IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 19, Issue 5, Ver. III (Sep.- Oct. 2017), PP 47-53
www.iosrjournals.org

Data Warehouse System for Higher Education Student Information System

Dr.Talib M. J. Al Taleb [1],Mustafa M. K. AL Saedi [2]
[1](*College Of Information Engineering/ Al-Nahrain University, Iraq)*
[2](*Programmer At ICT In The Ministry Of Oil, Iraq)*

Abstract: *this paper presents an architecture for the data warehouse of Higher Education Student Information System (DWHESIS) system as the single source for decision makers to obtain information. The most important motivations to implement this architecture are technology reliability; efficient data report platform and most importantly different data format integration from a different data source in the Higher Education Student Information. The indexed view is a technique utilized to query huge amounts of data. Consequently, the process of developing on-line analytical processing (OLAP) is proposed for decision support to find interesting information from large databases.*

Keywords : *Data warehouse (Star schema), Database of outpatient healthcare, Excel file, Extract Transform and Load (ETL) process, Indexed view, OLAP.*

Date of Submission: 19-09-2017 Date of acceptance: 06-09-2017

I. Introduction

The increased processing power and sophisticated analysis tools have put the solid establishment for the data warehouse to possibly resolve the problem of providing access to all staff in the organizations with the needed information for the necessary in an increasingly competitive world [1] [2]. A data warehouse is an essential part of the overall life cycle intelligence system [3]. An establishment possesses single information distribution center and information markets source their data from the information stockroom. Data warehouse is a central repository that collects data from various sources and formats, cleansed, quality assured, and released only when it is fit for user consumption by ETL (Extract, Transform and Load) process [4] [5]. Data warehouse has both a very detailed level of data and summarize data and it has ability to deal with current and historical data and creating analytical reports. This paper is based on Ralph Kimball's approach. SQL Server R2 is used to design the data warehouse of DWHESIS [6]. The data sources were selected to provide the data warehouse system with data: Higher Education Student Information System (HESIS) system. This paper Contribution is to develop a data warehouse architecture, as shown in figure (1).

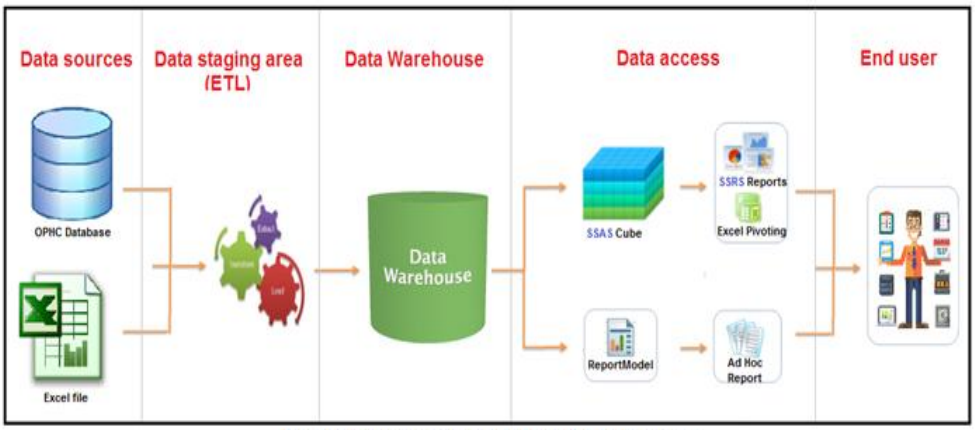

Figure (1) Architecture of the data warehouse

II. Design ETL Process For The Data Warehouse System

This process was designed by using (Visual Studio-Business Intelligence - Integration Service) software [6] [7]. Two packages of SSIS (SQL Server Integration Services) is created. Where ETL process was designed for the source (Database) system . Figure (2) refers to database of HESIS system. Figure (3) refers to ETL process for Database.

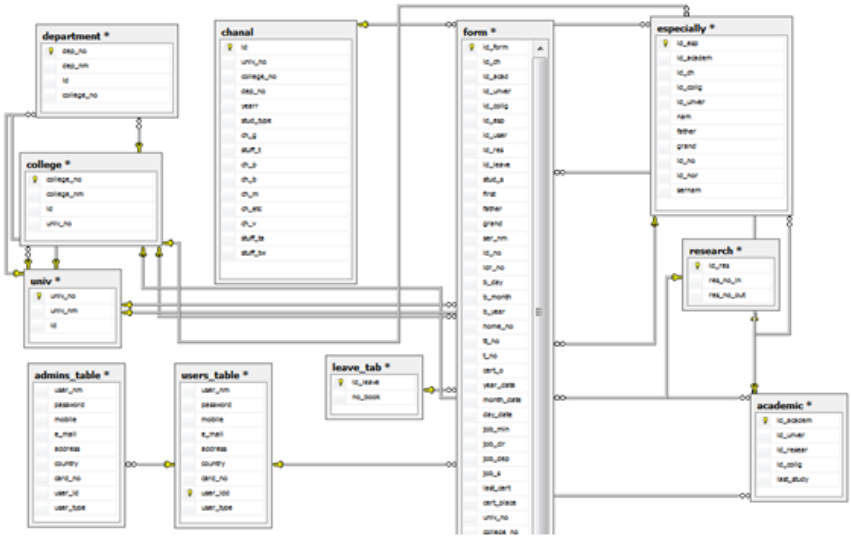

Figure (2) Database of HESIS system

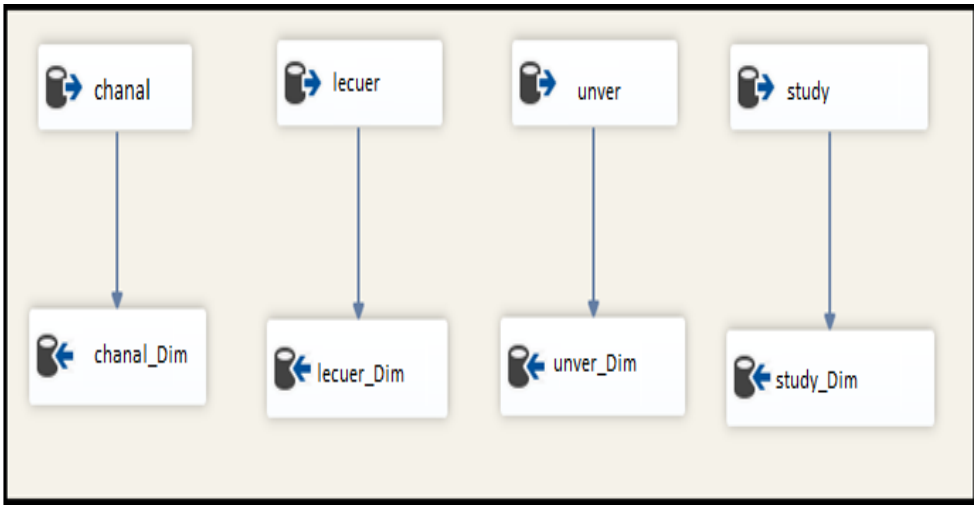

Figure (3) ETL Process for ETL process for Database

III. Dimensional data model

The first step of design the data warehouse is designed a dimensional model [1] [2]. Dimensional modeling is a logical design technique that tries to present the data model in an accessible and intuitive standard frame to provide a foundation for performing sophisticated data analysis. It should be designed to meet requirements of large organizations which must address the needs of managers and decision makers and contains information that can be easily accessible [8]. A fact table is the main table in every dimensional model. It contains measurements of the business and two or more foreign keys that relation to dimension tables. A dimension is one or many table that relation to the fact table. Each dimension contains descriptive textual information and it has a primary key that serves as the basis for referential integrity with fact table to which it is joined. Most dimension tables contain many attributes that contain data gathered from different sources [1] queries, fast aggregations and feeding cubes (OLAP cubes efficiently) [9] [10] [11].The Time dimension is not appeared in the star schema because it was created within (Visual Studio - Business Intelligence- Analysis Service) software [6]. See figure (5).

IV. Star schema

Star is the simplest type of schema, as shown in figure (4) which contains fact table sits in the center and associated with other dimension tables like a star. Each dimension has the primary key that is related to a foreign key in the fact table. There are many reasons lead to adopting this schema, these are:

1- Less complex
2- Easy to understand
3- Most widely used to develop data warehouses
4- More effective schema for handling simpler.

Queries, fast aggregations and feeding cubes (OLAP cubes efficiently) [9] [10] [11]. The Time dimension is not appeared in the star schema because it was created within (Visual Studio - Business Intelligence- Analysis Service) software [6]. See figure (5).

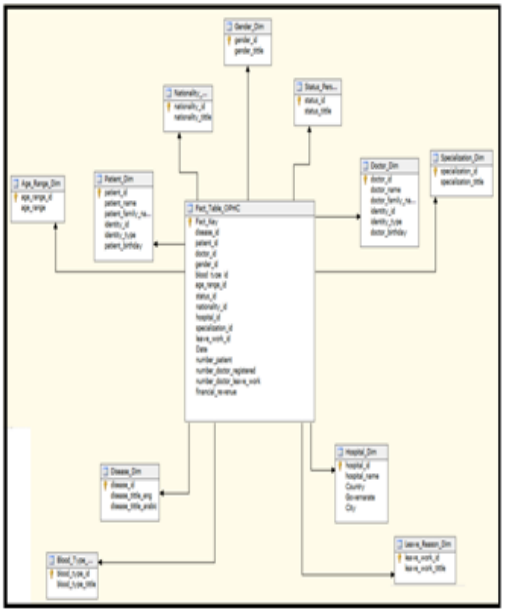

Figure (4) Star Schema

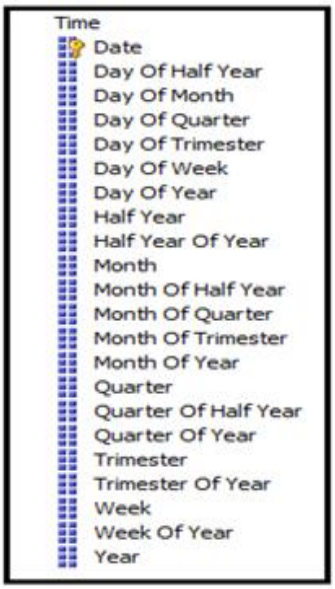

Figure (5) Time dimension

V. Design An Indexed View For The Data Warehouse System (11 Bold)

At the first, this article is SQL Server specific. Indexed views can be a powerful tool that can satisfy a query, then under certain circumstances, this can significantly reduce the amount of work that SQL Server needs to do to return the required data, and so improve query performance and solve the problem of decision support workloads. However, the more use indexed views lead to the more storage space is needed in a data warehouse. At first, view is created to meet the user's require for some important query or this view may be summary of data in the base table. Once the view has been created, a unique clustered indexes is created on it. The first index created on view that must be a unique index and must be clustered and this unique clustered index must be created before any other indexes can be created on the view. Once a unique clustered index is present for a view, further non-clustered indexes may be created. Non-clustered indexes on views can provide additional query performance and provide more options for the query optimizer to choose from during the compilation process. When the first index is added to a view, it is built using a b-tree structure [12]. An identity single column is created on the fact table (Fact _Table _HESIS) and made it a primary key called column Fact_Key. The alternative to a single column primary key is to have multi-column primary key. The primary key is created from a combination of dimension key columns. Usually not all dimension key columns, but only several of them. Only those whose combination enables us to uniquely identify each fact row

An identity single column primary key is better than multi-column primary key because the multi-column primary key may not be unique. Even if the combination of those columns is unique now, it does not guarantee that in the future, the primary key won't be a duplicate. The single-column approach guarantees that the primary key is unique because it is an identity column and this first reason of creating Fact _Key. The second reason of creating Fact _Key to improve insert performance because, in the case of multi-column primary key, SQL Server needs to identify first where to insert that row, based on the clustered column. That row does not automatically go to the end of the table; it goes in the middle of the table depending on the value of the clustered column. SQL Server spends a lot of time to identify where to insert each row. The data type of Fact _Key is a BIGINT.

In data warehouse, the primary consideration is the query performance and secondary consideration is the load performance. Single column primary key is better when need to balance between loading and query performance but multi-column primary key is slower loading and faster query than single column primary key.A

common way of indexing a fact table is by putting a clustered index on the fact key column, then put a non-clustered index on each of the dimension key columns. This will allow each of the non-clustered indices to be efficiently used, whilst ensuring that they are slim because, in SQL Server, a non-clustered index uses the clustered index key as the row locator. They are efficiently used because they are built on each of the dimension keys. Each of them as the first index key not as a second or third index keyThis ensures that they will be hit by the queries using that dimension key, because they are the first column in the index key.

Three indexed views created to improve the performance of the main queries that decision makers are asking about them constantly, which are:
1- Plan _VW 2- leave _VW 3- Introduction _ VW Each index (Clustered or Non-clustered) occupies from storage about 0.180 MB where row count = 2676 records related to doctors and the size increases with increasing the number of records. Note the elapsed time can be affected by several factors such as load on the server, input/output load, network bandwidth between server and client.

Table (1) compares the performance of regular view and indexed view of Plan _vw where default query and row count= 36633, Number of Threads=200, Number of Iterations=50.

Table (1) compares between regular view and indexed view of Plan _vw

Function	Regular view	Indexed view (ClusteredIndex(Fact _Key) only)	Indexed view (Clustered index (Fact_Key) with (Non-Clustered Index (Plan _id))
Elapsed Time	00:04:35.7427	00:03:45.2783	00:03:05.1483
Client Seconds /Iteration (AVG)	3.5339	2.6012	1.3110
Logical Reads/Iteration (AVG)	1283	222	222
CPU Seconds/Iteration (AVG)	0.0436	0.0344	0.0342
Actual Seconds/Iteration (AVG)	3.2672	2.4011	1.2163
Current space	Default	1.734 MB	2.609 MB

Table (2) compares the performance of regular view and indexed view of leave_vw where default query and row count= 798, Number of Threads=200, Number of Iterations=50.

Table (2) compare regular view and indexed view of leave _vw

Function	Regular view	Indexed view (Clustered Index(Fact_Key))	Indexed view (Clustered index (Fact_Key) with (Non-Clustered Index (specialization_id))	Indexed view (Clustered index (Fact_Key) with (1- Non-Clustered Index (specialization_id) 2- Non-Clustered Index(leave l_id))
Elapsed Time	00:01:14.5857	00:00:12.1886	00.00.07.0494	00:00:10.2889
Client Seconds /Iteration (AVG)	0.4119	0.0327	0.0148	0.0308
Logical Reads/Iteration (AVG)	1283	7	7	7
CPU Seconds/Iteration (AVG)	0.0187	0.0009	0.0010	0.0008
Actual Seconds/Iteration (AVG)	0.3315	0.0108	0.0023	0.0060
Current space	Default	0.055 MB	0.086 MB	0.125MB

VI. Data Access (OLAP)

A data warehouse is a place to store data with a design that makes analyzing data easier, and OLAP is a method to analyze multidimensional data as well as to provide self-service business intelligence capabilities to decision-makers to gain insight into data through fast, consistent, interactive access to a wide range of different views of information. The feature is essential of OLAP is “a multidimensional analysis”. In other words, the ability to analyze metrics in different dimensions such as time, geography, disease, gender, nationality, etc. [13] [14].

OLAP can build many OLAP cubes with many dimensions. An OLAP cube is a data structure that improved performance through overcoming the limitations of relational databases by providing fast analysis answers to complex queries and to find interesting information from the data warehouse. A cube is a set of related measures and dimensions. A dimension is a group of attributes that represent an area of interest related to the measures in the cube, and which are used to analyze the measures in the cube.

OLAP consists of three basic analytical operations:

1- Roll up
2- Drill down
3- Slice and dice.

Thus, the data can be rolled up, sliced, and diced as needed to handle the widest variety of questions that are relevant to a user’s area of interest [1] [3]. Physical storage options affect the performance and storage requirements for cubes. The cube can be stored in an MOLAP (multidimensional OLAP) structure, an ROLAP (relational OLAP) database, or an HOLAP (hybrid OLAP) combination of multidimensional structure and relational database. Visual Studio (Business Intelligence- Analysis Service) software adopted to create some OLAP cubes that meet the decision makers' queries [6]. Figure (6) illustrate example about how to use the operations of OLAP Cube, it shows the number of students applicants, successful and accepted for the specialization of computer science for the year 2009 for all universities.

year- month
January 2009

Universities	No.S. Applicants	No.S. Successful	Males No.S. Successful	Female No.S. Successful	Males No.S. Accepted	Female No.S. Accepted	seats
Al Mustansiriya	650	550	400	150	100	25	125
Baghdad	800	600	400	200	50	100	150
Nahrin	500	450	300	150	75	50	125
Technoloev	1000	700	300	400	100	100	200
Karbal	400	300	200	100	100	25	125
Anbar	200	175	100	75	75	25	100
Diyala	400	350	200	150	50	25	75
Basra	500	400	250	150	100	100	200
Nasirivah	300	250	200	50	50	20	70
Karbala	200	195	120	75	60	20	80
Naja	150	140	100	40	50	25	75
Kufa	300	250	175	75	50	25	75
Grand Total	5400	4360	2745	1615	860	540	1400

Figure (6) Result of OLAP Cube

VII. User Interface

This is the last stage in the process of construct the data warehouse (DWOP) system, during this stage, interfaces were designed which enable the decision makers to reach the required reports easily. ASP.NET 4.5, C#, HTML, CSS and JavaScript languages are used to design the interface pages. Figure (7) shows the main interface of the DWOP system and has called the login page. This page is designed to allow decision makers to enter the page of reports. SQL Server Report Services and Excel Power Pivot used to show reports [6].

Figure (8) refers to report which is designed by using SSRS. Figure (9) refers to chart which is designed by using Microsoft Excel 2013 Pivot Table.

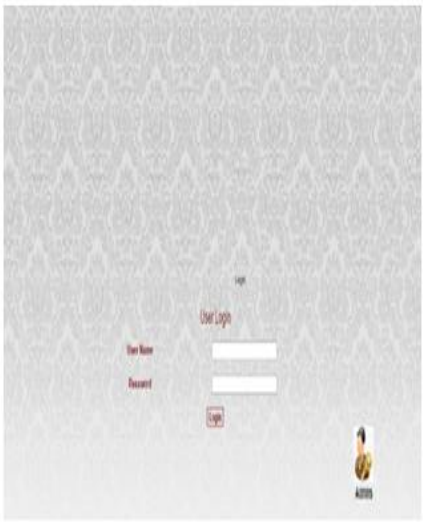

Figure (7) Main page of DWOP system

Report
Master and PhD

2011
science

Universities	Master	PhD
Baghdad		
chamistry	12	4
physics	5	1
AL-Mustansiriyah		
computer	10	3
atmospheric	10	2

Figure (8) Report about Disease

VIII. Conclusion

A data warehouse (DWHESIS) system has been built for an outpatient healthcare environment that collects data from various sources and stores in a single repository, and solved the problems faced by decision makers in a Higher Education Student Information, and improves speed to answer important queries needed by decision makers. Also, advanced analysis tools are utilized to provide an analytical information for decision makers from a different point of view and designed a reporting interfaces.

References

[1]. Ralph Kimball, Margy Ross, Warren Thornthwaite, Joy Mundy, Bob Becker, "The Data Warehouse Lifecycle Toolkit: Practical Techniques for Building Data Warehouse and Business Intelligence Systems", Second Edition John Wiley & Sons, Inc., 2008.
[2]. K. Laudon, J. Laudon, "Management Information Systems: Manage the Digital Firm", Twelfth Edition, Pearson Education, Inc., United States of America, 2012.
[3]. Paulraj Ponniah, "Data Warehousing Fundamentals", John Wiley & Sons, Inc. 2001.
[4]. Lekha Narra, Tony Sahama , Peta Stapleton, "Clinical Data Warehousing A Business Analytics approach for managing health data", Proceedings of the 8th Australasian Workshop on Health Informatics and Knowledge Management (HIKM 2015), Pages 101 104, Sydney, Australia, January 2015.
[5]. Ralph Kimball, Margy Ross, "The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling", Third Edition, John Wiley & Sons, Inc., Indiana, United States of America, 2013.
[6]. R. Mistry, W. Misner, "Introducing Microsoft SQL Server 2012",
[7]. First Edition, Microsoft Press A Division of Microsoft Corporation, United States of America, 2012.
[8]. Warren Thornthwaite, "Implementing a Microsoft SQL Server Parallel Data Warehouse Using the Kimball Approach", Copyright Microsoft Corporation, United States of America, June 2011.
[9]. MS. ALPA R. PATEL; PROF. (DR.) JAYESH M. PATEL, "DATA MODELING TECHNIQUES FOR DATA WAREHOUSE", International Journal of Multidisciplinary Research, Vol.2, Issue 2, Pages 240-246, February 2012.
[10]. Rajib Dutta, "Healthcare Data Warehouse System Architecture for Influenza (FLU) Diseases", Computer Science & Information Technology Journal, Vol. 3, No. 2, Pages 77-89, March 2013.
[11]. Dr. Osama E.Sheta and Ahmed Nour Eldeen, "BUILDING A HEALTH CARE DATA WAREHOUSE FOR CANCER DISEASES", International Journal of Database Management Systems (IJDMS) Vol.4, No.5, Page 39 – 46, October 2012.
[12]. TEH YING WAH, ONG SUAN SIM, "Development of a Data Warehouse for Lymphoma Cancer Diagnosis and Treatment Decision Support", Journal WSEAS TRANSACTIONS on INFORMATION SCIENCE and APPLICATIONS Vol. 6, Issue 3, Pages 530 -543, March 2009.
[13]. "SQL Server to SQL Server PDW Migration Guide (AU3)", Copyright Microsoft Corporation 2014.
[14]. Dr. Walid Qassim Qwaider, "Apply On-Line Analytical Processing (OLAP) With Data Mining for Clinical Decision Support", International Journal of Managing Information Technology (IJMIT) Vol.4, No.1, Pages 25 - 37, February 2012.
[15]. N.Colossi, W.Malloy, B.Reinwald, "relational extensions for OLAP", IBM systems journal, Vol. 41, No. 4, Pages 714-731, 2002.

IOSR Journal of Computer Engineering (IOSR-JCE) is UGC approved Journal with Sl. No. 5019, Journal no. 49102.

Dr.Talib M. J. Al Taleb . "Data Warehouse System for Higher Education Student Information System." IOSR Journal of Computer Engineering (IOSR-JCE), vol. 19, no. 5, 2017, pp. 47–53.