COMPARISON OF ETL EXECUTION SPEEDS WHEN USING SQL CODE AND WHEN USING SSIS BUILT IN COMPONENTS

Slaviša Vučetić[1], Slaviša Ilić[2], Slavica Savić[1] , Siniša Ilić[1]

[1] *Faculty of Technical Sciences, University of Priština in Kosovska Mitrovica*
[2] *University Singidunum, Belgrade*

Abstract

In this paper, we compare the execution speed of the ETL (Extract, Transform and Load) process, when it is implemented using SQL language code in the first case, and when it is implemented using the ETL tool SSIS (SQL Server Integration Services) in the second case. In the first case we present an example of how to extract data and perform certain transformations to data from tables of the source database by direct executing SQL code, and load the transformed data into a destination database. In the second case, we extract and transform the same data, but using the built in components of the SSIS ETL tool without the need to write any command. The main goal of the paper is to compare the execution speeds for running ETL job in both ways, and to obtain the order of magnitude of time needed to transfer and transform data from OLTP source to OLAP destination systems.

Keywords: ETL, OLTP, OLAP, SSIS

INTRODUCTION

Database technology is at the centre of many types of information systems [1]. Since the 1960s, various approaches have been developed for such systems, which have become known under many different names such as Management Information Systems (MIS), Decision Support Systems (DSS) or Executive Information Systems (EIS) [2]. The afore-mentioned Information Systems contain the database to save and retrieve data. The job of earlier on-line operational systems was to perform transaction and query processing [3], and that was the reason they were named OLTP - Online Transactional Processing systems. These systems are optimised for data entry by a large number of users, and they usually have a huge number (millions or even billions) of records.

Managers of the business areas that were using such systems needed complex data analyses (that could be retrieved from the systems by executing complex aggregation queries) in order to manage own business. Running such queries couldn't be executed in parallel with data entry without jeopardising the system performances, so people decided to create new type of the system specialised for running complex queries that would enable data analysis and brining important business decisions. Such systems can organize and present data in various formats in order to accommodate the diverse needs of the different users [3], and such systems need much less time to generate complex reports [4]. This type of system is OLAP (Online Analytical Processing) system that is optimised for on-line analytical data processing. Since the OLAP functionality and analysis is based on aggregation functions [5], these systems are suitable for quick generation of complex aggregation reports.

Manual data entry is not an option for loading the data to OLAP systems. The reason for quick execution of complex queries in OLAP systems is the way how data are stored and prepared for a query

execution. Actually, it is the process of extracting data from their source database(s), transforming the format of data and loading this data to destination (OLAP) system called ETL. The ETL (Extract-Transform-Load) processes are responsible for integrating data into a place called data warehouse [6]. A data warehouse is defined as a "subject-oriented, integrated, time variant, non-volatile collection" of data that serves as a physical implementation of a decision support data model and stores the information on which an enterprise needs to make strategic decisions [3]. The first step in the ETL process is to recognise and extract relevant data from the OLTP systems. In the second step appropriate transformations are implemented on extracted data such as: removing wrong data entries, joining data from multiple tables, modification of data, etc. in order to prepare data for faster execution of queries. In third step, the transformed data are loaded to OLAP system. Actually, Extraction-Transformation-Loading (ETL) tools are pieces of software responsible for the extraction of data from several sources, their cleansing, customization and insertion into a data warehouse [7]. By transforming data from source (OLTP) systems to OLAP systems, the query execution time in OLAP system is less because the number of single or multiple join operations and other backup operations is smaller.

When data are stored in OLAP systems, IT Engineers can execute complex – aggregation queries by writing SQL or MDX (programming language) code. But, business analysts – the resources that run data analysis do not write programming language codes. They use Business Intelligence (BI) tools - applications. In short, BI may be defined as the process that helps to highlight the most pertinent information from a vast amount of data [8]. Based on obtained analysis results, management of the companies can make important business decisions and marketing units can run activities for promotion of appropriate products. Actually, BI integrates products, technology, and processes to arrange the crucial data that management should use to boost the performance and revenues of the company [8].

When the amount of data to be transferred from OLTP to OLAP systems is huge, the ETL processes can take a time and they are usually executed out of working hours, and that is the reason the data in OLAP system are a day late compared to data in OLTP system. Since nowadays the business decisions need to be obtained frequently, it is important to OLTP and OLAP systems must be frequently be synchronised without any delay.

The main goal of the paper is to measure the order of magnitude of time needed to execute ETL process that extract, transform and load data from relatively huge OLTP database to OLAP database using modern ETL applications. Since the execution time depends on performance of computers where systems (OLTP system, OLAP system and ETL application) are installed, we measured the ETL execution time on several computers.

THE EXPERIMENT SETUP

We created the VMart OLTP database from the VMart OLAP database [9]. The OLTP database consists of 48 tables, where three tables are with 300.000 records each and four tables are with 5 million records each. Other tables are with 25.000 or less records. Since the complete scheme of the OLTP VMart database is too large, the part of the database scheme with 15 tables is shown in Fig 1. It can be clearly seen that some tables are grouped in order to be transformed in ETL process. The database is installed to SQL Server 2017 standard (DreamSpark Premium | Academic Software Discounts) edition. The SQL Server is installed on MS Server 2016 OS on the VMware virtual machine with 80GB disk space and 8GB RAM memory.

The tables with white background colour in Fig 1 will be transformed to Dimension tables, and tables with yellow (darker)

background colour will be transformed to Fact tables. Dimension tables consists of data related to place, time, and other "code" tables like: products, vendors, shippers, etc. Fact tables contain event (business process) data records: purchase, procurement, lending, payments, etc.

The target database VMartDWH is created with dimension and fact tables organised for OLAP DB. The OLTP Vmart database is our source of data and from its tables data will be extracted, will be transformed and will be loaded to OLAP VMartDWH database using Microsoft ETL application – MS SQL Server Integration Services (SSIS). In the first solution, the ETL process will be executed using queries written in SQL code, and in the second one, the ETL process will be executed using SSIS unit function tools from the toolbox. We will compare speed of ETL process when using two mentioned solutions.

From Fig 1 it can be seen that data from five tables from OLTP DB grouped and with

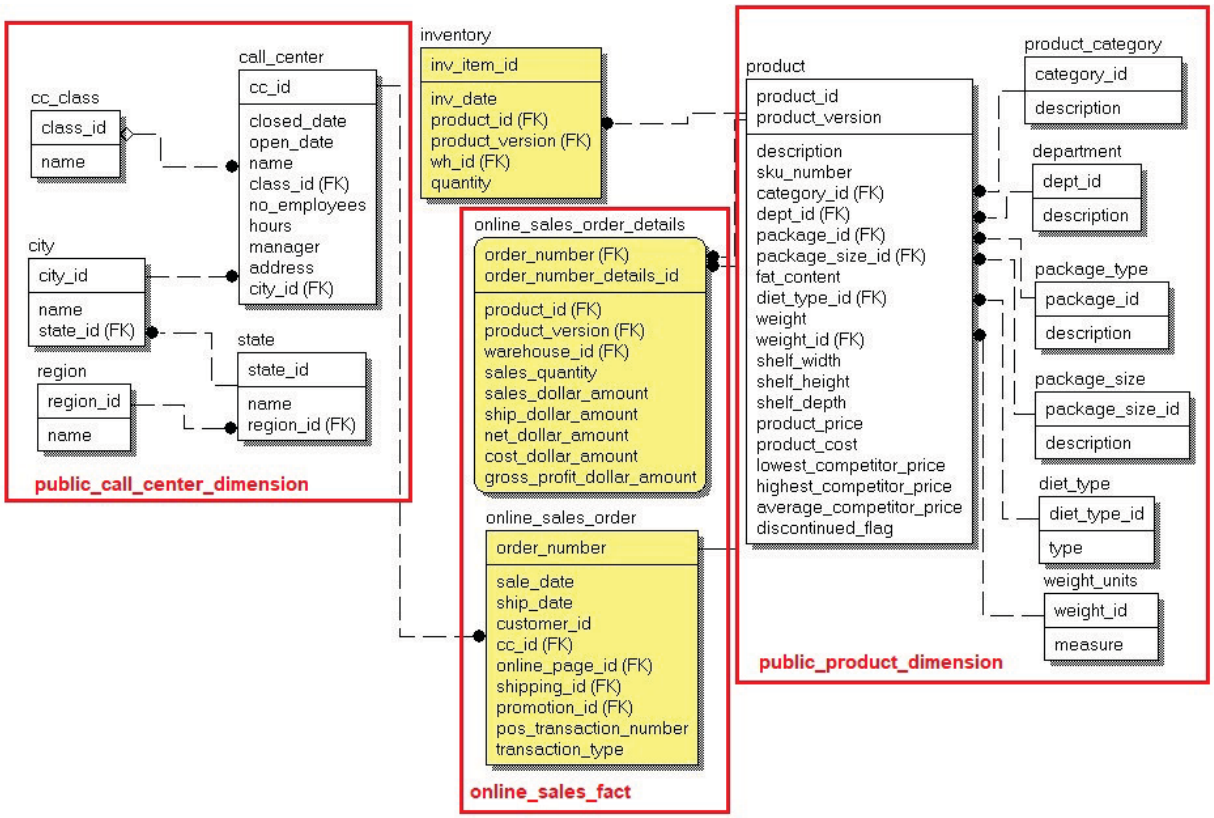


Fig. 1. Part of the VMart OLTP database scheme

relations to the table **call_center** (cc_class, city, state, region) will be merged (transformed) to a single dimension OLAP DB table **public_call_center_dimension**. Another six OLTP tables with relations to the table **product** (product_category, department, package_type, package_size, diet_type, and weight_units) will be also merged to a single OLAP dimension table **public_product_dimension**. The ETL process should extract (get) all data from mentioned groups of tables, perform join operations and load transformed data to appropriate dimension OLAP tables. Our ETL process will transform data from 41 OLTP tables to 11 OLAP dimension tables.

Also, data from four groups of tables similar to tables with yellow (darker) background colour in Fig 1 will be transformed and loaded to four fact tables. From Fig 1 it can be seen that data from two OLTP tables: **online_sales_order** and **online_sales_order_details** will be

transformed to OLAP fact table **online_sales_fact**.

In Fig 2 are presented target OLAP tables obtained from the ETL transformations from the source tables shown in Fig 1. One can see that data in attributes: cc_class, cc_city, cc_state and cc_region in the table **public_call_center_dimension** is extracted from the attributes "name" of the tables: **cc_class**, **city**, **state** and **region** respectively. Similar transformations of data from attributes "description", "type" and "measure" from OLAP tables can be seen in OLAP table **public_product_dimension**. Data from almost all attributes from OLTP tables **online_sales_order** and **online_sales_order _details** are merged into single fact OLAP table **online_sales_fact**.

It is easy to see that data from 15 OLTP tables shown in Fig 1 are transformed to the same amount of data, but organised to 5 OLAP tables in Fig 2.

ETL SOLUTION USING SQL QUERIES

One simple way to create ETL job is to use two SSIS components for loading data to each destination table, one that enable direct writing of SQL code for extraction and transformation of the source data and the second one for mapping columns of obtained SQL queries to the columns of destination table. Those two components: OLE DB Source and OLE DB Destination with the connection between them are shown in Fig 3.

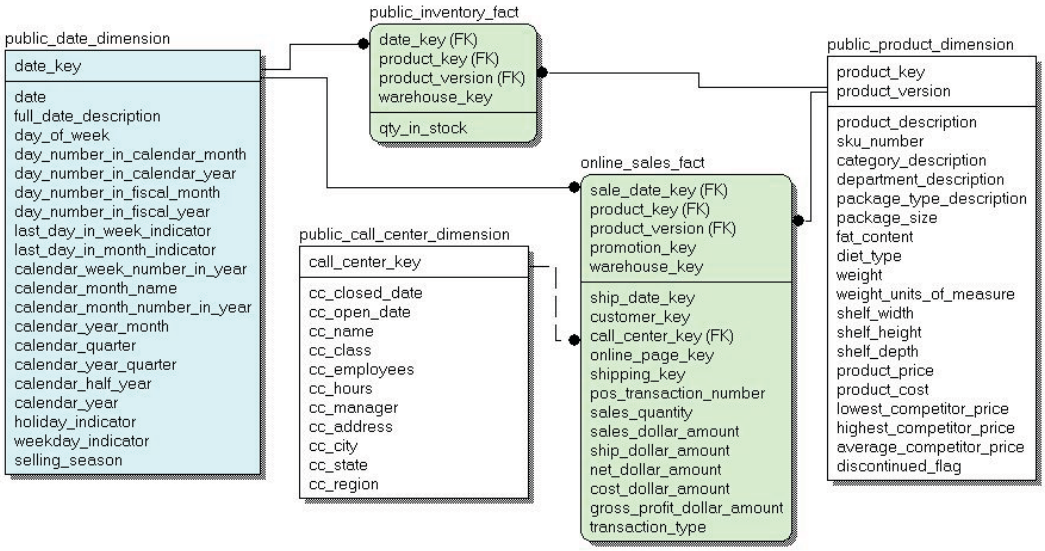


Fig. 2. VMart OLAP tables as ETL transformation of OLTP tables from Fig 1

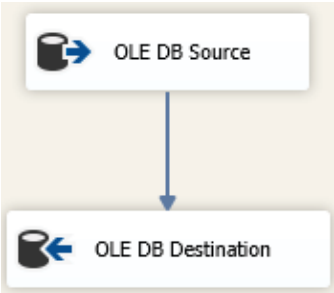


Fig. 3. OLE DB Source and OLE DB Destination components.

The SQL code for extraction and transformation of source data is shown in Fig 4. and it belongs to OLE DB Source component configuration window. The user must select appropriate source database (in this case OLTP DB) first. From the SQL code one can notice that the query is obtained by extracting data from afore-mentioned OLTP tables around **call_center** that are inner joined. The resulting columns

are chosen from the tables that participate in the join process.

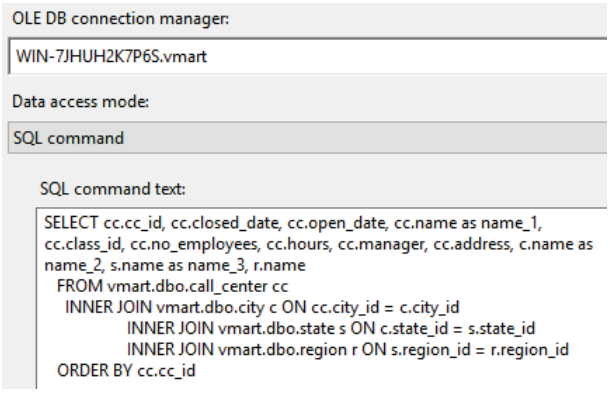
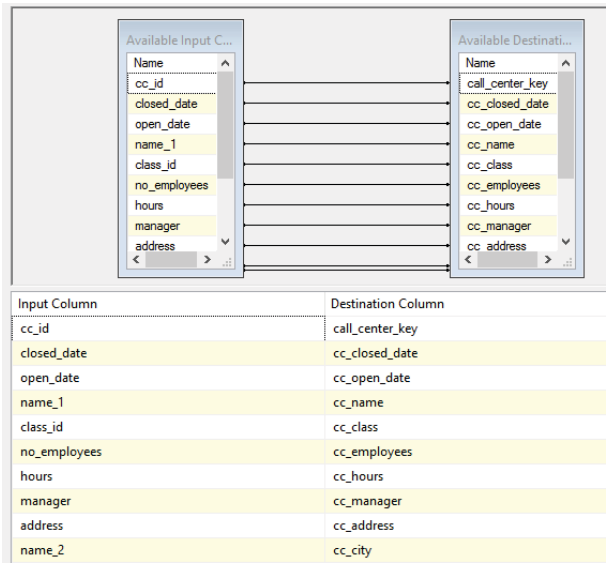


Fig. 4. "OLE DB Source Editor".

In the OLE DB Destination component, the OLAP DB has to be selected as destination DB, and also the destination table where extracted and transformed data are to be loaded. Since the column names and the sequence of columns in the query obtained by SQL command in OLE DB Source component need not to match with those in the destination table, user has to map query columns to the destination columns. In this way appropriate data will be stored in a destination table. The mapping configuration for loading data to **public_call_center_dimension** table is shown in Fig 5. The unit job explained is created for every destination (OLAP) table.



Input Column	Destination Column
cc_id	call_center_key
closed_date	cc_closed_date
open_date	cc_open_date
name_1	cc_name
class_id	cc_class
no_employees	cc_employees
hours	cc_hours
manager	cc_manager
address	cc_address
name_2	cc_city

Fig. 5. Mapping for table public_call_center_dimension.

ETL SOLUTION USING SSIS FUNCTIONAL COMPONENTS

Another way for creating the ETL job is to use unit functionalities of existing ETL application, in this case functionalities of SSIS toolbox. The unit job is created by connecting the SSIS components and by configuring tasks in the components, without need to write SQL or other programming language code. Unlike in previous case, where almost all job is executed in the DataBase Management System (DBMS), in this case data is extracted from DBMS and completely transformed in the ETL application.

In this solution the input for the OLE DB Source component is the source table (not the query). The join process is implemented by connecting two OLE DB source components (that extract data from two tables separately) to Sort components (each source table has to be sorted by column(s) that participate to join process) and then the outputs of Sort components are connected to the input of Merge Join component. When several join operations have to be executed, two Sort and one Merge Join components are needed for each one. The number of the components that can be used in SSIS is large, but we used in our example just the few one.

The components used and connected for extracting, transforming and loading data to the same **public_call_center_dimension** destination table are shown in Fig 7.

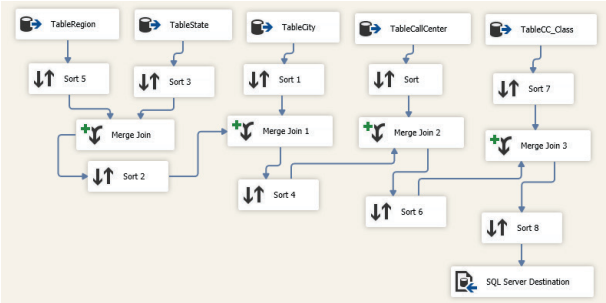


Fig. 7. The SSIS unit job (Data Flow) Editor

In the Fig 7 one can see five OLE DB Source components for five source tables, and four Merge join components for executing inner join operations. Sort

components are mandatory for sorting data before going to Merge join components. Unlike in DBMS, SSIS does not use column indexes that increase the speed of data sorting, thus the sorting speed is lower here.

Configuring of the Sort component can be seen in Fig 8. A user can select which column (or columns) are to be sorted by checking the checkboxes on the left side, and which columns to be passed through the sorting process. Sometimes, only couple of columns are needed to be passed through for the destination table, so user has to be careful in order not to load to much data and increase computer system load.

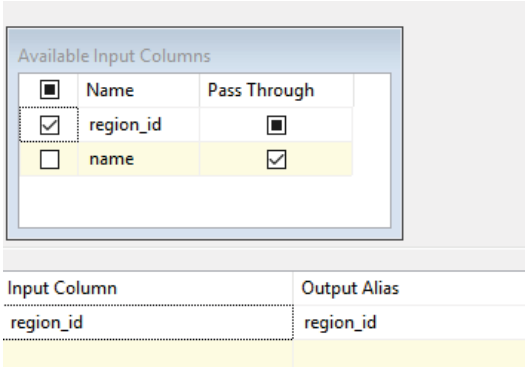


Fig. 8. *Sort Transformation Editor*

The sorted columns at the output of Source components are recognised by Merge Join component (see checkbox on the right side of the input tables in Fig 9), but user has to select which columns need to be passed through at the output of the component by checking the checkboxes on the left side of the input tables.

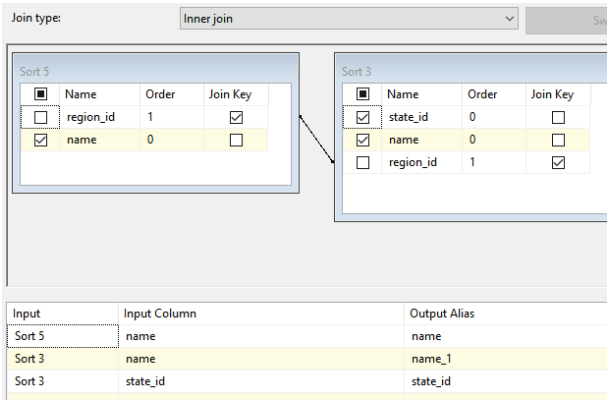


Fig. 9. *Merge Join Transformation Editor*

The last component in the unit job is OLE DB Destination component, where the destination database and destination table has to be selected. By using mapping functionality, the columns at the input of the component are mapped to the destination table columns in the same way as in the previous solution.

RESULTS AND DISCUSSION

ETL process is in our case organised in several tasks: 1) truncating all tables (deleting data from all tables), 2) transforming and loading data to Dimension tables (2 tasks), and 3) transforming and loading data to Fact tables (2 tasks). Out of eleven dimension tables, nine are with 2000 and less records (DimT1), two are with 50.000 and more records (DimT2). Out of four fact tables two are with 300.000 records (FactT1) and two are with 5 million records (FactT2). The time needed for completion of ETL job in three different computers using SQL and SSIS method is presented in Fig 10.

	I7-11800H, 2.3GHz SSD		i7-4700MQ, 2.3GHz, SSD		i7-2600K, 3.7 GHz, HDD	
	SQL	SSIS	SQL	SSIS	SQL	SSIS
truncate	0.079	0.047	0.109	0.094	0.156	0.156
DimT1	1.344	7.345	1.843	10.219	4.516	9.484
DimT2	0.890	3.609	1.344	5.156	2.907	8.39
FactT1	1.907	3.625	3.75	4.812	4.922	9.421
FactT2	43.14	72.516	72.859	84.266	136.094	150.595
Total	**47.36**	**87.142**	**79.905**	**104.547**	**148.595**	**178.046**

Fig. 10. *ETL job duration in seconds*

It is evident that ETL job's duration for mentioned OLAP tables is not so long and is of order of magnitude of couple minutes. Using the SQL code in implementing ETL job is faster around two times than when using the SSIS application if appropriate sorting and merge buffers are set correctly in SSIS. As expected running the job at computers with better processor is faster, but not always. As can be seen from Fig 10, execution time is the slowest when running on HDD and is almost two time slower compared to running on SSD.

CONCLUSION

Transforming and loading data from OLTP to OLAP systems with order of millions of records is executed in couple of minutes using the commercial platform MS SQL and MS SSIS. The duration may vary depending on network speed if two systems are connected by computer network, and on platforms (DBMS and ETL tool) used. We proved that the execution of ETL job is very fast and that business analyst should not hesitate to run ETL job more frequent in order to analyse fresh data.

ACKNOWLEDGMENTS

This work has been supported by the Ministry of Science and Technological Development of Republic of Serbia under Project No. TR-35026

REFERENCES

[1] S. S. Conn, "OLTP and OLAP Data Integration: A Review of Feasible Implementation Methods and Architectures for Real Time Data Analysis", IEEE Southeast Con, Ft. Lauderdale, 8-10 April 2005, 515-520.

[2] O. Azeroual and H. Theel, "The Effects of Using Business Intelligence Systems on an Excellence Management and Decision-Making Process by Start-Up Companies: A Case Study", International Journal of Management Science and Business Administration, Volume 4, Issue 3, March 2018, Pages 30-40.

[3] G. S. Reddy, R. Srinivasu, M. P. Chander Rao, S. R. Rikkula, "Data Warehousing, Data Mining, OLAP and OLTP Technologies Are Essential Elements to Support Decision-Making Process in Industries", International Journal on Computer Science and Engineering Vol. 02, No. 09, 2010, 2865-2873.

[4] S.Ilić, D. Miljković, A. Veljović, S. Obradović, B. Jovanović, "Comparison of Performance in Data Analysis in Dedicated and Traditional DBMS", Proc. of Int. Scientific Conference, Gabrovo, 18 – 19 November 2016.

[5] Hans-J. Lenz, B. Thalheim, "A Formal Framework of Aggregation for the OLAP-OLTP Model", Journal of Universal Computer Science, vol. 15, no. 1 (2009), 273-303.

[6] P. S. Diouf, A. Boly, S. Ndiaye, "Variety of data in the ETL processes in the cloud: State of the art", 2018 IEEE Int. Conference on Innovative Research and Development (ICIRD), Bangkok, Thailand, 2018, pp. 1-5, doi: 10.1109/ICIRD.2018.8376308.

[7] P. Vassiliadis, A. Simitsis, S. Skiadopoulos, "Conceptual modeling for ETL processes", Proc. of the 5th ACM international workshop on Data Warehousing and OLAP, November 2002, pp. 14-21.

[8] Ragazou K, Passas I, Garefalakis A, Zopounidis C (2023) Business intelligence model empowering SMEs to make better decisions and enhance their competitive advantage. Discov Anal 1(2):1–15. https://doi.org/10.1007/s44257-022-00002-3

[9]https://www.vertica.com/docs/10.0.x/HTML/Content/Authoring/GettingStartedGuide/IntroducingVMart/IntroducingVMart.htm