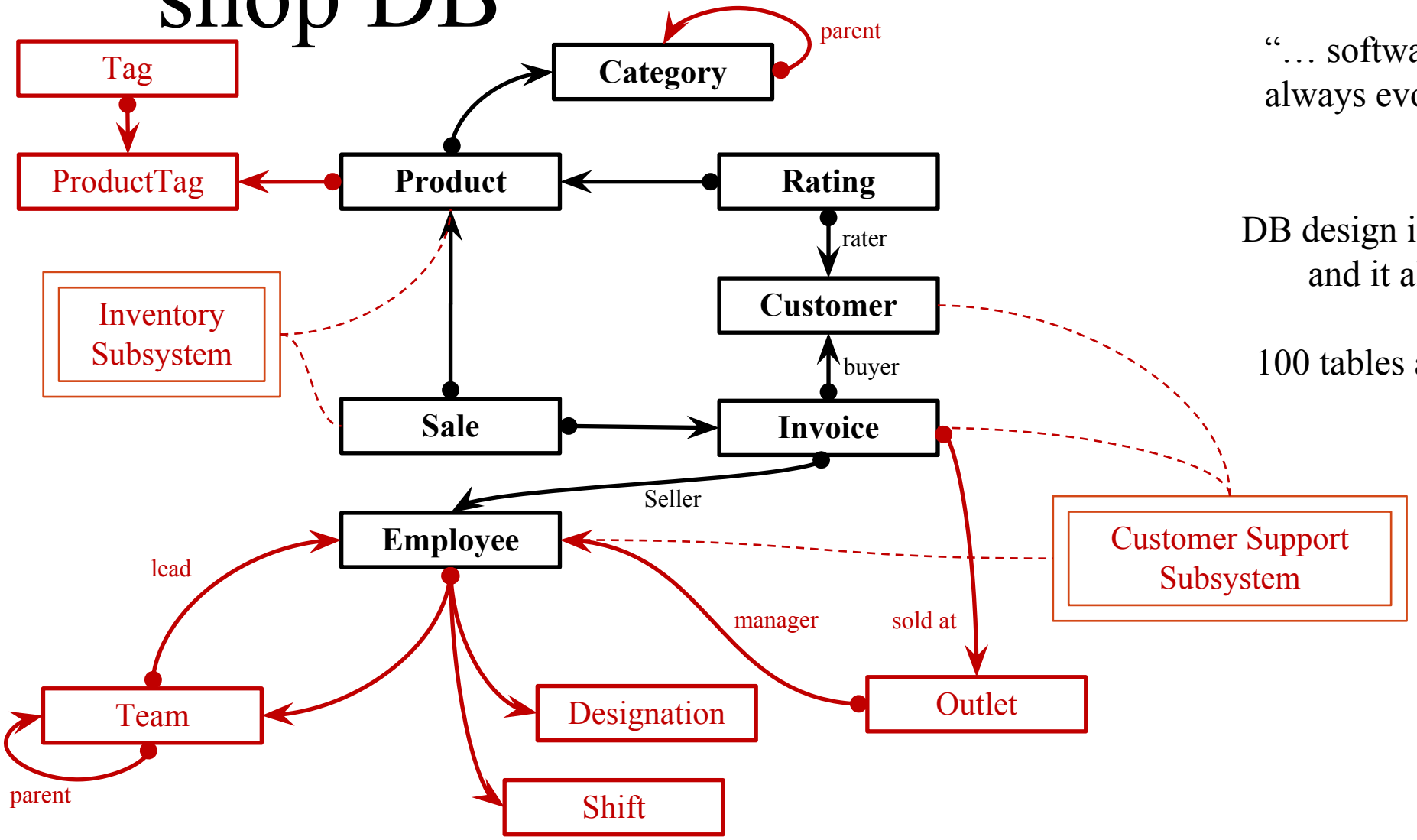
Reporting Database

Software Design and Architecture (SWE 4601)

Title Image Source: https://martinfowler.com/bliki/ReportingDatabase.html

More on the kid's shop DB



"…software designs tend to be incredibly large and complex."

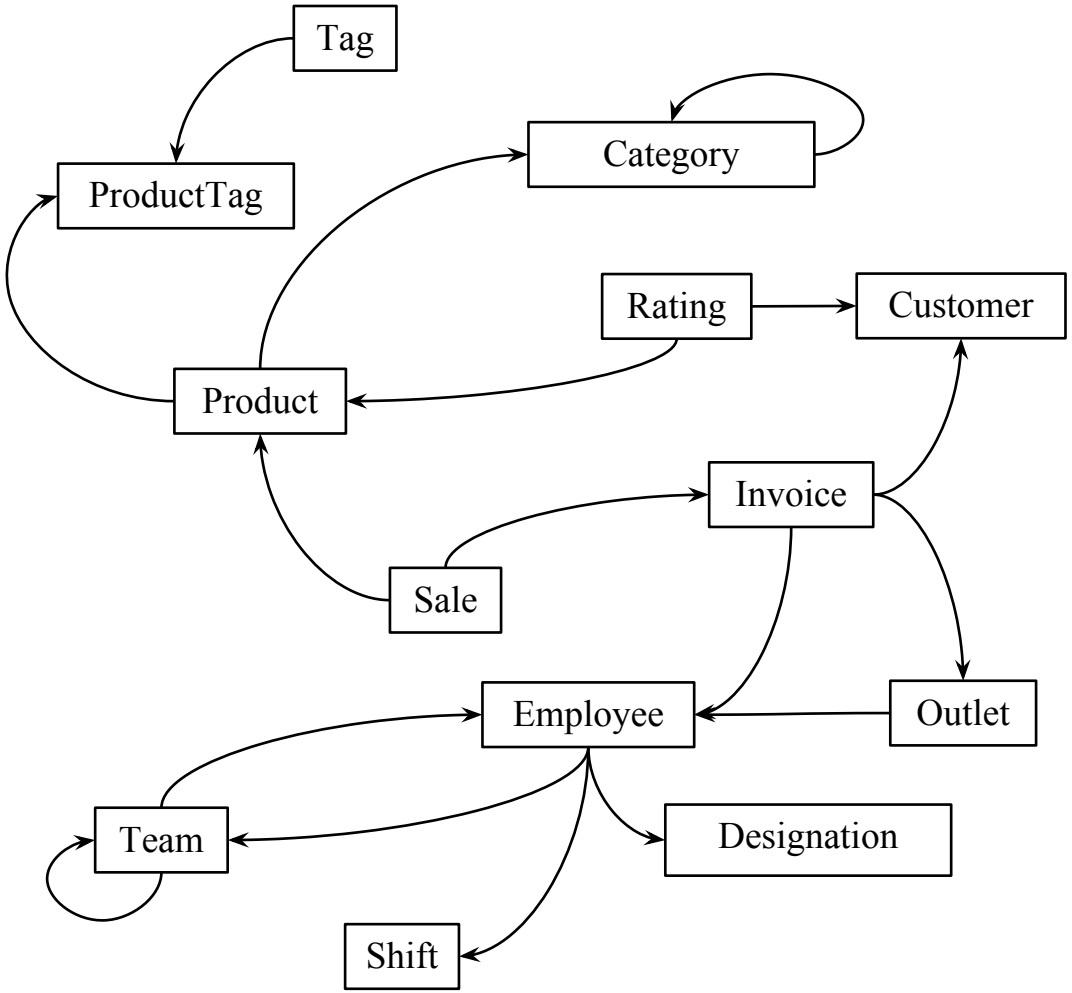
"… software designs are almost always evolving."

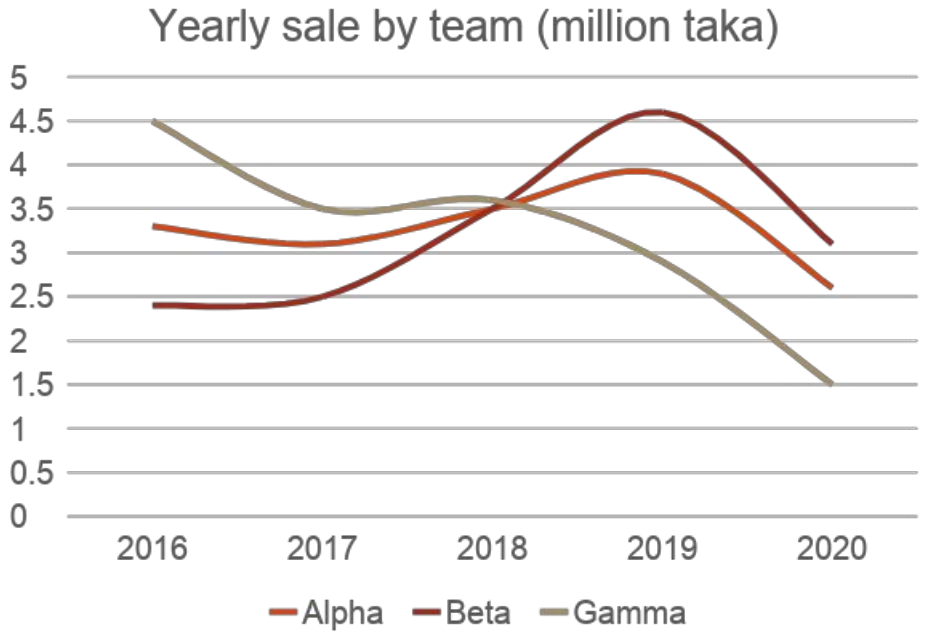
- Jack Reeves

DB design is part of the software design, and it also grows rather rapidly.

100 tables are many, but not too many.

Reporting queries





Managers want to know

- Which categories are popular in which outlets?
- Which shift can be served with fewer employees?
- Are some employees more productive than others?
- Which categories are no longer selling well?
- Which salespersons are good at selling not-so-popular products?
- Which product is not selling well but are filling the storage – run a promotion?

We want to keep normalization for consistent and fast update operation, we also want fast query

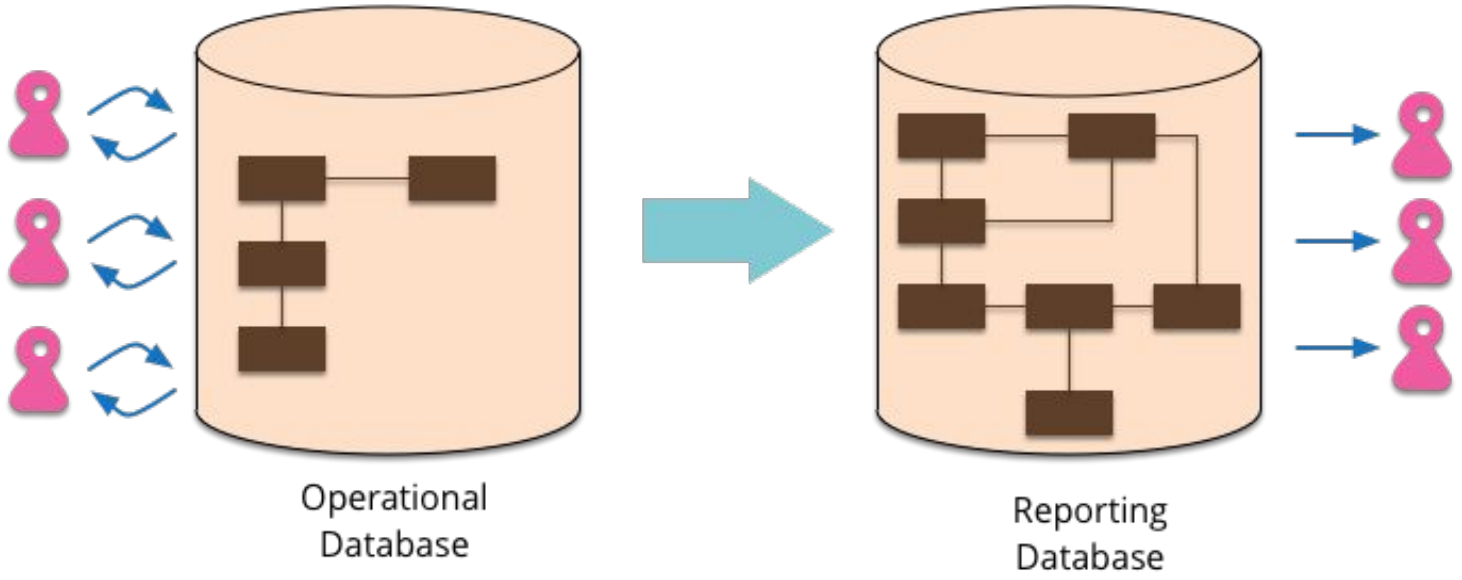
Normalization and fast query does not just work together

Attribute redundancy helps, but gets complicated over time

How to solve this problem?

Why not keep copies of normalized data in denormalized table(s)?

Introducing Reporting Database



Use separate DB for operations and reports
Build the report DB from Operational DB

https://martinfowler.com/bliki/ReportingDatabase.html

Benefits of Reporting Database

- Special design for reporting is possible
- It is read-only database, normalization is not necessary
- Refactoring operational database without needing to change the reporting database.
- If different DB instance is used
  - Load of query and update are separated
  - Horizontal scaling of reporting database is easy

A case study at Streams Tech

Situation

- 50+ tables
- Moderate data volume
- Very dynamic reporting, requiring almost all tables to be joined
- Some joins yield millions of records
- Result: slow/nonresponding reports

Solution

- Use a reporting table (in same database)
- Keep track of data updates
- Schedule data synchronization
- Result: smooth reports
- Schema: star

How to sync reporting database

Option 1: Scheduled update

- Daily, hourly or may be every 10 minutes
- Pro: Easy to manage
- Con: backdated data

Option 2: Messaging

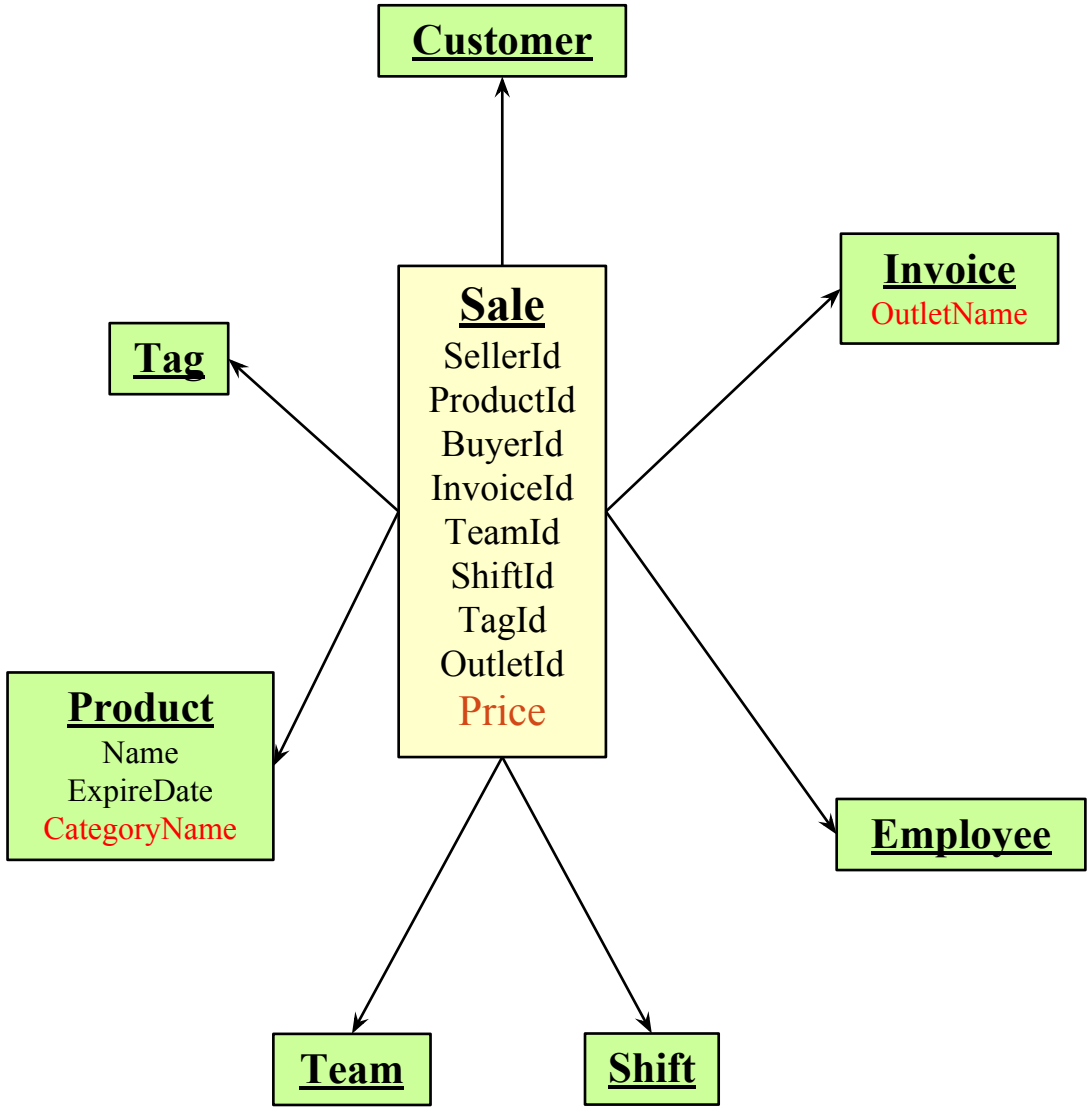
- Send message to reporting DB when there is an update in operation DB
- Pro: immediate update
- Con: difficult to manage

Reporting database terminologies

- Measure/fact: the data that we are interested about.
  - Example: selling price or sale.
- Fact table: the table that contains the fact/measure.
  - Example: the sale table
- Dimension: categories of fact.
  - Example: product
- Dimension table: the table that contains the dimension.
  - Example: product table

Star Schema

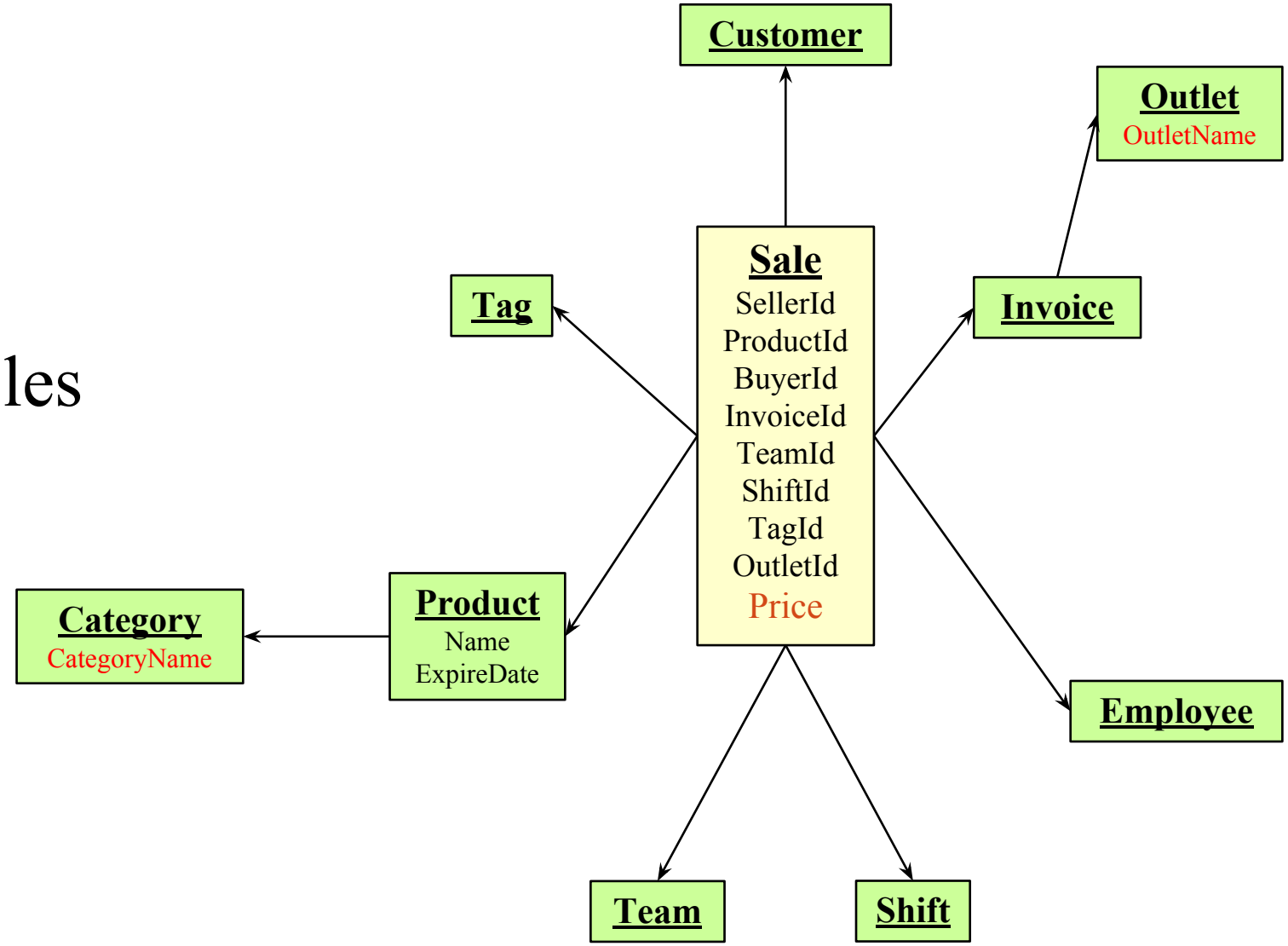
- Dimension tables surrounding a fact table
- One table per dimension
- Dimension table contains the set of attributes.
- The dimension table is joined to the fact table
- The dimension table are not joined to each other
- Fact table contain key and measure
- The dimension tables may not be normalized



https://www.guru99.com/star-snowflake-data-warehousing.html

Snowflake Schema

Normalized dimension tables



Star vs Snowflake

	Star Schema	Snowflake Schema
Normalization	Pure denormalized dimension tables	Have normalized dimension tables
Maintenance	More redundancy due to denormalized format so more maintenance required	Less redundancy so less maintenance
Query	Simple queries due to pure denormalized design	Complex queries due to normalized dimension tables
Joins	Less joins	More joins
Usage Guidelines	More than data integrity speed and performance is concern here	Concerned about data integrity and duplication