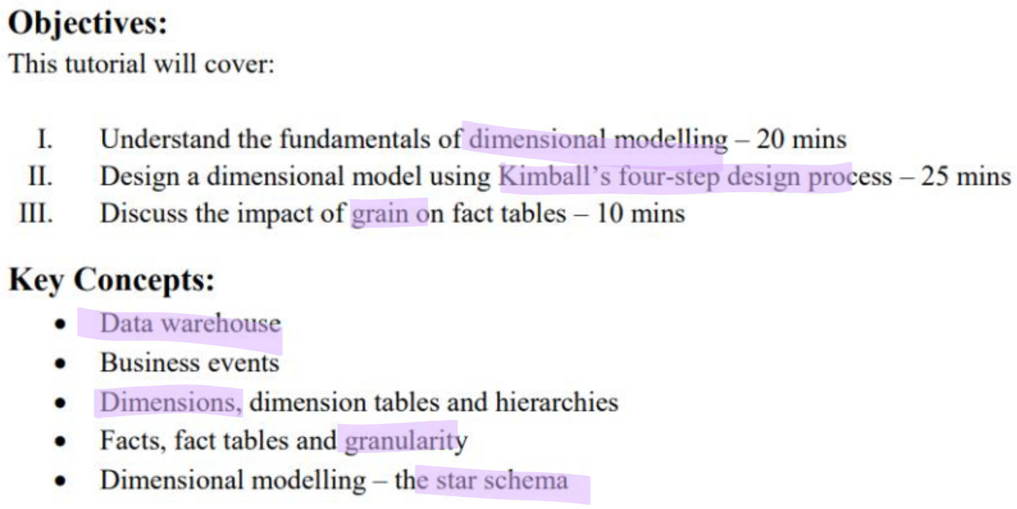
T11 Prashansa Singh

INFO20003 Tutorial 11

Starting ~2.20 pm

Today's tutorial

- Data Warehousing
- Exercises
  - group work



Objectives:

This tutorial will cover:

I. Understand the fundamentals of dimensional modelling – 20 mins
II. Design a dimensional model using Kimball's four-step design process – 25 mins
III. Discuss the impact of grain on fact tables – 10 mins

Key Concepts:

- Data warehouse
- Business events
- Dimensions, dimension tables and hierarchies
- Facts, fact tables and granularity
- Dimensional modelling – the star schema

Review of Data Warehousing

- Differences between
  - Transactional / Operation DB
  - Informational or Dimensional DB
  - ?

Transactional or Operation DB

- Normal DBs based on ER models seen in this subject
- Good for day-to-day operations
- E.g. a db for each dept:
  - accounting, sales, inventory etc (multiple dbs side by side)

Informational or Dimensional DB

- Just 1 single db storing all the information about an organisation
  - Don't need a db for each dept

- Useful for decision making processes by e.g. CEOs
  - Good to answer analytical questions
    That aggregate over events and look at many dimensions

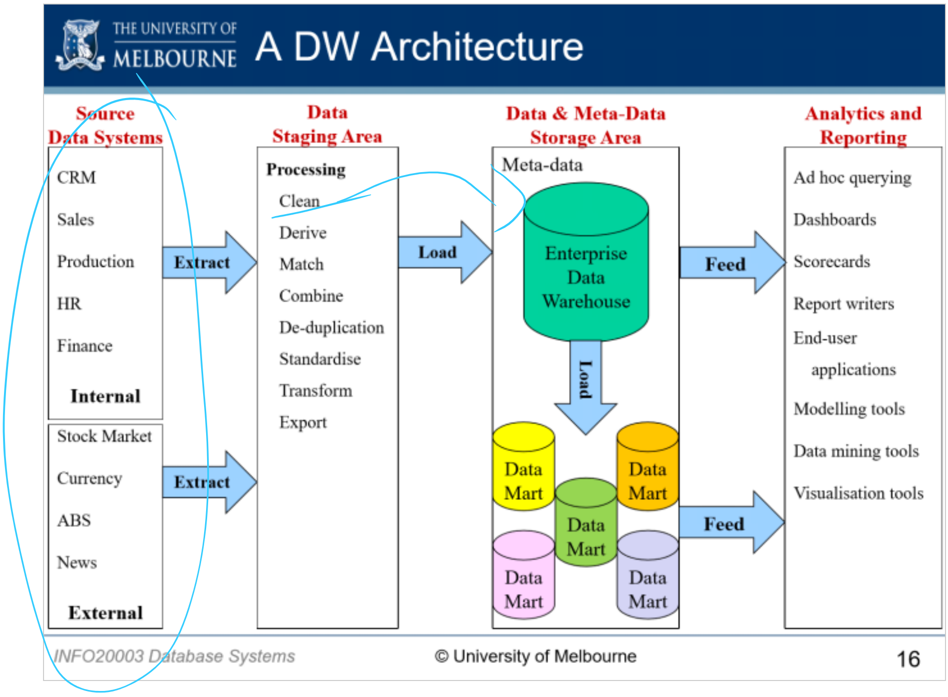
Data warehouse (DW)

A data warehouse is a **single** database of organisational data that allows **all** of the organisation's data to be stored in a form that **supports** managers' **decision making**.

The data is **integrated from multiple sources** internal and external to the organisation by converting it into a **common format** and **validating** before storing it to ensure credibility of the warehouse.

It keeps **historical** data (across time) and is available to the **managerial** bodies of the organisation to support **high-level decision-making** processes and data analysis.

Unlike an ER model, where data is organised around conceptual entities, the data in a data warehouse is organised around **business processes** such as **sales, finance, or marketing.**



THE UNIVERSITY OF MELBOURNE

A DW Architecture

Source Data Systems
- Internal: CRM, Sales, Production, HR, Finance
- External: Stock Market, Currency, ABS, News

Extract

Data Staging Area
- Processing: Clean, Derive, Match, Combine, De-duplication, Standardise, Transform, Export

Load

Data & Meta-Data Storage Area
- Meta-data
- Enterprise Data Warehouse
- Load
- Data Mart, Data Mart, Data Mart, Data Mart, Data Mart

Feed

Analytics and Reporting
- Ad hoc querying, Dashboards, Scorecards, Report writers, End-user applications, Modelling tools, Data mining tools, Visualisation tools

INFO20003 Database Systems © University of Melbourne 16

Business events

Data warehouses store information about **business events.**

A business event is an **event** that occurs as **part** of a **business process**.

Examples of Business events:

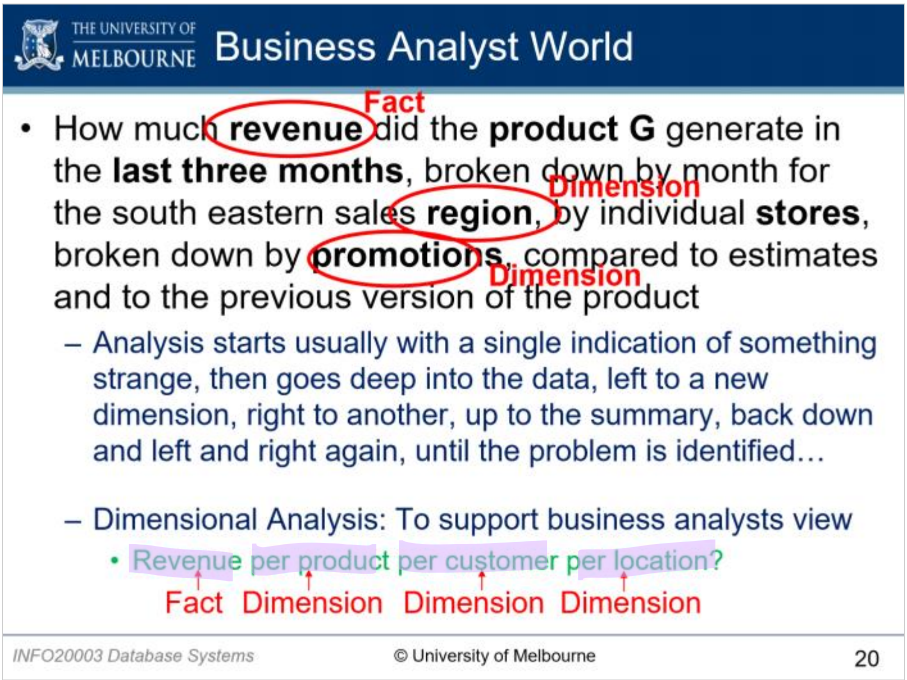
- For the **sales** business process, an **individual order or sale** would be considered a business event.

- For **finance**, a **payment** would be a business event.

- For a **marketing** data warehouse, it might be a **view** of a webpage or a **click** on an online ad.

Dimensions

A **dimension** is an entity that describes and gives **context** to a **business event.**
Some **examples** of commonly used dimensions are **time, customers, products and locations.**



THE UNIVERSITY OF MELBOURNE Business Analyst World

- How much **revenue** did the **product G** generate in the **last three months**, broken down by month for the south eastern sales **region**, by individual **stores**, broken down by **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?

INFO20003 Database Systems © University of Melbourne 20

For example, if a CEO is interested in a comparison of **revenue** of a new model of the product with the older model in every quarter of the year by customer demographic group, **the relevant dimensions are:**

- **...**

○ Time (quarter of the year)
○ Product (product version)
○ Customer (customer demographic).

Similarly, for an insurance company, an example of a key measurement (**fact**) is **claims**, and the **dimensions** could be **agent, policy, customer, and time (as these help describe the claim)**.

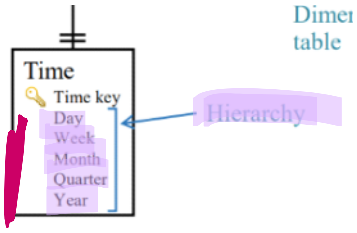
Dimension tables and hierarchies

Dimensions are represented in the data warehouse as **dimension tables.**
Within each dimension table, a range of attributes may be stored.

A sequence of attributes that describes a dimension across different levels of detail is called a **hierarchy**.

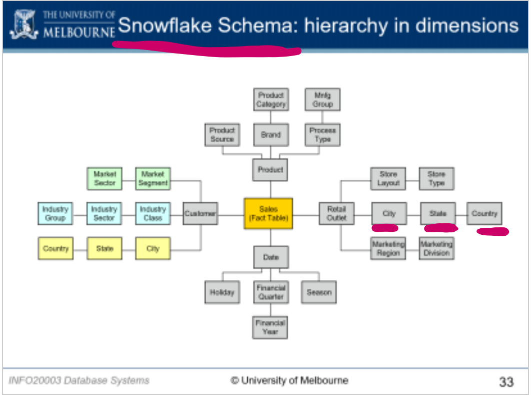
For example, for the dimension table '**Location**', the data can be stored at **various levels** (hierachy) **such as city, state**, or much higher level of **country** and so on.

Similarly, the **Time** dimension will have a **hierarchy of day, week, month, quarter and year.**



The **hierarchies** of dimensions are stored as **attributes** of the **dimensional tables** and all the related hierarchies are typically stored in a **single** dimension table

(Unless a snowflake schema is used, where parts of the hierarchy can become individual tables).

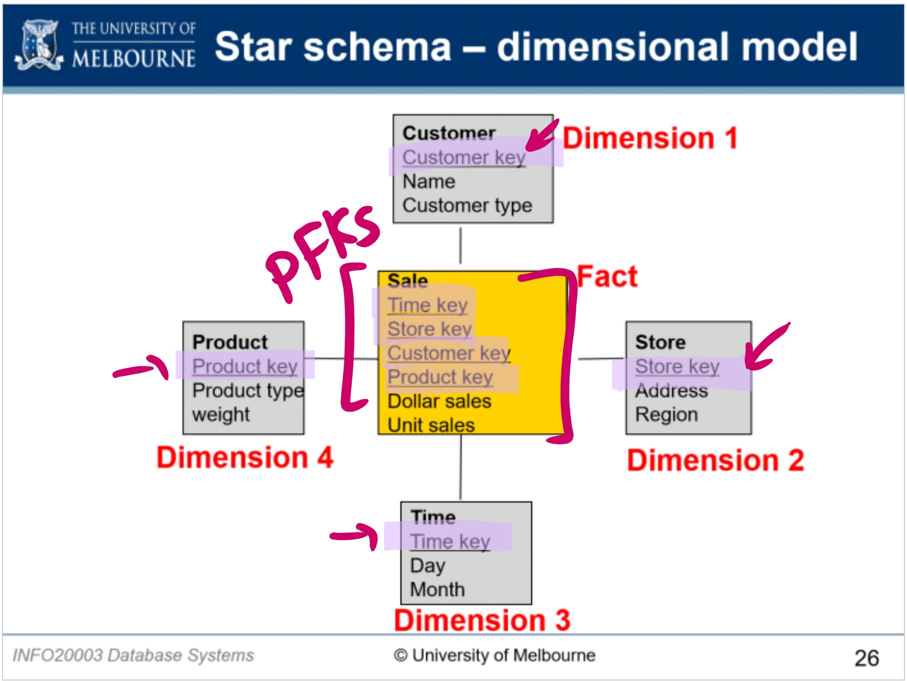


THE UNIVERSITY OF MELBOURNE Snowflake Schema: hierarchy in dimensions

INFO20003 Database Systems © University of Melbourne 33

- Will only use simple star schema in this subject

These **hierarchies** are **used** for **selecting and aggregating data** at the desired level of **detail**.
In other words, hierarchies help with slicing and dicing the data.



THE UNIVERSITY OF MELBOURNE **Star schema – dimensional model**

Customer
Customer key
Name
Customer type
Dimension 1

PFKS

Sale
Time key
Store key
Customer key
Product key
Dollar sales
Unit sales
Fact

Product
Product key
Product type
weight
Dimension 4

Store
Store key
Address
Region
Dimension 2

Time
Time key
Day
Month
Dimension 3

INFO20003 Database Systems © University of Melbourne 26

Facts and fact tables

A **fact** is a numeric **measurement** of a meaningful and significant business event.

- Something numerical we can measure

Consider the scenario where a customer buys a product at a certain location at a certain time. The **intersection** of these four dimensions constitutes a **sale** (the **business event**).

The sale can be measured in terms of the

- **amount of revenue generated**
- **number of items sold**
- **total profit earned, etc**

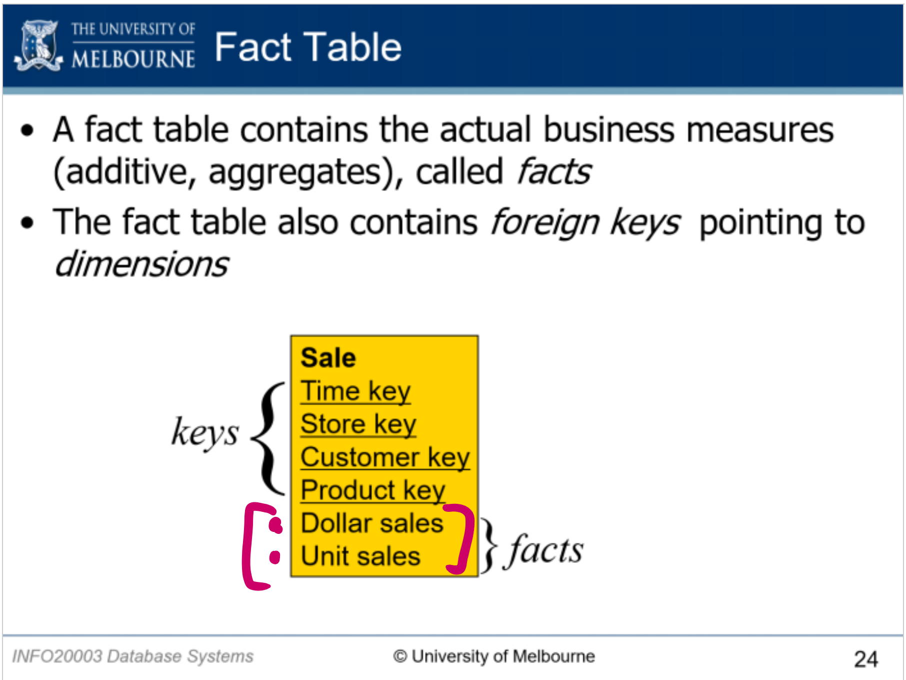
These are all **facts** relating to the sale.

In a data warehouse, the facts (numerical performance measurements) of a business are stored in a **fact table.**

A row in a fact table corresponds to one or more business events.

A data warehouse **fact table** is defined as an **intersection** of the **dimensions** that describe the business event.

In general, the fact table has a **PK** made up of the **foreign keys connecting it to the dimension tables. (PFKs)**



THE UNIVERSITY OF MELBOURNE

Fact Table

- A fact table contains the actual business measures (additive, aggregates), called *facts*
- The fact table also contains *foreign keys* pointing to *dimensions*

Sale
Time key
Store key
Customer key
Product key
Dollar sales
Unit sales

INFO20003 Database Systems © University of Melbourne 24

Notice: Difference between 'fact table' and 'fact'

Granularity

What is Granularity or Grain?

The **level of detail** present in a fact table is referred to as **grain** or **granularity**.

- ○ The **fact table can store each business event in its own row**
  - □ for example, preserving each sales event as an individual row
- ○ Or it can store **many business events aggregated together**
  - □ if sales data is aggregated down to one row per hour (or per day)

The **finer** the **granularity** is, the more **precisely** a query can **extract details** from the database.

<u>Dimensional modelling – the star schema</u>

The model in which the **fact table** consisting of numeric measurements (facts) is **related** to all the **dimension tables** storing descriptive attributes is termed the "**dimensional model"**.

The **fact table is at the centre** and the **dimensional tables are on the sides**, making a **star schema**.

Example: Figure 1 shows a star schema with **Sale as the fact table** and **Customer, Product, Store and Time as dimensions** of the business.

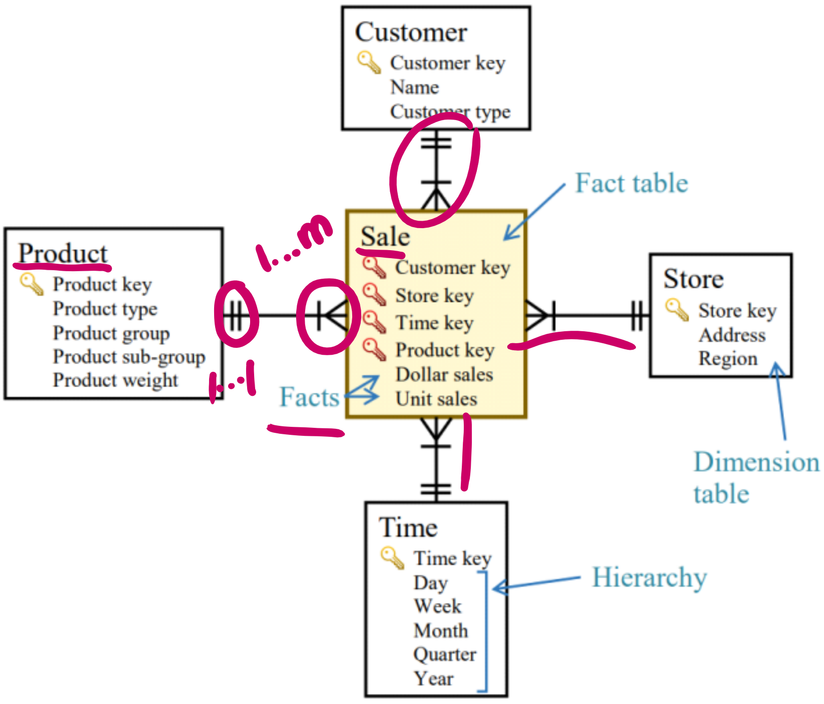
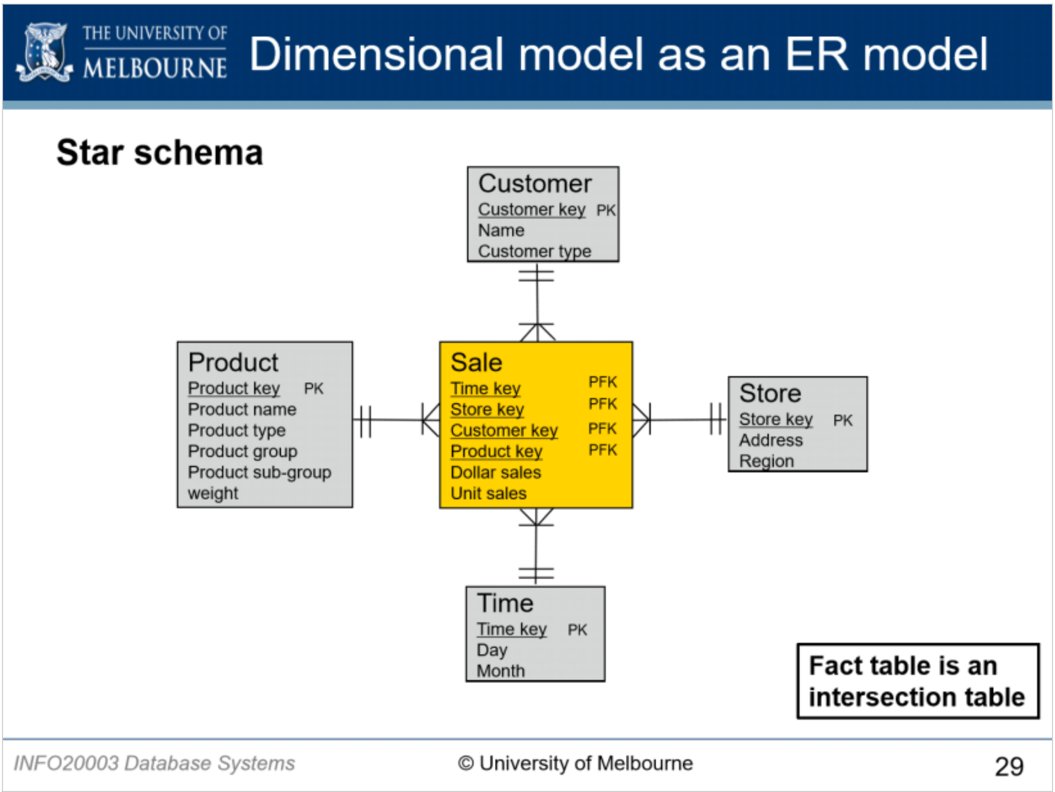


Figure 1: A simple star schema for a sales data warehouse with four dimension tables.



THE UNIVERSITY OF MELBOURNE

Dimensional model as an ER model

Star schema

Customer
Customer key PK
Name
Customer type

Product
Product key PK
Product name
Product type
Product group
Product sub-group
weight

Sale
Time key PFK
Store key PFK
Customer key PFK
Product key PFK
Dollar sales
Unit sales

Store
Store key PK
Address
Region

Time
Time key PK
Day
Month

Fact table is an intersection table

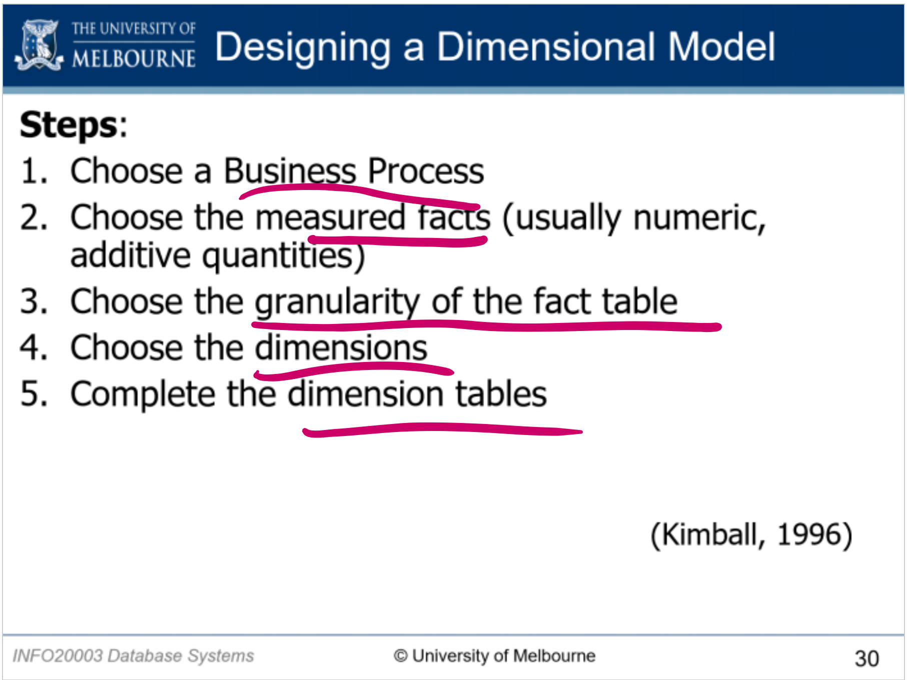
INFO20003 Database Systems © University of Melbourne 29

- PFKs referring to dimension tables

- Note cardinality
    - ○ What does this imply?

    - ○ e.g. 1 product /customer/store can have 1 to many (1...m) sales related to it
    - ○ However, each sale captured is for a single (1 and only 1 = 1...1) product, customer, store and single time

Kimball's four-step design process



THE UNIVERSITY OF MELBOURNE

Designing a Dimensional Model

Steps:

1. Choose a Business Process
2. Choose the measured facts (usually numeric, additive quantities)
3. Choose the granularity of the fact table
4. Choose the dimensions
5. Complete the dimension tables

(Kimball, 1996)

INFO20003 Database Systems © University of Melbourne 30

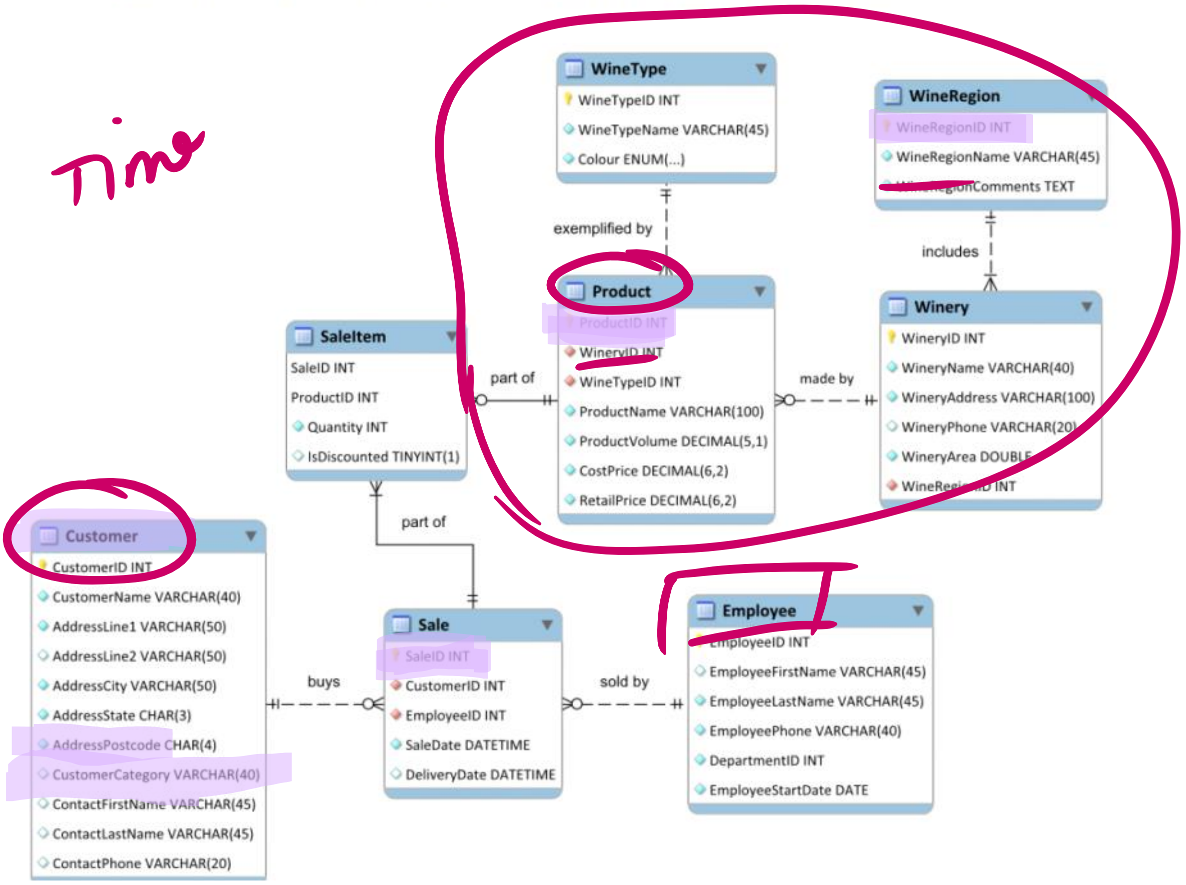
Group Work

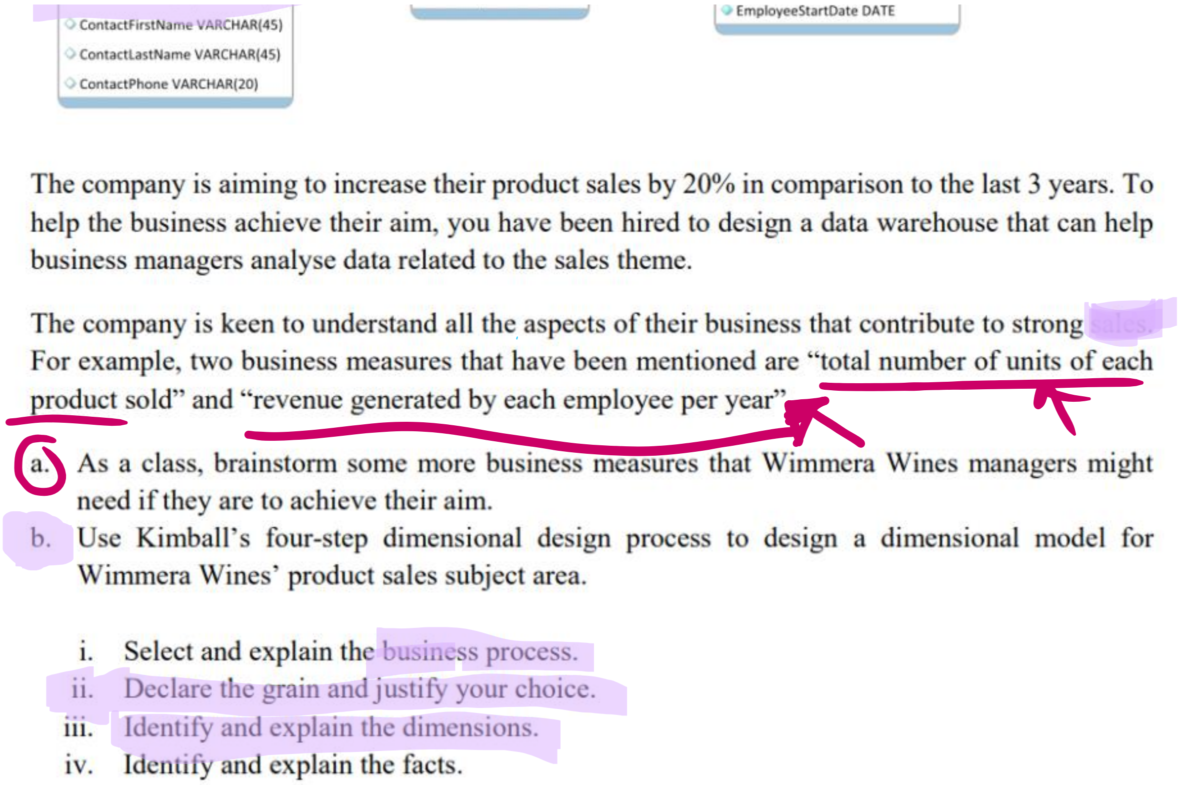
Exercise:

1. Designing a dimensional model

Wimmera Wines is a large company that takes deliveries of grapes from wine growers, produces and bottles wine, and sells those bottles to retailers and restaurants. They produce many different types of wine at a range of price points, from cheap cask wine to top-of-the-range vintage bottles.

Wimmera Wines' day-to-day OLTP database uses the following ER model:





ContactFirstName VARCHAR(45)
ContactLastName VARCHAR(45)
ContactPhone VARCHAR(20)

EmployeeStartDate DATE

The company is aiming to increase their product sales by 20% in comparison to the last 3 years. To help the business achieve their aim, you have been hired to design a data warehouse that can help business managers analyse data related to the sales theme.

The company is keen to understand all the aspects of their business that contribute to strong sales. For example, two business measures that have been mentioned are "total number of units of each product sold" and "revenue generated by each employee per year".

a. As a class, brainstorm some more business measures that Wimmera Wines managers might need if they are to achieve their aim.

b. Use Kimball's four-step dimensional design process to design a dimensional model for Wimmera Wines' product sales subject area.

i. Select and explain the business process.
ii. Declare the grain and justify your choice.
iii. Identify and explain the dimensions.
iv. Identify and explain the facts.

a) Brainstorming

- Number of products sold per year
- Sales by a particular state
- Sales of a product in a given quarter of a year
- Revenue generated from a particular customer category
- Which product is selling the best (hence generating the most revenue)?

All these measures (including the example ones given in the question)

- Are things that will potentially help us understand how to boost sales
- Who is the best performing employee - what are they doing?
- What is the most popular product

(Analytical questions we will use our star schema/ DW to answer)

b) Group work

b. Use Kimball's four-step dimensional design process to design a dimensional model for Wimmera Wines' product sales subject area.

i. Select and explain the business process.

As stated in the case study, product **sales is the business process**. (Sales is a very common business process)

Analyses related to sales can be of varying natures and may use different measures associated with Sales.

ii. Declare the grain and justify your choice.

Because Wimmera Wines sells to retailers and restaurants, they would not make a large **number** of sales (but **each** individual **sale** can include **large quantities** of items).

- So we're probably not making many sales e.g. not making sales every hour or even every day

It is appropriate to store **each sale item as its own row** in the fact table with **no aggregation**. This allows for most precise data.

If we did want to aggregate and perhaps weekly sales are sufficient, the data would be aggregated by **week**.

- (Not easy to specify a time grain)

This is just for the grain of the sales table, but can also think about the grain of the dimension tables.

iii. Identify and explain the dimensions.

Looking at the data available from the given ER model of the existing database, and considering the business process (sales), the following dimensions are relevant for evaluating business measures related to Sales:

- Employee
- Customer
- Time
- Product

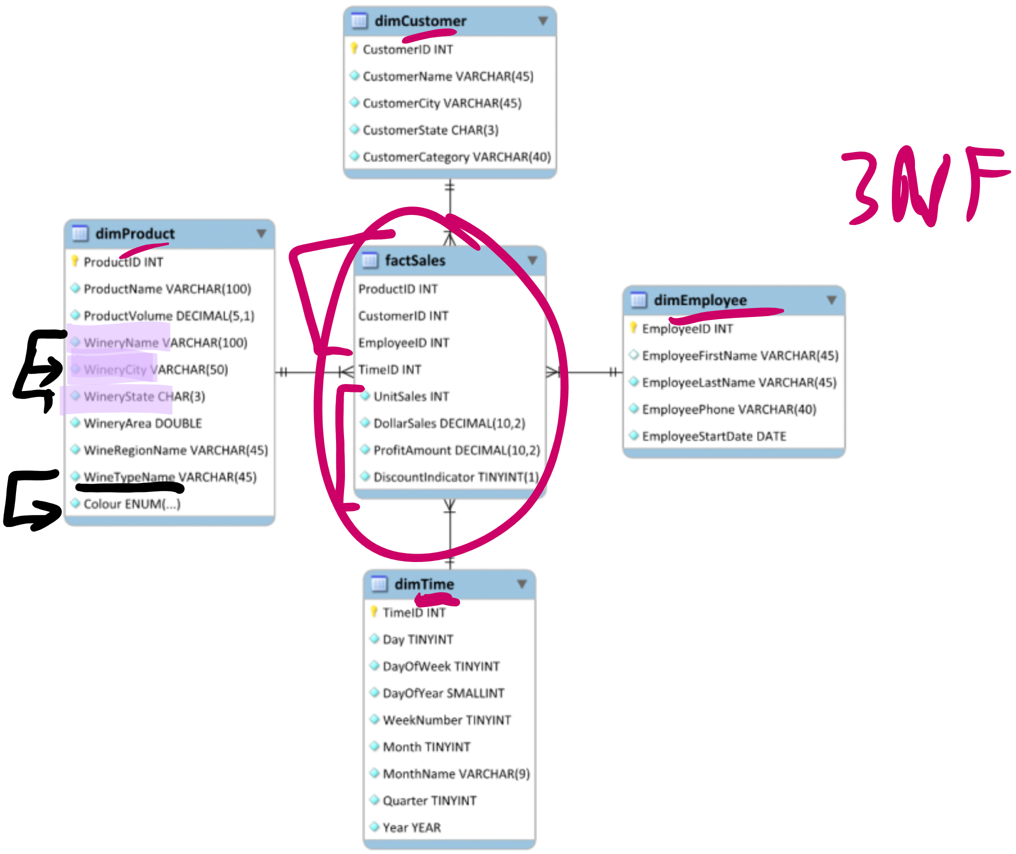
iv. Identify and explain the facts.

The following **sales-related facts** can be extracted from the source database:

- **Unit sales (# of products sold)**
- **Dollar sales**
- Profit amount
- Discount indicator (non-additive fact)
  - ▪ Non-additive means:
    - □ We are not aggregating/summing over anything
    - □ For each sale, we store whether it was discounted or not

Roughly what would the star schema for this DW look like?

Appendix: Question 1 star schema solution



dimCustomer
- CustomerID INT
- CustomerName VARCHAR(45)
- CustomerCity VARCHAR(45)
- CustomerState CHAR(3)
- CustomerCategory VARCHAR(40)

dimProduct
- ProductID INT
- ProductName VARCHAR(100)
- ProductVolume DECIMAL(5,1)
- WineryName VARCHAR(100)
- WineryCity VARCHAR(50)
- WineryState CHAR(3)
- WineryArea DOUBLE
- WineRegionName VARCHAR(45)
- WineTypeName VARCHAR(45)
- Colour ENUM(...)

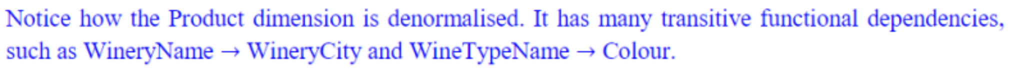
factSales
- ProductID INT
- CustomerID INT
- EmployeeID INT
- TimeID INT
- UnitSales INT
- DollarSales DECIMAL(10,2)
- ProfitAmount DECIMAL(10,2)
- DiscountIndicator TINYINT(1)

dimEmployee
- EmployeeID INT
- EmployeeFirstName VARCHAR(45)
- EmployeeLastName VARCHAR(45)
- EmployeePhone VARCHAR(40)
- EmployeeStartDate DATE

dimTime
- TimeID INT
- Day TINYINT
- DayOfWeek TINYINT
- DayOfYear SMALLINT
- WeekNumber TINYINT
- Month TINYINT
- MonthName VARCHAR(9)
- Quarter TINYINT
- Year YEAR

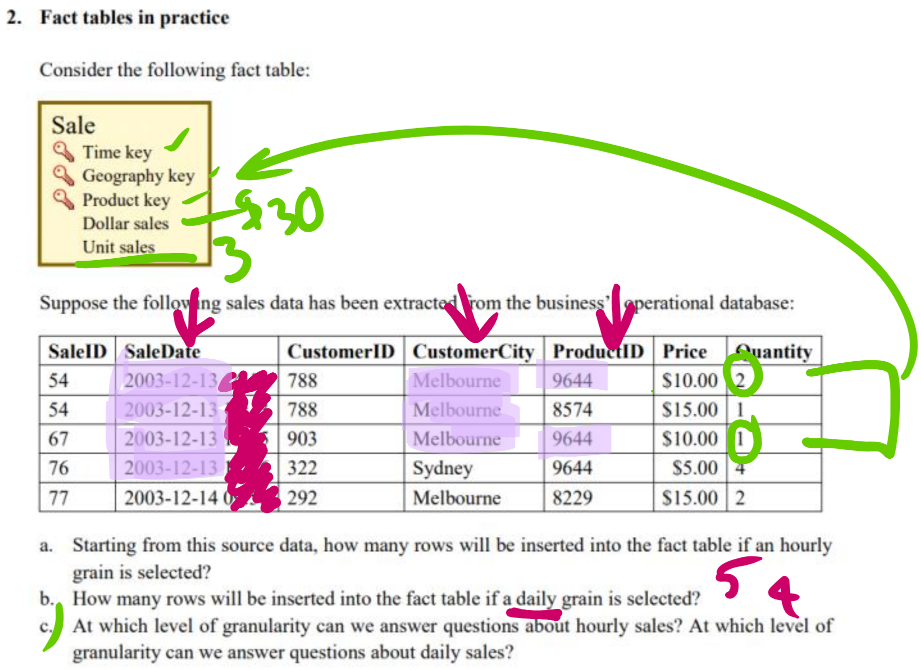
- Facts
- Dimensions

- Denormalised in this example. How can you tell?



Notice how the Product dimension is denormalised. It has many transitive functional dependencies, such as WineryName → WineryCity and WineTypeName → Colour.

Group Work



2. Fact tables in practice

Consider the following fact table:

Sale
Time key
Geography key
Product key
Dollar sales $30
Unit sales 3

Suppose the following sales data has been extracted from the business' operational database:

SaleID	SaleDate	CustomerID	CustomerCity	ProductID	Price	Quantity
54	2003-12-13	788	Melbourne	9644	$10.00	2
54	2003-12-13	788	Melbourne	8574	$15.00	1
67	2003-12-13	903	Melbourne	9644	$10.00	1
76	2003-12-13	322	Sydney	9644	$5.00	4
77	2003-12-14	292	Melbourne	8229	$15.00	2

a. Starting from this source data, how many rows will be inserted into the fact table if an hourly grain is selected? 5

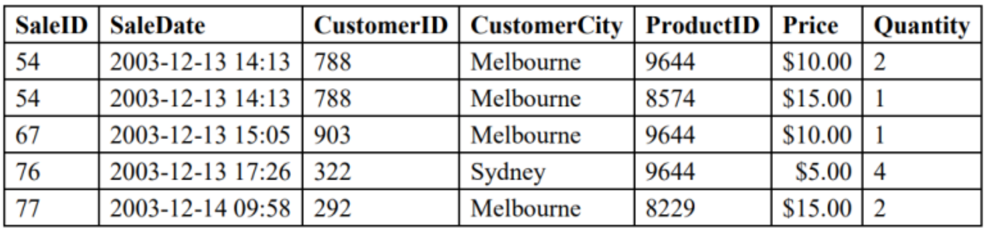
b. How many rows will be inserted into the fact table if a daily grain is selected? 4

c. At which level of granularity can we answer questions about hourly sales? At which level of granularity can we answer questions about daily sales?

When thinking about aggregation

- Must consider PKs. What are the PKs?

○ PKs are Time, Geography, Product = SaleDate, CustomerCity, ProductID:



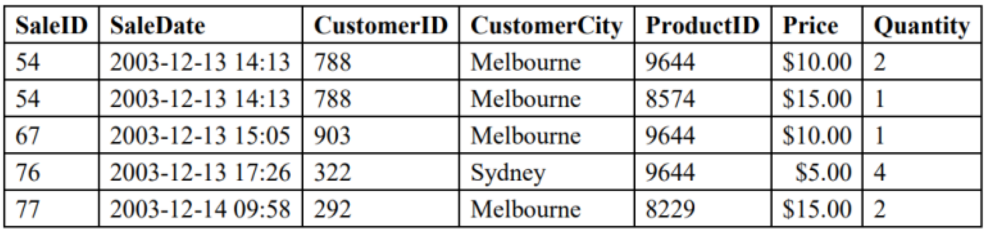
SaleID	SaleDate	CustomerID	CustomerCity	ProductID	Price	Quantity
54	2003-12-13 14:13	788	Melbourne	9644	$10.00	2
54	2003-12-13 14:13	788	Melbourne	8574	$15.00	1
67	2003-12-13 15:05	903	Melbourne	9644	$10.00	1
76	2003-12-13 17:26	322	Sydney	9644	$5.00	4
77	2003-12-14 09:58	292	Melbourne	8229	$15.00	2

- The PKs must have the same values for two or more rows to be aggregated

- SaleID is not PK.
- SaleID, CustomerID attributes are not even stored in fact table

a. Starting from this source data, how many rows will be inserted into the fact table if an hourly grain is selected?

None of these sale-item rows share the same hour, geography and product. No aggregation can be performed. Five rows will be inserted into the fact table.



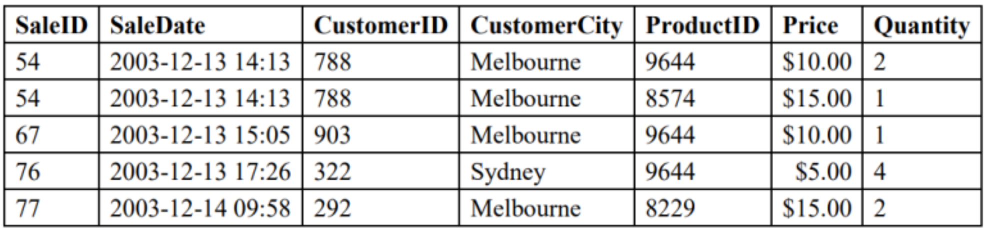
SaleID	SaleDate	CustomerID	CustomerCity	ProductID	Price	Quantity
54	2003-12-13 14:13	788	Melbourne	9644	$10.00	2
54	2003-12-13 14:13	788	Melbourne	8574	$15.00	1
67	2003-12-13 15:05	903	Melbourne	9644	$10.00	1
76	2003-12-13 17:26	322	Sydney	9644	$5.00	4
77	2003-12-14 09:58	292	Melbourne	8229	$15.00	2

b. How many rows will be inserted into the fact table if a daily grain is selected?

The first sale-item of sale **54**, and the sale-item of sale **67**, took place on the same day, at the same location and relate to the same product.

These two rows will be **aggregated** into a **single row** in the fact table with Dollar Sales = **$30.00** and Quantity = **3.**

In total, four rows will be inserted into the fact table.



SaleID	SaleDate	CustomerID	CustomerCity	ProductID	Price	Quantity
54	2003-12-13 14:13	788	Melbourne	9644	$10.00	2
54	2003-12-13 14:13	788	Melbourne	8574	$15.00	1
67	2003-12-13 15:05	903	Melbourne	9644	$10.00	1
76	2003-12-13 17:26	322	Sydney	9644	$5.00	4
77	2003-12-14 09:58	292	Melbourne	8229	$15.00	2

While there are more products sold on that day, note that we **cannot aggregate** such records because they are:

i) for a different product
    e.g. two sale-items of sale 54 for products 9644 and 8574

or

ii) for a different geography (region)
    e.g. sale 76 is for the Sydney region

c. At which level of granularity can we answer questions about hourly sales? At which level of granularity can we answer questions about daily sales?

Information about the hour when a sale was made is not stored if a daily grain is used.

Questions about hourly sales can only be answered when the grain is hourly (or finer).

We can answer questions about daily sales when the grain is daily (or finer).

We can also answer these daily questions from an hourly-grain fact table
○ How?

- up to 24 hourly rows can be combined (aggregated) into a single daily row when the fact table is queried, using a GROUP BY clause.