

<<會計資訊系統課程講義>>

關聯式資料庫與概念資料模式 (RDB & ERD)

周國華

國立屏東大學會計學系

初版：2007/7/24

本次修訂：2021/2/19

智慧財產權聲明

- 本文件係由周國華老師獨自撰寫，除引用之概念屬於原文作者外，其餘文字及圖形內容之智慧財產權當然屬於周老師獨有。
- 任何機構或個人，在未取得周老師同意前，不得直接以本文件做為學校、研究機構、企業、會計師事務所、政府機關或財團法人機構舉辦教學或進修課程之教材，否則即屬侵權行為。
- 任何機構或個人，在未取得周老師同意前，不得在自行編撰的教材中直接大量引用本文件的內容。若屬單頁內部分內容之引用，亦請註明出處。

第一部份

關聯式資料庫(RDB) 基本概念

RDBMS

- **RDBMS** (Relational Database Management System)：關聯式資料庫管理系統。使用者根據RDB model建立資料庫。
- RDB model係由E. F. Codd在1970年創建。資料儲存在二維表格中，每份表格有一欄為主索引(primary key)，兩份表格藉由外來鍵(foreign key)建立彼此間的關聯。
 - 主索引的值可唯一識別資料表的特定資料列，因此不可重複。主索引亦可由兩個或多個欄位共同組成，稱為複合主索引(composite primary key)。
 - 當資料表的某個欄位值，是另一個資料表的主索引值時，該欄位稱為外來鍵。表格之間藉由外來鍵建立關聯性，而不由儲存表格的實體位置來代表。
- RDBMS使用SQL語言建立、修改、移除資料庫物件及新增、刪除、修改資料。

SQL

- SQL (Structured Query Language, 結構化查詢語言), 可區分成三種次語言：
 - Data Definition Language (DDL, 資料定義語言)：用來建立及修改資料庫物件，包含create, alter, drop, rename, truncate等敘述。
 - Data Manipulation Language (DML, 資料操作語言)：用來修改資料庫表格中的資料，包含insert, select, update, delete, commit, rollback, savepoint等敘述。
 - Data Control Language (DCL, 資料控制語言)：用來配置安全性，以執行資料庫工作和操作資料庫物件，包含commit, rollback, grant, revoke等敘述。
- 不同資料庫軟體通常會使用各自延伸的SQL版本，例如微軟的SQL Server及Sybase DB使用T-SQL，Oracle DB使用PL/SQL，IBM的DB2使用SQL PL。

SQL範例

- 建立表格

```
CREATE TABLE 會計資訊系統成績單 (  
    學號          INT,  
    姓名          VARCHAR (10),  
    學期成績      INT,  
    PRIMARY KEY  (學號)  
);
```

- 插入資料

```
INSERT INTO 會計資訊系統成績單 (學號,  
    姓名, 學期成績) VALUES (92404060,  
    李小敏, 84);
```

- 查詢

```
SELECT 學號, 姓名  
FROM 會計資訊系統成績單  
WHERE 學期成績 >= 60 ;
```

- 確認及撤銷

```
INSERT INTO 會計資訊系統成績單 (學號,  
    姓名, 學期成績) VALUES (92404060,  
    李小敏, 84);  
COMMIT;  
  
INSERT INTO 會計資訊系統成績單 (學號,  
    姓名, 學期成績) VALUES (92404065,  
    王美麗, 83);  
ROLLBACK;
```

RDBMS產品

- 商用RDBMS軟體，大致由Oracle (42%)、Microsoft (24%)及IBM (13%)三家公司的產品主宰市場(括號內為2020/8市佔率)。
- Oracle是市占率最高的資料庫軟體公司，它最新的商用資料庫版本為21c。它也有Oracle Autonomous Database這個雲端化的資料庫平台服務。
- 微軟的資料庫產品有伺服器版的SQL Server、單機版的MS Access (最新版本均為2019版)，及雲端版的Azure SQL DB。
 - 本課程後續資料庫實作部分，將使用MS Access。
- IBM的資料庫產品稱為Db2，最新版本為11.5。
- 其他重要的資料庫軟體產品：
 - SAP Hana, Sybase
 - MySQL, MariaDB

ORDBMS

- **ORDBMS**：物件關聯式資料庫管理系統。
- 特性：
 - 與**RDBMS**概念完全相容。
 - 支援使用者定義資料類型。
 - 支援多媒體和其他大型物件，可在單一資料欄中儲存影音、圖片、大量文字。
- 例：**Oracle DB**自**Oracle 8**開始融入**ORDBMS**概念，**Oracle 10g**已成為標準**ORDMBS**。

XML DB

- 一般而言，**RDB**處理**XML**文件可分成三種模式：
 - **Textual fidelity**: 將**XML**文件內容以純文字型態儲存在**RDB**的欄位內。
 - **Relational fidelity (shredding + XML publishing)**: 將**XML**文件的內容及格式儲存在**RDB**的欄位內。
 - **XML fidelity**: 將**XML**文件按照**schema**結構以樹狀方式儲存。此模式稱為**Native XML**。
- 目前，主要的**DBMS**軟體都已在**RDB**中提供處理**XML**文件的機制，可透過一般**SQL**語法查詢**XML**文件；某些高階產品進一步提供符合**XQuery**標準的資料庫查詢功能。
 - **XBRL**文件也屬於**XML**格式，實務上，**XBRL**文件有兩種儲存方式，其一為單純以檔案方式存在資料夾內，其二則是以**Native XML**方式儲存在**XML DB**內，但此時**XML DB**需具有處理**XBRL**格式的能力。
 - 以**Oracle**為例，其**XML DB**可外加**Oracle**特別開發的**XBRL Extension**模組，**XML DB**就能處理**XBRL**格式文件。

JSON 文件的儲存

- **JSON (JavaScript Object Notation)**是一種輕量級的資料傳輸格式，目前已成為主流的文件交換格式。
- **JSON**將資料以**key:value**的配對方式表達，和**Python**語言的**dict**資料結構很像。它使用大括號{ }包住物件內容。
JSON範例：
`{"city": "台北市", "mayor": "柯文哲", "population": 2608332}`
- 目前，主要的**DBMS**軟體都已在**RDB**中提供處理**JSON**文件的機制，可以輕易地將文字資料輸出成**JSON**格式或是將**JSON**文件轉存進**RDB**內。
 - 大部分**DB**以純文字型態把**JSON**格式文件內容儲存在**RDB**的欄位內。
 - 某些**DB**將**JSON**的**key**和**value**分開儲存，以提高處理效率。
 - **Oracle DB 21c**有一個內建的**json**資料型態，可以更有效率處理**JSON**文件。

第二部份

個體關係模式 及 個體關係圖

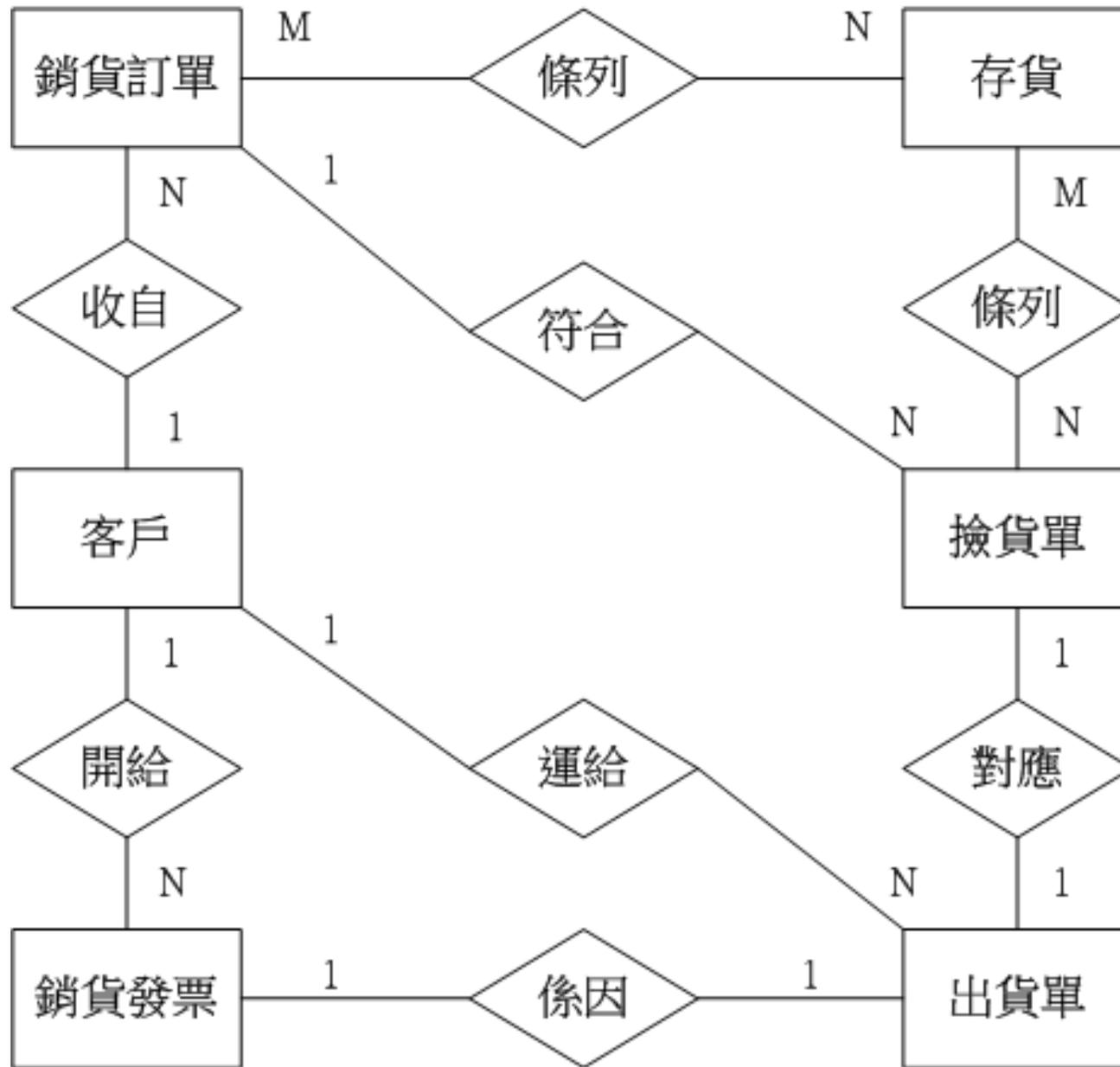
Data Modeling

- **Data Modeling:** 資料塑模，係指依據任何一種資料模式理論來建立資料模式(data model)的過程。
- **Data model：** 資料模式，又稱為綱要(schema)，可分成三種
 - **概念資料模式(conceptual data model)：** 以簡化的模型描述組織的運作內涵，例如，以ERD的個體(E)來代表欲蒐集資料的對象，並在兩兩個體之間建立關係(R)，以描述個體之間的互動。
 - **邏輯資料模式(logical data model)：** 以詳細的書面表格將概念資料模式中的個體結構呈現出來，並藉由表格內的主索引及外來鍵呈現表格之間的關係。
 - **實體資料模式(physical data model)：** 在特定RDBMS軟體中實作出邏輯資料模式之內涵。

ERD & ER Model

- ERD：個體關係圖(entity-relationship diagram)。
- ER model是由Peter Chen在1976年發表，是一種概念資料模式(conceptual data model)。根據ER model所建立的概念圖形稱為ERD。
- ERD內通常包含四種元素：
 - 個體(entity, 即E)：代表欲蒐集資料的對象，以矩形表示。
 - 關係(relationship, 即R)：在兩兩個體之間建立合乎邏輯的關係，以菱形表示。在簡化的ERD中，通常會將關係的菱形圖形省略。
 - 關係的實質內容是以基數性(cardinality)來表示，可分為1對1、1對多及多對多三種。
 - 屬性(attributes)：個體或關係內的資料項目(即表格中的欄位)，以圓矩形表示。當描述稍微複雜的系統時，ERD內通常會省略此元素。

ERD範例：銷貨系統



基數性：1對1

- 基數性(**cardinality**)：兩個個體之間若具有特定關係，可用基數性表示其關係之內涵。
- 1對1關係 (**1:1**)：表示兩個個體都僅能以一個實例(**instance**)參與此關係。
- 例如：在前述**ERD**範例之「銷貨發票」與「出貨單」之間的基數性關係為**1對1**，意含「一張銷貨發票的內容對應一張出貨單，一張出貨單只能開一張銷貨發票」。

基數性：1對多

- 1對多關係 (1:N 或 1:∞)：1方的一個實例，可以面對多方的多個實例；多方的一個實例，只能面對1方的一個實例。
- 例如：在前述**ERD**範例之「客戶」與「銷貨訂單」之間的基數性關係為 1對多，意涵「一位客戶可發出多次(張)銷貨訂單[客戶發出的進貨訂單，對供應商而言是銷貨訂單]，一張銷貨訂單只能來自一位客戶」。

基數性：多對多

- 多對多關係 (M:N)：一邊多方的一個實例，可以面對另邊多方的多個實例；反之亦然。
- 例如：在前述**ERD**範例之「銷貨訂單」與「存貨」之間的關係為多對多，意涵「一張銷貨訂單內可包含多種存貨，一種存貨可出現在多張銷貨訂單內」。

基數性的相對性

- 兩個個體之間的基數性，並非絕對的。其內涵受到企業政策之左右。
- 例如：「公務車」與「業務員」之間
 - 1對1：表示「一部公務車只能讓一位業務員使用，一位業務員只能使用一部公務車」。
 - 1對多：表示「一部公務車可讓多位業務員使用，每位業務員只能使用一部公務車」。
 - 多對1：表示「一部公務車只能讓一位業務員使用，一位業務員可使用多部公務車」。
 - 多對多：表示「一部公務車可讓多位業務員使用，一位業務員可使用多部公務車」。

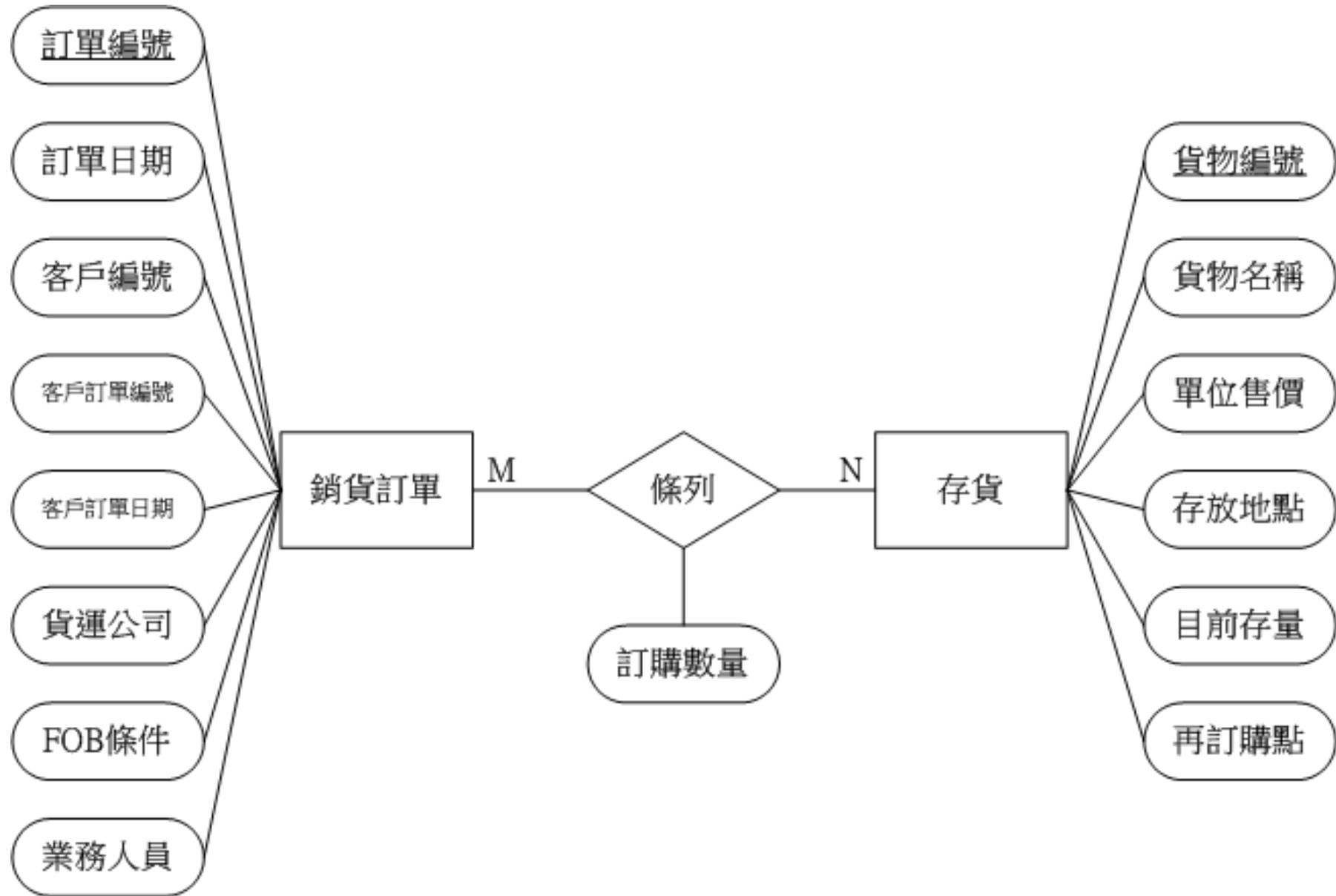
最小基數性

- 最小基數性(minimum cardinality)又稱為參與限制(participating constraint)，用以指明個體在特定關係中的最低參與程度。
- 例如：
 - 「員工」(1, 1)對「已完成工作」(0, N)：表示一名員工可能沒完成任何工作(新進員工)，也可能已完成許多工作；一份工作僅能由一位員工完成(無法自動完成、也不能由超過一位員工合作完成)。
 - 「學生」(10, 60)對「課程」(2, 5)：表示一名學生至少需修2門課、至多能修5門課；一門課至少需有10名學生選修，至多不能超過60名學生。

屬性

- 屬性是個體或關係內的資料項目。在將個體或關係轉成資料表後，屬性就是資料表內的各個欄位。
 - 一對一 或 一對多 的關係沒有屬性，只有多對多之間的關係才有屬性，這些屬性將成為關係資料表(通常會把兩個多方的主索引納入，成為複合主索引)的一部份。
- 每個個體都有多個屬性，有些屬性的值可決定其他數個屬性的值，此種屬性稱為**決定性屬性(determinant)**。資料表經過正規化(normalization)後，剩下的決定性屬性稱為**候選鍵(candidate key)**，此屬性的值具有不重複性、且可決定所有其他屬性的值。候選鍵若不只一個，資料庫管理師(DBA)須挑選一個做為**主索引(primary key)**，或稱**主鍵**。
 - 例如，「員工編號」及「身分證字號」都是「員工」資料表的候選鍵，可任選一個做為主索引。在**ERD**中主鍵名稱應加上底線。

屬性標示範例：訂單&存貨



將ERD轉換成邏輯模式

- **ERD**描繪完成後，可按以下步驟轉成邏輯資料模式：
 1. 為每個個體編製一張資料表(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。
 2. 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。
 3. 在多對多關係中，另外增加一個關係資料表，將兩個個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)。
 4. 在**1**對**1**關係中，把未來最有可能轉成多方的個體暫時當成多方，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。

邏輯資料模式範例

- 將前述銷貨系統ERD範例轉成邏輯資料模式如下(共8張表格)：

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	伍瑞子	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

外來鍵 & 參考完整性

- 多方資料表內任一筆資料列的外來鍵值，可連結至一方資料表內的特定資料列。為確保此外來鍵值所指向的資料列確實存在、未被刪除，可在**DBMS**內設定外來鍵的參考完整性(**referential integrity**)。一經設定後，一方的特定資料列的主索引值只要被多方特定資料列的外來鍵引用，該一方特定資料列即無法被刪除(除非修改多方的外來鍵值)。
- 有時，多方特定資料列的外來鍵的確無法在一方找到對應的主索引值，此外來鍵可設定為空值(**null**)。
 - 例如，在前述銷貨系統**ERD**範例之「銷貨發票」與「客戶」之間的關係為多對**1**，因此，銷貨發票資料表內會有一欄位為客戶編號(客戶資料表之主索引)。在現銷的情境下，許多客戶並不願留下任何記錄，故此類交易之銷貨發票資料表內客戶編號欄會留白。

實體資料模式：MS Access

- 將前述邏輯資料模式實作在MS Access資料庫所產生之資料庫關聯圖如右：

