

國立虎尾科技大學資訊工程系  
資料庫專題製作報告

# ETF 投資組合管理系統 (第八組)

參與成員：田晉嘉 周偉宸  
陳億穎 陳峻宇

中華民國 114 年 6 月 15 日

## 目錄

一、	介紹 .....	3
二、	應用情境與使用案例 .....	4
三、	系統需求說明 .....	6
四、	ER Diagram 及詳細說明 .....	7
五、	資料庫 Schema 及完整性限制 .....	12
六、	使用者權限設定 .....	25
七、	使用者 View(1~8) .....	28
八、	管理員 View(9~10) .....	40
九、	資料筆數 .....	45
十、	DEMO .....	45
十一、	備份&還原資料庫 .....	48
十二、	參考資料/處理方式 .....	49
十三、	心得 .....	51

## 一、 介紹

### - 專案簡介

該系統協助投資者整合與管理其 ETF 投資資訊，提供利用標籤查詢 ETF、即時交易與持倉更新，並產生投資組合明細與統計。

## 功能概述

- **用戶管理**：註冊、登入及個人資料設定
- **ETF 列表與篩選**：儲存各 ETF 的基本資料（如代碼、名稱、管理公司、規模等）下拉選單顯示所有 ETF 並支援查詢
- **歷史價格與漲跌幅**：可以查看台灣所有 ETF 創立至今(2025/6 月)K 線圖走勢 查詢指定期間收盤價並計算漲跌幅百分比
- **交易與持倉管理**：記錄用戶的每筆買賣交易 即時計算並更新持倉狀態、成本與盈虧
- **績效與風險評估**：產生投資組合資料（收益率、ETF 配置等圖表）

## 成員

組員一：資工系 - 41143210 - 田晉嘉 [JustinTien10](#)

負責項目：[ETF 歷史走勢爬取、編排文件、協助 View 製作、Github 編輯]

Email：41143210@nfu.edu.tw

組員二：資工系 - 41143219 - 周偉宸 [WeiChen-Zhou](#)

負責項目：[權限設計&實作、建立 View、Github 編輯]

Email：41143219@nfu.edu.tw

組員三：資工系 - 41143238 - 陳峻宇 [twl-Benchen](#)

負責項目：[值域規劃、MariaDB 建立&實作、建立 View、ETF 標籤&成分股的爬蟲與整理、網頁實作、Github 總編輯]

Email：41143238@nfu.edu.tw

組員四：資工系 - 41143239 - 陳億穎 [yiyingg1226](#)

負責項目：[資料庫概念層建立、ER digram 製作、Github 編輯、成分股 view 設計&實作]

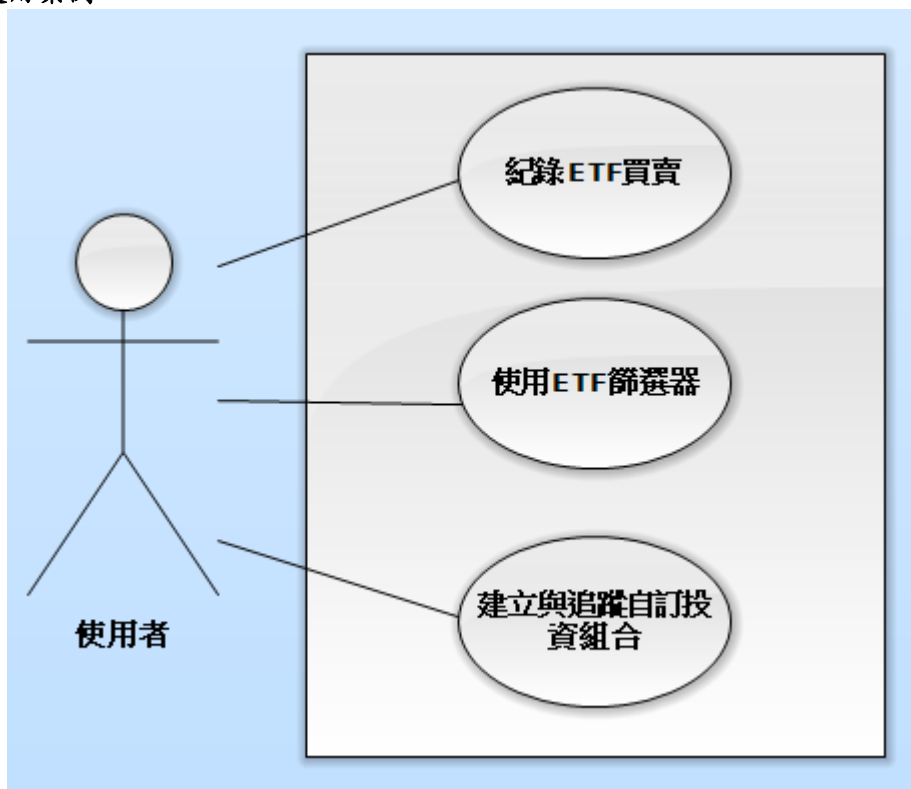
Email：41143239@nfu.edu.tw

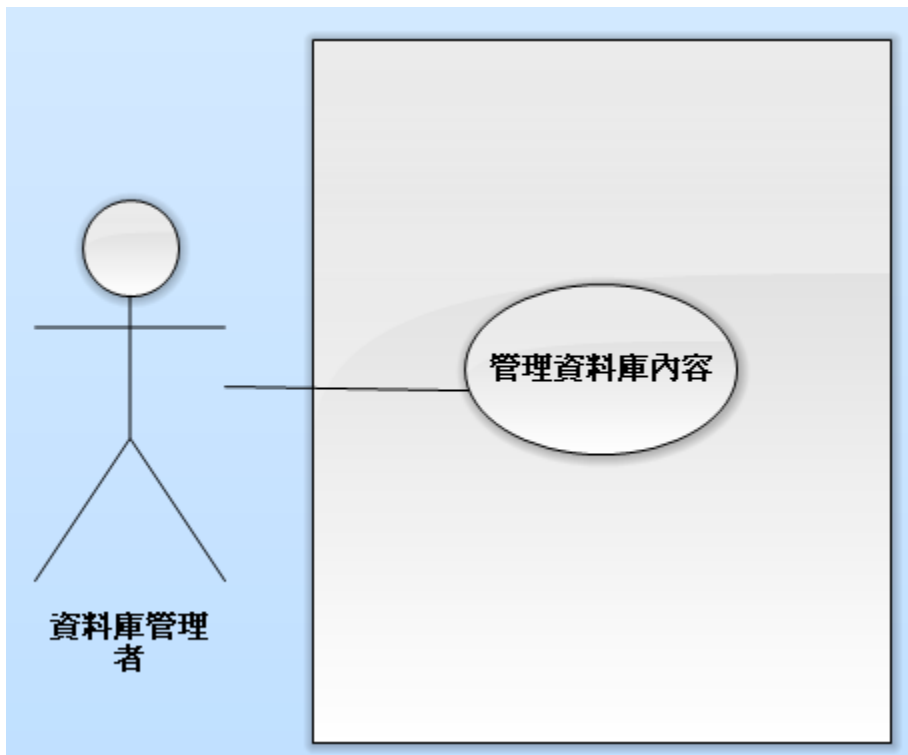
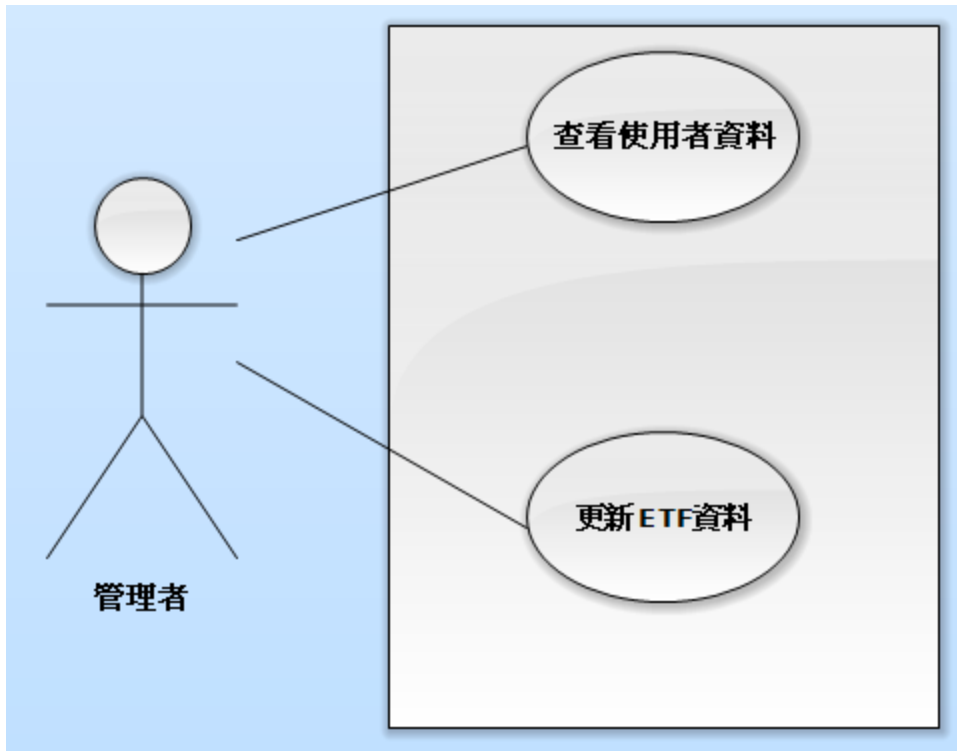
## 二、 應用情境與使用案例

### 應用情境

1. 小美登入後，系統立刻顯示她所有持有的 ETF 明細（名稱、股數、平均成本及總成本），並自動彙總出不同檔數、總持股與投資金額，同時在下方列出近期交易紀錄，讓她方便掌握自身的資產配置與操作狀況。
2. 小張想找到特定類型的 ETF，只要在分類介面先選「股票型」、再點「大型權值」，系統便列出所有符合條件的標的；若他需要更精準，也能直接輸入關鍵字（例如「元大」）進行搜尋，迅速鎖定符合策略的 ETF。
3. 小王若要分析一檔 ETF 的歷史表現，只要選擇期間（如 2024 至 2025 年），系統會計算該段漲跌幅，並提供每日 K 線與成交量資料，協助他評估這檔 ETF。

### 使用案例





- 使用者
  - 紀錄 ETF 買賣
  - 使用 ETF 篩選器
  - 建立與追蹤自訂投資組合
- 說明
  - 使用者可以買入或賣出 ETF，，對應「 Transaction(交易紀錄表)」
  - 依據標籤（如科技、能源、金融、高股息、ESG 等），快速找到符合需求的 ETF，對應「 Category\_Level1(父標籤)」與「 Category\_Level2(子標籤)」
  - 建立個人化投資組合，追蹤其報酬變化，對應「 Portfolio（持倉資料表）」
- 管理者
  - 查看使用者資料
  - 更新 ETF 資料
- 說明
  - 查看使用者帳號資料，對應「 Users(使用者基本資料)」
  - 更新 ETF 歷史數據、對應「ETF(基本資料表)」、「ETF\_HistoryPrice(ETF 歷史價格表)」
- 資料庫管理者
  - 管理資料庫內容
- 說明
  - 管理資料庫結構與資料，處理資料異常或系統升級

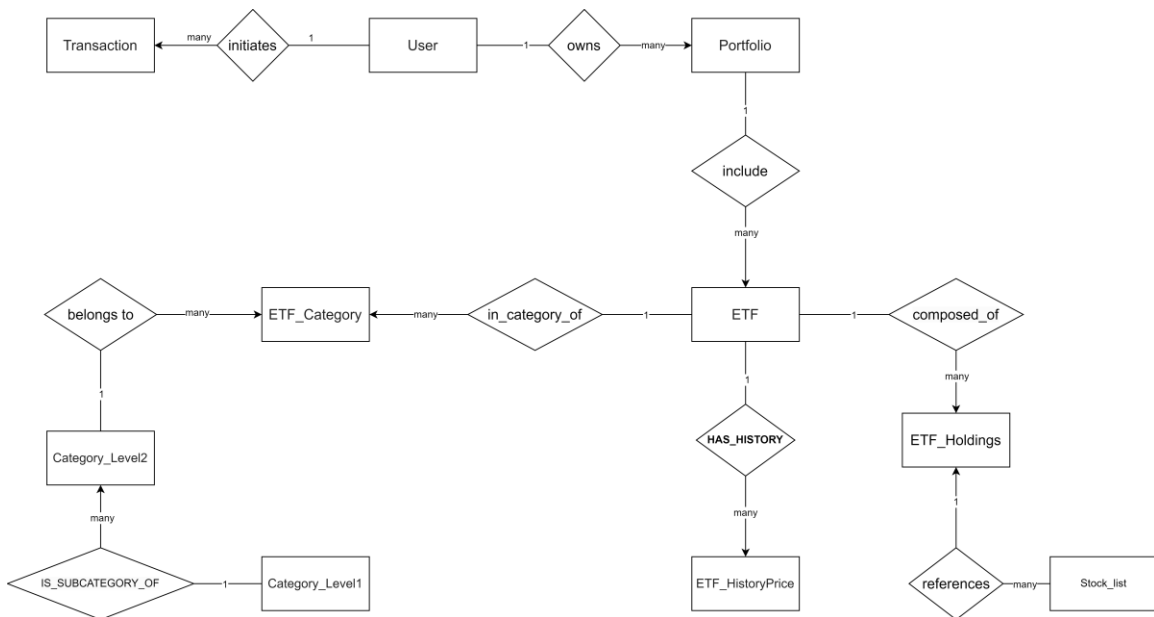
### 三、系統需求說明

- **交易與持倉管理**
  - 紀錄用戶每筆交易（含交易成本、價格、數量等）
  - 根據交易自動更新用戶當前持倉狀態與盈虧
  - 支援用戶調整 ETF 投資組合以因應市場變動
- **ETF 資料篩選**
  - 儲存與管理 ETF 基本資料（代碼、名稱、管理公司、規模等）
  - 記錄 ETF 的歷史價格與市場每日成交資訊

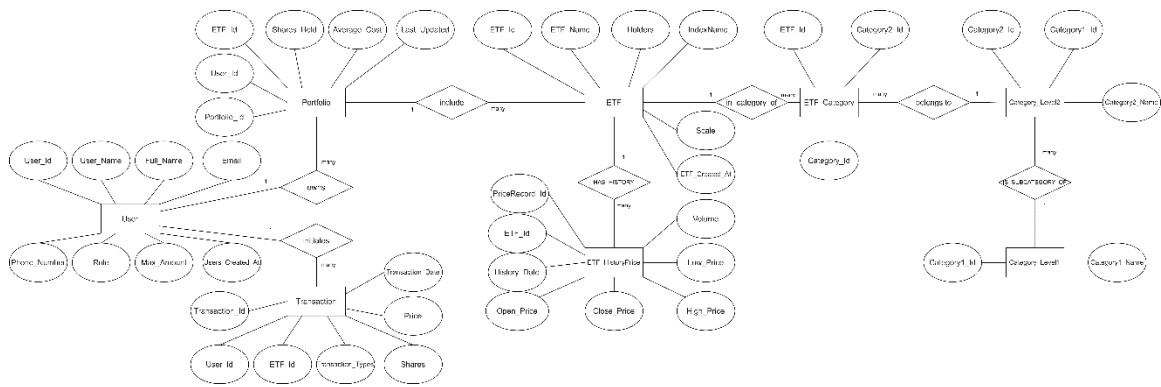
- 提供 ETF 分類、查詢與比較功能
- 支援自訂篩選條件以滿足用戶查詢需求
- 績效與風險評估
  - 生成 K 線走勢圖、以及交易量
- 用戶管理
  - 個人資料設定（姓名、信箱、單日買賣上限等）
  - 用戶可查詢歷史交易記錄與投資組合變化
- 安全性
  - 確保用戶個人資料與交易記錄的隱私性
  - 帳號與密碼分成兩個資料庫
  - 管理員定期備份資料庫

#### 四、ER Diagram 及詳細說明

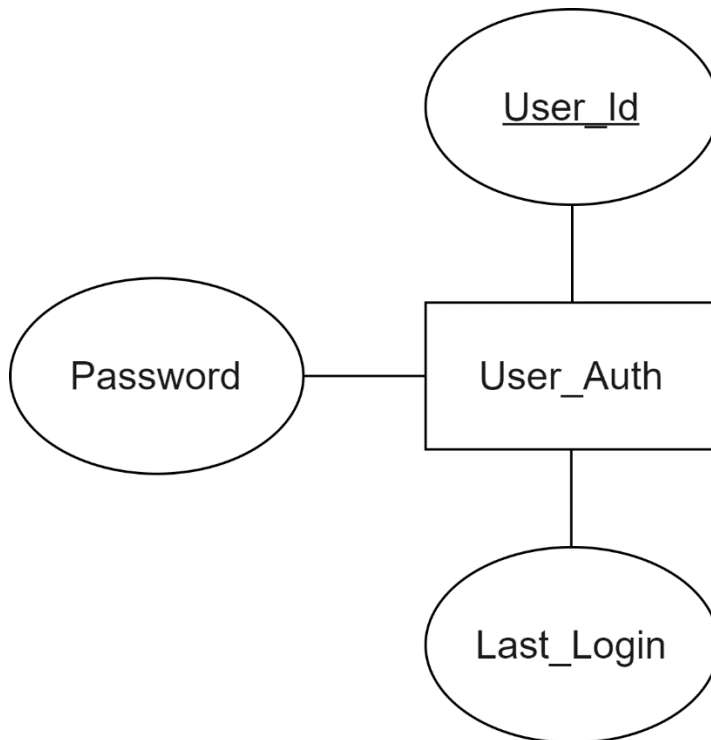
##### 簡略圖



##### 完整圖



密碼資料庫圖



### 1. 使用者密碼 (User\_Auth) 資料表屬性

- 使用者代號 (User\_Id)
- 使用者密碼 (Password)
- 最近登入 (Last\_Login)

### 2. 使用者基本資料 (Users) 資料表屬性

- 使用者代號 (User\_Id)
- 使用者名稱 (User\_Name)
- 全名 (Full\_Name)



- 電子郵件 (Email)
- 電話號碼 (Phone\_Number)
- 權限 (Role)
- 當日最大交易量 (Max\_Amount)
- 帳號創建日期 (Users\_Created\_At)

### **3. 交易紀錄表 (Transaction) 資料表屬性**

- 交易代號 (Transaction\_Id)
- 使用者代號 (User\_Id)
- ETF 代號 (ETF\_Id)
- 交易類型 (Transaction\_Type)
- 買賣股數 (Shares)
- 交易價格 (Price)
- 交易時間 (Transaction\_Date)

### **4. 持倉資料 (Portfolio) 資料表屬性**

- 持倉代號 (Portfolio\_Id)
- 使用者代號 (User\_Id)
- ETF 代號 (ETF\_Id)
- 持有股數 (Shares\_Held)
- 平均成本 (Average\_Cost)
- 最後更新日期 (Last\_Updated)

### **5. 成分股清單 (Stock\_list) 資料表屬性**

- 成分股代號 (Ticker\_Symbol)
- 成分股名稱 (Stock\_Name)
- 所屬產業類別 (Sector)

### **6. 成分股佔比 (ETF\_Holdings) 資料表屬性**

- ETF 代號 (ETF\_Id)
- 成分股代號 (Ticker\_Symbol)
- 該股票在 ETF 的比重 (Weight)

### **7. ETF 基本資料 (ETF) 資料表屬性**

- ETF 代號 (ETF\_Id)
- ETF 名稱 (ETF\_Name)
- 持有人數 (Holders)
- 追蹤指數 (IndexName)
- 規模 (Scale)
- 創立時間 (ETF\_Created\_At)

#### **8. ETF 歷史價格 (ETF\_HistoryPrice) 資料表屬性**

- 價格紀錄代號 (PriceRecord\_Id)
- ETF 代號 (ETF\_Id)
- 開盤價 (Open\_Price)
- 收盤價 (Close\_Price)
- 最高價 (High\_Price)
- 最低價 (Low\_Price)
- 交易量 (Volume)
- 日期 (History\_Date)

#### **9. 紀錄分類 (ETF\_Category) 資料表屬性**

- 紀錄分類代號 (Category\_Id)
- ETF 代號 (ETF\_Id)
- 子標籤代號 (Category2\_Id)

#### **10. 子標籤 (Category\_Level2) 資料表屬性**

- 子標籤代號 (Category2\_Id)
- 父標籤代號 (Category1\_Id)
- 子標籤名稱 (Category2\_Name)

#### **11. 父標籤 (Category\_Level1) 資料表屬性**

- 父標籤代號 (Category1\_Id)
- 父標籤名稱 (Category1\_Name)

#### **12. 關聯**

- 「使用者密碼 (Auth)」與「使用者基本資料表 (Users)」實體有一對一 (1..1) 的關係，表示：1..1 和 1..1。每筆使用者密碼只能對應一位使用者，而每位使用者也只能有一筆密碼資料。
- 「使用者基本資料表 (Users)」與「交易紀錄表 (Transaction)」實體有一對多 (1..N) 的關係，表示：1..1 和 0..\*。一位使用者可以有零到多筆交易紀錄，但每筆交易紀錄只能屬於一位使用者。
- 「使用者基本資料表 (Users)」與「持倉資料表 (Portfolio)」實體有一對多 (1..N) 的關係，表示：1..1 和 0..\*。一位使用者可以持有零到多筆持倉資料，但每筆持倉資料只能屬於一位使用者。
- 「持倉資料表 (Portfolio)」與「ETF」實體有一對多 (1..N) 的關係，表示：1..1 和 0..\*。每個 ETF 可以出現在多筆持倉資料中，但一筆持倉只能包含一個 ETF。
- 「ETF」與「ETF 歷史價格表 (ETF\_HistoryPrice)」實體有一對多 (1..N) 的關係，表示：1..1 和 0..\*。一個 ETF 可以有零到多筆歷史價格紀錄，但每筆歷史價格紀錄只能對應一個 ETF。
- 「ETF」與「紀錄分類表 (ETF\_Category)」實體有一對多 (1..N) 的關係，表示：1..1 和 0..\*。每筆分類可以對應多個 ETF，但每個 ETF 只能歸類於一個分類。
- 「紀錄分類表 (ETF\_Category)」與「子標籤表 (Category\_Level2)」實體有多對多 (M..N) 的關係，表示：0..\* 和 0..\*。一筆分類可以包含多個次分類，而一個次分類也可以屬於多個分類。
- 「父標籤表 (Category\_Level1)」與「子標籤表 (Category\_Level2)」實體有一對多 (1..N) 的關係，表示：1..1 和 0..\*。一個父標籤可以包含多個子標籤，但每個子標籤只能屬於一個父標籤。
- 「ETF 基本資料表 (ETF)」與「成分股佔比 (ETF\_Holdings)」實體有一對多 (1..N) 的關係，表示：1..1 和 0..\*。一個 etf 可以包含多個成分股。
- 「成分股清單 (Stock\_list)」與「成分股佔比 (ETF\_Holdings)」實體有一對多 (1..N) 的關係，表示：1..1 和 0..\*。一個成分股佔比可以包含多個成分股。

## 五、資料庫 Schema 及完整性限制

### ETF 基本資料表 (ETF)

欄位名稱	資料型態	是否可為空	欄位說明	值域	實際資料舉例
ETF_Id (PK)	VARCHAR(10)	N	ETF 代號	數字 + 英文字串	0050
ETF_Name	VARCHAR(100)	N	ETF 名稱	長度 1~100 的文字	元大台灣50
Holders	INT	N	持有人數	≥ 0 的整數	900000
IndexName	VARCHAR(50)	N	追蹤指數	長度 1~50 的文字	台灣50指數
Scale	INT	N	規模 (億)	≥ 0 的整數	5000
ETF_Created_At	TIMESTAMP	N	創立時間	時間格式：YYYY-MM-DD	2025-05-06

欄位名稱	值域限制說明	確認方式 (MySQL)
ETF_Id (PK)	必須為 1 到 10 個字元長度的字串，僅可包含阿拉伯數字 (0-9) 與英文字母 (A-Z、a-z)，且不可為空，用以唯一識別每檔 ETF。	<code>CHECK (ETF_Id REGEXP '^[0-9A-Za-z]{1,10}\$')</code>
ETF_Name	必須為 1 到 100 個字元長度的文字，可包含中英文、數字、空格及常見標點符號，且不可為空，用以顯示 ETF 的完整名稱。	<code>CHECK (CHAR_LENGTH(ETF_Name) BETWEEN 1 AND 100)</code>
Holders	必須為大於或等於 0 的整數，且不可為空，用以統計目前持有該 ETF 的投資人總數。	<code>CHECK (Holders &gt;= 0)</code>
IndexName	必須為 1 到 50 個字元長度的文字，可包含中英文、空格及常見標點符號，且不可為空，用以記錄該 ETF 所追蹤的基準指數名稱。	<code>CHECK (CHAR_LENGTH(IndexName) BETWEEN 1 AND 50)</code>
Scale	必須為大於或等於 0 的整數，且不可為空，以「億元」為單位表示該 ETF 的管理規模，實際儲存時以整數形式存放。	<code>CHECK (Scale &gt;= 0)</code>
ETF_Created_At	必須時間格式：YYYY-MM-DD	無需額外CHECK約束 (MySQL內建驗證)

```
-- 建立 ETF 資料表
CREATE TABLE ETF (
  ETF_Id VARCHAR(10) PRIMARY KEY,
  ETF_Name VARCHAR(100) NOT NULL,
  Holders INT NOT NULL,
  IndexName VARCHAR(50) NOT NULL,
  Scale INT NOT NULL,
  ETF_Created_At DATE NOT NULL,
  CHECK (ETF_Id REGEXP '^[0-9A-Za-z]{1,10}$'),
  CHECK (CHAR_LENGTH(ETF_Name) BETWEEN 1 AND 100),
  CHECK (Holders >= 0),
  CHECK (CHAR_LENGTH(IndexName) BETWEEN 1 AND 50),
  CHECK (Scale >= 0)
);

-- 範例：插入 0050 (台灣 50) 之 ETF 資料
INSERT INTO ETF (ETF_Id, ETF_Name, Holders, IndexName, Scale, ETF_Created_At)
VALUES ('0050', '元大台灣 50', 500000, '臺灣 50 指數', 250, '2003-06-25');
```

## 成分股清單 (Stock\_list)

欄位名稱	資料型態	是否可為空	欄位說明	值域	實際資料舉例
Ticker_Symbol(PK)	VARCHAR(10)	N	成分股代號	長度 1~10 的文字	2330、AAPL
Stock_Name	VARCHAR(100)	N	成分股名稱	長度 1~100 的文字	台積電
Sector	VARCHAR(50)	Y	所屬產業類別	長度 1~50 的文字	上市半導體業

```
-- 建立 Stock_list 資料表
CREATE TABLE Stock_list (
  Ticker_Symbol VARCHAR(10) NOT NULL PRIMARY KEY,
  Stock_Name VARCHAR(100) NOT NULL,
  Sector VARCHAR(50) DEFAULT NULL,
  CHECK (Ticker_Symbol REGEXP '^[A-Za-z0-9]{1,10}$'),
  CHECK (CHAR_LENGTH(Stock_Name) BETWEEN 1 AND 100),
  CHECK (Sector IS NULL OR CHAR_LENGTH(Sector) BETWEEN 1 AND 50)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;

-- 範例：插入成分股 (2330 台積電、半導體)
INSERT INTO Stock_list (Ticker_Symbol, Stock_Name, Sector)
VALUES ('2330', '台積電', '半導體');
```

## 成分股佔比 (ETF\_Holdings)

欄位名稱	資料型態	是否可為空	欄位說明	值域	實際資料舉例
ETF_Id	VARCHAR(10)	N	ETF 代號	長度 1~50 的文字	1
Ticker_Symbol	VARCHAR(10)	N	成分股代號	長度 1~100 的文字	2330
Weight	DECIMAL(5,2)	N	該股票在ETF的比重	0.00 到 100.00	12.75

欄位名稱	值域限制說明	確認方式 (MySQL)
ETF_Id	必須為 1 到 50 個字元長度的字串，僅可包含阿拉伯數字 (0-9) 與英文字母 (A-Z、a-z)，且不可為空，用以唯一識別每個股票。	<code>CHECK (ETF_Id REGEXP '^[0-9A-Za-z]{1,10}\$')</code>
Ticker_Symbol	必須為 1 到 100 個字元長度的文字，可包含中英文、數字、空格及常見標點符號，且不可為空，用以顯示股票的完整名稱。	<code>CHECK (Ticker_Symbol REGEXP '^[A-Z0-9]{1,10}\$')</code>
Weight	必須為 1 到 50 個字元長度的文字，可包含中英文、數字、空格及常見標點符號，且不可為空，用以顯示股票所屬的產業類別。	<code>CHECK (Weight &gt;= 0.00 AND Weight &lt;= 100.00)</code>

-- 建立 ETF\_Holdings 資料表

```
CREATE TABLE ETF_Holdings (
  ETF_Id VARCHAR(10) NOT NULL,
  Ticker_Symbol VARCHAR(10) NOT NULL,
  Weight DECIMAL(5,2) NOT NULL,
  PRIMARY KEY (ETF_Id, Ticker_Symbol),
  FOREIGN KEY (Ticker_Symbol) REFERENCES Stock_list(Ticker_Symbol),
  CHECK (ETF_Id REGEXP '^[A-Za-z0-9]{1,10}$'),
  CHECK (Ticker_Symbol REGEXP '^[A-Za-z0-9]{1,10}$'),
  CHECK (Weight >= 0.00 AND Weight <= 100.00)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
```

-- 範例：插入 ETF「0050」的成分股組成

```
INSERT INTO ETF_Holdings (ETF_Id, Ticker_Symbol, Weight)
VALUES ('0050', '2330', 45.00);
```

### 交易紀錄表 (Transaction)

欄位名稱	資料型態	是否 可為 空	欄位 說明	值域	實際資料舉 例
Transaction_Id (PK)	INT	N	交易 代號	$\geq 1$ 的整數	1
User_Id (FK)	VARCHAR(50)	N	使用 者代 號	參考 Users.User_Id	U000001
ETF_Id (FK)	VARCHAR(10)	N	ETF 代號	參考 ETF.ETF_Id	0050
Transaction_Type	ENUM('Buy','Sell')	N	交易 類型	僅可為 'Buy' 或 'Sell'	Buy
Shares	INT	N	買賣 股數	$> 0$ 的整數	100
Price	DECIMAL(10,2)	N	交易 價格	$\geq 0$ ，最多小數第 2 位	125.50
Transaction_Date	TIMESTAMP	N	交易 時間	時間格式：YYYY-MM-DD HH:MM:SS	2025-05-06 10:00:00

欄位名稱	值域限制說明	確認方式 ( MySQL )
Transaction_Id (PK)	必須為大於等於 1 的整數，且不可為空，用於唯一識別每一筆交易	CHECK (Transaction_Id >= 1)
User_Id (FK)	必須為長度不超過 50 個字元的字串，且不可為空，且其值必須對應至 Users 表中的 User_Id。	CHECK (char_length(User_Id) <= 50)
ETF_Id (FK)	必須為長度 1 至 10 個字元的字串，且不可為空，且其值必須對應至 ETF 表中的 ETF_Id。	CHECK (ETF_Id REGEXP '^[0-9A-Za-z]{1,10}\$')
Transaction_Type	僅可接受字串 'Buy' 或 'Sell'，且不可為空，用以區分買入或賣出交易類型。	已使用 ENUM('Buy','Sell') 約束
Shares	必須為大於 0 的整數，且不可為空，用以表示此筆交易的股數。	CHECK (Shares > 0)
Price	必須為大於或等於 0 且最多保留兩位小數的十進位數，且不可為空，用以記錄每單位交易價格。	CHECK (Price >= 0)
Transaction_Date	必須時間格式：YYYY-MM-DD HH:MM:SS	無需額外CHECK約束 ( MySQL內建驗證 )

-- 建立交易紀錄表

```
CREATE TABLE 'Transaction' (
  Transaction_Id INT PRIMARY KEY,
  User_Id VARCHAR(50) NOT NULL,
  ETF_Id VARCHAR(10) NOT NULL,
  Transaction_Type ENUM('Buy','Sell') NOT NULL,
  Shares INT NOT NULL,
  Price DECIMAL(10,2) NOT NULL,
  Transaction_Date TIMESTAMP NOT NULL,
  FOREIGN KEY (User_Id) REFERENCES Users(User_Id),
  FOREIGN KEY (ETF_Id) REFERENCES ETF(ETF_Id),
  CHECK (Shares > 0),
  CHECK (Price >= 0),
  CHECK (Transaction_Type IN ('Buy','Sell'))
);
```

-- 範例：記錄 001 使用者於 2025-04-29 買進 0050 100 股，單價 167.80

```
INSERT INTO 'Transaction' (User_Id, ETF_Id, Transaction_Type, Shares, Price,
Transaction_Date)
VALUES ('U000001', '0050', 'Buy', 100, 168.80, '2025-04-29');
```



## 持倉資料表 (Portfolio)

欄位名稱	資料型態	是否 可為 空	欄位說明	值域	實際資料舉例
Portfolio_Id (PK)	INT	N	持倉代號	≥ 1 的整數	1
User_Id (FK)	VARCHAR(50)	N	使用者代號	參考 Users.User_Id	U000001
ETF_Id (FK)	VARCHAR(10)	N	ETF 代號	參考 ETF.ETF_Id	0050
Shares_Held	INT	N	持有股數	>= 0 的整數	500
Average_Cost	DECIMAL(10,2)	N	平均成本	≥ 0，最多小數第 2 位	175
Last_Updated	TIMESTAMP	N	最後更新日期	時間格式：YYYY-MM-DD HH:MM:SS	2025-05-06 10:00:00
欄位名稱	值域限制說明			確認方式 ( MySQL )	
Portfolio_Id (PK)	必須為大於等於 1 的整數，且不可為空，用以唯一識別持倉紀錄			CHECK (Portfolio_Id >= 1)	
User_Id (FK)	必須為長度不超過 50 個字元的字串，且不可為空，對應 Users.User_Id。			CHECK (CHAR_LENGTH(User_Id) <= 50)	
ETF_Id (FK)	必須為長度 1 至 10 個字元的字串，且不可為空，對應 ETF.ETF_Id。			CHECK (ETF_Id REGEXP '^[0-9A-Za-z]{1,10}\$')	
Shares_Held	必須為大於等於 0 的整數，且不可為空，用以表示目前持有該檔 ETF 的股數。			CHECK (Shares_Held >= 0)	
Average_Cost	必須為大於或等於 0 且最多保留兩位小數的十進位數，且不可為空，用以記錄每股平均成本。			CHECK (Average_Cost >= 0)	
Last_Updated	必須時間格式：YYYY-MM-DD HH:MM:SS			無需額外CHECK約束 ( MySQL內建驗證 )	

```

-- 建立持倉資料表
CREATE TABLE Portfolio (
  Portfolio_Id INT PRIMARY KEY ,
  User_Id VARCHAR(50) NOT NULL,
  ETF_Id VARCHAR(10) NOT NULL,
  Shares_Held INT NOT NULL,
  Average_Cost DECIMAL(10,2) NOT NULL,
  Last_Updated TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
  FOREIGN KEY (User_Id) REFERENCES Users(User_Id),
  FOREIGN KEY (ETF_Id) REFERENCES ETF(ETF_Id),
  CHECK (Portfolio_Id >= 1),
  CHECK (CHAR_LENGTH(User_Id) <= 50),
  CHECK (Shares_Held >= 0),
  CHECK (Average_Cost >= 0)
);

-- 範例：使用者代號 1 持有 0050 100 股，平均成本 167.80
INSERT INTO Portfolio (User_Id, ETF_Id, Shares_Held, Average_Cost)
VALUES ('U0000001', '0050', 100, 167.80);

```

## ETF 歷史價格表 (ETF\_HistoryPrice)

欄位名稱	資料型態	是否可為空	欄位說明	值域	實際資料舉例
PriceRecord_Id (PK)	INT	N	價格紀錄代號	從 1 開始遞增的整數	1
ETF_Id (FK)	VARCHAR(10)	N	ETF 代號	參考 ETF.ETF_Id	0050
Open_Price	DECIMAL(10,2)	N	開盤價	≥ 0，最多小數第 2 位	125.00
Close_Price	DECIMAL(10,2)	N	收盤價	≥ 0，最多小數第 2 位	125.50
High_Price	DECIMAL(10,2)	N	最高價	≥ 0，最多小數第 2 位	126.00
Low_Price	DECIMAL(10,2)	N	最低價	≥ 0，最多小數第 2 位	124.80
Volume	BIGINT	N	交易量	≥ 0 的整數	3500000
History_Date	DATE	N	日期	時間格式：YYYY-MM-DD	2025-05-05

欄位名稱	值域限制說明	確認方式 ( MySQL )
PriceRecord_Id (PK)	必須為從 1 開始連續遞增且大於等於 1 的整數，且不可為空，用以唯一識別每筆紀錄。	使用 AUTO_INCREMENT 已確保遞增
ETF_Id (FK)	必須為長度 1 至 10 個字元的字串，且不可為空，對應 ETF.ETF_Id。	CHECK (ETF_Id REGEXP '^[0-9A-Za-z]{1,10}\$')
Open_Price	必須為大於或等於 0 且最多保留兩位小數的十進位數，且不可為空，記錄當日開盤價。	CHECK (Open_Price >= 0)
Close_Price	必須為大於或等於 0 且最多保留兩位小數的十進位數，且不可為空，記錄收盤價。	CHECK (Close_Price >= 0)
High_Price	必須為大於或等於 0 且最多保留兩位小數的十進位數，且不可為空，記錄當日最高價。	CHECK (High_Price >= 0)
Low_Price	必須為大於或等於 0 且最多保留兩位小數的十進位數，且不可為空，記錄當日最低價。	CHECK (Low_Price >= 0)
Volume	必須為大於或等於 0 的整數，且不可為空，用以表示當日成交量。	CHECK (Volume >= 0)
History_Date	必須時間格式：YYYY-MM-DD	無需額外CHECK約束 ( MySQL 內建驗證 )

-- 建立歷史價格表

```
CREATE TABLE ETF_HistoryPrice (
  PriceRecord_Id INT PRIMARY KEY AUTO_INCREMENT,
  ETF_Id VARCHAR(10) NOT NULL,
  Open_Price DECIMAL(10,2) NOT NULL,
  Close_Price DECIMAL(10,2) NOT NULL,
  High_Price DECIMAL(10,2) NOT NULL,
  Low_Price DECIMAL(10,2) NOT NULL,
  Volume BIGINT NOT NULL,
  History_Date DATE NOT NULL,
  FOREIGN KEY (ETF_Id) REFERENCES ETF(ETF_Id),
  CHECK (Open_Price >= 0),
  CHECK (Close_Price >= 0),
  CHECK (High_Price >= 0),
  CHECK (Low_Price >= 0),
  CHECK (Volume >= 0)
);
```

-- 範例：紀錄 2025-04-29 之 0050 開盤 167.15、收盤 167.80、最高 168.00、最低 166.50、成交量 10830

```
INSERT INTO ETF_HistoryPrice (ETF_Id, Open_Price, Close_Price, High_Price, Low_Price,
Volume, History_Date)
VALUES ('0050', 167.15, 167.80, 168.00, 166.50, 10830, '2025-04-28');
```

## 父標籤表 (Category\_Level1)

欄位名稱	資料型態	是否可為空	欄位說明	值域	實際資料舉例
Category1_Id (PK)	INT	N	第一分類代號	從 1 開始遞增的整數	1
Category1_Name	VARCHAR(20)	N	第一分類名稱	長度 1~20 的文字	股票型
欄位名稱	值域限制說明		確認方式 ( MySQL )		
Category1_Id (PK)	必須為從 1 開始連續遞增且大於等於 1 的整數，且不可為空。		使用 AUTO_INCREMENT 已確保遞增		
Category1_Name	必須為長度 1 到 20 個字元的文字，且不可為空，僅可包含中、英文與數字。		CHECK (CHAR_LENGTH(Category1_Name) BETWEEN 1 AND 20)		

-- 建立父標籤表

```
CREATE TABLE Category_Level1 (
  Category1_Id INT PRIMARY KEY AUTO_INCREMENT,
  Category1_Name VARCHAR(20) NOT NULL,
  CHECK (CHAR_LENGTH(Category1_Name) BETWEEN 1 AND 20)
);
```

-- 範例：新增父標籤「股票型」

```
INSERT INTO Category_Level1 (Category1_Name) VALUES ('股票型');
```

### 子標籤表 (Category\_Level2)

欄位名稱	資料型態	是否 可為 空	欄位 說明	值域	實際 資料 舉例
Category2_Id (PK)	INT	N	第二 分類 代號	從 1 開始遞增的整數	1
Category1_Id (FK)	INT	N	第一 分類 代號	參考 Category_Level1.Category1_Id	1
Category2_Name	VARCHAR(20)	N	第二 分類 名稱	長度 1~20 的文字	市 值 型
欄位名稱	值域限制說明			確認方式 ( MySQL )	
Category2_Id (PK)	必須為從 1 開始連續遞增且大於等於 1 的整數，且不可為空。			使用 AUTO_INCREMENT 已確保遞增	
Category1_Id (FK)	必須為大於等於 1 的整數，且不可為空，其值必須對應至 Category_Level1.Category1_Id。			外鍵約束已確保參照完整性	
Category2_Name	必須為長度 1 到 20 個字元的文字，且不可為空，僅可包含中、英文與數字。			CHECK (CHAR_LENGTH(Category2_Name) BETWEEN 1 AND 20)	

-- 建立子標籤表

```
CREATE TABLE Category_Level2 (
  Category2_Id INT PRIMARY KEY AUTO_INCREMENT,
  Category1_Id INT NOT NULL,
  Category2_Name VARCHAR(20) NOT NULL,
  FOREIGN KEY (Category1_Id) REFERENCES Category_Level1(Category1_Id),
  CHECK (CHAR_LENGTH(Category2_Name) BETWEEN 1 AND 20)
);
```

-- 範例：新增子標籤「大型權值」屬於父標籤 1

```
INSERT INTO Category_Level2 (Category1_Id, Category2_Name) VALUES (1, '大型權值');
```

紀錄分類表 (ETF\_Category)

欄位名稱	資料型態	是否 可為 空	欄位說明	值域	實際資料舉例
Category_Id (PK)	INT	N	紀錄分類代號	從 1 開始遞增的整數	1
ETF_Id (FK)	VARCHAR(10)	N	ETF 代號	參考 ETF.ETF_Id	0050
Category2_Id (FK)	INT	N	第二分類代號	參考 Category_Level2.Category2_Id	1
欄位名稱	值域限制說明			確認方式 ( MySQL )	
Category_Id (PK)	必須為從 1 開始連續遞增且大於等於 1 的整數，且不可為空。			使用 AUTO_INCREMENT 已確保遞增	
ETF_Id (FK)	必須為長度 1 到 10 個字元的字串，且不可為空，其值必須對應至 ETF.ETF_Id。			外鍵約束已確保參照完整性	
Category2_Id (FK)	必須為大於等於 1 的整數，且不可為空，其值必須對應至 Category_Level2.Category2_Id。			外鍵約束已確保參照完整性	

-- 建立 ETF 與分類對應表

```
CREATE TABLE ETF_Category (
  Category_Id INT PRIMARY KEY AUTO_INCREMENT,
  ETF_Id VARCHAR(10) NOT NULL,
  Category2_Id INT NOT NULL,
  FOREIGN KEY (ETF_Id) REFERENCES ETF(ETF_Id),
  FOREIGN KEY (Category2_Id) REFERENCES Category_Level2(Category2_Id)
);
```

-- 範例：將 0050 歸類至子標籤 1 (大型權值)

```
INSERT INTO ETF_Category (ETF_Id, Category2_Id) VALUES ('0050', 1);
```

## 使用者基本資料表 (Users)

欄位名稱	資料型態	是否 可為 空	欄位說明	值域	實際資料舉例
User_Id (PK)	VARCHAR(50)	N	使用者 代號	長度 1~50 的文字	U000001
User_Name	VARCHAR(50)	N	使用者 名稱	長度 1~50 的文字	alice
Full_Name	VARCHAR(100)	N	全名	長度 1~100 的文字	Alice Chen
Email	VARCHAR(100)	N	電子郵件	Email 格式	<a href="mailto:alice@example.com">alice@example.com</a>
Phone_Number	VARCHAR(10)	N	電話號 碼	長度固定為 10 碼	0912345678
Role	ENUM('user','admin')	N	權限	僅限 'user' 或 'admin'	user
Max_Amount	INT	N	當日最 大交易 量	≥ 0 的整數	1000000
Users_Created_At	TIMESTAMP	N	帳號創 建日期	時間格式：YYYY-MM-DD HH:MM:SS	2025-05-06 10:00:00
欄位名稱	值域限制說明			確認方式 ( MySQL )	
User_Id (PK)	必須為長度 1~50 的文字，且不可為空，由程式依需產生			CHECK (CHAR_LENGTH(User_Id) BETWEEN 1 AND 50)	
User_Name	必須為長度 1 到 50 個字元的文字，且不可為空，可包含英數字、底線及常見標點。			CHECK (CHAR_LENGTH(User_Name) BETWEEN 1 AND 50)	
Full_Name	必須為長度 1 到 100 個字元的文字，且不可為空，可包含中英文及空格。			CHECK (CHAR_LENGTH(Full_Name) BETWEEN 1 AND 100)	
Email	必須符合標準電子郵件格式，且長度不超過 100 個字元，不可為空，用以作為聯絡與驗證依據。			CHECK (Email REGEXP '^[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Za-z]{2,}\$')	
Phone_Number	必須為長度固定 10 碼且僅包含數字的字串，格式如 0912345678，且不可為空。			CHECK (Phone_Number REGEXP '^[0-9]{10}\$')	
Role	僅可接受字串 'user' 或 'admin' 其中之一，且不可為空，用以設定使用者權限等級。			CHECK (Role IN ('user','admin'))	
Max_Amount	必須為大於或等於 0 的整數，且不可為空，用以限制使用者於單日內可執行之最大交易數量。			CHECK (Max_Amount >= 0)	
Users_Created_At	日期時間格式：YYYY-MM-DD HH:MM:SS，預設為當前時間			無需額外CHECK約束 ( MySQL內建驗證 )	

```

-- 建立使用者資料表
CREATE TABLE Users (
  User_Id VARCHAR(50) PRIMARY KEY,
  User_Name VARCHAR(50) NOT NULL,
  Full_Name VARCHAR(100) NOT NULL,
  Email VARCHAR(100) NOT NULL UNIQUE,
  Phone_Number VARCHAR(10) NOT NULL,
  Role ENUM('user', 'admin') NOT NULL,
  Max_Amount INT NOT NULL DEFAULT 0,
  Users_Created_At TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
  CHECK (CHAR_LENGTH(User_Id) BETWEEN 1 AND 50),
  CHECK (CHAR_LENGTH(User_Name) BETWEEN 1 AND 50),
  CHECK (CHAR_LENGTH(Full_Name) BETWEEN 1 AND 100),
  CHECK (Email REGEXP '^[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Za-z]{2,}$'),
  CHECK (Phone_Number REGEXP '^[0-9]{10}$'),
  CHECK (Role IN ('user', 'admin')),
  CHECK (Max_Amount >= 0)
);

-- 範例：新增使用者 Bob
INSERT INTO Users (User_Name, Full_Name, Email, Phone_Number, Role, Max_Amount)
VALUES ('bob', 'Bob Lee', 'bob@example.com', '0987654321', 'user', 500000);

```

## 使用者密碼 (Auth)

[!NOTE] 存放在其他資料庫(auth\_db)

欄位名稱	資料型態	是否可為空	欄位說明	值域	實際資料舉例
User_Id (PK)	VARCHAR(50)	N	使用者代號	參考 Users.User_Id	U000001
Password	VARCHAR(255)	N	使用者密碼	長度 1~255 的文字	abcdef
Last_Login	TIMESTAMP	N	最近登入	時間格式：YYYY-MM-DD HH:MM:SS	2025-05-06 10:00:00

```

-- 建立使用者密碼資料表 Auth
CREATE TABLE Auth (
  User_Id VARCHAR(50) PRIMARY KEY,
  Password VARCHAR(255) NOT NULL,
  Last_Login TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
);

-- 範例：新增使用者密碼為 'abcd'
INSERT INTO Auth (User_Id, Password) VALUES ('U000001', '王小明');

```



## 六、 使用者權限設定

### 1. 使用者

ETF\_DB

資料表	權限	說明
ETF	SELECT	查看ETF
Stock_list	SELECT	查看成分股清單
ETF_Holdings	SELECT	查看成分股佔比
Transaction	SELECT,INSERT,UPDATE,DELETE	讀寫交易紀錄
Portfolio	SELECT,INSERT,UPDATE,DELETE	讀寫持倉
ETF_HistoryPrice	SELECT	查看歷史價格
Category_Level1	SELECT	查看父標籤
Category_Level2	SELECT	查看子標籤
ETF_Category	SELECT	查看分類
Users	SELECT	查看使用者基本資料

AUTH\_DB

資料表	權限	說明
Auth	SELECT	比對密碼輸入是否正確

### SQL 語法

```
CREATE USER 'user'@'%' IDENTIFIED BY '222';  
GRANT SELECT ON etf_db.* TO 'user'@'%';  
GRANT INSERT, UPDATE, DELETE ON etf_db.Transaction TO 'user'@'%';  
GRANT INSERT, UPDATE, DELETE ON etf_db.Portfolio TO 'user'@'%';  
GRANT SELECT ON auth_db.* TO 'user'@'%';  
FLUSH PRIVILEGES;
```

## 2. 管理者

### ETF\_DB

資料表	權限	說明
ETF	SELECT,INSERT,UPDATE,DELETE,CREATE VIEW,LOCK TABLES,SHOW VIEW,DROP	讀寫ETF及備份
Stock_list	SELECT,INSERT,UPDATE,DELETE,CREATE VIEW,LOCK TABLES,SHOW VIEW,DROP	讀寫成分股清單及備份
ETF_Holdings	SELECT,INSERT,UPDATE,DELETE,CREATE VIEW,LOCK TABLES,SHOW VIEW,DROP	讀寫成分股佔比及備份
Transaction	SELECT,INSERT,UPDATE,DELETE,CREATE VIEW,LOCK TABLES,SHOW VIEW,DROP	讀寫交易紀錄及備份
Portfolio	SELECT,INSERT,UPDATE,DELETE,CREATE VIEW,LOCK TABLES,SHOW VIEW,DROP	讀寫持倉及備份
ETF_HistoryPrice	SELECT,INSERT,UPDATE,DELETE,CREATE VIEW,LOCK TABLES,SHOW VIEW,DROP	讀寫歷史價格及備份
Category_Level1	SELECT,INSERT,UPDATE,DELETE,CREATE VIEW,LOCK TABLES,SHOW VIEW,DROP	讀寫父標籤及備份
Category_Level2	SELECT,INSERT,UPDATE,DELETE,CREATE VIEW,LOCK TABLES,SHOW VIEW,DROP	讀寫子標籤及備份
ETF_Category	SELECT,INSERT,UPDATE,DELETE,CREATE VIEW,LOCK TABLES,SHOW VIEW,DROP	讀寫分類及備份
Users	SELECT,INSERT,UPDATE,DELETE,CREATE VIEW,LOCK TABLES,SHOW VIEW,DROP	讀寫使用者基本資料及備份

### AUTH\_DB

資料表	權限	說明
Auth	SELECT	比對密碼輸入是否正確

### SQL 語法

```
CREATE USER 'admin'@'%' IDENTIFIED BY '111';  
GRANT SELECT, INSERT, UPDATE, DELETE, CREATE VIEW, LOCK TABLES, SHOW VIEW, DROP ON etf_db.*  
TO 'admin'@'%' ;  
GRANT SELECT ON auth_db.* TO 'admin'@'%' ;  
FLUSH PRIVILEGES;
```

### 3. 資料庫管理者

ETF\_DB

資料表	權限	說明
ETF	ALL	讀寫ETF及備份與還原
Stock_list	ALL	讀寫成分股清單及備份與還原
ETF_Holdings	ALL	讀寫成分股佔比及備份與還原
Transaction	ALL	讀寫交易紀錄及備份與還原
Portfolio	ALL	讀寫持倉及備份與還原
ETF_HistoryPrice	ALL	讀寫歷史價格及備份與還原
Category_Level1	ALL	讀寫父標籤及備份與還原
Category_Level2	ALL	讀寫子標籤及備份與還原
ETF_Category	ALL	讀寫分類及備份與還原
Users	ALL	讀寫使用者基本資料及備份與還原

AUTH\_DB

資料表	權限	說明
Auth	ALL	讀寫密碼和更新最後登入以及備份與還原

### SQL 語法

```
CREATE USER 'DBA'@'localhost' IDENTIFIED BY '000';  
GRANT ALL PRIVILEGES ON etf_db.* TO 'DBA'@'localhost';  
GRANT ALL PRIVILEGES ON auth_db.* TO 'DBA'@'localhost';  
FLUSH PRIVILEGES;
```

## 七、 使用者 View(1~8)

### 使用者查詢 ETF 標籤層次結構

```
-- 1.ETF 標籤層次結構 View
CREATE OR REPLACE VIEW vw_etf_category_overview AS
SELECT
    c1.Category1_Id,
    c1.Category1_Name,
    c2.Category2_Id,
    c2.Category2_Name
FROM
    Category_Level1 c1
LEFT JOIN
    Category_Level2 c2
ON
    c2.Category1_Id = c1.Category1_Id
ORDER BY
    c1.Category1_Name,
    c2.Category2_Name;
```

### 使用方式

```
--1.1 查看所有標籤層次結構
SELECT * FROM vw_etf_category_overview;
```

### 說明

- 功能與目的：用於檢索 ETF 的標籤列表，包含父標籤及其對應的子標籤，提供 ETF 分類層次的 View，該功能主要用於展示給使用者，提供標籤名稱的選擇。
- 詳情：選擇父標籤&子標籤的 ID、名稱等欄位後，使用 LEFT JOIN 確保包含所有父標籤，即使某些父標籤沒有對應的子標籤也會顯示，最後按父標籤名稱和子標籤名稱進行排序。

執行結果：

Category1_Id	Category1_Name	Category2_Id	Category2_Name
93	主題	1125	ESG
93	主題	1117	不動產
93	主題	1124	半導體
93	主題	1118	生技
93	主題	1119	科技
93	主題	1120	綠能
93	主題	1121	金融
93	主題	1122	電信通訊
93	主題	1123	電動車
92	債券型	1112	中國債
92	債券型	1102	中短債
92	債券型	1101	中長債
92	債券型	1103	主權債
92	債券型	1109	亞債

## 使用者依分類查詢 ETF 產品

```
-- 2. 建立特定分類 ETF View
CREATE OR REPLACE VIEW vw_etf_by_category AS
SELECT DISTINCT
  e.ETF_Id,
  e.ETF_Name,
  e.Holders,
  e.Scale,
  e.ETF_Created_At,
  c1.Category1_Name,
  c2.Category2_Name
FROM ETF e
LEFT JOIN ETF_Category ec
  ON e.ETF_Id = ec.ETF_Id
LEFT JOIN Category_Level2 c2
  ON ec.Category2_Id = c2.Category2_Id
LEFT JOIN Category_Level1 c1
  ON c2.Category1_Id = c1.Category1_Id
ORDER BY e.ETF_Id;
```

## 使用方式

```
--2.1 查看特定父標籤和子標籤的 ETF
SELECT *FROM vw_etf_by_category
WHERE Category1_Name = '股票型' AND Category2_Name = '大型權值';
```

## 說明

- 功能與目的： 檢索符合特定父標籤和子標籤條件的 ETF 資訊，根據特定分類條件過濾 ETF 並輸出詳細資料，方便使用者快速找到所需類別的 ETF 產品。

- 詳情：此查詢為選擇不重複的 ETF 詳細資訊，包括 ETF 編號、名稱、持有人數等名稱。透過關聯 ETF、ETF\_Category、Category\_Level2 和 Category\_Level1 表，建立 ETF 與其分類的完整連結，可依據指定的父標籤和子標籤進行篩選，並按 ETF 編號排序輸出結果，例如範例為附標籤為"股票型"，子標籤為"大型權值"，該 View 就會呈現同時符合這兩項標籤的 ETF。

執行結果：

ETF_Id	ETF_Name	Holders	Scale	ETF_Created_At	Category1_Name	Category2_Name
0050	元大台灣50	877483	5107	2003-06-30	股票型	大型權值
0057	富邦摩台	477	2	2008-02-27	股票型	大型權值
0061	元大寶滬深	9646	26	2009-08-17	股票型	大型權值
006201	元大富櫃50	1955	3	2011-01-27	股票型	大型權值
006203	元大MSCI台灣	1244	8	2011-05-12	股票型	大型權值
006206	元大上證50	4624	13	2012-05-11	股票型	大型權值
006207	復華滬深	2169	7	2012-06-22	股票型	大型權值
006208	富邦台50	727777	2135	2012-07-17	股票型	大型權值
00631L	元大台灣50正2	46538	458	2014-10-31	股票型	大型權值
00632R	元大台灣50反1	51723	407	2014-10-31	股票型	大型權值
00633L	富邦上証正2	7342	96	2014-11-25	股票型	大型權值
00634R	富邦上証反1	941	2	2014-11-25	股票型	大型權值
00637L	元大滬深300正2	29272	273	2015-05-18	股票型	大型權值
00638R	元大滬深300反1	692	4	2015-05-18	股票型	大型權值
00639	富邦深100	10340	20	2015-06-09	股票型	大型權值
00640L	富邦日本正2	1870	9	2015-09-10	股票型	大型權值
00641R	富邦日本反1	1201	4	2015-09-10	股票型	大型權值

## 使用者透過名稱搜尋 ETF

```
-- 3. 建立 ETF 清單 View
CREATE OR REPLACE VIEW vw_etf_dropdown AS
SELECT
    ETF_Id,
    ETF_Name
FROM ETF
ORDER BY ETF_Id;
```

## 使用方式

```
-- 3.1 取得所有 ETF 列表（用於網頁的下拉選單）
SELECT * FROM vw_etf_dropdown;

-- 3.2 搜尋包含特定關鍵字的 ETF
SELECT * FROM vw_etf_dropdown
WHERE ETF_Name LIKE '%元大%';
```

## 說明

- 功能與目的：此查詢檢索所有 ETF 的列表，提供簡單的 ETF ID 和名稱列表，供用戶在(網頁)介面中選擇。
- 詳情：使用 LIKE 語法進行關鍵字搜尋，輸入"元大"，就會匹配名稱包含"元大"的 ETF。

執行結果：

(3.1)

ETF_Id	ETF_Name
0050	元大台灣50
0051	元大中型100
0053	元大電子
0055	元大MSCI金融
0056	元大高股息
0061	元大寶滬深
006201	元大富櫃50
006203	元大MSCI台灣
006206	元大上證50
00631L	元大台灣50正2
00632R	元大台灣50反1
00635U	期元大S&P黃金
00637L	元大滬深300正2
00638R	元大滬深300反1
00642U	期元大S&P石油
00646	元大S&P500
00647L	元大S&P500正2
00648R	元大S&P500反1
00660	元大歐洲50

(3.2)

ETF_Id	ETF_Name
0050	元大台灣50
0051	元大中型100
0052	富邦科技
0053	元大電子
0055	元大MSCI金融
0056	元大高股息
0057	富邦摩台
0061	元大寶滬深
006201	元大富櫃50
006203	元大MSCI台灣
006204	永豐臺灣加權
006205	富邦上証(新臺幣)
006206	元大上證50
006207	復華滬深
006208	富邦台50
00625K	富邦上証+R(人民幣)

## 使用者查詢 ETF 歷史價格與漲跌幅

```
-- 4. 建立 ETF 歷史價格 View
CREATE OR REPLACE VIEW vw_etf_price_history AS
SELECT
    hp.ETF_Id,
    e.ETF_Name,
    hp.History_Date,
    hp.Close_Price
FROM ETF_HistoryPrice hp
JOIN ETF e ON hp.ETF_Id = e.ETF_Id
ORDER BY hp.ETF_Id, hp.History_Date;
```

## 使用方式

```
-- 4.1 期間漲跌幅計算
SELECT
    start_data.ETF_Id,
    start_data.ETF_Name,
    start_data.History_Date AS Start_Date,
    start_data.Close_Price AS Start_Price,
    end_data.History_Date AS End_Date,
    end_data.Close_Price AS End_Price,
    ROUND((end_data.Close_Price - start_data.Close_Price) / start_data.Close_Price *
100, 2) AS Price_Change_Percentage
FROM
    (SELECT * FROM vw_etf_price_history
    WHERE ETF_Id = '0050' AND History_Date >= '2024-01-01'
    ORDER BY History_Date ASC LIMIT 1) start_data
JOIN
    (SELECT * FROM vw_etf_price_history
    WHERE ETF_Id = '0050' AND History_Date <= '2025-05-30'
    ORDER BY History_Date DESC LIMIT 1) end_data
ON start_data.ETF_Id = end_data.ETF_Id;
```

## 說明

- 功能與目的：計算 ETF '0050' 在 2024 年 1 月 1 日至 2025 年 5 月 30 日期間的價格漲跌幅與百分比，展示 ETF 在指定期間價格漲跌幅的表現。
- 詳情：關聯 ETF\_HistoryPrice 和 ETF 表，選擇 ETF 代號、ETF 名稱、歷史日期和收盤價格等欄位，按 ETF 編號和歷史日期排序。透過子查詢方式取得指定期間的起始和結束價格，計算漲跌幅百分比公式為(結束價 - 起始價) / 起始價 \* 100，並四捨五入至小數點後兩位。

## 執行結果：

ETF_Id	ETF_Name	Start_Date	Start_Price	End_Date	End_Price	Price_Change_Percentage
0050	元大台灣50	2024-01-02	134.90	2025-05-29	179.75	33.25



---

## 使用者管理投資組合持倉

-- 5. 用戶持倉查看視圖（相同 ETF 合併顯示）

```
CREATE VIEW v_user_portfolio AS
SELECT
    p.User_Id,
    p.ETF_Id,
    e.ETF_Name,
    SUM(p.Shares_Held) as Shares_Held,
    SUM(p.Shares_Held * p.Average_Cost) / SUM(p.Shares_Held) as Average_Cost,
    MAX(p.Last_Updated) as Last_Updated
FROM
    Portfolio p
JOIN
    ETF e ON p.ETF_Id = e.ETF_Id
GROUP BY
    p.User_Id, p.ETF_Id, e.ETF_Name;
```

## 使用方式

-- 5.1 查看 user001 的持倉

```
SELECT * FROM v_user_portfolio WHERE User_Id = 'user001';
```

-- 5.2 使用者買入後更新交易紀錄跟投資組合（修改引號內的參數）

```
INSERT INTO Transaction (
    Transaction_Id,
    User_Id,
    ETF_Id,
    Transaction_Type,
    Shares,
    Price,
    Transaction_Date
) VALUES (
    (SELECT COALESCE(MAX(Transaction_Id), 0) + 1 FROM Transaction t),
    'user001',          -- 修改：用戶 ID
    '0057',            -- 修改：ETF 代碼
    'Buy',              -- 固定：買入
    100,                -- 修改：股數
    150.50,             -- 修改：價格
    NOW()
);
```

-- 更新持股數

```
INSERT INTO Portfolio (
    Portfolio_Id,
    User_Id,
    ETF_Id,
    Shares_Held,
```

```

        Average_Cost,
        Last_Updated
    ) VALUES (
        (SELECT COALESCE(MAX(Portfolio_Id), 0) + 1 FROM Portfolio p),
        'user001',          -- 修改：用戶 ID
        '0057',             -- 修改：ETF 代碼
        100,                -- 修改：股數
        150.50,             -- 修改：價格
        NOW()
    ) ON DUPLICATE KEY UPDATE
        Shares_Held = Shares_Held + VALUES(Shares_Held),
        Average_Cost = ((Shares_Held * Average_Cost) + (VALUES(Shares_Held) *
VALUES(Average_Cost))) / (Shares_Held + VALUES(Shares_Held)),
        Last_Updated = NOW();
SELECT * FROM v_user_portfolio WHERE User_Id = 'user001';

```

--5.3 使用者賣出後更新交易紀錄跟投資組合(股數為 0 則刪除 ETF)

-- 插入賣出交易紀錄

```

INSERT INTO 'Transaction' (
    Transaction_Id, User_Id, ETF_Id, Transaction_Type,
    Shares, Price, Transaction_Date
) VALUES (
    (SELECT COALESCE(MAX(Transaction_Id), 0) + 1 FROM 'Transaction' t),
    'user001', '0057', 'Sell', 50, 160.00, NOW()
);

```

-- 更新持股數

```

UPDATE Portfolio
SET
    Shares_Held = Shares_Held - 50,
    Last_Updated = NOW()
WHERE
    User_Id = 'user001'
    AND ETF_Id = '0057'
    AND Shares_Held >= 50;

```

-- 刪除股數為 0 的持倉

```

DELETE FROM Portfolio
WHERE User_Id = 'user001'
    AND ETF_Id = '0057'
    AND Shares_Held = 0;
SELECT * FROM v_user_portfolio WHERE User_Id = 'user001';

```

## 說明

- 功能與目的：提供用戶投資組合的整合檢視，將相同 ETF 的多筆持倉記錄合併顯示，計算總持股數量和平均成本，並支援買入賣出交易操作和自動更新投資組合記錄。

- 詳情：關聯 Portfolio 和 ETF 表，按用戶 ID、ETF 編號和 ETF 名稱進行分組，使用 SUM 函數計算總持股數量，透過加權平均公式計算平均成本，並取得最後更新時間。買入操作會自動插入交易記錄並更新投資組合，若已持有相同 ETF 則更新股數和重新計算平均成本。賣出操作會減少持股數量，當股數歸零時自動刪除該持倉記錄。

執行結果：

(5.1 初始持倉)

User_Id	ETF_Id	ETF_Name	Shares_Held	Average_Cost	Last_Updated
user001	0050	元大台灣 50	2150	116.368837	2025-06-03 22:39:39
user001	0051	元大中型 100	1000	179.750000	2025-06-01 14:46:38
user001	0056	元大高股息	150	153.666667	2025-06-03 15:04:58
user001	00633L	富邦上証正 2	1000	36.870000	2025-06-03 15:44:08
user001	00661	元大日經 225	1000	50.000000	2025-06-01 17:40:43

(5.2 買入 100 股的 0057)

User_Id	ETF_Id	ETF_Name	Shares_Held	Average_Cost	Last_Updated
user001	0050	元大台灣 50	2150	116.368837	2025-06-03 22:39:39
user001	0051	元大中型 100	1000	179.750000	2025-06-01 14:46:38
user001	0056	元大高股息	150	153.666667	2025-06-03 15:04:58
user001	0057	富邦摩台	100	150.500000	2025-06-07 18:00:33
user001	00633L	富邦上証正 2	1000	36.870000	2025-06-03 15:44:08
user001	00661	元大日經 225	1000	50.000000	2025-06-01 17:40:43

(5.3 賣出 50 股的 0057)

User_Id	ETF_Id	ETF_Name	Shares_Held	Average_Cost	Last_Updated
user001	0050	元大台灣 50	2150	116.368837	2025-06-03 22:39:39
user001	0051	元大中型 100	1000	179.750000	2025-06-01 14:46:38
user001	0056	元大高股息	150	153.666667	2025-06-03 15:04:58
user001	0057	富邦摩台	50	150.500000	2025-06-07 18:02:12
user001	00633L	富邦上証正 2	1000	36.870000	2025-06-03 15:44:08
user001	00661	元大日經 225	1000	50.000000	2025-06-01 17:40:43

## 使用者查看 ETF K 線資料與每日變動

```
-- 6. 建立 ETF K 線資料與每日變動 View
CREATE OR REPLACE VIEW vw_etf_daily_kline AS
SELECT
    ETF_Id,
    History_Date,
    Open_Price,
    High_Price,
    Low_Price,
    Close_Price,
    Volume,
    LAG(Close_Price) OVER (PARTITION BY ETF_Id ORDER BY History_Date) AS prev_close,
    Close_Price - LAG(Close_Price) OVER (PARTITION BY ETF_Id ORDER BY History_Date) AS
daily_change,
    CASE
        WHEN LAG(Close_Price) OVER (PARTITION BY ETF_Id ORDER BY History_Date) IS NOT
NULL
        THEN ROUND(((Close_Price - LAG(Close_Price) OVER (PARTITION BY ETF_Id ORDER BY
History_Date)) / LAG(Close_Price) OVER (PARTITION BY ETF_Id ORDER BY History_Date)) *
100, 2)
        ELSE NULL
    END AS daily_change_percent
FROM ETF_HistoryPrice
ORDER BY ETF_Id, History_Date;
```

## 使用方式

```
-- 6.1 查看特定 ETF 在特定期間的 K 線資料
SELECT * FROM vw_etf_daily_kline
WHERE ETF_Id = '0050'
    AND History_Date BETWEEN '2025-01-10' AND '2025-03-16'
ORDER BY History_Date;

-- 6.2 查看特定日期所有 ETF 的表現
SELECT * FROM vw_etf_daily_kline
WHERE History_Date = '2025-06-07';
```

## 說明

- 功能與目的：提供完整的 ETF 每日 K 線資料，包含開盤、最高、最低、收盤價格及成交量，並計算每日價格變動金額和變動百分比，用於技術分析和投資決策參考使用。
- 詳情：從 ETF\_HistoryPrice 表選擇完整的 K 線資料欄位，使用 LAG 函數取得前一日收盤價，計算每日變動金額(當日收盤價-前一日收盤價)，每日變動百分比(變動金額/前一日收盤價\*100)，並四捨五入至小數點後兩位。支援查詢特定 ETF 在指定期間的價格變動軌跡，或檢視特定日期所有 ETF 的表現狀況。

執行結果：

(6.1 ETF 0050 在區間內價格變動)

ETF_Id	History_Date	Open_Price	High_Price	Low_Price	Close_Price	Volume	prev_close	daily_change	daily_change_percent
0050	2025-01-10	199.00	199.55	198.00	199.00	8321324	199.05	-0.05	-0.03
0050	2025-01-13	199.00	199.00	194.10	194.40	20876179	199.00	-4.60	-2.31
0050	2025-01-14	194.70	195.90	194.05	195.60	10877775	194.40	1.20	0.62
0050	2025-01-15	196.00	196.00	193.75	194.10	17384117	195.60	-1.50	-0.77
0050	2025-01-16	197.90	199.50	197.40	198.05	15587098	194.10	3.95	2.04
0050	2025-01-17	197.00	197.35	194.40	196.30	15137566	198.05	-1.75	-0.88
0050	2025-01-20	197.15	198.65	197.15	198.30	7417571	196.30	2.00	1.02
0050	2025-01-21	197.30	199.20	196.70	198.70	10612738	198.30	0.40	0.20
0050	2025-01-22	200.30	201.70	200.30	201.30	12864399	198.70	2.60	1.31
0050	2025-02-03	192.30	193.00	190.65	192.70	55789778	201.30	-8.60	-4.27
0050	2025-02-04	193.65	194.55	193.05	193.85	19284119	192.70	1.15	0.60
0050	2025-02-05	195.75	197.80	195.75	196.80	15239625	193.85	2.95	1.52

(6.2 特定日期所有 ETF 的價格變動)

ETF_Id	History_Date	Open_Price	High_Price	Low_Price	Close_Price	Volume	prev_close	daily_change	daily_change_percent
0050	2025-05-07	171.80	172.50	169.95	170.35	13284346	171.00	-0.65	-0.38
0051	2025-05-07	70.00	70.00	69.35	69.50	167245	69.95	-0.45	-0.64
0052	2025-05-07	163.00	164.95	161.90	163.45	351250	163.70	-0.25	-0.15
0053	2025-05-07	90.20	90.20	89.60	90.00	7000	89.85	0.15	0.17
0055	2025-05-07	26.74	26.93	26.62	26.67	971900	26.74	-0.07	-0.26
0056	2025-05-07	32.71	32.92	32.61	32.64	25983215	32.68	-0.04	-0.12
0057	2025-05-07	127.35	127.90	127.35	127.90	18002	127.70	0.20	0.16
0061	2025-05-07	17.78	17.93	17.70	17.75	762000	17.61	0.14	0.80
006201	2025-05-07	18.77	18.77	18.55	18.55	26000	18.77	-0.22	-1.17
006203	2025-05-07	79.70	79.70	79.10	79.10	2000	79.55	-0.45	-0.57
006204	2025-05-07	103.60	103.60	103.60	103.60	1000	103.95	-0.35	-0.34
006205	2025-05-07	30.68	31.23	30.68	30.80	382300	30.56	0.24	0.79
006206	2025-05-07	29.68	29.68	29.34	29.48	21000	29.14	0.34	1.17
006207	2025-05-07	23.27	23.27	23.03	23.13	210000	22.91	0.22	0.96
006208	2025-05-07	100.40	100.95	99.45	100.95	7068996	100.15	-0.10	-0.10
00625K	2025-05-07	7.28	7.28	7.28	7.28	0	7.28	0.00	0.00
00631L	2025-05-07	176.20	176.90	172.45	174.25	9327908	174.75	-0.50	-0.29
00632R	2025-05-07	24.45	24.69	24.37	24.55	95362386	24.59	-0.04	-0.16
00633L	2025-05-07	36.02	36.78	36.02	36.24	7552030	35.42	0.82	2.32

使用者查詢某股票包含在哪些 ETF 裡

-- 7. 建立某檔成分股包含在哪些 ETF 裡 View

```
CREATE VIEW vw_Stock_In ETF AS
SELECT
    s.Ticker_Symbol,
    s.Stock_Name,
    s.Sector,
    e.ETF_Id,
    e.ETF_Name,
    h.Weight
FROM ETF_Holdings h
JOIN Stock_list s ON h.Ticker_Symbol = s.Ticker_Symbol
JOIN ETF e ON h.ETF_Id = e.ETF_Id;
```

## 使用方式

-- 7.1 查看特定股票在哪些 ETF 裡

```
SELECT * FROM vw_Stock_In ETF
```

```
WHERE Stock_Name LIKE '%台積電%' OR Ticker_Symbol = '2330';
```

### 說明

- 功能與目的：讓使用者輸入股票名稱或代碼，查詢這檔股票在哪些 ETF 裡，用於技術分析和投資決策參考使用。
- 詳情：從 ETF\_Holdings 表搭配 Stock\_list 表查詢，使用者可輸入股票代碼（如：2330）或股票名稱（如：台積電），系統將自動比對符合條件的股票，並回傳。

### 執行結果：

(7.1 股票編號 2230(台積電)在哪些 ETF 內)

```
MariaDB [etf_db]> SELECT * FROM vw_Stock_In ETF
-> WHERE Stock_Name LIKE '%台積電%' OR Ticker_Symbol = '2330';
```

Ticker_Symbol	Stock_Name	Sector	ETF_Id	ETF_Name	Weight
2330	台積電	半導體業	0050	元大台灣50	55.50
2330	台積電	半導體業	0052	富邦科技	65.39
2330	台積電	半導體業	0053	元大電子	51.51
2330	台積電	半導體業	0057	富邦摩台	50.78
2330	台積電	半導體業	006203	元大MSCI台灣	50.84
2330	台積電	半導體業	006204	永豐臺灣加權	34.07
2330	台積電	半導體業	006208	富邦台50	55.45
2330	台積電	半導體業	00690	兆豐臺灣藍籌30	30.95
2330	台積電	半導體業	00692	富邦公司治理	49.40
2330	台積電	半導體業	00728	第一金工業30	29.02
2330	台積電	半導體業	00850	元大臺灣ESG永續	27.41
2330	台積電	半導體業	00881	國泰台灣5G+	29.87
2330	台積電	半導體業	00891	中信關鍵半導體	19.02
2330	台積電	半導體業	00892	富邦台灣半導體	24.16
2330	台積電	半導體業	00894	中信小資高價30	28.22
2330	台積電	半導體業	00896	中信綠能及電動車	12.90
2330	台積電	半導體業	00900	富邦特選高股息30	3.79
2330	台積電	半導體業	00901	永豐智能車供應鏈	29.50
2330	台積電	半導體業	00904	新光臺灣半導體30	29.13
2330	台積電	半導體業	00905	FT臺灣Smart	27.72
2330	台積電	半導體業	00912	中信臺灣智慧50	26.68
2330	台積電	半導體業	00913	兆豐台灣晶圓製造	23.94
2330	台積電	半導體業	00921	兆豐龍頭等權重	2.74
2330	台積電	半導體業	00922	國泰台灣領袖50	28.76
2330	台積電	半導體業	00923	群益台ESG低碳50	27.51
2330	台積電	半導體業	00927	群益半導體收益	13.44
2330	台積電	半導體業	00935	野村臺灣新科技50	29.40
2330	台積電	半導體業	00952	凱基台灣AI 50	3.72
2330	台積電	半導體業	009803	保德信市值動能50	29.23

使用者查詢某股票包含在哪些 ETF 裡

```
-- 8. 建立使用者持有的股份有哪些成分股 View
CREATE VIEW vw_User ETF_Holdings AS
SELECT
    u.User_Id,
    u.Full_Name,
    e.ETF_Id,
    e.ETF_Name,
    s.Ticker_Symbol,
    s.Stock_Name,
    s.Sector,
    h.Weight
FROM Portfolio p
JOIN Users u ON p.User_Id = u.User_Id
JOIN ETF e ON p.ETF_Id = e.ETF_Id
JOIN ETF_Holdings h ON e.ETF_Id = h.ETF_Id
JOIN Stock_list s ON h.Ticker_Symbol = s.Ticker_Symbol;
```

使用方式

```
-- 8.1 查詢某股票，在哪些使用者的 ETF 中出現
SELECT
    User_Id,
    Full_Name,
    ETF_Id,
    ETF_Name
FROM vw_User ETF_Holdings
WHERE Ticker_Symbol = '2330'
ORDER BY User_Id, ETF_Id;
```

說明

- 功能與目的：顯示每位使用者持有的 ETF 以及該 ETF 的成分股清單與比重。
- 詳情：讓使用者輸入股票代號或名稱，查詢該股票目前被哪些 ETF 納入成分股，並顯示在各 ETF 中的持股比重。此功能有助於投資人掌握該股票的市場影響力與 ETF 資金配置情形，作為投資決策與技術分析的參考依據。

執行結果：

(8.1 使用者持有的 ETF，有哪些 ETF 包括股票編號 2230(台積電))

```
MariaDB [etf_db]> SELECT
-> User_Id,
-> Full_Name,
-> ETF_Id,
-> ETF_Name
-> FROM vw_User ETF_Holdings
-> WHERE Ticker_Symbol = '2230'
-> ORDER BY User_Id, ETF_Id;
```

User_Id	Full_Name	ETF_Id	ETF_Name
adminbenchen	陳峻宇	0050	元大台灣50
adminbenchen	陳峻宇	0053	元大電子
adminbenchen	陳峻宇	006208	富邦台50
user001	王小明	0050	元大台灣50
user001	王小明	0050	元大台灣50
user010	林怡君	0053	元大電子

## 八、管理員 View(9~10)

管理員查看用戶投資組合持股明細

```
-- 9. 建立用戶投資組合持股明細 View
CREATE OR REPLACE VIEW vw_portfolio_detail AS
SELECT
  p.Portfolio_Id,
  p.User_Id,
  u.Full_Name,
  p.ETF_Id,
  e.ETF_Name,
  p.Shares_Held,
  p.Average_Cost,
  (p.Shares_Held * p.Average_Cost) AS Cost_Basis,
  p.Last_Updated
FROM Portfolio p
JOIN Users u ON p.User_Id = u.User_Id
JOIN ETF e ON p.ETF_Id = e.ETF_Id
WHERE p.Shares_Held > 0
ORDER BY p.User_Id, e.ETF_Id;
```



## 使用方式

```
-- 9.1 查看所有用戶的投資組合詳細
SELECT * FROM vw_portfolio_detail;

-- 9.2 查看特定用戶的投資組合詳細
SELECT * FROM vw_portfolio_detail
WHERE User_Id = 'user001';

-- 9.3 統計用戶投資分佈
SELECT
    ETF_Id,
    ETF_Name,
    COUNT(*) AS Holders,
    SUM(Shares_Held) AS Total_Shares,
    SUM(Cost_Basis) AS Total_Cost
FROM vw_portfolio_detail
GROUP BY ETF_Id, ETF_Name
ORDER BY Holders DESC;
```

## 說明

- 功能與目的：提供管理員檢視所有用戶投資組合持股的詳細摘要，整合用戶資訊、ETF 資訊和持股詳情，並計算成本基礎，支援用戶投資分析和 ETF 持有狀況統計。
- 詳情：關聯 Portfolio、Users 和 ETF 三個表格，選擇投資組合編號、用戶編號、用戶姓名等欄位。僅顯示持有股數大於零的投資組合記錄，按用戶編號和 ETF 編號排序。支援查看所有用戶或特定用戶的投資組合詳情，並可統計各 ETF 的持有人數、總持股數和總成本分佈情況。

## 執行結果：

### ( 9.1 查看所有用戶投資組合 )

Portfolio_Id	User_Id	Full_Name	ETF_Id	ETF_Name	Shares_Held	Average_Cost	Cost_Basis	Last_Updated
5	adminbenchen	陳峻宇	0050	元大台灣50	900	91.25	82125.00	2025-06-01 17:15:30
6	adminbenchen	陳峻宇	0051	元大中型100	1000	179.75	179750.00	2025-06-01 17:28:32
4	adminbenchen	陳峻宇	0053	元大電子	1	179.75	179.75	2025-06-01 16:36:36
7	adminbenchen	陳峻宇	0056	元大高股息	100	50.00	5000.00	2025-06-01 17:34:18
16	adminbenchen	陳峻宇	006208	富邦台50	500	103.99	51950.00	2025-06-03 23:21:27
3	adminbenchen	陳峻宇	00635U	期元大S&P黃金	1	179.75	179.75	2025-06-01 16:36:30
9	user001	王小明	0050	元大台灣50	2100	115.33	242193.00	2025-06-03 22:39:39
10	user001	王小明	0050	元大台灣50	50	160.00	8000.00	2025-06-03 14:51:13
1	user001	王小明	0051	元大中型100	1000	179.75	179750.00	2025-06-01 14:46:38
11	user001	王小明	0056	元大高股息	100	150.50	15050.00	2025-06-03 15:04:31
12	user001	王小明	0056	元大高股息	50	160.00	8000.00	2025-06-03 15:04:58
13	user001	王小明	00633L	富邦上証正2	1000	36.87	36870.00	2025-06-03 15:44:08
8	user001	王小明	00661	元大日經225	1000	50.00	50000.00	2025-06-01 17:40:43
15	user002	林小美	00951	台新日本半導體	1000	7.36	7360.00	2025-06-03 23:02:23

### ( 9.2 查看特定用戶投資組合 )

Portfolio_Id	User_Id	Full_Name	ETF_Id	ETF_Name	Shares_Held	Average_Cost	Cost_Basis	Last_Updated
9	user001	王小明	0050	元大台灣50	2100	115.33	242193.00	2025-06-03 22:39:39
10	user001	王小明	0050	元大台灣50	50	160.00	8000.00	2025-06-03 14:51:13
1	user001	王小明	0051	元大中型100	1000	179.75	179750.00	2025-06-01 14:46:38
12	user001	王小明	0056	元大高股息	50	160.00	8000.00	2025-06-03 15:04:58
11	user001	王小明	0056	元大高股息	100	150.50	15050.00	2025-06-03 15:04:31
13	user001	王小明	00633L	富邦上証正2	1000	36.87	36870.00	2025-06-03 15:44:08
8	user001	王小明	00661	元大日經225	1000	50.00	50000.00	2025-06-01 17:40:43

( 9.3 查看 ETF 持有狀況)

ETF_Id	ETF_Name	Holders	Total_Shares	Total_Cost
0050	元大台灣50	3	3050	332318.00
0056	元大高股息	3	250	28050.00
0051	元大中型100	2	2000	359500.00
0053	元大電子	1	1	179.75
00635U	期元大S&P黃金	1	1	179.75
006208	富邦台50	1	500	51950.00
00951	台新日本半導體	1	1000	7360.00
00633L	富邦上証正2	1	1000	36870.00
00661	元大日經225	1	1000	50000.00

### 管理員查看近期交易記錄與統計分析

```
-- 10. 建立近期交易記錄 View
CREATE OR REPLACE VIEW vw_recent_transactions AS
SELECT
    t.Transaction_Id,
    t.User_Id,
    u.Full_Name,
    t.ETF_Id,
    e.ETF_Name,
    t.Transaction_Type,
    t.Shares,
    t.Price,
    (t.Shares * t.Price) AS Total_Amount,
    t.Transaction_Date
FROM 'Transaction' t
JOIN Users u ON t.User_Id = u.User_Id
JOIN ETF e ON t.ETF_Id = e.ETF_Id
ORDER BY t.Transaction_Date DESC;
```

### 使用方式

```
-- 10.1 查看最新 10 筆交易記錄
SELECT * FROM vw_recent_transactions LIMIT 10;

-- 10.2 查看今日所有交易
SELECT * FROM vw_recent_transactions
WHERE DATE(Transaction_Date) = CURDATE();

-- 10.3 統計交易量分析
SELECT
    ETF_Id,
    ETF_Name,
    Transaction_Type,
```

```

COUNT(*) AS Transactions_Time,
SUM(Shares) AS Total_Shares,
SUM(Total_Amount) AS Total_Amount
FROM vw_recent_transactions
WHERE Transaction_Date >= DATE_SUB(NOW(), INTERVAL 30 DAY)
GROUP BY ETF_Id, ETF_Name, Transaction_Type
ORDER BY Total_Amount DESC;

-- 10.4 查看用戶交易活躍度
SELECT
    User_Id,
    Full_Name,
    COUNT(*) AS Transaction_Count,
    SUM(Total_Amount) AS Transaction_Amount
FROM vw_recent_transactions
WHERE Transaction_Date >= DATE_SUB(NOW(), INTERVAL 30 DAY)
GROUP BY User_Id, Full_Name
ORDER BY Transaction_Count DESC;

```

## 說明

- 功能與目的：提供管理員檢視所有用戶 ETF 交易活動的完整記錄，整合交易資訊、用戶資訊和 ETF 資訊，支援交易筆數、用戶活躍度統計。
- 詳情：關聯 Transaction、Users 和 ETF 三個表格，選擇交易編號、用戶編號、用戶姓名等欄位，按交易日期降序排序以顯示最新交易。支援查看最新交易記錄、今日所有交易、近 30 天交易量分析和用戶交易活躍度統計。可統計各 ETF 的買賣次數、總股數和總金額，以及各用戶的交易次數和交易金額。

## 執行結果：

### (10.1 查看最新十筆交易)

Transaction_Id	Full_Name	ETF_Name	Transaction_Type	Shares	Price	Total_Amount	Transaction_Date
46	王小明	富邦摩台	Sell	50	160.00	8000.00	2025-06-03 16:26:56
45	王小明	富邦摩台	Buy	100	150.50	15050.00	2025-06-03 16:25:43
44	王小明	富邦上証正2	Buy	1000	36.87	36870.00	2025-06-03 15:44:08
43	王小明	元大寶滬深	Sell	1000	17.56	17560.00	2025-06-03 15:42:12
42	王小明	元大寶滬深	Buy	1000	17.56	17560.00	2025-06-03 15:42:00
41	王小明	富邦摩台	Sell	50	160.00	8000.00	2025-06-03 15:21:08
40	王小明	富邦摩台	Sell	50	160.00	8000.00	2025-06-03 15:11:57
39	王小明	富邦摩台	Sell	50	160.00	8000.00	2025-06-03 15:11:00
38	王小明	富邦摩台	Sell	50	160.00	8000.00	2025-06-03 15:10:29
37	王小明	富邦摩台	Buy	100	150.50	15050.00	2025-06-03 15:10:17

### (10.2 統計今日所有交易)

Transaction_Id	User_Id	Full_Name	ETF_Id	ETF_Name	Transaction_Type	Shares	Price	Total_Amount	Transaction_Date
79	user010	林怡君	0053	元大電子	Sell	3456	50.00	172800.00	2025-06-07 20:20:13
78	user010	林怡君	0053	元大電子	Buy	6000	50.00	300000.00	2025-06-07 20:20:03
77	user003	陳柏宇	00936	台新永續高息中小	Buy	333	50.00	16650.00	2025-06-07 20:19:34
76	user003	陳柏宇	007198	元大美債1-3	Buy	300	50.00	15000.00	2025-06-07 20:19:18
75	adminbenchen	陳峻宇	006208	富邦台50	Buy	200	50.00	10000.00	2025-06-07 20:18:00
74	adminbenchen	陳峻宇	00955	中信日本商社	Buy	3000	50.00	150000.00	2025-06-07 20:17:39
73	user001	王小明	0057	富邦摩台	Sell	50	160.00	8000.00	2025-06-07 18:03:30
72	user001	王小明	0057	富邦摩台	Sell	50	160.00	8000.00	2025-06-07 18:02:12
71	user001	王小明	0057	富邦摩台	Buy	100	150.50	15050.00	2025-06-07 18:00:33
70	user001	王小明	0057	富邦摩台	Sell	50	160.00	8000.00	2025-06-07 17:52:14
69	user001	王小明	0057	富邦摩台	Sell	50	160.00	8000.00	2025-06-07 17:51:00
68	user001	王小明	0057	富邦摩台	Buy	100	150.50	15050.00	2025-06-07 17:44:08
67	user001	王小明	0057	富邦摩台	Sell	50	160.00	8000.00	2025-06-07 16:42:03
66	user001	王小明	0057	富邦摩台	Buy	100	150.50	15050.00	2025-06-07 16:41:44
65	user001	王小明	0057	富邦摩台	Sell	50	160.00	8000.00	2025-06-07 16:37:47
64	user001	王小明	0057	富邦摩台	Sell	50	160.00	8000.00	2025-06-07 16:37:40
63	user001	王小明	0057	富邦摩台	Buy	100	150.50	15050.00	2025-06-07 16:36:54

(10.3 統計 ETF 近 30 天買賣次數與金額)

ETF_Id	ETF_Name	Transaction_Type	Transactions_Time	Total_Shares	Total_Amount
0050	元大台灣50	Buy	10	6602	678929.00
0057	富邦摩台	Sell	25	5950	661500.00
0051	元大中型100	Buy	3	2001	359679.00
0057	富邦摩台	Buy	11	2000	330250.00
0050	元大台灣50	Sell	6	2250	302250.00
0053	元大電子	Buy	2	6001	300179.75
0053	元大電子	Sell	1	3456	172800.00
00955	中信日本商社	Buy	1	3000	150000.00
006208	富邦台50	Buy	2	1200	113900.00
006208	富邦台50	Sell	1	500	51950.00
00679B	元大美債20年	Buy	1	1000	50000.00
00661	元大日經225	Buy	1	1000	50000.00
00633L	富邦上証正2	Buy	1	1000	36870.00
00679B	元大美債20年	Sell	1	1000	25210.00
0056	元大高股息	Buy	2	200	20050.00
006201	元大富櫃50	Sell	1	1000	18800.00
006201	元大富櫃50	Buy	1	1000	18800.00
0061	元大寶滬深	Sell	1	1000	17560.00
0061	元大寶滬深	Buy	1	1000	17560.00
00936	台新永續高息中小	Buy	1	333	16650.00
0056	元大高股息	Sell	2	100	16000.00
00719B	元大美債1-3	Buy	1	300	15000.00
00951	台新日本半導體	Buy	1	1000	7360.00
00635U	期元大S&P黃金	Buy	1	1	179.75
00637L	元大滬深300正2	Buy	1	1	50.00

(10.4 統計用戶近 30 天總交易次數與金額)

User_Id	Full_Name	Transaction_Count	Transaction_Amount
user001	王小明	52	1193548.00
adminbenchen	陳峻宇	22	1726169.50
user010	林怡君	2	472800.00
user003	陳柏宇	2	31650.00
user002	林小美	1	7360.00

## 九、 資料筆數

```
MariaDB [etf_db]> SELECT 'category_level1' AS 資料表, COUNT(*) AS 筆數 FROM category_level1
-> UNION
-> SELECT 'category_level2', COUNT(*) FROM category_level2
-> UNION
-> SELECT 'etf', COUNT(*) FROM etf
-> UNION
-> SELECT 'etf_category', COUNT(*) FROM etf_category
-> UNION
-> SELECT 'etf_historyprice', COUNT(*) FROM etf_historyprice
-> UNION
-> SELECT 'etf_holdings', COUNT(*) FROM etf_holdings
-> UNION
-> SELECT 'portfolio', COUNT(*) FROM portfolio
-> UNION
-> SELECT 'stock_list', COUNT(*) FROM stock_list
-> UNION
-> SELECT 'transaction', COUNT(*) FROM transaction
-> UNION
-> SELECT 'users', COUNT(*) FROM users
-> UNION
-> SELECT 'auth_db.auth', COUNT(*) FROM auth_db.auth;
```

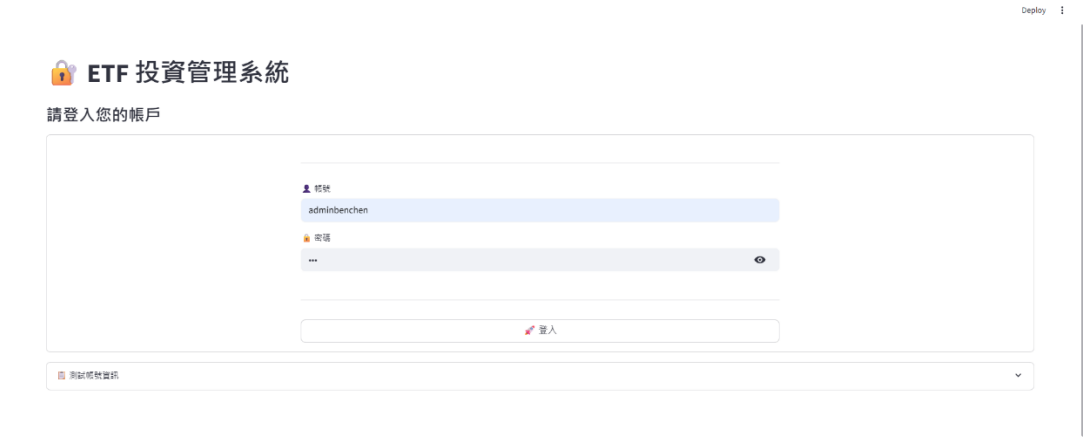
資料表	筆數
category_level1	7
category_level2	73
etf	287
etf_category	1791
etf_historyprice	414084
etf_holdings	3059
portfolio	18
stock_list	517
transaction	79
users	11
auth_db.auth	11

## 十、 DEMO

<https://www.youtube.com/watch?v=Hx2qvtQ5Txx>

頁面預覽

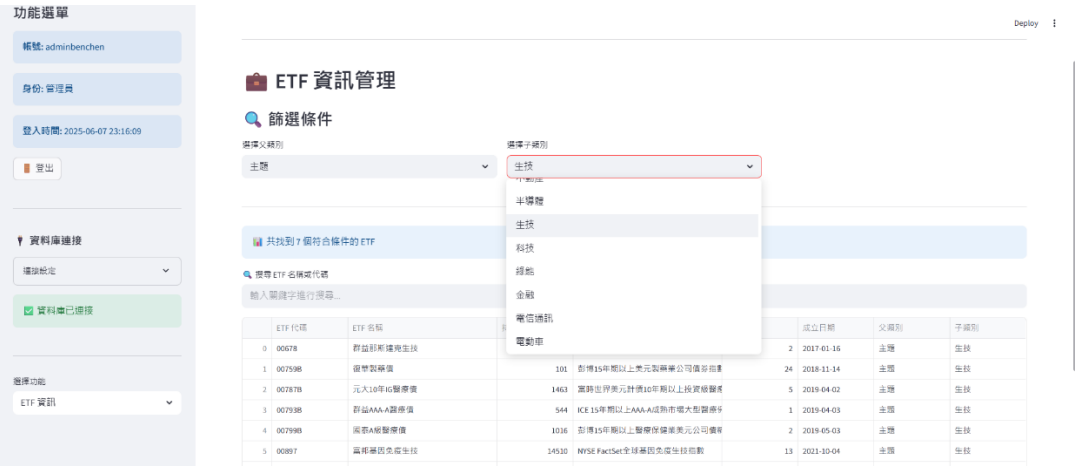
1. 帳密登入



2. 資訊總覽&查看最近交易紀錄



3. 透過附標籤跟子標籤篩選 ETF，也可以透過查詢名稱來篩選



## 4. 用戶管理

功能選單

帳號: adminbenchen

身份: 管理員

登入時間: 2025-06-07 23:16:09

登出

資料庫連接

選擇功能

用戶管理

ETF 投資管理系統

用戶管理

User_Id	User_Name	Full_Name	Email	Phone_Number	Role	Max_Amount	Users_Created_At
0	adminbenchen	陳裕宇	benchen@example.com	090000000	admin	2000000	2025-06-01 15:40:36
1	user001	john123	john.wang@example.com	0912345678	user	1000000	2025-05-28 00:41:04
2	user002	alice	alice.lin@example.com	0923000001	user	500000	2025-06-03 22:56:30
3	user003	bob	bob.chen@example.com	0933000002	user	300000	2025-06-03 22:56:30
4	user004	cindy	cindy.wu@example.com	0955000003	user	800000	2025-06-03 22:56:30
5	user005	david	david.chang@example.com	0966000004	user	250000	2025-06-03 22:56:30
6	user006	eva	eva.chen@example.com	0977000005	user	600000	2025-06-03 22:56:30
7	user007	frank	frank.huang@example.com	0988000006	user	450000	2025-06-03 22:56:30
8	user008	grace	grace.li@example.com	0911000007	user	700000	2025-06-03 22:56:30
9	user009	henry	henry.wu@example.com	0900000008	user	350000	2025-06-03 22:56:30

5. 即時交易，透過證交所抓取最新價格，用戶選擇好 ETF 後，可以按"使用此價格"後，數字會帶入到交易時的價格，下方會列出交易紀錄

功能選單

帳號: adminbenchen

身份: 管理員

登入時間: 2025-06-07 23:16:09

登出

資料庫連接

選擇功能

交易記錄

0050 - 元大台灣50

交易用戶: 陳裕宇

交易金額: \$50,000.00

提交交易

ETF 即時資訊

元大台灣50 (0050)

當前價格	\$181.95	最高價	\$182.40
開盤價	\$181.90	最低價	\$181.50
昨收價	\$181.65		
累積成交量	8146張		

選擇或交易

漲 \$0.30 (+0.17%)

更新時間: 2025/06/06 13:30:00

使用當前價格

6. 投資組合統計，可以查看用戶當前持倉佔比與 ETF 損益，上方可查看詳細持股清單



7. 顯示該區間 K 線圖表，下方附有交易量長條圖



## 十一、 備份&還原資料庫

管理員: etf\_db 備份

資料庫管理員: auth\_db 備份、auth\_db 還原、etf\_db 還原

etf\_db(一般內容)

備份:

```
mysqldump -u admin -p etf_db > etf_db_backup.sql
```

還原:

```
mysql -u DBA -p etf_db < etf_db_backup.sql
```



auth\_db(密碼)

備份：

```
mysqldump -u DBA -p auth_db > auth_db_backup.sql
```

還原：

```
mysql -u DBA -p auth_db < auth_db_backup.sql
```

## 十二、 參考資料/處理方式

ETF 基本資料表 (ETF):

- 證交所 api: <https://www.twse.com.tw/zh/ETFortune/ajaxProductsResult>
- 櫃買中心 api: <https://info.tpex.org.tw/api/etfFilter>

ETF 歷史價格表 (ETF\_HistoryPrice):

- 2008 年以前資料: finmind  
<https://finmindtrade.com/analysis/#/data/document>
- 2008 年以後資料: Yahoo finance 套件: yfinance (官網頁面顯示, 以 ETF:0050 舉例 <https://finance.yahoo.com/quote/0050.TW/history/>)

紀錄分類表 (ETF\_Category)、父標籤表 (Category\_Level1)、子標籤表 (Category\_Level2):

利用證交所、櫃買中心篩選器的篩選結果:

<https://www.twse.com.tw/zh/ETFortune/products>

- 比如總共有 A~E 這些 ETF, 選擇"股票型"標籤後, 剩下 A、C、D 這些 ETF, 就可以將"股票型"這個標籤分配給 A、C、D 這三檔 ETF, 其餘標籤同理
- 配息月份標籤: <https://www.twse.com.tw/zh/ETFortune/dividendCalendar>

網頁端即時股價

- 證交所 api(以 ETF:0050 舉例):  
[https://mis.twse.com.tw/stock/api/getStockInfo.jsp?json=1&delay=0&ex\\_ch=tse\\_0050.tw](https://mis.twse.com.tw/stock/api/getStockInfo.jsp?json=1&delay=0&ex_ch=tse_0050.tw)

網頁端實作

- 此專題網頁端實作有透過生成式 AI 協助製作
- streamlit: 網頁端框架

- Plotly : 圖表顯示
- 參考教學:
  - streamlit、streamlit 套件: <https://codinglu.tw/2024/10/streamlit-dashboard/>
  - 證交所 API: <https://hackmd.io/@aaronlife/python-ex-stock-by-api>
  - Yahoo finance 套件:<https://ithelp.ithome.com.tw/articles/10341344>
  - ETF 成分股(僅包含部分 ETF 的成分股資料):  
[https://mopsov.twse.com.tw/mops/web/t78sb03\\_q2](https://mopsov.twse.com.tw/mops/web/t78sb03_q2)

### 十三、心得

田晉嘉：

在學期實作的過程中，從資料表設計、值域設定到 view，以及匯入資料。view 以及 ER 圖的部分算是專題最麻煩的地方，要都先設定對才能去實作後面的部分。在專題最後整理 GitHub 時，因為不常接觸所以手忙腳亂，但因為以後工作或是研究所都會用到，所以是一次寶貴的學習經驗。

總之，這次專題讓我把資料庫的基本觀念跟工具實際用過一次，對資料庫設計和實作有更清楚的想法，也知道專題過程不是只有寫出來，要適時調整跟修正。

周偉宸：

在這次期末專案中，雖然我們各自都有主要負責的部分，但每一步我們都有深入討論交流，所以即便其他地方不是自己負責的，也有彼此學習到各個部份。

在我們討論如何設計資料表的時候，透過將需求條列出來，再將需求分析後畫成 ER Diagram，最後轉化為實際的資料庫 schema。整個過程讓我對於資料庫的規劃有更進一步的理解，也明白到資料的完整性與一致性及值域規劃對於資料庫的結構有多麼重要。而設計 VIEW 的時候也有遇到一些挑戰，像是規劃好每個使用者的權限、怎麼查詢會更有效率等等。透過與組員的討論和不斷嘗試後，最終也順利解決了這些問題。這些經歷讓我對於資料庫的實作層面有了更全面的認識，也讓我明白團隊合作的重要性。

陳峻宇：

在這次資料庫系統課程的專題中，會選擇以 ETF 為主題一方面是我平常有在投資 ETF，比較了解實際數據可以從哪裡獲取、資料表欄位會有什麼內容；另一方面是 ETF 本身數據很適合建立資料庫的 view 應用。

透過這次專題，讓我把課程學習到的內容實際應用，從概念層規劃、資料庫建置到實際的實作的過程都體驗一遍。在值域規範的部分，我學習到如何利用 check 語法和正則表達式讓定義更嚴謹；在 view 設計部分我也體驗到將資料邏輯封裝於 view 中，讓資料查詢跟處理更有效率，此外，在剛開始實作過程中，我常常遇到忘記刷新的問題，這讓我養成了只要一更改權限就會記得刷新的習慣；在網頁端實作則讓我了解到如何利用 streamlit 網頁框架搭配 plotly 圖表套件快速製作網頁，以及利用 python 將大量的 k 線數據匯入資料庫；最後在 Github 編排上我也學習到像是目錄、callout 等方式讓頁面更美觀與整潔。

陳億穎：

這次資料庫期末專題，我主要負責的部分包括概念層設計、ER 圖繪製、成分股相關的 View 撰寫，以及專題報告的整理。一開始雖然對 ETF 沒那麼熟，但透過查資料和跟組員討論，慢慢了解 ETF 的結構和資料特性，能夠設計出合適的資料表和欄位。

在建置資料庫的階段，我參與從零開始的資料表建立，包含欄位命名、資料型別設定、主鍵設計、外鍵參照等；值域限制部分，也有實作 CHECK 條件和正規表達式來確保資料正確性。透過這些設定，我更清楚什麼樣的欄位該有什麼規範，也體會到規劃清楚對後面查詢和操作的重要性。

在 ER 圖設計中，我根據功能需求先整理欄位，再將各表之間的關聯畫成關聯圖，這部分讓我更理解資料之間的邏輯結構。而在 View 的部分，像是 ETF 和成分股的關聯查詢，我設計了一個 View 將不同資料表整合，讓使用者可以輸入股票代碼就查出相關 ETF，這對查詢效率和資料整理都很有幫助。

整體來說，這次專題讓我把平常課堂學到的資料庫觀念實際動手做一次，從設計、建表、限制條件到查詢邏輯都有參與到，也加深我對資料庫系統設計和實務應用的理解。