

Microsoft
.net

Windows程式設計 ADO.NET資料存取(五)

講師：林賢達
Peter.lin@imestech.com

Microsoft Professional Microsoft Solution Developer Microsoft Application Developer

Microsoft
.net

課程大綱

- 並行控制(Concurrency Control)
- 交易處理(Transaction)
- 分散式交易處理(Distributed Transaction)

Microsoft
.net

並行控制

- 並行控制是離線存取架構必須面對的問題
- 當多個使用者試圖修改相同的記錄時，可能會引發Lost Update、Dirty Read、NonRepeatable Read與Phantoms Read等衝突
- 為了避免上述的衝突，必須對正在處理的資料進行鎖定，限制其他使用者存取這些資料

並行控制

- 悲觀鎖定，當某個使用者存取資料時，立刻對這些資料進行鎖定，不管使用者執行何種操作，即使是單純讀取資料
 - 可根本解決並行問題
 - 影響性能和處理容量
- 樂觀鎖定，只有當更新資料時，才進行鎖定
 - 在更新前偵測資料庫是否發生變更；如果尚未更新，則執行更新，否則拋出並行例外
 - ADO.NET 預設使用樂觀鎖定

並行控制

- 解決方案
 - 檢查欲修改的資料是否被修改


```
Update Employee Set DeptNo='A2' Where EmpNo='6905'
And DeptNo='A1'
```
 - 加入[時間戳記]欄位記錄最近更新的日期時間，作為更新資料時的條件。


```
Update Employee Set DeptNo='A2' Where EmpNo='6905'
And TimeStamp= '2007/10/4 12:30:00'
```

交易處理

- 交易處理主要確保ACID原則，也就是參與交易的所有動作不是全部成功，就是全部失敗，中間沒有任何灰色地帶
- 所謂ACID是指不可分割性(Atomicity)、一致性(Consistency)、隔離性(Isolation)、持續性(Durability)等原則
- 銀行轉帳便是一個非常典型的交易處理

Microsoft
.net

SqlTransaction類別

- ADO.NET的交易處理只能使用在單一資料庫連接，也就是無法跨越資料庫
 - 並非所有的SQL語法均可用於交易處理
 - 一個交易處理只有Commit或者Rollback兩種結果
- 開發步驟
 - 使用Connection的BeginTransaction方法取出一個Transaction物件，並設定使用何種隔離級別，支援ReadUncommitted, ReadCommitted(預設), RepeatableRead和Serializable

Microsoft
.net

SqlTransaction類別

- 開發步驟(續)
 - 將執行Insert、Update和Delete Command的Transaction屬性設定成Transaction物件
 - 執行異動命令，如果沒有發生錯誤，則使用Transaction的Commit方法確認所有的異動；否則使用Rollback方法復原所有的異動

Microsoft
.net

SqlTransaction類別

- 隔離級別

Isolation Level	Dirty Read	NonRepeatable Read	Phantoms Read
ReadUncommitted	Y	Y	Y
ReadCommitted	N	Y	Y
RepeatableRead	N	N	Y
Serializable	N	N	N

SqlTransaction類別

■ 使用範例

```
Dim trans As SqlTransaction
trans = cnn.BeginTransaction()
Try

    cmd.Transaction=trans
    cmd.ExecuteNonQuery()
    trans.Commit()
Catch
    trans.Rollback()
End Try
```

分散式交易處理

- DTC服務允許用戶端應用程式將許多不同來源的資料包含在同一筆交易中。DTC 負責協調交易中編列的所有伺服器之間分散式交易的確認。
- 為了執行跨資料庫的交易處理，需要使用 System.EnterpriseServices 向 DTC 註冊連接
- 開發步驟
 - 在 Web 服務專案中加入 System.EnterpriseServices 元件，使用 TransactionAttribute 類別

分散式交易處理

- 開發步驟(續)
 - 在 Web Method 使用 TransactionOption 特性指定是否建立交易
TransactionOption:=TransactionOption.Required
如果已經存在，則共用交易，否則建立新的交易
 - 在連接字串中使用 Enlist 參數指定此連接是否加入交易；如果 Enlist=False，可以使用 Connection 的 EnlistDistributedTransaction 方法加入交易

分散式交易處理

■ 控制分散式交易處理

- 使用AutoComplete屬性表示如果此方法正常執行完畢，則自動確認交易，否則回復交易
- 呼叫ContextUtil類別的SetComplete靜態方法來確認交易，或者呼叫SetAbout靜態方法回復交易

下課鐘響，回家真好...


