Lecture 2：關聯式資料庫(Relational Database)概念與應用

-- 掌握 AIS 的底層邏輯

編撰：國立屏東大學 周國華老師 **日期：**2026-02-25

一、課程導論與學習地圖

- **回顧上週：**AIS 簡介。
- **本週重點：**資料庫基礎 (Database Fundamentals)。這是 AIS 的底層邏輯與地基。
- **下週預告：**進入記憶體運算時代的 SAP HANA 資料庫技術。

前言（為什麼會計人需要懂資料庫？）：

- 在現代商業環境中，會計師與稽核人員不再只是查核紙本憑證，更多時候是在處理「數據」。
- 所有的會計分錄、銷售訂單、採購單，在系統中本質上都是資料庫裡的「紀錄 (Records)」。
- 若不懂資料庫邏輯，就無法有效進行數據分析 (Data Analytics) 或電腦稽核。

二、資料儲存方式的演進與痛點

- **問題思考：**如果在 Excel 裡紀錄全校學生的選課資料，一位學生修了 5 門課，他的學號、姓名與系級可能要重複輸入多次。這引出了資料冗餘 (Redundancy) 與維護困難的問題。
- **傳統檔案系統 (File System)：**
  - 早期資料分散在不同檔案中（如：業務部有自己的 Excel 客戶名單，會計部也有自己的名單）。
  - **缺點：**各自為政、資料重複 (Data Redundancy)、資料不一致 (Data Inconsistency)、版本不一且缺乏整合。
- **資料庫管理系統 (DBMS)：**(DBMS 產品排名請參閱次頁)
  - 集中管理數據的軟體系統（如：Oracle, Microsoft SQL Server, MySQL, SAP HANA）。
  - **優點：**集中管理、單一真實來源 (Single Source of Truth)、數據共享、減少冗餘並提升安全性與完整性。

DB-Engines 的 DBMS 排名 (2026-02 資料)

429 systems in ranking, February 2026

Rank			DBMS	Database Model	Score		
Feb 2026	Jan 2026	Feb 2025			Feb 2026	Jan 2026	Feb 2025
1.	1.	1.	Oracle	Relational, Multi-model	1203.51	-33.82	-51.31
2.	2.	2.	MySQL	Relational, Multi-model	868.22	+0.70	-131.77
3.	3.	3.	Microsoft SQL Server	Relational, Multi-model	708.14	+1.88	-78.73
4.	4.	4.	PostgreSQL	Relational, Multi-model	672.03	+5.77	+12.42
5.	5.	5.	MongoDB	Document, Multi-model	378.73	+1.99	-17.90
6.	6.	↑ 7.	Snowflake	Relational	208.14	+0.34	+52.56
7.	7.	↓ 6.	Redis	Key-value, Multi-model	147.04	+2.88	-10.87
8.	8.	↑ 13.	Databricks	Multi-model	144.51	+2.97	+54.48
9.	9.	9.	IBM Db2	Relational, Multi-model	111.22	-1.50	-14.21
10.	10.	↓ 8.	Elasticsearch	Multi-model	106.46	-0.69	-28.17
11.	11.	11.	Apache Cassandra	Wide column, Multi-model	101.60	+0.76	-0.98
12.	12.	↓ 10.	SQLite	Relational	99.19	-1.42	-14.63
13.	13.	↑ 14.	MariaDB	Relational, Multi-model	86.09	-1.64	-3.42
14.	↑ 15.	↑ 17.	Microsoft Azure SQL Database	Relational, Multi-model	73.66	-0.53	+0.90
15.	↓ 14.	15.	Splunk	Search engine	73.05	-2.25	-7.51
16.	16.	↑ 18.	Apache Hive	Relational	73.00	-0.48	+10.51
17.	17.	↓ 12.	Microsoft Access	Relational	69.47	-2.18	-27.07
18.	18.	↓ 16.	Amazon DynamoDB	Multi-model	67.87	-1.92	-7.71
19.	19.	19.	Google BigQuery	Relational	59.58	-4.02	+4.69
20.	20.	20.	Neo4j	Graph	49.26	-0.89	+3.92
21.	21.	↑ 24.	Apache Solr	Search engine, Multi-model	33.26	-0.84	+0.58
22.	22.	↑ 23.	Teradata	Relational, Multi-model	32.44	-1.25	-1.44
23.	23.	↓ 22.	SAP HANA	Relational, Multi-model	31.75	-0.72	-3.47
24.	24.	↓ 21.	FileMaker	Relational	30.19	-1.04	-10.03
25.	25.	25.	SAP Adaptive Server	Relational, Multi-model	26.18	-0.60	-2.44
26.	26.	↑ 27.	Microsoft Azure Cosmos DB	Multi-model	23.88	-0.92	+1.75

三、關聯式資料庫 (RDBMS) 核心要素與構造

這是目前商業世界最主流的資料庫模型。

- **資料表 (Table / Relation)：**
  - 資料庫的基本儲存單位，由「列」與「欄」組成，就像一個 Excel 工作表。
  - **實例：**在 SAP 中，存放會計憑證抬頭的表叫做 BKPF。

操作：在 t-code 欄輸入 **SE16N**，在 General Table Display 的 Table 欄中輸入 **BKPF**，在下方的 Selection Criteria 內輸入適當的篩選值，例如在 Company Code 的 From Value 及 To Value 中都輸入 **US00**，再按 **F8**。

- **紀錄 (Row / Record / Tuple)：**
  - 水平方向，表格中的每一列代表一筆完整的資料。
  - **實例：**發票資料表中一張特定發票的資訊。(Google 查詢：**營業人使用三聯式統一發票明細表**)
- **欄位 (Column / Field / Attribute)：**
  - 垂直方向，表格中的每一欄代表資料的屬性或類別。
  - **實例：** 發票號碼(2 位英文字母+8 位數字流水號)、客戶統一編號、銷售額、課稅別、稅額、備註。
- **主鍵 (Primary Key, PK, 又稱 主索引)：**
  - **定義：**能「唯一」識別表格中每一列資料的識別碼 (Unique Identifier)。
  - **原則與重要性：**不可重複、不可為空值 (Null)，確保資料不重複。
  - **實例：**身分證字號、學號、員工編號、訂單編號。DBMS 中通常會使用企業自身資訊系統生成的編號作為資料表的主鍵，例如，用**員工編號**作為員工資料表的主鍵，而非員工**身分證字號**；用**學號**作為學生資料表的主鍵，而非學生**身分證字號。**
  - **SAP 實例：**KUNNR (客戶編號)，MATNR (物料編號)。

    操作 1 (查詢客戶主檔 KNA1)：在 t-code 欄中輸入 **/n SE11**，進入 ABAP Dictionary 初始畫面。在 Database table 欄中輸入 **KNA1**，按左下方的 **Display**，在 key 欄中可看到 MANDT (用戶碼，屏大是 300)及 KUNNR (客戶編號)這兩欄被勾選做為複合主索引。

    操作 2 (查詢物料主檔 MARA)：按上方的**返回鍵 Back**，回到 ABAP Dictionary 初始畫面。在 Database table 欄中輸入 **MARA**，按左下方的 **Display**，在 key 欄中可看到 MANDT (用戶碼，屏大是 300)及 MATNR (物料編號)這兩欄被勾選做為複合主索引。
- **外來鍵 (Foreign Key, FK)：**
  - **定義與功能：**用來連結另一個表格主鍵的欄位，作為連結兩個表格的橋樑，這也是「關聯 (Relational)」一詞的由來。
  - **SAP 實例：**在 VBAK「銷貨單資料表」中有一個欄位是 KUNNR (客戶編號)，這個欄位就是外來鍵，用來連結到 KNA1「客戶主檔」。

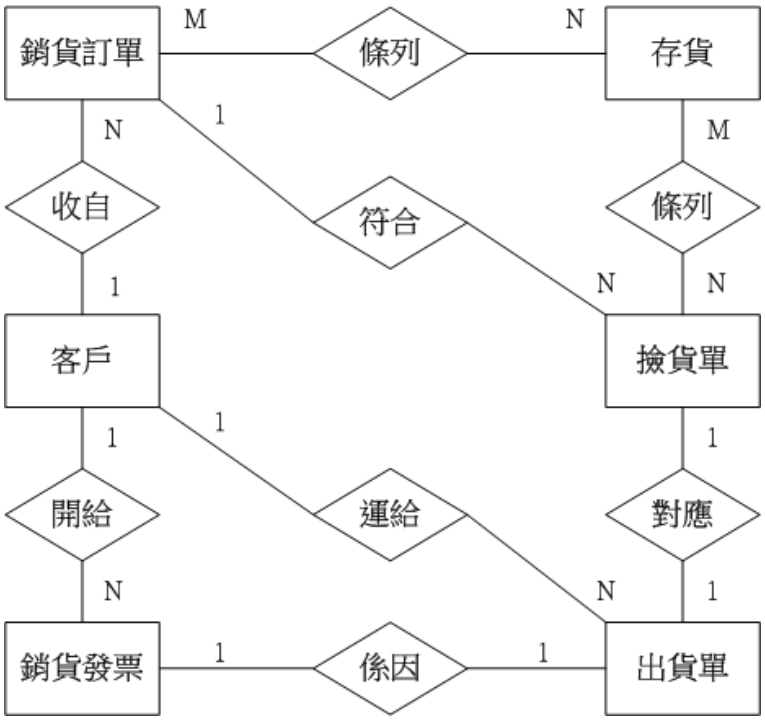
    操作 (查詢銷貨單資料表 VBAK)：按上方的**返回鍵 Back**，回到 ABAP Dictionary 初始畫面。在 Database table 欄中輸入 **VBAK**，按左下方的 **Display**，在 key 欄中可看到

MANDT (用戶碼，屏大是 300)及 VBELN (銷貨單編號)這兩欄被勾選做為複合主索引。往下查找欄位名稱可看到 KUNNR 這個外來鍵，右側描述欄顯示為 Sold-To Party。

四、實體關聯模型 (Entity-Relationship Model, E-R Model) 與正規化

- **實體關聯圖 (E-R Diagram) 簡介：**
  - 在 AIS 中，我們常用 E-R Model 來描述業務流程。根據 E-R Model 所建立的圖形稱為 E-R Diagram (ERD)。ERD 內通常包含 Entity、Relationship、Cardinality 及 Attribute 四個要素。
  - **Entity (實體):** 代表蒐集資料的對象，以長方形表示（例如：學生、課程）。
  - **Relationship (關係):** 兩個實體之間存在的關係，以菱形表示（例如：選修）。
  - **Cardinality (基數性/關係種類):** 通常有三種
    - **一對一 (1:1)：**例如，一位員工只有一個勞保帳號，一個勞保帳號只屬於一位員工。
    - **一對多 (1:N)：**例如，一個客戶可下多張訂單，一張訂單只能來自一個客戶。
    - **多對多 (M:N)：**例如，學生與課程的關係（一個學生可修多門課，一門課有多個學生修讀），在資料庫設計中通常會拆解成兩個 1:N 關係。
  - **Attribute (屬性)**: Entity 及 Relationship 可能有許多屬性，在把 Entity 及 Relationship 轉成資料表時，這些屬性就是資料表內的欄位。在繪製簡化的 ERD 時，通常會把屬性省略。

ERD 範例：銷貨系統



- **將 ERD 轉換成資料表結構**
    - ERD 描繪完成後，可按以下步驟轉成邏輯資料模式：
        - 為每個實體編製一張資料表(稱為 relation 或 table)，ERD 內之屬性即為資料表內的各個欄位，且每張資料表內須包含一個主索引欄位。
        - 在 1 對多 關係中，把 1 方資料表的主索引放在多方資料表內，此欄位稱為多方的外來鍵(foreign key)。表格間的關聯即藉此建立。
        - 在 多對多 關係中，另外增加一個關係資料表，將兩個實體的主索引納入此資料表，成為複合主索引(composite PK)。亦即，將原本的一組多對多關係，轉成兩組 1 對多關係(關係資料表為多方)。
        - 在 1 對 1 關係中，把未來最有可能轉成多方的實體暫時當成多方，然後按照 1 對多關係處理。若難以判定，亦可互將對方的主索引納入成為外來鍵。
    - 例如：若將前述銷貨系統 ERD 範例轉成邏輯模式，需包含 8 張資料表，其中兩張為關係資料表。(見次頁)

圖示：銷貨系統的邏輯資料表模式

客戶

客戶編號	客戶名稱	地址	城市	國家	郵遞區號	收貨人	送貨地址	送貨城市	送貨國家	送貨郵區	信用額度	最後修正	信用條件
A001	上華公司	民和路54號	高雄	中華民國	800	相同	相同	相同	相同	相同	5,000,000.00	20071012	2/10,n/30
A002	永康公司	健康路185號	屏東	中華民國	900	大方公司	馬甲路8號	上海	中國	n/a	850,000.00	20070630	n/60

銷貨訂單

訂單編號	訂單日期	客戶編號	客戶訂單編號	客戶訂單日期	貨運公司	FOB條件	業務人員
B001	20071013	A001	C00123	20070930	新竹貨運	起運點	李麗華
B002	20071014	A002	H34892	20071001	DHL	目的地	范小文

銷貨訂單 條列 存貨

銷貨訂單編號	貨物編號	訂購數量
B001	D001	150
B001	D003	80
B002	D004	90

揀貨單

揀貨單編號	揀貨日期	揀貨員	銷貨訂單編號
C001	20071015	張五哥	B001
C002	20071017	王唯一	B002

揀貨單 條列 存貨

揀貨單編號	貨物編號	揀貨磅數
C001	D001	150
C001	D003	80
C002	D004	90

存貨

貨物編號	貨物名稱	單位售價(磅)	存放地點	目前存量(磅)	再訂購點
D001	摩卡咖啡豆	500.00	倉一	1000	500
D002	爪哇咖啡豆	560.00	倉一	1500	600
D003	曼特寧咖啡豆	650.00	倉二	2300	800
D004	藍山咖啡豆	890.00	倉二	800	400

出貨單

出貨單編號	出貨日期	出貨人員	揀貨單編號	客戶編號
E001	20071018	陳錦芳	C001	A001
E002	20071020	伍瑞于	C002	A002

銷貨發票

發票編號	出貨單編號	發票日期	發票金額	客戶編號
F001	E001	20071018	127,000.00	A001
F002	E002	20071021	80,100.00	A002

- 正規化 (Normalization) - 概念：
  - 資料庫正規化 (normalization) 是從關聯式資料模式衍生出來的理論，目的是為大雜燴型資料表的分割提供一套可遵循的模式。它可分成以下幾種形式(normal form)：
    - 第一正規化形式(First Normal Form, 1NF)：一份資料表內所有資料列的每個欄位，都只包含一個資料項，則該資料表即達到 1NF。
    - 第二正規化形式(2NF)：一份已符合 1NF 規範的資料表，若其主索引以外欄位均完全相依於主索引欄位，該資料表即達到 2NF。
    - 第三正規化形式(3NF)：一份已符合 2NF 規範的資料表，若其主索引以外的欄位之間不存在功能相依性，該資料表即達到 3NF。
  - 在大部分資料庫管理實務上，資料表的正規化只需要達到 3NF 即可。如果資料表的主索引是由多個欄位組成，該資料表須進一步達到 BCNF (Boyce/Codd NF)。
    - Boyce/Codd 正規化形式(BCNF)：一份已符合 3NF 規範的資料表，若其主索引內的個別欄位並不相依於主索引以外的任何欄位，該資料表即達到 BCNF。
  - 資料庫正規化還有更複雜的第四正規化形式(4NF)、第五正規化形式(5NF)，但實務上罕見使用，在此從略。
  - SAP HANA 資料庫和傳統的資料庫不同，它是 反正規化 (De-normalization) 的，相關內容請詳後續課程介紹。

正規化觀念練習題：

Z 資料表有 z1,z2,z3,z4,z5,z6,z7 等七個欄位，其中 z1 及 z2 是複合主索引，且 z3,z4,z5,z6,z7 等五個欄位均完全相依於 z1 及 z2 兩個欄位：

(1).我們稱此資料表已達到_____正規化。

(2).若 z4,z5 另外相依於 z3，則可知此資料表必須進一步達到_____正規化。

(3).若 z3,z4,z5,z6,z7 之間並不存在相依現象，且 z1 或 z2 也不相依於 z3 或 z4 或 z5 或 z6 或 z7，則我們稱此資料表已達到_____正規化。

五、SAP 的三種資料類型

SAP 把資料分成三大類型：

- **主檔數據 (Master Data)**：
  - **定義與特徵**：企業營運中長期存在、靜態、共享且不常變動的基礎資料。
  - **四大主檔**：
    - 客戶 (Customer)：KNA1
      - 查詢 t-code：**SE16N**，在 Table 欄輸入 **KNA1**，按 Enter，下方會顯示客戶主檔的欄位，把預設最大顯示筆數 500 刪除，按 **F8**，系統會顯示客戶主檔的全部資料。
        - 查詢特定客戶：用 t-code **BP** 進入 維護企業夥伴 頁面進行操作。
      - 資料庫表格結構：用 t-code **SE11** 進入 ABAP Dictionary，在 Table 欄中輸入 **KNA1**，按 **Display**。
    - 供應商 (Vendor)：LFA1
      - 查詢 t-code：**SE16N**，在 Table 欄輸入 **LFA1**，按 Enter，下方會顯示供應商主檔的欄位，把預設最大顯示筆數刪除，按 **F8**，系統會顯示供應商主檔的全部資料。
        - 查詢特定供應商：用 t-code **BP** 進入 維護企業夥伴 頁面進行操作。(註：在 S/4HANA 的前端操作中，客戶與供應商已被整合為「企業夥伴 (BP)」)
      - 資料庫表格結構：先用 t-code **SE11** 進入 ABAP Dictionary，在 Table 欄中輸入 **LFA1**，按 **Display**。
    - 物料 (Material)：MARA

        - 查詢 t-code：**SE16N**，在 Table 欄輸入 **MARA**，按 Enter，下方會顯示物料主檔的欄位，把預設最大顯示筆數刪除，按 **F8**，系統會顯示物料主檔的全部資料。
            - 查詢特定物料：用 t-code **MM03** 進入 顯示物料 頁面進行操作。
        - 資料庫表格結構：先用 t-code **SE11** 進入 ABAP Dictionary，在 Table 欄中輸入 **MARA**，按 **Display**。
    - 總帳會計科目 (G/L Account)：SKA1
        - 查詢 t-code：**SE16N**，在 Table 欄輸入 **SKA1**，按 Enter，下方會顯示 G/L 主檔的欄位，把預設最大顯示筆數刪除，按 **F8**，系統會顯示 G/L 主檔的全部資料。
            - 查詢特定總帳科目：用 t-code **FS00** 進入 G/L 編輯頁面進行操作。
        - 核心資料庫表格：先用 t-code **SE11** 進入 ABAP Dictionary，在 Table 欄中輸入 **SKA1**，按 **Display**。

- **交易數據 (Transaction Data)：**
    - **定義與特徵：**企業日常營運產生的活動紀錄，具備時間性，動態且會隨時間快速頻繁增加。
    - **結構：**通常由 Header (抬頭資訊) + Line Items (項目資訊)兩張表格組合而成。
    - **例子與特性：**銷售訂單、採購單、會計傳票，每一筆交易數據都會引用主檔數據。例如，採購單的結構：
        - Header：採購單抬頭表格(EKKO)，紀錄採購日期、供應商編號(主檔數據)、內部採購組織。
        - Line Items：採購單明細表格(EKPO)，紀錄物料編號(主檔數據)、採購數量、價格。Header 及 Line Items 兩張表格透過 採購單編號(EBELN) 連結。
- **配置數據 (Configuration Data)：**
    - **定義與特徵：**系統設定參數，通常是在企業剛導入 SAP，或是業務發生重大改變時，由 SAP 顧問或 IT 人員進行設定的，一般使用者（如會計或業務）不會去更改它。
    - **例子：**公司代碼(Company Code)、會計年度起訖月份、自動過帳規則、付款條件..。

六、總結與預告：從 RDBMS 到 In-Memory

- **本週總結：**資料庫是 AIS 的地基，主鍵與外來鍵串起了所有的資料庫表格。
- **下次課程預告：**

- 傳統關聯式資料庫將資料存放在硬碟，受限於讀寫速度 (I/O)，運算效率差。
- 下次課程將介紹 SAP 的革新技術-- SAP HANA，它透過記憶體式運算 (In-Memory Computing)，顛覆傳統資料庫的效能限制。

附錄：REA 會計資料模型 (REA Data Model)

一、 什麼是 REA 模型？

REA 模型是由 William E. McCarthy 教授於 1982 年提出的一種「企業本體論（Enterprise Ontology）」。傳統會計系統是建立在「借貸法則（Debits and Credits）」與「分類帳」的基礎上；而 REA 模型則是專為「關聯式資料庫（Relational Database）」時代所設計的現代會計架構。

它的核心理念是：資料庫底層不應該為了迎合傳統報表而預先儲存「借」與「貸」，而是應該**純粹且客觀地記錄「商業事件的本質」**。只要將基礎的商業活動完整記錄下來，系統隨時可以透過資料庫查詢（SQL）自動產出資產負債表或損益表。

這項理論的遠見，完美契合了我們後續要探討的 SAP HANA「通用日記帳 (ACDOCA)」與「單一真相來源」的設計哲學！

二、 REA 的三大核心元素

在進行資料庫塑模（Data Modeling）時，REA 將企業所有的商業活動拆解為三個基本實體（Entities）類型：

1. R - 資源 (Resources)：

- **定義**：企業擁有或控制的、具有經濟價值且稀缺的有形或無形資產。
- **實例**：現金（Cash）、存貨（Inventory）、機器設備（Equipment）。在 SAP 中，這通常對應到「物料主檔 (MARA)」。

2. E - 事件 (Events)：

- **定義**：會改變資源數量的商業活動。REA 著重於「經濟事件（Economic Events）」。
- **實例**：銷貨（Sales，導致存貨減少）、收款（Cash Receipts，導致現金增加）。在 SAP 中，這對應到「交易數據表」，如銷售訂單 (VBAK) 或會計憑證 (BKPF)。

3. A - 參與者/代理人 (Agents)：

- **定義：**參與經濟事件的內部或外部人員、組織。
- **實例：**內部參與者（如：業務員、採購員、出納）、外部參與者（如：客戶、供應商）。在 SAP 中，這對應到「商業夥伴 (BP)」或員工主檔。

三、 REA 模型的「雙重性」與基本規則

在繪製 REA 實體關聯圖（E-R Diagram）時，必須遵守以下三大核心法則，這也是電腦稽核人員檢驗系統架構是否合理的標準：

- **規則一（事件與資源的關聯）：** 每一個「事件」都必須至少與一個「資源」產生關聯。（例如：發生銷貨事件，必定會牽動存貨資源）。
- **規則二（事件與事件的對偶性 / Duality）：** 每一個「經濟事件」都必須與另一個「反向經濟事件」產生關聯。這代表了商業上的「給予與獲取（Give and Take）」。實例：企業「給出」存貨（銷貨事件），是為了「獲取」現金（收款事件）。這兩個事件的緊密連結，在資料庫層面完美取代了傳統會計的複式簿記邏輯。
- **規則三（事件與參與者的關聯）：** 每一個「事件」都必須至少與兩個「參與者」產生關聯（通常是一個內部人員，對應一個外部人員）。實例：一筆銷貨事件，必然牽涉到負責接單的「業務員（內部）」與下單的「客戶（外部）」。

四、 REA 與本週「關聯式資料庫」的結合

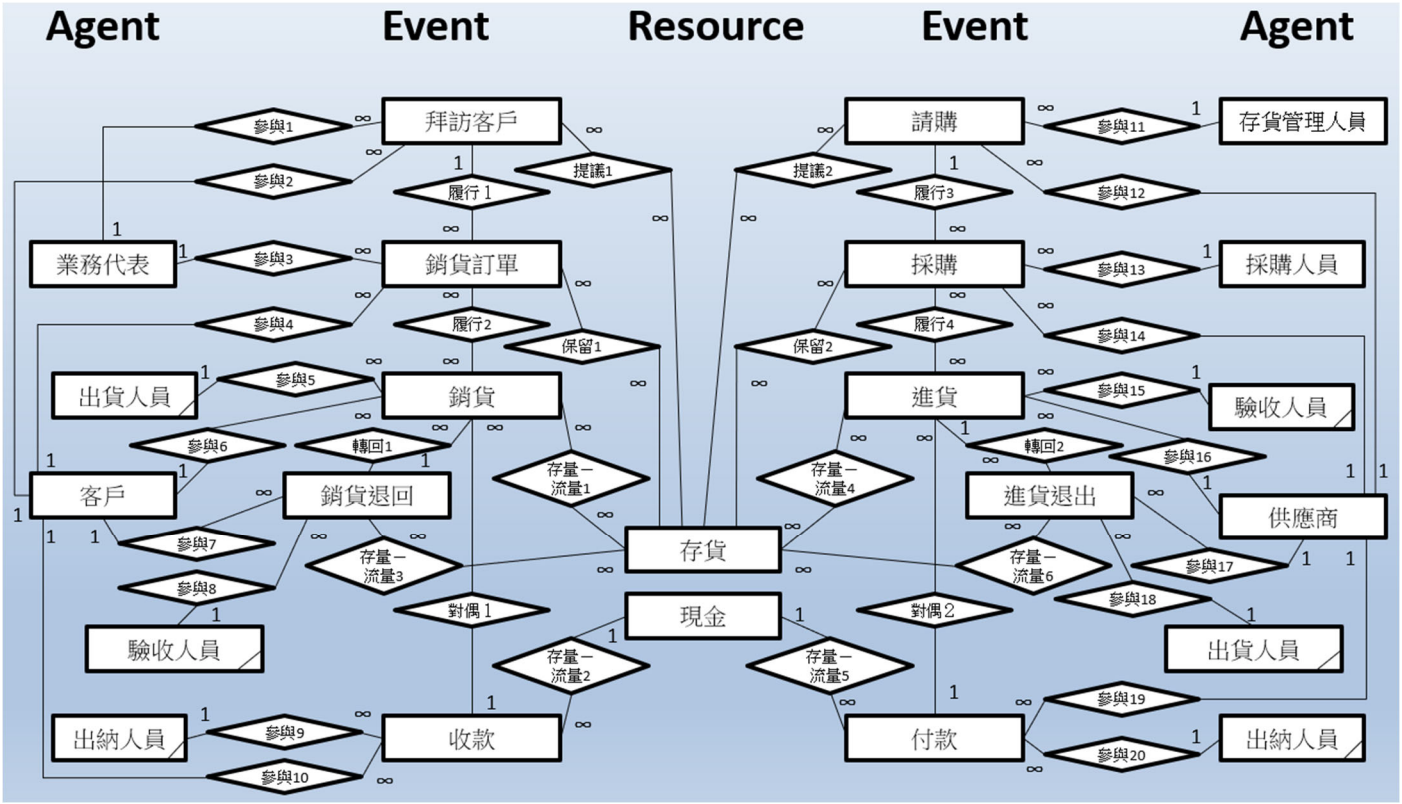
為何我們要在本週學習 REA？因為 REA 是將複雜的商業流程，轉化為關聯式資料庫表格（Tables）的最佳橋樑。

當我們運用 REA 方法論設計系統時：

1. **抓出實體：**先找出流程中的 R、E、A，建立各自的基礎資料表。
2. **建立關聯與外來鍵：**透過這堂課學過的「主鍵 (PK)」與「外來鍵 (FK)」，將參與者與資源的代碼，綁定到事件表中（例如：在銷貨單表中，記錄客戶編號與業務員編號）。
3. **處理多對多關係 (M:N)：**商業實務上，一張銷貨單通常會包含多種存貨（M:N 關係）。此時我們就會運用課堂上教過的技巧，拆解並建立一個「關聯實體表（如：銷貨單明細表）」，確保資料庫符合正規化（Normalization）的要求。

結語：REA 模型不僅是會計資訊系統的重要理論，更是我們在學習 ERP 系統底層邏輯時的絕佳透視鏡。了解 REA 後，大家就能明白為何 SAP 的「銷售訂單」畫面上，永遠強制要求輸入「客戶」與「物料」-- 這正是資料庫理論在真實商業世界的嚴謹實踐！

圖示：進銷存系統的 REA 模型 (見次頁)


Agent
Event
Resource
Event
Agent
拜訪客戶
請購
存貨管理人員
參與1
參與11
提議1
提議2
履行1
履行3
參與2
參與12
業務代表
銷貨訂單
採購
採購人員
參與3
參與13
履行2
履行4
保留1
保留2
參與4
參與14
出貨人員
銷貨
進貨
驗收人員
參與5
參與15
轉回1
轉回2
參與6
參與16
客戶
銷貨退回
進貨退出
供應商
存量－流量1
存量－流量4
參與7
存貨
存量－流量3
存量－流量6
參與17
參與8
參與18
對偶1
現金
對偶2
驗收人員
出貨人員
存量－流量2
存量－流量5
參與19
出納人員
參與9
收款
付款
出納人員
參與20
參與10