

# 薪資管理系統 的再生與繼往開來

詹雪梅

國立清華大學計算機與通訊中心校務資訊組

smjan@cc.nthu.edu.tw

## 摘要

網路與電腦的普及，傳統應用系統的設計與技術受到衝擊與考驗，本校自民國 75 年起即開始校務行政電腦化業務，早期的大型封閉式主機 HP3000 與網連式(Network)資料庫，隨著業務的持續增長與用戶層面的擴增，已面臨瓶頸，考量軟硬體的維護與成本，與圖形視窗介面的應用趨勢，及因應網際網路時代的來臨，遂於民國 87 年起開始規劃進行汰換主機、系統轉型等事宜。

本人首先將負責的各項系統之歷年來的需求提案單，加以分類統計，作成問題點分析。利用系統轉型，資料庫、程式介面可全面換新之時機，從改革資料庫結構、檢討流程、及將作業規則表格化等重點著手，徹底改善系統，期使新一代系統功能更趨完善。新系統工程已完成，選出其中最具代表性的薪資管理系統作為簡介，以它的再生案例，說明如何繼承往昔所有業務效能之持續運作，同時也為未來拓展應用鋪下便利之路。

關鍵詞：校務行政，薪資管理。

## 1. 薪資管理系統

本系統作業範圍廣泛，只要是校內人員屬於薪資所得之金額發放，皆屬其範疇，包含教職員工薪資、計畫薪資、契約工薪資、各類鐘點費、

考績及晉級、調薪追加、年終獎金、年節獎金、不休假加班費、各類補助費、各類代扣款、所得扣繳與申報、銀行轉帳及帳號管理...等子系統。

一般人會認為學校人員單純，薪資發放應屬易事，然而不幸的是，本校研究計畫眾多，受薪人員領域廣泛，包括專任教師、兼任教師、教官、職技員、工友、契約工、博士後研究員、助理、學生、計畫專案人員、聘僱人員、退休人員等。

不同的所得人編號，擷取不同的人事資料來源；不同的體制，產生不同的給薪標準；不同的所得性質，採用不同的扣繳標準；不同的身份，決定不同的代扣撥款來源；不定期的補助或獎金，可能併月薪或獨立機動放款...等等因素，建造出本系統如網狀般的複雜性，處理上的困難度很難被一般人所理解。

## 2. 提案單制度

本組自發展校務資訊系統以來，對於系統的開發與維護，一直採用提案單制度，業務單位如因環境因素、政策法令的變動，需要修改系統，或是業務上有新增功能之需求，均須先填寫『系統需求改善提案單』，向本組提出申請，經本組進行需求評估後，再核定是否實行。

本人自民國 79 年 9 月開始承接薪資管理系統，將歷年來的提案單作一統計歸類如表 1。

表 1 提案單需求類別分析統計表

年度 類別	79	80	81	82	83	84	85	86	87	88	89	90	91
需要變更資料結構	3	1	1	4	2	10	3	4	0	0	0	0	0
規則變動	4	5	2	5	15	5	1	2	5	1	1	2	0
配合其他系統	4	0	1	1	1	3	2	1	0	1	5	0	2
新增需求功能	9	11	21	26	8	7	8	6	3	2	1	2	6
報表內容調整	12	5	17	11	13	8	0	2	1	2	3	1	2
其他需求修改	5	7	8	5	3	3	4	2	2	3	1	1	1
資料處理	2	6	6	6	1	1	0	0	0	1	0	0	0
自提系統改善	3	15	11	8	6	7	2	0	1	1	1	1	0
其他單位支援	0	3	2	2	1	1	1	0	0	0	0	0	2
合計	42	53	69	68	50	45	21	17	12	11	12	7	13

### 3. 提案單需求類別分析

#### 3.1 需要變更資料結構

薪資系統中的資料檔案，以「個人薪資明細」為最主要，其檔案結構示意圖如圖 2。

先人建設之資料架構，有如傳統房屋建築模式般，必須預先建置好幾房幾廳的規模，一個房間供一個房客使用。為了未來的擴展性，初期大部分都會先預置幾個空房，雖然有點浪費空間，但對於往後有新加入房客的需求時，暫時提供了即時住房的便利性。例如：圖 2 中之 SAL-BANK-CRED1、SAL-BANK-CRED2、SAL-BANK-CRED3、SAL-BANK-CRED4 等資料項目，就是預留給銀行貸款使用的例子。

銀行貸款業務興盛，薪資代扣業務順勢而生，期間增加了不少數項代扣款項，如：公務人員健保費、勞工健保費、退休撫恤金、離職儲金、中興銀行貸款、台灣銀行貸款、退休撫恤金貸款、北院房租費、宿舍管理費、汽機車停車費、法院扣款...等。

系統原本的資料架構與應變措施，最終還是無法承載，最後一途仍是 - 房屋重建，重新配置空間。然而空間的擴充永遠不及未來需求的變化，這段期間還是數度修改檔案結構，加上每次都不可避免的程式修改工程，著實付出相當大的維護成本。

#### 3.2 作業規則變動

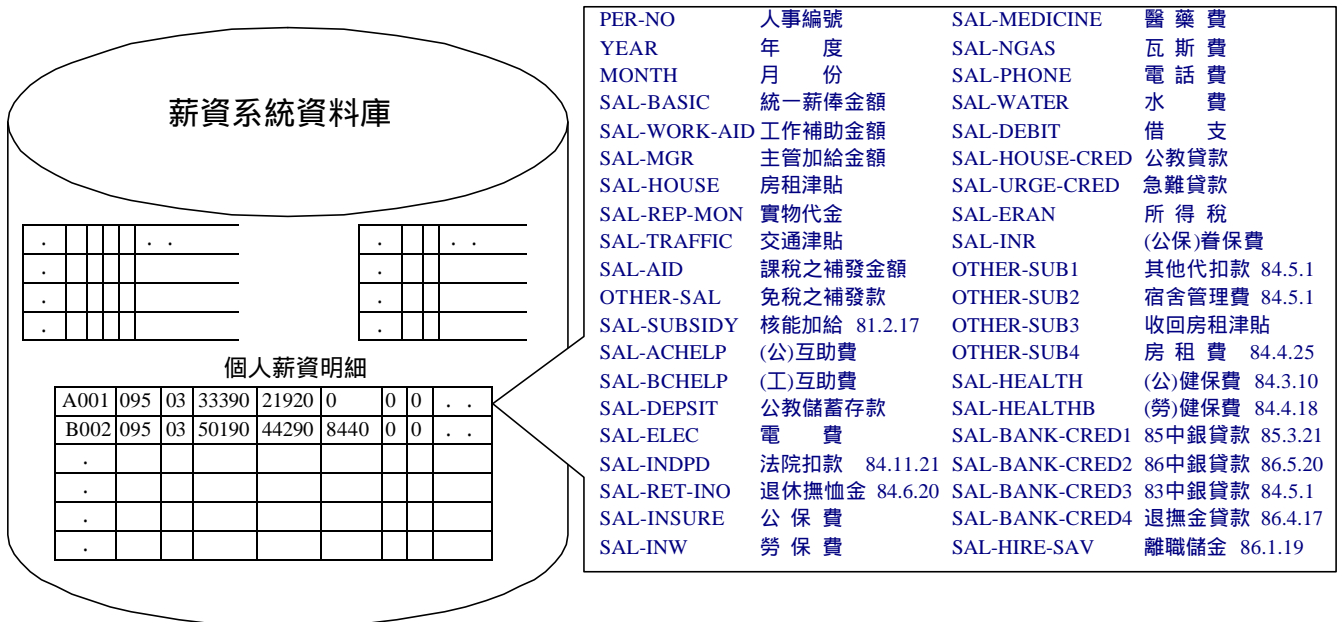


圖 2 舊系統檔案結構示意圖

對於資料結構的修改，資料庫大部分都會有管理工具可進行整個資料重建，資料量的多寡會影響執行的難易，但最困擾的部分是整建過程後接續的程式修改工程，要在系統中一一檢視程式是否需要修正，及真正對指令進行編輯，全靠人力，別無他法可取代。薪資系統的歷史悠久，資料量龐大，衍生的系統功能眾多，集所有不利條件於一身，沒有人會願意經常去更動結構。

上列的建屋架構，等住宿率達到滿點時，當時採取的應變措施是清理不用房間的房客，以新房客遞補該房，例如：圖 2 中之 SAL-RET\_INO，原本是退休保險費，因自 81 年 2 月後就沒有發生代扣事實，所以轉為給退休撫恤金使用。此法雖然暫時解決問題，但是對於資料的追蹤，卻無法提供忠實的參考史實。

民國 84 至 86 年間，因全民健保開辦，又值

作業規則的變動，大致上可分為兩類，一是發放所得的計算，一是代扣款項的計算。從表 1 中可觀察出每年幾乎都有因規則變動而需修改程式的提案單，其中九成都是與所得稅法有關。早期的設計，均是在程式中寫下所有的計算法則，寫下一堆 If ...Else ... 指令與設定扣繳稅率值，當變更稅法或是調整稅率時，就要由程式人員來修正所有相關程式。在 If ...Else ... 指令篩選的過程中，經常暗藏危機，如果沒有謹慎閱讀程式，檢查邏輯，常會遺漏一些特殊因子，當問題發生時，程式人員就有承擔責任的風險。

最顯著的例子是：教官原則上所得免申報，但其適用範圍僅限於統一薪俸、專業加給、交通津貼、子女教育補助費，其餘的津貼則一律要申報扣繳。開發初期，教官因無他項津貼，業務單位沒有告知但書，直到問題浮出檯面，才引發一

場爭議。

資料在整合體系中，最怕遇到的事是上游端業務單位遺忘與下游端業務單位的關係，常常總

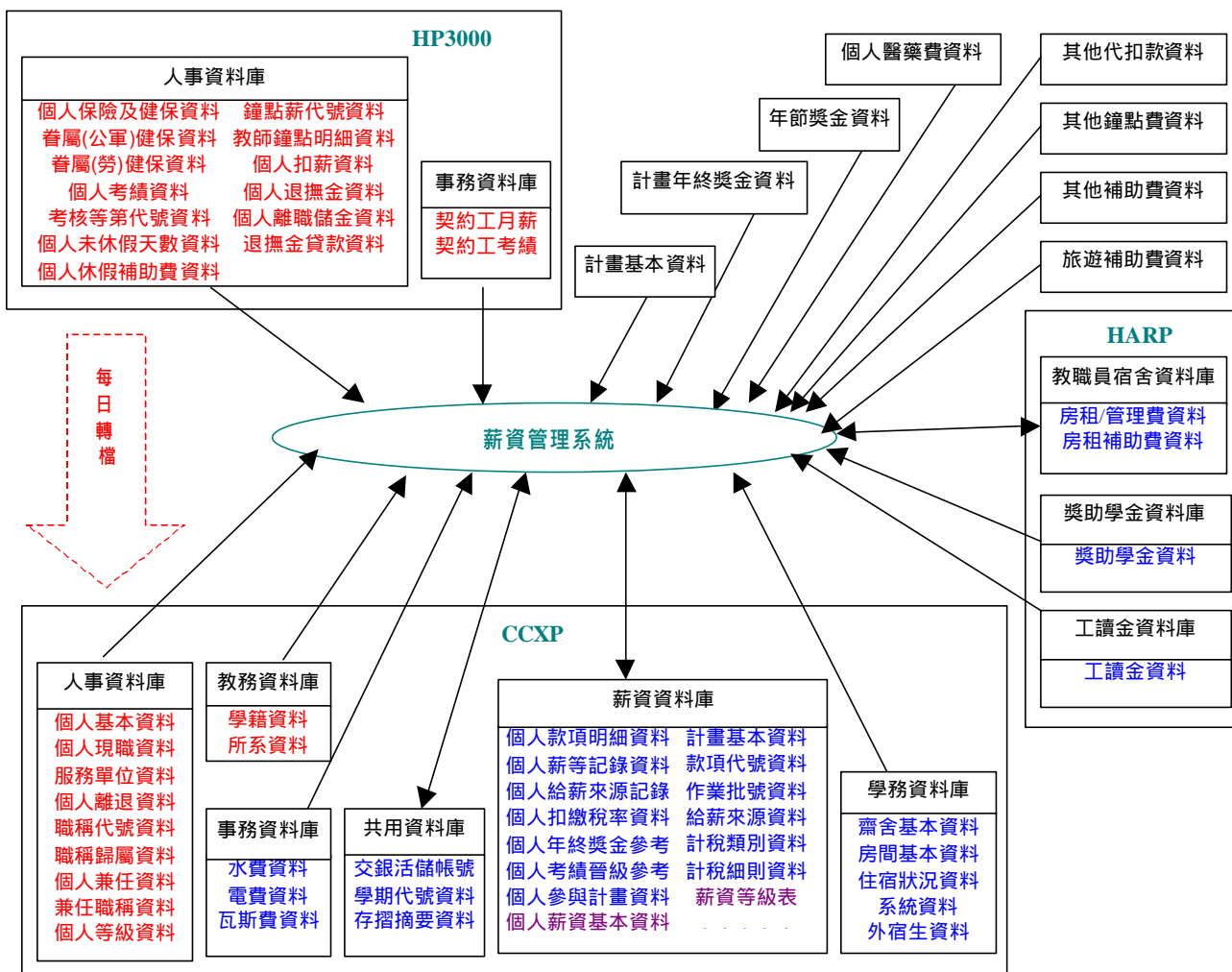


圖3 薪資管理系統關聯圖(93年9月版)

### 3.3 配合其他系統

在整合的校務資訊系統中，薪資系統屬於下游端系統，常常需要擷取利用各項上游端系統提供的資訊來檢核或產製自己的資料。例如：人事系統提供的教職員工資料，及教務系統提供的學生資料等，用來檢核個人身分狀況的正確性；事務系統提供的水、電、瓦斯費等，用來轉入個人薪資明細資料，作為代扣款項的依據。薪資系統與其他系統之資料關聯，參考圖3。

下游端系統能夠正確擷取上游端系統的資料，全仰賴遵守上游端系統資料的規範，當上游端系統資料結構有變化或來源處有異動時，下游端系統必須連帶地一併修正，才能維持系統原本正常的運作。像這類配合其他系統的提案需求總是無法避免，也是無法拒絕的，尤其在各系統紛紛轉型期間，與上游端系統的切換、銜接，總是絡繹不絕。

是等到下游端業務單位在線上作業處理時，發現異常訊息，才恍然獲知上游端業務有變動，此時才緊急提出修改需求，這種迫切任務如履薄冰、危如累卵，是非常難受的考驗。

### 3.4 報表內容調整

報表的問題永遠是本系統中最難克服的部分，無法永久固定格式，也不能用他法取代。

由於請款必定要有憑證以開立傳票，對於非費用性之發放款，印領清冊成為標準之附件依據。本校請款支付流程如圖4，由流程得知，製表的單位是出納組，也是本系統的需求提案者，但對所提出的報表提案，往往沒有完全的決策權，常會因為審查單位的人或事異動，而被迫要求修改報表規格，難以配合的層面涉及太廣，長久以來一直很難突破。

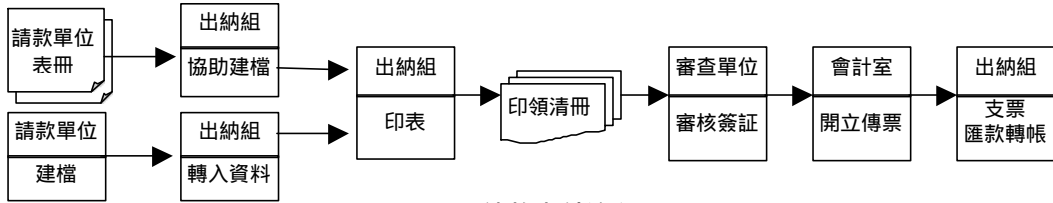


圖 4 請款支付流程

### 3.5 其他

早期的系統開發，並沒有全面思考完整的作業程序應備的相關功能，當處理程序中臨時有一環節出錯時，就只能直接對資料庫進行資料緊急救援的處理。本人承接系統後，慢慢發現系統中的瑕疵，自行利用空檔，逐一補足系統缺失的功能，所以表 1 中之“資料處理”與“自提系統改善”類別的次數出現左偏峰現象。

### 4. 資料庫改革

鑑於昔日資料結構變更提案單的痛苦經驗，及因應未來會不斷新增的需求量，以規畫具有高彈性的資料庫為原則，儘量降低未來檔案結構會變動的衝擊性，將原本的系統架構全面檢討改進，以「個人薪資明細」檔案為例，結構變更為「個人款項明細」，示意圖如圖 5。

新結構的設計觀念，繼前 3.1 節觀點所述，就像是摒除傳統房屋預定幾房的建築模式，改用一屋一房的活動屋，不用預先留置空房，當有房客要進住時，隨時再加掛活動屋即可，而連結每一活動屋的關鍵點就是所謂的索引鍵值。

舊架構中，以每一記錄代表每一個體每個月所有的收支相關資料，以資料項目名稱的不同代

表每一種款項，對於搜尋資料時可馬上展開陳列是其方便性，但卻有資料項目無法彈性擴充的缺失。

新架構中，以每一串記錄代表每一個體每個作業批次點所有的收支相關資料，以每一資料項目中的款項代號值，代表每一種性質的款項名稱，所以每一個體之記錄串的記錄數目不盡相同。對於搜尋資料時，必須連結款項代號的對照表，才能解釋出款項的意義，雖然有些不便，但在新系統 SQL 資料庫中，這類事件是很容易解決的事。

自 90 年 8 月新系統上線改用新結構後，需要變更資料結構的需求提案單已降至為零，這是改革成功的實證。

### 5. 作業規則表格化

以 3.2 節所言，對於所得稅扣繳，在新系統中改善的作法是將作業規則表格化，將一些變動因子改以資料表方式建立。

首先將薪資按性質與受薪人身分別分為幾大計稅類別，如：教職員工薪資、助理薪資、契約工薪資、學生薪資、各類鐘點費、年終獎金、各類補助費等，建立成「計稅類別資料表」如下：

類別	類別名稱	免預扣最大所得額	免預扣最大稅額
----	------	----------	---------

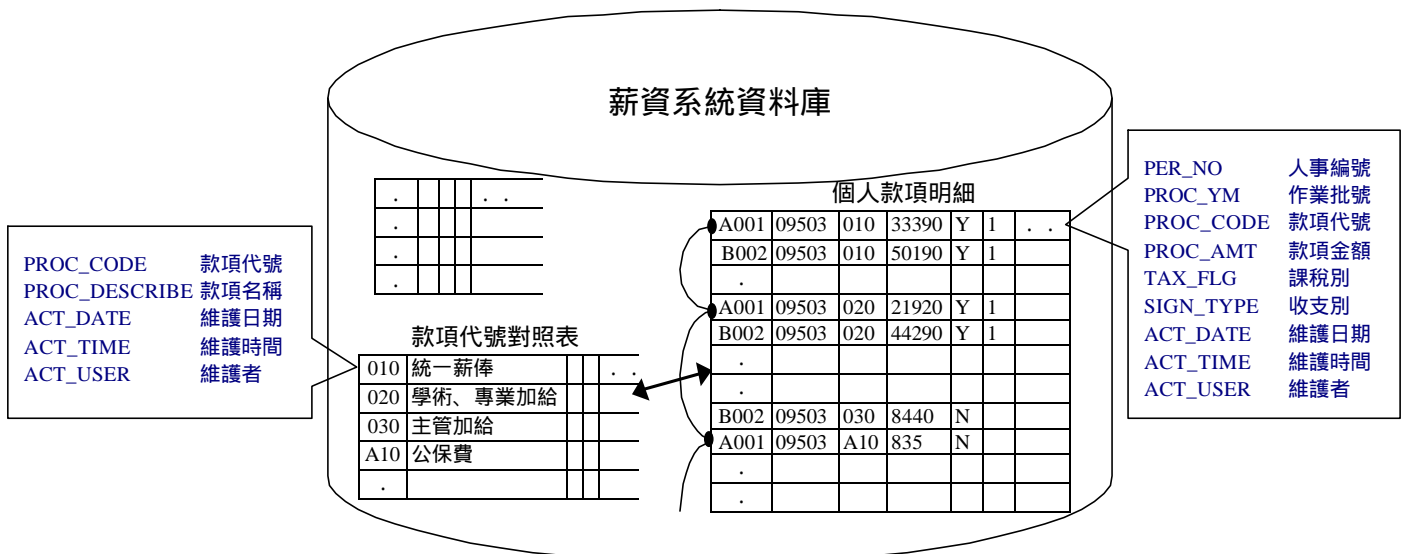


圖 5 新系統檔案結構示意圖

1	教職薪資	33333	2000
2	助理薪資	33333	2000
3	學生薪資	99999	2000
4	年終獎金	33333	2000
5	各類鐘點費	33333	2000

然後依計稅類別分別建立其下特殊款項的扣繳稅率，集合成「計稅細則資料表」如下：

計稅類別	款項代號	款項說明	扣繳率
1	011	統一薪俸補發	0.06
1	014	統一薪俸晉級差額	0.06
1	021	專業加給補發	0.06
1	024	專業加給晉級差額	0.06

另外依據稅捐處每月所得扣繳申報規定，建立「扶養親屬扣繳稅額表」如下：

所得等級	0人	1人	2人	3人	4人	5人	..
66001~66500	4330	3520	2720	0	0	0	0
66501~67000	4390	3590	2790	0	0	0	0
67001~67500	4460	3650	2850	2050	0	0	0

並提供外國人士(例如：外僑、大陸人士)固定扣繳稅率的輸入，建立「個人扣繳稅率資料表」如下：

計稅類別	人事編號	姓名	扣繳率
2	B020	庫馬克	0.1
5	B261	馬西	0.1
1	E371	朱創新	0.2

對於免稅部分，以人員性質制定「免稅職別與款項代號資料表」如下：

職別	款項代號	款項說明
教官	010	統一薪俸
教官	011	統一薪俸補發
教官	014	統一薪俸晉級差額

以所得性質，在「個人款項明細資料」中的課稅別資料項目區分為免稅或課稅，如下：

學號	作業批號	款項代號	款項名稱	金額	課稅別
934301	09506	911	獎學金	15000	N
940101	09506	913	工讀金	2400	Y

最後應用系統中之「計算稅額」作業功能，匯集當次作業批號點的所有收入，參考上列各項資料表的規則設定，計算出應繳之稅額。演算法則簡略摘要如下：

- ◆ 若「個人款項明細資料」符合免稅範圍（「免稅職別與款項代號資料表」），更新課稅別資料項目='N'
- ◆ 以作業批號取得個人計稅類別 A
- ◆ 讀取個人扶養親屬人數
- ◆ 讀取「個人款項明細資料」，計算課稅別資料項目='Y'之總所得 B
- ◆ 檢查「個人扣繳稅率資料表」，有無符合計稅類

別 A：

- 若有：  
總稅額 = (總所得 B) x 個人扣繳稅率

- 若無：  
If (總所得 B) <= 免預扣最大所得額  
then 總稅額 = 0

else

特定稅率所得 C =

Sum(「個人款項明細」符合「計稅細則資料表」計稅類別 A 下所列款項代號之所得額)

特定稅率所得稅 CT =

Sum(「個人款項明細」符合「計稅細則資料表」計稅類別 A 下所列款項代號之所得額 \* 「計稅細則資料表」計稅類別 A 下所列款項之扣繳稅率)

一般所得稅 DT =

[(總所得 B) - (特定稅率所得 C)] 再對照「扶養親屬人數扣繳稅額表」求得之稅額

總稅額 = CT + DT

If 總稅額 <= 免預扣最大稅額

then 總稅額 = 0

雖然要建立很多規則表格，初期工程不易，但對於目前中央財政窘困，經常調整稅收藉以增加財源的時代，這種可以由用戶隨時調整或新增作業規則內容，而不用再修改程式的方法，無疑是個很好的對應之策，對我們程式人員來說，不僅減少維護工作，也免於承擔責任的風險。

## 6. 流程改善

### 6.1 資料源建立 View

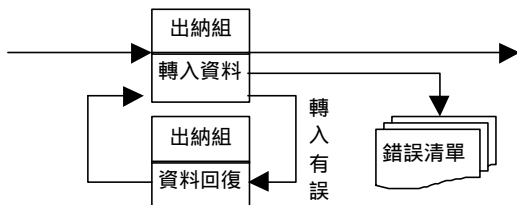
屬於下游端的薪資系統，必須要遵循上游端系統資料的規範，才能正確擷取利用各項上游端的資訊。在轉型期間，由於各系統規格重新規劃，仍在不斷研析修改中，資料定義尚未能明確訂立，此時類似虛擬 Table 的 View 就扮演著很重要的角色，它本身並沒有存放真正的資料，只是儲存實際參考資料的定義，當系統對 View 讀取資料時，透過 View 的定義解釋，才會抓取實際 Table 的資料。

新薪資系統的開發，皆以 View 作為擷取管道，不管實際資料是來自原本舊系統的資料，還是已上線的新系統資料，或是數個 Table 資料項目的結合體，只要修改調整 View 的定義，即可切換源頭，配合其他系統的運作，而不需再對程式作任何修正。

### 6.2 處理程序標準化

在 3.5 節中曾提及往昔的系統開發並沒有全面思考完整的作業程序，導致發生事變時，沒有應變措施，只能直接對資料庫進行緊急處理。這樣的處理模式，最怕的是資料庫處理不當後的追究責任歸屬。

以圖 4 請款支付流程為例，標準的資料轉入作業，應列示錯誤清單，對轉入的資料加以篩選檢核，並解釋錯誤原因。對於資料轉入有誤，欲重新再處理時，亦應提供資料復原相關程序。



另外對於類似的作業程序，儘可能變為共用模組型態，以減少程式維護工程，例圖 6 左半部為舊系統各種款項發放流程，由於每種款項的明細資料檔案名稱與規格不同，從第一道建檔程序至最後的銀行轉帳程序，均須量身訂作，不可被他人再利用，但圖 6 右半部新系統各種款項發放流程，經由資料轉入作業程序，將各種款項變成同一的檔案規格，使統一標準模組 - 銀行轉帳程序，不用撰寫即可套用。今年恰逢本校駐校銀行改組與他行合併，銀行轉帳規格變動，更顯出新系統流程改善後的優異之處。

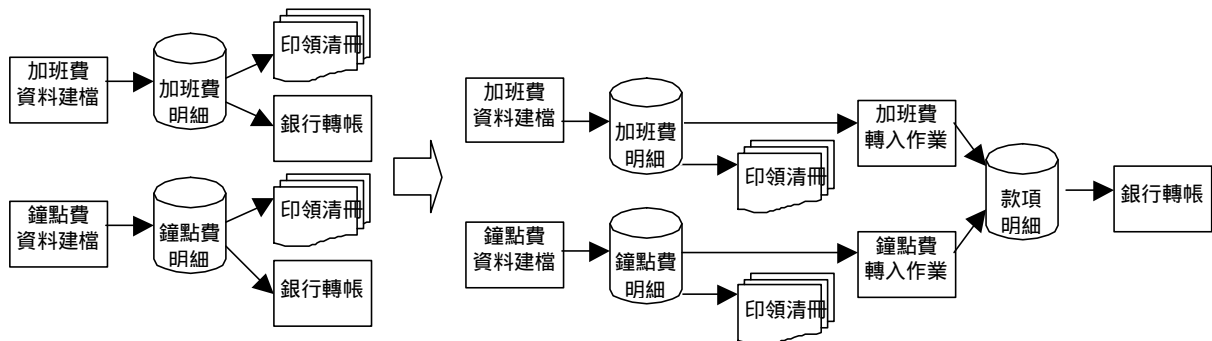


圖 6 舊系統與新系統之款項發放作業流程圖

### 6.3 提供通用匯入匯出管道

學校經常有很多臨時性的集體扣款或發放款，想要依附在薪資下支出，例如颱風賑災捐款、研究獎勵金等，由於性質單純，只是金額的收入或支出差異，與款項名目的不同而已，為了業務執行上的方便與彈性，於是提供了通用的匯入匯出管道作業。委託單位只要能提供符合通用匯入格式之資料，出納組即可依據轉入資料，處理後續的款項轉帳程序。對於匯出管道，則是製訂統一的簡易印領清冊格式，提供報表表頭名稱自訂性，使性質相近的報表，可直接套用。

### 6.4 程式流程改變

在應用系統與資料庫分散於不同主機的架構中，網路傳輸速率與流暢度，變成影響系統好壞的主要因素之一。

在程式撰寫中要有某些觀念的改變，例如讀取資料，應避免像傳統方式一樣，將所要參考的資料全部讀入應用程式中再處理，應該儘量在擷取資料的 SQL 指令中，建立好參考資料間的關聯性，並設定資料條件篩選範圍，使應用系統與資料庫主機間的 I/O 次數與資料流量減少。同理，對大量資料的異動，亦應善用 SQL 集體處理指令來取代以前逐筆處理的方式。

### 7. 新系統效益評估與未來展望

- ◆ 資料結構的改進，瓦解了存在已久的問題，瞬間使出納業務拓展開來，各類補助或代扣款項 (目前已有 377 種款項代號)，均可隨時併新處理，對於津貼的補發與扣回，或是費用的追繳與退還等，亦能以分開款項來詳細表達說明。
- ◆ 通用的匯入匯出管道，提供了隨到隨辦的功能，不用申請需求的提案，即可及時處理代發、代扣業務。
- ◆ 作業規則表格化後，減少程式修改的頻率；處理程序標準化，防止資料庫直接處理的風險，

亦釐清資料權責的關係；流程的改善，降低了程式維護的成本。

- ◆ 開放的關聯式資料庫結構，提供 ODBC 連結管道，讓用戶可依所需自行下載資料，自行編製簡易報表，解決一些臨時應急作業。
- ◆ 應用網際網路，提供個人各項查詢，節省核對紙本與薪資單等的印製，並省略人工分發的作業。即時的 E-mail 通知，更讓所得人永遠獲得第一線訊息。

科技的發達，技術不斷推陳出新，新的系統不僅要能承接以前的歷史包袱，更要能從容面對未來各項新的挑戰，能夠繼往開來，才是本系統再生的本質與目的。