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 | | | |
| 1. 初始化與設置    * 載入所需的模組和庫：      + mysql.connector.connect 和 mysql.connector.Error：用於連接 MySQL 數據庫和處理錯誤。      + py2neo.Graph, py2neo.Node, py2neo.Relationship：用於連接 Neo4j 數據庫，創建節點和關係。      + time：用於在重試連接 Neo4j 時進行延遲。      + re：用於正則表達式處理文本。 2. 連接到 MySQL 數據庫    * 嘗試連接到 MySQL 數據庫，並處理可能的連接錯誤。 3. 定義函數 connect\_to\_neo4j    * 輸入參數: retries（重試次數）和 wait（等待時間）。    * 在指定的重試次數內嘗試連接到 Neo4j，如果失敗，則等待一段時間再重試。    * 成功後返回 Neo4j 的 graph 對象；如果所有嘗試均失敗，則退出程式。 4. 連接到 Neo4j 數據庫    * 使用 connect\_to\_neo4j 函數嘗試連接到 Neo4j。 5. 查詢 MySQL 數據    * 使用 SQL 語句從 MySQL 數據庫中提取所需數據。    * 處理可能的查詢錯誤。 6. 定義價格和大小範圍    * 定義價格範圍 price\_ranges 和大小範圍 size\_ranges。 7. 定義函數 parse\_size    * 輸入參數: size\_str（大小字符串）。    * 使用正則表達式從字符串中提取數字並返回；如果提取失敗，則返回 None。 8. 處理數據並寫入 Neo4j    * 對於每個從 MySQL 提取的數據行 row：      + 提取並解析屬性信息（如 hid, pattern, size\_str, layer, type, price, address, subway, bus）。      + 如果大小無法解析，則跳過該屬性。      + 創建房屋節點：使用 Node 創建包含屬性信息的房屋節點，並將其寫入 Neo4j。      + 創建物件類型節點：創建或合併物件類型節點到 Neo4j。      + 創建物件類型（pattern）節點：創建或合併物件類型（pattern）節點到 Neo4j。      + 創建價格範圍節點：根據價格，創建或合併相應的價格範圍節點到 Neo4j，並創建與房屋節點的關係。      + 創建大小範圍節點：根據大小，創建或合併相應的大小範圍節點到 Neo4j，並創建與房屋節點的關係。      + 創建類型和 pattern 關係：為房屋節點創建與類型和 pattern 節點的關係。    * 處理可能的寫入錯誤。 9. 關閉 MySQL 連接    * 關閉 MySQL 游標和連接。 | | | |

▼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 | | | |
| 1. 初始化與設置  * 載入所需的模組和庫：   + json：讀取和解析 JSON 文件。   + py2neo.Graph, py2neo.Node, py2neo.Relationship：用於連接 Neo4j 數據庫，創建節點和關係。   + time：用於在重試連接 Neo4j 時進行延遲。   + os：用於檢查文件是否存在。  1. 定義函數 connect\_to\_neo4j  * 輸入參數: retries（重試次數）和 wait（等待時間）。 * 在指定的重試次數內嘗試連接到 Neo4j，如果失敗，則等待一段時間再重試。 * 成功後返回 Neo4j 的 graph 對象；如果所有嘗試均失敗，則退出程式。  1. 連接到 Neo4j 數據庫  * 使用 connect\_to\_neo4j 函數嘗試連接到 Neo4j。  1. 設定 JSON 文件路徑  * 指定 JSON 文件的路徑 json\_file\_path。  1. 檢查文件是否存在  * 使用 os.path.exists 函數檢查指定的 JSON 文件是否存在。 * 如果文件不存在，則輸出錯誤消息並退出程式。  1. 讀取 JSON 文件  * 使用 open 函數和 json.load 讀取並解析 JSON 文件的內容。 * 處理可能的 JSON 解碼錯誤。  1. 處理數據並寫入 Neo4j  * 對於 JSON 數據中的每個項目 entry：   + 提取房屋 ID (hid) 和商店列表 (stores)。   + 檢查 stores 是否為列表類型；如果不是，則跳過該項目。   + 查找對應的 Property 節點：根據 hid 在 Neo4j 中查找對應的 Property 節點。   + 如果找不到對應的 Property 節點，則跳過該項目。   + 創建或獲取 Store 節點並創建關係：     - 對於每個商店名稱 store，創建或合併對應的 Store 節點。     - 創建 Property與 Store 節點之間的 NEAR\_STORE 關係。 * 處理可能的數據處理錯誤。  1. 成功完成數據整合  * 輸出數據成功整合的消息。 | | | |

▼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 | | | |
| 1. 初始化與設置  * 載入所需的模組和庫：   + json：讀取和解析 JSON 文件。   + py2neo.Graph, py2neo.Node, py2neo.Relationship：用於連接 Neo4j 數據庫，創建節點和關係。   + time：用於在重試連接 Neo4j 時進行延遲。   + os：用於檢查文件是否存在。  1. 定義函數 connect\_to\_neo4j  * 輸入參數: retries（重試次數）和 wait（等待時間）。 * 在指定的重試次數內嘗試連接到 Neo4j，如果失敗，則等待一段時間再重試。 * 成功後返回 Neo4j 的 graph 對象；如果所有嘗試均失敗，則退出程式。  1. 連接到 Neo4j 數據庫  * 使用 connect\_to\_neo4j 函數嘗試連接到 Neo4j。  1. 設定 JSON 文件路徑  * 指定 JSON 文件的路徑 json\_file\_path。  1. 檢查文件是否存在  * 使用 os.path.exists 函數檢查指定的 JSON 文件是否存在。 * 如果文件不存在，則輸出錯誤消息並退出程式。  1. 讀取 JSON 文件  * 使用 open 函數和 json.load 讀取並解析 JSON 文件的內容。  1. 處理數據並寫入 Neo4j  * 對於 JSON 數據中的每個項目 entry：   + 提取房屋 ID (hid)、地鐵站列表 (subways) 和公車站列表 (buses)。   + 查找對應的 Property 節點：根據 hid 在 Neo4j 中查找對應的 Property 節點。   + 如果找不到對應的 Property 節點，則跳過該項目。   + 創建或獲取 Subway 節點並創建關係：     - 對於每個地鐵站名稱 subway\_station，創建或合併對應的 Subway 節點到 Neo4j。     - 創建 Property 節點與 Subway 節點之間的 NEAR\_SUBWAY 關係。   + 創建或獲取 Bus 節點並創建關係：     - 對於每個公車站名稱 bus\_stop，創建或合併對應的 Bus 節點到 Neo4j。     - 創建 Property 節點與 Bus 節點之間的 NEAR\_BUS 關係。 * 處理可能的數據處理錯誤。  1. 成功完成數據整合  * 輸出數據成功整合的消息。 | | | |