Spring 2007

Near Real-Time Extract, Transform and Load

Wei-Chwen Soon Wilson
Regis University

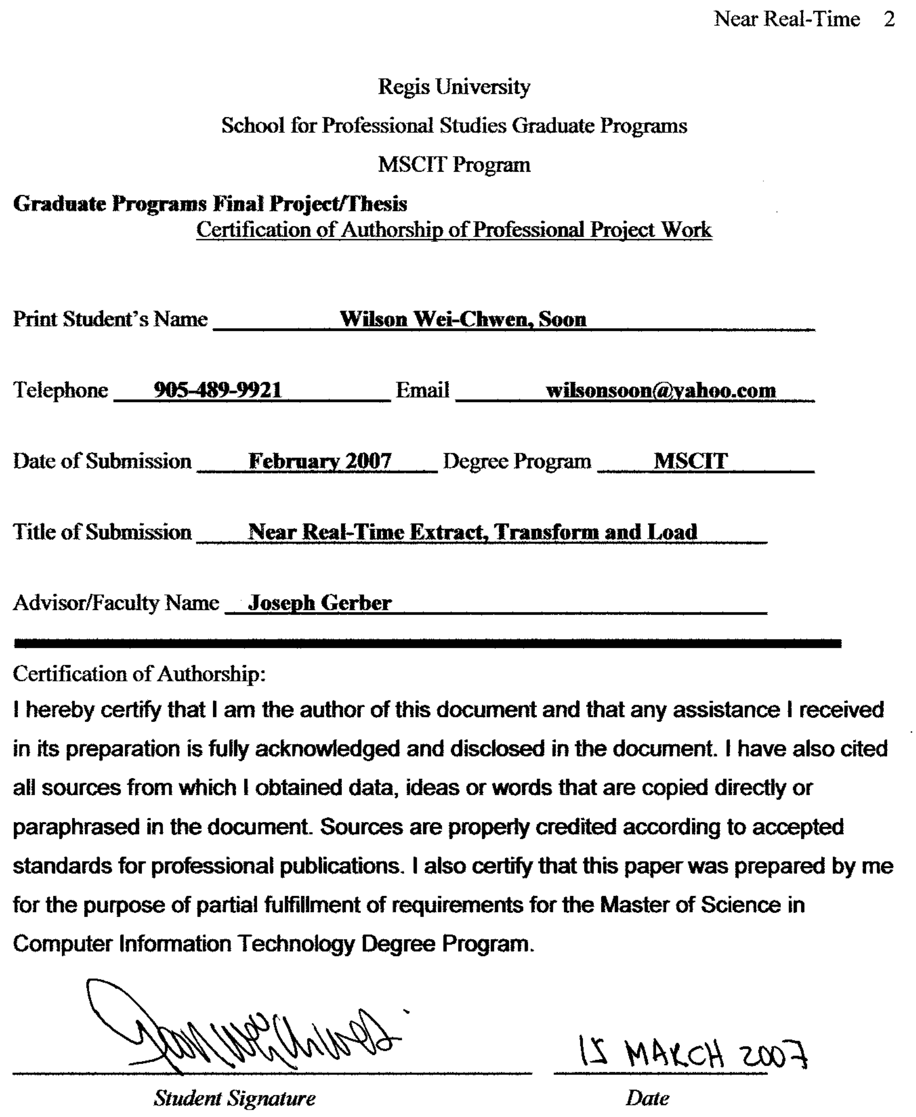
Near Real-Time Extract, Transform and Load

Wilson Wei-Chwen, Soon

Regis University

School for Professional Studies

Master of Science in Computer Information Technology



Regis University

School for Professional Studies Graduate Programs

MSCIT Program

Graduate Programs Final Project/Thesis

Certification of Authorship of Professional Project Work

Print Student's Name **Wilson Wei-Chwen, Soon**

Telephone **905-489-9921** Email **wilsonsoon@yahoo.com**

Date of Submission **February 2007** Degree Program **MSCIT**

Title of Submission **Near Real-Time Extract, Transform and Load**

Advisor/Faculty Name **Joseph Gerber**

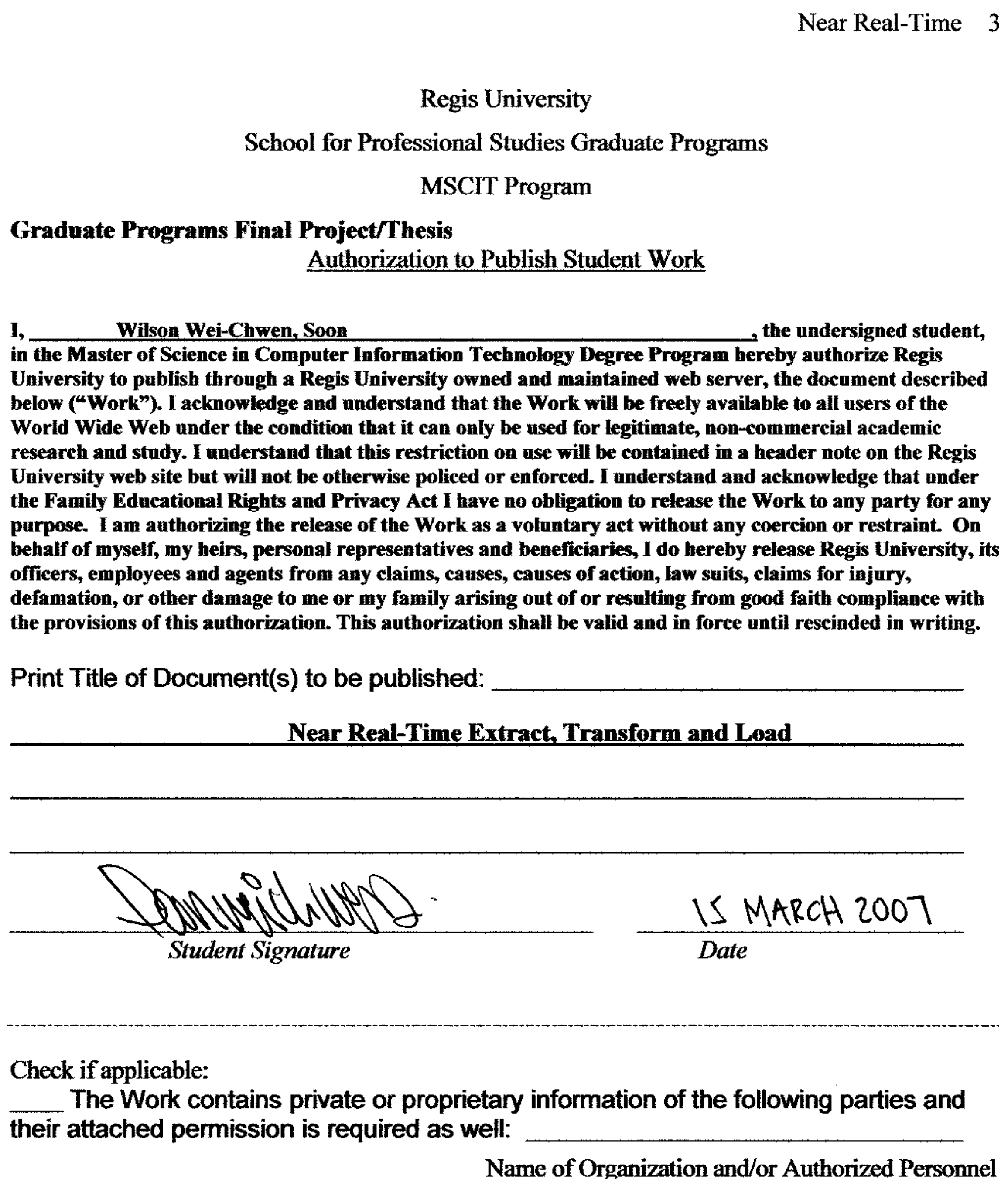
Certification of Authorship:

I hereby certify that I am the author of this document and that any assistance I received in its preparation is fully acknowledged and disclosed in the document. I have also cited all sources from which I obtained data, ideas or words that are copied directly or paraphrased in the document. Sources are properly credited according to accepted standards for professional publications. I also certify that this paper was prepared by me for the purpose of partial fulfillment of requirements for the Master of Science in Computer Information Technology Degree Program.

Student Signature

15 MARCH 2007

Date



Near Real-Time 3

Regis University

School for Professional Studies Graduate Programs

MSCIT Program

Graduate Programs Final Project/Thesis

Authorization to Publish Student Work

I, Wilson Wei-Chwen, Soon, the undersigned student, in the Master of Science in Computer Information Technology Degree Program hereby authorize Regis University to publish through a Regis University owned and maintained web server, the document described below ("Work"). I acknowledge and understand that the Work will be freely available to all users of the World Wide Web under the condition that it can only be used for legitimate, non-commercial academic research and study. I understand that this restriction on use will be contained in a header note on the Regis University web site but will not be otherwise policed or enforced. I understand and acknowledge that under the Family Educational Rights and Privacy Act I have no obligation to release the Work to any party for any purpose. I am authorizing the release of the Work as a voluntary act without any coercion or restraint. On behalf of myself, my heirs, personal representatives and beneficiaries, I do hereby release Regis University, its officers, employees and agents from any claims, causes, causes of action, law suits, claims for injury, defamation, or other damage to me or my family arising out of or resulting from good faith compliance with the provisions of this authorization. This authorization shall be valid and in force until rescinded in writing.

Print Title of Document(s) to be published: ______________________________

Near Real-Time Extract, Transform and Load

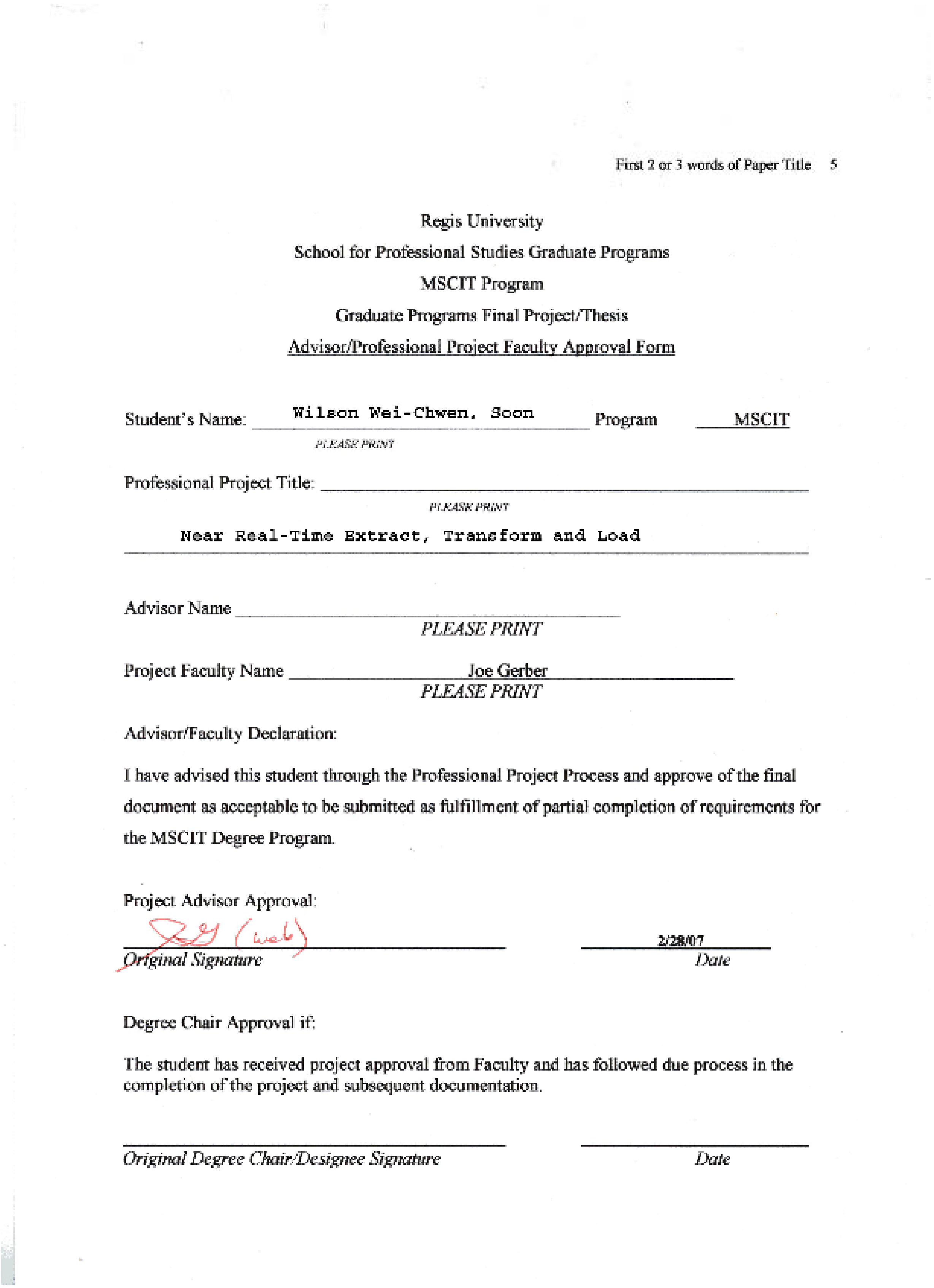
15 MARCH 2007

Student Signature   *Date*

Check if applicable:

____ The Work contains private or proprietary information of the following parties and their attached permission is required as well: ______________________________

Name of Organization and/or Authorized Personnel



First 2 or 3 words of Paper Title 5

Regis University

School for Professional Studies Graduate Programs

MSCIT Program

Graduate Programs Final Project/Thesis

Advisor/Professional Project Faculty Approval Form

Student's Name: Wilson Wei-Chwen, Soon Program MSCIT

PLEASE PRINT

Professional Project Title: ______________________

PLEASE PRINT

Near Real-Time Extract, Transform and Load

Advisor Name ______________________

PLEASE PRINT

Project Faculty Name Joe Gerber

PLEASE PRINT

Advisor/Faculty Declaration:

I have advised this student through the Professional Project Process and approve of the final document as acceptable to be submitted as fulfillment of partial completion of requirements for the MSCIT Degree Program.

Project Advisor Approval:

JG (web) 2/28/07

Original Signature *Date*

Degree Chair Approval if:

The student has received project approval from Faculty and has followed due process in the completion of the project and subsequent documentation.

______________________ ______________________

Original Degree Chair/Designee Signature *Date*

Regis University

School for Professional Studies Graduate Programs

MSCIT Program

Graduate Programs Final Project/Thesis
Advisor/Professional Project Faculty Approval Form

Student's Name: ___**WILSON WEI-CHWEN, SOON**_________ Program **MSCIT**______
PLEASE PRINT

Professional Project Title: ____
PLEASE PRINT

________**NEAR REAL-TIME EXTRACT, TRANSFORM AND LOAD**__________

Advisor Name ____________**BRAD BLAKE**______________________________
PLEASE PRINT

Project Faculty Name ________**JOSEPH GERBER**__________________________
PLEASE PRINT

Advisor/Faculty Declaration:

I have advised this student through the Professional Project Process and approve of the final document as acceptable to be submitted as fulfillment of partial completion of requirements for the MSCIT Degree Program.

Project Advisor Approval:

Brad Blake
___________________________ ______________________**02-27-2007**___
Original Signature ***Date***

Degree Chair Approval if:

The student has received project approval from Faculty and has followed due process in the completion of the project and subsequent documentation.

___________________________________ ____________________
Original Degree Chair/Designee Signature *Date*

Abstract

The integrated Public Health Information System (iPHIS) system requires a maximum one hour data latency for reporting and analysis. The existing system uses trigger-based replication technology to replicate data from the source database to the reporting database. The data is transformed into materialized views in an hourly full refresh for reporting. This solution is Central Processing Unit (CPU) intensive and is not scaleable. This paper presents the results of a pilot project which demonstrated that near real-time Extract, Transform and Load (ETL), using conventional ETL process with Change Data Capture (CDC), can replace this existing process to improve performance and scalability while maintaining near real-time data refresh. This paper also highlights the importance of carrying out a pilot project to precede a full-scale project to identify any technology gaps and to provide a comprehensive roadmap, especially when new technology is involved. In this pilot project, the author uncovered critical pre-requisites for near real-time ETL implementation including the need for CDC, dimensional model and suitable ETL software. The author recommended purchasers to buy software based on currently available features, to conduct proof-of-concept for critical requirement, and to avoid vaporware. The author also recommended using the Business Dimensional Lifecycle Methodology and Rapid-Prototype-Iterative Cycle for data warehouse related projects to substantially reduce project risk.

Acknowledgement

I would like to thank my wife Nancy and my daughter Kayla for their encouragement and understanding while I was working through the MSCIT program and this professional project paper. I dedicate this paper to Nancy and Kayla.

I would also like to thank my faculty advisor, Joe Gerber, my content advisor, Brad Blake, my peers Gary Howard, Michael Singleton and all those who have helped to review and to provide valuable comments. In particular, Joe's inspiring comments and advice, and the willingness to help whenever needed made possible the completion of this paper.

Finally, I would like to thank Regis University for the MSCIT program which provided me a wonderful and valuable learning experience.

Table of Contents

CHAPTER ONE: INTRODUCTION 10

Statement of the Problem to be Investigated and Goal to be Achieved 10
Relevance, Significance or Need for the Project 12
Barriers and/or Issues 13
Elements, Hypotheses, Theories, or Questions to be Discussed / Answered 14
Limitations/ Scope of the Project 14
Definition of Terms 15
Summary 19

CHAPTER TWO: REVIEW OF LITERATURE / RESEARCH 20

Overview of All Literature and Research on the Project 20
Literature and Research that is Specific / Relevant to the Project 20
Summary of What is Known and Unknown about the Project Topic 22
The Contribution this Project will Make to the Field 23

CHAPTER THREE: METHODOLOGY 24

Research Methods to be Used 24
Life-Cycle Models to be Followed 24
Specific Procedures 25
Formats for Presenting Results/ Deliverables 26
Review of the Deliverables 27
Resource Requirements 28
Summary 29

CHAPTER FOUR: PROJECT HISTORY 31

How the Project Began 31
How the Project was Managed 31
Significant Events/ Milestones in the Project 32
Changes to the Project Plan 34
Evaluation of Whether or not the Project Met Project Goals 35
Discussion of What Went Right and What Went Wrong in the Project 37
Discussion of Project Variables and Their Impact on the Project 38
Findings / Analysis Results 38
Summary of Results 41

CHAPTER FIVE: LESSONS LEARNED AND NEXT EVOLUTION OF THE PROJECT 43

What You Learned From the Project Experience 43
What You Would Have Done Differently in the Project 43
Discussion of Whether or not the Project Met Initial Project Expectations 43
What the Next Stage of Evolution for the Project Would be If It Continued 45
Conclusions / Recommendations 46
Summary 47

REFERENCES 49

EXHIBIT 52

LIST OF FIGURES

Figure 1. The Business Dimensional Lifecycle Diagram (Kimball, 1998)..................................25
Figure 2. Rapid-Prototype-Iterative Cycle (Wherescape, 2003) 26
Figure 3. Overview of the Near Real-Time ETL using Asynchronous Change Data Capture (Oracle Corporation, 2003) 52
Figure 4. Overview of the iPHIS Near Real-Time ETL Process Physical Schematic..................53
Figure 5. The iPHIS CDC Asynchronous Distributed Hotlog Logical Configuration 54
Figure 6. The iPHIS CDC Asynchronous Distributed Hotlog Physical Configuration................55
Figure 7. The iPHIS CDC Pilot Source and Target Tables Data Mapping..................................56
Figure 8. The iPHIS CDC Pilot Dimensional Data Model 57

Near Real-Time ETL

Chapter One: Introduction

Statement of the Problem to be Investigated and Goal to be Achieved

The Public Health System consists of 36 local Public Health Units (PHU) and a centralized Public Health Division (PHD). The mandates of the PHUs are to manage and prevent the spread of infectious disease and to promote and protect public health. The PHD is responsible for overseeing 36 PHUs as well as to liaise with the federal government and other provinces. The IT Cluster is one of the divisions within the Health Ministry which is mandated to provide Information and Information Technology expertise to the Health Ministry businesses such as the PHD. The Hosting Agency is an arm's length organization set up by the Health Ministry to provide IT infrastructure services such as secured network, email and data centre. The agency is fully funded by the Health Ministry.

The integrated Public Health Information System (iPHIS) is a disease case management application used by the PHUs to conduct contact follow-up and disease outbreak management. iPHIS is a web-based application with a centralized database that links all 36 PHUs and PHD over a secured network to allow data sharing. In addition, iPHIS offers near real-time reporting and analytical functionalities. iPHIS is expected to allow the PHUs and PHD to reduce the time it takes to recognize potential outbreaks of infectious disease and to trace the path of infectious disease propagation more quickly.

The iPHIS was implemented to all 36 PHUs and PHD in 2004 to replace the antiquated Reportable Disease Information System (RDIS) which were 36 independent stand-alone versions of the software that offered no data sharing. RDIS was deemed grossly inadequate by the public inquiry that was set up to investigate how the province had responded to Severe Acute

Respiratory Syndrome (SARS) outbreak and how well equipped the province is to tackle future public health problems.

Since iPHIS was put into production, system performance and availability problems began to surface which significantly impaired the PHUs and PHD mission to manage and report disease cases and to maintain disease outbreak readiness. The users had repeatedly threatened to abandon the system. In the summer of 2005, PHD retained a major consulting firm to conduct a 5 week end-to-end performance assessment. The consulting firm made recommendations to several areas of the iPHIS application to improve system wide performance. One area was the reporting system in which the consulting firm made several recommendations to improve reporting system performance. One of the recommendations was to replace the existing data replication system with an Extract, Transform and Load (ETL) process using a Commercial-Off-The-Shelf (COTS) ETL software to improve the performance of the reporting system. The ETL process involves extracting data from source databases, transforming it to fit business needs and ultimately loading it into the reporting database or data warehouse from which users can query to create reports or carry out analysis.

iPHIS system currently uses data replication feature of the database to replicate data from the source database to the reporting database for reporting. The replication process is automated by a hand coded computer program. The data replication process is hosted in the same server as the reporting database server. It is a Central Processing Unit (CPU) intensive process that seriously affects the performance of the rest of the system. In particular, data transformation, a step in the replication process, employs full data refresh and takes up 25% of the CPU which could be used to service users and to increase capacity to serve reporting users.

This pilot project was conducted to confirm that the reporting system performance can be improved by adopting an ETL process that utilizes an incremental data refresh to replace the current replication system which requires a full data refresh. Specifically, the ETL process is scaleable and flexible enough to allow future enhancement such as adding and modifying reporting data elements more easily.

In order to implement all of the consultant firm's recommendations, the Public Health Division (PHD) launched a project, the Integrated Project Team (IPT), with the intention of implanting all the consultant firm's recommendations. The project organized the recommendations logically into project work packages. The ETL work, which was one of the many project work packages, consists of three major phases. The first phase was to formulate a strategy to procure suitable ETL software which has been completed. The second phase, the subject of this report, was the pilot project to implement a prototype ETL process using the procured ETL software. The third phase was the full implementation of the ETL process using the ETL software to replace the replication process. In addition to testing the feasibility of the ETL process to replace the existing process, the pilot project was also expected to provide a roadmap for phase three full-scale ETL implementation.

This author was the technical lead and database administrator for the ETL pilot project. The author reviewed the consultant's work and made major technical decisions. The author was also responsible for signing off on the deployment package before it was sent to the hosting agency for production implementation.

Relevance, Significance or Need for the Project

The following are the risks of poor reporting system performance and response time:

- The epidemiologists may be prohibited from effectively support emergency disease outbreak activities
- The PHD may incur exponential and undetermined costs by having to add more hardware to solve performance issues as more users are added
- The system data replication time will continue to increase as database grows and eventually exceed the one hour data latency business requirement
- Users may stop inputting information into the transaction database if they cannot get reports on that information
- Users may try and circumvent the system in order to get reports that they actually want

Barriers and/or Issues

The integrated project team (IPT) set up to implement the consultant's recommendations to improve system wide performance was jointly managed by three stakeholders: the business (PHD), Information Technology (IT) Cluster (where the author is employed) and the hosting agency. The business controlled most of the budget, set project priority and scheduled the work packages according to business priorities. The IPT met every second week to review on-going projects, to prioritize and approve new work packages for development as appropriate. Due to the involvement of multiple stakeholders, the progress was sometimes spotty and unpredictable. Nevertheless, the business (PHD) eventually completed phase one of the work package which was the ETL strategy and procurement of the ETL software. The IT Cluster was assigned to lead the phase two of the ETL work package, the subject of this paper, which was to conduct a pilot project for the ETL software and process.

On the development side, most project staffs had higher priority operation tasks to do even though the staffs found the ETL work package interesting to do. In addition, there was a lack of in-house ETL experience. The IT cluster also faced with an inadequate development and testing environment.

Elements, Hypotheses, Theories, or Questions to be Discussed / Answered

There were two project elements. One of the elements was that the reporting system was required to be near real-time for disease surveillance and reporting. The epidemiologists needed near real-time disease case information in order to more quickly recognize potential infectious disease outbreaks, to more quickly trace the path of infectious disease propagation, and to more effectively create and monitor the effectiveness of containment strategy. The near real-time was described by PHD as having data latency no more than one hour. The other element was that ETL was a mature computer process, but near real-time ETL feature was relatively new.

There were three hypotheses. The first hypothesis was that the conventional near real-time data refresh could be modified and/or supplemented to enable near real-time ETL. The second hypothesis was that the full data refresh could be replaced by incremental data refresh and greatly improved performance and scalability. The last hypothesis was that the ETL process would allow reporting data element to be added and modified more easily.

The questions were how much performance improvement the ETL process would provide as compared to existing process and what metrics could be used to measure the improvement.

Limitations/ Scope of the Project

The pilot project was run by the IT cluster, and the reporting team lead was assigned to lead the project. The author was assigned as the technical lead and database administrator. The project had one full-time contract developer; other team members were drawn from other development projects as required.

The project team selected a small subset of nine source database tables for the pilot ETL process. The target database consisted of six tables. The project team created three aggregated materialized views (MV) which provided summary information of the target database data. Materialized views were database views that contained physical data to improve query performance. The project team also created two prototype reports based on data from the prototype target database.

Definition of Terms

CDC – Change Data Capture

Central Processing Unit - The brain of the computer where most calculations in a computer take place.

Change Data Capture - A feature of Oracle database which detects and captures source database changes such as insert, delete and update and saves those changes into a queue, typically in a staging database, to allow software such as ETL to process and retrieve the changes for further processing.

Commercial Off-The-Shelf - Ready-made software which is tested and available for sale to the general public.

Conformed Dimension – Dimensions that mean the same thing to all the facts tables linking to them.

COTS – Commercial Off-The-Shelf Software

CPU – Central Processing Unit

Data Latency - The delay between changes to source data and the reflection of those changes in the reporting data.

Data Warehouse - An enterprise data repository storing historical and integrated data for use in decision support systems.

Data Warehouse Bus Architecture – Consists of a suite of conformed dimensions and standardized definitions of facts. The business process data marts can plug into this bus to retrieve required data for analysis.

De-Normalized Entity-Relationship Data Model – A de-normalized database model that describes the attributes of entities and the relationships among them.

De-Normalized View – A database view created through de-normalization which is the process of optimizing the performance of a database by adding redundant data.

Decision Support System - A specific class of computer information system that supports business and organizational decision-making activities.

Dimension – A type of table in dimensional data model which represents business objects such as product, customer which typically used to filter data in fact table for analysis.

Dimensional Database Model – A model optimized for ease of use and query performance with end-user reporting and ad hoc analysis in mind.

Dimension Modeling – A methodology that logically models data into dimensions and facts for query performance and ease of use.

DSS – Decision Support System

Epidemiologist - A medical scientist who studies the transmission and control of epidemic diseases.

ETL – Extract, Transform and Load

Extract, Transform and Load - A process that loads data into database and most commonly into a data warehouse. The process involves extracting data from source database(s), transforming it to fit business needs and ultimately loading it into the reporting database or data warehouse from which users can query to create report or carry out analysis.

Facts – A type of table in dimensional data model consists of business measures and is usually numeric and typically additive.

Information Technology Infrastructure Library - The widely adopted standard for best practice in the provision of IT Service

Integrated Public Health Information System - A web-based application with a centralized database that offers data sharing and near real-time reporting, as well as comprehensive case management and outbreak management capabilities that are expected to reduce the time it takes to recognize potential outbreaks of infectious disease. iPHIS also enables health units to trace the path of infectious disease more quickly.

iPHIS – Integrated Public Health Information

ITIL – Information Technology Infrastructure Library

Materialized View - A database view is actually physically populated with data to improve query response time.

Near Real-Time ETL – Changes in source database are streamed continuously to the staging area and then transformed and loaded to the targets in incremental micro-batches in near real-time.

Normalized Database Model – A model optimized for transaction processing.

OLAP – Online Analytical Processing

OLTP – Online Transaction Processing

Online Analytical Processing - An analytical-oriented application typically provides answer to analytical queries that are dimensional in nature.

Online Transaction Processing - A transaction-oriented application typically used for data entry and retrieval transaction processing.

PHD - Public Health Division

PHU – Public Health Unit

PMI – Project Management Institute

Project Management Institute – An international organization that manages several levels of project management certification and published a number of project management standards including *A Guide to the Project Management Body of Knowledge* (PMBOK) Guide

Proof of Concept – A short exercise demonstrates the feasibility of a product or idea to verify that some concepts or theories are probably capable of exploitation in a useful manner.

Public Health Division - A division within the provincial health ministry to oversee local health units and to liaise with the federal health authority

Public Health Unit - A public health organization mandated to manage and prevent disease and to monitor and protect public health

SARS - Severe Acute Respiratory Syndrome

Severe Acute Respiratory Syndrome - A respiratory infection which caused the infection of the lung

Third Normal Form – As one of the several levels of data normalization. Third normal form is the process of organizing data to minimize redundancy. In third normal form, duplicate data is not permitted.

Vaporware – Software announced by the vendor in advance of its release but fails to emerge.

Summary

The iPHIS was experiencing system performance problem which significantly impaired the users' business functions. One of the identified system improvement project was the ETL project. The ETL project was created to replace the existing resource intensive data replication process which provided maximum one hour data latency for reporting. The ETL project consisted of three phases. Phase one was the creation of an ETL strategy and the purchase of ETL software, and was completed. Phase two was to conduct a pilot project to test the feasibility of the ETL software and process, the subject of this paper. Phase three is a future project which is the full-scale implementation of the ETL project to replace existing data replication process.

The major goals of this ETL pilot project were to:

- Test the feasibility of the ETL software and process to replace existing data replication process to maintain maximum one hour data latency and to greatly increase performance and scalability
- Provide a roadmap for phase three full-scale ETL implementation

Chapter Two: Review of Literature / Research

Overview of All Literature and Research on the Project

This project is related to data warehouse; therefore, the author started his research in the data warehouse area. A data warehouse is an enterprise data repository that stores historical and integrated data for use in decision support systems (DSS). The author researched on books and published articles that were written by the two pioneers in the data warehouse fields: Bill Inmon and Ralph Kimball. Bill Inmon is regarded as the father of data warehouse while Ralph Kimball is the creator of the immensely popular dimensional model and data warehouse bus architecture (Drewek, 2005). Data warehouse bus architecture consists of a suite of conformed dimensions and standardized definitions of facts. The business process data marts can plug into this bus to retrieve required data for analysis.

The author conducted a comparison on the advantages and disadvantages between the Inmon's and Kimball's approaches to data warehouse design especially in the choice of data warehouse data model. Data warehouse lifecycle methodology was also researched with the objective of selecting a suitable development lifecycle methodology for this ETL pilot project. The author also researched on the currently available technologies for real-time data warehouse and the Commercial-Off-The-Shelf (COTS) ETL software. In particular, the author reviewed ETL software packages that provide fully automated end-to-end support for near real-time ETL process.

Literature and Research that is Specific / Relevant to the Project

Bill Inmon (1992) first advocated the concept of corporate information factory in his book entitled *Building the Data Warehouse*. His book launched the data warehousing industry but also started enormous debate over the best model for the data warehouse. Inmon is a strong advocate of using third normal form relational data model for the data warehouse. Third normal form is one of the several levels of data normalization which organizes data to minimize redundancy. In third normal form, duplicate data is not permitted. Ralph Kimball, on the other hand, is regarded as the creator of the dimensional model for the data warehouse and advocates the data warehouse bus architecture. Ralph Kimball advocates the use of the dimensional model for data warehouse for its superior query performance and ease of use. Kimball's version of data warehouse is that an individual dimensional model is linked together with other dimensional models through conformed dimension to form the data warehouse. Conformed dimension is a dimension that means the same thing to every fact table within the data warehouse. To date, the industry is equally split with as many organizations implementing either Inmon's or Kimball's version of the data warehouse (Watson and Ariyachandra, 2005)

Although the dimensional model may not be suitable everywhere (Inmon, 2000), there is unanimous consensus that dimensional modeling is the best data model for the presentation layer because of its ease of use and superior query performance (Chuck, Amit, Vijaya, Nelson & Olaf, 2005; Inmon, 2000; Rittman, 2005). In addition, the dimensional model was also more flexible in terms of accommodating future enhancement such as adding and modifying reporting data fields (Chuck et al., 2005; Rittman, 2005).

For the ETL process, the author researched on Kimball and Caserta's book entitled *The Data Warehouse ETL Toolkit* which provides a comprehensive road map for planning, designing, building and running the ETL process. In this book, Kimball and Caserta characterize the ETL phase of the data warehouse development life-cycle as the most difficult, time-consuming, and labor-intensive phase of building a data warehouse which often consumes between 60-80% of the total resources of a typical data warehouse project. If the ETL process is executed successfully, companies will maximize their use of data storage; otherwise, they can end up wasting millions of dollars storing obsolete and rarely used data (Kimball and Caserta, 2004). Consequently, the ETL process is a considerably risky process in a data warehouse project.

Kimball and Caserta, in their book, further point out that the data warehouse is becoming more operational and real time. In a real-time operational data warehouse, the traditional batch-file oriented ETL processing is replaced by a streaming ETL processing. Change Data Capture (CDC) is by far the most common technology used to implement streaming ETL processing (Schmitz, n.d.; White 2005). Database CDC achieved streaming ETL by using a combination of database log mining, streaming and advanced queuing technology.

According to the Garner's research, IBM Ascential Software and Informatica Powercentre are the two leading ETL software packages in the market (Friedman, Beyer & Bitterer, 2006). The author conducted further research on the vendor web sites and also conversed with existing users of these two software packages. Based on this further research, the author confirmed that both of these leading ETL software packages have existing fully automated end-to-end support for near real-time ETL.

Summary of What is Known and Unknown about the Project Topic

From the research done in chapter two, the following is known

- The data warehouse industry is equally split with equal number of Inmon's and Kimball's version of data warehouse architecture implementation.
- Kimball's Dimensional Model is unanimously considered by the data warehouse industry as the best data model for the presentation layer.
- ETL process is the most resource intensive and risky process in the data warehouse project consuming between 60-80% of the total resources.

- In real-time operational data warehouse, traditional batch-file oriented ETL processing is replaced by a streaming ETL process which can be enabled by the Change Data Capture technology.
- IBM and Informatica are two leading ETL vendors which provide existing end-to-end support for near real-time ETL functionality.

The unknown about the project topic was whether the near real-time ETL process could replace the existing replication process to improve system performance and scalability which is the subject of this paper.

The Contribution this Project will Make to the Field

This project has the following contributions to the field:

- General contribution to the field of near real-time data warehouse
- Specific contribution to ETL process and especially near real-time ETL process using Change Data Capture
- Contribution to data warehouse data modeling
- Contribution to ETL software evaluation and selection

Chapter Three: Methodology

Research Methods to be Used

The author researched on books and published articles related to data warehouse architecture and lifecycle, dimensional modeling, ETL, near real-time ETL process and software. The author specifically studied the books and articles written by the two pioneers in the data warehouse fields: Bill Inmon and Ralph Kimball. The author also conducted the Internet search for related articles and research on the ETL vendor's websites.

Life-Cycle Models to be Followed

In the search for a development life-cycle model, the author read books and articles on methodologies such as the Waterfall, Rational Unified Process and Xtreme Programming. The author concluded that Kimball's Business Dimensional Lifecycle Methodology was most suitable for the ETL pilot project because of its comprehensive coverage of technical areas specific to data warehouse development and also on the build-in iterative approach within the methodology where successive data warehouse enhancement projects are built on the feedback from the previous enhancement projects. The Business Dimensional Lifecycle Methodology was presented in Ralph Kimball's 1998 book, *The Data Warehouse Lifecycle Toolkit—Expert Methods for Designing, Developing and Deploying Data Warehouses*. Kimball helped to formulate this methodology while working at Metaphor. The methodology subsequently evolved from his experience working with hundreds of data warehouse projects.

The Business Dimensional Lifecycle Methodology begins with project planning followed by business requirement and project management processes. The business requirements feed three streams of concurrent project activities: Technical Architecture, Data model/Database/ETL and End-user application. The three streams then are regrouped to merge into the project deployment stream. The following figure illustrates the Business Dimensional Lifecycle. The data staging design and development phase is the ETL process, the subject of this paper.

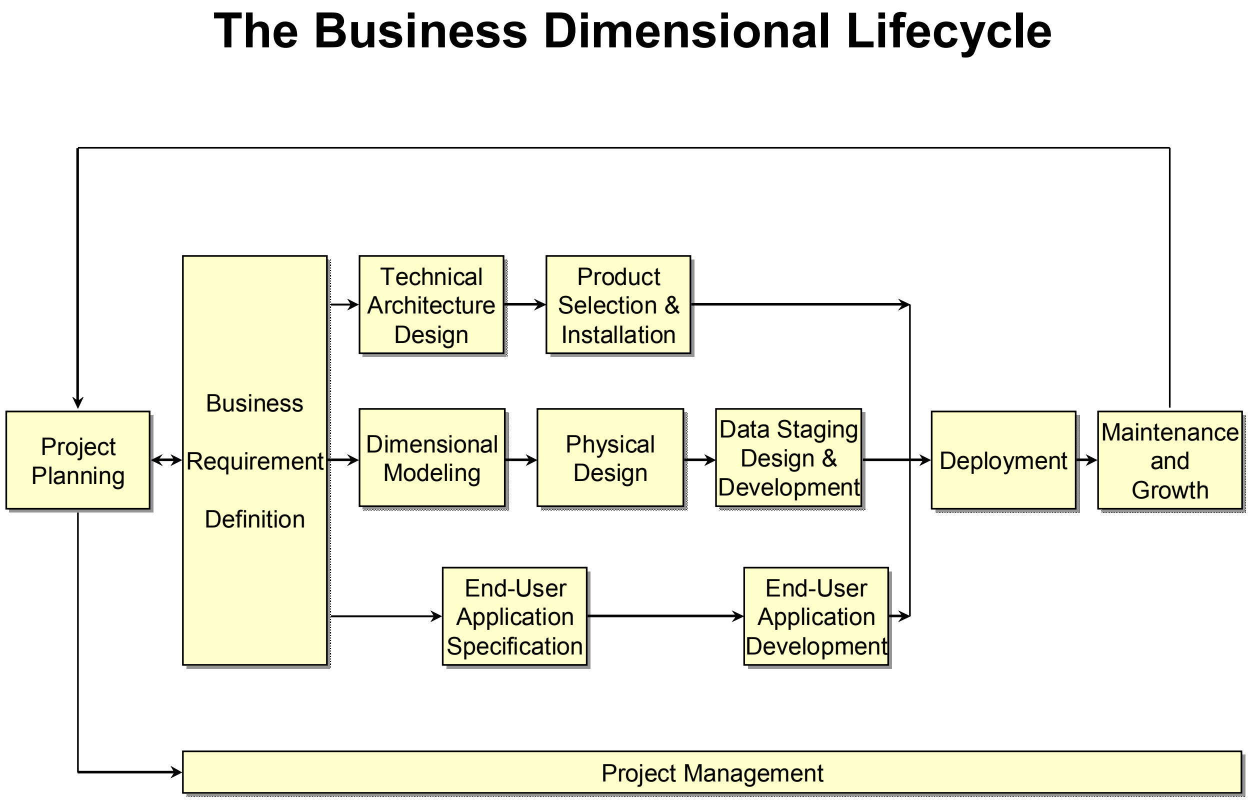

The Business Dimensional Lifecycle
Technical Architecture Design
Product Selection & Installation
Business Requirement Definition
Project Planning
Dimensional Modeling
Physical Design
Data Staging Design & Development
Deployment
Maintenance and Growth
End-User Application Specification
End-User Application Development
Project Management

Figure 1. The Business Dimensional Lifecycle Diagram (Kimball, 1998)

Specific Procedures

According to Inmon (1992), decision Support System (DSS) analysts operate in the exploration state of mind. DSS analysts can better articulate their requirements only after they have the opportunity to perform various kinds of analysis on the data. Consequently, iteration is a common practice in data warehouse project. Iteration is a complete development loop typically resulting in the release of one business subject area for the data warehouse. This subject area is a subset of the total data warehouse under development, which grows incrementally from the iterations to become the final system. One of the major benefits of iteration is to allow the project to mitigate risks by allowing the DSS analysts to incrementally work with the data and hence continuously refine the requirement as the project progressively builds the complete data warehouse. The following figure illustrates the rapid-prototype-iterative cycle.

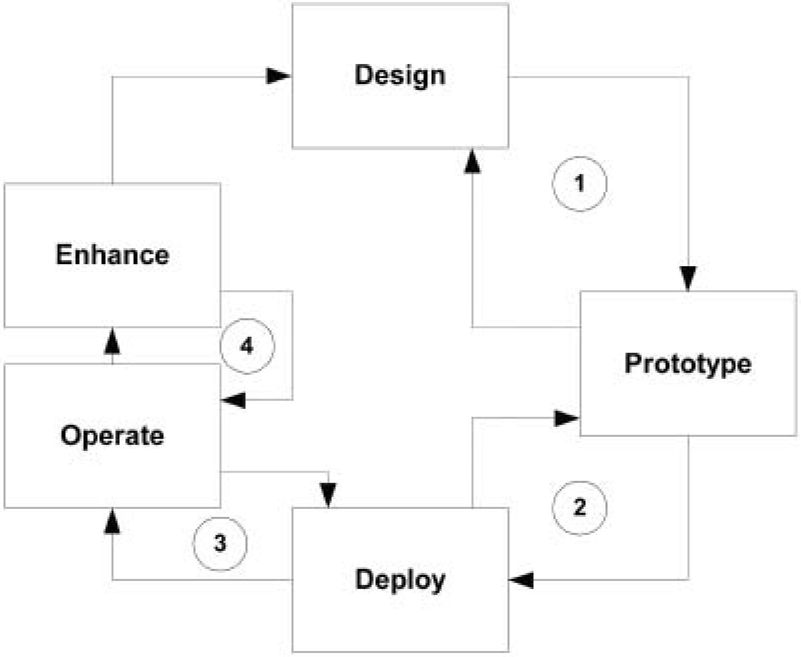


Figure 2. Rapid-Prototype-Iterative Cycle (Wherescape, 2003)

The author used the data warehouse business dimensional lifecycle model because it supported the rapid-prototype-and-iterative cycle. This pilot project can be considered as one of the iterative cycles.

Formats for Presenting Results/ Deliverables

The pilot project was expected to produce detailed documentation for each of the deliverables in the next section. This documentation was to be used for input to business requirement and as reference documentation for phase three full ETL implementation project. At the completion of the development, the development team provided a formal presentation of the pilot prototype to management. Following the presentation, the deployment package was formally released to the hosting agency and hosted at the production training environment for both business and IT to conduct detailed evaluation. The business and IT conducted detailed evaluation and provided reports from their own perspectives.

Review of the Deliverables

The pilot project team identified the following deliverables with respect to each major phase of the Business Dimensional Lifecycle Methodology:

Project Planning Phase

- Project charter and scope of work
- Integrated project plan and schedule
- Change and release management procedure
- Prototype analysis, findings and recommendations

Business Requirement Definition Phase

- Business processes
- Pilot project business requirement

Technical Architecture Design Phase

- Technical architecture plan

Product Selection and Installation Phase

- Server selection and specification
- Physical schematic and installation plan

Dimensional Modeling Phase

- Bus matrix
- Dimension data model

Physical Design Phase

- Database schema design

Data Staging Design & Development Phase

- Data mapping
- Transformation requirement

End-User Application Phase

- Specification
- Development plan

Deployment Phase

- Testing plan (Unit, Integration and User Acceptance Test)
- Training plan
- Production implementation package
    - Source database CDC setup document
    - Staging database CDC setup document
    - Data warehouse target schema setup document
    - ETL setup document
    - Data loading and materialized views creation script
    - CDC enabling script
    - ETL scheduler setup script

Resource Requirements

The pilot project required staff members to work on the deliverables in each phase. A business analyst was required to compile business requirement. A technical architect was required to design, specify and implement the development, testing and production environments including server, storage, security and network requirements. A Decision support application developer was required to design and develop analytical application and reports for the end users. For the ETL phase, multiple expertises were required. A data modeler was required to design and develop the target database data model. An ETL developer and a database administrator were required to design and develop data mapping and transformation between the source and target database. Since this was a pilot project, the scale was small; consequently, the above mentioned resources were mostly required on a part time, as required basis.

This author was the technical lead and database administrator for the ETL pilot project. The author reviewed the consultant's work and made major technical decisions. The author was also responsible for signing off on the deployment package before it was sent to the hosting agency for production implementation. The hosting agency was required to deploy the pilot project package to the production training environment for the pilot testing and evaluation. The project team identified two additional servers required for the pilot deployment.

Outcomes

The desired outcomes of this ETL pilot project were:

- The ETL software and process were confirmed to be feasible to replace the existing replication process and improved reporting system performance and scalability while maintaining maximum one hour data latency
- A roadmap was created for the full-scale ETL process implementation which identified critical technical gaps, if any, and possibly identified the workarounds

Summary

The author conducted research by reading data warehouse related books, searched the Internet for related articles and browsed the ETL vendor websites for ETL software inforamtion.

The author also completed research on the standard development methodologies and concluded that Kimball's Business Dimensional Lifecycle Methodology was most suitable for this pilot project because of its comprehensive coverage of technical areas specific to data warehouse development. In addition, the author introduced the Rapid-Prototype-Iterative cycle concept to highlight the iterative nature of the data warehouse project. The author also presented the list of deliverables, resource requirements, and desire outcomes for the pilot project.

Chapter Four: Project History

How the Project Began

The Severe Acute Respiratory Syndrome (SARS) outbreak in 2003 exposed the inadequacy of the existing outbreak management system, the Reportable Disease Information System (RDIS), to track disease cases and to provide epidemiologists with critical disease case information to monitor disease propagation and to devise effective containment strategy. Subsequently, the Integrated Public Health Information (iPHIS) was identified to replace RDIS. iPHIS is a mission critical application which requires near real time reporting system for disease outbreak surveillance and monitoring. The application was in production since December 2004 with near real time reporting (maximum 1 hour data latency). In early 2005, production performance problems emerged; users began to experience poor system response time and system availability. As the problem persisted, iPHIS users were getting impatient waiting for a solution and threatened to abandon the system. In summer 2005, the Public Health Division (PHD) commissioned a major consulting firm to conduct a 5-week performance assessment. The consultant made a series of recommendations to improve performance. One of them was to use the ETL software and process to replace existing custom coded data replication process.

How the Project was Managed

The PHD set up the integrated project team (IPT) to implement the consultant firm's recommendations. The IPT is made up of members from the business (PHD), IT Cluster (where the author is employed) and the hosting agency. The IPT organized the consultant firm's recommendations logically into work packages of the project. One of the work packages was the implementation of ETL process with a Commercial-Off-The-Shelf (COTS) ETL software. The work package consisted of three phases: Phase one was to create an ETL strategy and to purchase the ETL software. Phase two was to conduct a pilot project (the subject of this paper), to test the feasibility of the ETL process and the ETL software. Phase three was to implement the ETL process system wide. Depending on the content of the work packages, leadership role was assigned to a member group (PHD, IT or the hosting agency) with participation from other groups as required. The phase one of the ETL project was assigned to PHD and was completed. The phase two of the ETL work package, the subject of this paper, was assigned to IT cluster.

The reporting team lead was assigned to lead the pilot project. The author was assigned as the technical lead and database administrator. A contract developer was hired to work full time on the project. Other staffs including database programmer, report developer, tester, and business analyst were engaged on an as required basis. The pilot project was managed as a change request under the iPHIS change management procedure and executed under iPHIS release management procedure. Both the iPHIS change and release management were adopted from the IT infrastructure Library (ITIL) framework.

The author reviewed the consultant's work, made major technical decisions, and sign off on the deployment package before it was sent to the hosting agency for production implementation.

Significant Events/ Milestones in the Project

The contract developer started March, 2006. The consultant recommended the Change Data Capture (CDC) feature of the database to enable near real-time ETL. The author concurred with the recommendation based on his research on enabling technology for near real-time ETL (More information about the subject research can be found in chapter two).

Shortly after the development started, the contract developer discovered that the reporting data model was unnecessarily complex. The existing system was created with materialized views to serve the users' reporting needs. These materialized views were created through the de-normalization of the source database tables. Materialized view is a database view that is physically populated with data to improve query response time. These materialized views were joined together for drilling across data query to form a de-normalized entity-relational data model. Over time, the data model became extremely inflexible which made system enhancement such as adding and modifying data fields difficult and required lengthy regression test. The contract developer had to simplify the views in order to proceed with the ETL work. After substantial simplification, the consultant concluded that the logic was still too complex for the ETL process because of the patch work done on the views over several reporting changes.

Since the current model had almost come to a dead end because of evolving business requirement and associated patch work, it was a good opportunity to remodel the reporting database to a dimension model for the ETL project. In addition, the author has confirmed from his research that dimensional modeling was unanimously recognized as more superior data model for the presentation (Chapter Two). As a result, the author directed the developer to model the pilot subject area into a prototype dimensional model for reporting.

Another significant event was that the new version of the purchased ETL, which promised end-to-end automation of the near real-time ETL, was repeatedly delayed. Consequently, the developer had to hand code the configuration for the near real-time functionality to complete the pilot project. When the new version was eventually released, the near real-time functionality, which was promised by the sales engineer during pre-sale demonstration, was excluded.

Besides missing the important functionality, the purchased ETL software was also found to be not intuitive, not user-friendly and with lots of bugs. As manual coding and configuration was unmanageable and not cost effective for a full-scale ETL process, new ETL software would be required for the full-scale implementation. The author conducted research into other ETL software and identified two leading ETL vendors that provided near real-time functionality in their current release (Chapter Two).

The pilot project was completed in May 2006. The team made a presentation to the management, and it was well received. The deployment package was approved to be deployed to production training environment for evaluation which was done in November 2006 after long delays due to unavailability of hosting services. The pilot project was officially completed in December 2006.

Changes to the Project Plan

The original scope of the pilot project did not include the remodeling of the reporting database. However, during the pilot project, the remodeling was identified as necessary as the current model did not work efficiently with the ETL process. In addition, the proposed dimensional model will be more flexible in terms of accommodating future enhancements such as adding and modifying reporting data fields. Consequently, target database remodeling was added to the pilot project scope and was identified as an important piece of the roadmap for the full-scale implementation project.

The pilot project team had to abandon the purchased ETL software and resorted to hand coding the ETL process in order to complete the pilot project. This highlighted the danger of purchasing vaporware. Vaporware is software announced by the vendor in advance of its release but fails to emerge. The author recognized that ETL software was essential for full-scale implementation as hand coding was unmanageable and un-maintainable both for the full-scale ETL implementation and future maintenance. Therefore, new ETL software that supported the full automation of end-to-end near real-time ETL functionality would be required for the full-scale ETL project.

The pilot project scope included a performance test to quantify the performance improvement of the ETL process to further strengthen the business case for phase three full-scale ETL project. However, the testing environment was unavailable and the formal performance testing was eventually cancelled. This highlighted the importance of detailed planning. The pilot project was led by the IT cluster and the hosting agency was responsible for providing the testing environment. New servers were not purchased in a timely manner and caused the project to miss a deadline. In the full-scale implementation, the three parties in the team should be involved to create one integrated plan so that important and long lead time activities such as the performance testing could be properly planned and successfully executed.

Evaluation of Whether or not the Project Met Project Goals

Project planning and management were done very informally. There was little documentation created other that the project charter. Business requirement was based on the knowledge of the IT staffs on the existing system. Since this was a system improvement project involving the back end ETL process, there was very little new business requirement other than to improve reporting system performance. A technical architecture document was produced which detailed the requirement for development, testing and production environment such as the number of servers and the configuration. A dimensional model was created for the target reporting database which was consisted of one fact table with five dimensions. Physical database design was completed which included automated scripts to create database objects such as tables, indexes and views, to execute the initial load of data, and to create user security roles. Data staging design and development were designed with detailed extract, transform and load logic, data refresh frequency and error logging. End-user application including data model and two reports were created and tested. The complete system passed unit and integration tests.

Since the pilot ETL system was small and simple, no user training was scheduled. The system was eventually successfully deployed to production training environment for evaluation.

The pilot project provided a clearer roadmap for full-scale implementation. Major technical and project management gaps such as the need for a dimensional data model, a new ETL software, a performance test and an integrated project planning were identified.

Through unit and integration testing, the pilot project team confirmed that conventional ETL process could be modified to enable near real-time ETL by supplementing the process with Change Data Capture (CDC) feature of the database. During integration testing, multiple data were entered into the source database and the author was able to monitor the changes being streamed across to the staging database and queued up in the change tables. The manually coded batched program was executed as scheduled to accurately pick up the changes, transformed the changes as coded, and loaded to the reporting database. The pilot project therefore met one of its major goals that the existing near real-time replication process could be replaced by the conventional ETL process with CDC.

The pilot project also showed that the ETL process was able to support incremental data refresh and could be moved to its own dedicated server separated from the reporting server. Therefore, the reporting system performance should theoretically be substantially improved and definitely scaleable. However, the project team was not able to quantifiable the performance improvement by a performance test because of the unavailability of the testing environment. On the other hand, any load test was still subject to extrapolation as the performance testing was done on a prototype of the full-scale system. Consequently, at the moment, the team was satisfied with the observation that the ETL process would improve performance and could maintain the one hour maximum data latency requirement but would definitely be more scaleable because of the ability to incrementally refresh the data. Nevertheless, the performance of the ETL process was still a potential risk item for the full-scale ETL implementation; therefore, the author recommended the full-scale project to proceed with the Business Dimensional Lifecycle Methodology with Rapid-Prototype-Iterative Cycle, as stated in chapter three, in order to provide important performance check point to review performance and data latency in an incremental iteration so that project risk could be mitigated and minimized.

Overall, the pilot project was successful in confirming that the existing replication process could be replaced by the ETL process and provided a complete picture for the full-scale implementation. However, there was a lot of room for improvement in project planning and performance testing. Several project deliverables including integrated project plan and schedule should have been more complete so that project activities such as setting up the development, testing ,and production environment could be tracked more closely and in turn avoided major delays. In addition, the project team should have gotten more detailed business requirements so that the business users could provide comprehensive testing and analysis from the business perspective which would benefit the full-scale implementation.

Discussion of What Went Right and What Went Wrong in the Project

Right: Identified conventional ETL could be modified to implement near-real time ETL by using CDC feature of the database

Right: Identified Dimensional model as a better presentation model

Right: Confirmed ETL process could enable incremental data refresh and hence improved performance and scalability

Wrong: Allowed pilot project priority to be moved around by other higher priority projects.

Wrong: Hired contract developer with no direct experience with the purchased ETL tool

Wrong: Did not complete the planned performance test

Wrong: Did not conduct performance benchmarking to quantify the performance improvement

Wrong: Did not create detailed test plan and relied on subjective observation

Wrong: Did not talk with the end users about what they wanted and did not want

Discussion of Project Variables and Their Impact on the Project

One of the variables of the pilot project was its priority which was dependent on the factors such as whether the existing data replication process was causing performance problem at the moment and the staff's workload. Furthermore, improvement to the existing data replication process has improved iPHIS performance, which decreased the urgency of the ETL project.

There was time when the project staffs' priority was yielded to other higher priority development and production support work which caused spotty and erratic project activities and created considerable delays. There was also limited hardware to set up the development and test environment because of other development activities which caused less than ideal although somewhat adequate development and testing environment

Findings / Analysis Results

The pilot project was successfully executed with conventional ETL supplemented with Change Data Capture feature of the database to make near real-time ETL. CDC captured the data changes and delivered them to the staging area in a continuous stream where they were transformed and loaded to target reporting database in micro-batches and then made available to users for reporting. During pilot testing, the author found that the CDC feature of the database was very stable and reliable and was suitable for mission critical application such as iPHIS. In addition, CDC was easy to configure and the learning curve was small for the database administrator. The ETL scheduler was able to be scheduled to wake up every five minutes and to look for data changes and to process changes to the target schema.

The pilot project did not execute the planned performance test and hence was unable to quantify the actual near real-time ETL performance under typical users load. However, the changes appeared almost instantaneously on the reporting database when changes were made to the source database. The performance testing was cancelled because no suitable testing environment was available. The pilot project team should have engaged the technical architecture group and testing group earlier to prepare for these long lead time activities. The performance of this batch processing of the data changes was dependent on several factors including the complexity of transformation algorithm, the volume of data and data changes. Although the pilot project team believed that the ETL process would improve performance, and the performance on the prototype for one simplified transaction was good, the overall performance of the system could not be quantified because a suitable testing environment was unavailable to quantify the performance improvement.

The original data model was created by using de-normalized views organized into entity relation model for reporting. This model eventually came to a dead end as the business requirement continuously evolved. This model might have saved time in the beginning but would inevitability cost more in the long run. There was evidence in the iPHIS model where the addition and modification of new data fields became difficult. The author through his research (Chapter Two) and the recommendation of the developer had confirmed that the dimension model was a better option for the reporting presentation database. Consequently, the author directed the developer to design a prototype dimensional data model for the pilot project.

The ETL process was inherently using an incremental data refresh. However, the ETL is more complicated than the current full data refresh because of the complicated error trapping logic required to keep track of what was loaded and to manage loading exception. The performance was expected to improve because data could propagate incrementally instead of doing full propagation every time. Consequently, the ETL process was feasible to replace the existing full data refresh replication process. In addition, using ETL software provided substantial other enterprise level functionalities which were not available with custom coded program. These include data profiling, data cleansing, metadata repository, version control, and error auditing.

The project team had also determined that proper project management methodology should be used to plan, schedule, and track the project activities to ensure the project meet its objectives and completed on-time. In this case, one of the major project objectives regarding quantifying performance improvement was not done due to inadequate performance testing planning.

The evaluation of the ETL software would be more beneficial with an experienced user. This user could use the ETL software to run through this specific ETL process requirement to identify any major deficiency. During the pre-sale presentation, the ETL vendor promised the required near real-time ETL feature would be included in the upcoming release. The new release was delayed beyond the pilot project's schedule and when it was eventually became available, the required functionally was not included. Consequently, it was not advisable to buy software based on promises or vaporware.

Even though the ETL process was a backend process that users did not usually see, it was a good idea to get more involvement and support from the users as ETL process was a costly process (Chapter Two) that needed business support in terms of budget and feedback.

Nevertheless, the pilot project team did engage the business users to run the pilot project reports and give their comments. Since the ETL process was essentially a backend process, the users were unable to draw significant comments regarding the pilot prototype. However, the users mentioned that the pilot project in its current state was not adequate from a business perspective and would not satisfy their reporting requirements. This was understandable as the pilot prototype had only a very small subset of data element. The users encouraged IT to move forward with full implementation and reiterated their major business requirements:

- Full scope of fields to be available to report on
- Fields naming convention for ease of use by users
- Enhanced system performance

Finally, there was substantial difference in production environment as compared to the development environment. Consequently, extra configuration changes and testing were made to the production deployment package before the pilot project was able to successfully deployed.

Summary of Results

The ETL project was created to solve iPHIS reporting system performance problem. The objective of the ETL project was to replace the existing CPU intensive and not scaleable data replication process. The pilot project confirmed the feasibility of the ETL process and provided a more complete roadmap for full-scale ETL implementation. However, several deliverables were missed including detailed project plan, performance testing and business requirement. In regard to the roadmap for full-scale implementation, the pilot project identified the following gaps and workarounds:

- Identified CDC to enable near real-time ETL
- Decided to use dimensional model to replace existing reporting data model

- Discovered major deficiencies with the procured ETL software and the need for new ETL software that support near real-time ETL
- Identified requirement for detailed business requirement, project planning, and scheduling
- Uncovered differences between production and development-testing environment which were needed to be resolved to avoid rework for production deployment

Chapter Five: Lessons Learned and Next Evolution of the Project

What You Learned From the Project Experience

The major lesson learned from this pilot project was that a pilot project was crucial for the full-scale implementation of a new technology especially if there was limited in-house experience with the new technology such as the case with the author's project. The pilot project provided an opportunity to identify unknown pieces that were critical to the success of the full-scale implementation such as the need for the CDC, dimensional model, and appropriate ETL software. In addition, even if the pilot project failed, one could also learn from the failure and, depending on the nature, one could take necessary steps in the full-scale implementation to avoid repeating those mistakes, or conclude that full-scale implementation was not feasible.

What You Would Have Done Differently in the Project

There were three things that should have been done differently. One of them was that the project team should have requested the vendor to do a proof of concept of its ETL software for near real-time ETL processing. The other was that the project team should not have relied on the vendor's promise that the required feature would be included in future release. Lastly, the pilot project team could have done a better project planning so that performance test and other critical testing requirement, such as ease of adding and modifying data elements, could be identified earlier. This will allow the project team to create test plan, to recruit testers, and to setup testing environment to benchmark and quantify the performance improvement of the ETL process. With a detailed project plan and regular tracking, the pilot project could be completed earlier.

Discussion of Whether or not the Project Met Initial Project Expectations

The major goals of this ETL pilot project were to:

- Test the feasibility of the ETL software and process

- Provide a road map for full-scale implementation

The pilot project team determined that the purchased ETL software was unsuitable and hence the team needed to re-enter the market to procure ETL software that can support end-to-end near real-time ETL process. From the research in chapter two, the author identified two leading ETL vendors who provided this capability. The author confirmed that the ETL process was a suitable process to replace existing replication.

The pilot project team met most of the initial project expectations but missed some of the expectations. Although the results from the pilot project confirmed that there would be system performance improvement based on the fact that the ETL process could enable incremental refresh and could be moved off to its dedicated server separated from the reporting server, the pilot project team did not conduct a performance test to quantify the performance improvement. In addition, it was difficult to test whether the new ETL process could meet the maximum one hour data latency requirement with a small pilot project such as this one even when full user loads could be simulated in a performance test. Consequently, the full-scale project should proceed iteratively (Chapter Three) so that performance and data latency could be incrementally checked and remedied, if required. This approach will substantially reduce overall project risk.

The pilot project team mapped out the end-to-end ETL process including the source, staging, and target processes. In addition, the pilot project team also identified technology gaps including the need for CDC, dimension model, and ETL software with near real-time support. The pilot project team identified major deficiencies in testing infrastructure and capability in term of allowing the pilot testing to positively confirm and quantify performance improvement. The pilot project team identified significant differences between development environment and production environment, which were needed to be resolved in order to have a smooth production deployment. Although adopting dimensional model should theoretically make the system more flexible and allow users to add, delete, and update data elements more easily and quickly, the pilot project team did not carry out a test to confirm that.

Based on the analysis and results in chapter four, the pilot project team provided sufficient information to enable management to make the decision to carry out a full-scale implementation and hence the pilot project team has accomplished the second goal that was to provide a road map for the full-scale implementation.

In summary, the pilot project team achieved the two major project goals with several lessons learned.

What the Next Stage of Evolution for the Project Would be If It Continued

The next step is to prepare a business case to request approval and funding for the full-scale ETL project. At the meantime, the Integrated Project Team (IPT) should start to create a comprehensive integrated project plan for the full-scale ETL project using proper project management methodology such as those provided by the Project Management Institute (PMI), and to employ a proper methodology such as the Business Dimensional Lifecycle Methodology (Chapter Three). The project plan should include a strategy to maintain two infrastructures: the existing production system and the new ETL process, and a plan to gradually phase in the ETL process and to decommission the existing system. The IPT project team should start to solicit user performance criteria and metrics, and to plan for system testing. For deliverable requiring long lead time, such as creating a performance testing environment and converging the differences between the development and production environments, the IPT should start to plan in detail so that the project duration and budget of the full-scale project can be more accurately estimated for input to the business case.

The IPT team should start the tendering process for new ETL software and request specific vendors to conduct proof-of-concept testing to ensure the software can meet the near real-time ETL requirement.

Another parallel task the IPT can do is to start gathering detailed business requirement, to identify and prioritize business processes to be implemented, and to engage users to redesign the complete model to a dimensional model. Finally, a strategy needs to be formulated to ensure reporting gradually migrated to the new system

Conclusions / Recommendations

The author confirmed that ETL software and process could improve iPHIS reporting system performance and should be implemented to replace the existing replication process to improve reporting system performance. Similar to the replication system, the ETL process was able to process data in near real-time using CDC database feature. The ETL process could refresh data incrementally; therefore, should improve performance, be more scaleable, and more flexible and extensible. Since the ETL pilot project team did not complete the planned performance test and hence could not quantified the performance improvement, the full-scale ETL implementation should follow the Business Dimensional Lifecycle Methodology and proceed iteratively using Rapid-Prototype-Iterative Cycle, as described in chapter three, to provide performance check point and remedy, if required, in incremental iterations. This iterative approach will substantially reduce project risk.

On purchasing COTS software, the author concluded that software purchasers should conduct a proof-of-concept when evaluating and purchasing software which involved innovation process. Purchasers should make a purchase based on currently available features and not on promised features that would be included in future release as these features might never be included due to a variety of reasons.

The author concluded that a pilot feasibility study project should precede a full-scale implementation of the project especially when there was inadequate in-house expertise and experience with the technology involved. A pilot project was important for providing a roadmap to plan for full-scale implementation, to identify technology gaps and to confirm critical assumptions. Even though the pilot project team did not achieve all of its goals such as the performance testing, it had been useful in providing suggestion in moving ahead the full-scale project. By completing the pilot project, the project team was alerted to areas that needed attention. In addition, the pilot project team had identified that project planning was essential, regardless of the size of the project, in order to be on schedule and to meet all project goals.

Summary

In conclusion, the pilot project was considered successful with several lessons learned. The pilot project team successfully confirmed that the ETL process was suitable to replace existing data replication process to improve iPHIS reporting system performance. The pilot project team also successfully provided a roadmap for the full-scale implementation by identifying critical gaps such as the need for the CDC, dimensional model, and appropriate ETL software. The author made several recommendations based on the lesson learned from this pilot project. The author recommended that the pilot project should precede any full-scale project especially when a new technology was involved. When buying software, the author recommended purchasers to avoid buying vaporware and to do a proof of concept for the critical requirement before the actual purchase. Finally, the author recommended the use of the Business Dimensional

Lifecycle Methodology and Rapid-Prototype-Iterative Cycle for data warehouse related projects to substantially reduce project risk.

References

Ballard, C., Gupta, A., Krishnan, A., Pessoa, N. & Stephan, O. (2005). *Data Mart Consolidation: Getting Control of Your Enterprise Information*. Retrieved July 2006, from IBM Web site: http://www.redbooks.ibm.com/abstracts/sg246653.html?Open

Burleson, D. (2004). *New Developments In Oracle Data Warehouse*. Retrieved August 10, 2006, from Burleson Consulting Web site: http://www.dba-oracle.com/oracle_news/2004_4_22_burleson.htm

Drewek, K. (2005). *Data Warehouse Architecture: The Great Debate.* Retrieved March 10, 2006, from b.eye Web site: http://www.b-eye-network.com/view/693

Friedman, T., Beyer, M. A., & Bitterer A. (2006). *Gartner ETL Magic Quadrant*. Retrieved February 9, 2007, from Informatica Web site: http://www.informatica.com

Inmon, B. (1992). *Building the Data Warehouse.* New York: John Wiley & Sons, Inc.

Inmon, B. (2000). *Information Management: Charting the Course: The problem with Dimensional Modeling.* Retrieved January 35, 2007, from DM Review Web site: http://www.dmreview.com/article_sub.cfm?articleId=2184

Kimball, R. (1998). *The Data Warehouse Lifecycle Toolkit.* New York:John Wiley & Sons, Inc.

Kimball, R. & Caserta , J. (2004). *The Data Warehouse ETL Toolkit.* New York: John Wiley & Sons, Inc.

Kimball, R. & Ross, M. (2002). *The Complete Guide to Dimensional Modeling*, New York: John Wiley & Sons, Inc.

Oracle Corporation (2003). *On-Time Data warehousing with Oracle 10g – Information at the Speed of your Business.* Retrieved June 10, 2006, from Oracle Corporation Web site: http://www.oracle.com/technology/products/bi/pdf/10gr1_twp_bi_ontime_etl.pdf

Rittman, M. (2005). *An Update to the Dimensional Modeling Article.* Retrieved May 25, 2006, from DBAZINE.com Web site: http://www.dba-oracle.com/oracle_news/2005_8_31_Update_to_Dimensional_Modeling_Article.htm

Rittman, M. (2006). *Implementing Real-Time Data Warehousing Using Oracle 10g*. Retrieved May 15, 2006 from DBAZINE.com Web site: http://www.dbazine.com/datawarehouse/dw-articles/rittman5

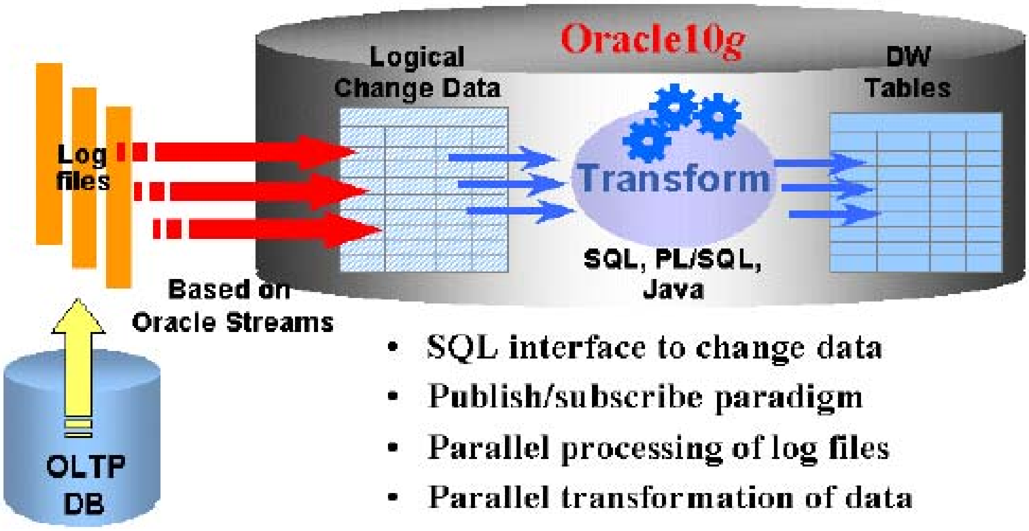
Schmitz, M. (n.d.). *Experiences with Real-Time Data Warehousing Using Oracle Database 10g*. Retrieved June 10, 2006, from http://download-east.oracle.com/owsf_2003/CDC_MSchmitzV2.ppt

Watson, J. H. & Ariyachandra, T. (2005). *Data Warehouse Architectures: Factors in the Selection Decision and the Success of the Architectures.* Retrieved April 25, 2006, from University of Georgia, Terry College of Business Web site: http://www.terry.uga.edu/~hwatson/DW_Architecture_Report.pdf

Wherescape Software Limited (2003). *Understanding the Data Warehouse Lifecycle Model.* Retrieved February 1, 2007, from WhereScape Web site: www.wherescape.com/Downloads/wherescape_dwlm[2].pdf

White, C. (2005). *Data Integration: Using ETL, EAI and EII Tools to Create an Integrated Enterprise*. Retrieved May 15, 2006, from ChannelWeb Web site: http://whitepaper.informationweek.com/search/index/cmpchannelweb/sol_summary/79326

Exhibit



- SQL interface to change data
- Publish/subscribe paradigm
- Parallel processing of log files
- Parallel transformation of data

Figure 3. Overview of the Near Real-Time ETL using Asynchronous Change Data Capture (Oracle Corporation, 2003)

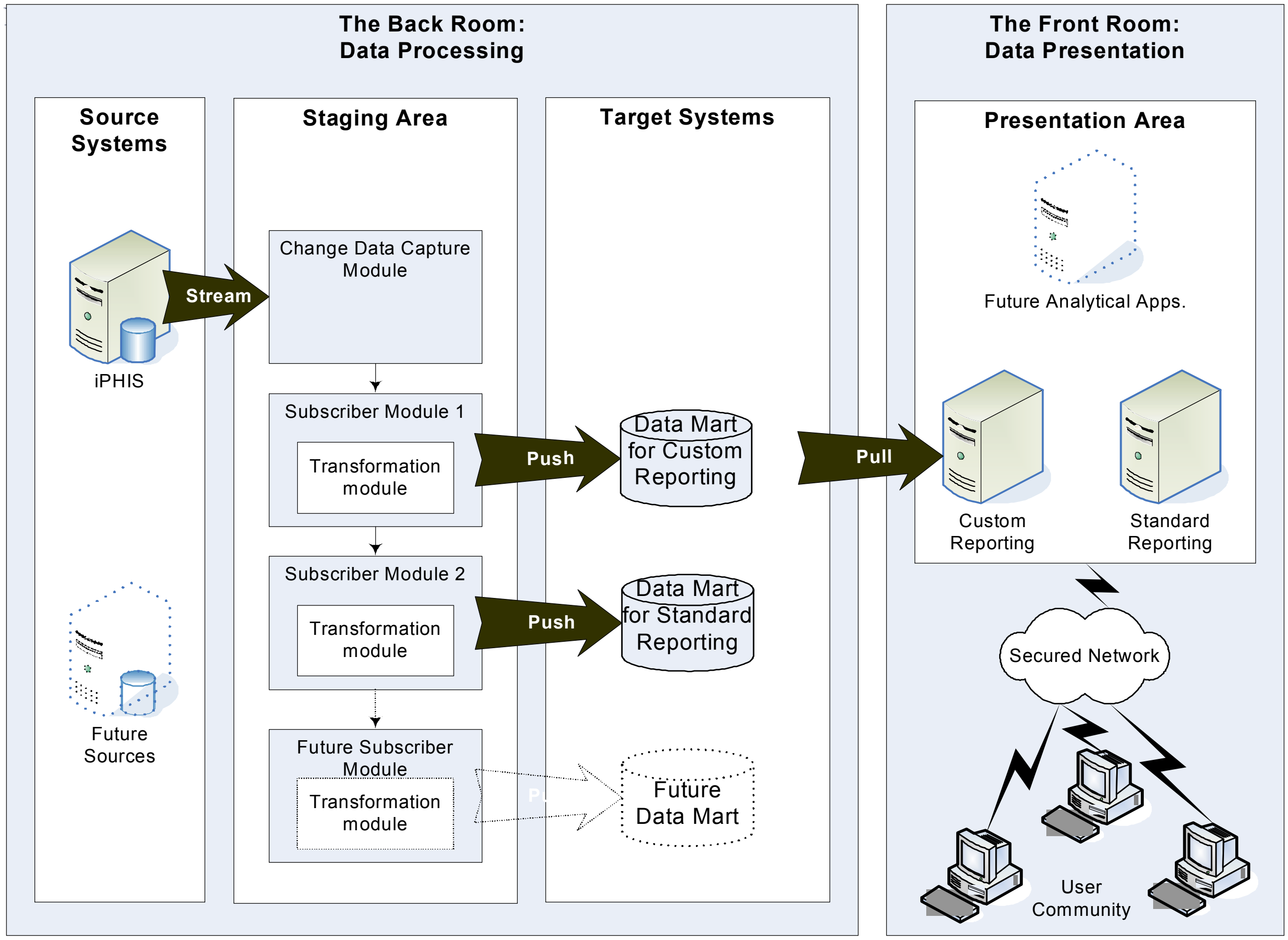


Figure 4. Overview of the iPHIS Near Real-Time ETL Process Physical Schematic

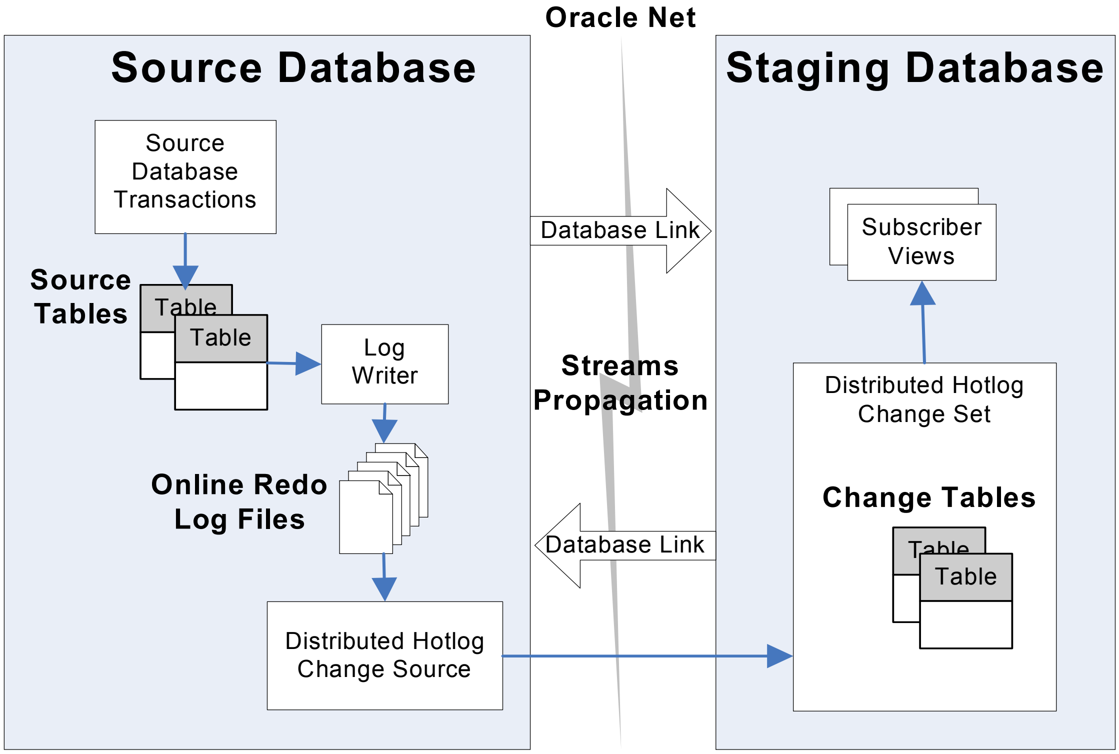
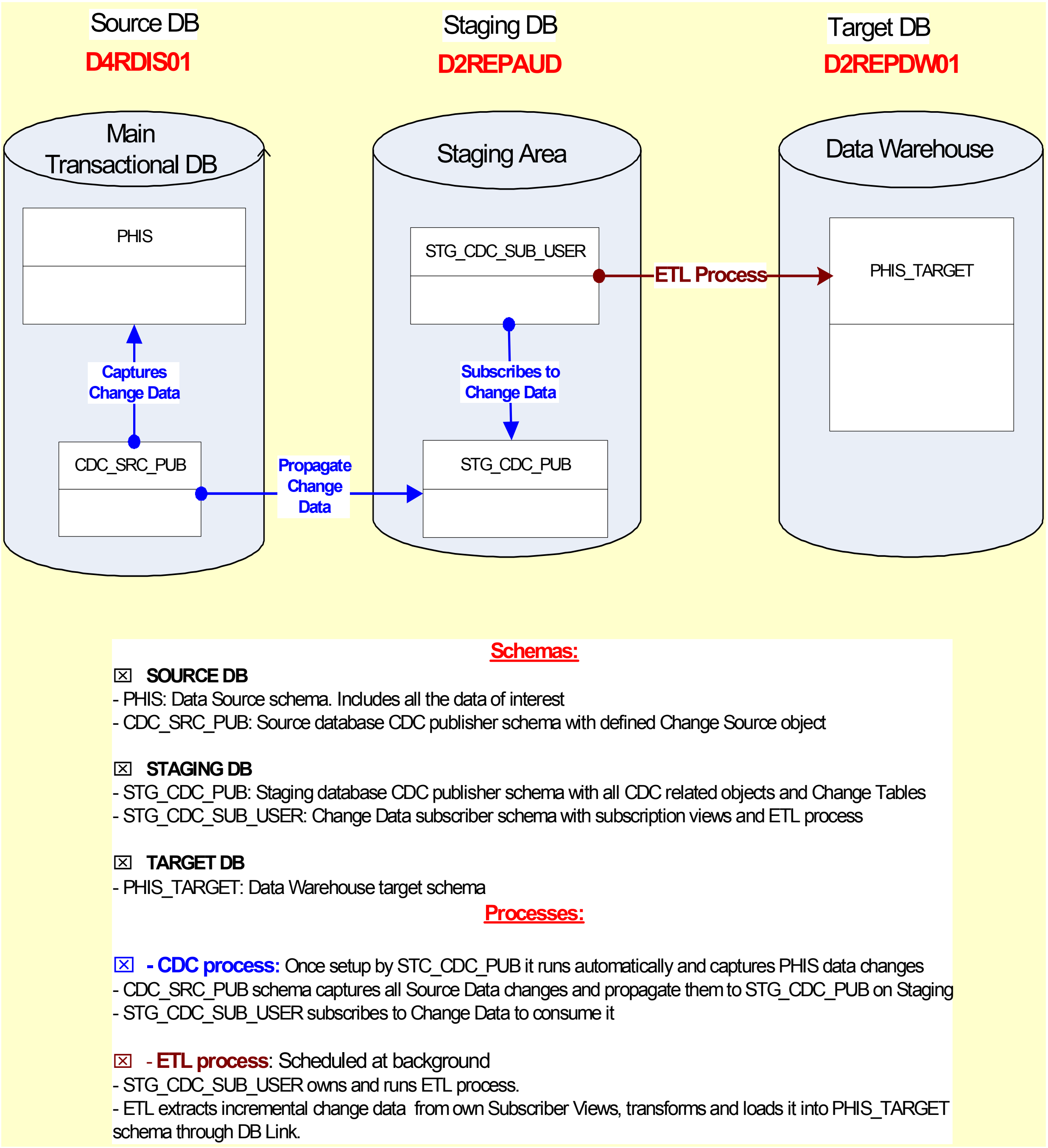


Figure 5. The iPHIS CDC Asynchronous Distributed Hotlog Logical Configuration



Schemas:

☒ **SOURCE DB**
- PHIS: Data Source schema. Includes all the data of interest
- CDC_SRC_PUB: Source database CDC publisher schema with defined Change Source object

☒ **STAGING DB**
- STG_CDC_PUB: Staging database CDC publisher schema with all CDC related objects and Change Tables
- STG_CDC_SUB_USER: Change Data subscriber schema with subscription views and ETL process

☒ **TARGET DB**
- PHIS_TARGET: Data Warehouse target schema

Processes:

☒ **- CDC process:** Once setup by STC_CDC_PUB it runs automatically and captures PHIS data changes
- CDC_SRC_PUB schema captures all Source Data changes and propagate them to STG_CDC_PUB on Staging
- STG_CDC_SUB_USER subscribes to Change Data to consume it

☒ - **ETL process**: Scheduled at background
- STG_CDC_SUB_USER owns and runs ETL process.
- ETL extracts incremental change data from own Subscriber Views, transforms and loads it into PHIS_TARGET schema through DB Link.

Figure 6. The iPHIS CDC Asynchronous Distributed Hotlog Physical Configuration

Source Table	Source Column	Target Column (table)
PHIS_GEN_CASE_INFOS		**PHIS_CASE**
	ID	PHIS_CASE_ID
	'CD'	DISEASE_TYPE_CD
	DCLT_ID	CLIENT_ID
	GBO_GHA_ID	HU_ID
	CASE_DATE	CASE_DATE
PHIS_OUT_EPISODE_INVSTGRS	GU_ID_CASE_INVESTIGATOR	INVESTIGATOR_USER_ID
PHIS_STD_CASE_INFOS		**PHIS_CASE**
	ID	PHIS_CASE_ID
	'STD'	DISEASE_TYPE_CD
	DCLT_ID	CLIENT_ID
	GBO_GHA_ID	HU_ID
	CASE_DATE	CASE_DATE
	GU_ID	INVESTIGATOR_USER_ID
PHIS_TB_CASE_INFOS		**PHIS_CASE**
	ID	PHIS_CASE_ID
	'TB'	DISEASE_TYPE_CD
	DCLT_ID	CLIENT_ID
	GHA_ID	HU_ID
	CASE_DATE	CASE_DATE
PHIS_TB_ENCOUNTERS	GU_ID	INVESTIGATOR_USER_ID
PHIS_GEN_USERS		**PHIS_USER**
	ID	USER_ID
	FAMILY_NAME	USER_FAMILY_NAME
	FIRST_NAME	USER_FIRST_NAME
PHIS_GEN_HEALTH_AREAS		**HEALTH_UNIT**
	ID	HU_ID
	HU_HD_NUMBER	HU_HD_NO
	DESCRIPTION_L1	HU_DESCRIP
	AREA_CODE	HU_AREA_CODE
PHIS_DEMG_CLIENTS		**CLIENT**
	ID	CLIENT_ID
	FAMILY_NAME	CLIENT_FAMILY_NAME
	FIRST_NAME	CLIENT_FIRST_NAME
	BIRTH_DATE	CLIENT_BIRTH_DATE
	GCL_ID_GENDER_CD	CLIENT_GENDER_CD
PHIS_GEN_CDS	EXPANDED_RESULT_L1	CLIENT_GENDER_DESCRIP

Figure 7. The iPHIS CDC Pilot Source and Target Tables Data Mapping

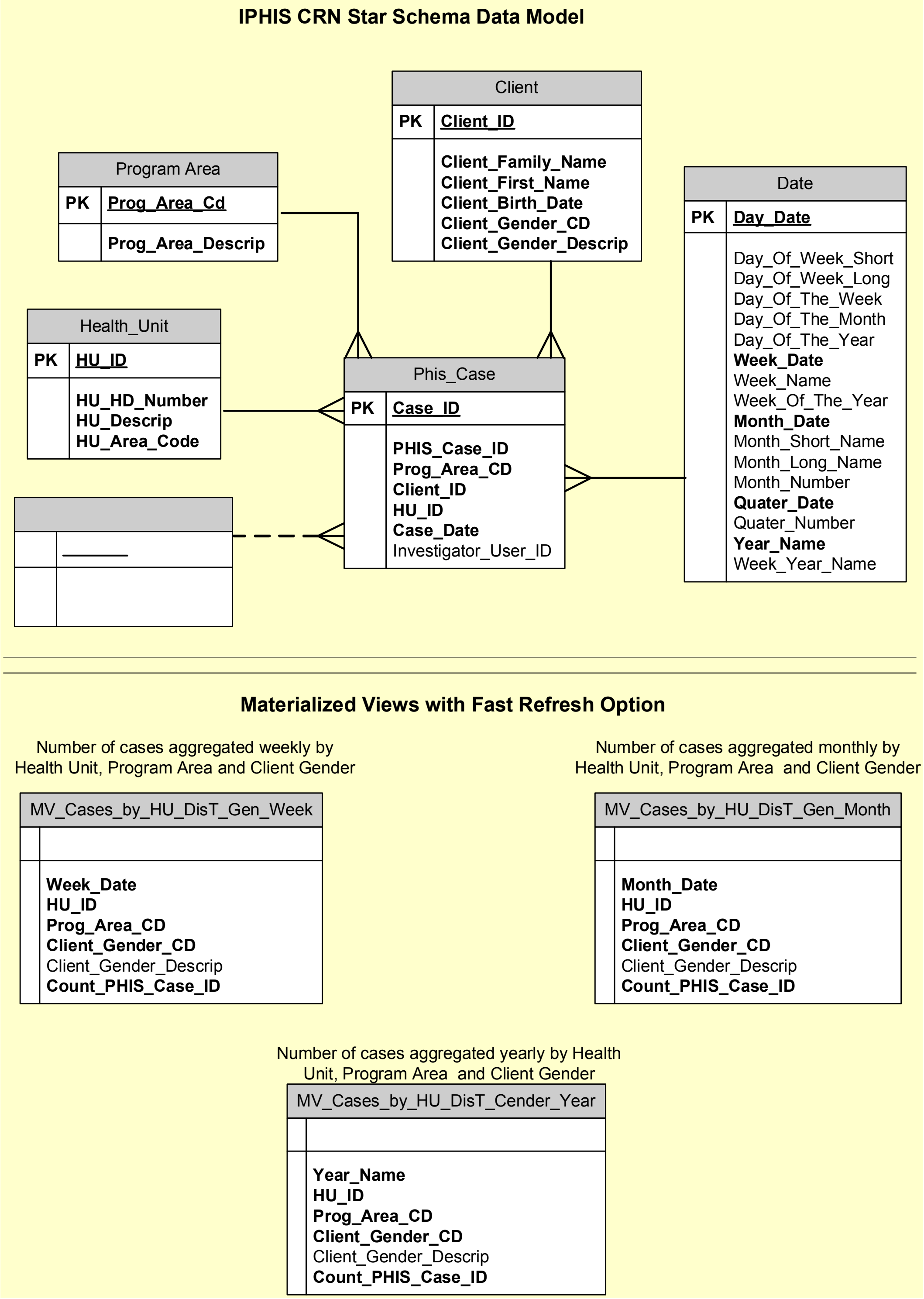
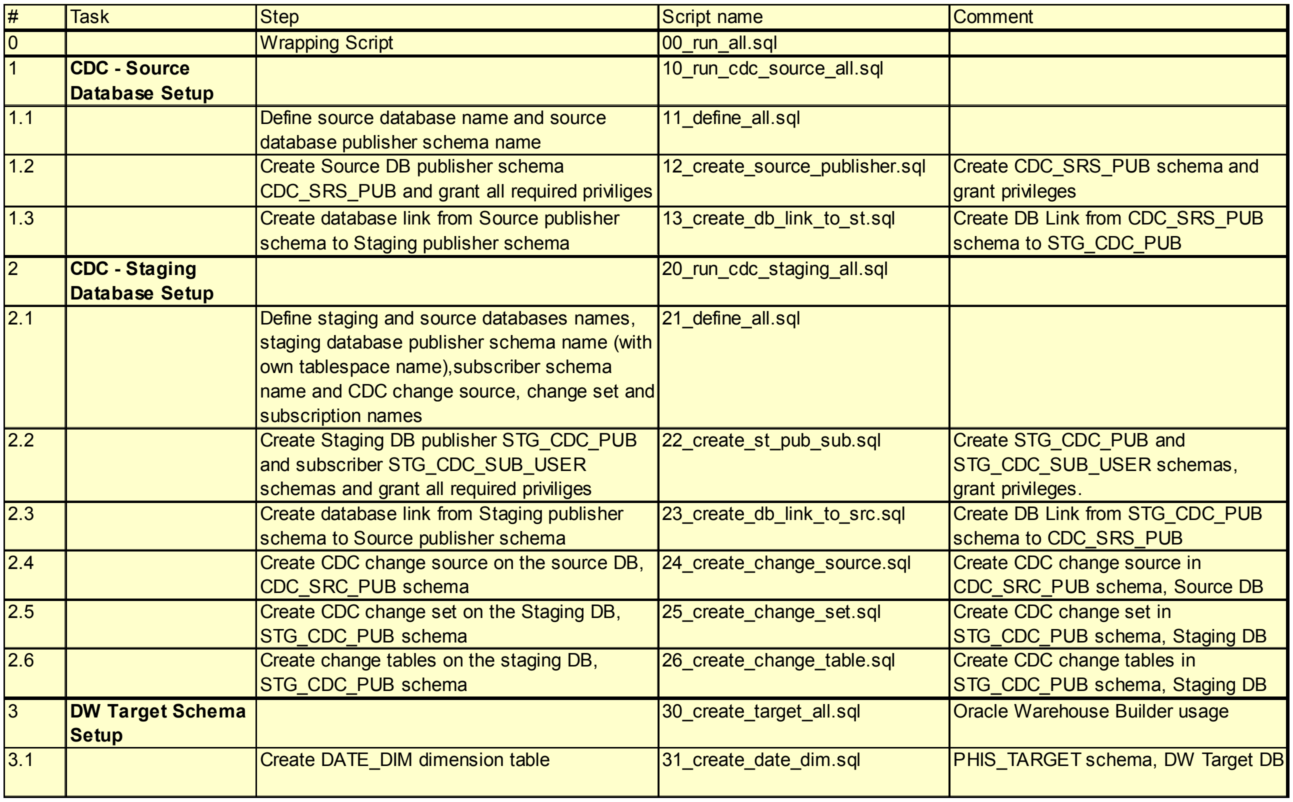


Figure 8. The iPHIS CDC Pilot Dimensional Data Model



#	Task	Step	Script name	Comment
0		Wrapping Script	00_run_all.sql	
1	**CDC - Source Database Setup**		10_run_cdc_source_all.sql	
1.1		Define source database name and source database publisher schema name	11_define_all.sql	
1.2		Create Source DB publisher schema CDC_SRS_PUB and grant all required priviliges	12_create_source_publisher.sql	Create CDC_SRS_PUB schema and grant privileges
1.3		Create database link from Source publisher schema to Staging publisher schema	13_create_db_link_to_st.sql	Create DB Link from CDC_SRS_PUB schema to STG_CDC_PUB
2	**CDC - Staging Database Setup**		20_run_cdc_staging_all.sql	
2.1		Define staging and source databases names, staging database publisher schema name (with own tablespace name),subscriber schema name and CDC change source, change set and subscription names	21_define_all.sql	
2.2		Create Staging DB publisher STG_CDC_PUB and subscriber STG_CDC_SUB_USER schemas and grant all required priviliges	22_create_st_pub_sub.sql	Create STG_CDC_PUB and STG_CDC_SUB_USER schemas, grant privileges.
2.3		Create database link from Staging publisher schema to Source publisher schema	23_create_db_link_to_src.sql	Create DB Link from STG_CDC_PUB schema to CDC_SRS_PUB
2.4		Create CDC change source on the source DB, CDC_SRC_PUB schema	24_create_change_source.sql	Create CDC change source in CDC_SRC_PUB schema, Source DB
2.5		Create CDC change set on the Staging DB, STG_CDC_PUB schema	25_create_change_set.sql	Create CDC change set in STG_CDC_PUB schema, Staging DB
2.6		Create change tables on the staging DB, STG_CDC_PUB schema	26_create_change_table.sql	Create CDC change tables in STG_CDC_PUB schema, Staging DB
3	**DW Target Schema Setup**		30_create_target_all.sql	Oracle Warehouse Builder usage
3.1		Create DATE_DIM dimension table	31_create_date_dim.sql	PHIS_TARGET schema, DW Target DB

Figure 9. The iPHIS Pilot Project Deployment Sample Script