

資料庫系統 期中考(1~8 章) A 卷重點整理

一、解釋名詞

1. DBMS: 資料庫管理系統，是一套讓使用者能夠新增、更新、管理取用及分析資料庫內容的工具、機能，以及介面的組合。
2. data independence: 資料獨立性。當存取方式的改變時，或者是資料庫儲存結構改變時，都不會去影響到原有的應用程式。
3. foreign key: 外部鍵，它是資料表中的其中一個欄位，其屬性必須與同一個表格或另一個表格的主鍵之屬性的 domain 一樣，進而建立表格間的關聯性。
4. primary key: 主鍵，從多個候選鍵中選出，是用來辨識紀錄的欄位。它具有唯一性，且不許重複。
5. candidate key: 候選鍵，在每一個關聯表至少擁有一個候選鍵，不只要滿足唯一性，還需要滿足最小性。
6. product-compatibility: 兩個關聯表 R1、R2 的表頭名稱不相交。
7. union-compatibility: 兩個關聯表 R1、R2 的屬性及必須具備相同的維度(Degree)，且 R1、R2 的相對屬性都必須定義於相同的 Domain 上。
8. closure: 封閉性，關聯表經過關聯式代數的運算子運算之後，其結果仍然還是一個關聯表。
9. relational algebra: 關聯式代數，是一種較低階的、程序性的、規範性之抽象的查詢語言，它是來描述如何產生查詢結果的步驟。
10. relationally complete: 關聯完整性，若有一查詢語言具備與 relational algebra 或者是 relational calculus 相同的能力時，則稱此語言具“關聯完整性”。
11. domain (of attribute) : The set of scalar value with the same type.
12. view (from relations) : 它定義於其他基底關聯表之上的虛擬資料表，它的目的是為了限制使用者去存取、讀取某些欄位及其資料。
13. snapshots (from relations) : 這是具名的導出關聯表，在同一個時間點的關聯表內容，不過它有真正的儲存資料，屬於一種唯讀資料表。
14. referential integrity rule: 關聯表中的所有外部鍵的值，都必須能參考到另一個關聯表的主鍵值。
15. entity integrity rule: 關聯表的主鍵不可以是空值。
16. intra-file clustering: Data item for a single stored file.
17. inter-file clustering: Data items in two or more files are stored together.
18. dense indexing: 密集索引，每一個索引檔中的區段指標皆會對應到資料檔中的每一筆資料。
19. nondense indexing: 非密集索引，每一個索引檔中的區段指標皆會對應到資料檔中的某一區塊的起始位置。所以並非所有資料檔中的每筆資料，在索引檔中都有一個搜尋鍵可以對應。
20. embedded SQL: 嵌入式 SQL，可以使用特殊語法將 SQL 指令包含在通用用途程式語言的程式碼檔案，例如: VB、C、C++、Java 等，這些語言稱為宿主語言。

二、問答題

1. Why database is superior to traditional file structure? i.e. What are the advantages of database?
(Why we need a database(DB) instead of files?)

Ans:

- (1) 減少資料重複性
- (2) 資料的完整性

- (3)資料的獨立性
- (4)資料的共享性
- (5)資料的一致性
- (6)有安全的限制與維護

2. What is union-compatibility? Which relational algebra operations have the property of union-compatibility?

Ans:

- (1)兩個關聯表 R1、R2 的屬性及必須具備相同的維度(Degree)，且 R1、R2 的相對屬性都必須定義於相同的 Domain 上。
- (2)Union(聯集)、Intersection(交集)、Difference(差集)

3. What are the responsibilities of the Database Administrator (DBA)?

Ans:

- (1)定義內部層
- (2)定義概念層
- (3)建立備用及修復
- (4)定義安全及限制
- (5)與使用者聯繫
- (6)監視運作並回復要求

4. What is the B-tree structure? What is the hashing function? What is the pointer chains? Please, describe the differences between these structures.

	B-tree structure	hashing function	pointer chains
定義	是一個多層次的索引，是 DBMS 中最常見的儲存結構	提供快速直接存取一個特定儲存紀錄，一個 key 值對應 address 函數	是一個父子結構，資料已經排列過的，每筆資料都會有一頭指向它，而它會指向下層的資料或指向同一層的不同筆資料。
優點	(1)可進行動態調整保持索引樹的平衡 (2)它是 notable	(1)較快找到資料 (2)節省空間	(1)插入集刪除資料比 index 簡單且有效率 (2)使用的空間比 index 少
缺點	(1)索引表占空間 (2)搜尋時很花時間	(1)可能發生碰撞 (2)logic 的順序與實際的順序不一致	(1)因為它是使用 sequential access 找到某一筆資料，所以搜尋速度較慢 (2)要建立出一個新的 pointer chain 需要資料庫的 reorganization

5. Describe the differences between relational algebra and relational calculus.

relational algebra	relational calculus
程序性語言	非程序性語言
近程式語言，較低階	近人類語言，較高階
必須指出運算順序(How)	不需指出運算順序， 只需說明查詢所需要的是什麼(What)
要提供基本運算	不需要提供基本運算
可直接實作	通常轉為關聯式代數來執行

6. For a n -attributes table, how many possible indexes could be established?

Ans: $2^n - 1$

7. For a n -attributes table, how many possible hashing functions could be established? Why?

Ans:

(1) 1

(2) 因為 hashing function 有可能發生碰撞，若建立 2 個，有些資料可能無法被包含，與 nondense 相似，碰撞機會也可能更多。

8. Why relational operation: Intersection is not a primitive operation?

Ans:

Intersection 可由聯集和差集組合運算，例如: $A \cap B = A \cup B - (A - B) - (B - A)$