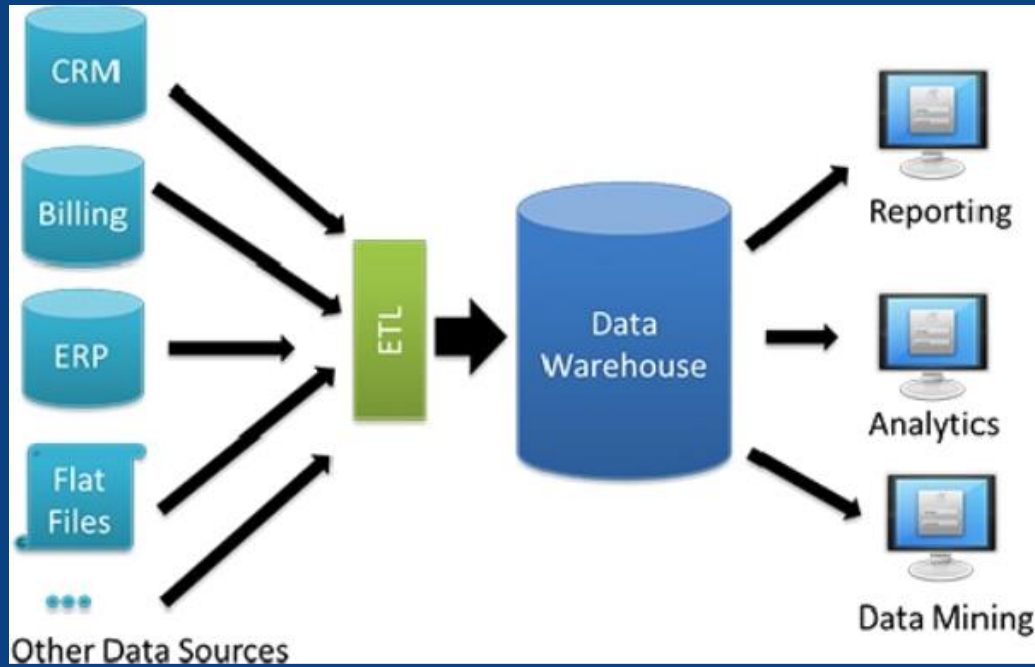
THE UNIVERSITY OF MELBOURNE

Data Warehousing

Database Systems & Information Modelling
INFO90002



Week 9 – DW
Dr Tanya Linden
Dr Renata Borovica-Gajic
David Eccles

This Lecture Discusses

The differences between **transactional** (operational) and **informational** (dimensional) databases

Characteristics of a Data Warehouse

Overall architecture of a Data Warehouse

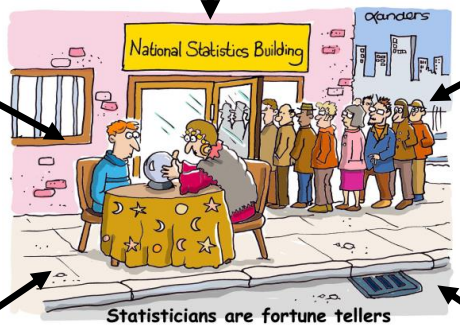
Star Schemas

Motivations: A manager wants to know….

Which are our lowest/highest margin customers ?

Who are my customers and what products are they buying?

What is the most effective distribution channel?



What product prom-otions have the biggest impact on revenue?

Which customers are most likely to go to the competition ?

What impact will new products/services have on revenue and margins?

BUT HOW? predict

Relational Databases for Operational Processing

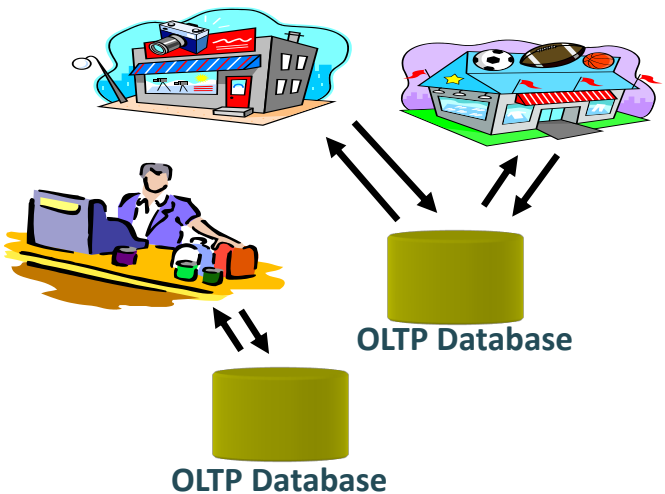
we use relational database for operational processing

Used to run day to day business operations

Automation of routine business processes

- Accounting
- Inventory
- Purchasing
- Sales

Created huge efficiencies



OLTP Databases

OLTP = "OnLine Transaction Processing"

1. Transaction processing supports daily (routine, repetitive) operations
   - Mundane but crucial
   - Become even more important with the growth of the internet

2. Definition:
   - 2-1 Collection of read/write operations
   - 2-2 Processed as one unit
   - 2-3 Reliably and efficiently processed
   - 2-4 No data loss due to interference and failures (operating system, program, disk, …)

OLTP is a collection of read and write operations.

OLTP Data Characteristics

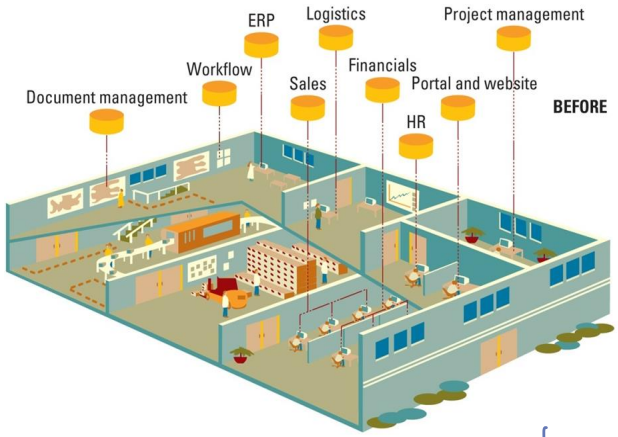
Characteristics of data:

- Transaction oriented
  - DML
  - **Inserts/Updates/Deletes**
- May be inconsistent and incomplete
  - Data may not be in its final form
- Volatile – continually changing
  - Data maybe subject to change
- Current
  - Data related to the operation of the business TODAY!

- Different entry standards and formats
  mm/dd/yyyy vs dd/mm/yyyy
- Missing / inaccurate / incomplete
  Online customers accidentally or intentionally enter inaccurate details

companies based on their needs might using different types of database.

Databases are great, BUT …



Too many of them

- Everybody wanted one, or two, or more
- Production, Marketing, Sales, Accounting …

Everybody got what was best for them

- IBM, Oracle, Access, Microsoft

Eventually this re-created the problem databases were meant to solve

1. Duplicated data → the same person could be in different database, when person update the data (ex: change surname), the data should be propagated across all different database.
2. Inaccessible data
3. Inconsistent data

But data is useful for analysis and decision making

some company are still use multiple system, because it's more interopeable now.

→ And each company need single point of truth, a way to aggregate

all data from entire organisations. from CRM. CSM from every system they're using.

What can be done about it? SPOT!

CRM: customer relationship management

SCM: supply chain management

Need an integrated way of getting the ENTIRE organisational data

Need an Informational Database, rather than a Transactional Database

- A single database that allows *all* of the organisations' data to be stored in a form that can be used to support **organisational decision processes**

So, information database is a centralised repository for decision making.

↳ It's about future.

A centralised repository for decision making

- Populated from operational databases and external data sources
- Integrated and transformed data
- Optimised for reporting

And it's optimised for reporting, this mean we don't need every individual transaction. We are happy to take transactions about this product for on daily or weekly base. We don't need every minutes what happened with this product.

Single Point of Truth (SPOT) about the data

transaction database that integrity is also important. Database need to be normalized.
And consistency is also important.

OLTP Versus OLAP Systems

OLTP = "OnLine Transaction Processing" (Transactional, Operational)

- Maintaining data integrity and effectiveness while dealing with numerous transactions simultaneously

OLAP = Online analytical processing (Informational, Dimensional)

- Purpose-built, optimised for handling data analysis queries
- Deals with fewer, but complex queries
- SELECT and aggregate queries

analytical database are different, it's read only. There is not writing data, because it's historical data, we shouldn't change it. So, the query are much complex. And quite often it's combination of selection and aggregation because we want to know total accross certain period of time. to know like is the product popular or not.

OLTP and OLAP complement each other.

So one data cannot exist without

another. because analytical database comes from transaction database.

Data Warehouse: An Informational Database

data warehouse is an anayli tical database. it's a single repository of historic organisational data.

Data Warehouse:

1. • A single repository of historic organisational data
2. • Integrates data from multiple sources
   – Extracts data from source systems, *transforms*, loads into the warehouse
3. • Makes data available to managers/users
4. • Supports analysis and decision-making
5. • Read only
6. Involve a large data store (often several Terabytes, Petabytes of data)

↳ start from the company collecting data, nothing is deleted.

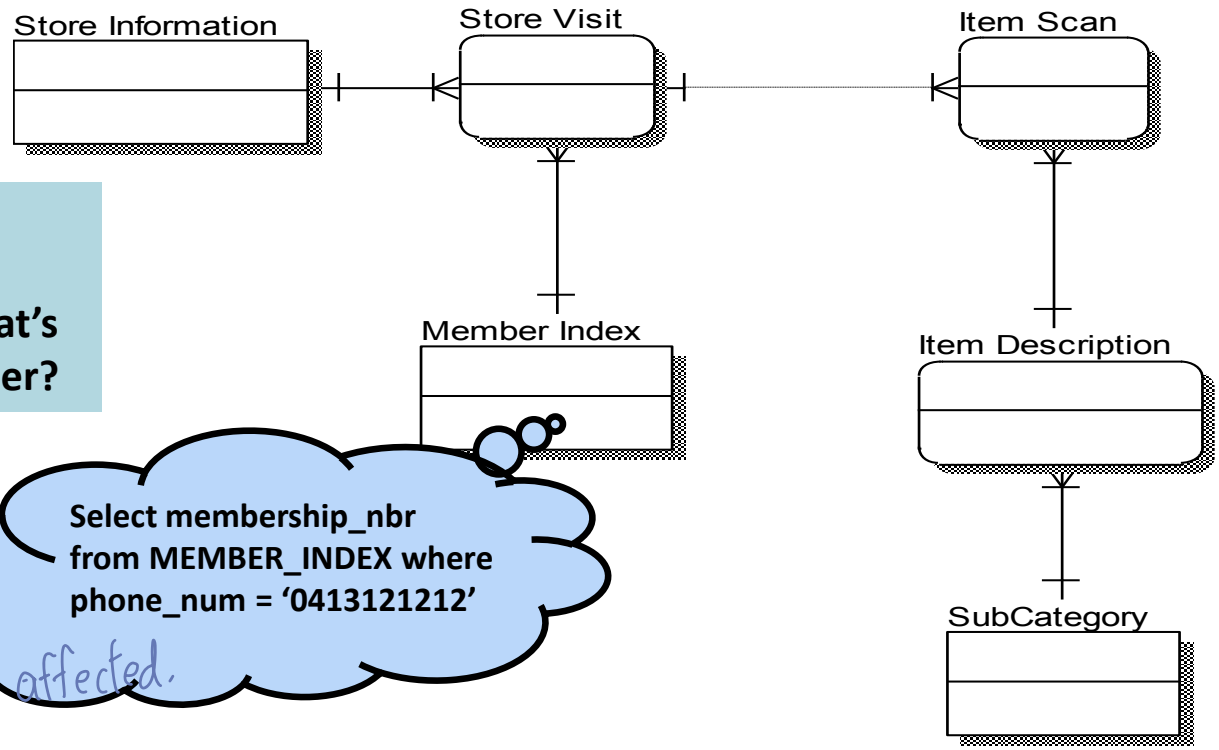
the data might come from multiple sources, and grouped by.

If it's a global appear, different level of granularity will be available.

Difference between Transactional and Informational Systems

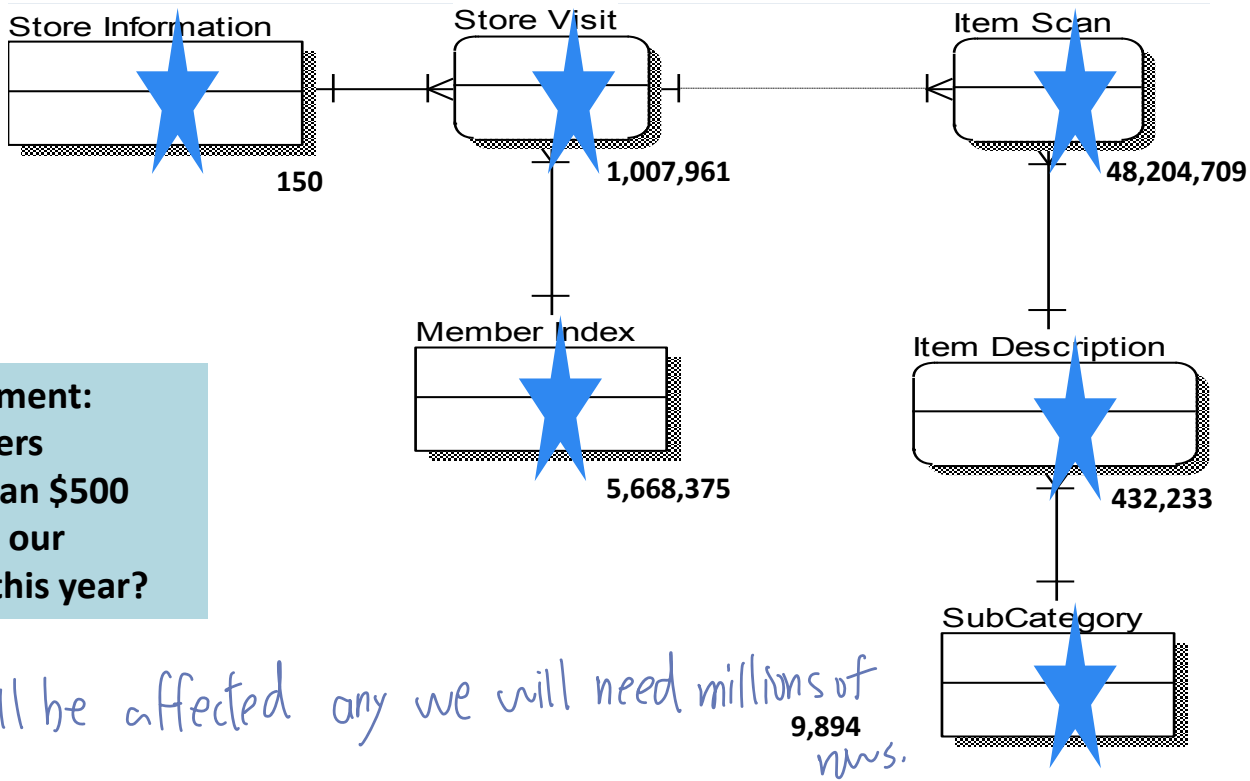
Characteristic	Transactional	Informational
Primary Purpose	Run the day-to-day business	Support decision making
Type of Data	Current data – representing the state of the business	Historical data – snapshots and predictions
Primary Users	Supports thousands/millions of users Customers, clerks and other employees	Supports hundreds of users Managers, analysts
Scope of Usage	Narrow, planned, fixed interfaces Fast and effective query processing and ensuring data integrity in multi-access environments	Broad, ad hoc, complex interfaces Emphasis on the response time to executing complex queries on large amounts of historical data aggregated from many rows
Design Goal	Performance and availability Data is stored in 3NF	Flexible use and data accessibility Data is denormalized to improve query performance
Volume	Many constant updates and queries on a few tables or rows	Periodic batch updates, complex querying on multiple or all rows

Transactional (Operational) Questions



How many tables affected? 1
How many rows have to be accessed? 1 (with an index in place)

Analytical Questions



Campaign Management: How many customers purchased more than $500 worth of alcohol in our Melbourne stores this year?

many tables will be affected any we will need millions of rows.

How many tables affected? 6
How many rows? millions

DW Supports Analytical Queries

A manager may be interested in numerical *aggregations*

- How **many**?
- What is the **average**?
- What is the **total cost**?

A manager may be interested in understanding *dimensions*

- Sales **by state by customer type**
- Sales **by product by store by quarter**

And there always have time dimensions.

Data Warehouse will help answer these questions

when we look into dimensions and numerical aggregations

They definitely have validated and integrated data. They move anything that is not complete.
↳ They deal in some ways will incomplete data.

Characteristics of a DW

Subject oriented

- Data warehouses are organised around particular subjects (sales, customers, products)

Validated, Integrated data

- Data from different systems converted to a common format: allows comparison and consolidation of data from different sources
- Data from various sources validated before storing it in a data warehouse

Problems

Incomplete Errors

- Missing Fields
- Records or Fields that, by design, are not recorded, e.g. the type of people that buy Big Issue from Big Issue Vendors when a sale is made

Incorrect Errors

- Wrong data entered into source system
  - E.g. manual entering of data will always have a percentage of incorrect data → human error

Characteristics of a DW

- Time variant
- Historical data
- Trend analysis crucial for decision support: requires historical data
- Data consists of a series of "snapshots" which are time stamped

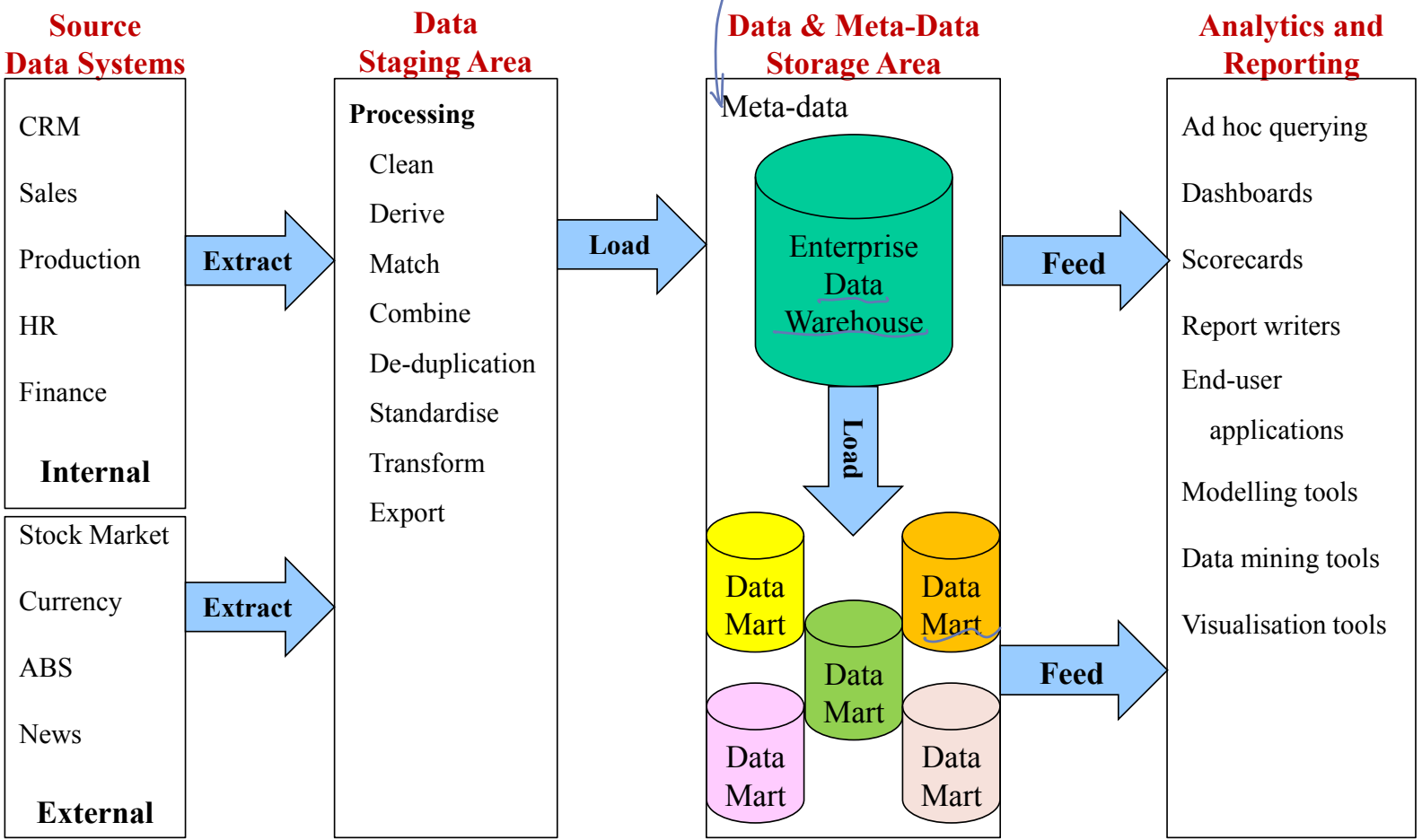
Non-volatile

- Users have Read access only – all updating done automatically by ETL* process and periodically by a DBA

*ETL stands for "extract, transform, load" - the three processes that, in combination, move data from one or more databases, or other sources to a unified repository, typically a data warehouse

A DW Architecture

data can be analysed through the whole data warehouse or all subset called data mart.



Data marts and data mining

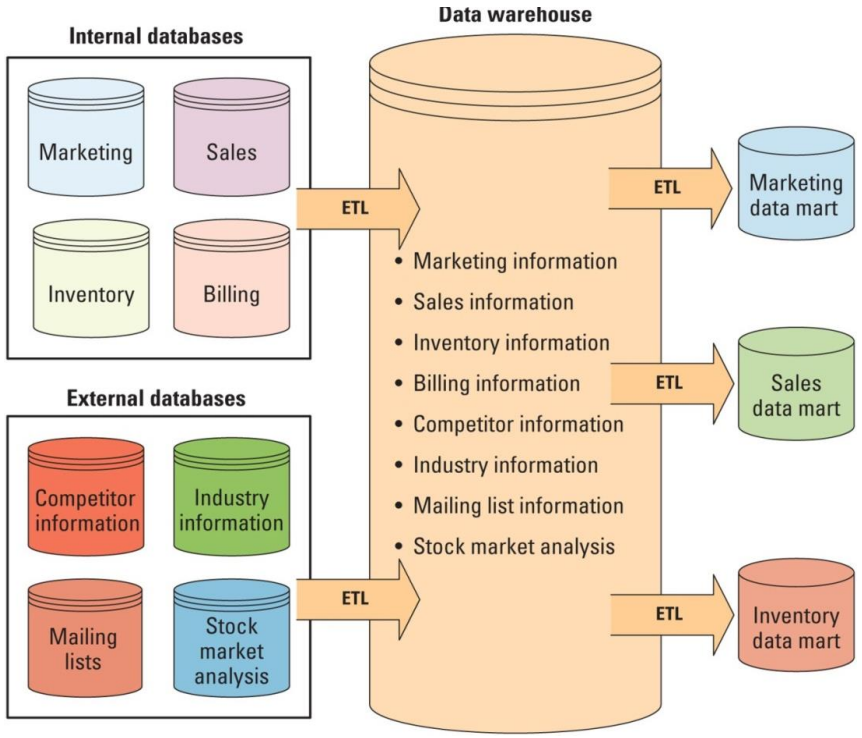
Data mart

- contains a subset of data warehouse information

when you work on data mart, the actual process when algorithm are applied to uncover things, it's called data mining.

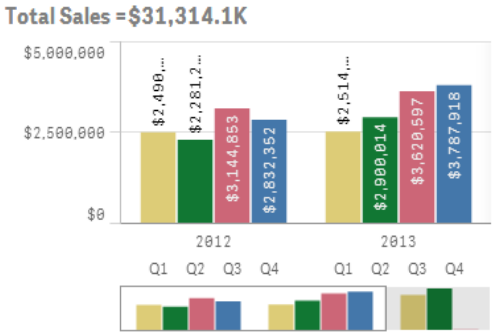
Data-mining

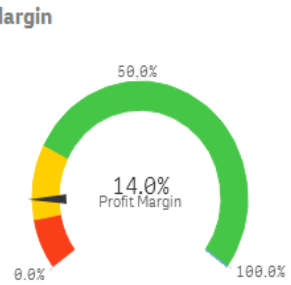
- A process in which algorithms are applied to information to uncover patterns and relationships which otherwise are difficult to find

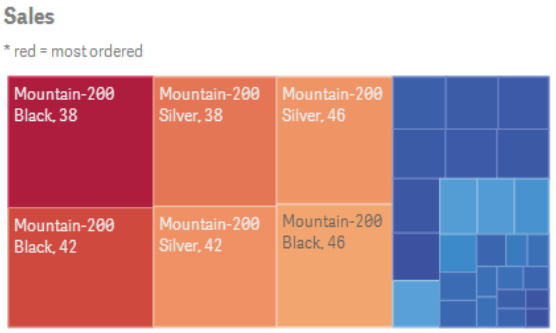


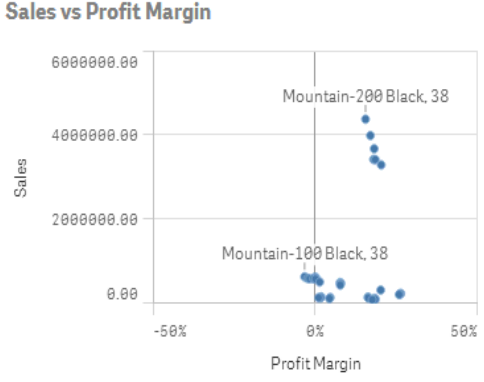
Baltzan, Lynch & Fisher 2015, p. 279

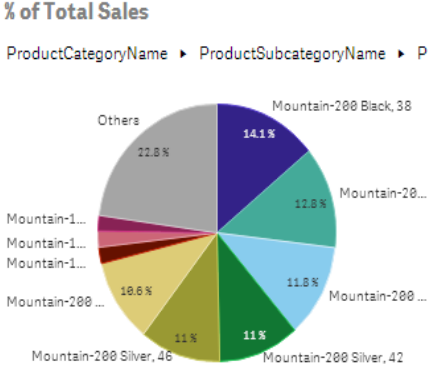
Business Intelligence Dashboard

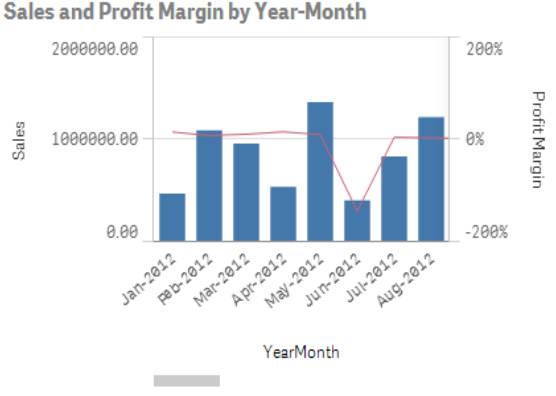


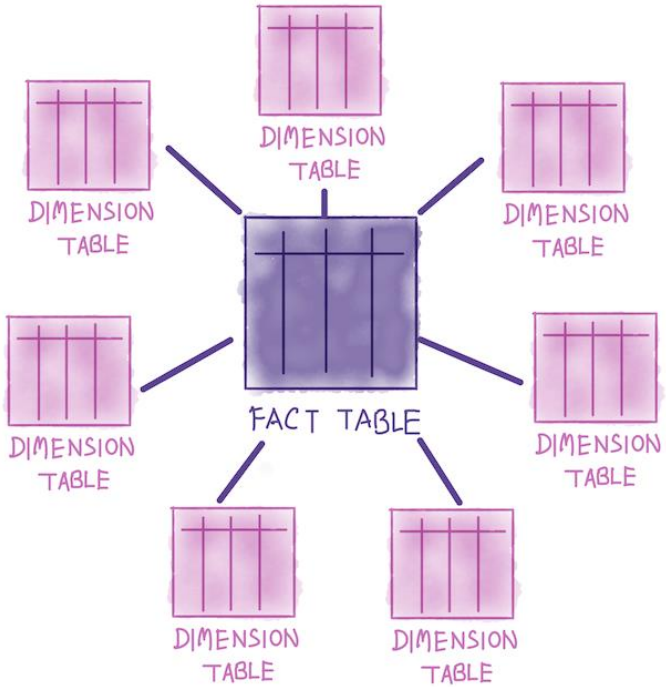












Dimensional Modelling

Introduction to Dimensional Modelling

Popularised by Ralph Kimball in the 1990s

Based on the *multi-dimensional* model of data and designed for retrieval-only databases

Very simple, intuitive, and easily-understood structure

Also known as *star schema* design

the central is the fact table.

Fact is what we need to analyse to know.

Dimension is what contributes to our facts.

A dimensional model consists of:

- Fact table
- Several dimensional tables
- (Sometimes) hierarchies in the dimensions

Essentially a simple and restricted type of ER model

Business Analyst World

Fact Dimension

How much **revenue** did the **product G** generate in the **last three months**, broken down by

Dimension

month for the south eastern sales **region**, by individual **stores**, broken down by

Dimension

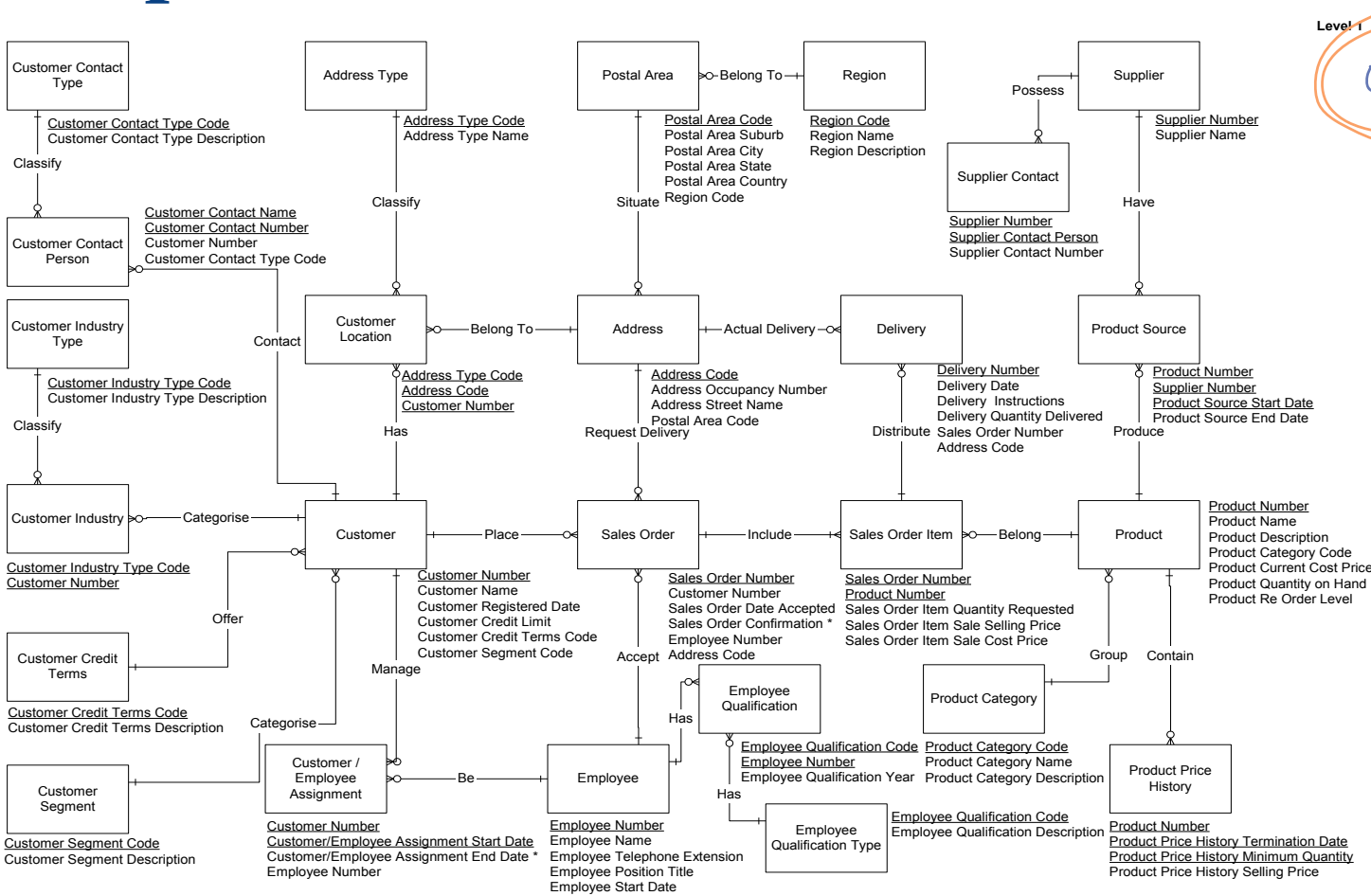
promotions, compared to estimates and to the previous version of the product

- Analysis starts usually with a single indication of something strange, then goes deep into the data, left to a new dimension, right to another, up to the summary, back down and left and right again, until the problem is identified…

- Dimensional Analysis: To support business analysts view
  – Revenue per product per customer per location?

Fact Dimension Dimension Dimension

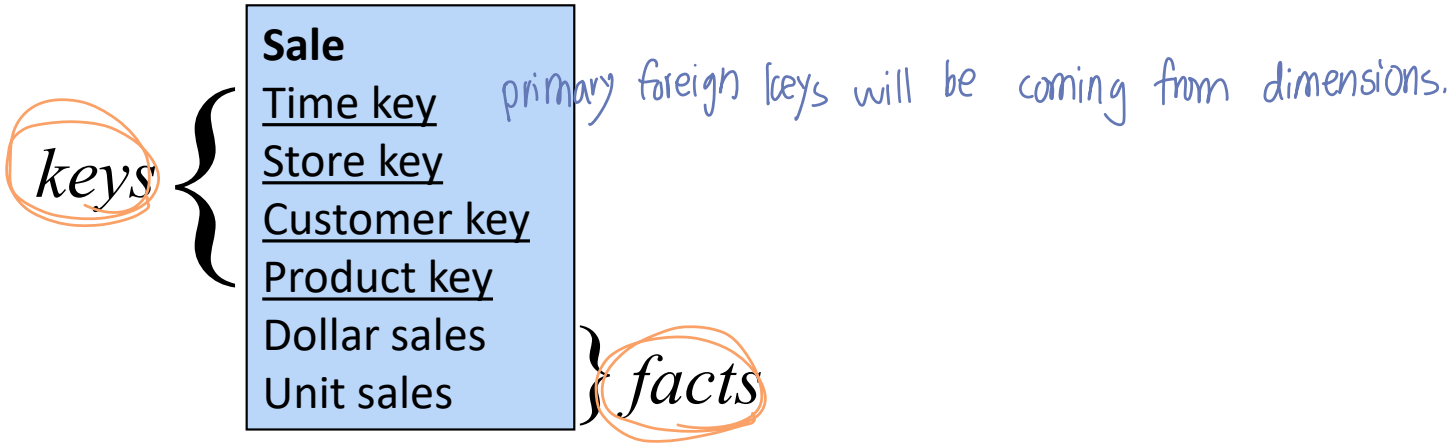
Example ER model



Fact Table

A fact table contains the actual business measures (additive, aggregates), called ***facts***

The fact table also contains *foreign keys* pointing to ***dimensions***



Fact Table - example

Actual data might look like this

Granularity, or level of detail, is a key issue

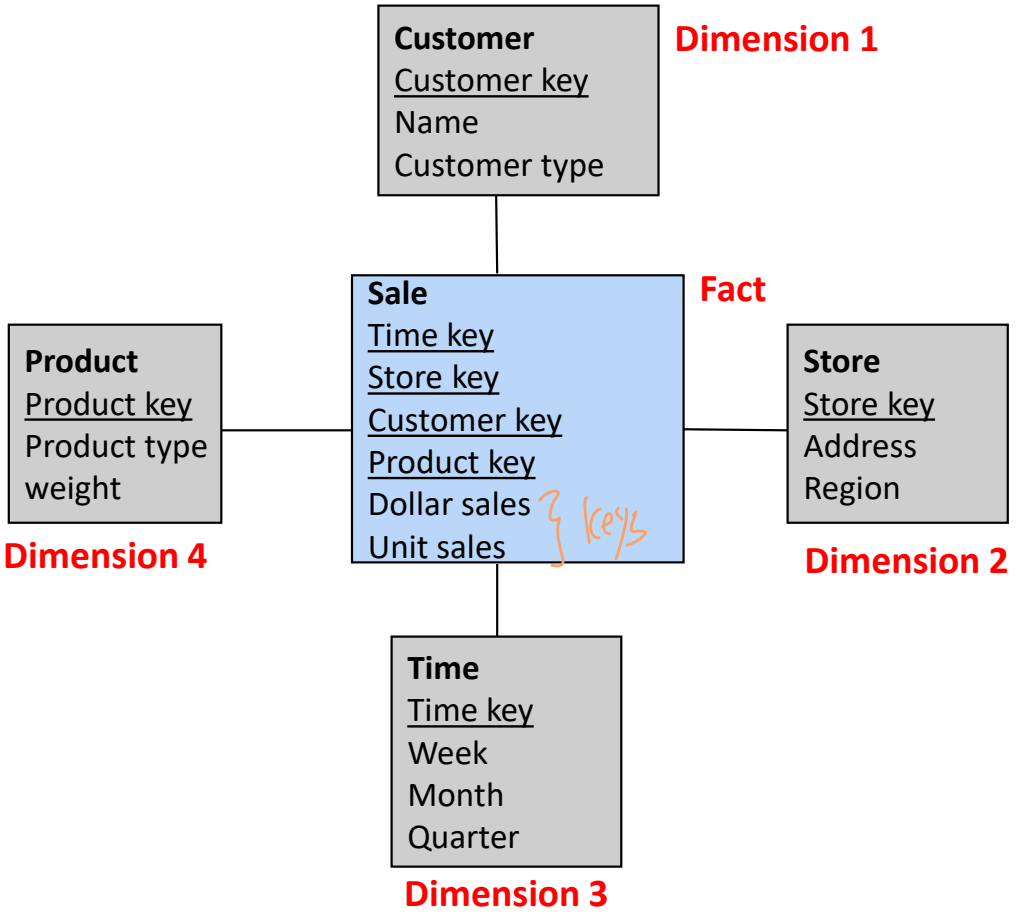
- Finest level of detail for a fact table, determined by the finest level of each dimension

there are time stamp.

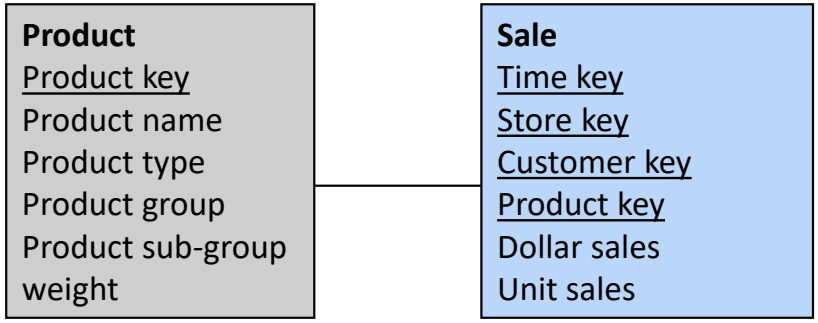
Time-id	*Store-id*	*Cust-id*	*Prod-id*	*Dollar sales*	*Unit Sales*
T100	S303	C101	P98	$120,000	5,000
T101	S303	C256	P98	$240000	10,000
T102	S387	C101	P10	$456,000	27,899
T100	S234	C400	P56	$100,200	5,600

Star schema – dimensional model

what, when, where, how



Dimension Hierarchies



Product name e.g. Hammer
- Product type e.g. Tool
  - Product group e.g. Hardware

Dimension Table - example

Actual data might look like this

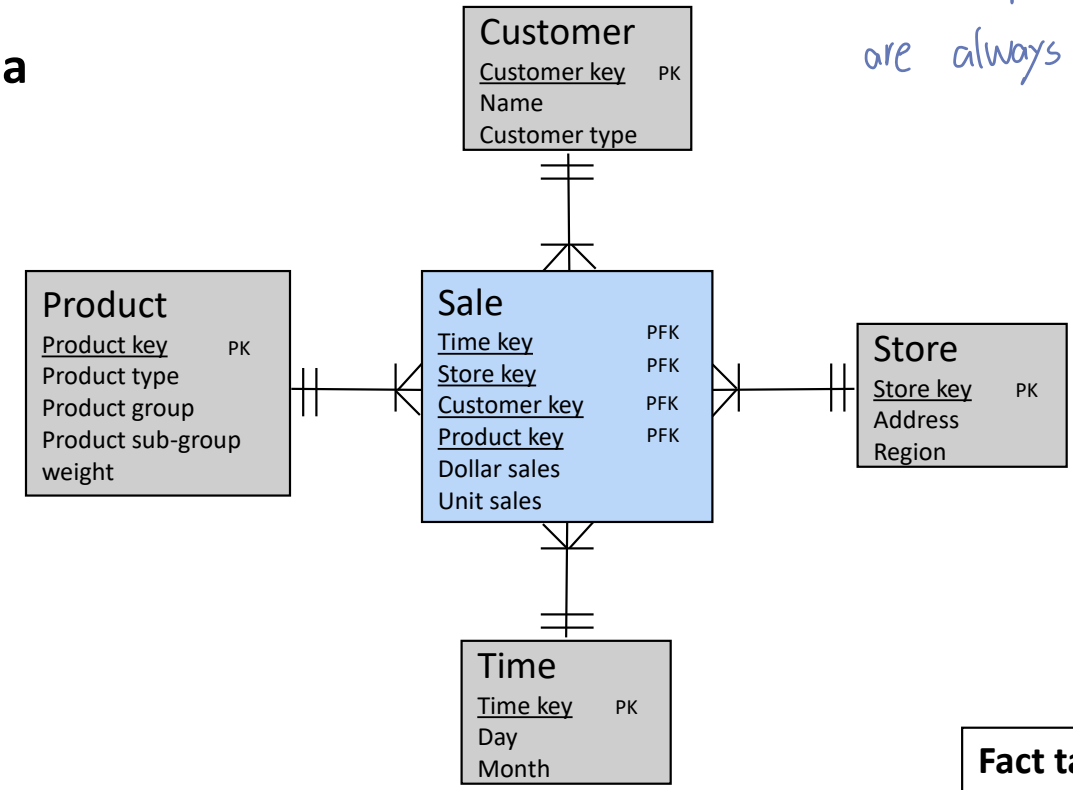
Hierarchy evident in data

Prod-id	*Prod-Name*	*Prod-Group*	*Prod-Subgroup*	*Weight*
P10	Hammer	Hardware	Tool	5kg
P56	10cm Nails	Hardware	Nails	1kg
P98	Plastic Pipe	Plumbing	Pipe	1kg

Dimensional model as an ER model

relationships between dimensions and facts are always one to many.

Star schema



Fact table is an intersection table

Designing a Dimensional Model

Steps:

1. Choose a Business Process what the manager could need.
2. Choose the measured facts (usually numeric, additive quantities)
3. Choose the granularity of the fact table
4. Choose the dimensions and whether you dimensions table will have sub dimensions.
5. Complete the dimension tables

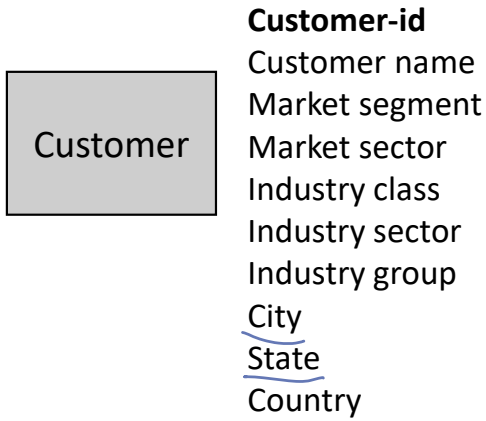
(usually numeric quantities or dollars)

decide the granularity in the fact table

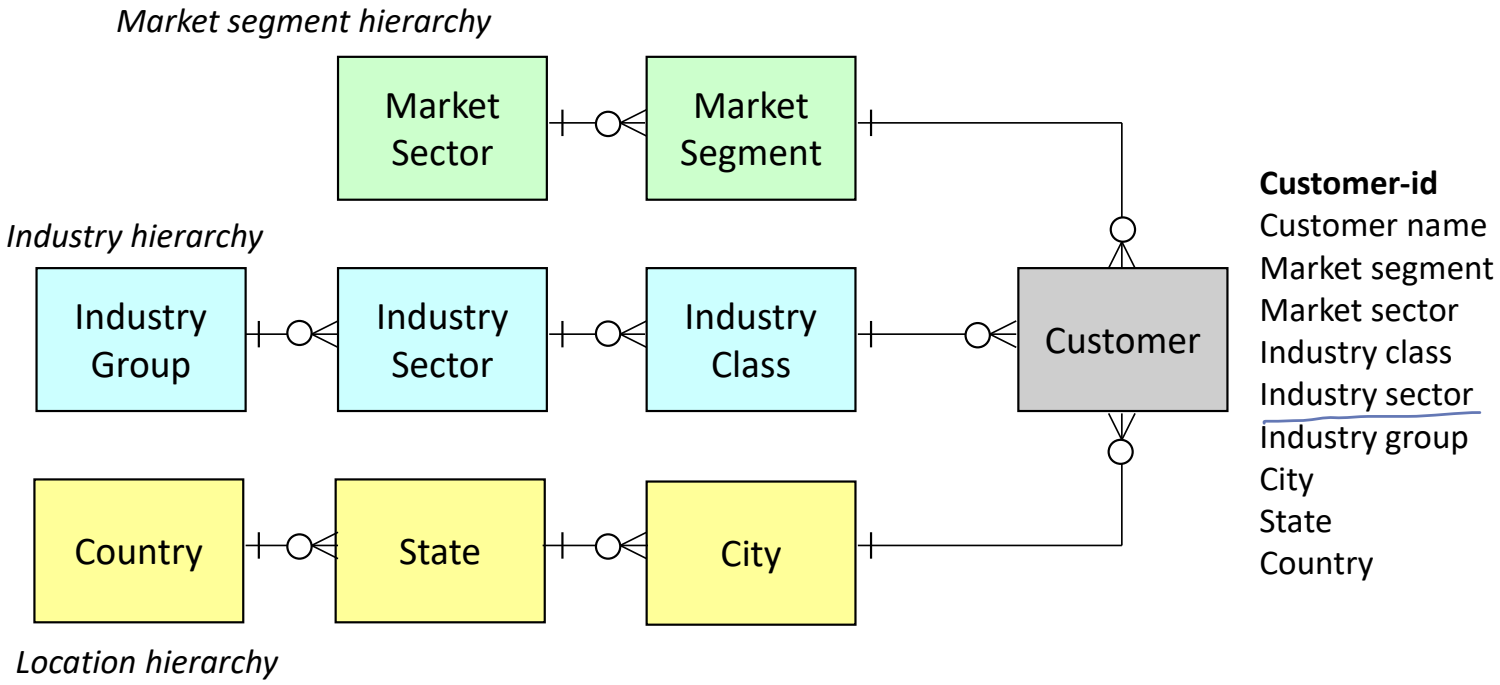
(Kimball, 1996)

Embedded Hierarchies in Dimensional Tables

sub - dimension

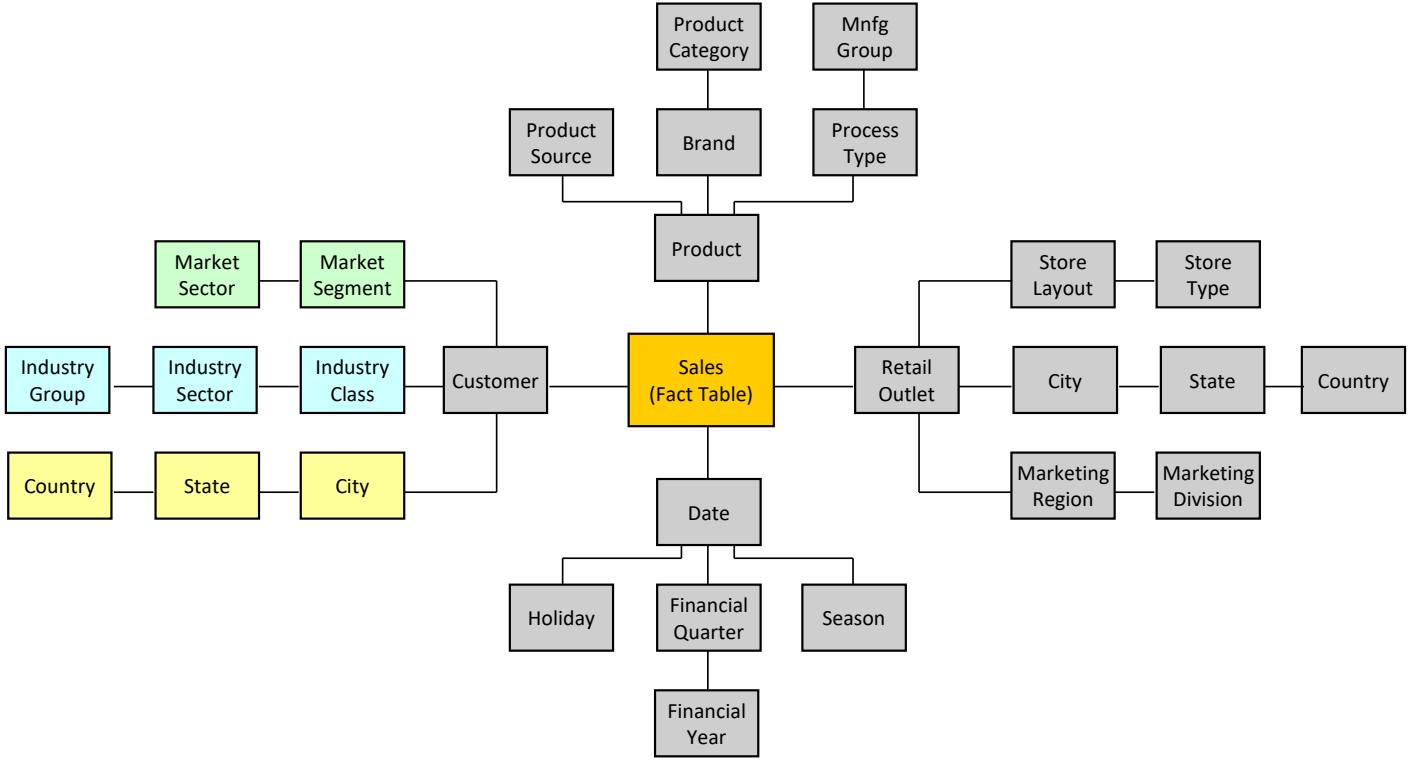


Embedded Hierarchies in Dimensional Tables



Snowflake Schema: hierarchy in dimensions

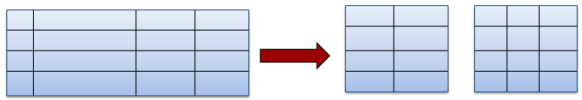
end up in real life.



Design Outcomes: Normalised or Denormalised?

Normalisation

- Eliminates redundancy
- Storage efficiency
- Referential Integrity



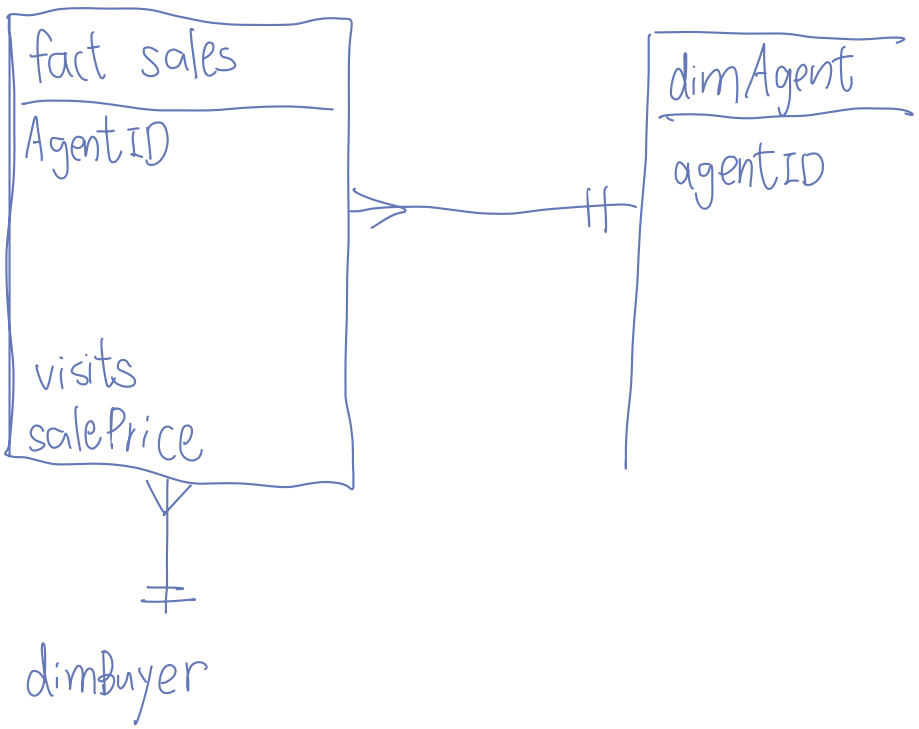
Denormalisation

- Fewer tables (fewer joins)
- Fast querying
- Design is tuned for end-user analysis

Exercise

We are making a data warehouse for a real estate agency. The company wants to track information about the **selling** of their properties. This warehouse keeps information about the **agents** (license#, first name, last name, phone #), **buyers** that come in (buyer id, first name, last name, phone #), and **property** (property#, property address, price). The information managers want to be able to find is **the number of times a property is viewed**, **sales price**. The information needs to be broken down **by rental agent, by buyer, by property** and **for different time** (day, week, month, quarter and year).

Draw a star schema to support the design of this data warehouse.


fact sales
AgentID
visits
salePrice
dimAgent
agentID
dimBuyer

What's examinable

- Differences between transactional and informational databases
- Modelling a star schema
- Identifying the best grain level
- Defining facts and dimension tables

THE UNIVERSITY OF MELBOURNE

Thank you