第9章　程式

9-1 元件清單及其規格描述

▼表9-1-2 元件清單及其規格描述表(後端)

|  |  |  |
| --- | --- | --- |
| Python檔案(後端)－圖形資料庫(Neo4j) | | |
| 編號 | 檔案名稱 | 功能 |
| 2-3-1 | neo4j\_transfer\_and\_creation.py | 將mysql資料導入neo4j，並建立節點(房屋物件、價格、房屋類型、房間類型、坪數)以及建立關聯 |
| 2-3-2 | neo4j\_nearby\_features.py | 將整理好的.json(附近特色:學校、商店、醫院……)導入先前建好的neo4j資料庫，建立節點及關聯 |
| 2-3-3 | neo4j\_transportation.py | 將整理好的.json(附近交通:公車、捷運)導入先前建好的neo4j資料庫，建立節點及關聯 |
| 2-3-4 |  |  |
| 2-3-5 |  |  |

9-2 其他附屬之各種元件

▼表9-2-1 部分程式碼－neo4j\_transfer\_and\_creation.py

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 對應9-1編號 | 程式名稱 | neo4j\_transfer\_and\_creation.py |
| 目的 | 將mysql資料導入neo4j，並建立節點(房屋物件、價格、房屋類型、房間類型、坪數)以及建立關聯 | | |
| 部分程式碼 | | | |
| # 從 mysql.connector 和 py2neo 模組導入所需的類別和方法  # 連接到 MySQL 資料庫  嘗試建立 MySQL 連接:  使用指定的主機、使用者、密碼和資料庫名稱  如果成功, 打印 "MySQL 連接已建立"  如果出現錯誤:  打印錯誤訊息並退出程式  # 定義重試連接 Neo4j 的函數  定義一個函數 connect\_to\_neo4j:  設定重試次數和等待時間  在每次嘗試連接 Neo4j 時:  如果連接成功:  打印 "Neo4j 連接已建立" 並返回連接物件  如果連接失敗:  打印當前的失敗嘗試次數和錯誤訊息  等待指定的時間後再次嘗試  如果所有嘗試都失敗:  打印錯誤訊息並退出程式  # 連接到 Neo4j 資料庫  呼叫 connect\_to\_neo4j 函數  # 查詢 MySQL 資料  嘗試從 MySQL 資料庫查詢數據:  使用 cursor 來執行查詢語句並獲取所有結果  如果成功, 打印 "MySQL 資料已成功檢索"  如果出現錯誤:  打印錯誤訊息, 關閉 MySQL 連接並退出程式  # 定義價格和大小的範圍  設置價格範圍的列表:  每個範圍包含一個最小值和最大值  設置大小範圍的列表:  每個範圍包含一個最小值和最大值  # 定義一個函數來解析大小  定義 parse\_size 函數:  使用正則表達式匹配並提取大小的數字部分  如果匹配成功:  返回提取的數字  如果匹配失敗:  返回 None  # 處理資料並寫入 Neo4j  嘗試處理每一行 MySQL 查詢結果:  解析數據字段  如果大小解析失敗:  打印錯誤訊息並跳過此物件  創建房屋節點  合併物件類型節點（避免重複創建）  合併物件類型（pattern）節點（避免重複創建）  根據價格建立價格範圍節點:  檢查房屋價格屬於哪個範圍  合併該價格範圍的節點（避免重複創建）  創建房屋與價格範圍節點的關聯  根據大小建立大小範圍節點:  檢查房屋大小屬於哪個範圍  合併該大小範圍的節點（避免重複創建）  創建房屋與大小範圍節點的關聯  創建與物件類型的關聯  創建與物件類型（pattern）的關聯  打印節點和關聯的創建訊息  如果出現錯誤:  打印錯誤訊息  # 關閉 MySQL 連接  關閉 cursor 和 MySQL 連接  打印 "MySQL 連接已關閉" | | | |

▼表9-2-2 部分程式碼－neo4j\_nearby\_features.py

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 對應9-1編號 | 程式名稱 | neo4j\_nearby\_features.py |
| 目的 | 將整理好的.json(附近特色:學校、商店、醫院……)導入先前建好的neo4j資料庫，建立節點及關聯 | | |
| 部分程式碼 | | | |
| # 從 json 和 py2neo 模組導入所需的類別和方法  # 定義一個連接到 Neo4j 資料庫的函數，帶有重試機制  定義函數 connect\_to\_neo4j:  設定重試次數和等待時間  在每次嘗試連接 Neo4j 時:  如果連接成功:  打印 "Neo4j 連接已建立" 並返回連接物件  如果連接失敗:  打印當前的失敗嘗試次數和錯誤訊息  等待指定的時間後再次嘗試  如果所有嘗試都失敗:  打印錯誤訊息並退出程式  # 連接到 Neo4j 資料庫  呼叫 connect\_to\_neo4j 函數  # 設定 JSON 文件的路徑  設置 json\_file\_path 為指定的文件路徑  # 檢查文件是否存在  如果文件不存在:  打印 "找不到文件" 的訊息並退出程式  # 讀取 JSON 文件  嘗試開啟並讀取 JSON 文件:  使用 utf-8 編碼讀取文件內容並解析為 JSON  如果讀取或解析失敗:  打印錯誤訊息並退出程式  # 處理資料並寫入 Neo4j  嘗試處理 JSON 文件中的每一個條目:  取得條目中的 hid 和 store 資訊  如果 store 不是列表類型:  打印錯誤訊息並跳過此物件  查找對應的 Property 節點  如果找不到對應的 Property 節點:  打印錯誤訊息並跳過此物件  對於每個 store 資訊:  創建或取得對應的 Store 節點（避免重複創建）  創建 Property 節點與 Store 節點之間的關聯  打印已創建關聯的訊息  如果處理過程中出現錯誤:  打印錯誤訊息  # 最後打印 "所有 store 資料已成功整合進 Neo4j" 的訊息 | | | |

▼表9-2-3 部分程式碼－neo4j\_transportation.py

|  |  |  |  |
| --- | --- | --- | --- |
| 編號 | 對應9-1編號 | 程式名稱 | neo4j\_transportation.py |
| 目的 | 將整理好的.json(附近交通:公車、捷運)導入先前建好的neo4j資料庫，建立節點及關聯 | | |
| 部分程式碼 | | | |
| # 從 json 和 py2neo 模組導入所需的類別和方法  # 定義一個連接到 Neo4j 資料庫的函數，帶有重試機制  定義函數 connect\_to\_neo4j:  設定重試次數和等待時間  在每次嘗試連接 Neo4j 時:  如果連接成功:  打印 "Neo4j 連接已建立" 並返回連接物件  如果連接失敗:  打印當前的失敗嘗試次數和錯誤訊息  等待指定的時間後再次嘗試  如果所有嘗試都失敗:  打印錯誤訊息並退出程式  # 連接到 Neo4j 資料庫  呼叫 connect\_to\_neo4j 函數  # 設定 JSON 文件的路徑  設置 json\_file\_path 為指定的文件路徑  # 檢查文件是否存在  如果文件不存在:  打印 "找不到文件" 的訊息並退出程式  # 讀取 JSON 文件  打開並讀取 JSON 文件:  使用 utf-8 編碼讀取文件內容並解析為 JSON  # 處理資料並寫入 Neo4j  嘗試處理 JSON 文件中的每一個條目:  取得條目中的 hid、subway 和 bus 資訊  # 查找對應的 Property 節點  查找對應 hid 的 Property 節點  如果找不到對應的 Property 節點:  打印錯誤訊息並跳過此物件  # 創建或取得每個 Subway 節點並創建關聯  對於每個 subway 資訊:  創建或取得對應的 Subway 節點（避免重複創建）  創建 Property 節點與 Subway 節點之間的關聯  打印已創建關聯的訊息  # 創建或取得每個 Bus 節點並創建關聯  對於每個 bus 資訊:  創建或取得對應的 Bus 節點（避免重複創建）  創建 Property 節點與 Bus 節點之間的關聯  打印已創建關聯的訊息  如果處理過程中出現錯誤:  打印錯誤訊息  # 最後打印 "所有交通資料已成功整合進 Neo4j" 的訊息 | | | |