

一、在 SQL 語言中 select 指令可以包含下列子句(clause): select, from, where, group by, having 等。說明這五個子句的意義及執行順序。(25 分)

(一) 意義:

(1) select: 選擇所要見到的屬性列, 類似關聯式代數中的投影

(project) 運算。

(2) from: 將參照的表格串列進行某些運算, 以供其他子句使用。

(3) where: 將 from 子句提供的表格做限制運算, 去掉不符合條件的值組。

(4) group by: 將依照屬性串列分成群重新安排, 以供其他子句使用。

(5) having: 將群中不符合條件者除去, 以供其他子句使用。

(二) 執行順序: from → where → group by → having → (order by) → select。

只有 select 和 from 是必要的。

二、試舉一例說明第三正規化(3rd Normal Form)的意義。(20 分)

調97.2

、試舉出五種可以提高資料庫執行效率的方法。(25 分)

透過適當設計的資料表和索引即可提高 DB 運作的效率:

(一) 索引的選定和建立: 依據以下3點找出最適合的屬性來建立索引。

(1) 每一個建立的索引都會被用於某些查詢句的最佳化。

(2) 一個索引最好能被多個查詢句使用到。

(3) 如果使用到一索引的查詢次數遠少於對該索引的修改次數, 慎重考慮是否要建立該索引。

ex: SELECT pName, name
FROM Product P, Author A
WHERE P.pNo = A.pNo AND
catalog = 'CD';

檢視 where 子句, find 有3個屬性
可建 index. Product.pNo, Author.pNo
- Product.catalog

↓
群聚索引

↓
一般索引

(二) 調整資料庫綱目: 3種方式

(1) 反正規化: 減少 JOIN, 故提高效率

→ T(pNo, tNo, amount, salePrice, invNo)

→ T1(pNo, tNo, amount, salePrice) } Need Join
T2(invNo, tNo)

Q: 列出售價超過 300 之交易商品的 pNo, tNo, salePrice, 則 T 效率佳。

(2) 資料垂直分割: 若付款相關 attribute 少用, 可分解如下:

表
T(tNo, transMid, transtime, tpNo, bankId, bankName, CardId)
→ T1(tNo, transMid, transtime)
→ T2(tpNo, bankId, bankName, CardId)

(3) 資料水平切割: 假設有很多查詢只針對書籍, 則可將 Product 水平切割成 Book 和 Other Product 2 資料表, 將 Product 設為 View

如右: CREATE VIEW Product AS(
SELECT *
FROM Book)
UNION
SELECT *
FROM otherProduct));

(三) 查詢句改寫:

(1) 有些 DBMS 不會試圖將巢狀查詢句改寫成一般查詢句再處理, 故能用一般查詢句完成, 就不要用巢狀查詢句。

(2) DISTINCT 費時, 盡量避免。

(3) SQL 敘述盡量不置於 loop 內, 以免要下達大量 SQL 敘述, 而效率變差。

(4) 少用 GROUP BY: 處理費時, 有些查詢的 GROUP BY 非必需。

- 四、假設國家檔案局開放民眾可以上網查詢與民眾相關的各機關公文檔案，試設計一個資料庫滿足這一個需求。注意：設計時應考慮民眾查詢的方便性、身分的適當性、公文的機密性、公文內容的不可修改性以及其他你覺得重要的特性。(30分)

使用者和 DB 關係圖：



(1) 一般 user 透過 DBMS 存取 DB。

(2) DBMS 驗證帳號、密碼、權限。

ex: Oracle, SQL server

查詢方便性：透過 DBMS 存取 DB

身分適當性：DAC → GRANT, REVOKE

MAC → 軍事級

RBAC → 最小權限

公文不可修改性：使用雜湊函數 MD5 驗證完整性

其他特性：

安全機制：

① 減少 repeat

① Data Mask

② 避免不一致

② DLP

③ 資料共用

③ DRM

④ 標準強迫推行

⑤ 確保安全性

⑥ 維持整合性