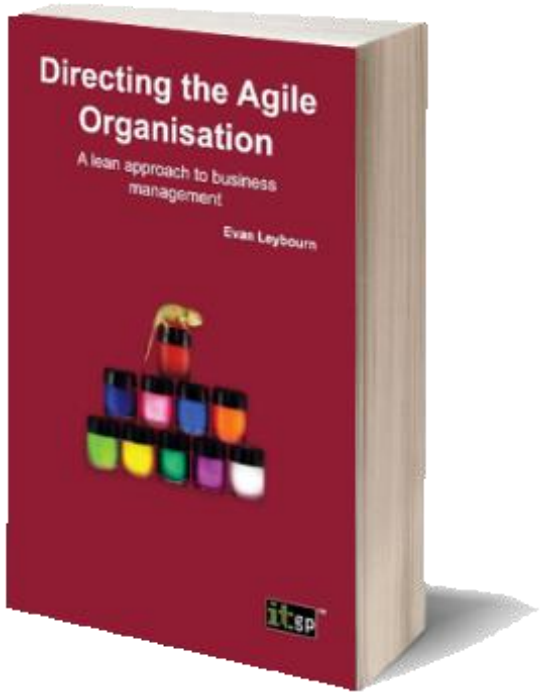
Introduction to Data Warehousing & Business Intelligence Systems

Student Guide

Evan Leybourn
evan@theagiledirector.com
Twitter: @eleybourn



OTHER WORKS BY EVAN LEYBOURN

DIRECTING THE AGILE ORGANISATION – BY EVAN LEYBOURN

http://theagiledirector.com/book

- Embrace change and steal a march on your competitors
- Discover the exciting adaptive approach to management
- Become the Agile champion for your organisation

Business systems do not always end up the way that we first plan them. Requirements can change to accommodate a new strategy, a new target or a new competitor. In these circumstances, conventional business management methods often struggle and a different approach is required.

Agile business management is a series of concepts and processes for the day-to-day management of an organisation. As an Agile manager, you need to understand, embody and encourage these concepts. By embracing and shaping change within your organisation you can take advantage of new opportunities and outperform your competition.

Using a combination of first-hand research and in-depth case studies, Directing the Agile Organisation offers a fresh approach to business management; applying Agile processes pioneered In the IT and manufacturing industries.

TABLE OF CONTENTS

INTRODUCTION

"Computers are getting smarter all the time. Scientists tell us that soon they will be able to talk to us. (And by 'they', I mean 'computers'. I doubt scientists will ever be able to talk to us.)"

- Dave Barry

Notes:

A Note About The Associated Slides

The presentation material for this course (also released under a Creative Commons BY-SA license) was created in Prezi.

You can locate the presentation here: https://prezi.com/pgituwqwxgiq/data-warehousing/

After Completing This Course

After completing this course, you should be able to do the following.

- Identify any organisational requirement for a Data Warehouse or Business Intelligence application.
- Understand how to improve an organisation's data and information.
- Understand what is involved in the creation and ongoing administration of an effective Business Intelligence system.
- Identify and analyse potential data sources inside and outside an organisation and how to use that data to improve the business intelligence of an organisation.
- Design a reporting plan that suits an organisational environment and improve information management.
- Use a Business Intelligence system to improve data integrity and quality.
- Create a database schema suitable for a Business Intelligence application.

Notes:

BUSINESS INTELLIGENCE PRINCIPLES

"Computers are useless. They can only give you answers."

- Pablo Picasso

Notes:

CREATING INFORMATION FROM DATA

The first step in any Business Intelligence project is to identify the data requirements of an organisation. There are two areas that need to be covered.

1. What data they have, and
2. What information they need.

For the sake of clarity, the terms data and information convey different meanings.

1. Data is the raw output of any database, website, log files or other data source.
2. Information is the processed and refined version of the data for human usage.

BUSINESS INTELLIGENCE PHASES

Designing a Business Intelligence system can be a complicated and time-consuming process. There are many factors, both technical and organisational, to consider and it is not always possible to resolve all of these in a timely manner.

There are four major areas to designing and creating a Business Intelligence system;

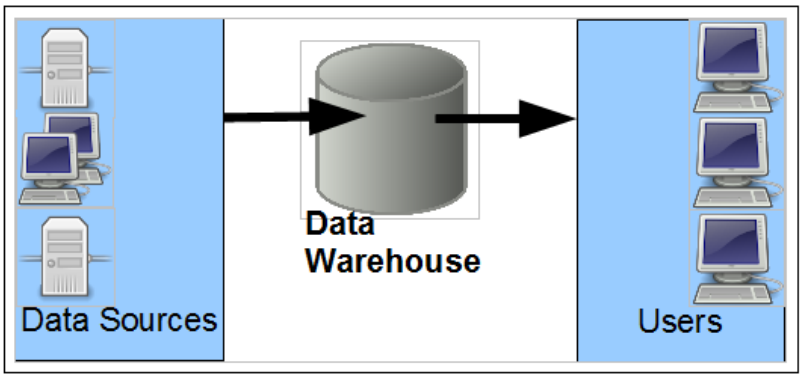
1. Analysis and Reverse Engineering
2. Design of the Consolidation DB and Data Marts
3. Extraction and Transformation of the data
4. Business level reporting on the data

Business Intelligence development can benefit from an iterative approach to development. Once you have identified your data sources, focus on a single data source and take it through all four areas of development.

Using an iterative approach you can have a limited production Business Intelligence system up and running in a short period of time, which users can be trained on and issues can be resolved.

Notes:

CENTRALISED REPOSITORY



All data within a Business Intelligence system is stored in a central repository called a "Consolidation Database".

A consolidation database allows an organisation to;

- Gain access to valuable data from multiple sources from a central location, increasing efficiency and reliability.
- Standardise and simplify the way information is managed within an organisation.
- Develop cross-application reporting to improve business intelligence.
- Identify obsolete, duplicate and redundant data across multiple applications.

COMBINING MULTIPLE SOURCES

As mentioned before, one of the most important aspects of a Business Intelligence system is the ability to combine multiple sources into a single consolidated database. However there are some issues that should be considered.

- How to resolve conflicts between data from different sources relating to the same record.
- How to correlate records that may have different primary keys to identify the record.
- How to accurately and efficiently store the data in the consolidation database, such that it does not duplicate existing records.

Notes:

- How to differentiate identical primary keys that refer to different records in the same consolidation table. (E.g. companies and universities in an organisation table.)

When a Business Intelligence system is being designed, it is important to keep these problems in mind. How each organisation resolves these problems is specific to each data source.

A good Extraction, Transformation and Load (ETL) design, and robust consolidation database will mitigate most of these problems. Thoroughly testing the ETL design with representative data from all data sources will limit potential issues.

Owning Your Data

One of the largest issues facing organisations today is the lack of data ownership. More and more organisations are purchasing software off-the-shelf and deploying them in mission-critical environments.

The problem occurs when it comes time to migrate from one vendor to another and the data is locked within a proprietary format, which you cannot access.

Open source software and some closed source vendors have started to promote the concept of "Owning Your Data". The premise is that your data should always be available and accessible regardless of license or software issues.

A Business Intelligence system can be central to this issue. By extracting information into the Business Intelligence system, you have complete ownership of all your data. It can also identify vendors and applications which restrict access to your data.

Case Study: A large professional organisation.

At a very early stage of the Business Intelligence development process, a vendor of a critical CRM application refused to allow analysis or extraction of the data into the Business Intelligence system, citing that by giving the organisation access they would have access to the vendor's intellectual property.

It was eventually decided to perform the extracts from the nightly backup dumps of the database, which were stored on the servers. Whilst not ideal, it was the only solution available.

Notes:

EXERCISE (YOUR ENVIRONMENT)

What BI Application do you use?

What do you want out of it?

What Information do you have?

What do you think is involved in a BI project (time, resources, etc)?

Notes:

INFORMATION MANAGEMENT

"On two occasions I have been asked, 'Pray, Mr. Babbage, if you put into the machine wrong figures, will the right answers come out?' [...] I am not able rightly to apprehend the kind of confusion of ideas that could provoke such a question."

- Charles Babbage, 1864

Notes:

GARBAGE IN, GARBAGE OUT (GIGO)

Garbage In, Garbage Out (abbreviated to GIGO) refers to the fact that, unlike people, computers will unquestioningly process all input data, regardless of purpose and meaning and produce nonsensical output.

In the realm of business intelligence this concept is doubly important, since the output of the Business Intelligence system is used the gauge the current state of the organisation.

At this time, some important things to note are.

- Ensure that all input to the system is correctly analysed and understood so that no errors occur from incorrect processing.
- Validate and any misleading or incorrect data is corrected during the extraction process. Also check the ETL tool error logs regularly to catch all errors as early as possible.
- Confirm the output of the Business Intelligence system against a secondary source before taking the results as correct. This should be done several times early on to ensure accuracy, and this should be regularly checked when new data sources are added.

Notes:

EXERCISE (GIGO)

Identify areas within your organisation with poor data?

How can you clean/transform that data to ensure accuracy?

Notes:

INFORMATION MANAGEMENT

Without good information management throughout your organisation, the results from the Business Intelligence system are untrustworthy and can, in extreme cases, harm your organisation. Keep in mind the following concepts;

- Ensure your data is consistent throughout its entire life-cycle, especially if you have multiple databases that track separate business tasks.
- Keep all records up to date, and validate against external sources as often as possible. A good example of this is address validation, customer addresses can be validated against your countries central postcode database
- Prioritise your information, and treat important information accordingly
- Train your staff in information management policies – A good worker who documents and accurately records their work is better than a great worker who doesn't.

Many large organisations still use Excel or Access to manage their data, especially if it is from non-core business area. With web based information systems incredibly cheap to buy or have custom built, there should be no excuse not get these business areas to upgrade.

A Business Intelligence system can help identify issues in your information management policies in two ways.

- During the business analysis phases it can identify business areas with weak information management systems.
- Once operational, reports can be compared with the expected results, and any anomalies can imply quality issues with the source data.

Notes:

DATA ACQUISITION

Data acquisition can be one of the hardest parts of a Business Intelligence process. Business areas do not want to share raw data, external data sources are slow and you have very little control over them, and you don't have access to competitor's data. This comes down to three types of data.

1. Data you Own
2. Data you can Access
3. Data you can Infer.

Data you own is the easiest form, these are data sources that belong to you organisation. Depending on the organisational structure, you may need to put cross-department data sharing policies in place, but that should be one of the first steps in a Business Intelligence process anyway.

Data you can access is more problematic. This includes public and government data, purchased data, and research data. Most of this data comes attached with licence restrictions and these must be considered carefully before inserting them into your Business Intelligence system. However some of the public and government data can be the most useful, as you can start to compare external factors (such as weather, geo-demographic, crime, interest rates, etc) against your internal data sources. This in turn can lead to some startling discoveries.

Data you can infer is the least accurate data, but often very useful. This includes information such as competitor prices, competitor income and expense, customer profiles, future trends, geo-demographic breakdown of your target audience, etc. By inferring this data you can act on the information and compare the inferred outcome against real, known data and over time start to gauge its accuracy. This is really good for identifying trends, advertising campaigns, and customer purchasing decisions.

Notes:

EXERCISE (DATA ACQUISITION)

What data do you own?

What data can you access?

What data can you infer?

Notes:

DATA REDUCTION

Always be aware that most senior executive in any organisation do not want to see large quantities of data.

What is important is to reduce the data to the core issues and features and whilst the Business Intelligence system should store as much data as possible, one of its primary goals is to report at the highest, most reduced, level.

REVERSE ENGINEERING

Reverse engineering is, in the context of data warehousing, the process of discovering the design principles and structure of a source system through analysis of its structure, function and operation.

During the analysis of the potential data sources, third party applications may need to be reverse engineered in order to understand the structure and availability of data within the application. This can be the only way to access your data from some vendors.

The process involves extracting the metadata on all the available relations and if possible the relationships from the data source, and building a data model and data dictionary from that.

Most E.R. modellers provide some reverse engineering tools.

It should be noted that, in many countries, even if an artefact or process is protected by trade secrets, reverse-engineering the artefact or process is lawful as long as it is obtained legitimately.

Notes:

DATA ANALYSIS

"Imagination is more important than knowledge. For knowledge is limited, whereas imagination embraces the entire world, stimulating progress, giving birth to evolution."

– Albert Einstein

Notes:

IDENTIFYING POTENTIAL DATA SOURCES

Before a Business Intelligence project can be started, potential data sources need to be identified. A data source is any object which contains business information.

Initially the data sources will be the main transactional and communication systems used within the organisation. Consider the following as potential sources;

- Human Resource Databases
- Client/Customer Relationship Management (CRM)
- Content Management Systems (CMS)
- Bespoke Application Databases
- Corporate Emails
- Reference Websites
- Wikis and Forums
- Documents and Spreadsheets
- Source Code Repositories
- Log Files

Systems

Notes:

EXERCISE (DATA SOURCES)

What Data Sources do you have (or want)?

Notes:

CREATING BUSINESS CASES FOR DATA SOURCES

Proper analysis and integration of foreign data sources into a Business Intelligence system can be a time consuming and complicated process. It is recommended that a brief business case is put together for each data source that is suggested. This can help prioritise data source analysis and extraction, as well as identify problem areas for each source.

An additional advantage is that it can help identify redundant systems and data.

Case Study: A large professional organisation.

When performing an analysis of data sources they identified that four regional divisions had developed a series of Microsoft Access databases to perform membership functions, duplicating the data within the primary CRM.

Not only did this analysis discover the databases, but they were able to identify flaws in the training of division staff.

If you have complete access to your data sources, and the system was written internally or you have access to the original data schemas, then the analysis can be fairly simple.

If it is a third party application, the analysis can be quite difficult and time-consuming. Things to consider are;

- What quantity of data is being dealt with?
- How frequently does this data change?
- When does this data change?
- How much of this data is duplicated elsewhere?
- How much of this data is obsolete or irrelevant?
- How is this data used?
- What reports currently use this data, and are they satisfactory?
- What access is available to the data, and how can we extract it?
- Is external or third party permission required before extracting this data, and how much will that cost?
- Does the organisation have the skills in-house to perform the analysis and extraction or will we have to out-source?

Notes:

EXERCISE (BUSINESS CASE)

Create a Business Case for a new database?

Notes:

TYPES OF SOURCES

When identifying data sources, it is important to understand the types of data sources that are available. These can include.

- Databases
- Documents
- Websites

- Emails
- Research
- Log Files

Each type has a unique method of extraction and translation. Databases are the most common source type, and unless otherwise mentioned are the default source type discussed during this course.

Notes:

DATABASES

The designer needs to be able to analyse the database schema and content. By the end of the data analysis stage the developer should be able to point at any column in any table and say what is contained therein and how it relates to the rest of the system.

To analyse a DBMS system access to the database schema is needed. In preference access should be via the schema diagram such as an E.R. Diagram. If this is unavailable, schema information can be extracted from a database connection such as ODBC/JDBC.

It may be impossible to extract the schema if there is only restricted access to the database. Delimited text extracts of the database should be made available that can be analysed. These should always be available for backup purposes. The analysis, at this point, is a highly manual process and will involve some guesswork.

If there is no success with any of these, then, as a final resort, the vendor should be able to provide the extraction queries. Understand that in this circumstance you will not “own” your data and will have no control of it.

What Databases would you include?

Notes:

DOCUMENTS

A good Business Intelligence system needs to be able to extract information from any available document. A lot of domain specific knowledge is stored within standard office documents, and can be very powerful to an organisation if collected. There are three primary document types that are important to store.

- Text files and word processor documents:

Information can be extracted from the file header, including author and modification dates. Depending on the format, these files can be separated by header for storage.

File formats include: Microsoft Word (doc), Open Document Format (odt), Rich Text Format (rtf), and Plain Text Files (txt).

- Spreadsheets:

These can be analysed as if they were a single database table.

File formats include: Microsoft Excel (xls), Open Document Format (ods) and Comma Separated Values (csv)

- Presentations:

Very little information can be extracted from these, but they can be split into slides if required. Most organisations would not store presentations, without a solid business case.

File formats include: Microsoft PowerPoint (ppt) and Open Document Format (odp).

What Documents would you include?

WEBSITES

Notes:

A lot of useful material is stored in websites, including research and reference material. Internal websites are usually database-driven, and a Business Intelligence system would normally access the database directly.

External websites can be very useful to extract data from, and can be used to improve the scope of available data for business intelligence and reporting.

Extracting from websites can be difficult as a website will normally contain a large quantity of extraneous information, such as advertising and navigation elements.

The extraction tool should be smart enough to follow links to extract the entire site if required.

What Websites would you include?

Notes:

RESEARCH

Because a lot of research is published online, a Business Intelligence system can extract a lot of this information from publicly accessible locations. It can be very useful for an organisation to keep abreast of current research.

Depending on the data source, and what access permission is available, the extraction tool can either gather research abstracts or complete papers.

What Research would you include?

LOG FILES

Corporate servers produce a lot of statistical data. Statistical information is usually stored in simple log files which can be processed and integrated into the Business Intelligence system.

This can be highly useful to correlate staff and clients with website and server usage.

What Logs would you include?

CORPORATE EMAILS

Notes:

A highly under-utilised aspect of data warehousing is the ability to extract from and report on corporate emails. There can be privacy issues with extracting from personal email automatically, but addresses such as sales@company.com, info@company.com and contact@company.com can contain a lot of very good information regarding who is emailing you, how often they email and what they are asking about.

A lot of information can be extracted from the email headers including who sent the email, where it came from, when, how it was routed to your servers and even information about their mail client.

Processing the body of the email can be difficult, with freeform text processing still in its infancy. However some information can be mined from it. It is important to make sure you strip all HTML and images from the email for maximum efficiency. Although, as with the attachments, can be inserted into the Business Intelligence application separately.

What Email Accounts would you include?

Notes:

Accessing the Data Sources

Depending on your data source there are numerous ways to extract the data from it. Remember, when deciding your extraction method, that you need to decide the connection method or protocol, as well as the query method.

Source Type	Connection	Query
Databases	ODBC, JDBC, Remote TCP/IP, Local path	SQL
Documents	HTTP(S), Local path	URL
Websites	HTTP(S)	URL
Research	HTTP(S)	URL
Log Files	HTTP(S) Local path	URL
Email	POP, IMAP(S)	

Building a Data Dictionary

A Data Dictionary is a set of metadata that contains definitions and representations of data elements. In the context of Business Intelligence, multiple data dictionaries are needed, one for each of the data sources, Data Marts and the consolidation database.

A good data dictionary may include both semantics and representational definitions for data elements. The semantic components focus on creating precise meaning of data elements. Representation definitions include how data elements are stored in a computer structure such as an integer, string or date format.

Initially, the data dictionaries will be a collection of database columns and the definitions of the meaning and types of data. Ideally, the data dictionary for a Data Warehouse will be comprehensive and simple to understand.

For a Data Warehouse a data dictionary should also contain a definition of how historical data is stored, and extracted from the data sources. This is critically important to understand, when developing a good Business Intelligence system.

Notes:

EXERCISE (DATA DICTIONARY)

Create a Data Dictionary for your data sources?

Notes:

A Business Intelligence System is Never Complete

It is good to think of Business Intelligence as a process, not a product. One of the most important lessons regarding data warehousing, and business intelligence in general is that it is never complete. The following need to be done periodically;

- New data sources need to be analysed and added as required for reporting.
- Custom reports will regularly need to be written. Even with an easy to use reporting tool, some people will turn to the BI team for complicated and large reports.
- Data validation and integrity checks need to be performed and followed up on.
- Standard database maintenance and tuning needs to be performed.

Long Term benefits

A good Business Intelligence system brings many long term benefits to any organisation, and if designed well and implemented correctly should support ongoing business intelligence improvement in the organisation.

- Improved decision making.
- A complete understanding of the data requirements and data production of an organisation.
- Break from vendor lock-in and simplify application migration.
- Giving end-users more access to organisation data.
- Build trend reports over the life of the Business Intelligence system to improve business capability and identify organisational weakness.

However, there are some concerns that will need to be addressed.

- The ETL process is very time-consuming
- Computer processing requirements are very high in any large Business Intelligence environment.
- Data security can be an issue. A good security policy is required to cover data access and storage.
- A strict design, scope, prototyping and release cycle is required to deliver a working system in a timely manner.

Notes:

DATA WAREHOUSE DESIGN

"First, solve the problem. Then, write the code."

- John Johnson

Notes:

THE CONSOLIDATION DATABASE

When building a Data Warehouse you are actually building numerous databases, not just one. The first database you need to design is the consolidation database. This database should contain all the information from your data sources.

Periodically, the extraction system will extract data from your sources and insert it into the consolidation database. As this will store all the data from every source it will become very large. Historical information from these sources must be kept forever, so an incremental extraction/insertion system is the best solution.

Both the consolidation database and the Data Marts are storage mechanisms for read-only, historical, aggregated data. By read-only, we mean that the person looking at the data won't be changing it. If a user wants to look at yesterday's sales for a certain product, they should not have the ability to change that number.

DATA MARTS

Data Marts are small subsets of the consolidation database. Consolidation databases are generally slow to query. They are designed for bulk inserts with complicated triggers (explained later in this chapter). The Data Marts are databases designed for queries and reporting. If done well, they can be orders of magnitude faster to query than the consolidation.

Data Marts are very specific to individual reporting needs, so they must be designed with those needs in mind. However remember these four things:

1. The Data Marts should also contain historical / incremental data, so the table structure should be similar to the consolidation database.
2. When designing Data Marts forget everything you know about normalisation. Merge table, duplicate data, anything goes. Although counter-intuitive, it will greatly improve performance.
3. Finally, make judicious use of indexes. As this is not a transactional system and is primarily used to report on data, the overhead of indexes on inserts and updates is insubstantial.
4. Your denormalised schema for the Data Mart should be purpose designed for reports from a subset of the data.

Notes:

EXERCISE (DATA SOURCES)

What is the relationship between each of your data sources (Diagram)?

Notes:

DBMS Choices

There are a lot of Database Management Systems (DBMS) on the market today, both open and closed source. Two of the most powerful, and popular systems are Oracle and PostgreSQL.

When developing a Data Warehouse consider the skills within the organisation. If the organisation uses another DBMS for most of its production systems, utilise the skills and value within the organisation and develop the Data Warehouse on that DBMS.

Most of the examples in this course will cover building a Data Warehouse on either Oracle or PostgreSQL, but the rest of the topics remain vendor neutral. Both DBMS's have wide experience with Data Warehouse applications, are incredibly powerful and can scale to enormous data sets.

Both of these databases have similar capabilities and are highly compliant with the SQL standard. PostgreSQL is a very professional open source application which is free of charge and free of ongoing licences. Oracle, whilst expensive, is an excellent DBMS and has a lot of integral functionality designed for Data Warehouses.

A Data Warehouse Table

```
CREATE TABLE person (
  person_id TEXT,
  title TEXT,
  family_name TEXT,
  given_names TEXT,
  gender TEXT,
  date_of_birth DATE,
  passphrase TEXT,
  start_timestamp_id TIMESTAMP REFERENCES dw_timestamp,
  end_timestamp_id TIMESTAMP REFERENCES dw_timestamp,
  UNIQUE (person_id, start_timestamp_id)
);
```

Both the consolidation database and Data Marts are built with a very similar structure, even though they have very different purposes.

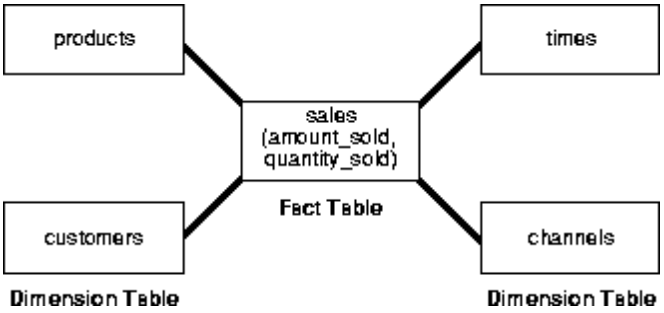
Notes:

There are five aspects of this table to note.

1. person_id: This is the primary key from the source database. Due to the incremental nature of this dataset, there will be more than one row with the same person_id. This also means that person_id, and person_timestamp_id should be used as a composite foreign key.
2. start_timestamp_id: This is a date/time of the start of the insertion process. Every row you insert, in a given ETL process, will have the same start_timestamp_id
3. end_timestamp_id: When did this row become obsolete? If this is a non current value then the row has either been deleted from the source database, or has been updated, in which case there will be another row in the consolidation database with the same person_id. The end_timestamp_id will be updated with the new start_timestamp_id.
4. There is a unique constraint against the person_id combined with start_timestamp_id. This ensures that the person_id is unique in each incremental snapshot.
5. OPTIONAL: Each type is TEXT, because it's best not to trust the source database. If an assumption is made on the content of the data and it breaks that assumption, the row will not be inserted and you will have an incomplete extraction. You should be running complete sanity checks and transformations against every insert to correct these errors before they get into the database, and to restrict the data type appropriately.

Notes:

STAR SCHEMA



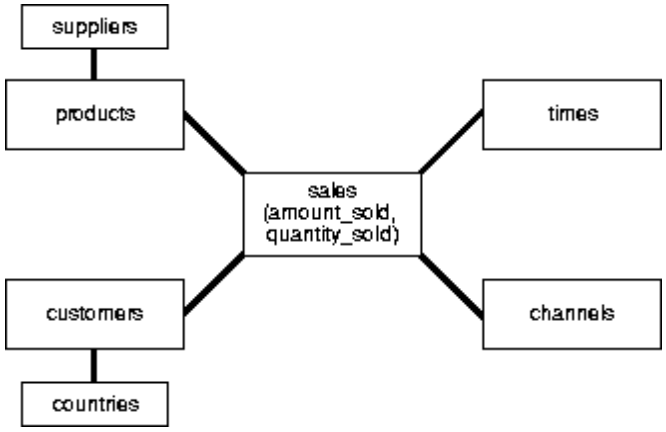
The star schema is perhaps the simplest Data Warehouse schema. The centre of the star consists of a large fact table, with a compound primary key, with one segment for each dimension and additional facts.

The points of the star are the dimension tables, or lookup tables, each of which contains information about the entries for a particular attribute in the fact table.

A good reason for using star schema is its ease of understanding. Fact tables in star schema are mostly in third normal form (3NF) meaning that there is no duplicate data across any table, but dimensional tables are in de-normalised second normal form (2NF) which has some duplication of data. If you want to normalise dimensional tables, they look like snowflakes (see snowflake schema) and the same problems of relational databases arise - you need complex queries and business users cannot easily understand the meaning of data.

Notes:

SNOWFLAKE SCHEMA



The snowflake schema is a type of star schema. The difference is that it's a more complex Data Warehouse model.

Snowflake schemas normalise dimensions to eliminate redundancy. That is, the dimension data has been grouped into multiple tables instead of one large table. While this saves space, it increases the number of dimension tables and requires more foreign key joins. The result is more complex queries and reduced query performance.

Also, many Business Intelligence systems are designed to be used by business users, and the added complexity of the snowflake schema will often preclude non-specialist users from forming their own queries.

Notes:

EXERCISE (SCHEMA)

Star Schema Workshop

Notes:

HISTORICAL DATA

Producing reports based on historic trends is critically important for most organisations and one of the primary advantages of a good Business Intelligence system. Each record in the Data Warehouse has a start and end timestamp attribute. These timestamps detail when a record was inserted and when it was updated or deleted from the source.

Each group of like records make up an incremental snapshot of a single data source record. For each record in the data source:

- Unchanged: Update the end timestamp of the current row to the current timestamp.
- Updated: Create a new row in the Data Warehouse with the start (and end) timestamp set to the current timestamp. The old row should stay the same with the end timestamp marking the date the record was obsoleted.
- New (Inserted): Create a new row in the Data Warehouse with the start (and end) timestamp set to the current timestamp.
- Deleted: This row will never be extracted from the data source (because it doesn't exist). Thus the current row's end timestamp will mark the date the record was obsoleted.

DATABASE TRIGGERS

Many data warehouse systems utilise a series of predefined triggers to perform the incremental extracts. These triggers will run through the inserts and check them against the current row, updating the end_timestamp_id as they go. The current row is the one where the person_id matches and there is no end_timestamp_id.

- If there is no current row (if it is a new entry) then the new row is inserted.
- If there is a difference between the row to be inserted and the current row (there has been an update to the source database) then the current row is marked obsoleted, and the new row is inserted.
- If there is no difference then the insertion is skipped.
- Finally, you check for any rows that have not been touched and you mark them as obsoleted (They have been deleted from the source_db).

INDEXES

Notes:

A database index is a data structure that improves the speed of operations in a table. Indices can be created using one or more columns, and for every insert or update those columns get duplicated in the index in alphabetical order.

Indices increase the time taken to insert and update a record within the database, but significantly reduce query time. A Data Warehouse is, by its nature, a highly query-based system and as such indices greatly increase efficiency.

Indices are automatically created for all unique columns and all primary and foreign keys. Additional indices can be added for any columns that are frequently used within query constraints. For example:

- Age / Date of Birth
- Gender
- Family Name
- Invoice / Receipt Date
- Invoice / Receipt Amount
- Membership Date

Notes:

EXTRACTION, TRANSFORMATION AND LOAD

"Any fool can use a computer. Many do."

- Ted Nelson

Notes:

DATA VALIDATION

Often, applications do not perform sufficient validation checks when accepting user input. Off-the-shelf software is particularly prone to validation errors. The extraction process can validate the data and raise a flag if it fails.

Example: Date of Birth

An off-the-shelf application may not check that the date of birth of a staff member falls within a predefined range.

Validation can check;
01-01-1930 < DOB < 01-01-1990
i.e., the staff member is under 75 and over 15.

If Data fails the validation checks, you can either:

- Insert it into the consolidation database, and alert the administrators who can then check all errors in the database, and correct them in the data source. This method ensures that all data is captured in the Data Warehouse.
- Not insert into the consolidation database but still alert the administrators who can then check the insertion errors, and correct them in the data source. This method ensures that only accurate data is captured in the Data Warehouse.

Notes:

DATA INTEGRITY

Organisations which have large datasets and multiple database systems often have problems with data integrity. Minor corruptions and data errors can develop slowly over time, which can cause large reporting errors, and even corrupt an entire data system. Many older applications are prone to these integrity problems.

The extraction process can automatically identify most of these problems and should alert the administrators to these errors.

Case Study: A large professional organisation.

Due to license issues with a third party application provider, all data was being extracted from a series of text database dumps.

During the early tests of the extraction process, it was discovered that within the data files were several binary streams of data which had corrupted a large block of important data.

The ETL process identified these problems and they were fixed in the source system.

Notes:

DATA STANDARDISATION

With data from multiple sources all stored within the consolidation database, it is important to be able to specify a data convention that can apply across every system.

During the extraction process it is important to identify similar and duplicated data across all available data sources and ensure it is transformed into a standard form.

Case Study: A Sydney based University

Three different application databases, the HR database, course management database, and library management database each stored similar staff details, but in different formats. The simplest example was the HR and Course Management systems. They added a different prefix to the staff number, which needed to be standardised.

During the extraction process, a series of transformations were written which automated this standardisation process.

Common standardisations;

- Gender: Male / Female => M / F etc.
- Case Sensitivity: university => University
- Addresses: Line 1 / Line 2 => Company / Street

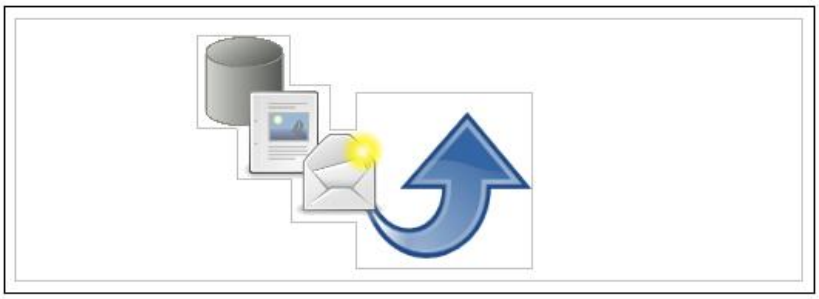
Notes:

EXERCISE (VALIDATION)

Data Validation/Standardisation Workshop

Notes:

EXTRACTION



The first step in the ETL process is extraction.

Extraction exports data from the data sources on a periodic basis. The different source types each have their own extraction methods.

- Databases: Data is extracted via remote SQL queries, usually an ODBC connection.

SELECT * FROM PERSON WHERE expired=FALSE;

- Documents: Data is extracted via local directory paths or remote URLs, usually a HTTP/S connection.

/Documents/Corporate/design.odt

- Websites: Data is extracted via remote URLs, usually a HTTP/S connection.

http://www.url.com/file.html

- Log Files: Data is extracted via local directory paths or remote URLs, usually a HTTP/S connection.
- Email: Data is extracted through an email connection, usually a POP or IMAP connection.

Most ETL applications require a series of control files to be written which control the extraction (as well as the transformation and load).

Notes:

TRANSFORMATION

The second step in the ETL process is transformation.

Transformation iterates through the extracted data and applies a series of predefined rules to each row of data. During this stage the ETL tool performs all the validation, standardisation and modification requirements.

Common transformations are;

- Adding arbitrary content
- Removing arbitrary content
- Modifying the content of the cell.
- Splitting a cell into two.
- Joining cells together.
- Dropping an entire row from insertion.

Notes:

ADDING DATA

Arbitrary columns, with pre-set values, can be added to the consolidation database during the extraction process.

This is most often used to add descriptive details alongside extracted information.

Real World Example: Organisation Table

A source database with two tables; one for companies and another for tertiary institutions. Within the consolidation database is one table called organisations. When extracting and loading these columns into the Data Warehouse we want to add an arbitrary column that specifies what type of organisation it is.

```
<input>
      <select>SELECT * FROM company</select>
      <transform>
            <type>add</type>
            <out>organisation_type</out>
            <value>Company</value>
      </transform>
</input>
<input>
      <select>SELECT * FROM ti</select>
      <transform>
            <type>add</type>
            <out>organisation_type</out>
            <value>Tertiary Institution</value>
      </transform>
</input>
```

Notes:

DELETING DATA

Any column that has been extracted from the data source can be explicitly deleted from the load.

If your extraction query contains a wildcard (*), then you may wish to delete extraneous columns.

Real World Example: Person Table

Because of the incremental nature of the Data Warehouse, we do not need to store the time the customer record was modified.

As there are 20 columns in the customer table, we use the '*' wildcard, to minimise the query length.

```
<input>
      <select>SELECT * FROM customer</select>
      <table>person</table>
      <transform>
            <type>delete</type>
            <in>modified_time</in>
      </transform>
</input>
```

Notes:

MODIFYING DATA

Any column that has been extracted from the data source can be modified and transformed into any style you require.

Different applications do this in different ways. One of the most powerful methods is regular expressions. A regular expression is a string that describes or matches a set of strings, according to certain syntax rules. Regular expressions are used by many text editors and utilities to search and manipulate bodies of text based on certain patterns.

Real World Example: Corporate Emails

When storing corporate emails, we only want to store plain text. The following regex will strip all HTML elements.

```
<input>
      <table>email</table>
      <transform>
            <type>modify</type>
            <in>body</in>
            <out>body</out>
            <action>s/<[^<>]+>//g</action>
      </transform>
</input>
```

Notes:

A QUICK OVERVIEW OF REGULAR EXPRESSIONS (REGEX)

A regular expression, or regex for short, is a pattern describing a certain amount of text, and optionally describing a replacement string.

Regex Format

Different regular expression engines can use different formats to wrap the expression itself. The most common is the Perl regex Engine. The format is;

Match a string.

```
m/SEARCH/
```

Search and Replace

```
s/SEARCH/REPLACE/
```

Global Search and Replace

```
s/SEARCH/REPLACE/g
```

Literal Characters

The most basic regular expression consists of literal characters, e.g.: “a”. It will match the first occurrence of those characters in the string. If the string is “Data warehousing can improve business intelligence”, it will match the first a after the D.

This regex can match the second a too. It will only do so when you tell the regex engine to start searching through the string after the first match.

Character Classes

A character class will match only one out of a range of possible characters. To match an "a" or an "e", use “[ae]”. You could use this in “gr[ae]y” to match either “gray” or “grey”. A character class matches only a single character. “gr[ae]y” will not match “graay”, “graey” or any other thing.

It is important to note that the order of the characters inside a character class does not matter.

Notes:

You can use a hyphen inside a character class to specify a range of characters. “[0-9]” matches a single digit between “0” and “9”. You can use more than one range. “[0-9a-fA-F]” matches any single hexadecimal digit, case insensitively.

Typing a caret “^” after the opening square bracket will negate the character class. The result is that the character class will match any character that is not in the character class. “h[^o]” matches “he” in “hello”, but it will not match “hope”.

There are some default shorthand character classes which allow you to match objects very quickly.

- “\d” matches a single character that is a digit
- “\w” matches a "word character" (alphanumeric characters plus underscore)
- “\s” matches a whitespace character (includes tabs and line breaks).

Special Characters

There are eleven characters (or metacharacters) with special meanings;

Sign	Name	Description	Example
[	Open Square Bracket	Start a character class.	“*gr[ea]y”* will match “*grey”* and “*gray”*
****	backslash	Turns metacharacters into normal characters.	“*3\.14”* will match “*3.14”*
^	caret	Within a character class. This will negate the character class. Outside a character class. This will match the beginning of a line.	“*gr[^ea]y”* will match “*grqy”* etc, but not “*grey”* or “*gray”.* “*^the”* will match *“the world”* but not “*world, the”*
$	dollar sign	This will match the end of a line.	“*the$”* will match *“world, the”* but not “the *world”*
.	period or dot	Matches one character regardless of the character.	“*3.14”* will match “*3x14”, “3514”, “3f14”* etc

Notes:

|	vertical bar or pipe symbol	Matches either or.	"*foo|bar"* will match "*foo"* and "*bar"* but not "*baz".*
?	question mark	Matches the preceding character 0 or 1 times.	"*m?ethane"* will match *"ethane"* and *"methane".*
*****	asterisk or star	Matches the preceding character 0 or more times.	"*m*ethane"* will match *"ethane"* and *"methane"* and "*mmethane"* etc.
+	plus sign	Matches the preceding character 1 or more times.	"*m+ethane"* will match *"methane"* and "*mmethane"* etc.
(	opening round bracket	Start a character group	"*a(bc)?"* will match "*a"* and "*abc"*
)	closing round bracket	End a character group	

To match "1+1=2", the correct regex is "1\+1=2". Otherwise, the plus sign will have a special meaning.

Notes:

SPLITTING DATA

Your ETL system needs to be able to split apart a data based on a given delimiter.

The delimiter can either be a string or a regular expression.

Real World Example: Email Table

We needed to be able to split an email address into the username and the domain. This is a very simple split based on the @ symbol.

```
<input>
      <select>SELECT * FROM email</select>
      <table>email</table>
      <transform>
            <type>split</type>
            <in>email_address</in>
            <out>username,domain</out>
            <action>@</action>
      </transform>
</input>
```

Notes:

JOINING DATA

Your ETL system needs to be able to join multiple columns together from the source data into a single column in the Data Warehouse.

The join should be separated by an optional delimiter

Real World Example: Email Table

This is the exact opposite of the example above. Whilst the organisation required the email split in the consolidation database, they needed a complete email address within one of the Data Marts.

```
<input>
      <select>SELECT * FROM email</select>
      <table>email</table>
      <transform>
            <type>join</type>
            <in>username,domain</in>
            <out>address</out>
            <action>@</action>
      </transform>
</input>
```

Notes:

DROPPING ROWS

For data validation purposes you must be able to drop entire rows from the load based on the value, or transformed value, of a given column.

This ensures that your Data Warehouse (or Data Mart) is in a stable and consistent state at all times.

Real World Example: Email Table

Email data can be critical to an organisation, so a lot of effort often goes in to making the data stable. At this point we want to drop (and raise an alert) all rows which contain a non-valid email address using regex.

```
<input>
      <select>SELECT * FROM email</select>
      <table>email</table>
      <transform>
            <type>drop</type>
            <in>email_address</in>
            <action>/^([a-zA-Z0-9_\.\-])+\@(([a-zA-Z0-9\-])+\.)+([a-zA-Z0-
9]{2,4})+$/</action>
      </transform>
</input>
```

Notes:

EXERCISE (TRANSFORMATION)

Transformation Workshop?

Notes:

LOAD

The final step in the ETL process is to populate the transformed records into the Data Warehouse. As each row is inserted into the consolidation database, the Data Warehouse triggers are fired.

Depending on the DBMS and Business Intelligence application used, these triggers can perform additional data validation and the incremental checks.

Depending on the complexity of the triggers and the indexes on the database tables, this process can be very time-consuming. Depending on the database design, it can be faster to drop all the indices on the tables for the load and recreate them after the ETL process.

Notes:

RESOLVING ERRORS

During the ETL process, records can be discovered that contain errors. As discussed before, there are two types of errors.

- Records that fail a transformation (e.g. for validation or standardisation reasons) are marked as erroneous records.
- Records that fail to load in the database based on data integrity issues are also marked as error records.

All erroneous records should be flagged for administrator or DBA attention, to be addressed as early as possible.

Erroneous records should always be resolved in the master data source to ensure stable and reliable data. These erroneous records can also identify logical flaws in the design of the Business Intelligence system, and are important to address early in the development cycle.

HISTORICAL EXTRACTION

In principle, a Business Intelligence system should contain a complete historical log of all changes made within the original data sources. Depending on organisational requirements, it is very important to retroactively insert historical data.

- This is a very time consuming process, and can take weeks or months to complete a full historical regeneration, when dealing with large amounts of data.
- Not all data sources will store historical data which can be extracted.
- Data sources which have historical data can have incomplete or simplistic data, such as;
    - A record may store the fact that it is no longer current, e.g. cancelled.
    - A record may store the modified by and modified time, but not the actual changes.
    - A record may store the changes in a audit log file, which may be difficult to process and extract from. Especially if they use natural language as part of the record. e.g. Person Name has Changed To X From Y.
- Ideally the changes of a record will be stored in a series of incremental records.

Notes:

REPORTING

"No matter how slick the demo is in rehearsal, when you do it in front of a live audience, the probability of a flawless presentation is inversely proportional to the number of people watching, raised to the power of the amount of money involved."

- Mark Gibbs

Notes:

WRITING GOOD REPORTS

There are three critical elements required for a Business Intelligence system to be useful to an organisations decision making capability.

- The data sources that supply the data to the Business Intelligence system must be appropriate to the business requirements. Irrelevant information can complicate the Business Intelligence system and in the worst case provide misleading information.
- The input data must be cleansed and corrected. Incorrect or misleading data will corrupt the results of any reporting.
- Most importantly, the person writing the report must understand the data, what it means and how to extract that information.

Any user who is expected to report on data from the Business Intelligence system must be given training on the usage of the Business Intelligence software, and just as importantly on the available data and structure within the Data Marts.

By training a user in the data structure and content, as well as providing reference materials (such as a simplified data dictionary) users can create reports effectively and accurately.

Notes:

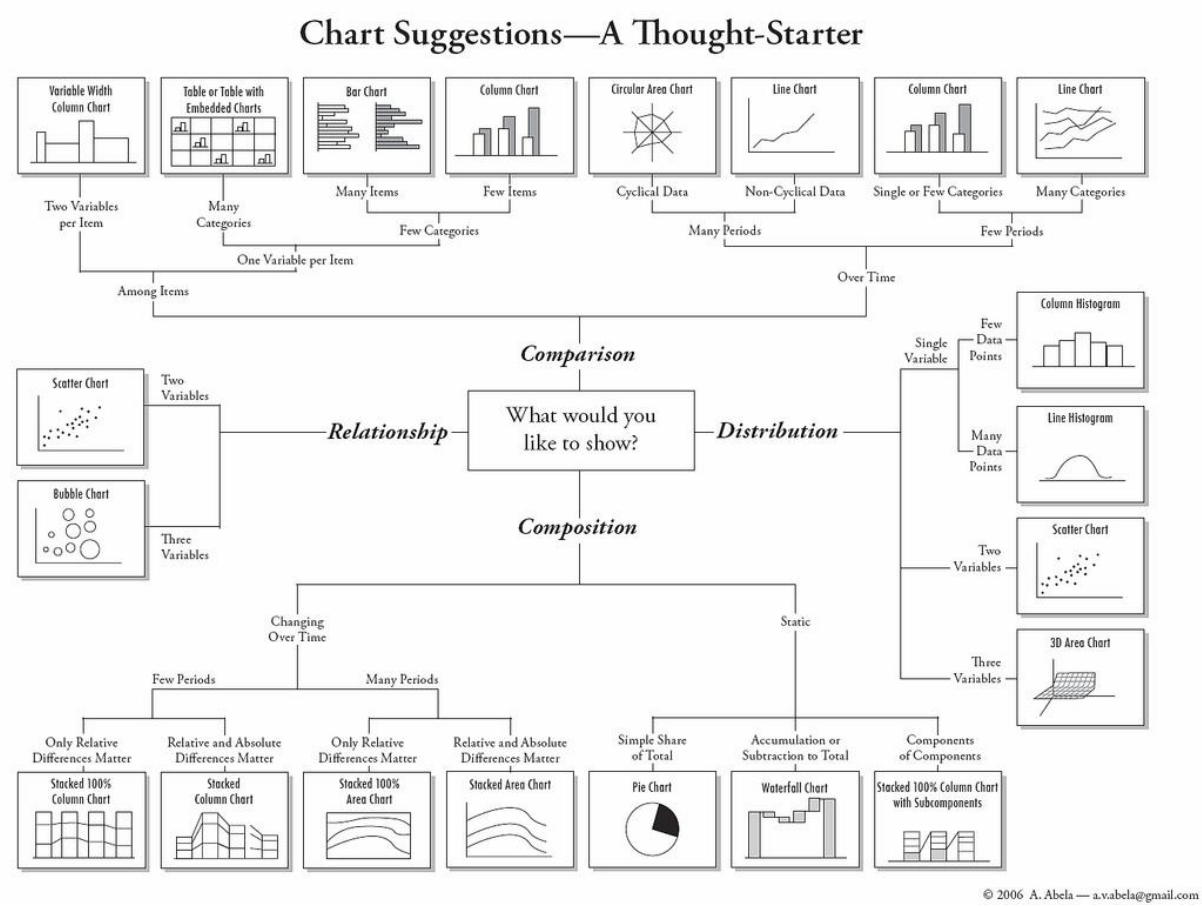


Chart Suggestions—A Thought-Starter
Variable Width Column Chart
Table or Table with Embedded Charts
Bar Chart
Column Chart
Circular Area Chart
Line Chart
Column Chart
Line Chart
Two Variables per Item
Many Categories
Many Items
Few Items
Few Categories
Cyclical Data
Non-Cyclical Data
Single or Few Categories
Many Categories
Many Periods
Few Periods
One Variable per Item
Among Items
Over Time
Comparison
Scatter Chart
Two Variables
Relationship
What would you like to show?
Distribution
Bubble Chart
Three Variables
Composition
Single Variable
Few Data Points
Column Histogram
Many Data Points
Line Histogram
Two Variables
Scatter Chart
Three Variables
3D Area Chart
Changing Over Time
Static
Few Periods
Many Periods
Only Relative Differences Matter
Relative and Absolute Differences Matter
Only Relative Differences Matter
Relative and Absolute Differences Matter
Simple Share of Total
Accumulation or Subtraction to Total
Components of Components
Stacked 100% Column Chart
Stacked Column Chart
Stacked 100% Area Chart
Stacked Area Chart
Pie Chart
Waterfall Chart
Stacked 100% Column Chart with Subcomponents
© 2006 A. Abela — a.v.abela@gmail.com

Notes:

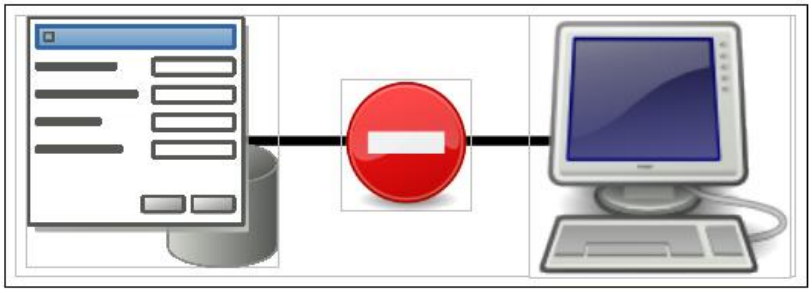
EXERCISE (REPORTS)

What Does Your Organisation Need to Know?

What Reports Does Your Organisation Require?

Notes:

DATA ACCESS CONTROL



Data security and access control is critical in any data-rich environment. Sensitive and personal information should not be available to users who do not have a need to know. Within a Business Intelligence environment, only top level directors and database administrators should have top level access to the data.

Users should have their access to the data and stored reports restricted based on the business requirements and trust level. At the simplest level, an Access Control List (ACL) will list all users with permission to access a particular dataset. Restrictions can be implemented at the database, table or column level. Fine grained restrictions allow for a lot of control, however more administration required to maintain them.

Several data security companies exist who can audit your Business Intelligence system and associated security procedures.

Notes:

ORGANISATIONAL STATUS AND DASHBOARDING

Not all reports from a Business Intelligence system are comprehensive analytical reports. A report can be as simple as a single number or percentage to represent a Key Performance Indicator (KPI) for the organisation.

These KPI's can be used to.

- Identify and correct negative trends.
- Identify new trends.
- Measure efficiencies/inefficiencies.
- Monitor the performance of individual sections.

By integrating the KPI reports from a Business Intelligence system with a digital dashboard managers can rapid and real time statistics about an organisation.

A digital dashboard (also known as an enterprise dashboard or executive dashboard), is a business management tool used to visually ascertain the status or health of a business enterprise though these KPI's. Digital dashboards use visual, at-a-glance displays of data pulled from disparate business systems to provide warnings, action notices, next steps, and summaries of business conditions.

Notes:

EXERCISE (DASHBOARD)

What would you put on the Senior Executive's Dashboard?

Notes:

SCHEDULED REPORTING

End user experience has shown that there are two primary types of reporting that are used.

- Ad-hoc reporting: This type of report is written to answer a single business requirement. Normally it does not need to be run again, and can be discarded after use.
- Scheduled Reporting: This type of report is written to meet a general business requirement. Once written is run on a scheduled basis where users can view the output. Users who do not have permission to create their own reports can often access the pre-run scheduled reports.

By automatically running scheduled reports, interaction required by the Business Intelligence administrators is minimised. It also allows complicated and processor intensive reports to be run during non-peak times.

NARROWING YOUR RESULTS

Most reports from a Business Intelligence system display aggregated information to represent an organisational fact. Such as a graph of the number of new clients over the previous year.

Some of the more advanced reporting tools have the ability to “drill down” into reports to extract more detailed information regarding each data point.

This functionality allows users to look at an organisation level of the data and see trends and statistics and to identify a data point of interest and further interrogate the information.

Example: Sales Report

An example sales trend report, displaying sales by region over time.

The manager is able to drill down into a data point identifying a region and time. This could then return a table report displaying total sales by salesperson and store.

This could then be further narrowed to see a details of each sale and customer sold to.

MASHUPS

Notes:

Mashups are defined as a hybrid application (usually web based) which combines data, presentation and functionality from two or more sources to create a new service. The concept of mashups originate in the early days of the Internet and, with the advent of Web 2.0 and organisations publishing open API's to their products (such as Google or Twitter), have become a lot more prolific in the last 3-4 years.

Combining your data with these public visualisation and functionality API's can add a level of depth and understanding of your data that would previously have been unknown. See below for some examples of popular and interesting Mashups.

If your data could be interesting to the general public, and bearing in mind privacy and security concerns, it can be beneficial to your organisation to publish the data and allow others to create their own mashups. Many of the examples above utilise proprietary information from companies such as Amazon or Google, which in turn have added value to those companies.

WEB SERVICES

A web service is a collection of protocols and standards used for exchanging data between applications or systems. Software applications written in various programming languages and running on various platforms can use web services to exchange data over computer networks.

If the Business Intelligence system is accessible by a well defined web service other internal and third party applications can be written to query the Business Intelligence system.

Case Study: A large professional organisation.

The deployment of a new corporate web portal is required to allow members to login and view technical information.

By writing a web service to query the Data Warehouse, this information was easily and quickly accessible. The alternative was to modify the primary CRM, which was a long and costly process.

This topic cannot be covered as a small subsection of this course, and is left as the responsibility of the reader to further research the concepts and potential advantages raised.

Notes:

Notes:

REFERENCES

PostgreSQL: http://www.postgresql.org

Oracle: http://oracle.com

Data Warehouse.com: http://www.datawarehouse.com/

DM Review: http://www.dmreview.com/index.cfm

Pentaho: http://www.pentaho.com/

Jasper Reports: http://jasperforge.org

YALE: http://rapid-i.com/

OpenI: http://openi.sourceforge.net/

Business Objects: http://www.businessobjects.com

Netezza: http://www.netezza.com/

SAP BI Warehouse: http://www.sap.com

Teradata: http://www.teradata.com/

Notes: