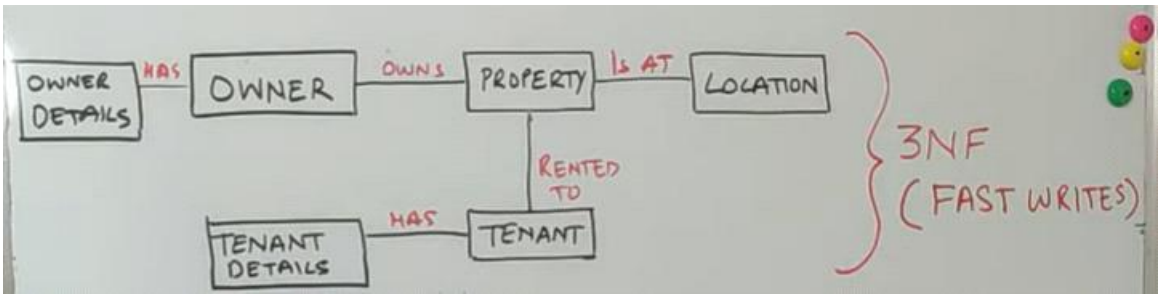
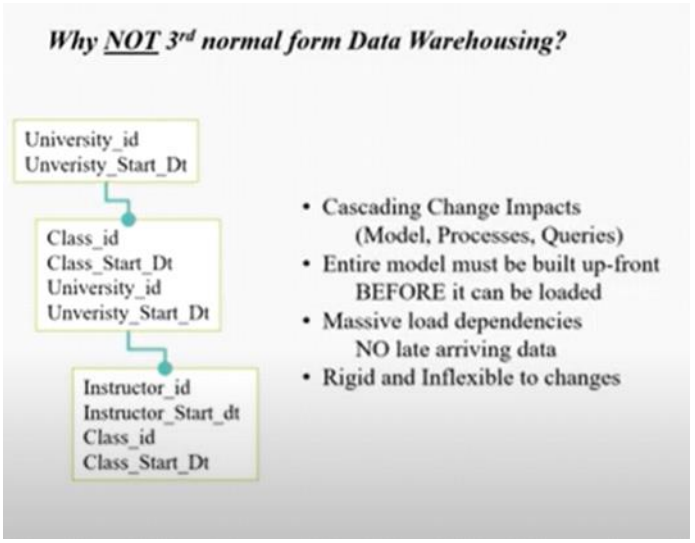
# Data Model

14 September 2022 10:14

## 3NF



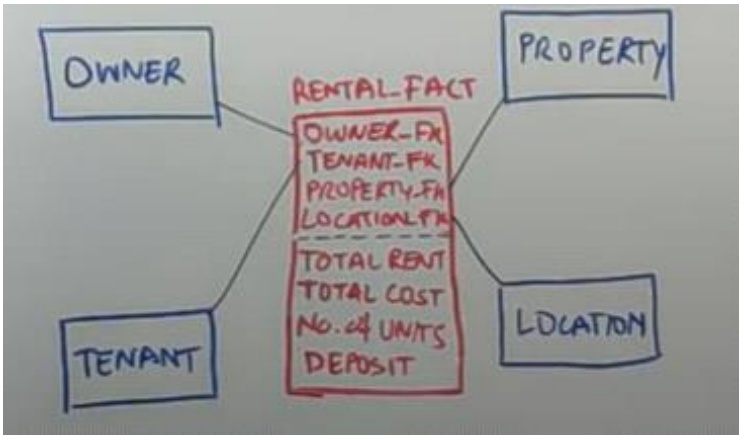

Pros:

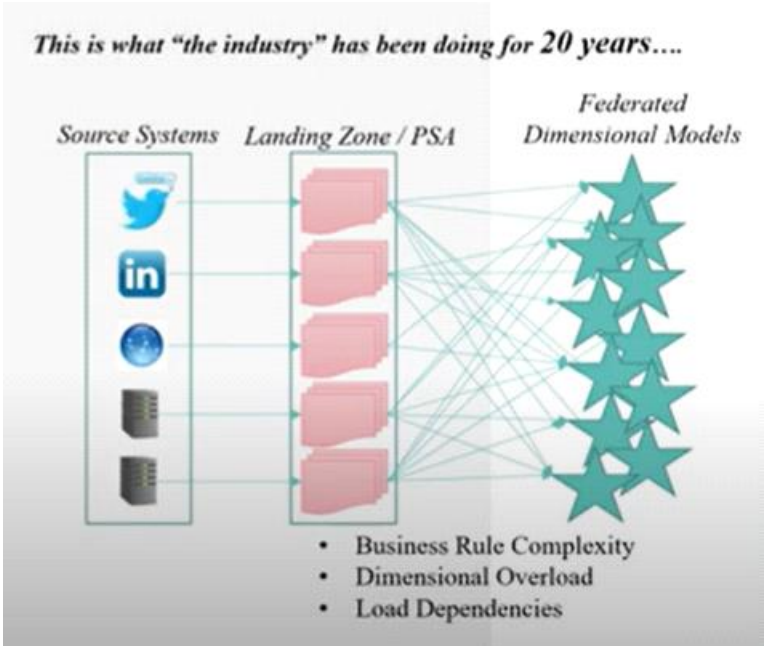
- Normalised
- Faster Writes (Good for transactions, not good for data overtime)
- You can drill down to data in different granularity (Grain is the lowest level up until which data is stored in a Fact table in context with a specific Dimension)

Cons:

- Any changes in the parent table (i.e. primary key) has a cascading effect on the child table
- Entire model must be built upfront BEFORE it can be loaded
- Massive load dependencies (NO late arriving data / real-time data / streams)
- Rigid and inflexible to changes

## Dimensional (Star Schema)





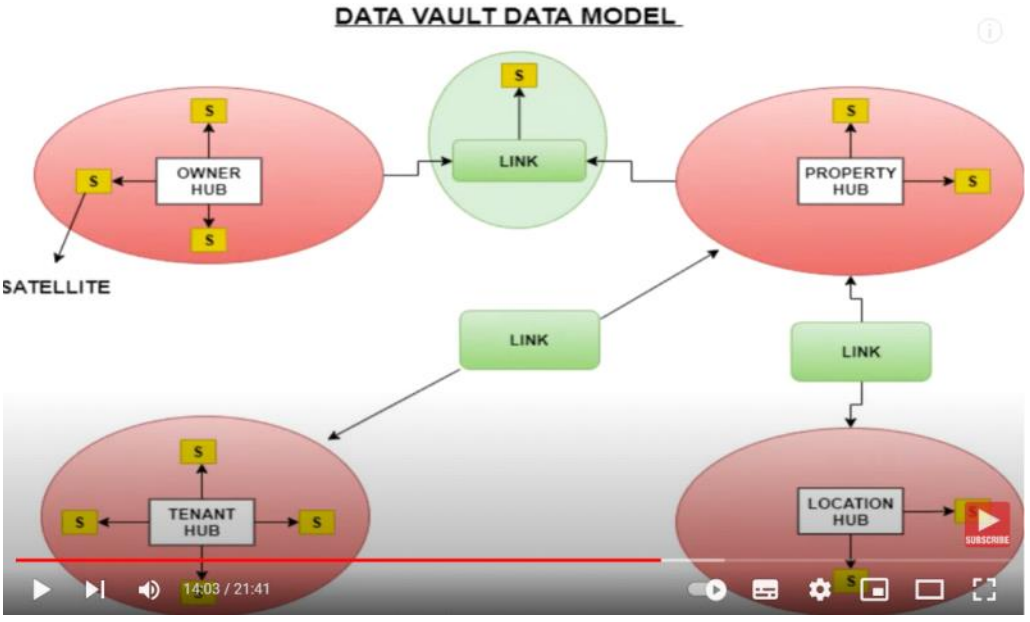

Pros:

- Fact and Dimensional Table
- Only one layer of normalisation (less joins)
- Faster Reads

Cons:

- **Hard to adjust to change in business requirements** (i.e. city level -> postcode level). When grain in fact table is reloaded, there is a risk of losing history. If a new fact table (i.e. postcode level) is added, you have to maintain multiple fact tables
- **Not scalable** (Complex Patterns = High rate of Production Failures and Slow load times over Big volumes)

## Data Vault




**Hub**: Business Keys (i.e. Employee ID)
**Link**: Relationships between hubs
**Satellite:** Business context of hubs (i.e. employee name, DOB, salary) . It is where all the descriptive (i.e. non-key) columns reside. Also Change Data Capture (DCD) is done and history is restored.

The Primary Key for a Satellite has 2 sparts:

1. The PK from its Parent Hub (or Link)

2. The LOAD_DTS

Pros:

- New source systems can be easily integrated into existing Enterprise Datawarehouse (EDW) making it agile (i.e. adding new Satellite)

Cons:

- Data Vault is not meant for Business Intelligence (BI) reporting and analysis, for that we need to derive Data Marts from our EDW built on Data Vault
- May not be suitable for systems with few sources and static data as setup overheads may override business value

Challenge:

- When you have a model that come from multiple sources,

Versioning of Data Model: To be in sync with the application deployment

Logs -> transactions

Cumulative dimensions

Previous dimensions -> users (is_loyal? Array of ) one row per user per day, array of sum of transactional amount, average / 30 > 5

Dimension

Daily growth:

One day -> # loyal customers / total customers

Looking at month-to-month (month end) -> changes in % of loyal customers

Is_loyal

Whether more discounts will get you more loyal customers?

Dimension history: transaction
Dimension latest: customer

Dimension cumulated: customer, array(transcaiton amount)