
Using Business Intelligence to Improve DBA Productivity

Eric A. Mortensen^A, En Cheng^B

^A Sherwin-Williams, 101 Prospect Avenue, Cleveland, OH 44115, USA, eric.a.mortensen@sherwin.com

^B Computer Science Department, The University of Akron, Akron, OH 44325, USA, echeng@uakron.edu

ABSTRACT

The amount of data collected and used by companies has grown rapidly in size over the last decade. Business leaders are now using Business Intelligence (BI) systems to make effective business decisions against large amounts of data. The growth in the size of data has been a major challenge for Database Administrators (DBAs). The increase in the number and size of databases at the speed they have grown has made it difficult for DBA teams to provide the same level of service that the business requires they provide. The methods that DBAs have used in the last several decades can no longer be performed with the efficiency needed over all of the databases they administer. This paper presents the first BI system to improve DBA productivity and providing important data metrics for Information Technology (IT) managers. The BI system has been well received by Sherwin Williams Database Administrators. It has i) enabled the DBA team to quickly determine which databases needed work by a DBA without manually logging into the system; ii) helped the DBA team and its management to easily answer other business users' questions without using DBAs' time to research the issue; and iii) helped the DBA team to provide the business data for unanticipated audit request.

TYPE OF PAPER AND KEYWORDS

Regular research paper: *Business Intelligence, Data Warehouse, DBA, ETL, Capacity Planning*

1 INTRODUCTION

Companies and organizations are gathering and storing large amounts of data on a daily basis. A Database Administrator (DBA) builds and maintains the databases that store data and support the data needs of a company or organization. Presently, 120,400 DBAs are employed throughout the United States, and 24,400 job openings are expected over the next 10 years [1]. According to research from University of California, Berkeley, more data has been created in the last three years than in all the past 40,000 years. The total amount data will quadruple in the next two years, said Stephen A. Brobst, chief technology officer of Teradata

Corporation [2], at The Data Warehouse Institute (TDWI) [3] World Conference in 2012. This trend has required DBAs to manage and administer a larger number of databases than they have in the past. The explosion of data requires the ability to store, secure, and manage the physical data, DAMA International [4] president John Schley said, but it also demands that the stored data be useful and meaningful. An example of this trend can be seen at Sherwin Williams in Table 1.

Sherwin Williams [5] is a Fortune 500 company that develops, manufactures, and sells paints and other coatings to professional, industrial, commercial, and retail customers in over one hundred different

Table 1: A summary of total number of Databases and DBAs from 2003 to 2013

Year	Total Databases	DBAs	Support DBAs	Databases Per DBA	Databases Per Support DBA
2003	92	3	3	30.67	30.67
2004	115	4	4	28.75	28.75
2005	136	6	6	22.67	22.67
2006	167	7	7	23.86	23.86
2007	190	10	7	19.00	27.14
2008	221	10	7	22.10	31.57
2009	303	17	8	17.82	37.88
2010	572	18	9	31.78	63.56
2011	732 (see note)	20	9	36.60	81.33
2012	707	21	11	33.67	64.27
2013	762	25	11	30.48	69.27

(Note: in 2011 Sherwin Williams was in the process of converting some databases to new hardware and new software. For example, our primary Oracle ERP system was migrated from old hardware to Oracle Exadata [6] hardware and the Oracle E-Business Suite 11.x was upgraded to 12.1 software. There was a period of time when both were running in 2011. This is why there is that spike in the number of databases in 2011 and a decrease in 2012.)

countries. Its total revenue in 2013 was 10,186 million U.S. dollars. This growth has been one of the primary factors in the large increase of databases at Sherwin Williams.

In 2013 there were eight times more database instances than there were just ten years prior. This growth has required Sherwin Williams to continue to hire DBAs in order to manage, administer, and assist projects/applications using those databases. In Table 1, the main difference between the *DBAs* column and the *Support DBAs* column is that the *Support DBAs* column identifies the number of DBAs who are primarily responsible for the administration and support of the database they manage. Since 2007, the DBAs' responsibilities have expanded to play an important role in helping different Sherwin Williams projects during their development and quality assurance phases as well as being responsible for the administration and support of the databases they manage. DBA assistance is highly needed in complex software such as implementing Enterprise Resource Planning (ERP) software such as the Oracle E-Business Suite. This has required some database administrators to transition from traditional DBA database support to project-oriented DBA tasks.

Figure 1 shows the trend between the number of databases per DBA and the number of databases per

Support DBA. According to Figure 1, the number of Databases per DBA at Sherwin Williams didn't show significant change from 2003 to 2013, while the number of databases per Support DBA is continuing to increase, although there is a spike in the number of databases in 2011 and a decrease in 2012 (see note below Table 1).

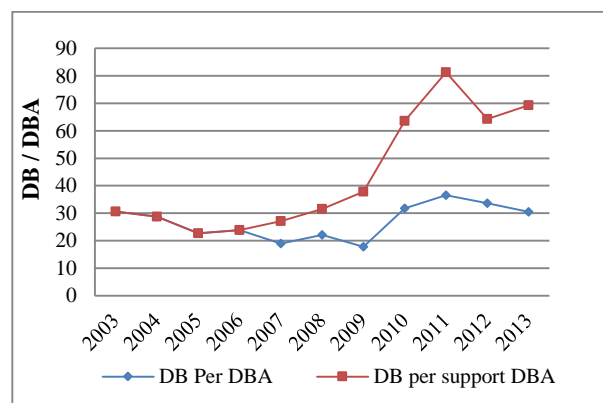


Figure 1: Database per DBA and Databases per Support DBA growth rate

With the increasing number of databases per support DBA, it has become more difficult for the

Sherwin Williams DBA team to continue to provide the same level of administration service to the business. DBAs could no longer log into each database periodically and run scripts to check the health of each database. This caused some issues, such as not identifying that a database was using nearly all of its allocated storage space until all free space had been consumed. At the same time, Information Technology (IT) managers were also beginning to ask questions about IT systems. Questions such as: i) Based on our current usage, how much additional storage space will we need to budget for next year; ii) How many concurrent programs are running daily over all of Oracle ERP systems and how many of these programs experienced errors; iii) How fast is a particular database growing over time; and iv) Which databases are using the most resources on a particular server.

Business Intelligence (BI) is now widely used to describe analytic applications in companies and organizations. BI has become a strategic initiative and is now recognized by chief information officers and business leaders as instrumental in driving business effectiveness and innovation. Business intelligence systems combine operational data with analytical tools to present complex and competitive information to decision makers and planners. BI is a process that includes two primary activities: getting data in and getting data out [7]. Getting data in, traditionally referred to as data warehousing, involves moving data from a set of source systems into an integrated data warehouse. A data warehouse is a structured repository of integrated, subject-oriented, enterprise-wide, historical, and time-variant data aimed at supporting decision-making processes [8]. Getting data out, commonly referred to as BI, consists of business users and applications accessing data from the data warehouse to perform enterprise reporting, querying, and predictive analytics. For example, BI technology is used in manufacturing for order shipment and customer support, in retail for user profiling to target grocery coupons during checkout, in financial series for claims analysis and fraud detection, in telecommunications for identifying reasons for customer churn, in utilities for power usage analysis, and in health care for outcomes analysis. Today, it is difficult to find a successful enterprise that has not leveraged BI technology for its business. For successful DBA teams it is important to empower the staff and management with BI technologies for strategic decision making.

This paper presents a Business Intelligence (BI) system called as *DBA Metrics Repository & Dashboard* aiming at improving DBA productivity and providing important metrics for IT managers. The BI system for Sherwin Williams DBA team has several important features. The system is capable of easily

adding or removing a database as a source of data due to the speed at which databases are being added, moved, or removed from our systems. The data extraction, transformation and loading (ETL) jobs are fully automated such that no manual intervention is needed to re-run the job if it failed. The system has the capability of knowing not to run ETL jobs against a database if maintenance work is being performed on the database. The BI system can assist DBAs in a wide range of areas. For instance, using the *DBA Metrics Dashboard* the DBA team can: i) quickly identify which tablespaces need additional space among the over 700 databases we manage; ii) accurately perform capacity planning for existing hardware; iii) efficiently identify performance issues with applications; div quickly identify which backups are failing, which databases do not have enough storage space allocated to be backed up, and which systems are not being backed up at all. These tasks were time consuming to perform prior to this system.

The DBA team has also been able to use *the DBA Metrics Repository & Dashboard* to perform tasks not envisioned in the original scope of the project. Some of these tasks are: i) helping the security team identify when specific user accounts were logged into the system; ii) helping developers understand how their code is performing in production servers they do not have permissions to log into; iii) helping the DBA manager easily answer questions asked by other business managers without interrupting a DBA. The DBA team could not have easily performed these unexpected tasks if *the DBA Metrics Repository & Dashboard* had not been designed using existing Data Warehousing and BI techniques proposed by Ralph Kimball [4].

The remainder of this paper is structured as follows: Section 2 describes the details of the Data Warehousing environment proposed to support DBA teams; Section 3 discusses our Dashboard which we developed on top of *the DBA Metrics Repository*. Section 4 presents evaluation results for our BI system; Section 5 discusses related work; and Section 6 concludes with discussions and future work.

2 A DATA WAREHOUSING ENVIRONMENT: DBA METRICS REPOSITORY

The BI system for Sherwin Williams DBA team is internally known as *DBA Metrics Repository & Dashboard*. The system has the capability of easily adding or removing a database as well as monitoring ETL jobs that are run against any database where the data was pulled from. The top four features of this BI system are summarized as follows:

- 1) A framework that can automatically create and run all defined ETL jobs for a data source when it is created, based on the data source type;
 - 2) Ability to temporarily halt a scheduled ETL job if the source of data is moved into a blackout state;
 - 3) Automatic purging of old fact table data based on configurable parameters;
 - 4) Ability to gracefully handle unexpected errors so that unexpected ETL errors do not prevent other ETL job from running and allow the ETL job that received the error to run on its next scheduled execution time.
- 3) Needing to collect server information (e.g. Server Usage Statistics) which did not exist in the database and could not be gathered from code executing in a database session;
 - 4) Some data could only be gathered when it existed in the system (e.g. individual Oracle Sessions data).

There were many challenges in the early development of the *DBA Metrics Repository & Dashboard*. The top four challenges were:

- 1) Needing to pull data from multiple Oracle Enterprise Repositories and Oracle Database instance of different versions;
- 2) Gathering session level data from over 400 Oracle database instances every five minutes without using all of the CPU, I/O, and memory resources on the server;

The *DBA Metrics Repository & Dashboard* application as designed today has resolved all of these challenges. An overview of the architecture for *DBA Metrics Repository & Dashboard* is shown in Figure 2. In summary, this system pulls/extracts data from different source databases through an ETL process and loads the data into the *DBA Metrics Repository*. After the data has been loaded into the database, a DBA can access the data using a Dashboard written with Oracle Application Express software.

Currently the *DBA Metrics Dashboard* is only accessible in the internal Sherwin Williams network on an administrator's desktop or laptop. The dashboard's web pages are not designed to display the content on a phone or tablet at the current stage. If there is a need to display contents on mobile devices in the future, the Oracle Application Express software, with which the

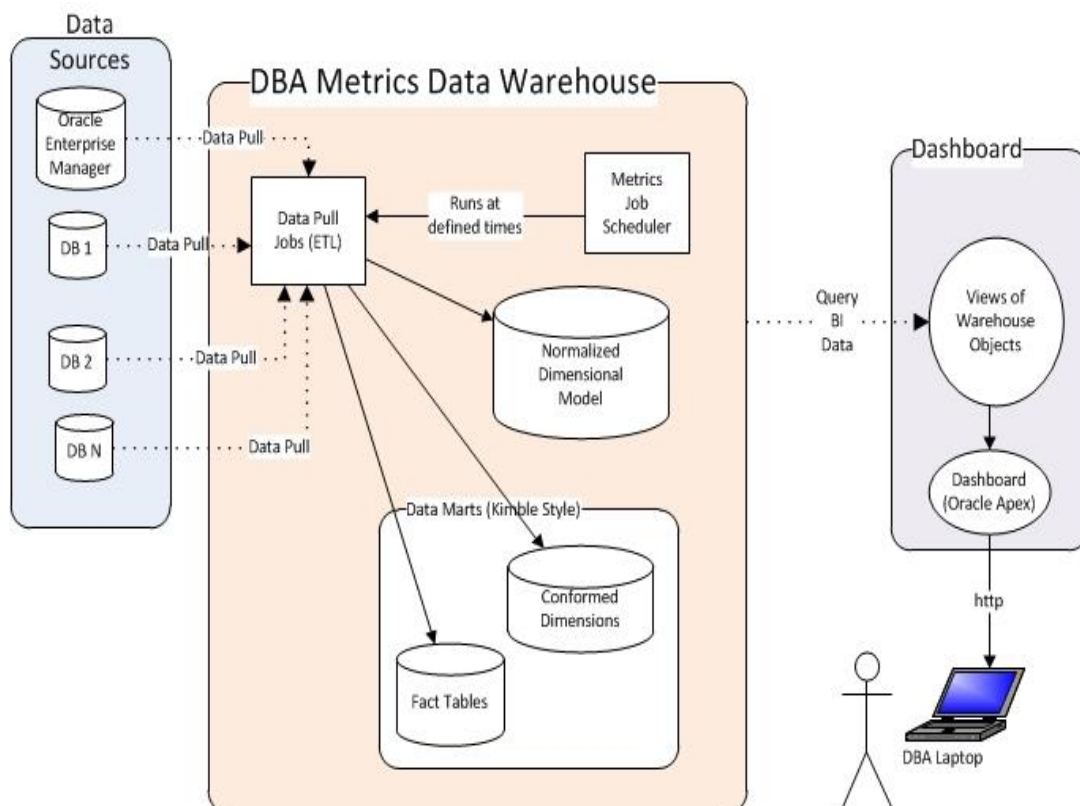


Figure 2: The Architecture for the *DBA Metrics Repository & Dashboard*

DBA Metrics Dashboard is written, has features that would allow the Dashboard to be effectively displayed on mobile devices.

2.1 Data Sources for the *DBA Metrics Repository* System

The data for the *DBA Metrics Repository* system are pulled from two distinct data sources: the *Oracle Enterprise Manager* repository and individual Oracle databases at Sherwin Williams.

Most of dimensional data for the *DBA Metrics Repository* system is pulled from the *Enterprise Manager*. The *Enterprise Manager* system is also used to identify when ETL jobs cannot be performed on a particular database. This is done by occasionally identifying which databases have been placed in blackout mode. Once a database is placed in the blackout mode, the *DBA Metrics Repository* stops pulling data from that database until the blackout has been lifted.

The *DBA Metrics Repository* also pulls data directly from Oracle Databases when the data needed for dimensions or fact tables are not found in *Enterprise Manager*. Examples of these techniques are provided for the *Oracle RMAN* job data mart (Oracle RMAN is the tool that Oracle DBAs use to backup an Oracle database) in Section 2.2.3.

2.2 DBA Metrics Repository - Data Warehouse

2.2.1 ETL Data Pulls

ETL jobs were created to pull data from Data Sources described in section 2.1. These ETL jobs were written in Oracle's PL/SQL language. PL/SQL was chosen over existing ETL tools (e.g. Informatica [9], Oracle Data Integrator [10], and IBM's Infosphere [11]) for several reasons. The primary reason was our team had expertise in writing PL/SQL code and Sherwin Williams did not own any licenses for the other ETL tools.

Using PL/SQL as the primary ETL tool resulted in several important advantages. The primary advantage is performance. Writing the ETL code directly allowed the development team to use the most efficient methods to pull data from different data sources. The best example of this is pulling live session data from Oracle databases. The goal was to load data about each session running in each Oracle database every five minutes. Having over 400 Oracle data sources defined in the *DBA Metrics Repository* requires that a new Database Session ETL Data Pull starts every second. Since this ETL job pull takes longer than a second to run, it would have been difficult to have achieved the

necessary performance using any existing ETL software. Based on the performance of our ETL data pulls over two years, there are no plans to investigate using a professional ETL tool in the near future.

The primary limitation of using PL/SQL as the *DBA Metrics Repository* ETL tool is that PL/SQL can only access data that can be read in an Oracle database session. For instance, it is impossible to directly run UNIX level commands from PL/SQL. The second limitation is that the manually written ETL jobs take time to write and debug where professional ETL tools allow you to define how data fields are mapped between the source and target systems and also handle creating the code that runs to move data into the data warehouse.

2.2.1.1 Dimension Table ETL Data Pulls

Once dimensional data is read from the source system, it is compared against a Normalized Model of the dimensional data (stored in the *DBA Metrics Repository*) to identify which records are new dimensional records, which records are updates to existing dimensional records, and which dimensional records no longer exist in the source systems. In Type 2 Slowly Changing Dimensions, a new record is added to the dimension instead of having the current data updated as defined by Ralph Kimball [12]. The new record gets its own primary key. Therefore, both the original and the new record values will be present for Type 2 Slowly Changing Dimensions.

Most dimensions in the *DBA Metrics Repository* are Type 2 slowly changing dimensions. A dimensional staging record is created for a Type 2 dimension to identify the changes that have been made to the dimension and to assist in identifying which dimensional record a fact table record needs to be tied to. No dimensional staging record needs to be created for Type 1 Slowly Changing dimension.

For most dimension tables, the ETL Data pulls follow the same basic steps. The procedure for pulling dimension table data is defined in Figure 3. A graphical representation on how the data flows based on the Server and Database tables is shown in Figure 4.

In the *DBA Metrics Data Warehouse*, all dimensional data is stored as a *Normalized Dimensional Model* (the *SERVER* and *DATABASE* tables shown in Figure 4 are examples of this). This *Normalized Dimensional Model* has many benefits including: identifying records that will be displayed on the *DBA Metrics Dashboard*; as a snapshot of the current systems managed by DBAs; and as the starting point for calculating how fast a system is growing when performing capacity planning.

2.2.1.2 Fact Table ETL Data Pulls

No two fact table ETL data pull algorithms follow exactly the same logic. This is caused by how the fact data is organized in the source system. Even with these variations, the algorithms for most fact table ETL data

pulls follow a generic pattern. Figure 5 shows the generic procedure used for Fact Table ETL data pulls. A graphical representation of the data flows based on the Server and Database tables is shown in Figure 6.

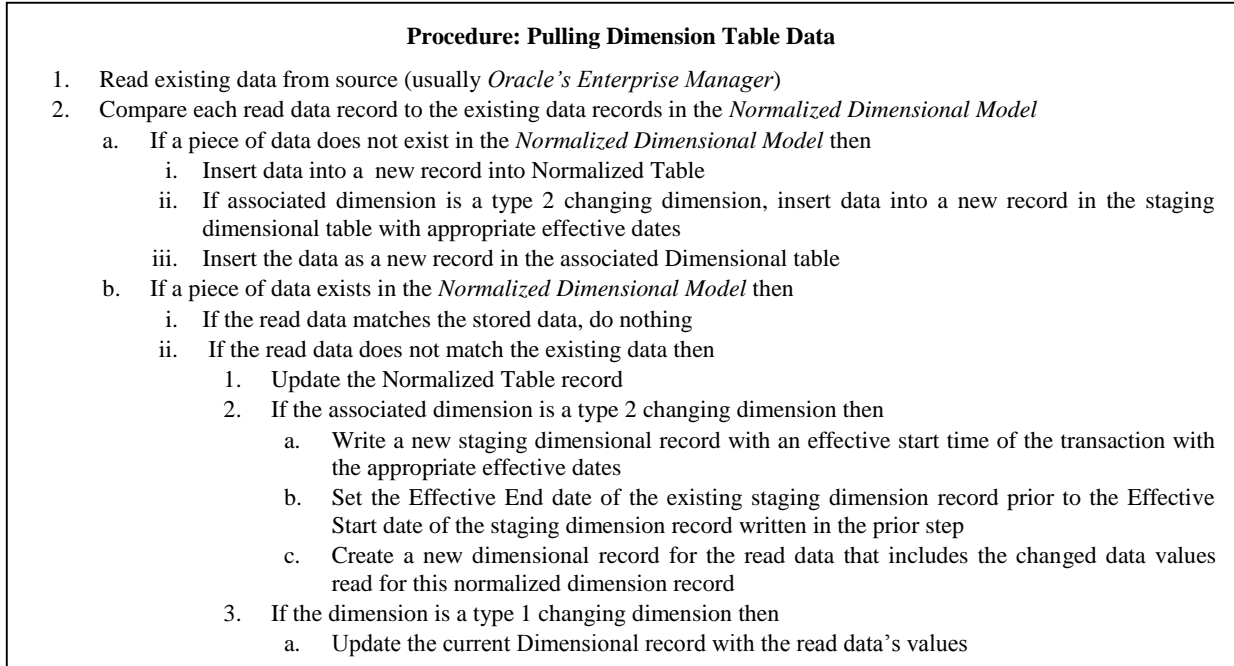


Figure 3: Procedure: Pulling Dimension Table Data

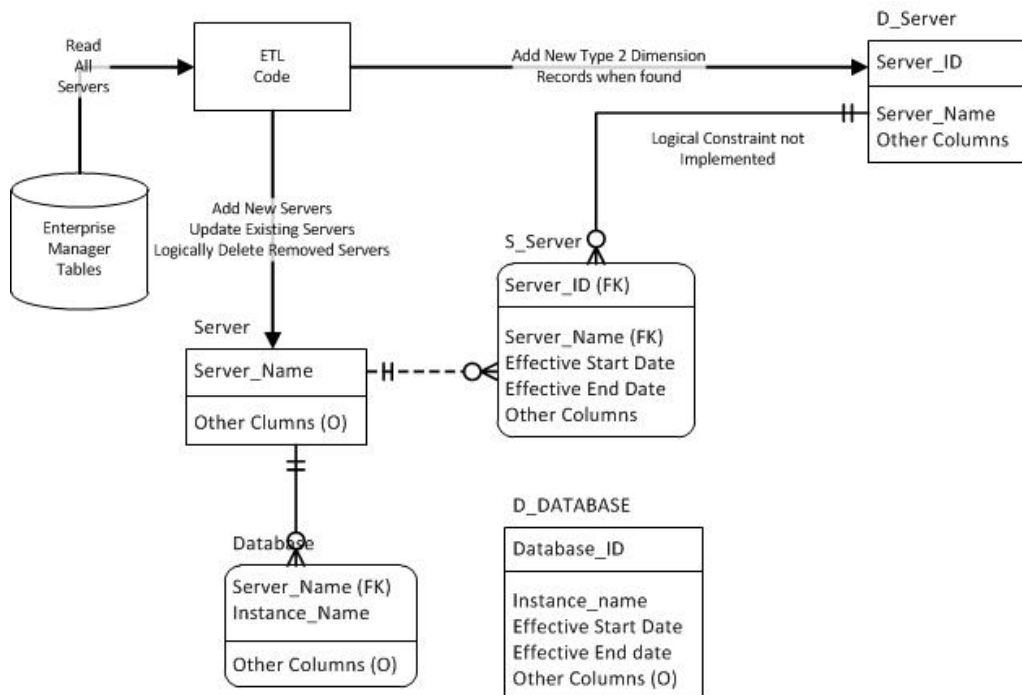


Figure 4: Dimensional Table ETL Pull based on the Server and Database table

- Procedure: Pulling Fact Table Data**
1. Determine which data needs to be read from the source system as identified by the data in the index table
 2. Read (Extract) only the data from the Database as determined in the prior step
 3. Transform the data into the fact tables format (this may include aggregation of data if the fact table records do not represent distinct data points from the source database) and load into the Fact staging table
 4. Read and load the Fact Staging Records into the fact table by identifying the correct dimensions the fact record is tied to
 5. Delete the staging table records used to load the fact table
 6. Update the load index to identify the correct location to start loading data from this database during the next fact table pull

Figure 5: Procedure: Pulling Fact Table Data

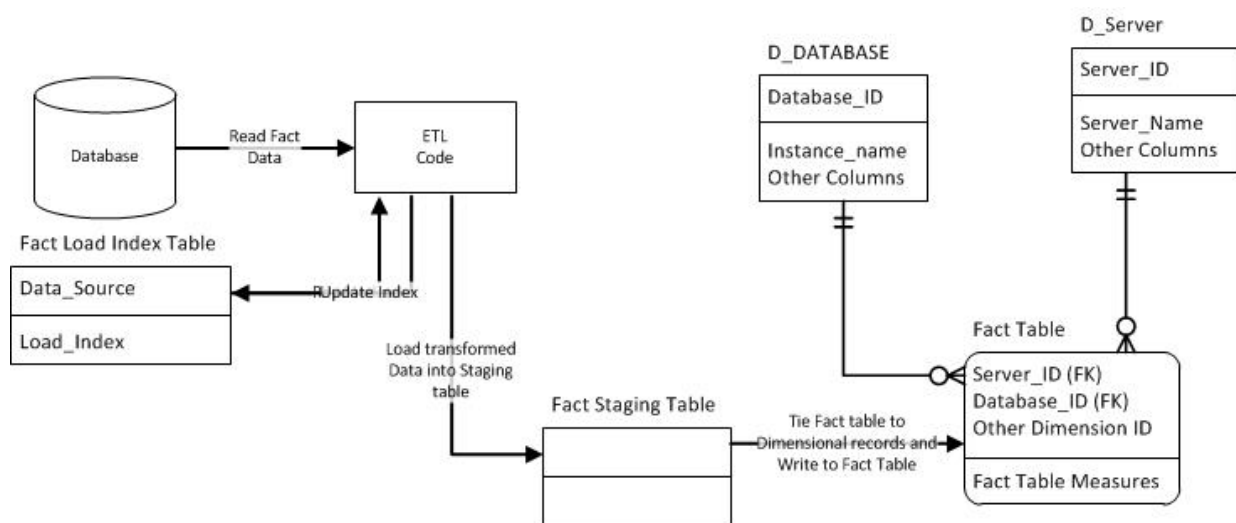


Figure 6: Fact Table ETL Pull based on the Server and Database tables

2.2.2 Data Marts defined in the DBA Data Warehouse

Sherwin Williams has currently implemented ten data marts in *the DBA Metrics Repository*. Table 2 shows a list of these data marts as well as the purpose of each data mart. In order to illustrate how the architecture for *the DBA Metrics Repository* works, the following section describes how data is loaded into the *Oracle Recovery Mananager (RMAN)* job data mart (Oracle RMAN is the tool Oracle DBAs use to backup Oracle databases).

2.2.3 Example: the Oracle RMAN Job Data Mart

One of the biggest responsibilities of a database administrator is to ensure that the database is successfully backed up on a regular basis. Backups serve two purposes: to help recover or restore the database if needed; and to create a copy of an existing

database for development or quality assurance purposes. A data mart storing *Oracle RMAN* job information was created because it became increasingly difficult to manually validate if scheduled backup jobs had successfully run.

Some of the metrics this data mart provides are:

- What percentage of RMAN jobs have run successfully in the last week and in the last month?
- Which databases do not have current backups that could be used, and how many days has it been since they were backed up?
- What is the average size of a full backup for a given database?
- What is the average size of an incremental backup for a database?
- What is the average time it takes to back up a database?

Table 2: A list of Data Marts defined in the DBA Metrics Repository

Data Mart	Purpose
Server I/O and CPU Utilization	<ul style="list-style-type: none"> Growth and Trends of a Servers CPU wait percentages and I/O operations over time Data used to: assist in capacity planning; identify when server is highly utilized
Storage Mount Point Growth	<ul style="list-style-type: none"> Growth of used and allocated storage over time by mount point by server Data used to: identify when and how much storage will be needed by current trends (for Capacity Planning)
Oracle Database Session Rollup	<ul style="list-style-type: none"> Aggregation of Oracle Session data by database by fifteen Minute interval Data used for: User access audits; Investigate Performance Issues; Performance feedback to developers; Investigating database session resource utilization
Oracle Database Unique Session	<ul style="list-style-type: none"> Number of unique sessions by database by fifteen minute interval Data used for: Capacity Planning; User Session Audits; Application configuration planning; identifying which (user, programs) have the most sessions in the database
Oracle Database System Change Number	<ul style="list-style-type: none"> Growth of the Oracle System Change Number (SCN) for all databases Data used for: to ensure that the issue described in [13] is not occurring
Oracle Database Data File Statistics	<ul style="list-style-type: none"> Amount of used and allocated bytes for each data file defined in an Oracle Database; various stats on the database interaction with the database (number of physical reads and write, average block write time, etc.) by database instance by day Data used for: Capacity Planning; ensure all tablespace have enough space to operate; Investigations for database performance issues related to disk I/O
Oracle ERP Concurrent Job Data	<ul style="list-style-type: none"> Data about each concurrent job run in every Oracle ERP system including completion data and any captured session data for that job Data used for: Concurrent Manager tuning; Investigating Concurrent Program performance issues; Identify amount of resources used by a typical execution of this concurrent job
Oracle ERP Hourly Run Concurrent Jobs	<ul style="list-style-type: none"> Data about the number of concurrent jobs run by Oracle ERP system by hour Data used to: ensure data not lost when data is purged from ERP system; Used as a rollup of Oracle ERP Concurrent Job data mart for dashboard display purposes
Oracle RMAN Jobs	<ul style="list-style-type: none"> Data about all <i>Oracle RMAN</i> backup jobs per Database Data used to: Identify issues with <i>Oracle RMAN</i> jobs; identifying which database are having backup problems; Validate that backups are occurring on the database as requested by the service level agreements with the business
Oracle RMAN Backup Set Used Storage	<ul style="list-style-type: none"> Data about the amount of space each <i>Oracle RMAN</i> backup set (file) is using Data used to: Identifying the amount of space each databases need to backup; Capacity Planning (to ensure that enough storage space exists on server for projected backup storage growth)

The Sherwin Williams development team chose to use the Kimball data mart model approach [12] in designing the DBA metrics data warehouse because this model is designed to easily answer the metrics defined above. In the *Oracle RMAN* data mart, the dimensions *D_CALENDAR*, *D_TIME_OF_DAY*, and *D_RMAN_JOB_STATUS* are predefined dimensions representing date, time, and the various job completion status values which are found in an Oracle ERP system. The dimensions *D_DATABASE* and *D_SERVER* are Type 2 Slowly Changing dimensions. The data found in these dimension are coming from the *Enterprise Manager* repository. The fact table *F_RMAN_JOBS* is loaded from the database where the

backups are run, because this data does not reside in the *Enterprise Manager* database. Figure 7 shows the fact and dimension tables that were created to store *Oracle RMAN* information, and provide the data for the metrics defined above.

3 DASHBOARD

Initially the data stored in the *DBA Metrics Repository* was only accessible through hand-written SQL queries. Few Database Administrators had time to learn efficient ways to access the data which was stored in the data mart to assist their daily tasks. There was an

urgent need to provide an easy and efficient way for the DBAs to be able access and utilize this data.

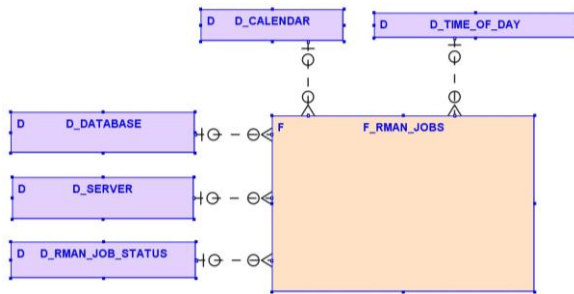


Figure 7: Oracle RMAN Jobs Data Mart

In the November 2010 TDWI [14] Fall conference session [15] the director of research at TDWI, Wayne W. Eckerson presented a session on dashboards for BI systems. He made a statement that only twenty percent of the potential users of a system would be power users in a BI system and directly query the data. The remaining eighty percent of users would need a dashboard in order to use the data collected by the BI system.

At Sherwin Williams, the development team developed a dashboard on top of *the DBA Metrics Repository* to make the collected data more easily accessible to the DBA team. Not owning any dashboard tools, the team created their dashboard in Oracle's Application Express tool. A high level dashboard page was created for each unique data mart in *the DBA Metrics Repository*. Table 3 lists four of the major *DBA Metrics Dashboards*; the information contained in each dashboard; and the information the DBA can drill into. Screenshots of each dashboard are shown in Figures 8 - 11. Additional dashboard pages have also been created to help administer *the DBA Metrics Repository & Dashboard*. The creation of the dashboard has enabled most DBAs to access the data in *the DBA Metrics Repository* and to use that data to help them in their daily DBA tasks.

DBAs can drill into each system displayed on the dashboard. This allows each DBA to find more specific information about a system. The high level *Oracle RMAN Dashboard* contains basic information about each database that has RMAN backups. A series of high level data points are also provided to help the DBA see what is happening in that database. On the dashboard, the DBA clicks on the associated system to see more detailed data about the system. Figure 11 shows a portion of the top level page for a given database from the *Oracle RMAN* data mart. The dashboard page in Figure 12 provides more

information about how Oracle RMAN backups are performing in the database system on which the DBA clicked. Along with the additional information about the system, the DBA can further drill into the RMAN data by clicking on the different navigation buttons at the top of the page. These links allow the DBA to further drill into the RMAN data collected and stored in the RMAN fact table.

4 EVALUATION – CASE STUDY

The DBA Metrics Repository & Dashboard has helped DBAs become more efficient in providing services to Sherwin Williams. The DBA team has experienced numerous benefits from *the DBA Metrics Repository & Dashboard*. In this section, we chose two common DBA tasks performed at Sherwin Williams to illustrate the significant contribution of our BI system. The tasks are: i) ensuring that all databases always have sufficient unused space to allow new data records to be written; ii) to accurately perform capacity planning on all servers, storage devices, and databases for the systems we manage.

4.1 Managing Unused Space

One of the primary DBA jobs is to ensure that each database has enough free space to allow new records to be stored in the database. A database that runs out of free space will prevent new records from being inserted into the database. Applications using that database experience errors when writing data to the database. There are several traditional approaches that DBAs have been using to prevent this from occurring.

The first method is to log into each database and run a series of scripts to compare the amount of allocated space against the amount of used space in the database. If the amount of used space approaches the amount of allocated space, the DBA would be required to add additional free space to the database to prevent the database from running out of unused space. The DBA could also store this information in a spreadsheet and compare it with the data collected the last time the DBA performed this check on the database. This step allows the DBA to determine the amount of storage that database will need in the next month, next quarter, or the next year. The DBA can compare the estimated amount of database storage needed to the current amount of database storage. If the estimated amount of storage is greater than the current storage, the DBA knows they need to increase the allocated storage for the database.

Table 3: A Summary of Four DBA Metrics Repository Dashboards

Dashboard	Information in Dashboard	Drilldown Areas
Oracle Database (See Figure 8)	<ol style="list-style-type: none"> 1) All Enterprise Manager monitored databases 2) Aggregation of Database Session Information 3) Database Disk Storage Information 4) Database Internal Identifier Changes, System Change Numbers Changes (ensure transactional integrity), Database Backup (RMAN) Summary Information, Database Server Memory Configuration 	<ol style="list-style-type: none"> 1) Database Session Information 2) Database Disk Storage Information 3) Database backup information
Server (See Figure 9)	<ol style="list-style-type: none"> 1) All Enterprise Manager monitored Servers 2) Storage Mount Points (Oracle ASM storage and traditional Server storage (SAN, NAS)) 3) Server Wait and I/O Utilization 4) List of Databases running on Server 	<ol style="list-style-type: none"> 1) Storage Mount Points 2) Server Wait and I/O Utilization
Oracle Applications (See Figure 10)	<ol style="list-style-type: none"> 1) All Oracle Application Systems 2) Information about each Oracle Application System 3) Which servers and Oracle Databases the Oracle ERP system is deployed on 4) Resources and time associated with Concurrent Managers 5) Concurrent Programs, and Concurrent Jobs 	<ol style="list-style-type: none"> 1) Overall Resource utilization and job completion success by: whole system, Concurrent Manager, or Concurrent Program 2) Tools to help identify concurrent programs that are running outside the defined min and max time for each concurrent manager
Oracle Database Backup (Oracle Recovery Manager - Oracle RMAN) (See Figure 11)	<ol style="list-style-type: none"> 1) Listing of all Databases using Oracle RMAN backups 2) Estimated recovery window for database 3) Instance backup sets can be applied to; Information about last backup, last level 0 (full)backup, and last level 1 (incremental) backup; Backup job information 4) Listing of Backup files from current and prior backups still active in system 	<ol style="list-style-type: none"> 1) Backup Job Information 2) Storage Growth of Backup Files

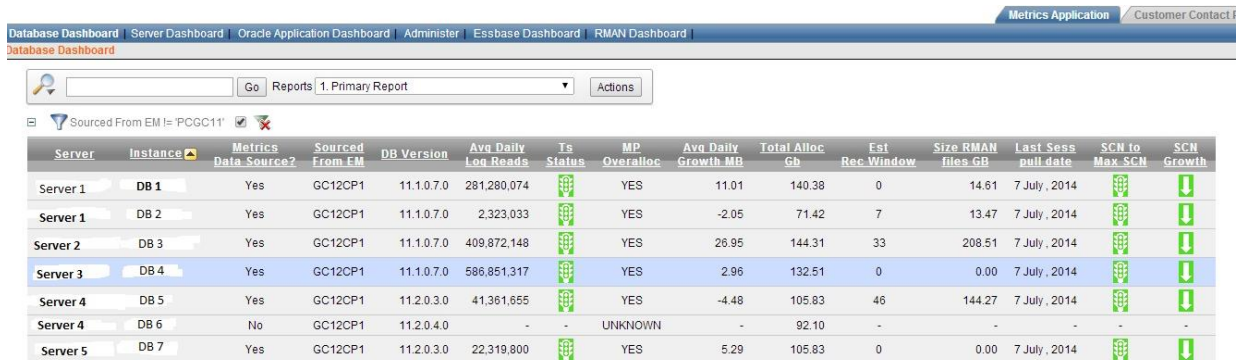


Figure 8: Oracle Database Dashboard

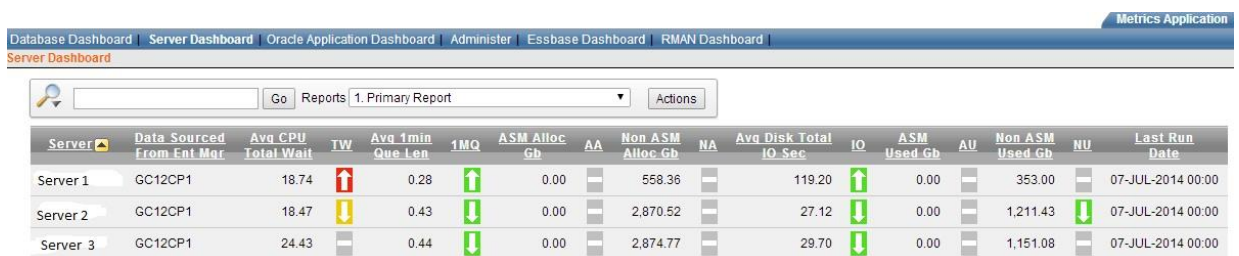


Figure 9: Server Dashboard



Figure 10: Oracle Application Dashboard

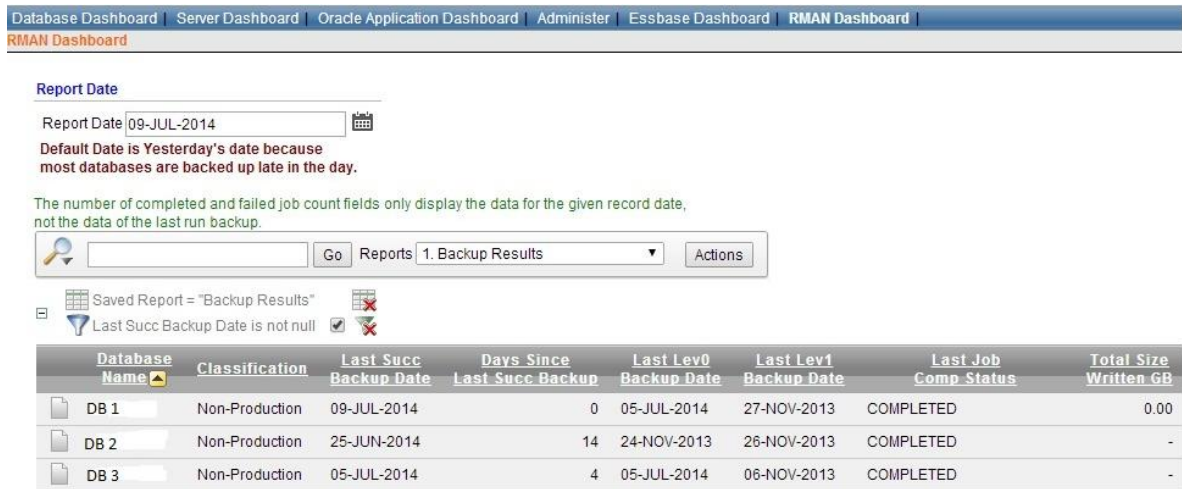


Figure 11: RMAN Jobs Data Mart

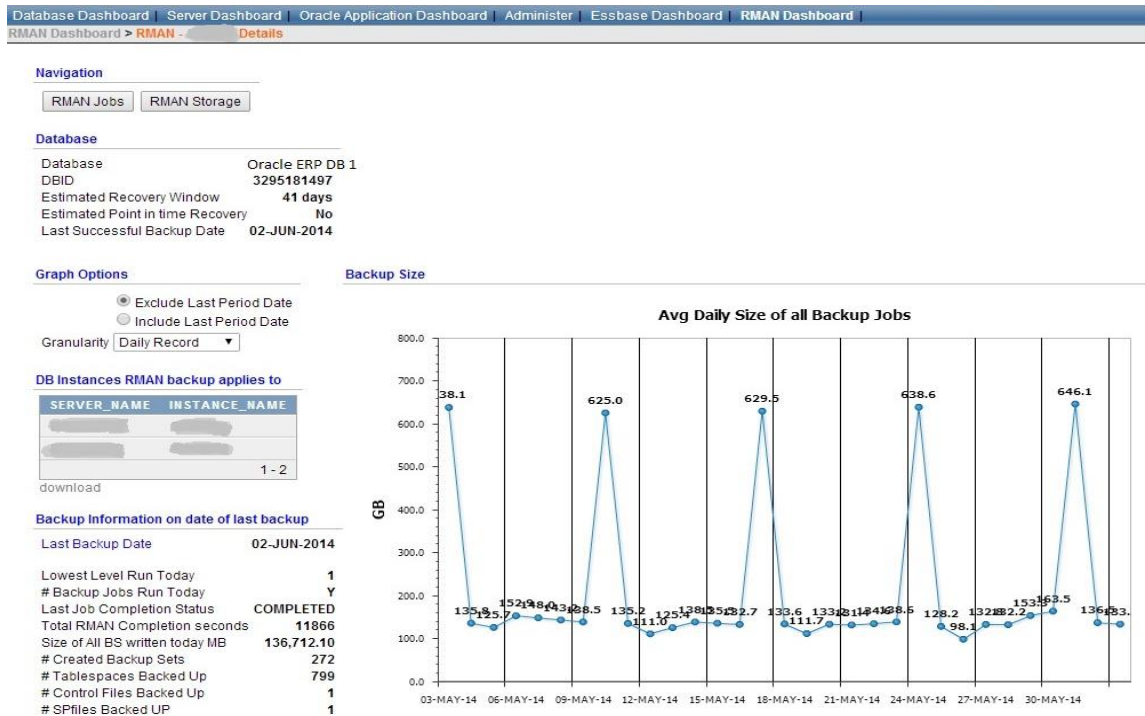


Figure 12: Top Level Oracle RMAN DBA Page

There are several limitations of this method, although it has been effective in preventing a database from running out of unused space. The main limitation is that it is a manual process which requires DBAs to manually log into the database to do this analysis frequently. This method becomes inefficient for the DBAs when the number of databases they manage grows rapidly.

The second method is to use the alerting features of a database monitoring tool like *Enterprise Manager*. DBAs need to configure the monitoring tool to send an alert when the percentage of used space to allocated space rises above a specific value (often 90% or 95% utilized). When the DBA receives the alert, he logs into the database and add additional free space to ensure the database has enough free space for future operations.

Most DBA teams use the second method when they administer a large number of databases, but this method also has its limitations. One limitation is caused by creating an alert based on a metric of a percentage of utilized data in the database. For example, the size of unused space in a 90% utilized database is a small amount for a 100 Megabyte sized database, but a large amount for 1 Terabyte sized database. (E.g. 10 MB for the 100 MB database as compared to 100 GB for the 1 TB database). This size difference makes it challenging for the DBA team to create a single alert using this metrics against database with different sizes and growth patterns. Another limitation is that the DBA learns about the issue, through an alert. Alert messages are generally delivered to the DBA through an e-mail, text message, or sometimes by a page. If the alert message is surrounded by hundreds of other alert messages (or the alert is seen but not addressed immediately because the DBA is working on other critical tasks), it is most likely the DBA cannot fix the issue immediately.

These commonly used methods are not 100% successful in preventing this issue from occurring. With the creation of *the DBA Metrics Repository* and the Oracle Database Data File Statistics data mart, the Sherwin DBA team was finally able to develop a method to prevent this issue from occurring.

The metrics development team was able to write a script against the fact table to determine the average daily growth in space for each database by month, quarter, and year. With this data they can accurately estimate when the amount of used space exceeds the amount of allocated space. With this script, they changed their standard operating procedures. Currently this script runs every day against the data stored in *the DBA Metrics Repository*. Any database which is estimated to fill its current unused space in the next

thirty days will have the amount of allocated space increased. The script provides the average daily growth in storage for this database. With this daily growth information, DBAs can easily calculate the exact amount of storage to add to the database. Since this Standard Operating Procedure has been implemented at Sherwin Williams, no database that has been monitored by *the DBA Metrics Repository* has run out of free space. The above task is normally done by directly querying *the DBA Metrics Repository*, but the information that these scripts use have also been added to the *DBA Metrics Dashboard* for each database. Figure 13 shows an example of the information presented to the DBA.

4.2 Capacity Planning

Capacity Planning is identifying the amount of computer system resources (such as servers, memory, storage, databases) that are needed in order to meet current and future demand. At Sherwin Williams, capacity planning is usually done when budgets are prepared for next year. The DBA team needs identify the number of additional resources needed for next year. An estimated cost for these needed items will be added to the next year's computer hardware budget. The DBA team is responsible for providing this data to ensure that we have enough resources to meet the expected system growth next year.

Capacity Planning requires DBAs to identify the expected growth for each system. Prior to *the DBA Metrics Repository*, the DBA team had no tools to calculate the growth of each system they administer. The task required manually gathering data from each system. This data includes the amount of resources used on each system (e.g. the average daily server CPU utilization, the amount of server memory used by the system, the amount of disk storage needed to store the systems data.). Two sets of these data points would be needed, the current value, and a value from an earlier period of time. The time period varies between several months to a year. With these two data points, the average daily growth in resource usage can be calculated using the following formula:

$$Avg\ Growth = \frac{(Current\ Value - Prior\ Value)}{\#\ of\ days\ between\ collection}$$

This Average Daily Growth metric can then be used to estimate the amount of system resources that may be needed by each system in the next year.

Calculating accurate Capacity Planning using this method is a time-consuming and complex task. It was a task that often could not be done at Sherwin Williams

due to the time it took to manually gather the data. Another complicating factor was that Oracle's *Enterprise Manager* tool does not retain the data points we needed over a sufficient period of time, which prevents DBAs from accurately performing these calculations. As a result, the Capacity Planning process becomes an educated guess about the amount of additional resources needed in the next year. This method proved to be inaccurate in estimating the expected growth in resource usage for the database we administer.

With the creation of the *DBA Metrics Repository*, determining the amount of additional resources needed becomes easier to perform. With up to three years of data on our systems, DBAs can accurately identify the amount of resources currently used, and the amount of additional resources to cover operations through the end of the next budget year according to the current growth trends indicated on DBA Metrics Repository. This enables the DBAs to provide accurate Capacity Planning numbers quickly and effectively. Due to the complexity of the Capacity Planning work, the data used for Capacity Planning is not gathered from the *DBA Metrics Dashboard*. It is queried directly using SQL by Sherwin Williams DBAs.

5 RELATED WORK

Business Intelligence (BI) has become a strategic initiative and is now recognized by chief information officers and business leaders as instrumental in driving business effectiveness and innovation. BI tools help companies and organizations gain more insight, across more data, and drive better outcomes in every aspect of their business. Today, it is difficult to find a successful enterprise that has not leveraged BI technology for its business. For brevity, we focus on discussing some BI

applications for supply chain management, customer relationship management, and higher education.

BI tools for supply chain management [16, 17] offers comprehensive BI solutions customized to i) improve supplier management through better information on margins, quality, delivery reliability, and supplier market shares; ii) detect possibilities to attune inventory levels to the anticipated demand and thus reduce inventory costs and capital demands. Through the sharing of information about inventories, purchasing conditions, expected purchases, resource planning, distribution, delivery times and suppliers, BI solutions ensure that the supply chain can be well-organized and efficient.

BI tools for customer relationship management (CRM) system [18, 19, 20, 21] provide cost-effective solutions to improve customer profiling, measure the success of the company in satisfying its customers, and create a comprehensive customer relationship management. BI and CRM software consists of leading providers of BI systems such as Salesforce.com CRM, Microsoft Dynamics CRM, Oracle Siebel CRM, and SAP CRM. BI and CRM solutions are capable of gathering and analyzing data which is crucial to efficient company managing.

Business Intelligence Solutions for Higher Education [22] can enhance institutions to i) expand their use of advanced analytics and complex event processing to address more complex problems; ii) provide a greater return on investment. BI systems for higher education consist of leading providers of administrative systems such as SunGard Higher Education, Campus Management, and Oracle/PeopleSoft. The solutions are often based on a partnership with a toolset provider or an analytical application provider and include a data warehouse. The solutions are developed for and integrated with administrative systems.

Tablespace	Type	Used GB	Alloc GB	Max Growth GB	Available GB	Avg Daily Growth MB	Max Daily Growth MB	Max Growth Date	# days fill (Avg Growth)	# days fill (Max Growth)
METRIC_CONC_JOB_I	PERMANENT	539.11	555.45	680.00	124.55	776.87	1,057.00	18 June, 2014	185.71	136.49
METRIC_CONC_JOB_T	PERMANENT	46.38	157.75	320.00	162.25	81.60	467.06	23 June, 2014	3,433.51	599.89
METRIC_DIM_I	PERMANENT	50.79	52.50	64.00	11.50	-3.03	108.88	9 June, 2014	-	124.29
METRIC_DIM_T	PERMANENT	3.94	4.25	64.00	59.75	0.50	8.00	2 June, 2014	121,831.67	7,687.70
METRIC_OPP_I	PERMANENT	160.08	177.42	200.00	22.58	-224.22	580.75	9 June, 2014	-	70.38
METRIC_OPP_T	PERMANENT	159.25	173.05	224.00	50.95	-209.84	666.00	9 June, 2014	-	99.56
METRIC_RMAN_FACT_I	PERMANENT	2.34	2.50	64.00	61.50	1.51	10.00	14 June, 2014	41,891.27	6,313.90
METRIC_RMAN_FACT_T	PERMANENT	3.74	4.00	64.00	60.00	0.68	2.25	24 June, 2014	91,352.34	27,425.22
METRIC_RMAN_OPP_I	PERMANENT	0.74	0.78	64.00	63.22	0.13	1.19	20 June, 2014	499,016.07	54,548.32
METRIC_RMAN_OPP_T	PERMANENT	0.14	0.16	64.00	63.84	0.07	0.50	20 June, 2014	971,521.57	130,781.75
SYSAUX	PERMANENT	37.49	48.00	48.00	0.00	255.16	684.50	4 June, 2014	42.17	15.72
SYSTEM	PERMANENT	6.62	6.63	16.00	9.38	2.58	64.00	5 June, 2014	3,728.45	150.12
TEMP	TEMPORARY	512.00	512.00	512.00	0.00	0.00	0.00	26 June, 2014	-	-
UNDOTBS1	UNDO	21.99	32.00	32.00	0.00	84.55	2,298.06	24 June, 2014	121.18	4.46
UNDOTBS2	UNDO	2.84	16.00	16.00	0.00	-3.73	2,260.00	13 June, 2014	-	5.96

Figure 13: Tablespace Expected Growth Data

DBA Metrics Repository & Dashboard is the first BI system to improve DBA productivity, although some DBAs use Oracle Enterprise Manager to facilitate their daily tasks. Oracle Enterprise Manager [23] is a tool which aids system administrators (e.g. DBA, Web Engineer, Administrators of Oracle's Engineered Systems) to perform administrative tasks. The administrative tasks include: i) extending the amount of storage allocated to a database; ii) adding a new database instance to an Oracle Real Applications Cluster (RAC); iii) scheduling Oracle Recovery Manager (RMAN) jobs to run against an Oracle database. It also gathers information about the databases and the other systems it monitors to assist administrators' tasks. It also collects different metric data points that the administrators can use to identify performance issues, or to send out alerts (warnings and critical alerts) when one component of that system has hit a specific threshold (e.g. a database tablespace now has 95% of its allocated space used; an Oracle RMAN job failed to execute; a servers job run queue is backing up). Enterprise Manager also provides a set of views that can be used to query the data stored in Enterprise Manager.

One of the primary limitations of Enterprise Managers is that it is an administration tool, and not a tool to view, compare, and drill down into the collected data. Enterprise Manager only keeps the collected data in an un-summarized form for seven days, making it impossible to accurately perform capacity planning with just the data stored in Enterprise Manager's repository. Another limitation of Enterprise Manager is its alerting system. Its default alerting thresholds results in the DBAs receiving too many alert messages. While it is possible to change these alert thresholds, they need to be done individually for each database, which is challenging to do for companies that have a large number of databases.

6 CONCLUSIONS & FUTURE WORK

In this paper, we presented the first Business Intelligence system to improve DBA productivity. *The DBA Metrics Repository & Dashboard* has been well received by Sherwin Williams Database Administrators. It has assisted the DBA team in its ability to improve the services and increase the trust between the company's different business units and the DBA team. It has i) enabled the DBA team to quickly determine which databases needed work by a DBA without manually logging into the system; ii) helped the DBA team and its management to easily answer other business users' questions without using DBAs' time to research the issue; and iii) helped the DBA

team to provide the business data for unanticipated audit request.

The DBA Metrics Repository & Dashboard system has also succeeded because: the system has been easy for the DBA team members to use; it allows for the easy addition of new databases along with the automated creation of the ETL jobs to run against those database; the system gracefully recovers from unexpected data source errors so that no intervention is needed to ensure that the next scheduled ETL job is run. Without these features *the DBA Metrics Repository & Dashboard* could not have succeeded.

The Sherwin DBA leadership continues to find value in this system and continues to ask for additional features to be added. Some of future projects that will be implemented in this system are:

- Better estimation on the growth of server storage to ensure each server has enough storage to meet current growth patterns;
- Collecting additional information about other parts of technology used by the Oracle ERP system (such as information about when users log into the system and their experience with the web application components of the system);
- Add data about non-oracle database and web application servers to have a larger picture of planning for combined systems;
- Adding other status indicators to better help the administrators see where potential issues exist;
- Create a new data mart to analyze the Oracle *Enterprise Manager* alerts to see trends, configuration issue, and identify potential configuration bugs in the Oracle *Enterprise Manager* system.

Despite the success of *the DBA Metrics Repository & Dashboard*, the full benefits of this system have yet to be realized. Most DBA team members are only just starting to understand the full capabilities of this system. DBA's have only just begun to utilize the data available to them in *the DBA Metrics Repository & Dashboards* in their day to day administrative tasks. The continued use of this system will become more essential as it will become impossible to continue to grow the DBA team at the same rate as has been done over the last decade.

ACKNOWLEDGEMENTS

The authors thank Sherwin Williams DBA team manager Mike Caldwell for persistent support. This project has also benefited from the support and feedback from the other Database Administrators at

Sherwin Williams. We would like to thank anonymous reviewers and editors for their valuable comments.

REFERENCES

- [1] Database Administrator Review 2014: <http://computer-careers-review.toptenreviews.com/database-administrator-review.html>.
- [2] TeraData: <http://www.teradata.com/?LangType=1033>.
- [3] TDWI: <http://tdwi.org/Home.aspx>.
- [4] DAMA International: <http://www.dama.org/i4a/pages/index.cfm?pageid=1>:
- [5] Sherwin-Williams: <http://www.sherwin-williams.com/>.
- [6] Oracle Exadata: <http://www.oracle.com/us/products/database/exadata/overview/index.html>.
- [7] H. J. Waston, and B. H. Wixom: "The Current State of Business Intelligence," IEEE Computer, 40(9): 96-99, 2007.
- [8] R. Kimball: "Data warehouse toolkit: Expert Methods for Designing, Developing and Deploying Data Warehouses", Wiley, 1998.
- [9] Informatica: <http://www.informatica.com/us/>.
- [10] Oracle Data Integrator: <http://www.oracle.com/technetwork/middleware/dataintegrator/overview/index.html?ssSourceSiteId=opn>.
- [11] InfoSphere: <http://www-01.ibm.com/software/data/infosphere/>.
- [12] R. Kimball, and M. Ross: "The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling", 3rd Edition, Wiley, July 2013.
- [13] TechWorld: <http://features.techworld.com/data-centre/3331020/crucial-oracle-flaw-revealed-but-is-the-problem-solved/>.
- [14] W. W. Eckerson: "A Practical Guide to Analytics: Putting People, Process, and technology to Work to Deliver Deeper Insights", TDWI Fall Conference, Nov. 7, 2010.
- [15] W. W. Eckerson: "Performance Dashboards: Measuring, Monitoring, and Managing Your Business", 2nd Edition, Wiley, October 7, 2010.
- [16] L. Luhao: "Supply Chain Integration through Business Intelligence", International Conference on Management and Service Science (MASS), 2010.
- [17] N. Stefanovic, and D. Stefanovic: "Supply Chain Business Intelligence: Technologies, Issues and Trends", IFIP International Federation for Information Processing, 2009.
- [18] A. Habul: "Business intelligence and customer relationship management", International Conference on Information Technology Interfaces (ITI), 2010.
- [19] J. H. Lee, and S. C. Park: "Intelligent profitable customers segmentation system based on business intelligence tools", Expert Systems with Applications, 29(1):145-152, 2005.
- [20] C. G. Yee, M. F. A. Aziz, and S. S. Hasan: "Applying Instant Business Intelligence in Marketing Campaign Automation", In proceeding of 2th International Conference on Computer Research and Development, May 2010.
- [21] D. D. Phan, and D. R. Vogel: "A model of customer relationship management and business intelligence systems for catalogue and online retailers". Information & Management, 47(2):69-77, March 2010.
- [22] M. B. Piedade, and M. Y. Santos: "Business intelligence in higher education: Enhancing the teaching-learning process with a SRM system". In Proceeding of 5th Iberian Conference on Information Systems and Technologies, pp. 1-5, June 2010.
- [23] Oracle Enterprise Manager: <http://www.oracle.com/us/products/enterprise-manager/index.html>

AUTHOR BIOGRAPHIES



Eric A Mortensen works at Sherwin-Williams as a Data Architect where he helps developers with database table design. Eric is the primary designer and developer of *the DBA Metrics Repository & Dashboard*. Eric also teaches database theory at a local branch of ITT Technical College in Strongsville, Ohio. Prior to Sherwin-Williams Eric worked at Everstream where he helped design and develop BI reporting systems on cable companies Video on Demand systems. He also worked at Royal Appliance as an Oracle DBA, and as a Data Architect and Software Engineer on the software for blood diagnostic equipment at Bayer Diagnostics. Eric is also Vice President of the local Oracle user group "Northeastern Ohio Oracle User Group" (NEOUG) where he helps with group operations and planning for the groups annual Great Lakes Oracle Conference.

Eric has spoken at several NEOOUG quarterly meetings and once at the Great Lakes Oracle Conference in May of 2013. Eric received a Masters of Computer Science from Brigham Young University where he studied database theory and software engineering in 1991. Eric received his Bachelors of Computer Science from Cleveland State University in 1989.



Dr. En Cheng received her Ph.D. degree at Case Western Reserve University and joined The University of Akron as an assistant professor in 2012. Cheng had the opportunity to experience internships in diverse research centers, including Microsoft Research Asia, IBM T.J. Watson Research Center, and Cleveland Clinic Foundation. Her current research interests include Data Integration, Database Systems and Applications, Mobile Applications, Semantic Web, Business Intelligence, Bioinformatics, and Bio-Medical Informatics.