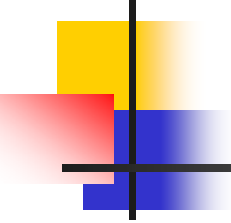




# 資料並行性與資料一致性

---



# 資料並行性與資料一致性

---

- 資料並行性(Data Concurrency)
- 資料一致性(Data Consistency)



# 異動交易的四大特性

---

- 單元性(Atomicity)
- 一致性(Consistency)
- 隔離性(Isolation)
- 持續性(Durability)



# 單元性(Atomicity)

- 一個異動交易(Transaction) 中包含了數個以上資料庫動作(Actions)。
- 整個異動交易必須視為一個不可分割的單元。也就是說整個異動交易的所有動作必須全部做完，若異動交易的中間過程有任何錯誤產生時，必須撤回(Rollback) 到未執行異動交易前的原點，也就是整個異動都不做。



# 單元性的例子

你到提款機前領錢，領錢的全部過程即是一個異動交易，其主要可分為二大部分：

- (1) 銀行從你的帳戶內扣款。
- (2) 自動提款機給錢。若這異動交易過程中，因某些原因，例如：帳戶內餘額不足無法扣款，而無法完成全部異動交易的動作，則整個異動交易會被撤回。



# 一致性(Consistency)

---

- 異動交易(Transaction) 在處理過程前後，必須使資料庫處理保持一致的狀態，雖然異動交易的過程中未必保持一致性。



# 一致性的例子

甲先生轉帳 1000 到乙先生的帳戶，這異動交易包含二大部分：<1> 銀行由甲帳戶先扣款 1000 元，<2> 銀行將 1000 元加入乙帳戶內。在這異動交易前的甲與乙帳戶金額總和一定會和異動交易後的甲與乙帳戶內的金額總和相同。雖然在這異動的第一部分動作（銀行由甲帳戶中先扣款 1000 元）做完後，資料庫正處於不一致狀態（因為金額總和少了 1000 元），但等到第二部分動作（銀行將 1000 元加入乙帳戶），資料庫又會保持一致性的狀態。



# 隔離性(Isolation)

---

- 異動交易(Transaction) 執行的過程中，所用到的資料或所產生的中間結果，都不能透露給其它異動交易讀取或更改，稱之為異動交易的「隔離性」。





# 隔離性的例子

甲欲由自動提款機中提款 1000 元而乙欲由自己帳戶中轉 5000 元到甲的帳戶中，若甲的異動交易先執行，則乙的異動交易必須等待甲的異動交易委付確認 (Commit) 後，才能執行將 5000 元加入甲的帳戶內的動作。其中乙帳戶扣款的動作，可與甲的異動交易同時並行執行，不必等待。



# 持續性(Durability)

---

- 如果異動交易執行過程都完全正常，且異動交易被委付確認(Commit)後，無論事後系統發生錯誤，則當系統排除錯誤，恢復正常時，原異動交易被委付確認的資料仍必須存在。