

# Lecture 3：SAP 系統管理(BASIS) 與 HANA 資料庫基礎(1)

編撰：國立屏東大學 周國華老師 本次修正：2026-03-29

## 一、課程導論

在上週，我們建立了關聯式資料庫 (RDBMS) 的基礎觀念。本週我們將進入 SAP 的技術深水區，探討支撐整個 SAP S/4HANA 運作的兩大技術支柱：

1. **SAP BASIS**：負責維持系統運作的「管理員」與「中介軟體」。
2. **SAP HANA**：負責高速運算與儲存的「核心引擎」。

## 二、什麼是 SAP BASIS？

1. **定義**：BASIS (Business Application Software Integrated Solution) 是 SAP 系統的「技術底層」或「作業系統」。它就像是電腦的 Windows/Linux OS，負責讓上層的應用程式（如 FI, CO, MM 模組）能夠順利執行，並管理硬體資源。

### 2. BASIS 的核心職責：

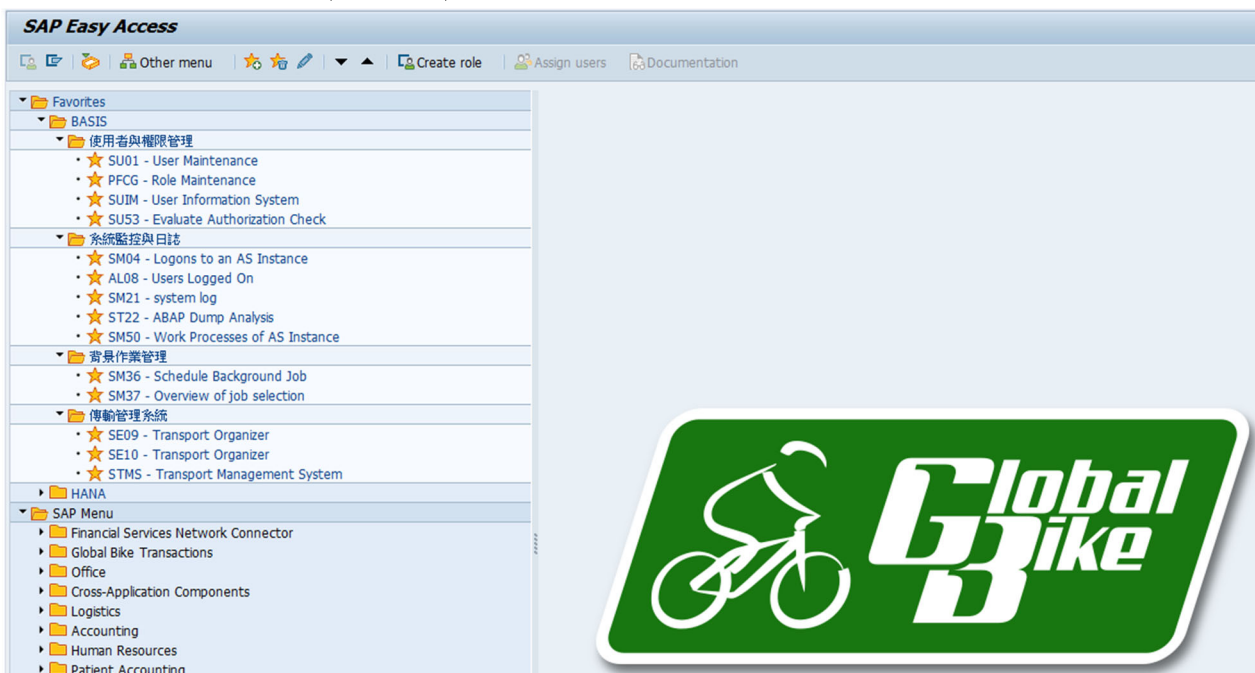
- **系統安裝與配置**：搭建 SAP 環境。(註：請參閱本次課程另附三份參考講義)
- **使用者權限管理 (User Administration)**：決定誰可以登入？誰可以用什麼功能？（這與會計的內部控制息息相關）。
- **系統監控 (Monitoring)**：確保系統不當機、效能不卡頓。
- **傳輸管理 (Transport Management System, TMS)**：管理程式碼從「開發區 (DEV)」移轉到「測試區 (QAS)」再到「正式區 (PRD)」的過程。這對稽核至關重要，因為它確保了程式變更的完整性與授權。
- **背景作業管理 (Background Jobs)**：設定半夜自動執行的報表或過帳程序。

### 3. BASIS 常用的 t-code：

- **使用者與權限管理 (User & Role Administration)**
  - 使用者維護 (User Maintenance)： **SU01**
  - 角色維護 (Role Maintenance)： **PFCG**
  - 使用者資訊系統 (User Information System)： **SUIM**
  - 權限檢查分析 (Evaluate Authorization Check)： **SU53**
- **系統監控與日誌 (System Monitoring & Logs)**

- 線上使用者清單 (User List) : **SM04 / AL08**
- 系統日誌 (System Log) : **SM21**
- ABAP 執行階段錯誤分析 (ABAP Dump Analysis) : **ST22**
- 工作流程監控 (Work Process Overview) : **SM50**
- **背景作業管理 (Background Jobs)**
  - 定義背景作業 (Define Background Job) : **SM36**
  - 監控背景作業 (Monitor Background Job) : **SM37**
- **傳輸管理系統 (Transport Management System, TMS)**
  - 傳輸組織器 (Transport Organizer) : **SE09 / SE10**
  - 傳輸管理系統 (Transport Management System) : **STMS**

圖示：周老師我的最愛(Favorites)中的 BASIS 交易代碼



### 三、SAP HANA：革命性的記憶體式資料庫

#### 1. 傳統資料庫的瓶頸：

傳統 RDBMS (如 Oracle, MS SQL Server..) 將資料存在硬碟(Disk)。當需要運算時，必須將資料從硬碟搬到記憶體(RAM)，運算完再寫回。這個「I/O 過程」是系統效能的最大瓶頸。

#### 2. SAP HANA (High-performance ANalytic Appliance) 的突破：

- **In-Memory Computing (記憶體式運算)**：HANA 將所有資料都常駐在記憶體中。CPU 可以直接存取資料，速度比讀取硬碟快 10,000 倍以上。

- **不再需要彙總表 (Aggregates)**：傳統系統因為慢，所以需要預先建立「月報表」、「年報表」存起來。HANA 速度夠快，可以隨時「即時計算」當下的報表，這就是 S/4HANA 能做到 Real-time 的原因。

### 3. 硬碟的角色：

當系統啟動，HANA 從硬碟載入資料，存入記憶體，之後的新增、刪除、修改、查詢直接在 RAM 運算，系統不再頻繁讀寫硬碟。但 HANA 會定期將資料儲存到硬碟，所以硬碟仍扮演永久保存與備援的角色。

## 四、欄式儲存 (Columnar Store) vs. 列式儲存 (Row Store)

這是 HANA 效能極大化的關鍵技術之一。

### 1. 列式儲存 (Row-oriented)：

- **原理**：將一筆資料（如：買受人 8 位數統編, \$1890, 應稅, \$95）存在一起。
- **優點**：寫入一筆完整交易很快（如：新增一張銷貨發票）。
- **缺點**：分析很慢。如果要算「本月發票稅額總計」，系統必須讀取本月發票資料表每一列的所有欄位，即使我們只需要「稅額」欄位。
- **適用場景**：傳統 OLTP (線上交易處理)。

### 2. 欄式儲存 (Column-oriented) - HANA 的強項：

- **原理**：將同一個欄位的資料（如：買受人的「稅額」）存在一起。
- **優點**：分析極快。要算稅額，只需讀取「稅額」這一長條資料，完全略過統編、銷售額、課稅別等無關欄位。
- **資料壓縮率高**：因為同一欄位的資料類型相同（如都是「課稅別」），內容重複率高(例如都是[應稅]、[零稅率]、[免稅])，HANA 可以將資料壓縮 5-10 倍。
- **適用場景**：現代 OLAP (線上分析處理) 與 OLTP 混合。
- **缺點**：要新增一筆資料（一整列）時，資料庫必須跳轉到磁碟的不同位置，分別寫入各個「欄」的區塊中。(請詳下次課程深入分析)

## 五、SAP HANA 的資料型態

SAP HANA 資料庫的資料型態(data types)分為原生 SQL 資料型態及 ABAP 字典型態兩個層級。

**1. SAP HANA 原生 SQL 資料型態**：這是最底層的資料庫標準，定義貼近 ANSI SQL 國際標準。以下為核心分類：

- **數值型態 (Numeric Types)**：這是財務系統中最核心的型態，區分為「精確數值」與「近似數值」。
  - **INTEGER (標準整數)**：4 Bytes，用於系統流水號、訂單數量..等。

- **DECIMAL(p, s) (精確小數)**：會計必用！ p 為總位數，s 為小數點後位數。用於精確記錄金額、匯率、單價..等。
  - **DOUBLE / REAL (浮點數)**：用於科學運算或統計分析。**絕對不能用於財務金額計算**，會產生浮點數誤差。
  - **字元與字串型態 (Character & String Types)**
    - **VARCHAR(n)**：可變長度字元，用於純英數代碼（如公司代碼 US00、總帳科目編碼）。
    - **NVARCHAR(n)**：可變長度 Unicode 字元，儲存包含中文等多國語言的文字（如客戶名稱）。
    - **ALPHANUM(n)**：HANA 特有型態，儲存純數字時會自動補前導零 (Leading Zeros)，優化 SAP 傳統單據號碼。
  - **日期與時間型態 (Date & Time Types)**：在電腦稽核中，時間戳記是查核的鐵證。
    - **DATE / TIME**：紀錄憑證日期 (YYYY-MM-DD) 或系統登錄時間 (HH:MI:SS)。
    - **TIMESTAMP (精確時間戳記)**：精確到小數秒，是稽核軌跡與系統 Log 的關鍵，能區分同一秒內發生的多筆高頻交易先後順序。
  - **大型物件型態 (LOB Types)**
    - **BLOB (二進位大物件) / CLOB (字元大物件)**：儲存非結構化數據，如圖片、PDF 憑證附件、長篇合約條款。
- 2. ABAP 字典型態 (DDIC)**：在應用程式介面（如 SE11）看到的型態，為了讓系統能支援多種資料庫(Oracle、SQL Server、HANA)，SAP 設計了 ABAP Dictionary「翻譯層」，不僅定義資料長度，更被賦予「商業與會計語意」。以下是在 SE11 看到的常見型態及其商業意義：
- **系統與文字基礎型態**
    - **CLNT (Client / 用戶端)**：區分不同集團的代碼（如屏大的 300），是多數資料表（如 MANDT）的主鍵型態。
    - **CHAR**：這是 SAP ABAP Dictionary (DDIC) 中最常見、也最泛用的文字型態。它可以容納英文字母、數字以及特殊符號，長度設定範圍可以從 1 到 30,000 個字元。
    - **LANG (Language Key)**：單一字元的語言代碼（如 E 代表英文，M 代表繁體中文，1 代表簡體中文，3 代表韓文），實現多國語言切換。
    - **NUMC (Numeric Character)**：這是字串而非數值，**不能用來做數學運算**。特性是會自動在前方補零（如輸入 123 會存成 000000123），常用於客戶編號 (KUNNR) 或物料編號 (MATNR)。
  - **日期與時間 (字串儲存)**：在 ABAP 字典中，日期與時間是以「純文字字串」儲存。
    - **DATS**：8 碼字元，格式永遠為 YYYYMMDD。
    - **TIMS**：6 碼字元，格式永遠為 HHMMSS。
  - **金額與數量的「雙生綁定」**：在 SAP 中，金額和數量**絕對不能單獨存在**，必須與「貨幣代碼」或「單位」成對綁定，這是防範財務災難的核心內控。

- **CURR (貨幣金額) + CUKY (貨幣代碼)**：CURR 對應底層的精確小數。系統強制規定 CURR 必須關聯一個 CUKY (如 USD, TWD)，否則無法存檔建立表格。
- **QUAN (數量) + UNIT (計量單位)**：紀錄買了幾個、生產了幾公斤，QUAN 必須強制綁定 UNIT (如 PC, KG) 才能生效。
- 沒有特定單位的數字：
  - **FLTP**：浮點數 (Floating Point)，在電腦底層是「近似值」的數字，用來儲存在公式換算後可能出現無窮小數的數字。FLTP 也可以指給金額欄位，這就需要和貨幣代碼雙生綁定。FLTP 型態的金額欄位**不能用在會計分錄內**(因為微量近似值差異就會導致借貸不平衡)，通常用在信用額度這類風險管理的情境中。(實務上，客戶訂單佔信用額=單價\*數量\*匯率，單價及匯率都可能以多位小數表達，計算結果經常出現無窮小數，如果信用額度欄位用 FLTP 作為資料型態，就方便 ERP 系統進行自動比對。)
  - **DEC**：用來記錄沒有特定單位的精確數字，性質和原生 SQL 形態中的 DECIMAL 相同。DEC 型態的欄位要給定總位數和小數位數，經常用在「匯率」、「百分比」及需要「毫秒必爭」的「長時間戳記，例如 YYYYMMDDhhmmssmmuuun」這類欄位上。
- 結構型態 **STRU**：STRU (Structure) 並非儲存單一資料的基本型態，而是「一組欄位的集合體」(如將姓名、電話打包為聯絡資訊)。它僅存在於記憶體運算中，不會在底層建立實體表格。

在撰寫 ABAP 客製化程式時，應優先使用 DDIC 型態，ABAP 執行環境會自動對應到該資料庫正確的原生型態。

**操作 1**：同學可以用 t-code **SE11** 進入 ABAP Dictionary 初始畫面，在 Database table 欄中輸入四大主檔的代碼(客戶 **KNA1**，供應商 **LFA1**，物料 **MARA**，G/L 科目 **SKA1**)，按 Display，查閱四大主檔內各欄位的資料型態。以客戶主檔 **KNA1** 為例，約有 240 個表格欄位(Field)，在對應的 Data Type 中可看到上述全部的 DDIC 型態，包含兩種類型的雙生綁定型態。

**操作 2**：接續 操作 1 的 **KNA1** 主檔，點按最上方選單的 [Utilities] => [Database Object] => [Display]，畫面變成顯示 **KNA1** 主檔 240 個欄位的**原生 SQL 資料型態**，包含資料長度及預設值。從頭到尾檢視一遍，可知 **KNA1** 主檔使用的原生資料型態僅有 **NVARCHAR** 及 **DECIMAL** 這兩種。

## 六、總結與預告

- **回顧與本週總結**：上週我們學會了傳統關聯式資料庫 (RDBMS) 如何透過主鍵與外來鍵將資料表連結起來。本週我們探討了支撐 SAP 運作的雙引擎：**BASIS (系統管家)**及 **HANA (運算引擎)**，以及 HANA 的資料型態。
- **下次課程預告**：雖然 HANA 的「欄式儲存」讓讀取與分析變得極快，但在日常頻繁「寫入」會計分錄時，卻會面臨解壓縮與資料重排的效能瓶頸。下週我們將深入探討 HANA 如何巧妙解決這個寫入難題 (透過：Delta Merge 增量合併)，以及這項底層技術的革新，最終如何促成 S/4HANA 財務會計的歷史性變革--「通用日記帳 (Universal Journal, ACDOCA)」的誕生。