Ch14 :

平面文件儲存 (Flat-file storage) : 是一種簡單的數據存儲方法，所有數據都保存在單個文件或多個無結構的文本文件中，這些文件通常不具有像關聯數據庫那樣的複雜性。數據存儲在平面文件中，通常以純文本或簡單的表格形式呈現，如 CSV（逗號分隔值）文件。

主要特點：

1. 單一結構：所有數據都保存在一個文件中，這個文件的結構通常是平坦的、無層次的。每一行表示一條記錄，字段之間可能由逗號、空格、或其他分隔符隔開。
2. 簡單易用：平面文件存儲非常直觀和簡單，任何文本編輯器都可以查看和編輯數據。這也是它的主要優勢之一。
3. 靈活性低：由於缺乏數據結構或約束，平面文件存儲在處理大量數據或複雜查詢時表現不佳，並且容易出現冗餘數據和數據不一致的問題。
4. 應用場景：
   * 小型應用程序：當數據量較小，且不需要複雜的查詢功能時，平面文件是一個合適的選擇。
   * 數據交換：平面文件常用作不同系統之間數據交換的格式，特別是 CSV 文件在報表和導入導出數據上應用廣泛。
   * 配置文件：很多應用程序使用平面文件來保存配置數據，這些文件格式通常是 XML、JSON 或 YAML。

優點：

* 簡單和低成本：無需額外的數據庫管理系統，維護成本低。
* 可移植性強：平面文件可以輕鬆地在不同的系統之間傳遞。

缺點：

* 不適合大規模數據管理：在處理大數據量、複雜查詢、數據一致性和多用戶操作時，Flat-file storage 很容易暴露性能和數據完整性問題。
* 缺乏數據完整性和約束：沒有像關聯數據庫中的主鍵或外鍵約束來確保數據的一致性和完整性。

數據庫（Database）是指一種有組織的數據集合，專門設計用來高效存儲、管理和檢索數據。數據庫可以讓用戶和應用程序快速、方便地存取大量的結構化或非結構化數據，從而支持各種應用程序的運行需求。

數據庫的核心概念

1. 數據庫管理系統（DBMS）：
   * DBMS 是用來管理數據庫的軟件，它提供數據的存儲、查詢、更新、刪除等基本操作。
   * 常見的 DBMS 包括 MySQL、PostgreSQL、SQL Server、Oracle 和 SQLite 等。
2. 數據模型：
   * 數據庫根據數據的組織方式，可以使用不同的數據模型來表示數據之間的關係。
   * 常見的數據模型包括：
     + 關聯模型：數據存儲在表格（表、行、列）中，表之間可以通過鍵（如主鍵、外鍵）來建立關聯。例如：SQL 數據庫。
     + 層次模型：數據按照層次結構組織，數據之間呈現父子關係。
     + 網絡模型：類似於層次模型，但允許一個節點有多個父節點。
     + 文檔模型：用於存儲半結構化數據（如 JSON、XML 文檔）的 NoSQL 數據庫。
     + 鍵-值模型：存儲鍵值對，用於高效的查找和存取操作。
3. SQL vs NoSQL：
   * SQL（Structured Query Language）數據庫是基於關聯模型的數據庫，使用 SQL 查詢語言進行數據操作，強調數據的結構化和一致性。例如：MySQL、PostgreSQL、SQL Server。
   * NoSQL數據庫適合存儲非結構化或半結構化數據，提供靈活的數據模型和可擴展性。例如：MongoDB、Cassandra、Redis。

數據庫的基本操作

1. 插入（Insert）：將新數據添加到數據庫中。
2. 查詢（Query）：檢索存儲在數據庫中的數據。SQL 查詢語句例如 SELECT \* FROM users; 會返回整個 users 表中的數據。
3. 更新（Update）：修改現有數據。例如 UPDATE users SET age = 30 WHERE id = 1; 將修改 id 為 1 的用戶的年齡。
4. 刪除（Delete）：刪除數據。例如 DELETE FROM users WHERE id = 1; 將刪除 id 為 1 的記錄。

數據庫的特性

1. ACID：關聯數據庫強調的四個特性，用來保證數據的完整性和一致性：
   * A（Atomicity, 原子性）：操作要么全部成功，要么全部失敗，保證數據的完整性。
   * C（Consistency, 一致性）：數據在操作前後保持一致，符合數據庫的規則。
   * I（Isolation, 隔離性）：多個事務同時運行時，互不干擾。
   * D（Durability, 持久性）：一旦事務提交，數據將永久保存，不會因系統崩潰丟失。
2. CAP 理論：分布式數據庫中，必須在一致性（Consistency）、可用性（Availability）和分區容錯性（Partition Tolerance）之間做出權衡，最多只能滿足兩者。
   * 一致性（C）：所有節點的數據保持一致。
   * 可用性（A）：每個請求都能獲得響應。
   * 分區容錯性（P）：數據在網絡分區中也能繼續運作。

數據庫類型

1. 關聯數據庫：
   * 使用表來表示數據及其關係，通過 SQL 進行操作。
   * 常見的有 MySQL、PostgreSQL、SQL Server、Oracle 等。
2. NoSQL 數據庫：
   * 提供靈活的數據模型，適合存儲非結構化或大規模數據。
   * 類型包括文檔型（如 MongoDB）、鍵值型（如 Redis）、列族型（如 Cassandra）、圖形型（如 Neo4j）。
3. 嵌入式數據庫：
   * 數據庫集成到應用程序中，適合桌面或移動設備，如 SQLite。
4. 雲數據庫：
   * 託管在雲端的數據庫服務，提供可擴展的數據存儲和管理，如 AWS RDS、Google Cloud SQL。

數據庫應用場景

* Web 應用：網站、電商平台使用數據庫存儲用戶信息、產品資料等。
* 數據分析：數據庫提供高效的數據查詢和統計，支持大數據分析和商業智能應用。
* 金融系統：金融交易記錄、銀行賬戶管理等都依賴高可靠性的數據庫。

關聯數據庫 (Relational database, RDB) : 是一種基於關聯模型的數據庫。RDB 使用表格（表、行、列）來存儲和管理數據，表之間通過共同的字段（鍵）建立關聯，以實現數據的組織、查詢和管理。

RDB 的關鍵特徵

1. 關聯模型：數據以表格的形式組織，每個表由行和列組成，行表示一條數據記錄，列表示數據的屬性。表之間的關係通過鍵（如主鍵、外鍵）來表達。
2. SQL（Structured Query Language）：關聯式數據庫使用 SQL 作為主要的查詢語言，用於數據插入、查詢、更新和刪除。SQL 讓用戶能夠方便地對數據進行操作。
3. 數據一致性：RDB 強調數據一致性，通過 ACID 原則（原子性、一致性、隔離性、持久性）來保證數據操作的安全性和可靠性。
4. 鍵：
   * 主鍵（Primary Key）：每個表中的唯一標識符，用來唯一確定表中的每一行。
   * 外鍵（Foreign Key）：用來引用其他表的主鍵，從而建立表與表之間的關聯。

RDB 的優點

1. 數據結構化：數據以嚴格的結構存儲，易於組織和管理。
2. 數據一致性和完整性：基於 ACID 原則，保證數據在操作過程中保持一致和完整。
3. 靈活的查詢：使用 SQL，可以執行複雜的查詢、過濾和聚合操作。
4. 標準化：關聯式數據庫有著標準的設計和操作模式，使用廣泛，便於維護和擴展。

常見的 RDBMS（關聯式數據庫管理系統）

* MySQL：開源且流行的 RDBMS，適合 Web 應用。
* PostgreSQL：功能強大的開源 RDBMS，支持更多的數據類型和高級功能。
* Oracle：商業級 RDBMS，適合大型企業和高要求的業務系統。
* Microsoft SQL Server：由微軟開發，廣泛應用於商業環境。

RDB 的應用場景

* 電子商務平台：存儲用戶、產品、訂單等結構化數據。
* 企業資源管理系統（ERP）：管理企業內部數據，如財務、物流、人事等。
* 銀行和金融系統：保障交易的安全性和一致性。

每秒輸入/輸出 (Input/Output Operations Per Second, IOPS) : 是一個用於電腦儲存裝置效能測試的量測方式，可以視為是每秒的讀寫次數。和其他效能測試一様，儲存裝置製造商提出的IOPS不保證就是實際應用下的效能。 IOPS可以用應用程式來量測。