Introduction to Database Systems

Term Project: Report

0716214 江岳勳

0716222 黃偉傑

1. Data
   1. Introduction

　　我們的資料以公司資訊、職缺資訊、地域、職缺類別、使用者資訊和最愛組成。其中公司資訊、職缺資訊、地域(不確定)和職缺類別的來源，是我們由[1111人力銀行](https://www.1111.com.tw/)網站爬蟲下來所得。而使用者資訊和最愛，是在application運行時，由使用者執行相對應的操作之後，執行insert指令所得。

* 1. Description of the tables

　　首先以一個表格來呈現資料之於table的構成。

|  |  |  |  |
| --- | --- | --- | --- |
| Table | Columns | Primary Key | Description |
| jobinfo | **job\_id**, job\_name, degree, low\_salary, high\_salary, exp\_year, job\_type, worktime, is\_night, needed\_num | job\_id | 職缺資訊 |
| company | **com\_id**, com\_name, capital, emp\_number, addr | com\_id | 公司資訊 |
| position | **pos\_id**, pos\_field, pos\_name | pos\_id | 職缺的職務與  領域類別 |
| localarea | **area\_id**, area\_cc\_name, area\_td\_name, area\_cctd\_name | area\_id | 臺灣各行政區 |
| job | **job\_id**, com\_id, pos\_id, area\_id | job\_id | 職缺與公司、職缺類別、地域之間的關係 |
| user | **username**, passwd | username | 使用者資訊 |
| favorite | username, job\_id | N/A | 使用者加入最愛 |

　　以下每個表格分別代表著一個table，包含了column在MySQL內儲存的資訊與功能描述。

* **jobinfo**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Type | Null | Key | Default | Description |
| job\_id | int | NO | PRI | NULL | 職缺id |
| job\_name | varchar(100) | NO |  | NULL | 職缺名稱 |
| degree | tinyint | NO |  | NULL | 最低學歷要求，見下方說明 |
| low\_salary | int | YES |  | NULL | 薪資下限，面議時為4萬 |
| high\_salary | int | YES |  | NULL | 薪資上限，面議時為NULL |
| exp\_year | tinyint | NO |  | NULL | 最低年資要求 |
| job\_type | tinyint(1) | NO |  | NULL | 工作型態，見下方說明 |
| worktime | float | NO |  | NULL | 工作時間（單位為小時） |
| is\_night | tinyint | NO |  | NULL | 日／夜班，見下方說明 |
| needed\_num | tinyint | YES |  | NULL | 職缺需求人數 |

\*說明：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Field | Explanation | | | | | |
| degree | 0（不拘） | 1（國中以下） | | 2（高中／職） | | 3（專科） |
| 4（大學） | | 5（碩士） | | 6（博士） | |
| job\_type | 0（正職） | | | 1（兼職） | | |
| is\_night | 0（大夜班） | 1（日班） | | 2（夜班） | | 3（皆有） |

* **company**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Type | Null | Key | Default | Description |
| com\_id | int | NO | PRI | NULL | 公司id |
| com\_name | varchar(100) | NO |  | NULL | 公司名稱 |
| capital | varchar(10) | YES |  | NULL | 資本額 |
| emp\_number | int | YES |  | NULL | 員工人數 |
| addr | varchar(100) | YES |  | NULL | 公司地址 |

* **position**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Type | Null | Key | Default | Description |
| pos\_id | int | NO | PRI | NULL | 職務id |
| pos\_field | varchar(8) | NO |  | NULL | 職務領域 |
| pos\_name | varchar(20) | NO |  | NULL | 職務名稱 |

\*說明：此處的資訊是求職網站為職缺所立的類別，例如「業務／業務助理」或「光電半導體／半導體工程師」等。

* **localarea**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Type | Null | Key | Default | Description |
| area\_id | smallint | NO | PRI | NULL | 地域id |
| area\_cc\_name | varchar(10) | NO |  | NULL | 縣市名稱 |
| area\_td\_name | varchar(10) | NO |  | NULL | 鄉鎮市區名稱 |
| area\_cctd\_name | varchar(10) | NO |  | NULL | 前兩者之連接 |

* **job**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Type | Null | Key | Default | Description |
| job\_id | int | NO | PRI | NULL | 職缺id |
| com\_id | int | NO |  | NULL | 公司id |
| pos\_id | int | NO |  | NULL | 職務id |
| area\_id | smallint | NO |  | NULL | 地域id |

* **user**

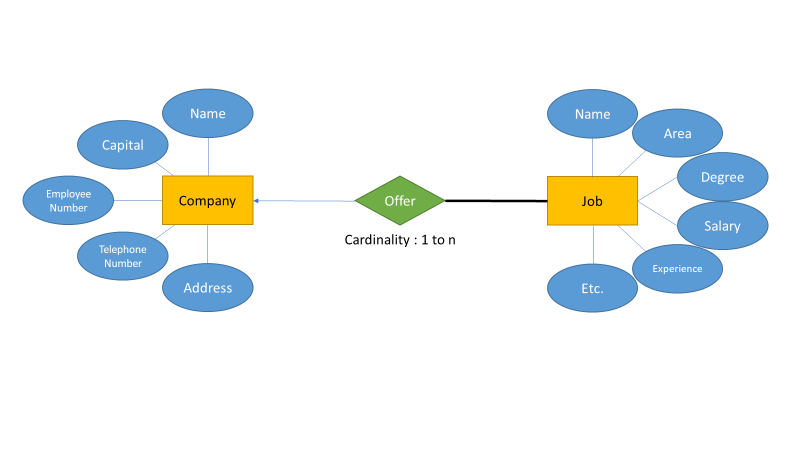
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Type | Null | Key | Default | Description |
| username | varchar(15) | NO | PRI | NULL | 使用者名稱 |
| passwd | char(64) | NO |  | NULL | 使用者密碼 |

* **favorite**

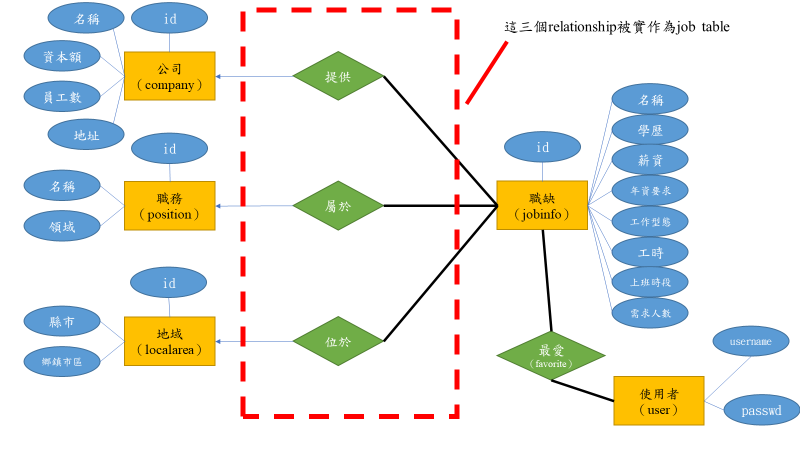
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Field | Type | Null | Key | Default | Description |
| username | varchar(15) | NO |  | NULL | 使用者名稱 |
| job\_id | int | NO |  | NULL | 職缺id |

\*說明：此功能為使用者將職缺加入最愛，我們也會在功能說明的部分描述這項功能。

* 1. ER Models



　　上圖為原本的ER Model，下圖為最後的ER Model。從下圖可以看到，職缺（jobinfo）和公司（company）、職務（position）、地域（localarea）三者之間的關係是多對一，而職缺（jobinfo）和使用者（user）的「最愛（favorite）」關係是多對多。



1. Database
   1. Introduction / Maintenance

我們使用的資料庫是MySQL。

當我們要更新資料庫時，例如加入新的職缺資訊、公司資訊（陸續爬蟲下來），我們會利用update.sql（/src/sql/update.sql）來更新，其中包含了刪除資料庫內舊的資訊和載入新的資訊，並會在載入新資訊後檢查favorite table內的資料是否過時，若存在過時資料則會將其刪除。

除此之外，在運行階段時也會有動態的資料被加進資料庫中/從資料庫移除。這主要分為兩類：

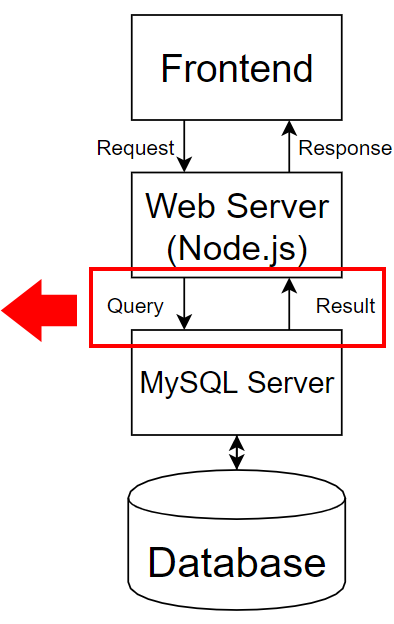
1. 有新用戶註冊

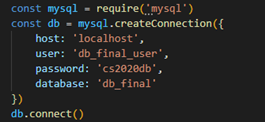
當有新用戶要註冊時，若用戶資料皆符合要求，則會在user table上新增一筆(username, passwd)的資料，其中passwd為經過SHA-256加密過的64位的十六進位字串。

1. 用戶新增/移除了一筆最愛職缺

當用戶新增一筆最愛職缺時，會向favorite table增加一筆(username, job\_id)的資料；反之當用戶移除某筆最愛職缺時，符合(username, job\_id)的資料列則會從favorite table被移除。

* 1. Connection / Interaction

我們連接資料庫的方式是透過Node.js架設的後台在啟動時會require npm 中的mysql模組，建立一個資料庫的connection實例，之後的query便都經由此實例來與資料庫互動。



在前端呼叫後端存取資料庫時，前端會經由HTTP POST和Web Socket發送一條serialized成字串的JSON，包含著必要的資料(通常是job\_id等資訊)，後端收到前端的資料後會視情況(像是存取用戶最愛列表時)加入用戶目前session的資料，將資料渲染成相對應的SQL指令，並向MySQL的connection發出query請求，在結果回傳後會再將結果渲染成相應的HTML或JSON，並回傳給前端顯示。

至於Database exception handling的部分，當資料庫的query出現錯誤時，後端會停止這個頁面的渲染，在HTTP response中寫入error.html的內容並回傳給前端。

1. Application
   1. Interface
   2. Functions
2. Others