

CS683-2001B-01 – Data Warehouse

Data Warehouse Design Document: Angel Children's Hospital

[STUDENT NAME]

[CURRENT DATE]

Table of Contents

- Data Warehouse Requirements..... 4
 - Company Outline – Angel Children’s Hospital..... 4
 - Data Acquisition 5
 - Production Data Sources 5
 - External Data 6
 - Data Staging Area 6
 - Data Storage..... 7
 - Data Warehouse and Components 7
 - Data Marts 8
 - Meta Data 8
 - Information Delivery 9
 - Multidimensional Database and OLAP 9
 - Data Mining 9
 - Report and Query 10
 - Data Management and Control 10
- Design Requirements 11
 - Data Warehouse Design Requirements 11
 - Conceptual Design 11
 - Logical Design 12
 - Physical Design..... 13
- Load Data..... 14
 - ACH Data Load Analysis 14
 - The Data Sources 14
 - ACH Source Data 14
 - Phynd Source Data..... 15
 - Data Transformation 16
 - ETL Completion 17
 - ACH Data Warehouse: ACH_PatientDW1 17
- Data Analysis 19
 - ACH and Data Analytics 19
 - Defining OLAP 19
 - OLAP Main Types and Processing Tools..... 20

- ACH and OLAP Analytical Tools 21
- Maintenance, SQL Script and Conclusion..... 23
 - ACH Data Maintenance Plan..... 23
 - Database Monitoring and Performance Tuning 23
 - Backup and Recovery Plans..... 24
 - SQL Data Warehouse Scripting..... 26
 - Data Warehouse Creation Script 26
 - Data Warehouse Table Creation Scripts 26
 - Referential Key Creation Scripts..... 29
- In Closing..... 30
- References 31

Data Warehouse Requirements

Company Outline – Angel Children's Hospital

Angel Children's Hospital (ACH) is a fictitious children's hospital that is based out of Columbus, Ohio. The hospital is a non-profit organization and its payer-mix comes from the following multiple areas: direct pay, medical insurance coverage, government grants, and programs, as well as fundraising and donations. The hospital consists of the main campus located in the heart of Columbus and four smaller satellite campuses in the greater surrounding Columbus area. The main hospital is a 500-bed facility, supporting full pediatric care which includes a full Lab and Testing Center, Surgery Center and Pharmacy, as well as departments for Ambulatory Care, Cardiology, Pulmonary, Orthopedic, and Mental Health. Each of ACH's satellite facilities has an ambulatory care department, specialty care departments, full lab and testing centers and a pharmacy.

Angel Children's Hospital employs over 5100 regular employees and has a medical professional staff that consists of over 300 surgeons, physicians, and residents. The hospital is also proud to support over 1200 part-time volunteers. ACH also contracts with different third-party vendors to help support their applications and to ensure their facilities and assets are protected and stay secure.

Due to the sensitive data the hospital houses, there are certain rules and regulations the hospital must remain compliant with. "Health Insurance Portability and Accountability Act (HIPAA)" is the regulatory standard that drives both the privacy and protection of an individual's health information (HIPAA 101, 2020). This includes patients as well as hospital employees and volunteers. ACH must also remain PCI compliant since they hold information dealing with credit card payments.

Data Acquisition

Upper management has stressed the need for creating a single reporting environment that contains related information from the various systems used by ACH. The decision has been made to move forward with creating a data warehouse. The initial phase of this environment is to contain information dealing with Patients and their attending physicians, as well as staff tracking information.

The data warehouse for ACH is to contain two main types of data sources: Production data and External Data. Both types of environments will have data extraction processes (ETL) where information will be extracted nightly, data transformations implemented, and then the data will be loaded into various staging tables of a Staging Database. Full details outlining this environment are discussed in later sections.

Production Data Sources

ACH currently utilizes different systems to support the different areas of care for their patients as well as track information on their employees and volunteers. ACH uses Epic as their patient's electronic medical record (EMR) source system. This system is the main source for patient demographics, ambulatory care information which includes diagnosis, prognosis, surgical information, lab tests, and results and billing. The source data for Epic is hosted on-premise on virtual Windows-based SQL Servers.

The hospital uses Workday as their enterprise resource application and houses all information dealing with the Employees at ACH. The source data is housed on-premise and resides on a virtual Windows-based SQL Server. OnGuard is ACH's badging system and houses information dealing with active, lost or stolen, and deactivated badges on all staff, volunteers and contractors. The data for this application is housed in a virtual Window's based SQL Server.

External Data

When dealing with physician information like credentialing, ACH has contracted with an external third-party vendor Phynd. Phynd is a company that tracks provider information. They supply provider information dealing with credentials, specialties, services, and locations. This source of data is used to track sensitive information about providers that work at ACH as well as outside network provider resources. The source data for this application is hosted by Phynd. A .CSV flat file is sent nightly using ACH's SFTP site.

Data Staging Area

The Data Staging environment consists of a virtualized Window's based server that has been set up with SQL Server 2016 Standard Edition as its database management system (DBMS). The Staging database created on this server will be used to house all incoming operational data before the final load into the data warehouse. Various tables have been created to support loading the information easily into the database. A file share has also been created on this server for housing the .csv files sent nightly from Phynd.

The data staging component includes three main processes or stages: Data Extraction, Data Transformation, and Data Loading. Separate Extraction, Transformation, and Loading (ETL) processes will be created and used for each dataset. Visual Studio 2017 was loaded along with SQL Server Data Tools (SSDT). Using SSDT, we will create SQL Server Integration Service (SSIS) packages to perform the different processes to extract, transform and load from each operational source system. SSIS packages will also be created to import the external .csv flat files from Phynd. All datasets will be imported into raw staging tables within the staging databases.

Each SSIS package will have an extraction process on the ACH production data. A data flow process using T-SQL select statements are run to extract the application data. The next step is the transformation process. All data will then run through a series of transformation checks to clean up any duplicate values, correct spelling errors, configure calculated values, and then data is standardized, filtered, and sorted before being imported into the data warehouse tables.

The initial dataset loads will bring in historical data from each of the production systems. The external data coming from Phynd will only bring in information dealing with active physicians. After the initial feed, the ETL processes will run nightly to pull new, modified or deleted information for the past 24-hour timeframe.

Data Storage

Data Warehouse and Components

A data warehouse (DW) is a database management system (DBMS) that will be a central repository storing information from the various systems within ACH. The DW will have the following characteristics: “subject-oriented”, integrated and standardized, has a “time-variant”, and is “non-volatile which means the previous data is not erased when new information is written” (Naveen, 2016). The data warehouse environment for ACH will be Microsoft's SQL Server 2016 Enterprise Edition as the database management system (DBMS). The DBMS resides in a Window's server-based virtualized computing environment. The ACH data warehouse was developed using the three-tier architecture which includes a bottom, middle and top-tier model. The bottom-tier serves as the repository for storing all data extracts of the hospital's currently operational, external, and archived data. The bottom tier is the level where the staging data ETL process is handled. The middle tier is a Windows-based virtualized server that is set up using

Microsoft Online Analytical Processing (OLAP). This layer utilizes Multidimensional Online Analytical Processing or MOLAP which indexes directly into a multidimensional database (MDDDB). The top-tier supports the information delivery process and sits as the client layer that holds the different tools for querying, analyzing, reporting and data mining.

Data Marts

ACH has requested data marts made available for each of their main departments. A data mart is a “customized” data subset of the information stored within the data warehouse (CertBuddyz, 2018). They are utilized to make accessing, consuming and reporting on data easier for the end-user. It has been decided that a new separate database will be created to house the different data marts. Each data mart will be filtered by departmental specifications and contain data for a rolling fourteen-day period.

Meta Data

Data warehouses contain an amazing amount of information. One of the key components of the data warehouse system is the development and use of metadata. Metadata is considered to be “data about data” “that defined the data warehouse” (CertBuddyz, 2018). Metadata is an essential part of being able to access, build, maintain and manage the DW. There are two different types of Metadata: Technical and Business Metadata. Technical Metadata contains information specifically about the data warehouse and the ETL processes and is mainly used by the warehouse administrators, designers, and developers. The business metadata contains the “high-level business information” that will help the end-user to easily understand the data stored within the data warehouse and information about the “cubes and data marts” (ETL-tools, n.d.).

Information Delivery

The main reason to have a data warehouse is to aggregate information from disparate systems and use it to create an overall picture of business operations by placing data from multiple systems into one centralized repository. Speed and data accuracy are the major players in information delivery.

Multidimensional Database and OLAP

A multidimensional database, also known as a cube is a specialized construction that allows summarized data to be accessed very quickly by the end-user. A multidimensional database is considered a type of Online Analytical Processing (OLAP). OLAP is a “software technology” used for business intelligence operations such as analytical calculations, data discovery, predictive analysis planning, and reporting (JavaTpoint, n.d.). OLAP also allows for end-users to “perform ad hoc analysis” (JavaTpoint, n.d.).

Data Mining

Data mining is defined as the process used to find patterns, correlations, and anomalies within large amounts of datasets. The information gained by data mining is then used to help predict future outcomes and trending. Some techniques used are “tracking patterns” within the data, classifying information, and using “clustering” where you “group chunks of data together” based on similarities (Alton, 2017). After much consideration and reviews, ACH will use FrontlineSolvers Analytic Solver Data Mining tool (formerly XLMiner). This is a comprehensive data mining tool that offers an easy to use interface with a variety of features, which includes forecasting, classification, and predictive methods, as well as offering “example datasets” and “online Help” (FrontlineSolvers, n.d.).

Report and Query

There were a few considerations when looking at what reporting and querying tools to implement with the ACH data warehouse environment. The main purpose of ACH's data warehouse is to provide to their business users complete information pertinent to making strategic business decisions. The reporting aspect is the top-tier of the three-tier architecture and where the users will interact with the data warehouse. A report writing team has been created to handle the development of key business analysis reports that will be consumed by upper and departmental management. Crystal Reports will be the reporting tool for all reporting at this level. Power BI is Microsoft's business intelligence and data visualization tool. End-users will be able to access and create reports and dashboards easily and have drill-down functionality that enables the users to focus deeper into a specific area.

Data Management and Control

Strong data management and control over the data warehouse environment is essential to ensuring high availability of the data warehouse. Data management includes monitoring all jobs and ETL processes are complete without error, ensuring the proper security measures are in place and user access is granted, data is tested for accuracy and properly maintained and a backup discovery plan is in place. The Data Infrastructure team at ACH has been trained in both the administrative and development areas to ensure the proper monitoring and alerts are in place in case of an event.

Design Requirements

Data Warehouse Design Requirements

The following sections deal with designing ACH’s data warehouse. There are three main models used for building the initial data warehouse design: Conceptual, Logical and Physical (Guru99, n.d.). Each model or tier is designed to facilitate building the true foundation of the data warehouse.

Conceptual Design

The first tier used for designing the data warehouse is the Conceptual Model. This model creates the initial layout and focuses on creating the entities that will make up the data warehouse as well as showing the relationships. Figure 1 shows the initial layout for ACH’s data warehouse.

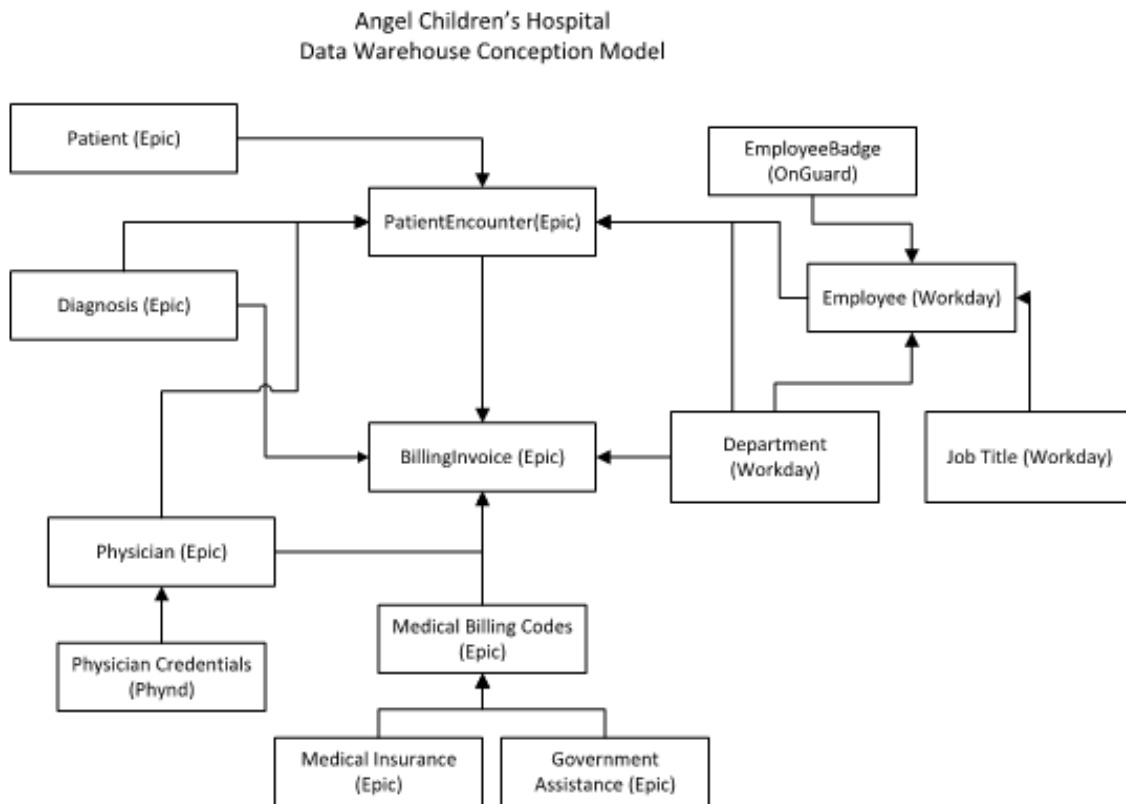


Figure 1. ACH Conceptual Model

Logical Design

The second tier used for designing the data warehouse is the Logical Model. This model builds off of the initial tier and focuses on including the different attributes within each entity. This model identifies the primary key within each entity and shows the relationships between entities using the foreign key relationship. Figure 2 shows the modified layout for ACH's data warehouse which includes the attributes and Primary/Foreign key relationships.

ACH Data Warehouse Logical Design

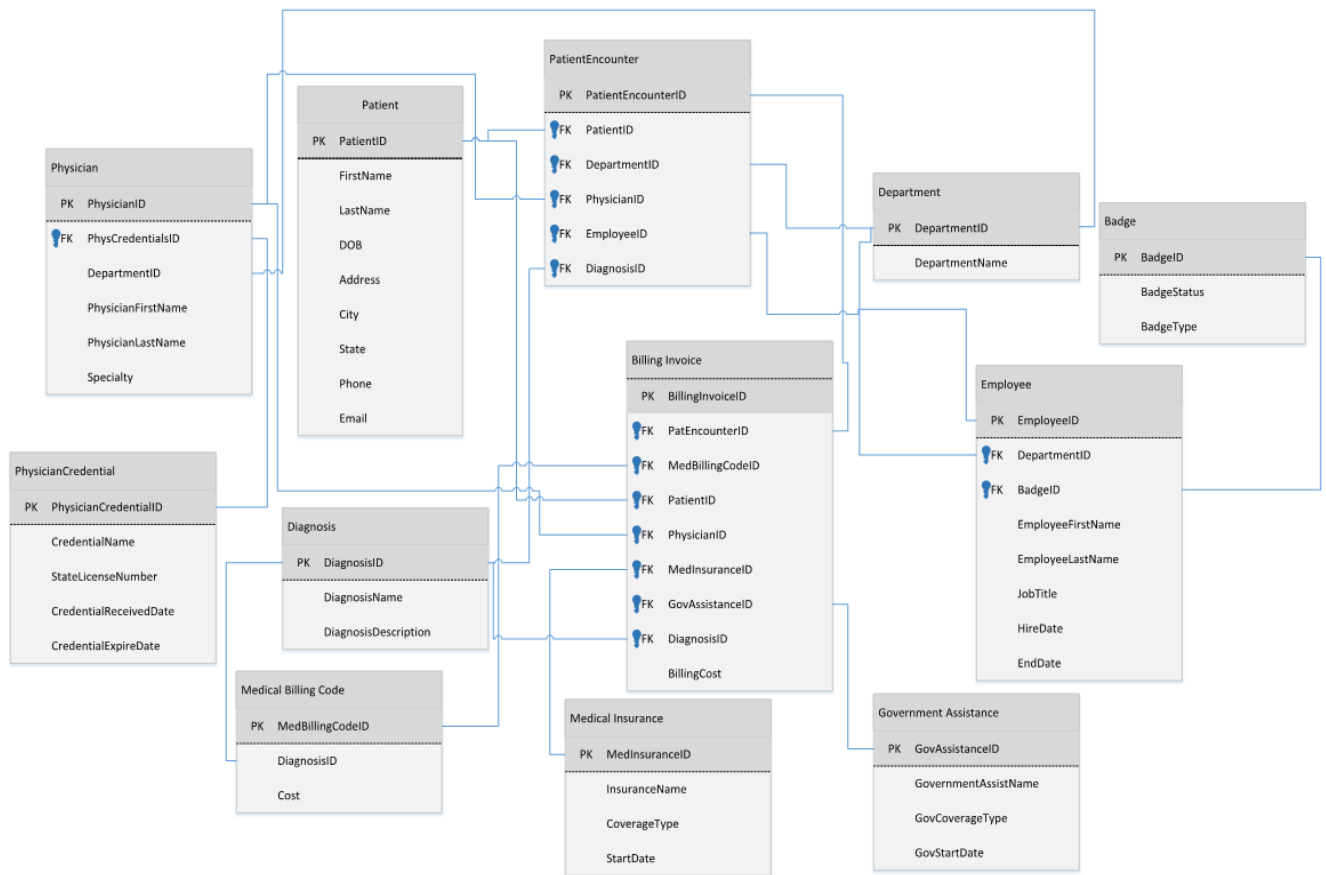


Figure 2. ACH Logical Model

Physical Design

The third and final tier used for designing the data warehouse is the Physical Model. This model builds off of the second, Logical tier and focuses on creating the true structure of the data warehouse. This model identifies the tables and data types while showing the Primary and Foreign key relationships. Figure 3 shows the physical layout which includes the new table structure with data types and key relationships for ACH's new data warehouse.

ACH Data Warehouse Physical Design

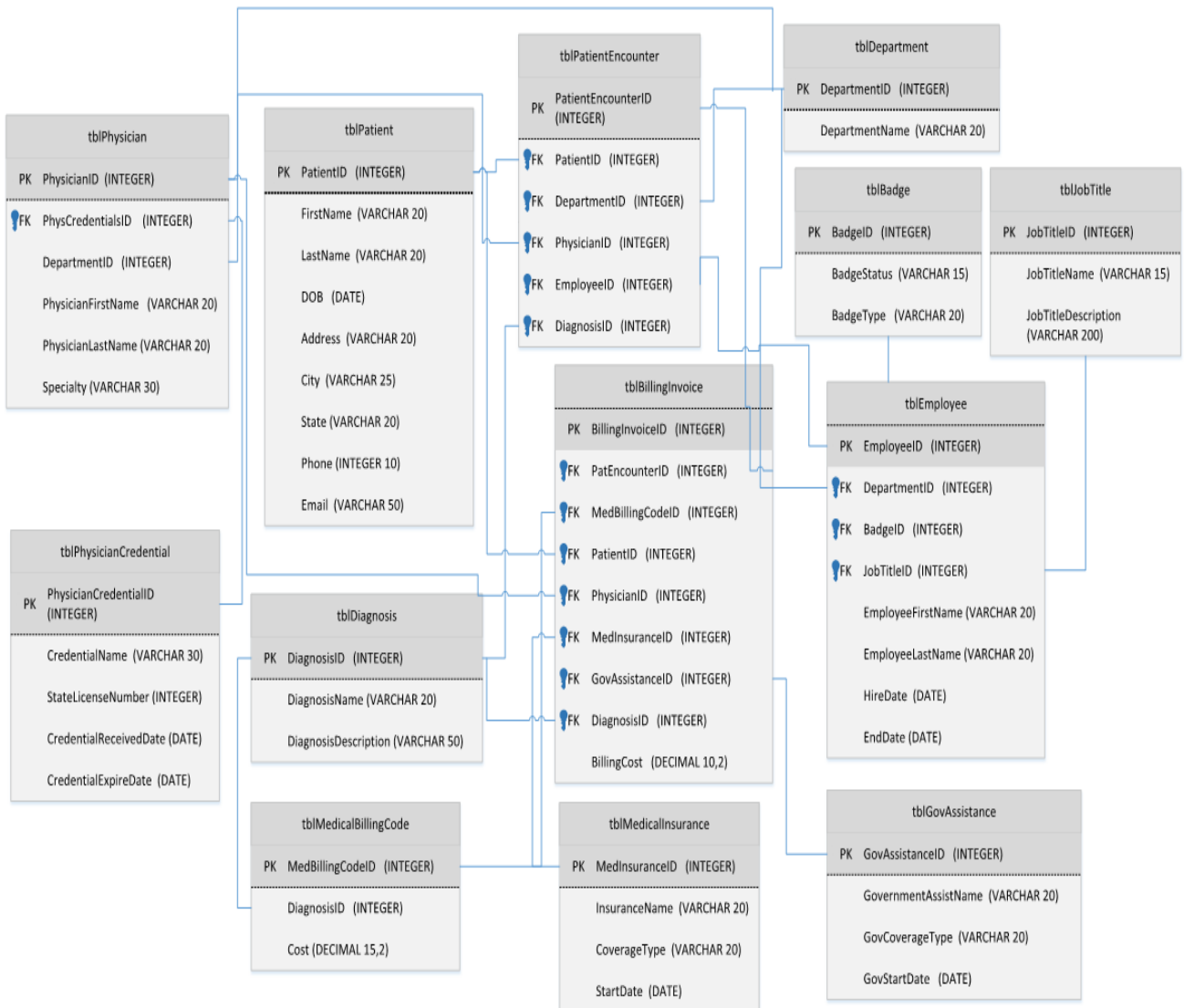


Figure 3. ACH Physical Model

Load Data

ACH Data Load Analysis

ACH Data Warehouse will be developed from multiple data sources. SQL Server Integration Services (SSIS) will be the tool used to create the packages for the different ETL processes needed to extract, transform and load each dataset. Three main areas support the ETL process; the Source Data which will come from either SQL Server or Flat Files, the ACH Staging area which is a SQL Server environment supporting the transformation section, and the ACH Data Warehouse area which is the destination area residing in a SQL Server 2016 Enterprise Edition environment and supports the organization's business intelligence analytics and reporting. The ETL process runs nightly to update the data warehouse. The following sections map out the main steps used to load the data warehouse named ACH_PatientDW1.

The Data Sources

ACH Source Data

The data from ACH comes from three main systems: Epic, OnGuard, and Workday. Each of these applications resides within a SQL Server repository. Through SSIS a data flow task will be used to run individual select statements needed to extract the correct data. Through this data flow task, the data will be extracted and loaded into a set of staging databases within the ACH_Staging database. Table 1 shows an example of a small dataset from the Patient table within ACH's Epic production system.

Table 1

ACH – Source Epic System Patient Table Data Sample

PatientID	FirstName	LastName	DOB (Date of Birth)	Sex	City	State
742951	Alan	Torres	2015-01-23	M	Columbus	Ohio
624588	Becky	Driver	2010-09-22	F	Pickerington	Ohio
507142	Johnny	Conklin	2020-02-29	M	Columbus	Ohio
359217	Brenda	Jarvis	2009-10-11	F	Dublin	Ohio

569920	Andrea	Stevens	2017-12-14	F	Columbus	Ohio
--------	--------	---------	------------	---	----------	------

Table 2 maps out examples of ACH’s application source tables, source data columns, data types, and related Data Warehouse Lookup tables and columns.

Table 2
Source to Data Warehouse Data Examples

Source Table	Source Column	Source Data Type	Lookup Table	Lookup Column
PatientEncounter				
	PatientEncounterID	Int	None	None
	PatientID	Int	tblPatient	PatientID
	DepartmentID	Int	tblDepartment	DepartmentID
	PhysicianID	Int	tblPhysician	PhysicianID
	EmployeeID	Int	tblEmployee	EmployeeID
	DiagnosisID	Int	tblDiagnosis	DiagnosisID
BillingInvoice				
	BillingInvoiceID	Int	tblBillingInvoice	BillingInvoiceID
	PatientID	Int	tblPatient	PatientID
	DiagnosisID	Int	tblDiagnosis	DiagnosisID
	PhysicianID	Int	tblPhysician	PhysicianID
	PatientEncounterID	Int	tblPatientEncounter	PatEncounterID
	MedicaInsuranceID	Int	tblMedicalInsurance	MedicaInsuranceID
	GovAssistanceID	Int	tblGovAssistance	GovAssistanceID
	MedicalBillinCodeID	Int	tblMedicalBillinCode	MedicalBillinCodeID
	BillingCost	Decimal (10,2)		
Physician				
	PhysicianID	Int	tblPhysician	PhysicianID
	PhysCredentialsID	Int	tblPhysCredentials	PhysCredentialsID
	DepartmentID	Int	tblDepartment	DepartmentID

Phynd Source Data

Each night at 8:00 PM an extract is run from the Physician Credentialing database housed within Phynd’s hosted environment, and a Flat file in .csv format is generated and placed within a shared folder housed on the ACH_Staging server. Through SSIS flat file task, the comma-delimited .csv file is picked up, processed, converted and formatted before loading into the

PhysicianCredentialStg table of the ACH data warehouse named ACH_ StagingSQL1601.

Figure 3 shows sample data from the Phynd Physician Credential .csv file.

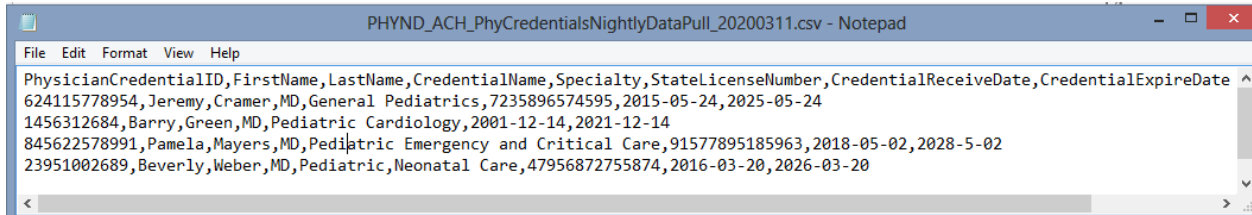


Figure 3: ACH - Phynd Physician Credential Sample Data

Data Transformation

Once the different datasets have been extracted from their source tables, they are loaded into staging tables. These tables are first truncated to clear the data from the previous night's run. Each dataset is then traversed through to check for anomalies and the data types are converted to match the Lookup table columns. A GlobalID is assigned to each record as it is loaded into the staging area. This column is a unique identifier and the naming convention for this GlobalID is the first four letters of the source system name with a number concatenated at the end of the string. The number increments by one with each new record per system.

SSIS task jobs are triggered to complete various data transformations needed before loading into the data warehouse. Next, each dataset within the staging area runs through a series of cleanup and filtering steps. During this process, data is traversed through to search for NULL values and errors are handled with default values. Validation and business rules are applied, data column values are merged and the datasets are optimized. During the cleanup stage data values are changed and values are standardized. An example within the Patient table deal with the column representing the child's sex. Here the value 'M' will be changed to Male and the value of 'F' will be changed to Female. Table 3 shows an example of the Staging Patient data formatted within the Staging area.

Table 3

ACH - Staging Patient table after initial data transformations

GlobalID	PatientID	FirstName	LastName	DOB	Sex	City	State
Epic410	742951	Alan	Torres	2015-01-23	Male	Columbus	Ohio
Epic411	624588	Becky	Driver	2010-09-22	Female	Pickerington	Ohio
Epic412	507142	Johnny	Conklin	2020-02-29	Male	Columbus	Ohio
Epic413	359217	Brenda	Jarvis	2009-10-11	Female	Dublin	Ohio
Epic414	569920	Andrea	Stevens	2017-12-14	Female	Columbus	Ohio

ETL Completion

ACH Data Warehouse: ACH_PatientDW1

ACH_PatientDW1 is the destination data warehouse for Angle Children's patient-related information including billing. SSIS jobs have been created to handle all tasks related to the ETL processes. Sequence Containers are used to process each stage of the ETL processes. The initial sequence container contains the data flow process for extracting the data from the source systems tables or files. These processes refer to the Select statements mentioned in the above sections. The second sequence container contains the transformations tasks which include processes like data cleansing, and data type conversions (also covered in the section above). One of the transformation processes deals with Patient Encounter information. This task handles the date timestamp column. Here the column is separated into two separate columns, one containing the date of the scheduled appointment and the other column containing the timestamp. The final sequence container contains the final data flow task that handles loading the data within the data warehouse. This data flow task is responsible for mapping the columns from staging to the data warehouse tables. Here field lengths are checked along with data and time formats. Next, the tables are loaded in a distinctive order of operation where Lookup tables are loaded before the Fact tables. Table 4 shows examples of both the Lookup and Fact tables.

Table 4

ACH Data Warehouse Lookup and Fact Table Examples

Table Type	Table Name	Table Column	Data Type
Lookup	tblDepartment		
		DepartmentID	Int
		DepartmentName	VARCHAR(20)
Lookup	tblBadge		
		BadgeID	Int
		BadgeStatus	VARCHAR(15)
		BadgeType	VARCHAR(20)
Lookup	tblJobTitle		
		JobTitleID	Int
		JobTitleName	VARCHAR(25)
		JobTitleDescription	VARCHAR(200)
Fact	tblEmployee		
		EmployeeID	Int
		DepartmentID	Int
		BadgeID	Int
		JobTitleID	Int
		EmployeeFirstName	VARCHAR(20)
		EmployeeLastName	VARCHAR(20)
		HireDate	DATE
		EndDate	DATE

Data Analysis

ACH and Data Analytics

The main driver for creating ACH's data warehouse is to create a single repository that connects essential ACH disparate systems. Having a single source of truth will ensure consistent data that it trusted throughout the hospital and creates a venue that takes data analytics for ACH to a whole new level. The following sections touch on the Business Intelligence (BI) side of reporting, online analytical processing (OLAP) and how ACH can utilize OLAP for advanced analytics.

Defining OLAP

Online Analytical Processing (OLAP) is a software technology that allows multiple users to concurrently access information for analysis from "multiple database systems" (Guru99, n.d.). OLAP data is stored as multidimensional cubes. Each cube houses data from some type of categorized dimension (such as Patients, Diseases, Census time periods, Departments, and hospital processes or services). Figure 4 shows an example of an OLAP cube with defined dimensions for Physicians, Diseases and their Care Process.

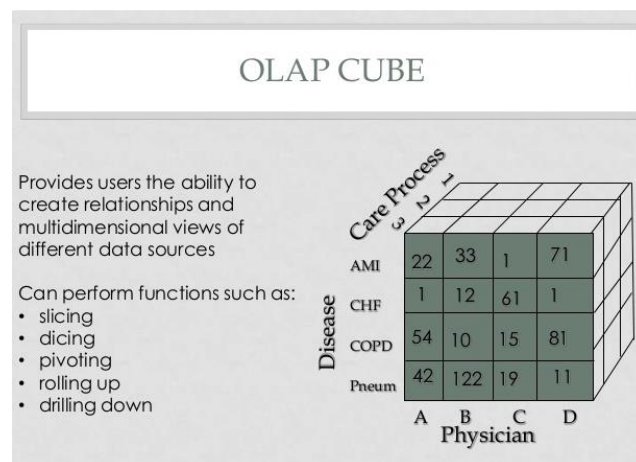


Figure 4: OLAP Cube describing Physicians, Diseases and Care Processes. Retrieved from Sullivan (2013).

This data comes from the “multidimensional tables” within the “data warehouse” (Educba, 2020). Here the end-user can access and perform OLAP operations which include “Drill-up/Roll-up, Drill-down, Pivot, and Slice-and-Dice” (Olap Operations in Data Mining, Slice and Dice, Drill Up and Drill Down with example, 2019).

OLAP Main Types and Processing Tools

There are three main types of OLAP systems: ROLAP for Relational OLAP which is an application based on relational database management systems, MOLAP for Multidimensional OLAP which is an application based on multidimensional database management systems, and HOLAP for Hybrid OLAP which is an application that uses both multidimensional and relational methods. At this time ACH is interested with the specifics of the MOLAP and ROLAP systems.

MOLAP is the classic form of the OLAP systems and may also be referred to as OLAP. Data is stored in optimized multidimensional array storage and requires storing information within a cube and involves pre-computations. Some advantages of using MOLAP are high performance with fast retrieval times and is considered the best choice for slicing and dicing data operations. MOLAP, however, is less scalable than ROLAP, which means it has limitations where data size is considered. So in this case, smaller datasets are needed and work best with MOLAP. Two top tools supporting MOLAP are: “IBM Cognos” which provides reporting, analysis, event monitoring and metrics, and “Microsoft Analysis Services” (SSAS) which is Microsoft’s analytical tool that “supports multidimensional models, data mining”, Power BI, and Reporting Services (Pat Research, 2020).

The ROLAP or Relational Online Analytical Processing system has its data stored in “relational databases and cubes” (1keydata, n.d.). An advantage with ROLAP deals with the ability to handle “large amounts of data” (Pat Research, 2020). Performance can be an issue as ROLAP is SQL based so querying large amounts of data can see a slowness in run times. “IBM Cognos” and “Microsoft Analysis Services” also support ROLAP systems (Pat Research, 2020).

ACH's decision to move forward with MOLAP was based on their organization's smaller datasets, their need for a high-performance system that supports accessing data quickly and easily, while having the ability to develop advanced analytic reporting with dashboards that support mobile and drill-down capabilities.

ACH and OLAP Analytical Tools

The executive leaders have stressed their need to be able to track and easily view vital statistics that support the organization. Through using OLAP, ACH will have the ability for tracking metrics and key performance indicators (KPIs) which in turn will support making better business decisions in clinical, operational and financial areas. ACH is a Microsoft based organization and has decided to use the tools they have under their contract with Microsoft. This includes utilizing Microsoft's SQL Server Analysis Services (SSAS) to create the various OLAP cubes and Data Marts, and Reporting Services and Power BI as their base reporting and advanced analytics tools. Power BI also has mobile capabilities that will allow the end-user to view reports from any smartphone device with connectivity.

Along with OLAP Cubes, Data Marts will initially be developed and built out for key departments like Emergency, Surgery, Cardiology and Pediatric Mental Health. At this level,

department information will be readily available, and access will be based on the end user's department and the permissions requested by the department manager.

The organization has created an Advanced Analytics team that includes data developers and report writers. This team will work with the individual department to create custom reports using Reporting Services and/or Power BI with all reporting capabilities (Drill-up, Drill-down, Slice, Dice, and Pivot). Power BI will also be used to create leadership dashboards for viewing KPIs like Average Hospital Stay, Patient Wait Times, ER Wait Times, and Patient Satisfaction. Financial leaders will have reports created that deal with claims Denial Rates, Average Insurance Claims Processing Times and Costs, Costs by Payer, and Average Treatment Costs. Leadership has expressed their excitement with the hospital's new direction with the data warehouse and OLAP capabilities for reporting.

Maintenance, SQL Script and Conclusion

ACH Data Maintenance Plan

As a Children's hospital, ACH knows the importance of keeping their data safe and secure. Part of this process deals with ensuring all databases are maintained, monitored, and backed-up with a formal recovery plan in place. For the end-user, collecting, processing, and delivering time-sensitive data is key to creating a BI environment that produces results in an efficient and effective manner. The ACH Data Infrastructure Team (DI Team) along with their Networking Team have developed and implemented a solid Data Maintenance Plan which includes the data warehousing environment and the servers on which the databases reside. The following sections focus on these areas as it relates to the ACH data warehouse.

Database Monitoring and Performance Tuning

As the company grows and moves forward, so does their data. ACH's DI Team has developed important processes and procedures to ensure high-availability is preserved while high-performance is maintained. To accomplish this, both database tuning and performance monitoring will need to be continually observed, managed and tested. ACH contracted with Solarwinds, which offers a third-party application used to "monitor and alert" specific designated users when a server and/or database issue arises (Solarwinds, n.d.). One of the key reasons ACH chose this company was the product's ability for the ACH Networking and DI Teams to easily setup and monitor customized metrics at both the database and server level.

At the server level, for all servers where databases reside, alerts have been setup to trigger when any drive size reaches 80% capacity and then again at 90% capacity. Alerts for CPU utilization have also been set up to trigger when CPU utilization reaches 85% or higher.

These alerts show on the application's Management Console as well as sends email messages automatically to both the Network and DI Team members listed within an assigned group.

At the database level, monitoring and managing both the data environment and data growth are essential parts of keeping the Data Warehouse performing at its optimal level. The DI Team will run daily checks on all nightly jobs, which includes all ETL processes and backup maintenance jobs. Monitoring using "SolarWinds' Management Console", will immediately enable the team to view issues with hanging or failing processes and monitor the "health of the database" (Solarwinds, n.d.). The team will also monitor the data flow process and watch for timing issues and any errors that are reported and need immediate attention. They will be looking for any performance issues and check to see if any new indexes need to be created or if the ETL process needs to be updated in any way. The team is also in charge of performing data validation checks which include using summary reports and running scripts to ensure data integrity is kept.

Backup and Recovery Plans

At the database level there are multiple backup and recovery plans in place that support the ACH Data Warehouse. All source application databases will have a FULL Backups run nightly with DIFFERENTIAL backups running every two hours. Transaction Log Backups are run on databases using the FULL Recovery Model. Monthly scripts are run against the database to track all active Logins. This active login file is kept on the local server and is copied over to the South Satellite Campus.

For the Data Warehouse environment a FULL backup will be performed weekly starting at 10 PM every Friday and Tuesday night. DIFFERENTIAL backups are to be performed Daily, starting at noon and running every six hours, excluding Friday and Tuesday Nights. The data

warehouse is setup for reporting purposes only and therefore only modified through the nightly ETL process. The data warehouse is set using the SIMPLE Recovery Model so no Transaction Log backups are needed at this time. All backups are stored locally on the server's backup drive and are available for immediate restores if needed for recovery. These backup drives are also copied over to tape nightly and are housed at an off-site location.

At the server level, there are two separate types of recovery processes that happen to ensure the availability of the backup files in case of a disaster event. The first process deals with the Network Team. They have setup nightly copies of all SAN volumes which includes all backup data to our South Satellite Campus. This campus is located an hour away from the main campus and retrieval processing times are routinely tested. The second process uses the services of Iron Mountain, a company which "transports and stores information" on backup storage media (Iron Mountain, n.d.). ACH has contracted with Iron Mountain to transport and store the backups on tape to their facilities. Per the contract, there is a four hour window for retrieval time and monthly tests are performed on key applications.

SQL Data Warehouse Scripting

This section deals with the SQL scripts used to develop the Data Warehouse for ACH.

The scripts below are designed from the Physical model depicted in Figure 3 within the Design Requirements section.

-- Data Warehouse Creation Script

```
CREATE DATABASE ACH_PatientDW1  
GO
```

-- Data Warehouse Table Creation Scripts

```
USE ACH_PatientDW1  
GO
```

```
CREATE TABLE tblDepartment(  
DepartmentID INT IDENTITY (1,1) PRIMARY KEY,  
DepartmentName VARCHAR(30) NOT NULL  
);
```

```
CREATE TABLE tblBadge (  
BadgeID INT IDENTITY(1,1) PRIMARY KEY,  
BadgeStatus VARCHAR(15) NOT NULL,  
BadgeType VARCHAR(20) NOT NULL  
);
```

```
CREATE TABLE tblJobTitle (  
JobTitleID INT IDENTITY (1,1) PRIMARY KEY,  
JobTitleName VARCHAR (15),  
JobTitleDescription VARCHAR(200)  
);
```

```
CREATE TABLE tblDiagnosis (  
DiagnosisID INT IDENTITY (1,1) PRIMARY KEY,  
DiagnosisName VARCHAR(20) NOT NULL,  
DiagnosisDescription VARCHAR(50)  
);
```

```
CREATE TABLE tblMedicalInsurance (  
MedicalInsuranceID INT IDENTITY (1,1) PRIMARY KEY,  
InsuranceName VARCHAR(20) NOT NULL,  
CoverageType VARCHAR(20) NOT NULL,  
StartDate DATE NOT NULL  
);
```

```
CREATE TABLE tblGovAssistance (  
GovAssistanceID INT IDENTITY (1,1) PRIMARY KEY,  
GovAssistanceName VARCHAR(20) NOT NULL,  
GovAssistanceType VARCHAR(20) NOT NULL,  
GovStartDate DATE NOT NULL  
);
```

```
CREATE TABLE tblPatient (  
PatientID INT IDENTITY (1,1) PRIMARY KEY,  
FirstName VARCHAR(20) NOT NULL,  
LastName VARCHAR(20) NOT NULL,  
DOB DATE NOT NULL,  
PatAddress VARCHAR(20) NOT NULL,  
City VARCHAR(20) NOT NULL,  
PatState VARCHAR(20) NOT NULL,  
Phone INT,  
Email VARCHAR(20)  
);
```

```
CREATE TABLE tblPhysicianCredential (  
PhysicianCredentialID INT IDENTITY (1,1) PRIMARY KEY,  
CredentialName VARCHAR(30) NOT NULL,  
StateLicenseNumber INT,  
CredentialReceivedDate DATE NOT NULL,  
CredentialExpireDate DATE  
);
```

```
CREATE TABLE tblMedicalBillingCode (  
MedBillingCodeID INT IDENTITY (1,1) PRIMARY KEY,  
DiagnosisID INT NOT NULL,  
Cost DECIMAL(15,2)  
);
```

```
CREATE TABLE tblPhysician (  
PhysicianID INT IDENTITY (1,1) PRIMARY KEY,  
PhysCredentialID INT,  
DepartmentID INT,  
PhysicianFirstName VARCHAR(20) NOT NULL,  
PhysicianLastName VARCHAR(20) NOT NULL,  
Specialty VARCHAR(30) NOT NULL  
);
```

```
CREATE TABLE tblEmployee (  
EmployeeID INT IDENTITY (1,1) PRIMARY KEY,  
DepartmentID INT NOT NULL,  
BadgeID INT NOT NULL,  
JobTitleID INT NOT NULL,  
EmployeeFirstName VARCHAR(20),  
EmployeeLastName VARCHAR(20),  
HireDate DATE,  
EndDate DATE  
);
```

```
CREATE TABLE tblPatientEncounter (  
PatientEncounterID INT IDENTITY (1,1) PRIMARY KEY,  
PatientID INT NOT NULL,  
DepartmentID INT NOT NULL,  
PhysicianID INT NOT NULL,  
EmployeeID INT NOT NULL,  
DiagnosisID INT NOT NULL,  
PatEncDate DATE NOT NULL  
);
```

```
CREATE TABLE tblBillingInvoice (  
BillingInvoiceID INT IDENTITY (1,1) PRIMARY KEY,  
PatientEncounterID INT NOT NULL,  
MedBillingCodeID INT NOT NULL,  
PatientID INT NOT NULL,  
PhysicianID INT NOT NULL,  
MedicalInsuranceID INT NOT NULL,  
GovAssistanceID INT NOT NULL,  
DiagnosisID INT NOT NULL,  
BillingCost DECIMAL(10,2)  
);
```

-- Referential Key Creation Scripts

```
ALTER TABLE tblMedicalBillingCode
ADD FOREIGN KEY (DiagnosisID) REFERENCES tblDIAGNOSIS(DiagnosisID);
```

```
ALTER TABLE tblPhysician
ADD FOREIGN KEY (PhysCredentialID) REFERENCES
tblPhysicianCredential(PhysicianCredentialID);
```

```
ALTER TABLE tblPhysician
ADD FOREIGN KEY (DepartmentID) REFERENCES tblDepartment(DepartmentID);
```

```
ALTER TABLE tblEmployee
ADD FOREIGN KEY (DepartmentID) REFERENCES tblDepartment(DepartmentID);
```

```
ALTER TABLE tblEmployee
ADD FOREIGN KEY (BadgeID) REFERENCES tblBadge(BadgeID);
```

```
ALTER TABLE tblEmployee
ADD FOREIGN KEY (JobTitleID) REFERENCES tblJobTitle(JobTitleID);
```

```
ALTER TABLE tblPatientEncounter
ADD FOREIGN KEY (PatientID) REFERENCES tblPatient(PatientID);
```

```
ALTER TABLE tblPatientEncounter
ADD FOREIGN KEY (DepartmentID) REFERENCES tblDepartment(DepartmentID);
```

```
ALTER TABLE tblPatientEncounter
ADD FOREIGN KEY (PhysicianID) REFERENCES tblPhysician(PhysicianID);
```

```
ALTER TABLE tblPatientEncounter
ADD FOREIGN KEY (EmployeeID) REFERENCES tblEmployee(EmployeeID);
```

```
ALTER TABLE tblPatientEncounter
ADD FOREIGN KEY (DiagnosisID) REFERENCES tblDiagnosis(DiagnosisID);
```

```
ALTER TABLE tblBillingInvoice
ADD FOREIGN KEY (PatientEncounterID) REFERENCES
tblPatientEncounter(PatientEncounterID);
```

```
ALTER TABLE tblBillingInvoice
ADD FOREIGN KEY (MedBillingCodeID) REFERENCES
tblMedicalBillingCode(MedBillingCodeID);
```

```
ALTER TABLE tblBillingInvoice
ADD FOREIGN KEY (PatientID) REFERENCES tblPatient(PatientID);
```

```
ALTER TABLE tblBillingInvoice  
ADD FOREIGN KEY (PhysicianID) REFERENCES tblPhysician(PhysicianID);
```

```
ALTER TABLE tblBillingInvoice  
ADD FOREIGN KEY (MedicalInsuranceID) REFERENCES  
tblMedicalInsurance(MedicalInsuranceID);
```

```
ALTER TABLE tblBillingInvoice  
ADD FOREIGN KEY (GovAssistanceID) REFERENCES  
tblGovAssistance(GovAssistanceID);
```

```
ALTER TABLE tblBillingInvoice  
ADD FOREIGN KEY (DiagnosisID) REFERENCES tblDiagnosis(DiagnosisID);
```

In Closing

Angel Children's Hospital is committed to offering the best possible care to their patients. Adding a data warehouse environment creates a solid reporting base to analyze essential information. ACH is very excited at having the ability to gain new perspectives over their data to help drive better business decisions. This in turn will help them to reach their goals of continuous improvements, and become better healthcare providers.

References

- 1keydata. (n.d.). What Is OLAP. Retrieved from <https://www.1keydata.com/datawarehousing/what-is-olap.html>
- Alton, L. (2017, December 22). The 7 Most Important Data Mining Techniques. Retrieved from <https://www.datasciencecentral.com/profiles/blogs/the-7-most-important-data-mining-techniques>
- CertBuddyz. (2018, July 2). Data Warehouse Concepts, Architecture and Components – CertBuddyz. Retrieved from <https://certbuddyz.com/data-warehouse-concepts-architecture-and-components/>
- Educba. (2020, February 3). What is OLAP? Retrieved from <https://www.educba.com/what-is-olap/>
- ETL-tools. (n.d.). Metadata - business and technical. Retrieved from <https://etl-tools.info/en/metadata.html>
- FrontlineSolvers. (n.d.). solver. Retrieved from <https://www.solver.com/>
- Guru99. (n.d.). What is Data Modelling? Conceptual, Logical, & Physical Data Models. Retrieved from <https://www.guru99.com/data-modelling-conceptual-logical.html>
- Guru99. (n.d.). What is OLAP (Online Analytical Processing): Cube, Operations & Types. Retrieved from <https://www.guru99.com/online-analytical-processing.html>
- HIPAA 101. (2020, February 23). ▷ HIPAA Regulatory Rules. Retrieved from <https://hipaa-101.com/hipaa-rules.htm>
- Iron Mountain. (n.d.). Retrieved from <https://www.ironmountain.com/>
- JavaTpoint. (n.d.). What is OLAP (Online Analytical Processing)? Retrieved from <https://www.javatpoint.com/what-is-olap>
- Olap Operations in Data Mining, Slice and Dice, Drill Up and Drill Down with example. (2019, February 7). Retrieved from <https://galaktika-soft.com/blog/olap-operations-in-data-mining.html>
- Naveen. (2016, January 2). Data Warehouse Overview by IntelliPaat. Retrieved from <https://intellipa.com/blog/tutorial/data-warehouse-tutorial/data-warehouse-overview/>
- Pat Research. (2020, January 3). What are Online Analytical Processing Tools?. Retrieved from <https://www.predictiveanalyticstoday.com/online-analytical-processing-tools/>

Solarwinds. (n.d.). Server & Application Monitor. Retrieved from
<https://www.solarwinds.com/server-application-monitor>

Sullivan, N. (2013, March 12). Figure 4. Healthcare business intelligence. Retrieved from
<https://www.slideshare.net/dachsund12/healthcare-business-intelligencensullivan>