Data Warehouse

1. Defining a Data Warehouse

A data warehouse is literally that a fairly large warehouse but one filled with data

* Data warehouse not the same with database

A Data warehouse is typically built on top of some type of a database. So you can thing data warehouse as the usage (cách sử dụng), and the database as the platform (nền tảng)

Data comes from elsewhere (other words, our operational systems and sometimes also external sources), we don't create data for the first time in a data warehouse as part of some type of a transaction

Those transactions occur and are recorded in various operational systems and their data is then subsequently (về sau) sent down to the data warehouse

Possibly dozens of data sources

Many data warehouses have dozens of data sources and you can also assume that there's a linear relationship in play here

The more sources, the more complex the overall environments

Data is copied ... not moved

Data remains (còn) in our source systems and the copies are made and send into data warehouse

Rule that goven (chi phối) how we built our data warehouses and how whe organize and store our data

1. Data warehouse is an integrated environment

2. Data warehouse should be subject-oriented (định hướng theo chủ thể) (bất kể)
regardless of how many systems and which data come from which systems, we need to reorganize the data by subjects (chủ đề)

3. Time variant

Data warehouse contain historical data, not just current data

4. Non volatile (Tính e biến động)

Traditionally, we will periodically (định kỳ) load data into a data ware house Think of it as refreshing a data warehouse to keep it current and we do so in batches (nhóm , lô)

Batwoen the time of the last refresh and the next one the data warehouse stays as is , even if thousands, or perhap millions of transactions are occurring in our transactional systems . That is non-volatile

=> Data warehouse ramain stable is between refreshes

So we can the do things like stategic planning without the data chaing underneath us

We bring data to data ware house

=> We'll typically restructure and reorganize it to make it more useful for analysis

5. Why ?

To support data - driven (hướng dữ liệu) decision making

Reasons for you to built a Data Warehouse

1. Making data - driven decisions (quyết định theo hướng dữ liệu)

(trực giác)

Rather than having rely on solely on experience and intuition and even hunches (linh cảm)

2. One stop shopping (1 cửa hàng mua sắm)

In other words, the data that we need is all in a single location, rather than scattered (rải rác) among the transactional and operational applications where we get that data from

Making data driven decision

+ Past
+ Present
+ Future
+ Unknow

Compare Data Warehouse and Data lake

1. Data warehouse is often built on top of relational database

Some time data warehouse built on top of a multi dimentional database that's typically know as cube

2. Data lake built on top of some soft big data environment rather than a traditional relation data base

Diffrences

1. Volume

Big data and it usage in datalakes help us manage extremly large volume of data larger than we typically would include even the largest of data warehouses

Volume is one of the traditional three V's of big data

2. Velocity

(nhanh) (tiếp nhận)

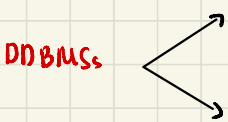
Big data also supports very rapid intake of new and change data much more rapidly than we typically do with traditional data warehousing

Variety (đa dạng)

Big data easily supports along with semi-stuctured data such as text messages, email, blogs and complex documents, as well as unstuctured data such as audio and video, ...

Compare data ware house to Data Virtualization

The root of data ware housing



Data virtualization (ảo hoá)

Data warehousing

Data virtualization

+ Can be thought of as a read-only distributed database
+ In-place data access. We access it from its original locations at the time we need to do so for reports and analytics
+ Many name over the years

Data virtualization use case

1. Simple transformations

If we have data that requires simple tranformations or perhaps even no transformations to use in BI and analytics

2. Small number of data source

3. Relaxed response time

Look at a simple end-to-end Data Warehousing Environment
(từ đầu → cuối)

1. A typical data warehousing environment

Data sources $\xrightarrow{\text{ETL}}$ Data warehouse

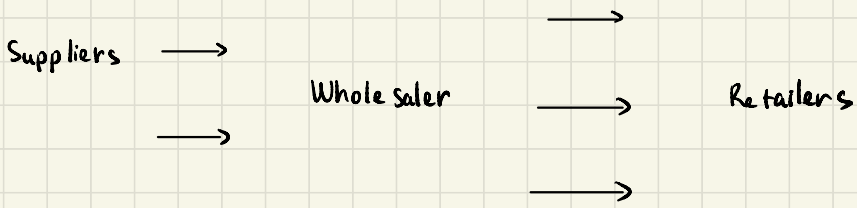
ETL

* Extract
* Tranform
* Load

2. Adding Complexity

Data source $\xrightarrow{\text{ETL}}$ Data Warehouse $\xrightarrow{\text{ETL}}$ Data marts

Ex:



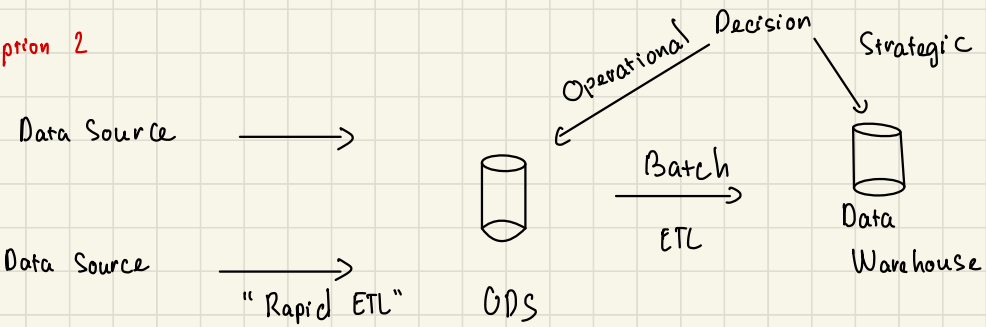
Built a centralized Data warehouse

1. Is a single data warehouse environment
2. With centralize data ware house you have a single database. Everything from all of your sources will feed into that single database
3. Support one-stop shopping

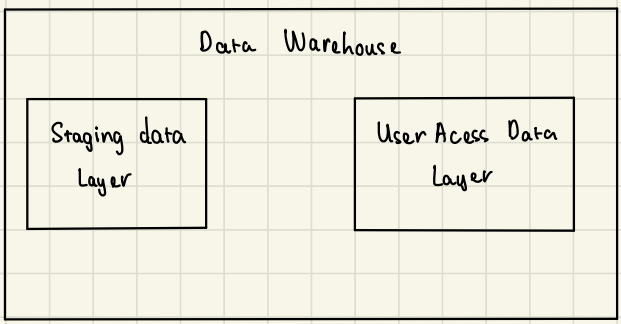
Historical challenges

1. Technology: face with large data volume
2. Work processes :
3. Organizational and human factors

Option 2



Role of staging Layer



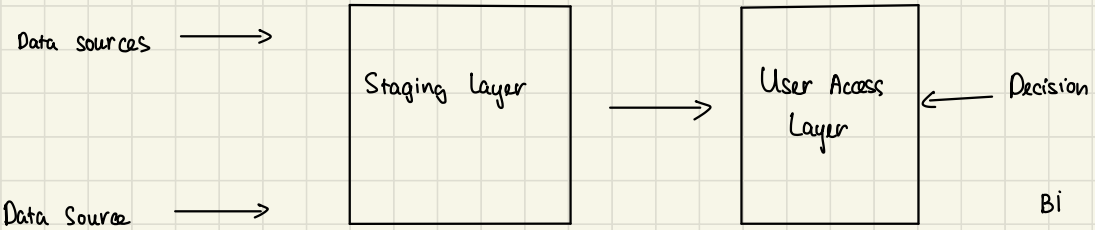
Staging Layer

- "Landing Zone"
- "E" within ETL
- 2 variations

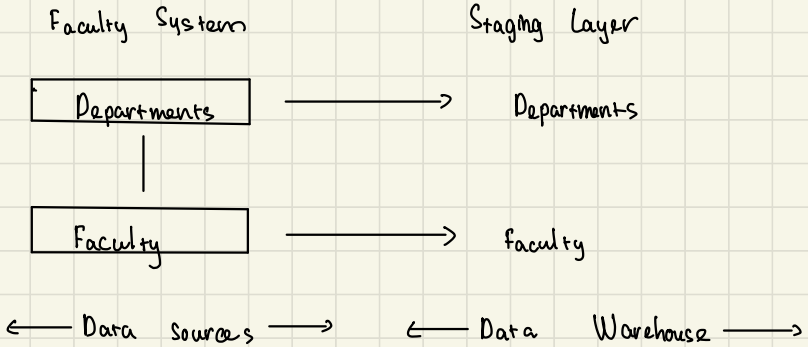
User access layer

- Where users go
- Dimensional data

Expanding our data warehousing architecture



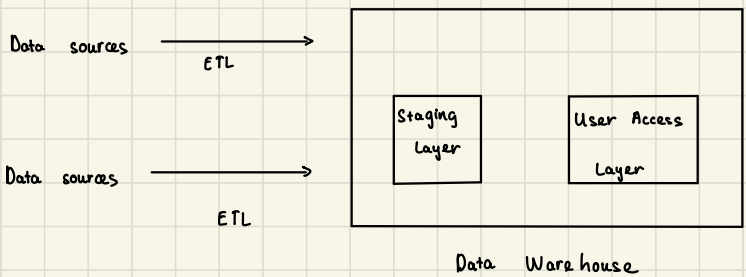
Inside staging area



Compare ETL and ELT

Extract

- Quickly pull data from source applications
- Traditionally done in "batches" (lô <=> gom nhóm dữ liệu)
- Raw data ... errors and all
- Land in data warehourse staging layer



Transform

- "Apples to apples"
- Prepare for uniform data in user access layer
- Can be very complex

Load

- Final stop along the data path way
- Store uniform data in user access data

Challenges with traditional ETL

- Significant business analysis before storing data
- Significant data modeling before storing data

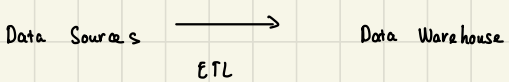
Change the order

ELT

- "Blast" data into big data environment
- Raw form in Hadoop HDFS, AWS S3, etc ...
- Use big data environment computing power to tranform when needed
- "Schema on read" vs. "Schema on write"

Initial Load ETL

A typical data warehousing environment



Two difference of ETL

- Initial (ban đầu)
- Incremental (tăng dần)

Initial ETL (ETL ban đầu)

- Normally one time only
- Right before the data warehouse goes live
- All relevant data necessary for BI and analytics
- Redo if data warehouse "blows up"

+ Mục tiêu : thực hiện lần tải đầu tiên của toàn bộ dữ liệu của hệ thống nguồn và hệ thống đích

Incremental ETL (ETL gia tăng)

- Incrementally "refreshes" the data warehouse
- New data : employees, customers, products, ...
- Modified data : employ promotions, product price change, ...
- Sepecial handling for deleted data

+ Purpose : Bring the data warehouse up to date

4 major incramental ETL patterns

- Append
- In-place update
- Complete replacement
- Rolling append

ETL today

- Append
- In-place update

Making data driven decisions

- One or more measurements
- Dimensional context for each measurement

Dimensional context : "by" vs "for"

Wording	Usage
By	"Sliced and grouped" by values of the entire dimension
For	One or more specific values from within the entire dimension

Non - additive facts

- Store underlying components in fact tables
- Possibly store non-additive fact also for individual row easy access (minimal calculations)
- Calculate aggregate averages, ratios, percentages, etc ... from totals of underlying components

Semi - additive facts

- Sometimes you can add these facts
- But other time you can't add them
- Typically used in periodic snapshot fact table

---

1. Fact là gì

Fact chứa dữ liệu chính, thường là các số liệu hay chỉ số định lượng có thể đo lường được. Fact chứa các thông tin được tổng hợp từ các giao dịch

- Chứa các số liệu có thể đo lường
- Chứa các khoá ngoại

2 Dimension shema là gì ?

- Chứa dữ liệu mô tả của thuộc tính
- Kết nối với bảng fact thông qua khoá ngoại

Như vậy bảng fact ghi nhận sự kiện là giao dịch , còn bảng dimension giúp cung cấp thông tin về khách hàng , sản phẩm , thời gian và địa điểm liên quan đến giao dịch đó

Star schema vs snow flake schema

Star schema : 1 bảng fact ở trung tâm và các bảng dimenson toả ra xung quanh

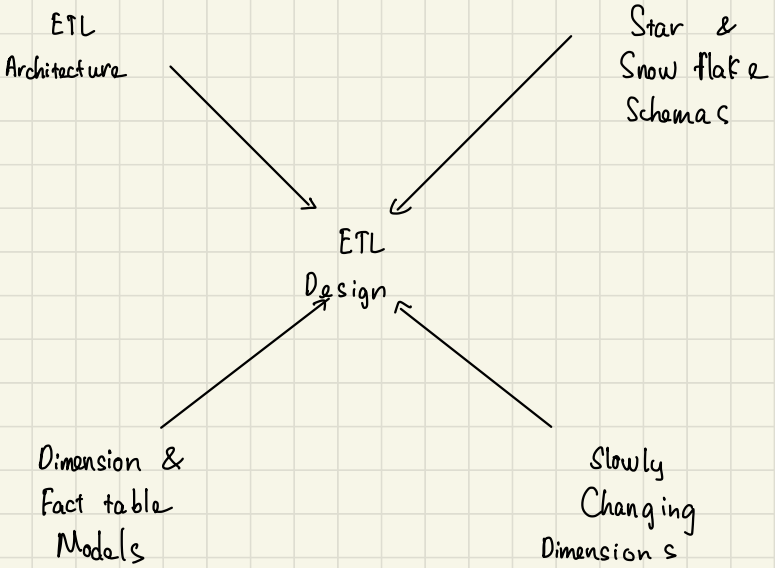
Snow flake schema : Là biến thể của star schema trong đó các bảng dimencion được chuẩn hoá hơn để tránh trùng lặp dữ liệu

2 Khác nhau

Slowly Changing Dimensions (SCDs) and Data Warehouse history

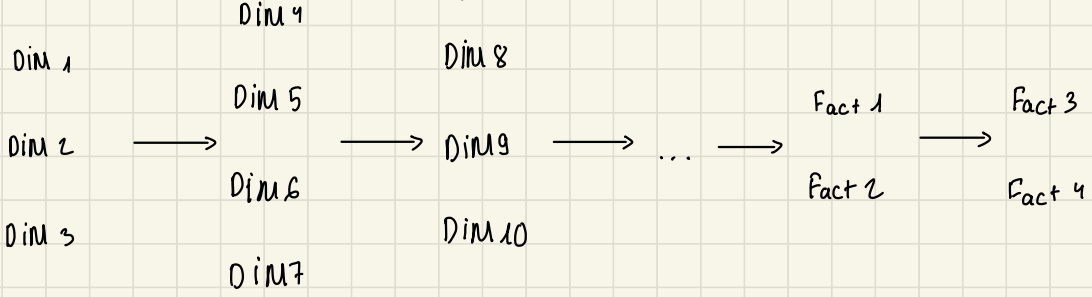
1. Slowly Changing Dimensions (SCDs)

- Techniques to manage history within data warehouse



ETL best practices and guidelines

- Limit amount of incoming data to be processed
- Process dimension tables before fact table
- Opportunities for parallel processing



DIMENSION TABLE Incremental ETL

Step 1 : data preparation

All possible operational data from all source → "Change Data Capture" → New and modified data for DW

"Change Data Capture" techniques

- Transactional data time stamps
- Database logs
- Last resort : data base scan - and - compare ,

Step 2 : data tranformation

Step 3 : process new dimension rows

Step 4 : process SCD type 1 changes

Step 5 : process SCD type 2 changes

More accurately

For each row :

If new : add to DIM table

If not new : process any Type 1 and Type 2 changes