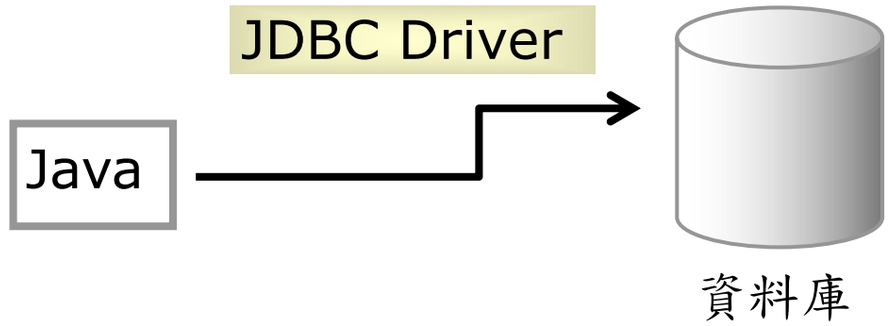
CH02 JDBC

- 2-1 JDBC與ODBC Driver
- 2-2 MySQL JDBC Driver下載與安裝
- 2-3 java.sql套件
- 2-4 註冊JDBC Driver
- 2-5 建立連結
- 2-6 建立Statement
- 2-7 執行SQL語法
- 2-8 交易與還原
- 2-9 利用執行錯誤啟動還原機制
- 2-10 Savepoint

黃彬華編撰

2-1 JDBC與ODBC Driver

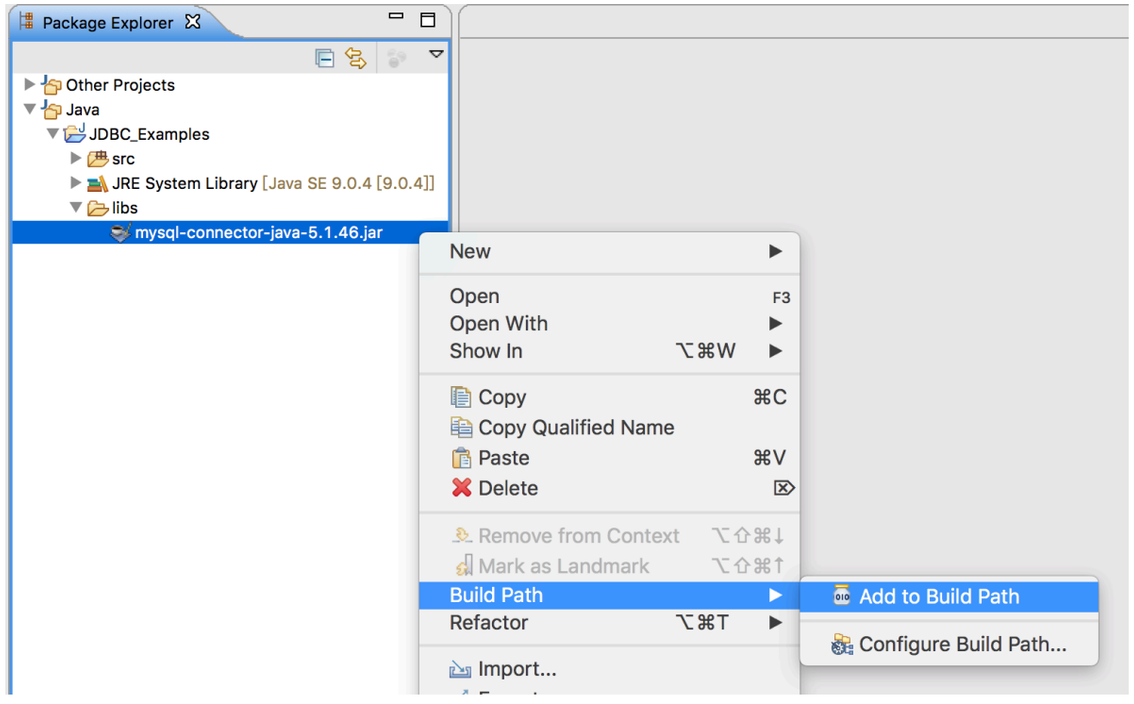
- JDBC (Java Database Connectivity) Driver
  - Java透過JDBC Driver操控資料庫
  - JDBC Driver由資料庫廠商提供
  - 通常以JAR檔案形式存在



- ODBC (Open Database Connectivity) Driver
  - 提供一種標準的API來存取資料庫管理系統
  - 資料庫廠商提供的ODBC Driver並非針對Java設計，所以Java API有提供連結ODBC Driver的套件
  - ODBC Driver並非為Java量身訂作，所以效能不如JDBC Driver

2-2 MySQL JDBC Driver下載與安裝

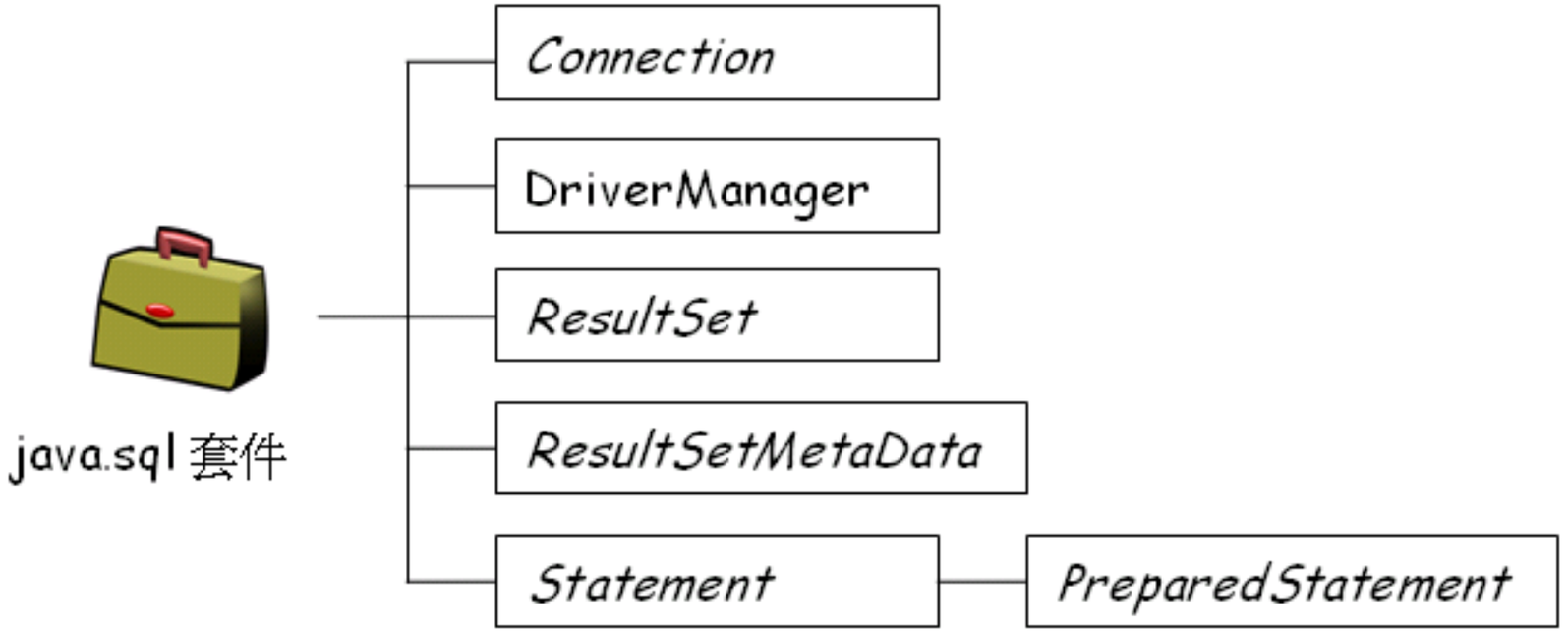
- 搜尋「download MySQL JDBC」
  - 下載「MySQL Connector/J」的ZIP壓縮檔
  - 解壓縮後找到JDBC的JAR檔案
- 安裝至Eclipse專案
  - 建立Eclipse專案，並建立一個名為libs的目錄
  - 將JDBC的JAR檔案複製到libs目錄
  - 對著JAR檔案右鍵 > Build Path > Add to Build Path



2-3 java.sql套件

- 要連結資料庫並執行SQL語法需要java.sql套件內的功能

java.sql套件架構圖



2-4 註冊JDBC Driver

- 必須先註冊JDBC driver後方能初始化driver物件
  - 一個app只需要以下列語法註冊一次
    - Class.forName("com.mysql.jdbc.Driver");
  - 執行完註冊會初始化driver物件並存放在記憶體內，而該driver物件就是Java API關於JDBC介面的實作物件
- 有可能產生ClassNotFoundException
  - 註冊時找不到指定名稱的類別

2-5 建立連結

- 建立連結物件
  - 需提供資料庫的URL、user帳號與密碼
    - Connection connection = DriverManager.getConnection(URL, USER, PASSWORD);
  - 可能產生SQLException
    - 連結不成功時發生，例如：帳號、密碼錯誤
- 當JDBC程式即將結束前要關閉連結以釋放資源
  - 將關閉連結的程式碼放在finally區塊內
  - 或是使用Java 7的try-with-resources語法自動關閉
- 範例：ConnectTestDemo

2-6 建立Statement

- Statement
  - 建立完connection可以取得statement
  - 負責將SQL語法送到資料庫端執行並將結果回傳至Java端
- PreparedStatement
  - Statement子介面，建議使用
  - 除了具備Statement功能外還提供SQL語法預先編譯的功能，這樣可以更有效地重複利用SQL語法
  - 提供SQL參數設定功能
    - INSERT INTO PUBLISHER (PUBLISHER_ID, PUBLISHER_NAME, CONTACT, PHONE) VALUES (**?, ?, ?, ?**)
    - preparedStatement.setXXX(index, value)設定上述「?」對應的值
      - 例如：preparedStatement.setString(1, "P00Z");
- 不使用時要關閉statement以釋放資源

2-7 執行SQL語法

- 異動語法
  - 呼叫statement.executeUpdate()，會回傳異動成功的資料筆數
  - 適用INSERT、UPDATE、DELETE等會改變資料庫內容的語法
  - 範例：InsertDemo, UpdateDemo, DeleteDemo
- 查詢語法
  - 呼叫statement.executeQuery()方法，會回傳查詢結果ResultSet物件 (不會為null)
  - 適用SELECT等查詢資料庫的語法
  - resultSet.next()將指標移動至下一個位置
  - resultSet.getXXX(index)搭配欄位索引 (1-based)取值
  - 範例：ConnectTestDemo
- SQL與Java型別對照表
  - https://www.tutorialspoint.com/jdbc/jdbc-data-types.htm

2-8 交易與還原

- 一個完整交易 (transaction) 可能是由多個異動指令組成
  - 例如：新增訂單包含新增資料至訂單主檔與訂單明細，所有資料都新增成功才算交易成功，只要有一筆資料新增失敗就需要還原 (roll back)
- Auto Commit
  - JDBC connection預設為auto commit模式，所以每一個SQL異動指令一經完成即確定為一個交易，確定後無法還原
- 關閉Auto Commit
  - 若希望自行界定一個完整交易，必須關閉auto commit模式
  - 呼叫connection.setAutoCommit(false)即可關閉
  - 要還原可呼叫connection.rollback()
  - 確定交易完成呼叫connection.commit()，但就不可還原了
- 範例：RollbackDemo

2-9 利用執行錯誤啟動還原機制

- JDBC技術存取資料庫過程中常會發生SQLException執行錯誤
- 在catch區塊處理錯誤時，將資料異動還原，以避免錯誤結果保留在資料庫內
- 範例：RollbackErrorDemo

2-10 Savepoint

- 設定savepoint (儲存點) ，可以roll back到當初設定的savepoint
- 範例：RollbackSavepointDemo

黃彬華編撰