

Enterprise Data Warehouse Governance Best Practices

Nayem Rahman, Intel Corporation, Hillsboro, OR, USA

ABSTRACT

Maintaining a stable data warehouse becomes quite a challenge if discipline is not applied to code development, code changes, code performance, system resource usage and configuration of integration specification. As the size of the data warehouse increases the value it brings to an organization tends to increase. However these benefits come at a cost of maintaining the applications and running the data warehouse efficiently on a twenty four hours a day and seven days a week basis. Governance is all about bringing discipline and control in the form of guidelines for application developers and IT integration engineers to follow, with a goal that the behavior of a data warehouse application becomes predictable and manageable. In this article the authors have defined and explained a set of data warehouse governance best practices based on their real-world experience and insights drawn from industry and academic papers. Data warehouse governance can also support the development life cycle, maintenance, data architecture, data quality assurance, the Sarbanes-Oxley (SOX) Act requirements and enforce business requirements.

KEYWORDS

Best Practices, Data Quality, Data Warehouse, Data Warehouse Governance, Data Warehouse Maintenance, ETL, Performance Optimization

1. INTRODUCTION

A data warehouse is considered one of the six major IT infrastructures (Weill, et al., 2002; Rahman, 2013a; Rahman et al., 2014) of business organizations. The warehouse expands as the customer base grows as new business requirements are identified. A data warehouse can contain hundreds of applications and thousands of objects. These objects include base tables, global temporary tables, base views, and business views for analytical purposes, stored procedures, macros, join indexes, and triggers.

Data warehouse governance is all about bringing discipline into coding, design and maintenance standards of a Data warehouse. It also helps in making sure that the design and development allows for reusability and code executions are optimal on the target platform. Such optimization (Rahman, 2013b) helps in running multiple applications on the data warehouse with the funded hardware and software capacity of underlying platforms. This reduces the TCO (total cost of ownership) for an IT organization and helps in capacity planning.

In this article we provide a set of data warehouse governance rules including use of automation tools and control criteria, which helps make ETL (Extract-Transform-Load) development (Kimball, 2013) and object migration flawless and the data warehouse environment stable, more efficient, and maintainable. Well executed governance can help an organization achieve the strategic objective of creating and maintaining a state-of-the-art data warehouse to support the ultimate goal of helping business executives make better strategic and tactical decisions based on the information stored in the warehouse (March and Hevner, 2005; Aiken et al. 2011).

DOI: 10.4018/IJKBO.2016040102

2. LITERATURE REVIEW

Data warehouses have the potential to provide business intelligence solutions for companies looking for competitive advantage (Rahman, 2013a). Fortune 1000 companies make strategic and tactical business decisions using the data warehouse as the central repositories of their enterprise data (Wixom & Watson, 2001). In an enterprise data warehouse new projects land over the years and a lot of enhancement and maintenance activities occur as part of day to day operations. All these activities require new objects installation or changing existing objects in the data warehouse. Given these activities how do we ensure that these day to day activities do not make data warehouse environment unstable, cause data quality issues, and impact analytical activities?

Based on real world observations of data warehousing projects implementation and past research findings (Arnott, 2008; Rahman, 2013a; Aiken et al. 2011; Bellatreche & Kerkad, 2015; Rabuzin, 2014; Zolait, 2012) the authors have determined that certain key areas of data warehouse activities need to be governed in a disciplined way. The authors believe that data warehouse objects development, installations, measurement, data quality monitoring, performance monitoring are critical for data warehouse implementation and maintenance. All these are needed to ensure that an organization can develop superior firm-wide IT capability to successfully manage their IT resource and realize agility (Lu and Ramamurthy, 2011; Mithas et al. 2011; Rahman et al., 2011; Roberts and Grover, 2012; Akhter & Rahman, 2015).

Data warehouse implementation has been a research topic for more than a decade. Most previous work on data warehousing focused on design issues (Rahman, 2013a; Rahman, 2014), data maintenance strategies in connection with relational view materialization (Rahman, 2013b) and implementation aspects (Rahman, 2010a; Rahman, 2014). A lot of research work has been done in the field of data warehouse refresh using ETL (Extract-Transform-Load) tools, with different alternative tools being proposed (Simitsis, et al., 2005). Significant amount of research work has also been done to address the issues of data inconsistency and quality (Ballou & Tayi, 1999; Rahman, 2013a). Towards this endeavor our work we focuses on identifying the best practices that can be followed in building and maintaining an enterprise data warehouse. Following the best practices will help in the maintenance, stability, and quality data refreshes for the data warehouse implementation. A consistent and standard enforcement of these best practices are referred to collectively as Data Warehouse Governance.

Data Warehouse Governance has become an important research agenda lately. This is largely due to rapid growth of data in every company's data warehouse. To the best of our knowledge, there are few publications (Watson, et al., 2004) available on data warehouse governance. They talk about some aspects of governance parameters, for example, the organization structure (Watson, et al., 2004). We have in turn chosen to address issues from the standpoint of ETL development, batch cycles refresh methodologies, data warehouse performance, stability and maintenance of data warehouses. There are other critical parameters influencing governance for a data warehouse implementation such as Corporate, Regulatory and Information Technology levels of governance. In our approach we focus on ETL design (Kimball, 2013) and development standardization, performance optimization of ETL objects as well as batch cycles, controlling of ETL objects so they follow certain standards and run efficiently. We also focus on metadata driven batch cycle refresh.

3. EVOLUTION OF A DATA WAREHOUSE

As time passes data tends to grow in a data warehouse to a very large volumes. Then there comes the significant challenge of maintaining a data warehouse efficiently in terms of space, performance, scalability, and stability (Feinberg and Beyer, 2010; Rahman, 2010b, Rahman, 2013b). Figures 1, 2 and 3 represent how a real-world production DataWarehouse has evolved over a period of time in terms of space occupied by databases/ DW subject areas, active users and database quantities.

Figure 1. Growth of space of a production data warehouse

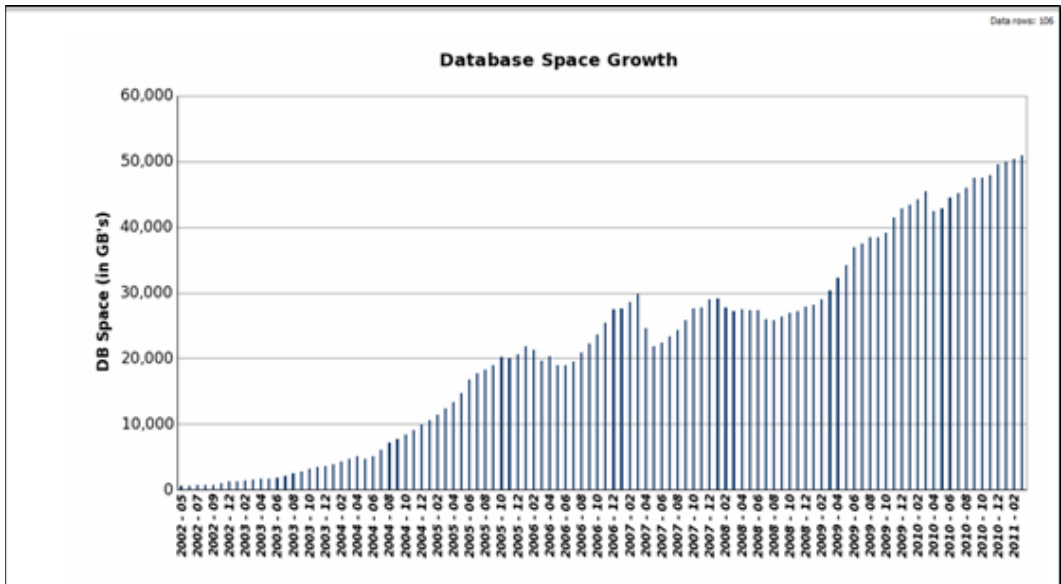


Figure 2. Growth of users in a production data warehouse

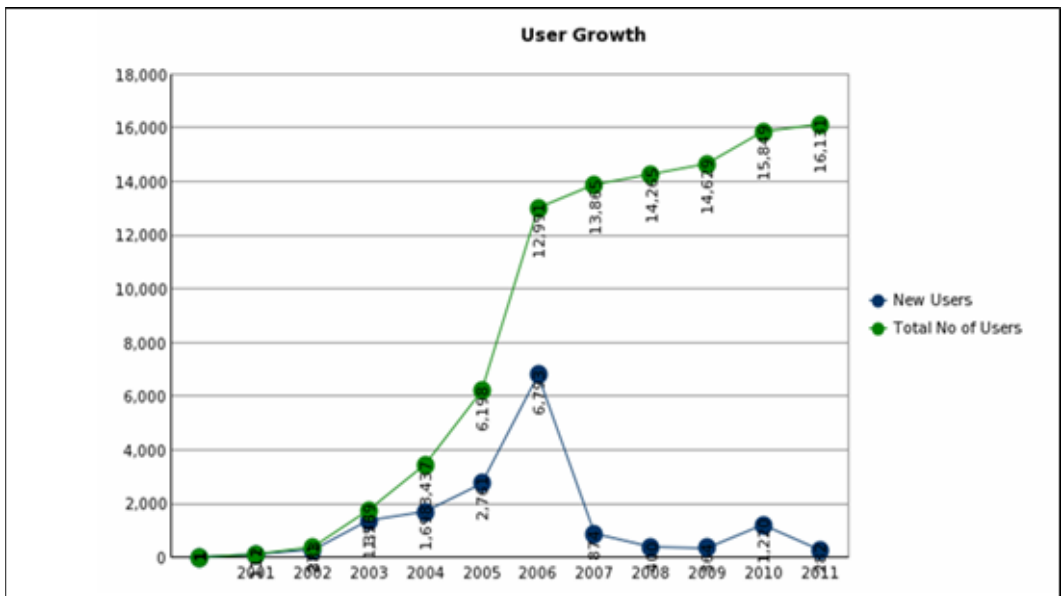
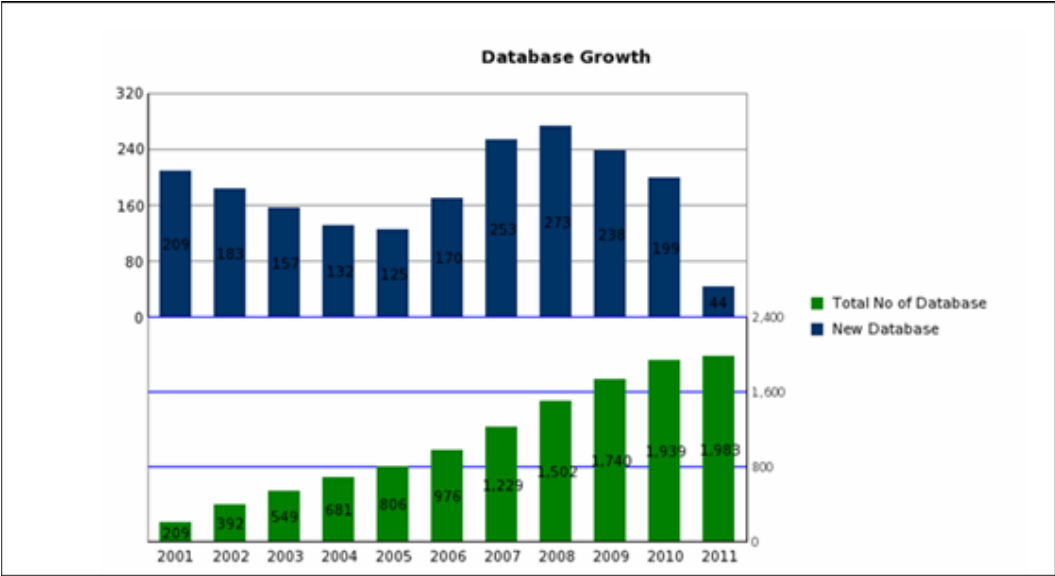


Figure 1 shows the growth of data in a data warehouse over a period of time. The data warehouse is not merely a database. It consists of hundreds of subject areas. An enterprise data warehouse is a central repository of an organization's data. As a result organizations gradually bring operational data from heterogeneous sources including enterprise resource planning (ERP), different relational databases, flat files, xml files and spread sheets. As data grows in tables query performance comes into question. In order to makes sure performance is reasonable the data architects need to make sure

Figure 3. Growth of databases in production data warehouse



logical and physical design of tables are carefully designed (Rahman, 2013b). The data warehouse application developers need to make sure table indexes and SQL (Stonebraker, 2012) are written such as way that performance is optimal given data growth in tables.

Figure 2 shows cumulative increase (the curve on the top: in green) of total number of users in the data warehouse. The other curve (the curve on the bottom: in blue) shows new users added each year on top of existing total number of users prior to that year. The vertical line (Y-axis) represents the number of users and the horizontal line (X-axis) represents period in terms of years. The number of report users and others in the BI community exponentially grow as they find there is timely and quality data available in a data warehouse. In order to make sure the user community can get query results against data warehouse tables/views within a reasonable time the data warehouse architects and administrators need to keep data warehouse healthy in terms of performance and the system is up and running (Rahman, 2013b). This also speaks for the importance of a data warehouse governance mechanism in place.

Figure 3 shows cumulative increase (bottom one: in green) of total number of databases in the data warehouse. The other curve (top one: in blue) shows new databases added each year on top of existing total number of databases prior to that year. As mentioned at the beginning of this section that in data warehouses tables are organized in terms of subject areas. Here databases and subject areas are synonymous. This figure indicates that a data warehouse that is successful grows rapidly. A data warehouse governance mechanism must be in place to make sure it is manageable and operational twenty four hours a day and seven days a week (Rahman, 2013a).

4. HIGH LEVEL DESIGN OF A DATA WAREHOUSE

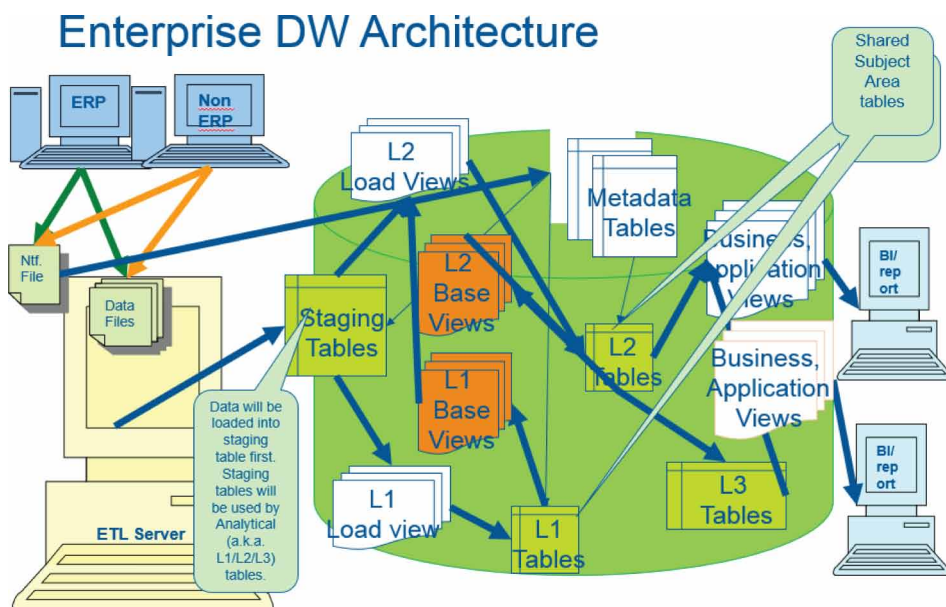
“A data warehouse is a subject-oriented, integrated, time-variant and non-volatile collection of data in support of management’s decision making process” (Inmon, 2005). Designing and rolling out to different subject areas and analytical environment of data warehouse is a complex process (Rahman, 2013a).

Figure 4 presents an architectural representation of an enterprise data warehousing environment. The data originates from operational databases. The source systems provide change timestamp for temporal data (Rahman, 2014). In a data warehouse source data is initially landed in staging databases. The data in staging database tables are accessed via L1 (layer-1) views. L1 view provide access to raw data in staging tables. Access to this raw data is restricted; users are not allowed to directly consume data in L1 views. This data is used to build L2 and L3 layers. An extract-transform-load (ETL) process is used in moving data from L1 to L2 and L3 tables. Users can access data using either L2 or L3 structures depending upon their specific needs. L2 subject areas are shared subject areas and used by any application(s) that needs data. L3 subject areas are typically dedicated to a specific application. Each L3 subject area provides an analytical environment used for reporting and Business Intelligence (BI) purposes (Herschel, 2012). Figure 4 also shows different kinds of views. The based views point to tables in different layers of subject areas. Business views point to base views. Users are not allowed to directly access the tables. They are provided with different kinds of views in order to allow them to access data depending on their business roles and data needs. So data access restriction is enforced via logic and filters applied to the business views.

5. BEST PRACTICES OF DATA WAREHOUSE GOVERNANCE

There are many aspects to look at as part of data warehouse governance. The following sub-sections will approximately follow the lifecycle of ETL (extract-transform-load) code as it is designed, developed, implemented and maintained in the production environment. Although many layers of governance will be covered, this is not an exhaustive list (Kimball, 2013). As with any discussion in regards to best practices, not all will be applicable to every data warehouse environment. A key takeaway should be there are levels of maturity (Sen et al., 2006) to be achieved by the deployment, implementation and maintenance of governance. As a data warehouse grows in size and complexity or begins to house sensitive data, governance must continually be adapting and growing.

Figure 4. An enterprise data warehousing environment (Rahman, 2014)



There are a large number of developers that work in a data warehouse. They often work on one another's code. Data warehouse object definitions (definitions of tables, views, stored procedures, macros, and indexes) change as part of enhancements or bug fixes. The versions of these objects also must be in sync across all data warehouse environments including development, testing, and production. As a result, it is important to store and control these objects via a source control tool as team foundation services (Rahman et al., 2010). When working on an object for enhancement or bug fix or moving objects from one DW environment to another these objects must be copied from the source control and an automated tool should be used to ensure that the correct versions of objects are selected via source control labels specified by tool's package builder.

To move objects from one data warehouse server (development to testing to production) to another an automated tools could be used as it will take significantly less time to install objects compared to a manual process (Rahman et al., 2010). ETL code development and changes must go through a rigorous review process such as a governance board to ensure they follow design, coding standards and performance optimization criteria. An automated tool needs to be used to enforce governance or built-in business rules such as SOX regulation compliance, password use compliance (an encrypted application password not a personal password is used) that increases security, and enforcement of naming conventions.

5.1. Data Warehouse Governance and Standards

Given the data warehouse is subject-area driven and consists of thousands of databases it is important to follow a set of guidelines and standards in naming objects, and maintaining coding standards and using proper documentations. This will also help with global search of objects from the data dictionary or metadata tables.

5.1.1. Data Warehouse Objects Naming Conventions

The purpose of following naming conventions in enterprise data warehouse is to make sure that subject area object names (tables, views, stored procedures, and macros) follow certain naming conventions. For example, table names should have prefix dim_, fact_, etc.; views should have prefix v_ followed by underlying table names; stored procedure names should have prefix pr_ followed by underlying target table names. A data warehouse can consist of subject areas with thousands of tables and columns. The column names should also use the same name across tables when the column represents exactly the same business object (Rahman et al., 2012). An example would be an organization name should always be "org_cd" regardless of where the table resides. There needs to be consistent abbreviations and column names that should be used throughout the entire data warehouse.

Following a naming convention helps to pull similar objects from data warehouse data dictionary via a global search. This allows for quick information retrieval and impact analysis (Rahman, 2010a; Rahman, 2013b). Naming conventions exist to ensure the integrity of the enterprise data warehouse environment and to ensure that object naming is consistent for better understanding, reusability, supportability and transportability (March and Hevner, 2007, Rahman, 2013a).

5.1.2. ETL Design and Coding Standards

The purpose of the ETL design and coding standards compliance is to ensure the integrity and stability of the data warehouse environment (Inmon, 2005). The ETL design process is to make sure that the ETL part of the process is not sub optimized for the sake of its efficiency. The ETL design also needs to take performance into consideration so that the reporting layer does not become inefficient as a result of performance issues. For example, to make load process efficient ETL design might prefer normalized tables but from reporting perspective side joining to so many tables might cause a performance bottleneck. By the same token to improve the report performance, if tables are designed highly de-normalized the ETL process might suffer from a performance hit during table refresh process (Rahman, 2013a). So, the ETL design must take into consideration the reporting environment side

and take a balanced approach. The ETL design must ensure that load process is not skewed. The ETL design also needs to make sure that the application and batch cycle is efficient from production support standpoint. Building an application with unnecessary complexity puts a significant burden on the production support team. The design review process should include addressing ease of supportability.

Design review is done before the start of ETL development. In the design review process stakeholders from different teams are invited to a design review meeting which may be face to face or virtual. The design review could also be off-line. Stakeholders may include product owners, systems analysts, data modelers, ETL developers, and the analytical community. Design review process makes sure subject areas and tables are landed at the appropriate subject areas in enterprise data warehouse. ETL design review makes sure the batch cycle does not become inefficient due to a design flaw, is leveraging the best source and target environments and will meet the business needs.

5.1.3. Batch Name Standards and Master Scheduling

In some data warehouses thousands of batch jobs run daily. Those jobs either refresh the target tables in staging databases (the as-is data from operational databases) or analytical databases (transformed and aggregated data) that are used by reporting and other analytical tools. Jobs are also run to extract data from a data warehouse and deliver it for use by downstream applications (Rahman, 2013a). All jobs need to be added to batch cycles scheduling tools so they are visible and manageable by a support team. For that purpose, scheduling standards need to be followed.

Batch jobs naming standards (using pre-determined structure and codes) will enable efficient monitoring and support. Using coding that matches job to support structures means failure can more easily be identified and routed to the respective developer team or organization. Adding information in the coding to communicate other pertinent information 'at-a-glance' also improves the supportability of the jobs.

Figure 5 provides an example of job naming standards. Batch jobs in different subject areas are run in varied frequencies based on service level agreement (SLA) with the application owners and customers. The SLA is an agreement that is established during the development process that sets the criteria for performance expectations with the application owners and customers (Rahman, 2013a). There is a constant balancing act to meet or exceed SLA goals while controlling costs driven by expanding capacity demands of a shared data warehouse environment. A valuable technique to help load-balance demand is to place batch jobs into low-consumption windows outside of user-driven and existing cycle-driven demand; sometimes referred to as Master Scheduling.

At its core, master scheduling requires knowing where existing demand peaks and valleys; the business requirements for query performance and availability of data (i.e. when cycles must finish) and the best estimate for demand/runtime for new cycles or queries landing. The day-to-day activities of performing master scheduling consists of maintaining up-to-date details of these three areas and negotiating with stakeholders of a given cycle or query to find the optimal run window. The optimal window will satisfy business need (SLA) while simultaneously not driving peak demand to the point where additional capacity must be purchased.

In smaller or more static environments this tracking and demand management may be accomplished within a spreadsheet while larger and more dynamic systems will require sophisticated monitoring and metrics managed with automation. Although solutions can vary widely, they will share these common elements: a record of reference for the expected consumption (when demand was requested; may be SLA documentation); a method for reading demand on the live-environment (when is usage actually happening); a means to attribute demand to the process which is driving it (process ID, application or user name, etc.); a way to negotiate with the owners of processes driving demand to find a more mutually beneficial time when conflicts occur. This can be a time consuming and complex process, but if performed with consistency the benefits in customer satisfaction and cost-avoidance will easily offset the effort.

Figure 5. Example batch job naming standard

Job Naming Standards

a	i	x	x	x	x	a	a	a	a	#	#	#	a
1	2	3	4	5	6	7	8	9	10	11	12	13	14

Columns 1 - 3

Column 4

Columns 5 - 6

Columns 7 - 10

Columns 11 - 12

Column 13

Column 14

Reserved - Acronym for Owning application or group

Reserved - Type of job (f = FileWatcher, s = Send, d = Delete c = Command, x = Box)

Reserved - Used for frequency and geographic identifiers (if applicable)

Four-letter acronym identifying the subject area

Numeric ID. Should use the same two-digit number for all jobs associated with a given target table

Identifies type of processing:

0 Start-up / initialization

1 FTP and/or unzip

2 Drop & create table

3 Load / update table

4 Drop & create table

5 Transformation

6 Collect statistics

7 Multi-table transformation

8 Preparation of inactive table

9 Clean-up / completion

a – m for standard job; n – z for "hook" job

5.2. Performance Perspectives of Data Warehouse Governance

Managing a data warehouse performance is so important in order to keep the data warehouse stable and running. To run batch cycles efficiently and in a timely fashion it is important to ensure that ETL code and jobs perform efficiently. On the other hand, from the reporting standpoint all reporting and business intelligence queries must run within a reasonable time that is acceptable to the data warehouse users and analytical community.

5.2.1. SQL Scorecard and Code Review Process

Often both the ETL and BI reports use complex SQL. In most cases the SQL runs against a huge volume of data in the data warehouse. As a result it is important to make sure SQL is efficient in both loading and reporting processes. One unique measurement ensuring SQL efficiency is to run the SQL code through a proposed SQL scorecard tool and processes which measure computing resources is used by SQL (Rahman, 2013b). In fact it measures parallel efficiency of SQL code in terms of CPU, IO and spool space used by the SQL code. In real-world applications in a large data warehouse the authors found it a very effective means to identify the inefficient SQL in development and testing phases. This allows for preventing inefficient SQL code in the production environment.

The SQL block is score carded based on SQL performance measurement criteria. The first six columns, in Figure 6, show the SQL performance in terms of CPU, IO and spool space usage and their respective parallel efficiencies. The remaining six columns in Figure 6 are auto populated to provide results if the SQL has failed in one or more of the following areas: CPU, IO or spool usage and PE (parallel efficiency). The last column in Figure 6 provides overall scorecard result such as 'Pass' or 'Fail'.

The scorecard results in Figure 6 show that the SQL failed the scorecard. To address this we redesign the SQL and split it into five SQL blocks after taking database parallel processing architecture into consideration. We made sure we're not exposing huge amount of data to joins unnecessarily. We redesign the index of the intermediary temporary tables and the final target table to avoid all-row scanning which is very expensive. Finally, we performed index-based join and update operations for select and update SQL to allow block by block operation as opposed to row-level operation.

Following improvements in SQL, Figure 7 shows that each of the individual SQL blocks passed scorecard. The scorecard process is performed for each individual SQL.

Figure 6. Scorecard results with failed score

CPU Evaluation		I/O Evaluation		Spool Evaluation		EXPLAIN Evaluation		System Rating			Overall Score
Total CPU	CPU Parallel Efficiency (%)	Total I/O	I/O Parallel Efficiency (%)	Total Peak Spool	Spool Parallel Efficiency (%)	Statistics on all Joins & Filters?	Joins or Filters on Derived Attributes?	CPU : I/O Ratio	Resource Usage Rating	Parallel Efficiency Rating	
1,642	71.65	1,231,559	31.5	5,618,799,616	5.71	YES	NO	1.33	FAIL	FAIL	FAIL
						***	***				
						***	***				
						***	***				
						***	***				

Figure 7. Scorecard results - pass

SQL Scorecard Results

Project Name:Capital_DRV

Developer Name:Wally Heaven

System used for Testing:TDEDW1

ETL

SQL-ID	Report Name (SQL)	CPU Evaluation		I/O Evaluation		Spool Evaluation		EXPLAIN Evaluation		System Rating		
		Total CPU	CPU Parallel Efficiency (%)	Total I/O	I/O Parallel Efficiency (%)	Total Peak Spool	Spool Parallel Efficiency (%)	Statistics on all Joins & Filters?	Joins or Filters on Derived Attributes?	CPU : I/O Ratio	Resource Usage Rating	Parallel Efficiency Rating
	pr_Fpo_lmt_rpt	8	79.01	53,118	95.4	40,359,936	91.24	YES	***	0.15	PASS	PASS
	pr_Fpo_lmt_rpt1	11	64.92	29,582	66.27	57,808,384	96.8	YES	***	0.36	PASS	PASS
	pr_Fpo_lmt_rpt2	170	86.93	179,584	87.82	1,643,832,832	60.73	YES	***	0.95	PASS	PASS
	pr_Fpo_lmt_rpt2 - UPDATE	10	70.7	31,369	77.99	1,327,616	93.14	YES	***	0.33	PASS	PASS
	pr_Fpo_lmt_rpt - FINAL INSERT	13	70.03	21,908	66.34	76,179,968	77.11	YES	***	0.59	PASS	PASS
		212		315,561		1,819,508,736		***	***	0.67	PASS	

There is a strong evidence of the benefits of software traceability (Neumuller and Grunbacher, 2006). The ETL code review process ensures that SQL code is readable, maintainable and follows coding and naming standards. The review process also makes sure that SQL code is optimized. A code review team administers all aspects of the code review process, communicates process updates and issues, provides consultation and recommendations on issues related to process, reviews code for standards compliance, makes sure best known methods (BKM) are followed, reviews SQL scorecard results, documents meeting results and updates and provides Go/ No-Go decisions. Application of design and code review in a real-world production data warehouse system has improved the quality of code and significantly contributed toward maintaining stable data warehouse environment. Data warehouse system quality positively influences information quality (Hwang and Xu, 2008).

ETL code must go through an ETL Governance Go/No-Go Code Review, and obtain a 'Go', prior to the code being migrated to the data warehouse environment. It is important to ensure 100% of the ETL code has passed SQL scorecarding and results are consolidated for review and documentation purposes. ETL code review checklist may be used to help consistency and repeatability, communicate expectations prior to review and improve the efficiency of the scoring and reviewing process.

The review process should utilize an ETL code review check-list. Participants can fill in appropriate fields as the meeting progresses, based on comments and required corrections necessary to ETL code. For each object reviewed, the team provides comments on adherence to coding and naming standards and solicits additional comments from other attendees. As output, the board grants 'Go' or 'No-Go' based on review of ETL objects and SQL scorecard results with action items documented if additional work is needed.

5.2.2. Performance Metrics Review and Improve

The purpose of capturing and publishing performance metrics for a data warehouse is to keep visibility on the overall performance and identify any slowness and missing SLAs. Publishing this kind of health check increases the confidence of the user-community of a data warehouse (Rahman, 2013b). An end-to-end health check of a data warehousing environment is needed and encompasses the operation source system (e.g., ERP), ETL tools, data warehouse (DBMS) platform and reporting environments (Reporting, BI, and data mining tools).







Figure 8 provides more detailed information about the performance of a data warehouse system. There are thousands of lines SQL executed in a data warehouse on a daily basis. SQL is executed via data warehouse-specific software such as stored procedures, views, SQL via reporting tools, SQL via extract tools, and SQL generated ad-hoc queries. The load jobs in batch cycles use data warehouse systems resources – CPU, IO, spool space (2010b). Reporting SQL and ad-hoc queries also uses huge amount of resources. In order to provide users a good query performance (Allen & Parsons, 2010) it is important that a data warehouse is up and running with enough systems resources available.

In Figure 8, we have identified several indicators under the first column, including system availability, system computing resources utilization, query performance, OLA performance with business partners and analytical community, and reporting environment performance. The second column shows guidelines for each indicator – green, yellow and red and the last column gives some additional details in terms business and query performance.

5.2.3. Reporting and Analytical Tools' Performance

In order to provide the data warehouse user community the best performance, two important things need to be done from a data warehouse governance standpoint. The data warehouse systems administrators or Database Administrators (DBA) need to make sure sufficient resources are allocated for use by the reporting and analytical tools to meet or exceed Service Level Agreement (SLA). Normally, this is ensured via priority scheduling for data warehouse systems resources. The second aspect of

Figure 8. Data warehouse platform indicators

	Indicator	Guideline	Details
	System Availability and Performance	Green – No unplanned downtime in last 24 hours Yellow – Slowness, deadlocks, skewing, TPA reset Red – Outage or node failure, resource constraints	
	CPU Utilization	Green – Below 85 Yellow – 85-89 Red – 90-100	24 hour Average CPU busy 74.70% Since midnight the average CPU has been 69.99%
	Canary Query	Green – under 1 second Yellow – 1-2 seconds Red – 2-5 seconds	DBC Average = .60 seconds (0 queries ran greater than 1 second)
	Business Partner OLA Performance	Green – 1-2 Yellow/Red OLAs Yellow – 3-4 Yellow/Red OLAs Red – > 4 Red OLAs	Note: Channel Management has complex dependency issues and is not part of this metric.
	System Availability	Green – no unplanned downtime in last 24 hours Yellow – Degraded system performance Red – Outage	
	BW System Availability	Green – no unplanned downtime in last 24 hours Yellow – Degraded system performance	

governance is to have ETL programmers work closely with reporting and analytical tools developers so the SQL (Adomavicius et al. 2011) they run via those tools is efficient. The data warehouse governance board for reporting tools must review each SQL being run by the reporting and analytical tools. The review process includes having ETL developers or report developers scorecard the SQL to measure the performance.

A code review team administers all aspects of the code review process, performing reviews on code for standards; ensures SQL has been optimized (Allen and Parsons, 2010), reviews SQL scorecard results, documents meeting results and updates; and provides Go/ No-Go decisions. Application of design and code reviews in a real-world production data warehouse system has proven to improve the quality of reporting tools' code significantly. Report SQL code must also go through an ETL Governance Go/No-Go Code Review, and obtain a 'Go', prior to the code being migrated to testing environment. Change control measures ensure 100% of the SQL code has passed SQL score-carding (Rahman, 2013b) and results are consolidated into a spreadsheet for review and documentation purposes.

5.3. Operational Aspects of Data Warehouse Governance

In day to day operations and maintenance of a data warehouse objects and code must be in sync among development, testing and production environment. A standard repository of objects and code needs to be maintained. In a data warehouse, subject area refreshes are performed via batch cycles. These batch cycles needs to be metadata driven to keep data warehouse operations simple and easy. A dedicated response team needs to be assigned to monitor, troubleshoot and keep the data warehouse cycles running per the SLA established with the application owners and the user community.

5.3.1. Objects Migration to Data Warehouse Environments

Installing objects and keeping them synchronized across all warehouses (migration) is often a challenge due to sheer number of objects and dependency complexities. For many IT shops object software installations are a manual effort (Rahman et al., 2010). There is a corresponding high cost due to long hours spent annually on this activity. Project timelines often do not factor in the time required for manual procedures. Developers can spend a disproportionate amount of time preparing installation and migration scripts which are prone to run time syntax errors. There is an additional cost for migrating objects multiplies each time developers have to rewrite scripts and re-migrate them. This contradicts the strategic objectives (better, faster, and cheaper) of many data warehousing projects (Rahman et al., 2010).

We advocate use of an automated tool which helps make object migration less error prone and therefore, more efficient and more effective (Rahman et al., 2010). Incorporating automated migrations can help an organization achieve the strategic objective of creating and maintaining a state-of-the-art data warehouse directly supporting the primary goal of helping business executives make better strategic and tactical decisions based on the information stored in the warehouse.

5.3.2. Data Warehouse Operations Team Involvement

In data warehouses it is typical for hundreds of applications to be running on different schedules for different business drivers. A data warehouse is therefore a highly shared environment and expected to be available on a twenty four by seven basis. Every single planned structural change must be communicated to the Operations team early on (Rahman, 2013a). This means that enhancements to existing subject areas or landing of a new subject area in the environment must be done in coordination with supporting teams. The developer community must communicate with support teams in terms of early engagement so that they know what are they going to support coming forward and can plan accordingly. One method is to have Operations representation in the design and code review processes as involved stakeholders.

Similarly, Operations team governance policies must include timely communication and stakeholder involvement in planning any activities which will impact the development lifecycle. This includes scheduled downtimes, patching, and software or hardware changing which could impact the design assumptions or performance of code. Including development distribution lists in incident communication and escalation processes is a simple step that can improve the ability of development stakeholders to mitigate impacts to their project timelines.

5.3.3. Metadata-Driven Batch Cycles

An efficient, flexible and general data warehousing architecture (Ariyachandra and Watson, 2010) requires a number of technical advances (Rahman, 2013a; Rahman et al., 2014). A metadata-driven batch cycle refresh of a data warehouse is one of these technical advances. Research suggests that data warehouse experience relatively high failure rates (Ramamurthy et al. 2008). Successful data warehousing is dependent on maintaining data integrity and quality in table refreshes. Metadata-driven refreshes plays a prominent role in this regard. Loading inconsistent data negatively impacts data quality and is likely to be a repeating issue if an efficient metadata model is not devised.

In data warehouses, each subject area is refreshed through a batch cycle (Rahman, 2010b). Under the batch cycle, jobs are run in different boxes to load the target tables in order of dependency on other tables in the subject area. Jobs in a subject area run under different conditions: some jobs load tables with full refreshes while other jobs refresh the tables incrementally; some other jobs in the cycle skip performing incremental loads if source data has not changed; some jobs are set to do incremental loads but end up doing full refreshes when the source has a large number of new records or the source or target table row count does not match after an incremental refresh is performed (Sugumaran, 2012). An ETL metadata model (Rahman et al. 2012) could be used to control the table load to automatically satisfy the load conditions from the options above.

The metadata model is based on several metadata tables, utility stored procedures, wrapper stored procedures, and full and delta load stored procedures for individual table loads (Rahman, 2010a; Sugumaran, 2012). With support from a metadata model the batch processing can be automated and cycle refresh could be performed by loading only those tables for which there are new records in the source tables. Inspection of the latest observation timestamp in a metadata data-table for each of the source tables referenced in the incremental load stored procedures detects the arrival of fresh data. The full and incremental load procedures are bypassed if the source data has not changed (Rahman, 2010a; Sugumaran, 2012). By providing the source and the target table last load timestamp, row count, and load threshold information, the model allows accurate incremental load processing and enables an automated decision about whether to perform full or incremental refresh.

5.4. Data Warehouse Governance and Data Quality

A data warehouse cannot survive if it cannot ensure data integrity and quality in the tables. For the sake of referential integrity it is important that data is in sync among the subject areas (that hold data relating to procurement, supply, finance, etc.). If customers lose confidence in data quality, the data warehousing might end up with a failed project. Protecting access to sensitive data is so critical after promulgation of SOX regulations a few years ago.

5.4.1. Enhancing and Maintaining Data Quality

As a corporation grows and evolves new business models and products, corresponding additions and changes to the data and structures in the data warehouse are necessary. Changes to the data warehouse to integrate new use cases are also triggered as the corporate community knowledge of data and analysis techniques evolves. Any growth in or changes to existing data and structures have a potential for negative impact on the quality of the data in the warehouse.

If the data in the data warehouse is unreliable, the analysis that a corporation depends on will be inaccurate, diminishing the value of the data warehouse and putting the corporation's success at

risk. It is easier and less costly to address quality during design and definition phases, it becomes significantly more difficult and expensive during maintenance or sustaining phases. This is why we have identified Maintaining/Enhancing Data Quality as one of the best practices to employ in the governance of a data warehouse. Data quality compromises can happen in various ways such as missing or incomplete data entry controls, duplication of data (Elmagarmid et al. 2007), inconsistent representation of data across dimensions, inconsistent usage/interpretation of the data by users and obsolete data.

One element to enforce data quality is having a good review team structure for the data (Watson et al, 2004). Watson et al. (2004) describe an organization model to enforce data quality in the warehouse using a hierarchical oversight team structure. At the top level is a Vice President Data Oversight Team that enforces the data warehouse direction alignment to the corporate direction. The next level is a Data Development Oversight Team which is responsible for addressing issues encountered during the implementation of the data warehouse. The actual development of the data warehouse is managed via a cross-functional team representing both the development team and the use-case experts. This is especially useful when adding new dimensions or analysis systems into the data warehouse.

Another element to enforcing data quality is to determine whether to address the data quality issue once it is detected (Ballou and Tayi, 1999). In this article, they advocate a systematic approach and provide a model to assess effectiveness at the onset of a data quality enhancement effort on a data warehouse. In their study, they identified a number of steps were identified that can help the success of data quality enhancement efforts. The first step is to clearly identify the organizational activities that the data warehouse is going to support and prioritize these activities so that the data quality enhancement investment can be focused on them. The second step is to clearly identify data sets that are required to support the identified activities. The required data set can be internal existing traditional data files, any external existing data files, or nonexistent data which can be created and set to support such need. The third step is to identify any critical problems with the data sets according to the dimensions of data quality guide.

Figure 9 shows some of the tools/methods to enforce data quality within a data warehouse life cycle. Using reports as a method of identifying data quality issues and driving cross-group standardization of data representation was identified as a benefit in the data warehouse implementation at Blue Cross and Blue Shield of North Carolina (Watson et al 2004).

5.4.2. SOX Compliance

Since its passing, the Sarbanes-Oxley (SOX) Act requirements have grown into one of the key constraints to which data warehouse implementations need to conform. It has been proposed that the enforcement of business rules be SOX compliant ((Agrawal et al., 2006; Brown & Nasuti, 2005) including automation of the enforcement of business rules. Password use should also be SOX compliant – for example an encrypted application password is used as opposed to a personal password. It is beyond the scope of this article to explore all the impacts SOX may have on governance of a data warehouse. Future study can go into depth on the ways to integrate legal regulations into day-to-day operations and therefore reduce cost and impact while accomplishing compliance.

6. CONCLUDING REMARKS

The focus of this article was to present data warehouse governance guidelines. These guidelines mainly come from the standpoint of data warehouse performance and stability. The ultimate goal is to foster confidence in report performance, query performance, data warehouse system availability, and information reporting quality.

We described the value in following naming standards on data warehouse objects. We emphasized the need to go through a rigorous review of ETL design to make sure performance and data integrity are optimized before beginning ETL design and programming and validated during testing with the

Figure 9. Tools and processes to maintain data quality in a data warehouse



illustrated SQL scorecard process. The overall goal of which is to optimize SQL blocks to best take advantage of the parallel processing architecture of the target DBMS. Other best practices discussed using automated tools to scorecard, run metadata-driven batch cycles, achieve SOX compliance, and install objects across all development, testing, and production data warehouse environments.

Having a data warehouse governance process in place and actively managed will greatly support the necessity of maintaining a stable data warehouse for an organization. Properly designed and implemented data warehouse governance enables success across the ETL development life cycle, environment maintenance, data quality assurance, Sarbanes-Oxley (SOX) Act requirement (Brown & Nasuti, 2005) and delivery to business requirements. With our proposed governance best practices we know a data warehouse product owner can get buy-in and support and in return deliver piece of mind to stakeholders including the analytical community, IT management, finance, users and customer management of their organization.

ACKNOWLEDGMENT

The author is grateful to anonymous reviewers whose comments have improved the quality of the article substantially. The author also thank Joan Schnitzer, Senior Systems Analyst at Intel, for doing an excellent editing job.

REFERENCES

- Adomavicius, G., Tuzhilin, A., & Zheng, R. (2011). REQUEST: A query language for customizing recommendations. *Information Systems Research*, 22(1), 99–117. doi:10.1287/isre.1100.0274
- Agrawal, R., Johnson, C., Kiernan, J., & Leymann, F. (2006). Taming compliance with Sarbanes-Oxley internal controls using database technology. *Proceedings of the 22nd International Conference on Data Engineering (ICDE'06)*. IEEE Computer Society. doi:10.1109/ICDE.2006.155
- Aiken, P., Gillenson, M., Zhang, X., & Rafner, D. (2011). Data management and data administration: Assessing 25 years of practice. *Journal of Database Management*, 22(3).
- Akhter, S., & Rahman, N. (2015). Building a customer inquiry database system. *International Journal of Technology Diffusion*, 6(2), 59–76. doi:10.4018/IJTD.2015040104
- Allen, G., & Parsons, J. (2010). Is query reuse potentially harmful? Anchoring and adjustment in adapting existing database queries. *Information Systems Research*, 21(1), 56–77. doi:10.1287/isre.1080.0189
- Arnott, D. (2008, December 3-5). Success factors for data warehouse and business intelligence Systems. *Proceedings of the 19th Australasian Conference on Information Systems (ACIS 2008)*, Christchurch, Australia (pp. 55-65).
- Ballou, D. P., & Tayi, G. K. (1999). Enhancing data quality in data warehouse environments. *Communications of the ACM*, 42(1), 73–78. doi:10.1145/291469.291471
- Bellatreche, L., & Kerkad, A. (2015). Query interaction based approach for horizontal data partitioning. *International Journal of Data Warehousing and Mining*, 11(2), 44–61. doi:10.4018/ijdwm.2015040103
- Brown, W., & Nasuti, F. (2005). Sarbanes-Oxley and enterprise security: IT governance - What it takes to get the job done. *Information Systems Security*, 14(5), 15–28. doi:10.1201/1086.1065898X/45654.14.5.20051101/91010.4
- Elmagarmid, A. K., Ipeirotis, P. G., & Verykios, V. S. (2007). Duplicate record detection: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 19(1), 1–16. doi:10.1109/TKDE.2007.250581
- Feinberg, D., & Beyer, M.A. (2010). Magic quadrant for data warehouse database management systems. *Gartner Research*, Gartner, Inc, ID No. G00173535.
- Herschel, R. T. (2012). *Principles and applications of business intelligence research* (1st ed., pp. 286–300). USA: IGI Global.
- Hwang, M. I., & Xu, H. (2008). A structural model of data warehousing success. *Journal of Computer Information Systems*, 49(1), 48–56.
- Inmon, W. H. (2005). *Building the data warehouse* (4th ed.). John Wiley & Sons.
- Kimball, R. (2013). *The data warehouse toolkit: The definitive guide to dimensional modeling* (3rd ed.). John Wiley & Sons.
- Lu, Y., & Ramamurthy, K. (2011). Understanding the link between information technology capability and organizational agility: An empirical examination. *Management Information Systems Quarterly*, 35(4), 931–954.
- March, S. T., & Hevner, A. R. (2007). Integrated decision support systems: A data warehousing perspective. *Decision Support Systems*, 43(3), 1031–1043. doi:10.1016/j.dss.2005.05.029
- Mithas, S., Ramasubbu, N., & Sambamurthy, V. (2011). How information management capability influences firm performance. *Management Information Systems Quarterly*, 35(1), 237–256.
- Neumuller, C., & Grunbacher, P. (2006). Automating software traceability in very small companies: A case study and lessons learned. *Proceedings of the 21st IEEE International Conference on Automated Software Engineering (ASE'06)*. IEEE Computer Society. doi:10.1109/ASE.2006.25
- Rabuzin, K. (2014). Deductive data warehouses. *International Journal of Data Warehousing and Mining*, 10(1), 16–31. doi:10.4018/ijdwm.2014010102

- Rahman, N. (2010a). Incremental load in a data warehousing environment. *International Journal of Intelligent Information Technologies*, 6(3), 1–16. doi:10.4018/jiit.2010070101
- Rahman, N. (2010b). Saving DBMS resources while running batch cycles in data warehouses. *International Journal of Technology Diffusion*, 1(2), 42–55.
- Rahman, N. (2013a). Measuring performance for data warehouses - A balanced scorecard approach. *International Journal of Computer and Information Technology*, 4(1), 1–7.
- Rahman, N. (2013b, May 5 - 8). SQL optimization in a parallel processing database system. *Proceedings of the IEEE 26th Canadian Conference of Electrical and Computer Engineering (CCECE 2013)*, Regina, Saskatchewan, Canada. doi:10.1109/CCECE.2013.6567832
- Rahman, N. (2014). Temporal data update methodologies for data warehousing. *Journal of the Southern Association for Information Systems*, 2(1), 25–41. doi:10.3998/jsais.11880084.0002.103
- Rahman, N., Burkhardt, P. W., & Hibray, K. W. (2010). Object migration tool for data warehouses. *International Journal of Strategic Information Technology and Applications*, 1(4), 55–73. doi:10.4018/jsita.2010100104
- Rahman, N., Marz, J., & Akhter, S. (2012). An ETL metadata model for data warehousing. *Journal of Computing and Information Technology*, 20(2), 95–111. doi:10.2498/cit.1002046
- Rahman, N., Rutz, D., & Akhter, S. (2011). Agile development in data warehousing. *International Journal of Business Intelligence Research*, 2(3), 64–77. doi:10.4018/jbir.2011070105
- Rahman, N., Rutz, D., Akhter, S., & Aldhaban, F. (2014). Emerging technologies in business intelligence and advanced analytics. *ULAB Journal of Science and Engineering*, 5(1), 7–17.
- Ramamurthy, K., Sen, A., & Sinha, A. P. (2008). An empirical investigation of the key determinants of data warehouse adoption. *Decision Support Systems*, 44(4), 817–841. doi:10.1016/j.dss.2007.10.006
- Roberts, N., & Grover, V. (2012). Leveraging information technology infrastructure to facilitate a firm's customer agility and competitive activity: An empirical investigation. *Journal of Management Information Systems*, 28(4), 231–270. doi:10.2753/MIS0742-1222280409
- Sen, A., Sinha, A. P., & Ramamurthy, K. (2006). Data warehousing process maturity: An exploratory study of factors influencing user perceptions. *IEEE Transactions on Engineering Management*, 53(3), 440–455. doi:10.1109/TEM.2006.877460
- Simitsis, A., Vassiliadis, P., & Sellis, T. (2005). Optimizing ETL processes in data warehouses. *Proceedings of the 21st International Conference on Data Engineering, ICDE*, Tokyo, Japan. doi:10.1109/ICDE.2005.103
- Stonebraker, M. (2012). New Opportunities for new SQL. *Communications of the ACM*, 55(11), 10–11. doi:10.1145/2366316.2366319
- Sugumaran, V. (2012). *Insight into advancements in intelligent information technologies* (1st ed., pp. 161–177). USA: IGI Global.
- Watson, H. J., Fuller, C., & Ariyachandra, T. (2004). Data warehouse governance: Best practices at blue cross and blue shield of North Carolina. *Decision Support Systems*, 38(3), 435–450. doi:10.1016/j.dss.2003.06.001
- Weill, W., Subramani, M., & Broadbent, M. (2002). Building IT infrastructure for strategic agility. *MIT Sloan Management Review*, 44(1), 57–65.
- Wixom, B. H., & Watson, H. J. (2001). An empirical investigation of the factors affecting data warehousing success. *Management Information Systems Quarterly*, 25(1), 17. doi:10.2307/3250957
- Zolait, A. H. S. (2012). *Knowledge and technology adoption, diffusion, and transfer: International perspectives* (1st ed., pp. 118–132). USA: IGI Global. doi:10.4018/978-1-4666-1752-0

Nayem Rahman is a Senior Enterprise Application Developer in IT Business Intelligence (BI) at Intel Corporation. He has implemented several large projects using data warehousing technology for Intel's mission critical enterprise DSS platforms and solutions. He is currently working toward the PhD degree in the Department of Engineering and Technology Management at Portland State University, USA. He holds an MS in Systems Science (Modeling & Simulation) from Portland State University, Oregon, USA and an MBA in Management Information Systems (MIS), Project Management, and Marketing from Wright State University, Ohio, USA. His most recent publications appeared in Proceedings of the IEEE 26th Canadian Conference of Electrical and Computer Engineering (CCECE 2013) and the Journal of the Southern Association for Information Systems (JSAIS). His principal research interests include Big Data Analytics, Active Data Warehousing, Data Mining for Business Intelligence, Intelligent Data Understanding using Simulation, and Simulation-based Decision Support System (DSS).