

101年特種考試地方政府公務人員考試試題

等 別：三等考試

類 科：資訊處理

科 目：資料庫應用

一、假設有一公司之關聯式資料庫 (Relational database) 包含三個表格 (Table)，其綱要 (Schema) 如下，有底線之屬性 (Attribute) 為該表格之主鍵 (Primary key)，其中“經理編號”是該名經理的“員工編號”：

員工 (員工編號，員工姓名，部門編號，上班開始日，職稱編號，薪水)

部門 (部門編號，部門名稱，經理編號，所在地編號)

所在地 (所在地編號，城市，國家)

(一)請寫出關聯代數 (Relational algebra) 來列出所有在紐約市工作的員工編號、員工姓名、職稱編號和部門名稱。

(二)請用SQL的Subquery敘述 (Statement) 來列出員工編號“E101”此員工的經理姓名。

(三)下列SQL敘述有錯，請說明其錯誤的原因。

```
Select a. 部門編號, a. 職稱編號, a. 員工姓名, avg(a. 薪水)
```

```
From 員工a, 部門b
```

```
Where a. 部門編號=b. 部門編號
```

```
Group by a. 部門編號, a. 職稱編號
```

```
Having max(a. 薪水)>10000
```

```
Order by a. 部門編號, a. 職稱編號
```

【擬答】：

(一) π (員工編號, 員工姓名, 職稱編號, 部門名稱) (σ (員工. 部門編號=部門. 部門編號 AND 部門. 所在地編號=所在地. 所在地編號 AND 所在地. 城市=' 紐約市') (員工×部門×所在地))。

(二)SELECT A. 員工姓名

FROM 員工 A

WHERE A. 員工編號 = (

SELECT B. 經理編號

FROM 員工 B, 部門

WHERE B. 部門編號 = 部門. 部門編號 AND

B. 員工編號 = ' E101') ;

(三)SQL語法GROUP BY表分群統計，SELECT後面只能接<分群屬性>或聚合函數；上述SQL中a. 員工姓名非分群屬性，故「SELECT a. 員工姓名」錯誤。

二、資料庫設計的概念結構設計中，有一種技術稱為“視界的整合” (View integration approach)，請列出並說明此技術的設計步驟。

【擬答】：

先依每個使用者 (或應用程式) 需求，設計概念綱要 (視界，可多個)，再經下列步驟整合成單一的整體概念綱要的設計方法，稱視界整合法，整合後個別視界可重新建構成資料庫的外部綱要。

(一)確認不同綱要之間的相對關係與衝突：綱要構件常代表真實世界中的某個概念，個別綱要表示法可能發生下列衝突，在視界整合前需加以確認。

1. 命名衝突：2 大類型。

(1)同義字：概念相同但名稱不同的實體或關係類型。

(2)同音異義字：概念不同但名稱相同的實體或關係類型。

2. 型態衝突：相同的概念可能用不同的構件 (如實體或屬性) 來表達。

3. 定義域衝突：相同屬性可能有不同的定義域。

4. 限制之間的衝突：兩個綱要可能設定不同的限制，如相同實體類型卻選用不同主鍵。

公職王歷屆試題 (101 地方政府特考)

(二)修改視界使彼此互相配合：即修改某些綱要，以解決上述衝突。

(三)視界的合併：即合併個別綱要來建立整體綱要，並指定視界與整體綱要之間的對映關係；對大型資料庫設計而言，可能牽涉數百個實體與關係，故為耗時且很難達成的步驟，只能儘量協商，找出最合理的解決方案。

(四)重組：簡化和結構化以便移除任何重複性或不必要的複雜性。

三、有一種分散式結構性資料儲存的技術稱為“NoSQL”，請列出並說明NoSQL資料庫主要的四個特徵。

【擬答】：

NoSQL，即 Not Only SQL，意指下一代非關聯式資料庫技術，以支援分散式儲存，並具備開放原始碼與容易進行水平式擴充 (Horizontally scalable) 等精神，有四大特徵。

(一)無綱要 (Schema-free)：網際網路多為半結構性 (如 HTML 或 XML 文件) 與非結構性 (如影音或圖片) 資料，通常無法事先明確定義實體間的關係與結構，故 NoSQL 資料庫採取無綱要策略，在資料庫提供的資料結構基礎上，允許直接儲存並取用資料實體，無須事先定義資料綱要。

(二)高延展性 (High scalability)：直接以開放標準之大眾化硬體設備 (避免 Vendor Lock 現象)，快速且低成本地擴充儲存容量與處理能力，稱水平式擴充，可滿足雲端計算的需求，故 NoSQL 資料庫必須具備自動資料複製 (Replication)、切割與轉移等能力。

(三)簡單的存取介面：為配合無綱要策略，NoSQL 資料庫依其提供的資料結構基礎，直接以支援開發語言 (如 Java 或 C++) 的 API 取代 SQL 做為資料存取介面，故 SQL 中複雜功能 (如 JOIN)，通常不提供。

(四)最終的一致性 (Eventually consistent)：NoSQL 以分散式資料複製技術 (一份資料可複製、存放在多個不同地理位置的資料中心)，以提升系統的可用性 (Availability，保證資料一定可以被存取)；多個複本之間則採用最終一致性策略，即資料更新時，僅保證在某段夠長時間內，逐步更新到所有複本，而非立即同步。

四、假若有一家房屋仲介公司要建置一個資料庫，其資料需求如下，請畫一個實體關係圖 (Entity-relationship diagram) 以充分表達其資料的需求。

1. 房屋：每一間房屋有一個房屋編號、街道名稱、房屋種類、訂價。每一個房屋的房屋編號是唯一的。
2. 買主：每一位買主有一個買主編號、買主姓名、買主住址、數個手機號碼。每一位買主的買主編號是唯一的，一位買主可購買數間房屋，每一次購屋皆記錄其購屋日期、最後成交價、契約書編號、代書編號、交屋日期。
3. 屋主：每一位屋主有一個屋主編號、屋主姓名、屋主住址、數個屋主電話。每一個屋主的屋主編號是唯一的，一個屋主可同時提供數間房屋託售，每一次託售皆記錄其託售時間、契約書編號、佣金成數。

【擬答】：

