

《資料處理概要》

試題評析

今年的資料處理概要考題著重在資料庫與物件導向技術兩方面，其中關聯式資料庫的Relational Algebra與Relational Calculus就分佔第一、三兩題，這部份對一般考生可能稍微陌生。另外，物件導向開發的Aggregation、Inheritance及Encapsulation分佔第二、四兩題，因是目前熱門的趨勢，也類似去年高考的題型，考生作答應無問題。最後一題整合模式語言(UML)的Sequence Diagram因屬較新的應用領域，所以大多數的考生容易忽略。綜觀本份考題對一般的考生較為困難，預估平均分數約為50分，優秀考生可有70分。

一、(一)說明何謂Relational Calculus？(8分)

(二)說明Relational Algebra與Relational Calculus之主要差異。(8分)

(三)說明何謂關連式完整語言(Relationally Complete Language)。(8分)

答：(一)Relational Calculus：在資料庫管理中是指操作關係(表)的一個非程序性方法。目前關連式微積分包含兩個系列：全域微積分和元組(tuple)微積分。這兩個系列的關連式微積分在數學的領域上是彼此對等且都屬於關係代數，所以使用其中的一個系列就能以目前資料庫中既有的關係為基礎，用公式化的方式來描述所需求的關係。

(二)Relational Algebra與Relational Calculus的差異為：

1.Relational Algebra為一程序性(procedural)查詢語言，比較低階。而Relational Calculus為非程序性(non-procedural)查詢語言，接近人類使用的自然語言，比較高階。

2.Relational Algebra必須明白指出一串操作過程，描述「如何(how)」去做。而Relational Calculus不須明白指出操作過程，只須描述所要的是「什麼(what)」。

(三)Relationally Complete Language：若在一種關聯式查詢語言所能表示的所有查詢，均能用Relational Calculus(關聯式計算法則)表示，則稱此種語言為Relationally Complete Language。

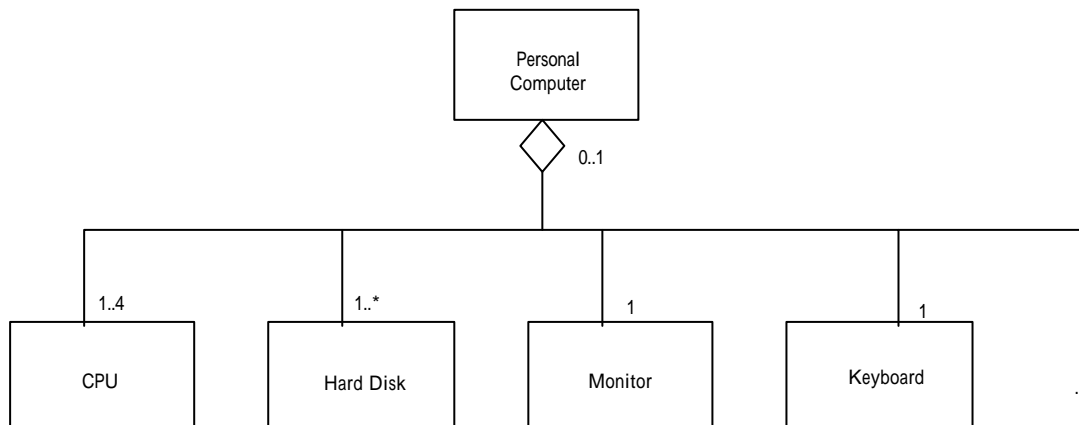
二、在物件導向開發中：

(一)說明何謂Aggregation，舉例說明之。(8分)

(二)說明何謂Inheritance，舉例說明之。(8分)

(三)從軟體重用(Software Reuse)觀點，比較Aggregation與Inheritance。(8分)

答：(一)Aggregation(聚合)：Aggregation(聚合)牽涉到一群不同的物件實例，其中一個是由其他的物件組成。Aggregation表達出成分物件與聚合物件之間的part - of關聯，它是一種比較強的結合關聯形式(因為多了part - of語意)，而且在聚合端是用中空的菱形表示。例如，Personal Computer物件與CPU物件有關，後者是前者的一部份。圖中顯示一台個人電腦(Personal Computer)是由CPU(最多4顆)、硬碟(Hard Disk)、顯示器(Monitor)、鍵盤(Keyboard)、滑鼠等物件所組成。



- (二)Inheritance(繼承)：Inheritance的觀念允許由一個以上的既存類別(稱為Super-class)中去導出另一個新的類別(Subclass)，子類別除了自己本身的屬性與運算方法之外，所有Super-class或更上層所繼承而來的所有屬性與運算方法，都會由Subclass來繼承。例如：若蔬菜為一個類別，則豆類及根菜類可以繼承自蔬菜，並且繼承了蔬菜的名稱、生長季節等等。
- (三)兩者都可提供可達成code reuse的功能。Aggregation將異質性(指不同類別)的物件組合起來，構成了物件導向架構下的橫向關係，而Inheritance則允許由現有類別來導出另一個新的類別，構成了物件導向架構下的縱向關係。

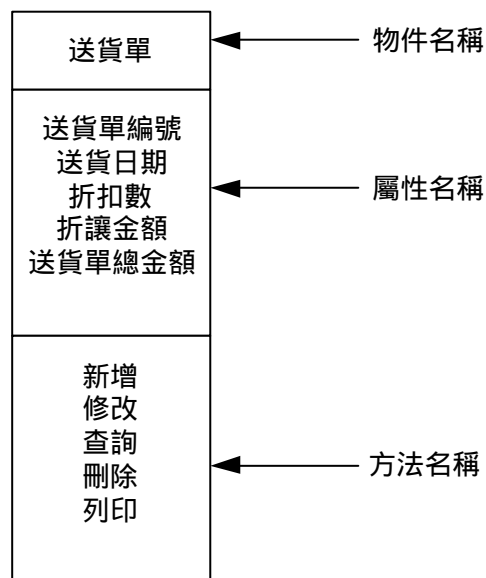
三、(一)說明何謂Relational Algebra？(8分)

- (二)舉出兩個Relational Algebra中之集合運算(Set Operations)。(8分)
- (三)舉出兩個Relational Algebra中之運算(Operations)，但不是集合運算。(8分)
- (四)在Relational Algebra中，何謂完整集合(Complete set)？(8分)

- 答：**(一)Relational Algebra(關連式代數)：係指一個允許關係(表)操作的規則和運算子。Relational Algebra通常用於敘述的運算子有：Select(或Restrict)、Project、Product、Union、Intersect、Difference、Join和Divide等。在關連式資料庫中，Relational Algebra是用於以目前既有的關係為基礎來發展建立新關係的程序。
- (二)在聯集(Union)、交集(Intersect)、差集(Difference)、卡式積(Product)四者中可任擇其二即可。
- (三)在限制(Select或Restrict)、投影(Project)、合併(Join)、除法(Divide)四者中可任擇其二即可
- (四)完整集合(Complete set)是指由五個基本運算子Restrict、Project、Union、Product、Different所成的集合。

四、在物件導向開發中，說明何謂封裝(Encapsulation)，並舉例說明之。(10分)

- 答：**Encapsulation(封裝)在物件導向開發中，是指將某物件的各種屬性(特性)及功能(方法或行為)加以包裝，以建立一個實質上是一個黑箱的物件，但其內部結構依然保有私密性且只能藉由其他物件經由清楚的定義介面透過訊息傳遞存取。Encapsulation封裝將物件的實作細節隱藏，使其與外界環境隔離，而只允許該物件所包含之操作修改其資訊，稱為資訊隱藏(Information Hiding)。物件之操作執行物件之行為，並可修改物件之資訊。使用物件時，僅需知道物件提供何種操作，而不需知道其內部之資訊或行為是如何表達或執行。
- 例：在下圖的送貨單物件中，使用者只要經由送貨單所提供的修改方法，即可更新原有內容，而不必知道送貨單內部的詳細資料結構及如何製作該方法。



細資料結構及如何製作該方法。

五、目前常用的整合模式語言(Unified Modeling Language)中，說明何謂Sequence Diagram，舉例說明之。(10分)

- 答：**Sequence Diagram(循序圖)用以描述系統運作時，物件間的互動行為且著重以時間為主軸的處理程序。下圖為一個接電話的Sequence Diagram範例，圖中縱軸是時間軸，橫向箭頭是每一個訊息，在此Sequence Diagram上，除了看到時間順序之外，還看到一些限制的條件，如本例中電話訊號進來和電話響之間最多不能超過多少時間。

