

# 資料倉儲(Data Warehouse, DW)

**Dr. Tun-Wen Pai**

- 1) Introduction to Data Warehouse
- 2) Operational and Informational Systems
- 3) DW Architecture / data models
- 4) Data Cube(slicing/dicing/roll-up/drill-down/pivot)

March 8, 2023

# Definitions

- **Data Warehouse**

- A **subject-oriented, integrated, time-variant, non-updatable collection of data** used in support of management **decision-making** processes
  - **Subject-oriented:** e.g. employee, customers, students, patients, achievement, products, performance, conditions
  - **Integrated:** consistent naming conventions, synonyms, formats , encoding structures, from multiple data sources, inconsistent key structures, Inconsistent data values, missing data
  - **Time-variant:** can study trends and changes
  - **Non-updatable:** read-only, periodically refreshed

- **Data Mart**

- A data warehouse that is limited in scope

資料倉儲就是一種經過優化的儲存過程，用來儲存結構化的資料，以進行後續快速的資訊查詢，及時提供商業決策依據。

# DATA WAREHOUSE (DW)

- DW supports business reporting and data mining activities.
- DW enables a comprehensive/integrated view of corporate data, all cleaned and organized.
- The objective of DW is to provide **business knowledge to support decision making.**

***“operational systems → informational systems”***

# Operational → Informational Systems

- **Operational system** – a system that is used to run a business in real time, based on current data; also called a system of record (SOR)

操作系統是指用於處理組織的日常事務的系統。這些系統的設計方式是有效執行日常交易的處理，並保持交易數據的完整性。

- **Informational system** – a system designed to support decision making based on **historical point-in-time** and **prediction data** for complex queries or data-mining applications

訊息系統是指用於收集，存儲和處理數據組件的集成，其中數據用於提供信息，知識貢獻以及促進決策的數字產品。

# Data Warehouse (DW)

- 資料倉儲有四種主要元素：中央資料庫、資料整合工具、後設資料及資料存取工具。所有這些元素都以速度為設計理念，旨在讓使用者能快速取得結果並即時分析資料

**中央資料庫：**就地部署或在雲端執行的標準關聯式資料庫。具有即時性、能大幅降低 RAM 成本的記憶體式運算資料庫正迅速竄起，廣受歡迎。

**資料整合：**從來源系統抓取資料後，根據資訊進行修改以快速分析消化，採用的資料整合方法包括 ETL（擷取、轉換、載入）ELT、即時資料複製、大量處理、資料轉換、資料品質以及擴充服務。

**後設資料：**後設資料是描述資料的資訊。說明資料集的來源、使用、價值及其他特徵。例如學校學生的後設資料後設資料會說明過去學習背景。而實習課程中繼資料則說明課程內容及實習操作方式。

**資料倉儲存取工具：**存取工具能讓使用者與資料倉儲的資料互動。存取工具的範例包括：查詢與報告工具、應用程式開發工具、資料採集工具以及 OLAP 工具。

## Comparison of Operational and Informational Systems

Characteristic	Operational Systems	Informational Systems
Primary purpose	Run the business on a current basis	Support managerial decision making
Type of data	Current representation of state of the business	Historical point-in-time (snapshots) and predictions
Primary users	Clerks, salespersons, administrators	Managers, business analysts, customers
Scope of usage	Narrow, planned, and simple updates and queries	Broad, ad hoc, complex queries and analysis
Design goal	Performance: throughput, availability	Ease of flexible access and use
Volume	Many constant updates and queries on one or a few table rows	Periodic batch updates and queries requiring many or all rows



# Design Considerations for DW

1. Subject oriented 主題導向 – single subject
2. Integrated 整合性 – comprehensive view of subject
3. Time-variant (時間序列) 時變性 – grow daily/intervals
4. Nonvolatile 穩定性 -- consistently available
5. Summarized 摘要整理 – reduce variables and dimension
6. Not normalized 非標準化 – star scheme (look-up tables)
7. Metadata 元資料(後設資料) – derived from other primitive variables
8. Near Real-time and/or right time (active) 即時/合適的時間 (主動)



# DW開發方法

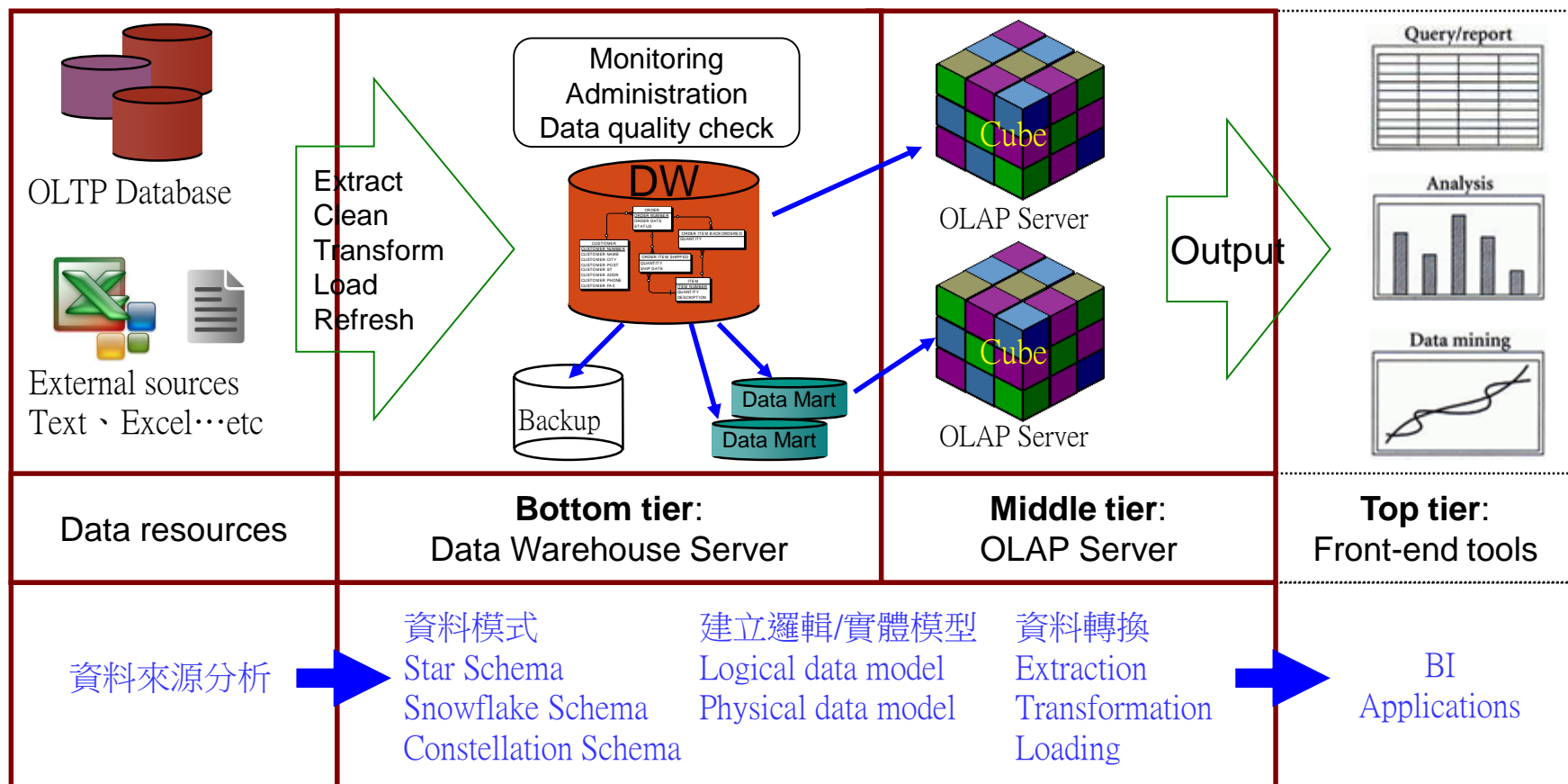
## ■ Different approaches to develop DW

1. **top down** : make a comprehensive DW that covers all the reporting needs of the enterprise.
2. **bottom up** : produce small data marts, for the reporting needs of different departments or functions, as needed. The smaller data marts will eventually align to deliver comprehensive Enterprise DW(**EDW**) capabilities.
3. **Combined Approach**



# 資料倉儲結構設計

由上而下法(Top-Down)  
由下而上法(Bottom-Up)  
並行法(Combined Approach)



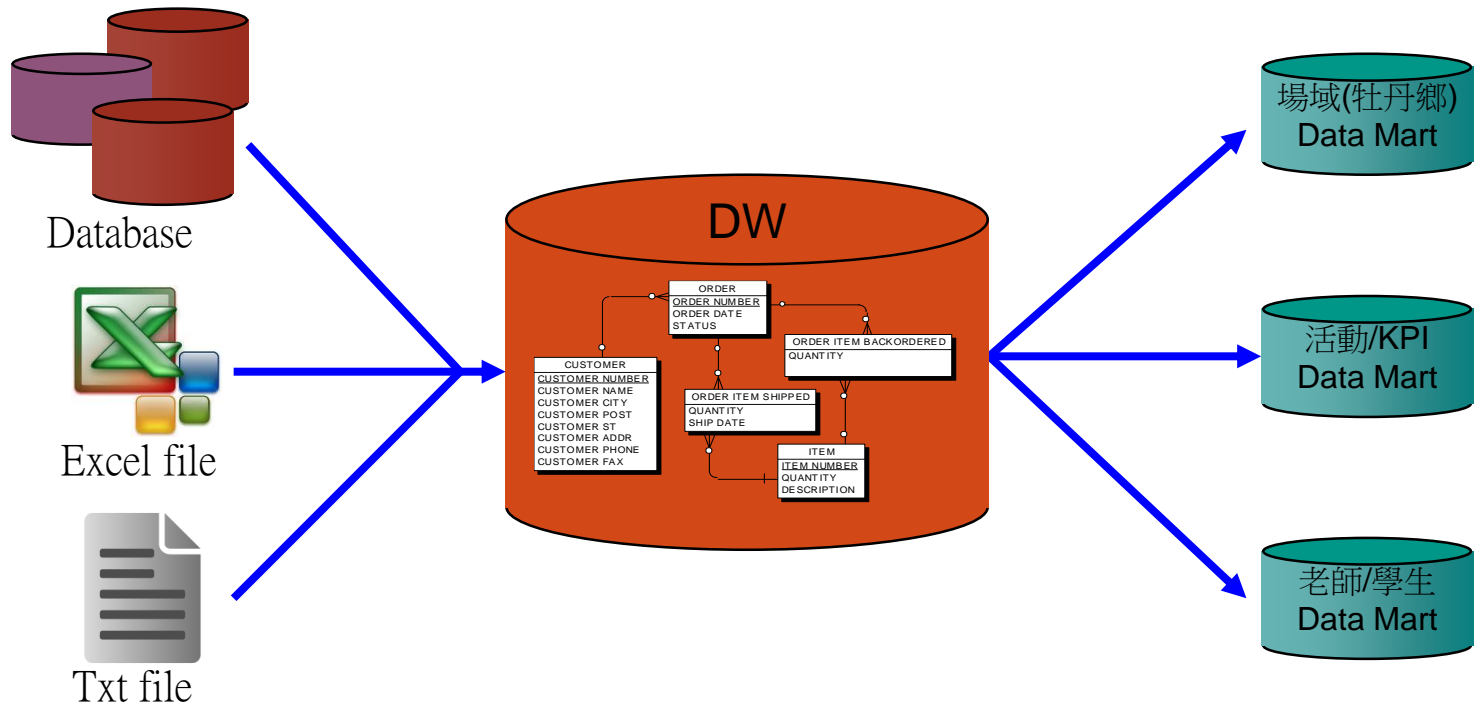
# 資料倉儲結構設計

- 由上而下(Top-Down)

以整體需求設計規劃，資料倉儲使用正規化的實體關聯模式(E-R Model)。收集傳統操作系統資料，經過Extract/Clean/Transform 機制，將資料Load載入資料倉儲中，再根據各部門決策需求複製到資料超市中。

優點：資料具完整性、一致性；

缺點：缺乏彈性，先建資料倉儲涉及整合企業異質性資料，需要花費更多的人力、財力與時間來建置。



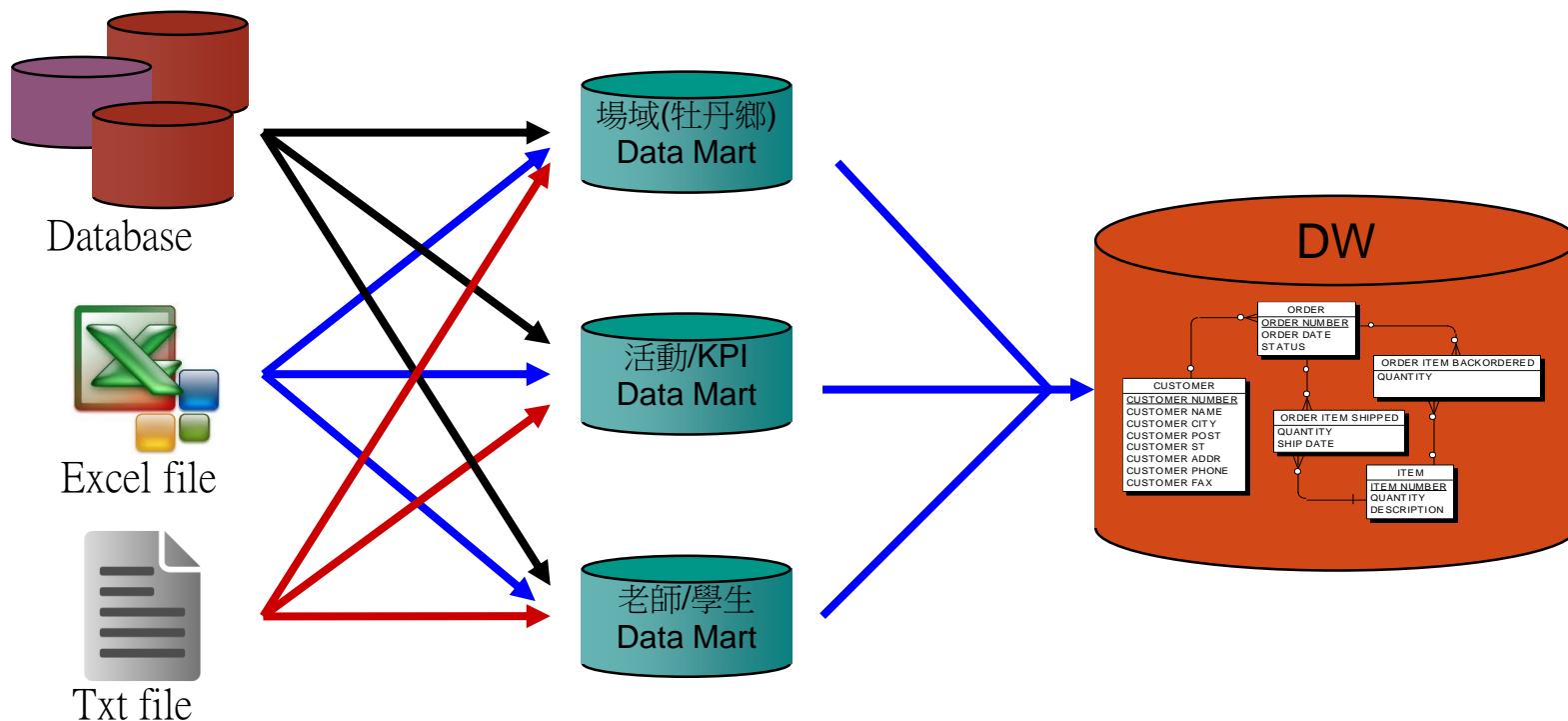
# 資料倉儲結構設計

- 由下而上法(Bottom-Up)

收集日常作業資料，資料由各部門自行Extract/Clean/Transform機制，將資料Load載入資料超市中，再將資料透過複製的機制上載、彙集至資料倉儲中。

優點：依特定需求建置資料超市，簡單快速低成本、較有彈性。

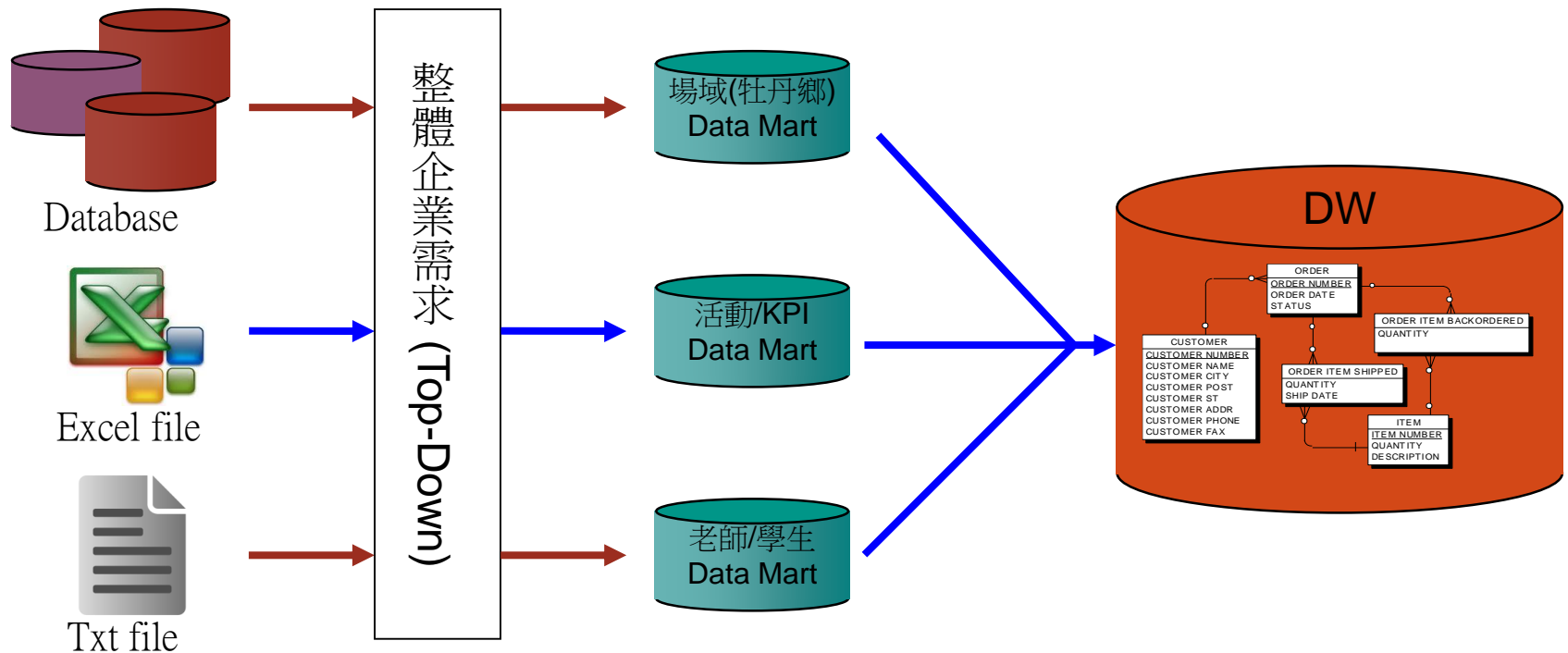
缺點：不同的異質資料來源，經資料超市分別收集再整合至資料倉儲，可能會有不一致狀況，獨立性資料超市並不包含詮釋資料，使得資料倉儲整合困難。



# 資料倉儲結構設計

- 並行法(Combined Approach)

同時使用”由上而下”及”由下而上”之優點，先就企業整體需求及資料模式進行規劃，再開發各部門資料超市，並將資料彙集至資料倉儲中。

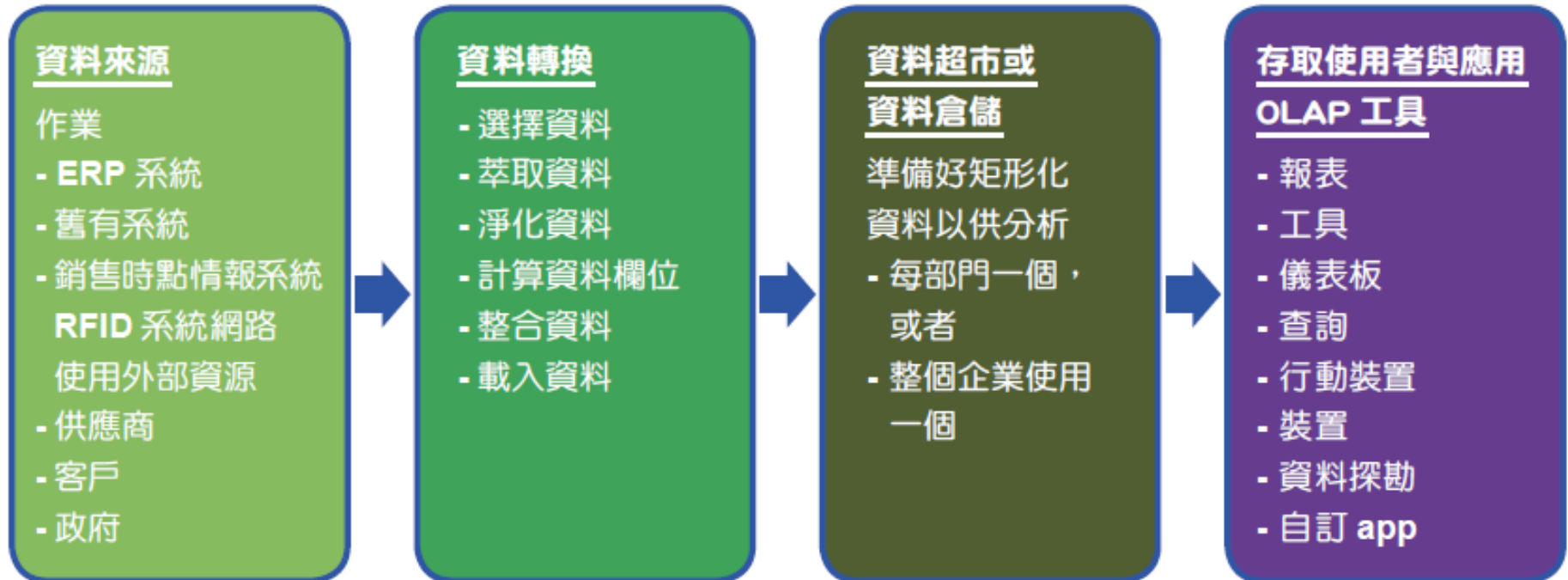


# DW 開發方法

▼ 表 3.1：資料超市與資料倉儲的比較

	功能性資料超市	企業資料倉儲
範圍	單一主題或功能性領域	完整企業資料需求
價值	功能性領域報表與見解	連接多重功能性領域的更深入見解
目標組織	去中心化管理	集中管理
時間	低至中	高
成本	低	高
規模	小至中	中至大
方法	由下而上	由上而下
複雜度	低（較少資料轉換）	高（資料標準化）
技術	較小規模的伺服器與資料庫	產業強度

# DW Architecture



▲ 圖 3.1：資料倉儲架構

## Four key elements:

**data sources** that provide the **raw data** / **data transform** / methods of regularly and accurately data loading of that data into **EDW or data marts** / **data access and analysis**

# 資料載入流程 Data Transform/Loading



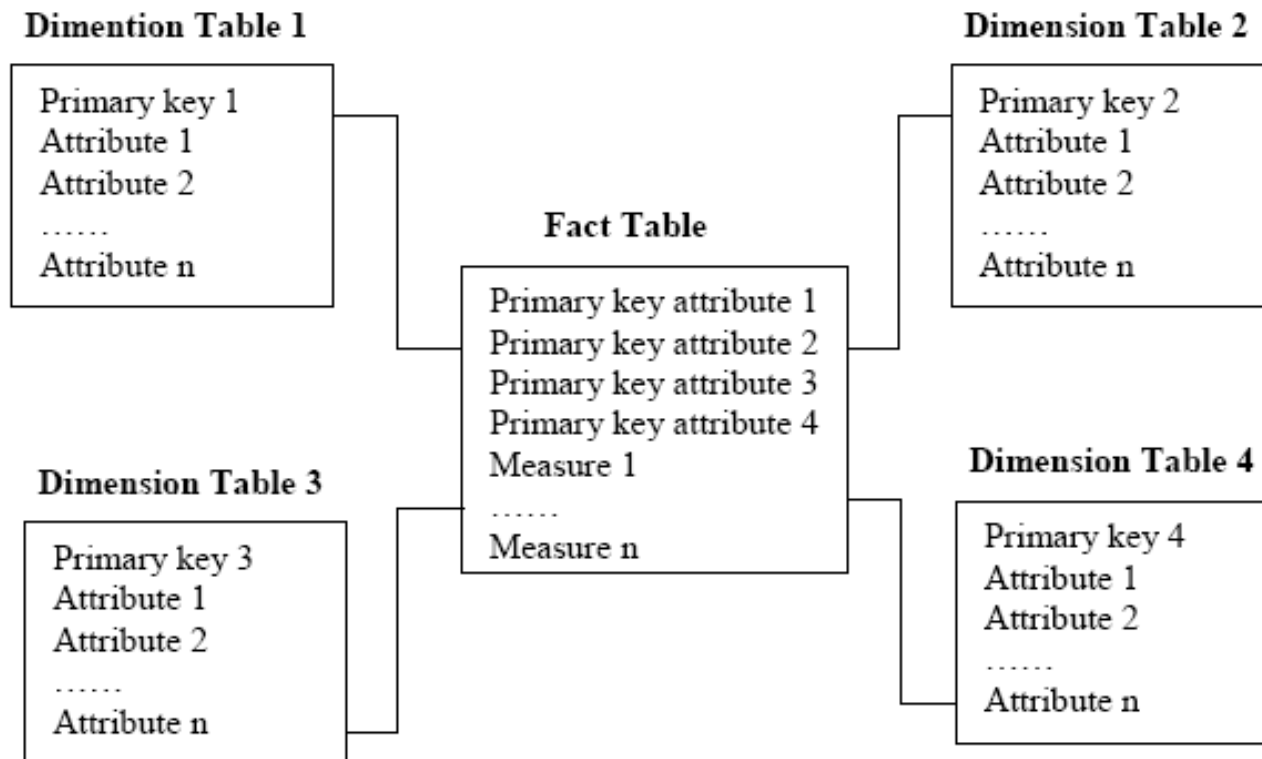
The heart of a useful DW is the processes to populate the DW with good quality data. This is called the **Extract-Transform-Load (ETL) cycle**.

提取-轉換-載入 循環

1. Regular extraction from the operational database sources (定期從操作 ( 交易 ) 資料庫來源，以及其他應用中取得)
2. Data aligned with key fields and integration (萃取的資料應該配合關鍵欄位，並整合至單一資料集中)
3. Loading calculated and transformed data (經轉換的資料接著便會載入DW)

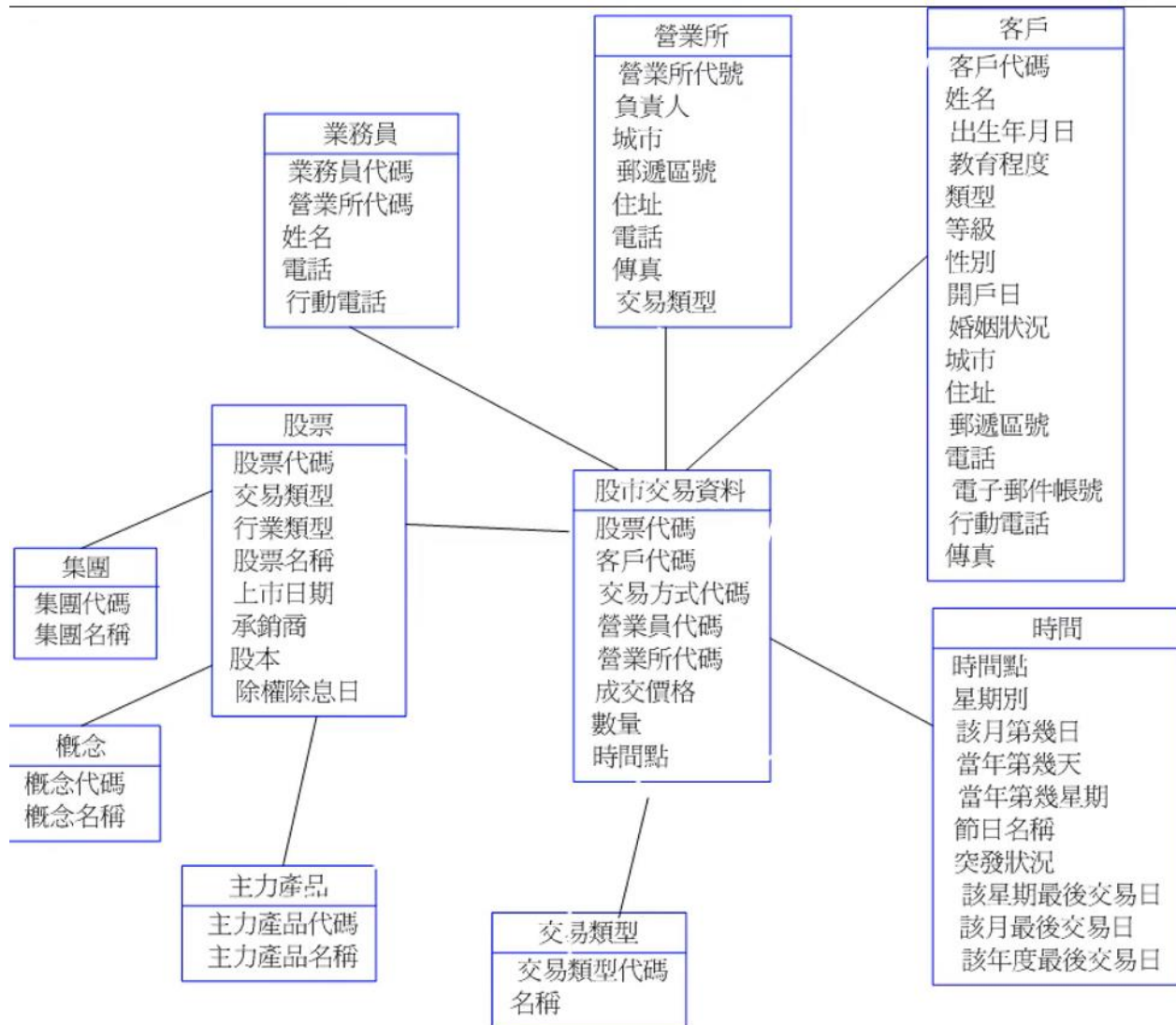
# 資料倉儲的資料模式 (Data model)

- 資料倉儲的資料模式都是多維度模式，包含星狀綱要（Star Schema）、雪花狀綱要（Snowflake Schema）及事實星座綱要（Fact Constellation Schema）。
- 一個完整的多維度模式包含一個以上的事實表格（Fact Table）及多個維度表格（Dimension Table）。



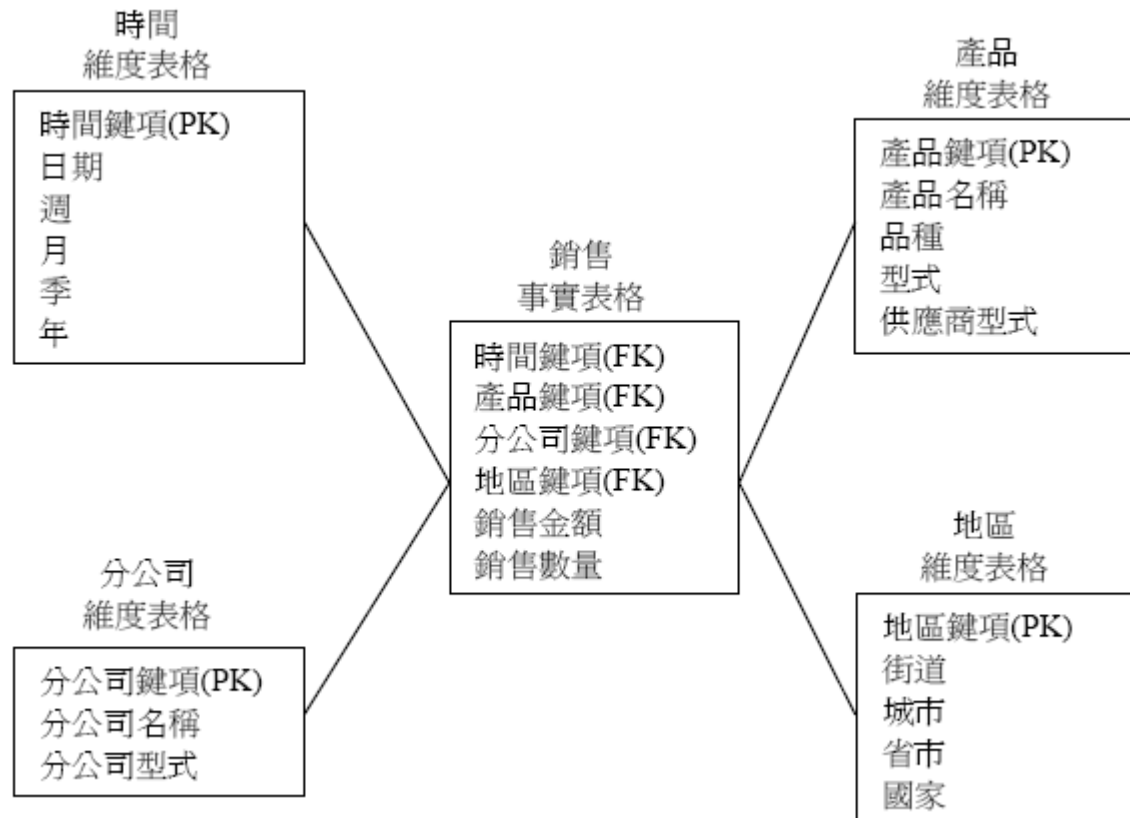


# 股票交易資料範例



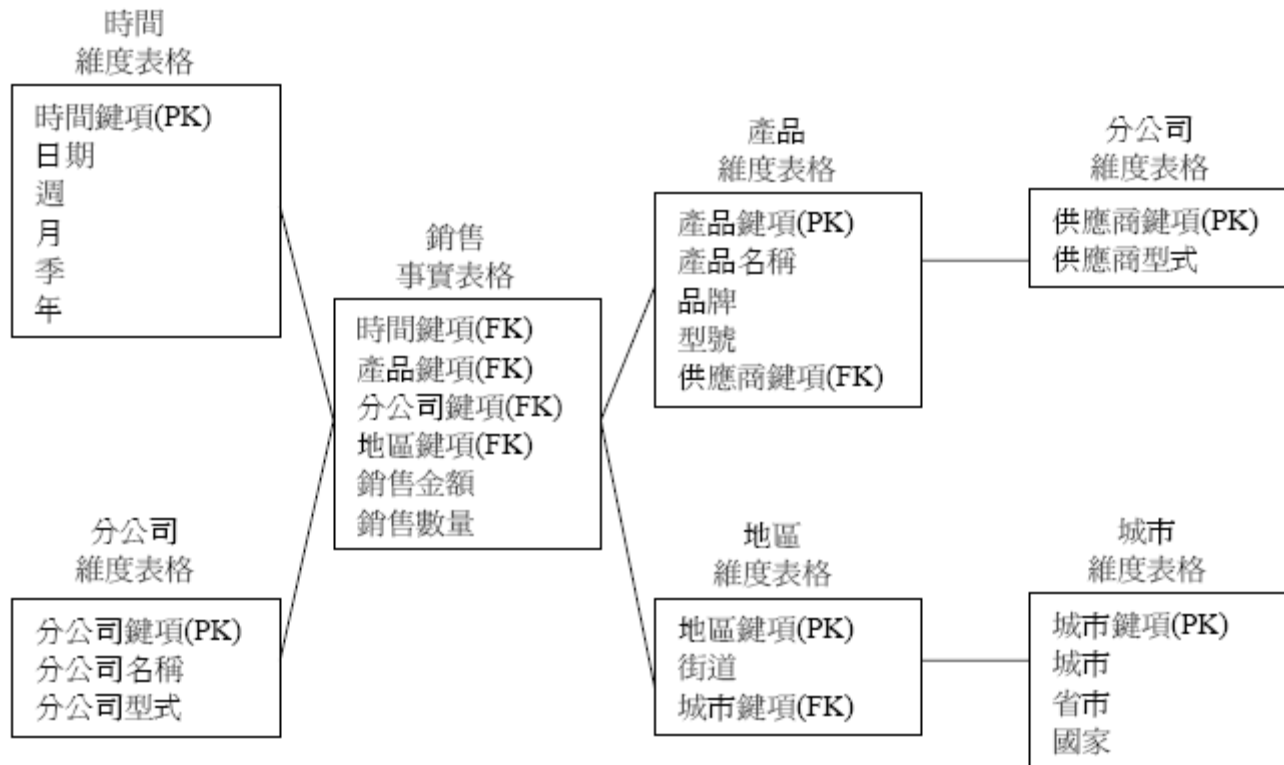
# 資料倉儲的資料模式

- **星狀綱要**：一個中心表格（事實表格）有大量不重複的資料，以及較小的附屬表格（維度表格）。（最常被使用）



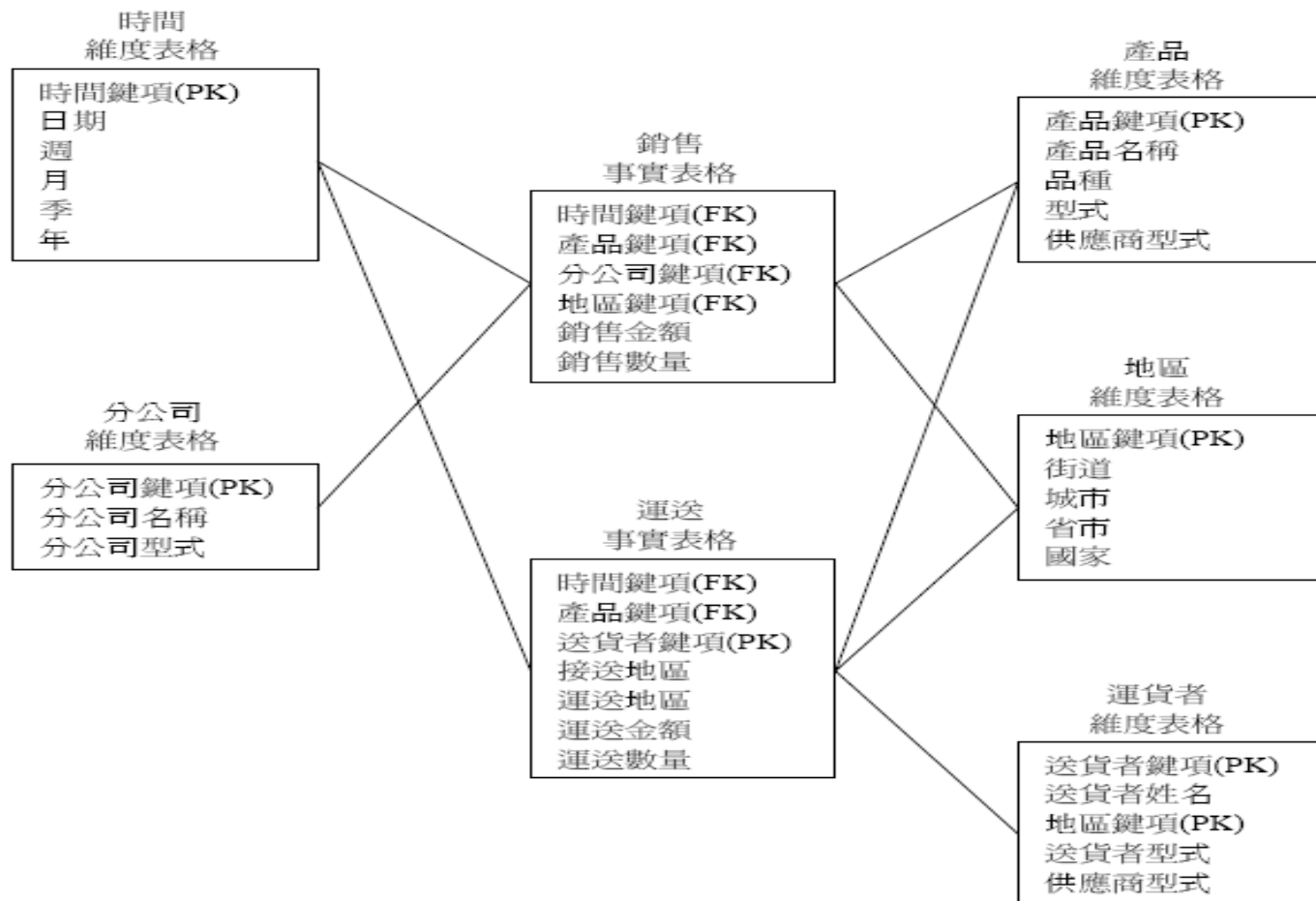
# 資料倉儲的資料模式

- **雪花狀綱要：**雪花狀綱要是星狀綱要的變形，部份維度表格經正規化後，進而分裂成新維度表格。雪花綱要模式與星狀綱要模式最主要的差異，在於雪花模式的維度表格被正規化，以減少重複、容易維護及節省儲存空間。

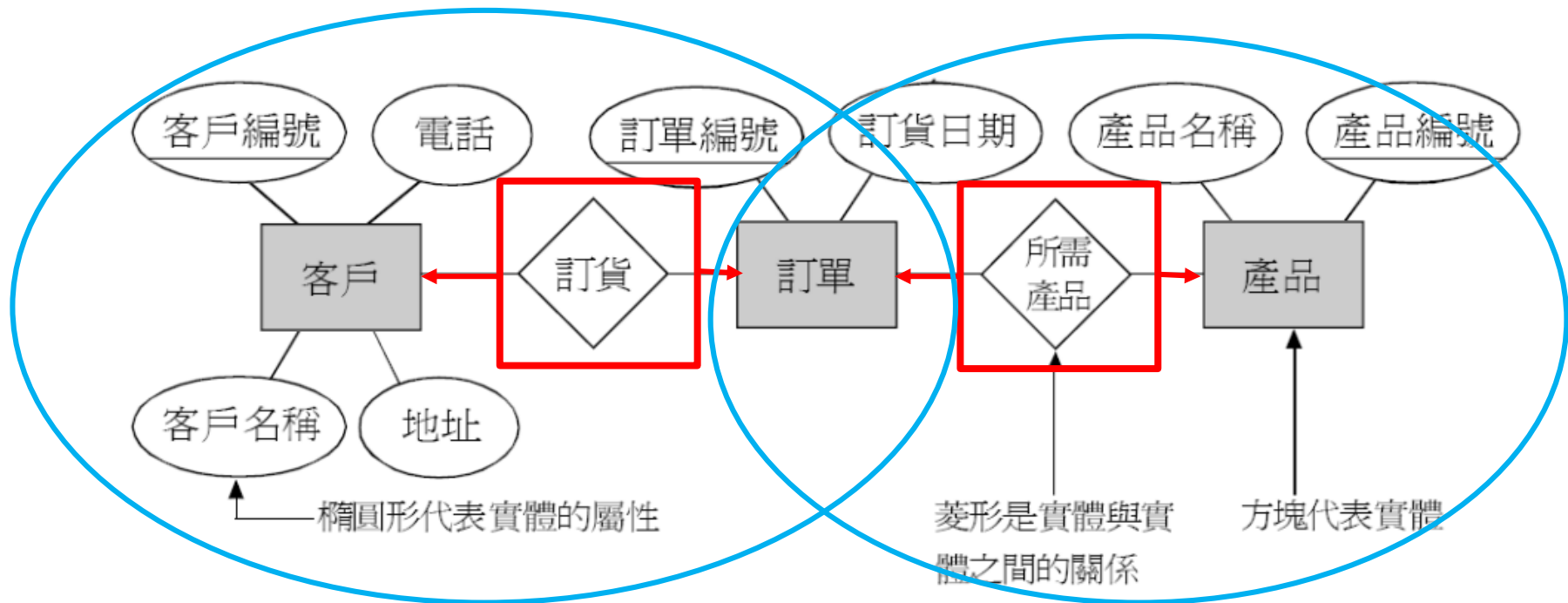


# 資料倉儲的資料模式

- 事實星座網要：以多個事實表格共用維度表格，使用到多個事實表格及需要不同階層的彙總資料。



# DB ER Diagram to DW schema

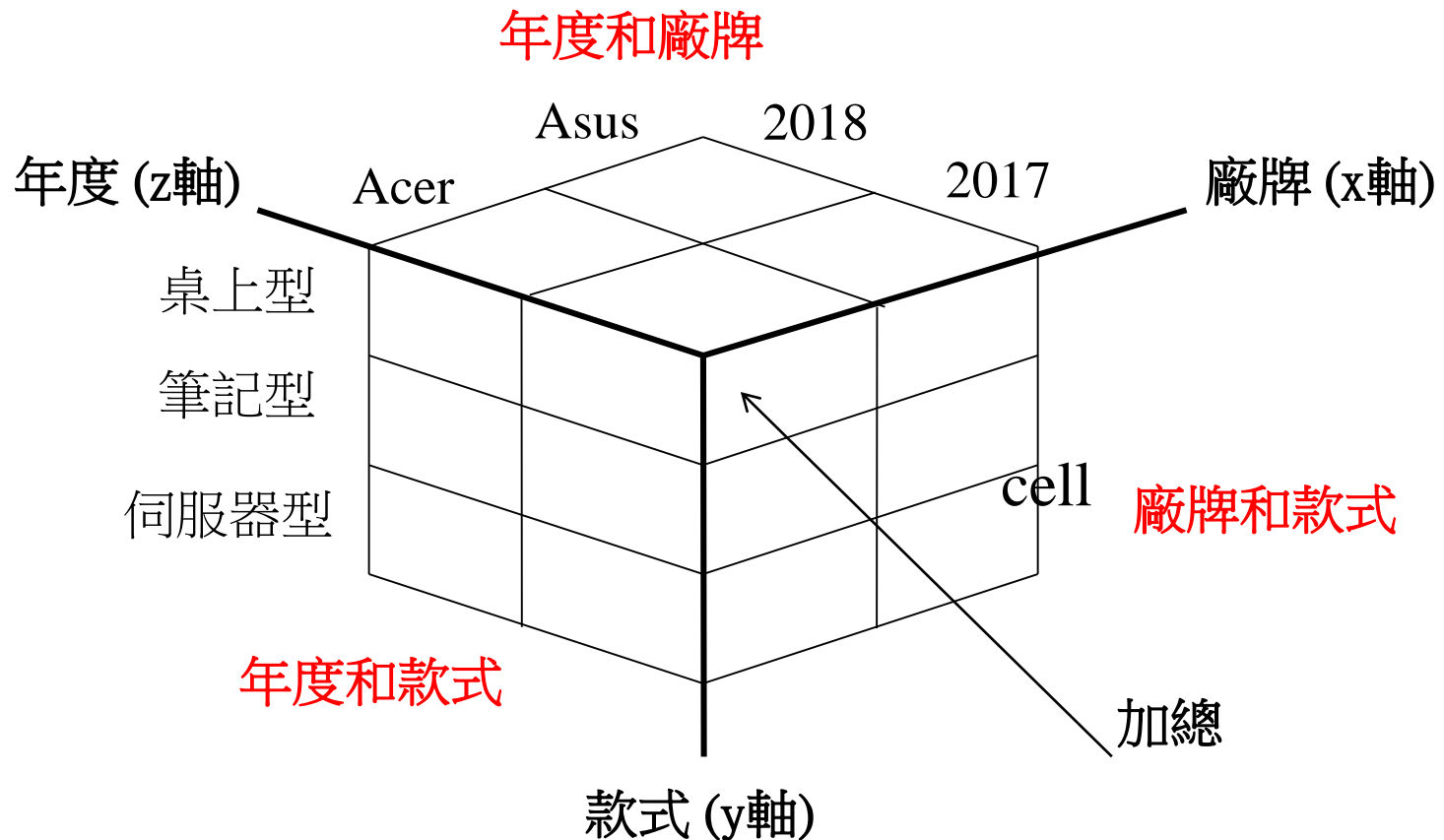


# 資料倉儲的儲存架構

- 一般常用的儲存架構在邏輯設計上稱為多維度資料庫結構（ multidimensional database structure ），但是實際的儲存實體結構可以是**關聯式資料庫**，或是**多維度資料立方體**（ multidimensional data cube ），或是兩者混合的結構。
- 存放在多維度資料庫結構的資料稱為**資料立方體**（ data cube ），它是由**維度**（ dimensions ）與**事實**（ facts ）組合而成。資料立方體提供了資料的多維度觀察，並允許事先計算好彙總值以便將來快速存取彙總的資料。

# Data Cube 的定義

- 是一種多維資料模型
- 將經常會被查詢之資料，事先加以運算、彙總與儲存
- 以立體多維資料結構，提供快速線上查詢與分析
- 呈現範例

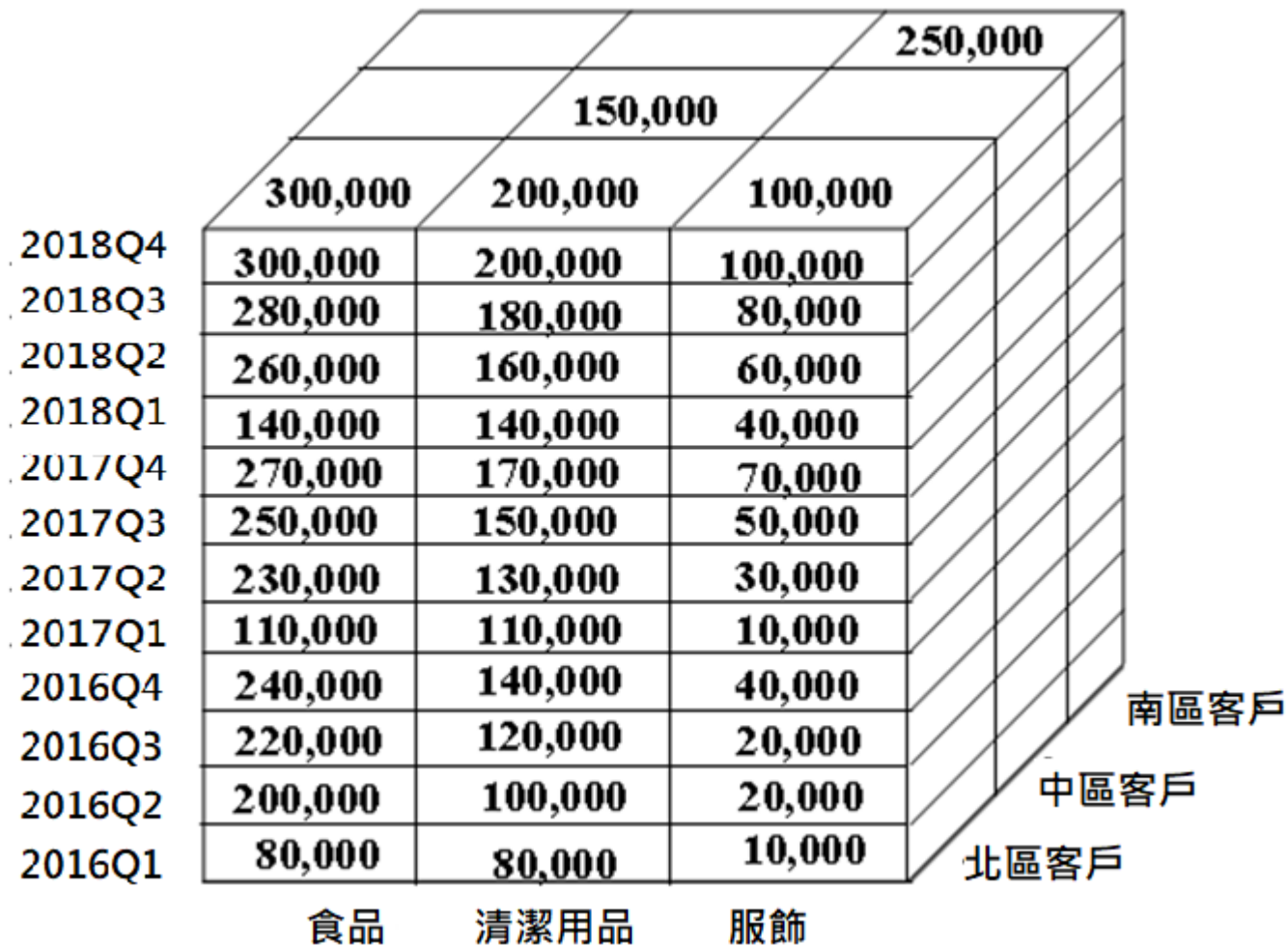


# 練習範例

- 假設一個購物網站預建置一個資料倉儲系統，主要作為**銷售貨品金額**的分析
  - 資料紀錄**時間**範圍從 2016 ~ 2018 總共三年。
  - 將客戶依照**地理區域**分為北、中、南三區，北部包含台北、桃園與新竹；中部包含台中、彰化與南投；南部包含嘉義、台南與高雄。
  - 販售**商品**分成服飾、食品、清潔用品三大類；服飾分成女裝、男裝及童裝三類；食品分成蔬菜、肉類及點心；清潔用品分成客廳、廚房、衛浴相關清潔用品。



# 多維度立方體示意圖



# 多維度立方體示意圖說明

- 此Data Cube總共有三個維度，垂直的維度代表時間、橫軸的維度代表商品類別，最後一個客戶區域維度代表消費者所在的區域與縣市。
- 維度上的單位或是階層將立方體劃分成許多小單元（cell），例如以年、季、月來刻畫時間維度，而三者又有階層關係，每個小單元則存放某個彙總的**量值（measure）**，在本例中所存放的量值就是**銷售金額**。例如左上角的單元，代表2018年第四季，食品賣給北部客戶總共金額是300,000元，它被存放在（2018 Q4，食品，北部）的座標單元之內。

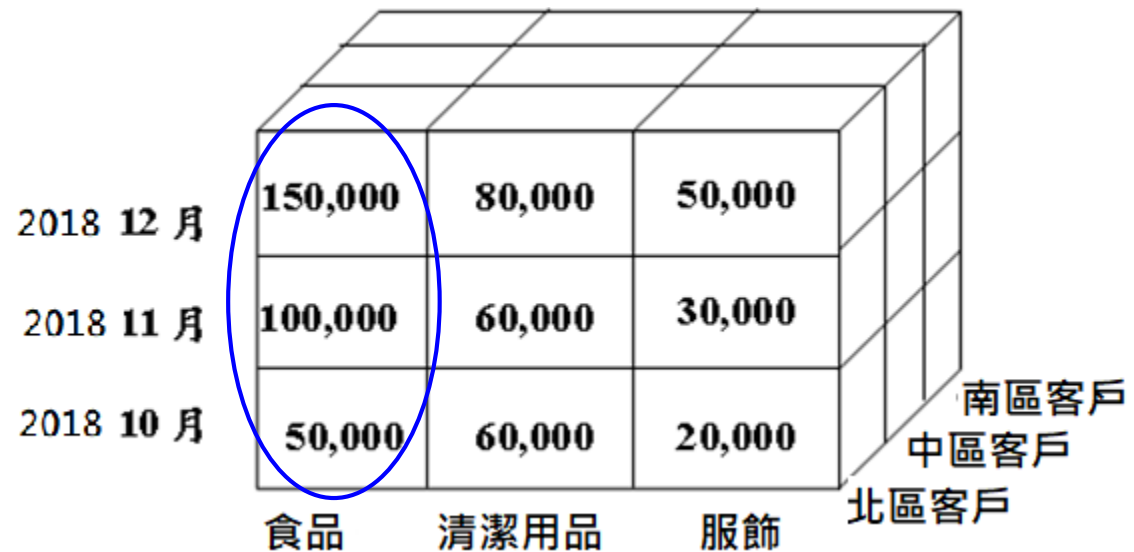
# 資料倉儲的OLAP操作方法

- 刻畫維度的單位可以設計成有階層關係存在，稱之為**概念階層**（concept hierarchy）。例如在時間維度，階層關係是年、季、月、日；商品維度的階層關係是商品類別、商品次類別、商品。
- 較高的階層可以包含數個層次較低的階層，可以利用類似拉近、拉遠（zoom in/out）的方式快速瀏覽各階層的彙總資料。而較高層次的彙總資料則可以由較低的階層之彙總資料快速組合而成，這也是資料倉儲可以快速回答查詢的原因之一。

# 資料倉儲的操作方法(下鑽)

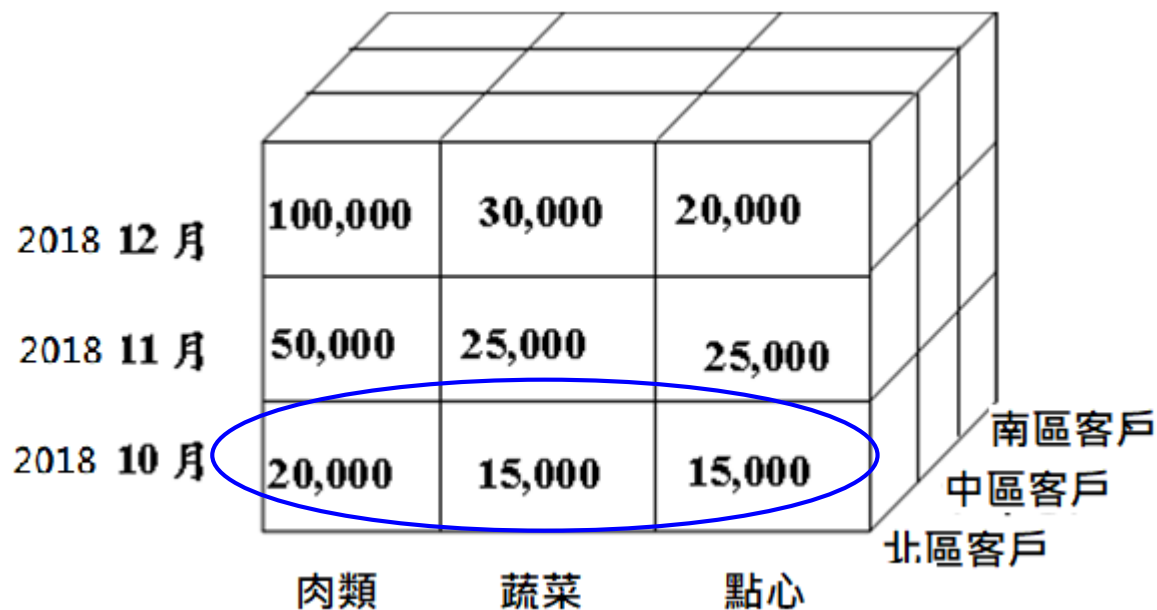
- 下鑽(drill-down)方法讓使用者可以更深入一層瀏覽彙總資料。

➤ 例一：想要進一步瞭解 2018 Q4 內每個月的銷售情況，則可以在時間維度利用下鑽操作，看到 2018 年 10 ~ 12 月的銷售金額。



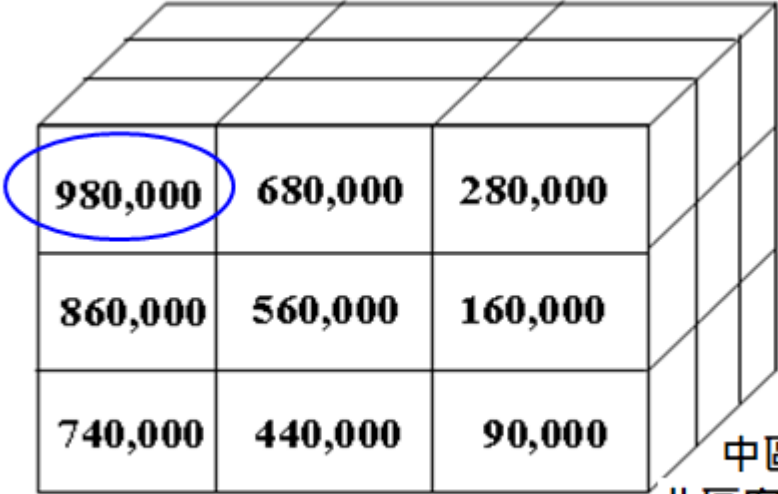
# 資料倉儲的操作方法(下鑽)

- 例二：在商品維度使用下鑽操作，去細看食品類別裡的蔬菜、肉類及點心三個小類分別的銷售金額。



# 資料倉儲的操作方法

- **上捲(roll-up)**方法讓使用者提高觀看的層次，去瞭解更概觀的情況。
  - 範例：想知道每年的銷售金額（不需要細分到季別），透過時間維度的上捲操作，將看到更高層級的彙總資料。

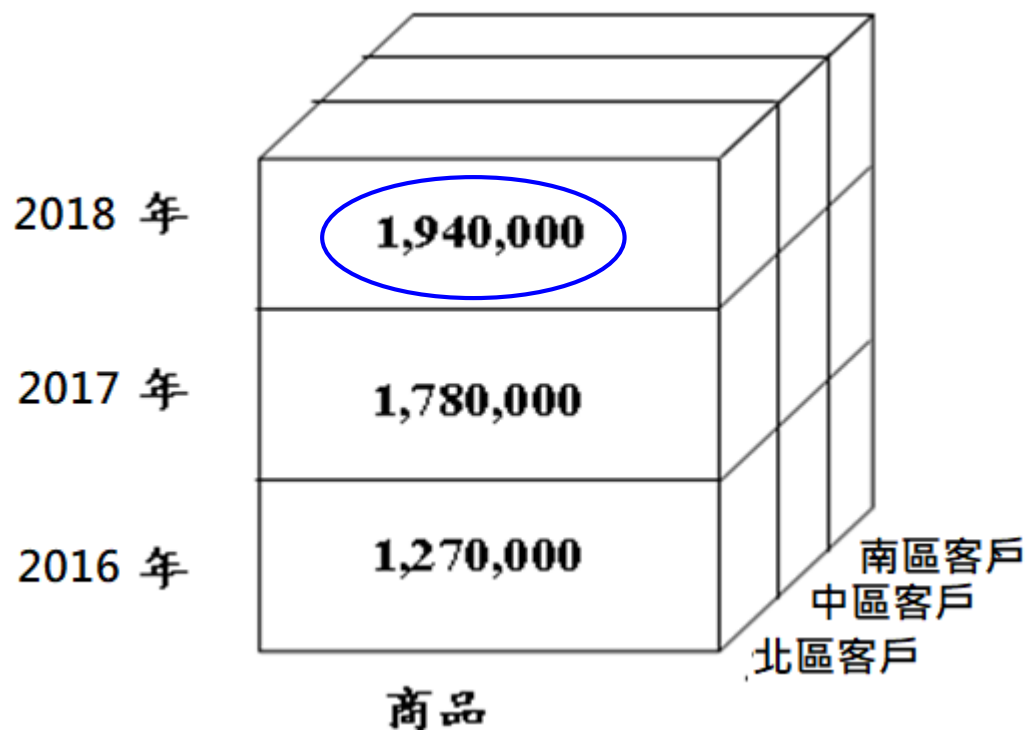


2018 年	980,000	680,000	280,000
2017 年	860,000	560,000	160,000
2016 年	740,000	440,000	90,000
	食品	清潔用品	服飾

南區客戶  
中區客戶  
北區客戶

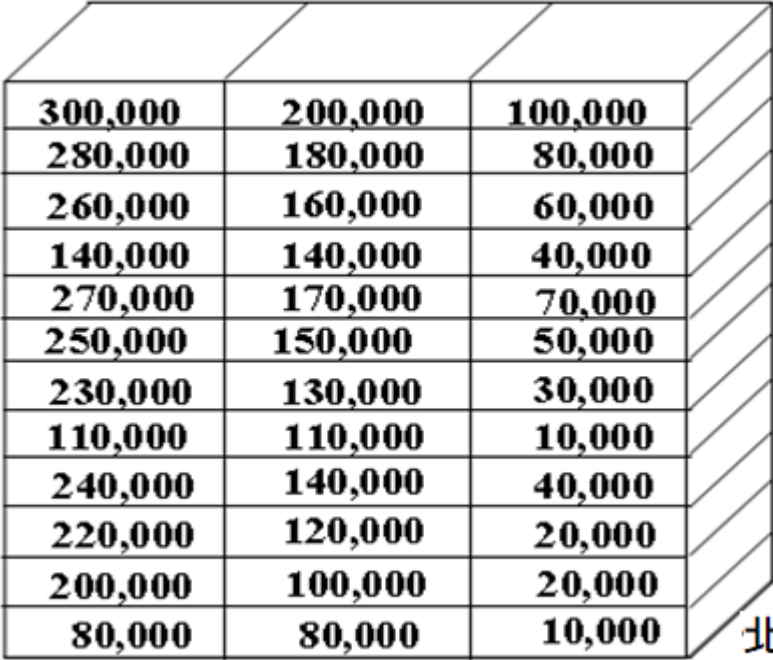
# 資料倉儲的操作方法(上捲)

- 例二：只希望看到每年北、中、南三地區客戶購買的總銷售金額（不需要細分商品類別），同樣利用商品維度的上捲操作，將看到更高一層的統計資料。



# 資料倉儲的操作方法(切片)

- 切片(slice) 是在單一維度上進行條件設定與資料選擇，進而產生出一個子立方體，讓使用者能夠切割某一層面的資料。
  - 範例：主管只想看北部客戶的購買情況，就可以利用切片操作，將只包含北部的資料切割出來。



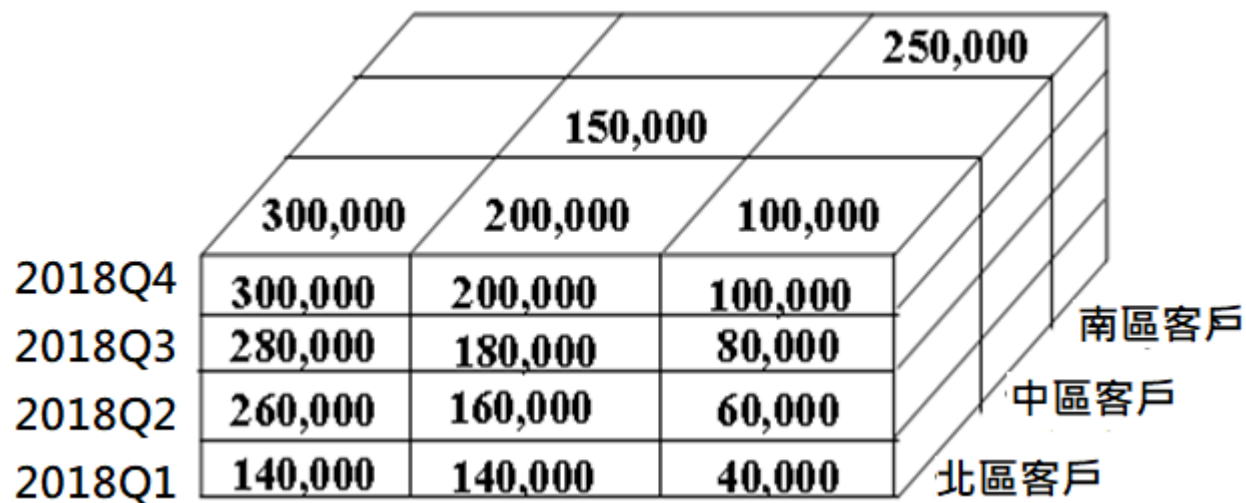
2018Q4	300,000	200,000	100,000
2018Q3	280,000	180,000	80,000
2018Q2	260,000	160,000	60,000
2018Q1	140,000	140,000	40,000
2017Q4	270,000	170,000	70,000
2017Q3	250,000	150,000	50,000
2017Q2	230,000	130,000	30,000
2017Q1	110,000	110,000	10,000
2016Q4	240,000	140,000	40,000
2016Q3	220,000	120,000	20,000
2016Q2	200,000	100,000	20,000
2016Q1	80,000	80,000	10,000
	食品	清潔用品	服飾

北區客戶



# 資料倉儲的操作方法(切片)

- 範例二：只想看最近一年 2018 的資料，同樣利用切片操作，切割出 2018 年的資料。



# 資料倉儲的操作方法(切塊)

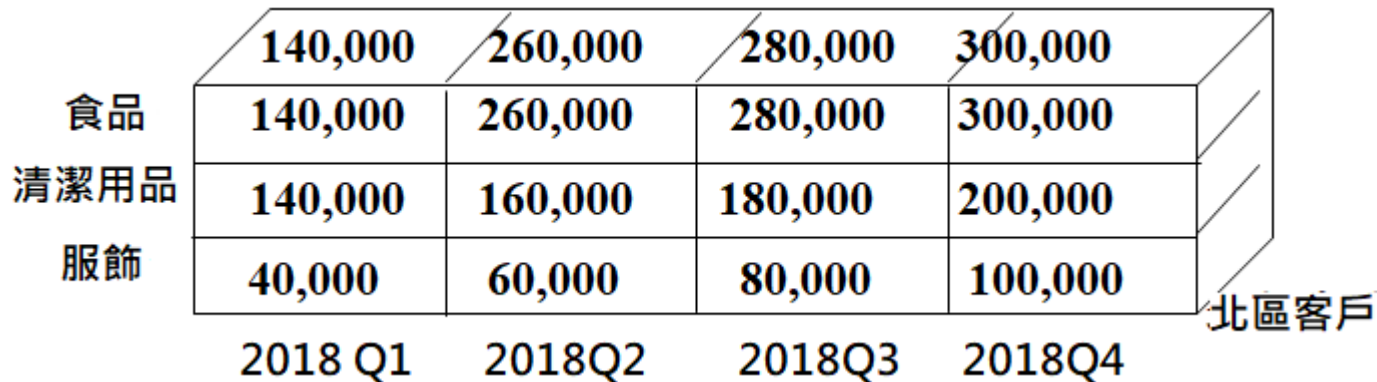
- **切塊 ( dice )** 方法是對多個維度進行條件設定的資料選擇，進而產生一個子立方體。
  - 例如主管只想看北部客戶在2018 年的購買情況，則可以利用客戶區域維度與時間維度切塊方式切出子立方體。

	300,000	200,000	100,000
2018Q4	300,000	200,000	100,000
2018Q3	280,000	180,000	80,000
2018Q2	260,000	160,000	60,000
2018Q1	140,000	140,000	40,000
	食品	清潔用品	服飾

北區客戶

# 資料倉儲的操作方法(轉軸)

- **轉軸 ( pivot )** ( 又稱旋轉 : **rotate** ) 讓使用者可以轉動 2D 切片或是 3D 的立方體，從不同的視角來觀看資料。
- 例如將前頁圖之子立方體做一 90度的旋轉。



	140,000	260,000	280,000	300,000
食品	140,000	260,000	280,000	300,000
清潔用品	140,000	160,000	180,000	200,000
服飾	40,000	60,000	80,000	100,000
	2018 Q1	2018Q2	2018Q3	2018Q4

北區客戶

# 資料倉儲的查詢處理

- **資料倉儲**基本上是建立在一個**多維度資料庫結構**上的一個儲存體，而多維度資料分析的核心是如何有**效率的計算出多個維度集合上的統計值**（例如2018年每一季，每個區域服飾的銷售金額），以支援快速查詢之用。
- 若是使用一般關聯式資料庫，就是利用「group by」語法來求得這些統計值（group by 季、分店、服飾），只是SQL的group by指令是在執行時，才即時計算結果，當資料量大需要等待上一段時間。

# 資料倉儲的查詢處理

- 前範例的多維度結構包含時間、商品、區域三個維度與一個**銷售金額**量值，使用者可能使用下面任何一種的查詢方式來分析資料：
  - (1) 依照時間、商品、區域 分組來計算銷售總金額（形成時間、商品、區域三維度的立方體）
  - (2) 依照時間、商品分組來計算銷售總金額（形成時間、商品二維度的立方體）
  - (3) 依照商品、區域分組來計算銷售總金額（形成商品、區域二維度的立方體）
  - (4) 依照時間、區域分組來計算銷售總金額（形成時間、區域二維度的立方體）

# 資料倉儲的查詢處理

- (5) 依照時間分組來計算銷售總金額（形成時間一維度的立方體）
- (6) 依照商品分組來計算銷售總金額（形成商品一維度的立方體）
- (7) 依照區域分組來計算銷售總金額（形成區域一維度的立方體）
- (8) 所有銷售總金額（形成零維度的立方體）

# Multi-Dimension eXpression (MDX)

- MDX query from CUBE

```
SELECT column_set0 ON AXIS(0) [, column_set1 on AXIS(1),...]
FROM cube_name
[WHERE (member-of-dim0, member-of-dim1,..., member-of-dimn) ]
```

```
SELECT column_set0 ON COLUMNS, column_set1 on ROWS, ...
FROM cube
[WHERE (member-of-dim0, member-of-dim1,..., member-of-dimn) ]
```

- Columns / Rows / Pages / Chapters / Sections
- 在 MDX 中，SELECT 陳述式會指定一個結果集，內含從 Cube 傳回的多維度資料子集。若要指定結果集，MDX 查詢必須包含下列資訊：
- 結果集包含的軸數目。在一個 MDX 查詢中，您最多可指定 128 個座標軸。
- 要在 MDX 查詢的每一個軸上包含的成員或 Tuple 集合。
- 設定 MDX 查詢內容的 Cube 的名稱。

# 資料倉儲的查詢處理

- 為快速回答查詢，可以將所有可能組合查詢的彙總資料都事先計算並加以儲存，計算的過程可以由下而上聚集而成。
- 若是記憶空間足夠，可以事先將以上所列的3D~0D資料立方體的彙總資料，全部事先算出並加以儲存，若是空間不夠，則可以只計算部分彙總資料，遇到查詢時，再即時由下層資料立方體以聚集方式快速算出。
- 一般關聯式資料庫遇到每一次查詢，都必須一筆一筆慢慢累積計算，在效能上自然無法跟已經事先計算出彙總值的多維度資料庫結構相比。



# DW 最佳實作



A data warehousing project reflects a significant investment into information technology (IT). All of the best practices in implementing any IT project should be followed.

1. The DW project should align with the **corporate strategy** (DW 專案應該與企業策略保持一致) Business professionals + IT for objectives of financial viability. DW design should be carefully tested before beginning development work.
2. It is important to manage user expectations. The data warehouse should be built incrementally.
3. **Quality** and **adaptability** should be built in from the start. Only relevant, cleaned, and high-quality data should be loaded.