



# 常見的並行控制問題

---



# 常見的並行控制問題

---

- 遺失更新問題(Lost Update Problem)
- 未委付確認相依問題(Uncommitted Dependency Problem)
- 不一致分析問題 (Inconsistent Analysis Problem)

# 遺失更新問題(Lost Update Problem)

時間點 <sup>↵</sup>	異動交易 A <sup>↵</sup>	異動交易 B <sup>↵</sup>
<u>t1</u> <sup>↵</sup>	Read X→a1 <sup>↵</sup>	<sup>↵</sup>
<u>t2</u> <sup>↵</sup>	<sup>↵</sup>	Read X→a2 <sup>↵</sup>
<u>t3</u> <sup>↵</sup>	<sup>↵</sup>	a2+5→a2 <sup>↵</sup>
<u>t4</u> <sup>↵</sup>	a1+20→a1 <sup>↵</sup>	<sup>↵</sup>
<u>t5</u> <sup>↵</sup>	Write a1→X <sup>↵</sup>	<sup>↵</sup>
<u>t6</u> <sup>↵</sup>	<sup>↵</sup>	Write a2→X <sup>↵</sup>
<u>t7</u> <sup>↵</sup>	Commit <sup>↵</sup>	<sup>↵</sup>
<u>t8</u> <sup>↵</sup>	<sup>↵</sup>	Commit <sup>↵</sup>

# 未委付確認相依問題(Uncommitted Dependency Problem)

時間點 <sup>↵</sup>	異動交易 A <sup>↵</sup>	異動交易 B <sup>↵</sup>
t1 <sup>↵</sup>	Read X → a1 <sup>↵</sup>	↵
t2 <sup>↵</sup>	a1+10 → a1 <sup>↵</sup>	↵
t3 <sup>↵</sup>	Write a1 → X <sup>↵</sup>	↵
t4 <sup>↵</sup>	↵	Read X → a2 <sup>↵</sup>
t5 <sup>↵</sup>	↵	a2+5 → a2 <sup>↵</sup>
t6 <sup>↵</sup>	Rollback <sup>↵</sup>	↵

# 不一致分析問題 (Inconsistent Analysis Problem)

時間點 <sup>↵</sup>	異動交易 A <sup>↵</sup>	異動交易 B <sup>↵</sup>
<u>t1</u> <sup>↵</sup>	Read x → a1 <sup>↵</sup>	↵
<u>t2</u> <sup>↵</sup>	↵	Read x → a2 <sup>↵</sup>
<u>t3</u> <sup>↵</sup>	a1-10 → a1 <sup>↵</sup>	↵
<u>t4</u> <sup>↵</sup>	↵	Sum+a2 → Sum <sup>↵</sup>
<u>t5</u> <sup>↵</sup>	Write a1 → X <sup>↵</sup>	↵
<u>t6</u> <sup>↵</sup>	Read Y → a3 <sup>↵</sup>	↵
<u>t7</u> <sup>↵</sup>	a3+10 → a3 <sup>↵</sup>	↵
<u>t8</u> <sup>↵</sup>	Write a3 → Y <sup>↵</sup>	↵
<u>t9</u> <sup>↵</sup>	↵	Read Y → a4 <sup>↵</sup>
<u>t10</u> <sup>↵</sup>	↵	Sum+a4 → Sum <sup>↵</sup>
<u>t11</u> <sup>↵</sup>	↵	Commit <sup>↵</sup>
<u>t12</u> <sup>↵</sup>	Commit <sup>↵</sup>	↵



# 排程(Schedule)

---

- 序列化排程(Serial Schedule)
- 非序列化排程(Non-serial Schedule)
- 可序列化排程(Serializable Schedule)



# 序列化排程 (Serial Schedule)

---

- 所謂序列化排程 (Serial Schedule) 是指一個異動交易 (Transaction) 緊接著另一個異動交易循序地執行，而且不相互交叉執行。



# 序列化排程對於資料庫一致性的意義

- 一個一致性的資料庫經過任何序列化排程運作後，該資料庫還是合乎一致性的。





## 非序列化排程(Non-serial Schedule)

- 指多個異動交易(Transaction) 同時交錯地執行，然而個別異動交易內各動作(Action)，仍遵守它在該異動交易內的先後順序。



## 可序列化排程(Serializable Schedule)

---

- 若一非序列化排程(Non-serial Schedule)，其執行結果與任一個序列化(Serial Schedule)的執行結果相同，則該非序列化排程稱為“可序列化排程”(Serializable Schedule)。

# 可序列化排程的例子

時間點 <sup>⊖</sup>	異動交易 TA <sup>⊖</sup>	異動交易 TB <sup>⊖</sup>
<u>t1</u> <sup>⊖</sup>	Read X→a1 <sup>⊖</sup>	⊖
<u>t2</u> <sup>⊖</sup>	a1 * 2 → a1 <sup>⊖</sup>	⊖
<u>t3</u> <sup>⊖</sup>	Write a1 → X <sup>⊖</sup>	⊖
<u>t4</u> <sup>⊖</sup>	⊖	Read X → a2 <sup>⊖</sup>
<u>t5</u> <sup>⊖</sup>	Read Y → b1 <sup>⊖</sup>	⊖
<u>t6</u> <sup>⊖</sup>	⊖	a2 + 5 → a2 <sup>⊖</sup>
<u>t7</u> <sup>⊖</sup>	b1 * 2 → b1 <sup>⊖</sup>	⊖
<u>t8</u> <sup>⊖</sup>	⊖	Write a2 → X <sup>⊖</sup>
<u>t9</u> <sup>⊖</sup>	Write b1 → Y <sup>⊖</sup>	⊖
<u>t10</u> <sup>⊖</sup>	⊖	Read Y → b2 <sup>⊖</sup>
<u>t11</u> <sup>⊖</sup>	⊖	b2 + 5 → b2 <sup>⊖</sup>
<u>t12</u> <sup>⊖</sup>	⊖	Write b2 → Y <sup>⊖</sup>



# 鎖定(Locks)

---

- 在多使用者的資料庫(Multi-user Database) 中，一般都採取某些資料鎖定(Locks) 來解決並行處理(Concurrency) 中資料的一致性(Consistency) 與整合性(Integrity) 問題。



# 主要的兩種鎖定(Lock)

---

- 互斥鎖定(Exclusive Lock)
- 共享鎖定(Share Lock)



# 互斥鎖定(Exclusive Lock)

- 它禁止相關資源共享。如果一異動交易(Transaction) 以互斥鎖定方式封鎖某資源，則僅有該異動交易(Transaction) 被允許更新該資源的資料，直到該異動交易釋放對該資源的鎖定。



# 共享鎖定(Share Lock)

---

- 它允許相關資源可以被共享(Share)，例如：多個使用者可以讀取(Read) 相同的資料。多個異動交易(Transaction) 可以對同一資源獲得共享鎖定。

# 死結的例子

異動交易 T1	時間點	異動交易 T2
SQL> update scott.emp 2 set sal = sal +100 3 where empno=7844;↓ ↓ 更新了 1 列	t1	
	t2	SQL> update scott.emp 2 set sal = sal -200 3 where empno = 7839;↓ ↓ 更新了 1 列
SQL> update scott.emp 2 set sal = sal -300 3 where empno= 7839;↓ ↓ (等待 T2 所鎖定的資料)	t3	
	t4	SQL> update scott.emp 2 set sal = sal +300 3 where empno = 7844;↓ ↓ (等待 T1 所鎖定的資料)
update scott.emp↓ * ↓ ERROR 在行 1:↓ ORA-00060: 等候資源時偵測 到一個死結	t5	





# 兩種不同層次一致性

---

- 敘述層級讀取一致性(Statement-level Read Consistency)
- 異動交易層級讀取一致性(Transaction-level Read Consistency)



# 敘述層級讀取一致性

- ORACLE 系統通常實施敘述層級讀取一致性，以保證個別查詢所傳回的資料與該查詢開始時一致。因此一個查詢從不會看到查詢過程中其他異動交易所做的委付確認(Commit) 的任何更新資料。



# 異動交易層級讀取一致性

- **ORACLE** 允許選擇實施異動交易層級讀取一致性，它保證在同一個時間點，同一個異動交易內的所有查詢資料是一致的。因此異動交易層級讀取一致性可以允許重覆性的讀取，仍保證資料的一致性。



## 兩種保證異動交易層次讀取一致性的方法

---

- 唯讀異動交易(Read –only Transaction)
- 互斥表鎖定與列鎖定(Exclusive Table and Row Lock)