKey;
```

-- The number of finance records where the amount is more than the average amount

```
select
  count(*)
from
  FactFinance
where
  Amount > (select avg(amount) from FactFinance);
```

-- Product categories for surveys received from German customers in 4th quarter
-- of 2002

```
select
  distinct
    City,
    StateProvinceName,
  DimProductCategory.EnglishProductCategoryName
from
  FactSurveyResponse, DimCustomer, DimProductCategory, DimDate, DimGeography
where
  FactSurveyResponse.CustomerKey = DimCustomer.CustomerKey and
  FactSurveyResponse.ProductCategoryKey =
  DimProductCategory.ProductCategoryKey and
  FactSurveyResponse.DateKey = DimDate.DateKey and
  DimCustomer.GeographyKey = DimGeography.GeographyKey and
  CountryRegionCode = 'DE' and
```

```
CalendarYear = 2002 and  
CalendarQuarter = 4  
order by  
StateProvinceName, City, EnglishProductCategoryName;
```

```
-----  
-- Geographies for each customer, including geographies without any customer  
-----
```

```
select  
  *  
from  
  DimGeography g  
  left join DimCustomer c  
    on g.GeographyKey = c.GeographyKey;
```

```
-----  
-- Top 3 quotas applied to every sales person  
-----
```

```
select  
  FirstName, LastName, SalesAmountQuota  
from  
  DimEmployee, (select top 3 SalesAmountQuota from FactSalesQuota order by  
SalesAmountQuota desc) t2  
where  
  SalesPersonFlag = 1  
order by  
  LastName, FirstName, SalesAmountQuota desc;
```