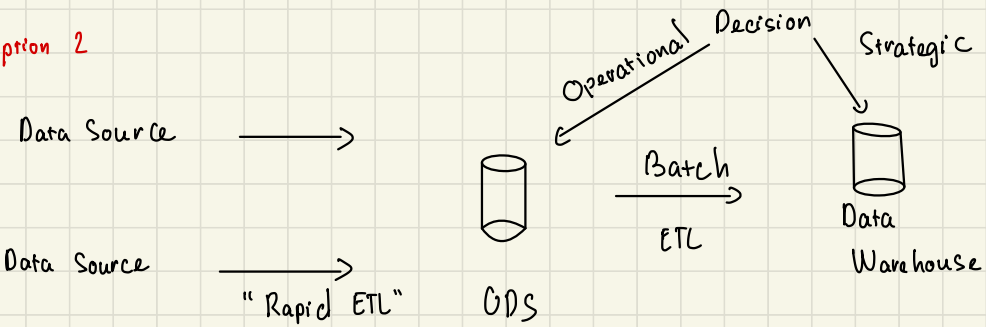
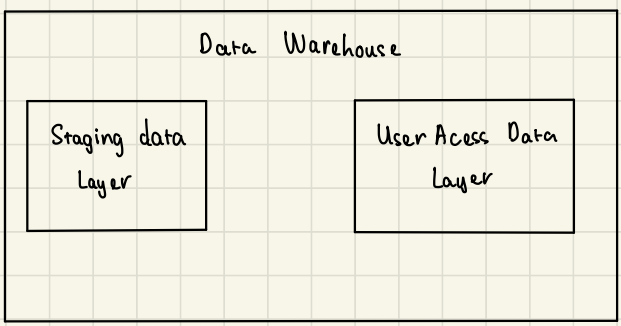
Option 2



Role of staging Layer



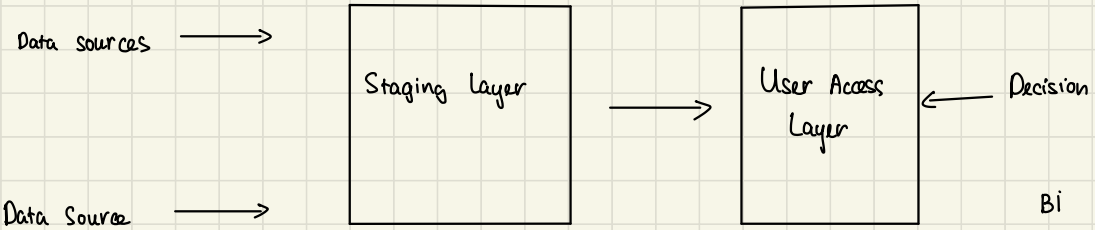
Staging Layer

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

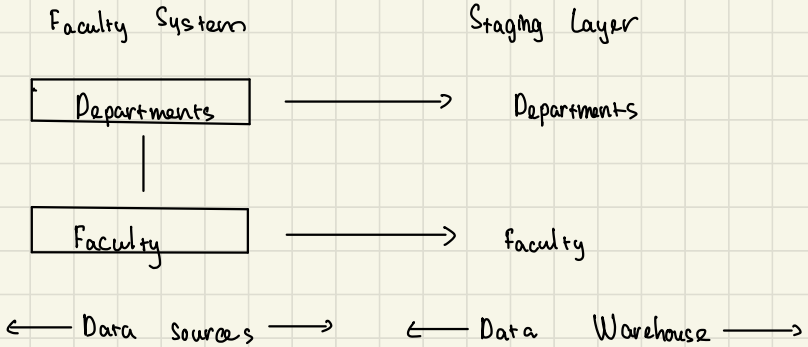
User access layer

- Where users go
- Dimensional data

Expanding our data warehousing architecture



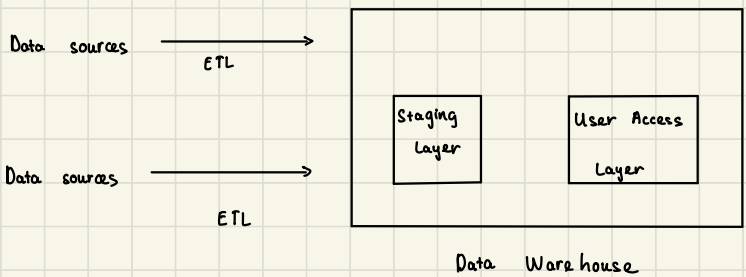
Inside staging area



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

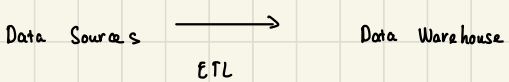
Change the order

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

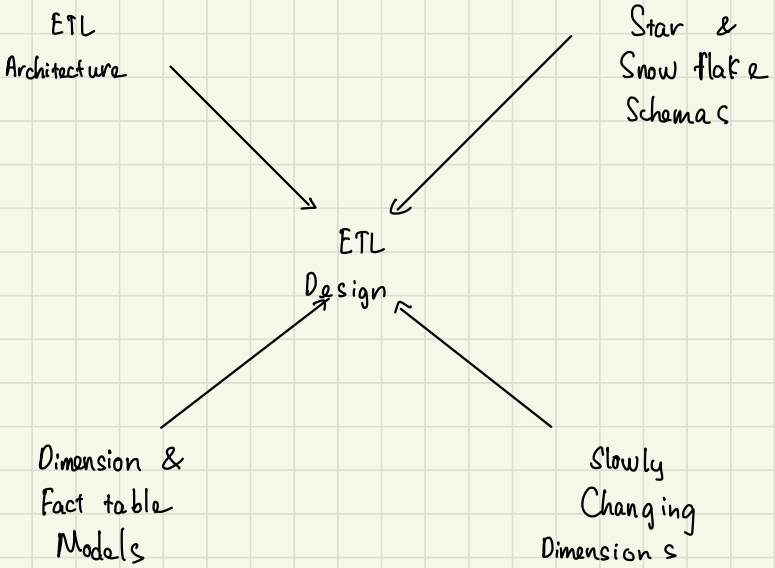
Snow flake schema : Là biến thể của star schema trong đó các bảng dimencion được chuẩn hoá hơn để tránh trùng lặp dữ liệu

2 Khác nhau

Slowly Changing Dimensions (SCDs) and Data Warehouse history

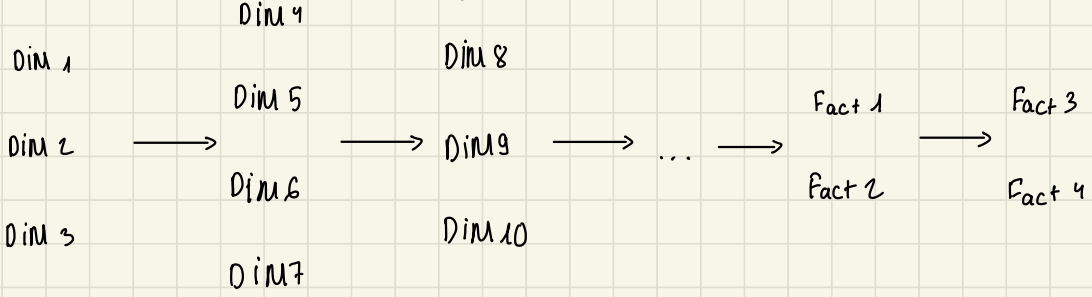
1. Slowly Changing Dimensions (SCDs)

- 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