

ETF 投資組合 管理系統

41143210 - 田晉嘉

41143219 - 周偉宸

41143238 - 陳峻宇

41143239 - 陳億穎

第八組



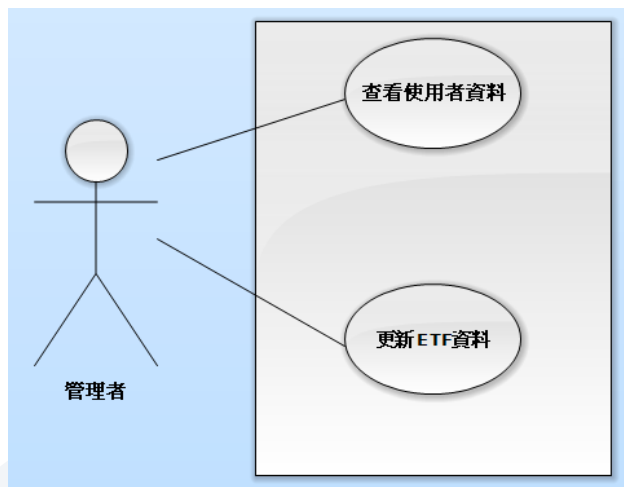
目錄

- 系統概述
- 使用案例
- 應用情境
- 功能需求
- ER Diagram
- 完整性限制
- 使用者權限設定
- View
- DEMO

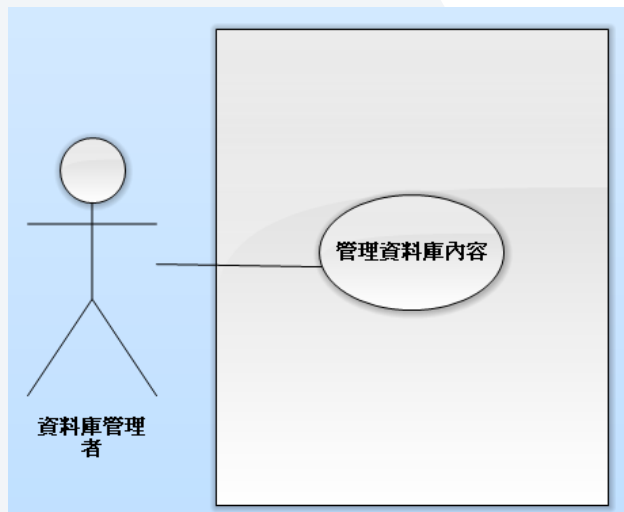
系統需求說明-系統概述

本系統旨在協助投資者有效整合與管理ETF投資資訊，從ETF基本資料分類、交易與持倉記錄，到即時績效與損益等。系統將透過穩定的數據存取與更新機制，幫助用戶精確掌握投資配置狀態，從而根據自身投資偏好及風險需求進行調整。

使用案例

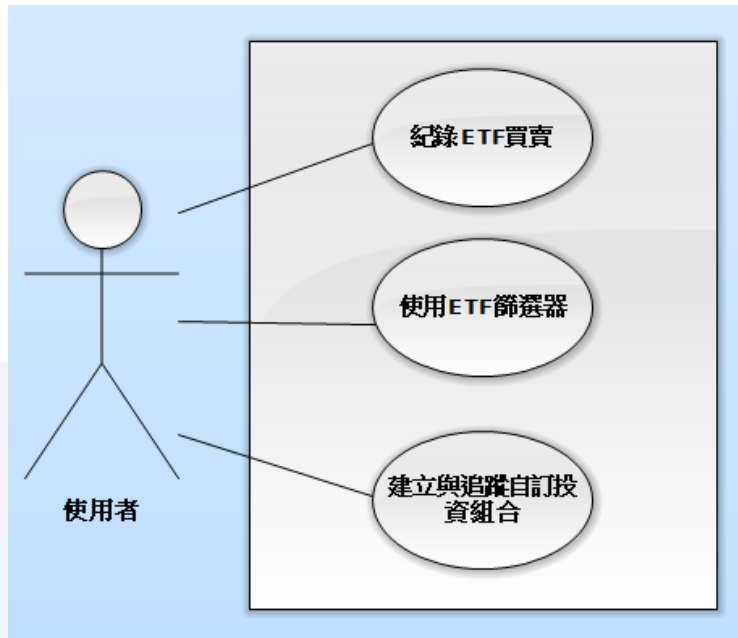


查看使用者帳號資料，對應
「Users(使用者基本資料)」
更新ETF歷史數據、對應
「ETF(基本資料表)」、
「ETF_HistoryPrice(ETF 歷史價格表)」



管理資料庫結構與資料，處理
資料異常或系統升級

使用案例



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(第一分類)」與「Category_Level2(第二分類)」
建立個人化投資組合，追蹤其報酬變化，對應「Portfolio (持倉資料表)」

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。

應用情境

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。

系統需求說明-功能需求

(1) 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

(2) ETF資料整合

- 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

系統需求說明-功能需求

(3) 績效與風險評估

- 生成K線走勢圖、以及交易量

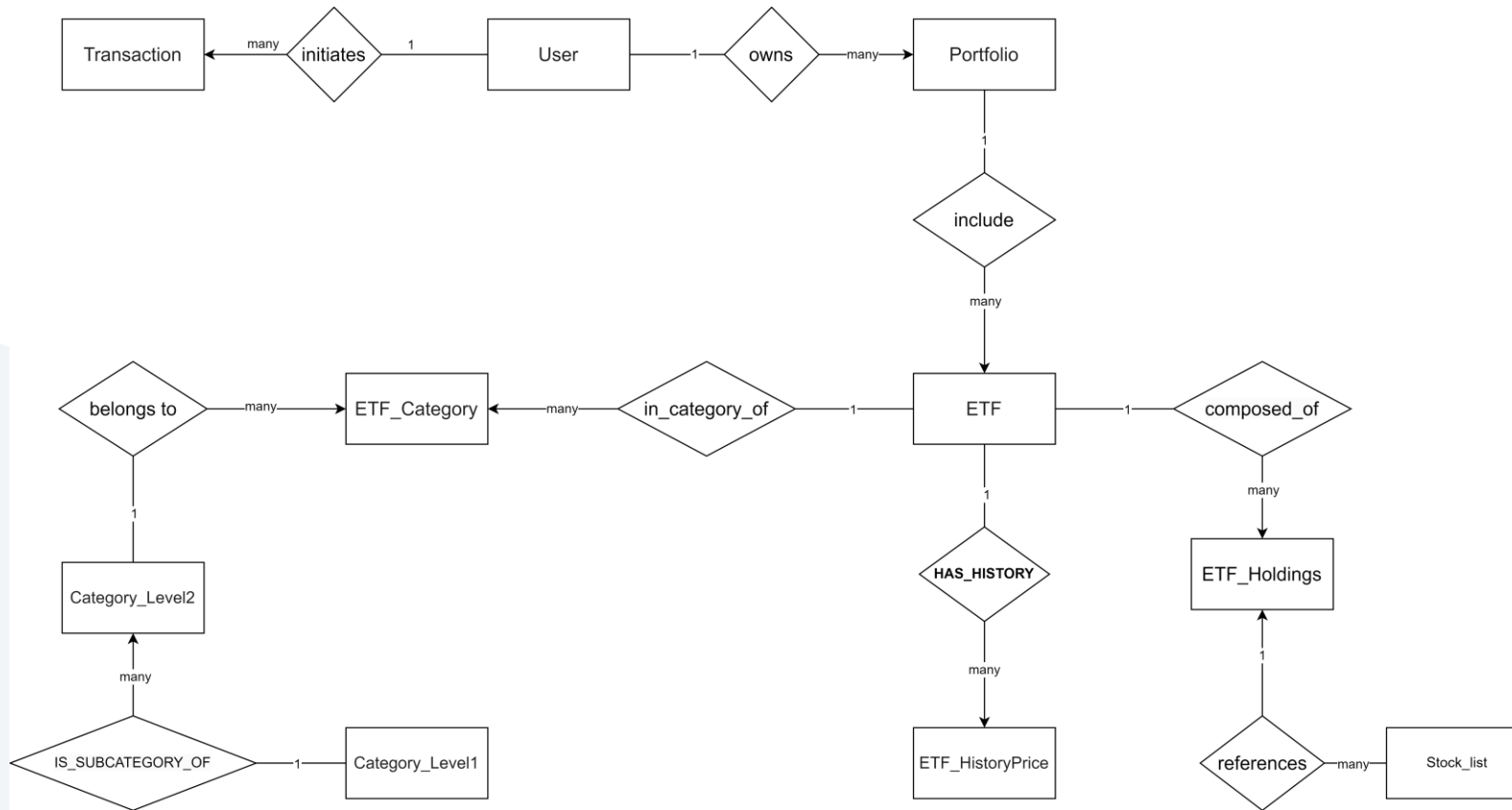
(4) 用戶管理

- 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

(5) 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
- 管理員與資料庫管理員定期備份資料庫

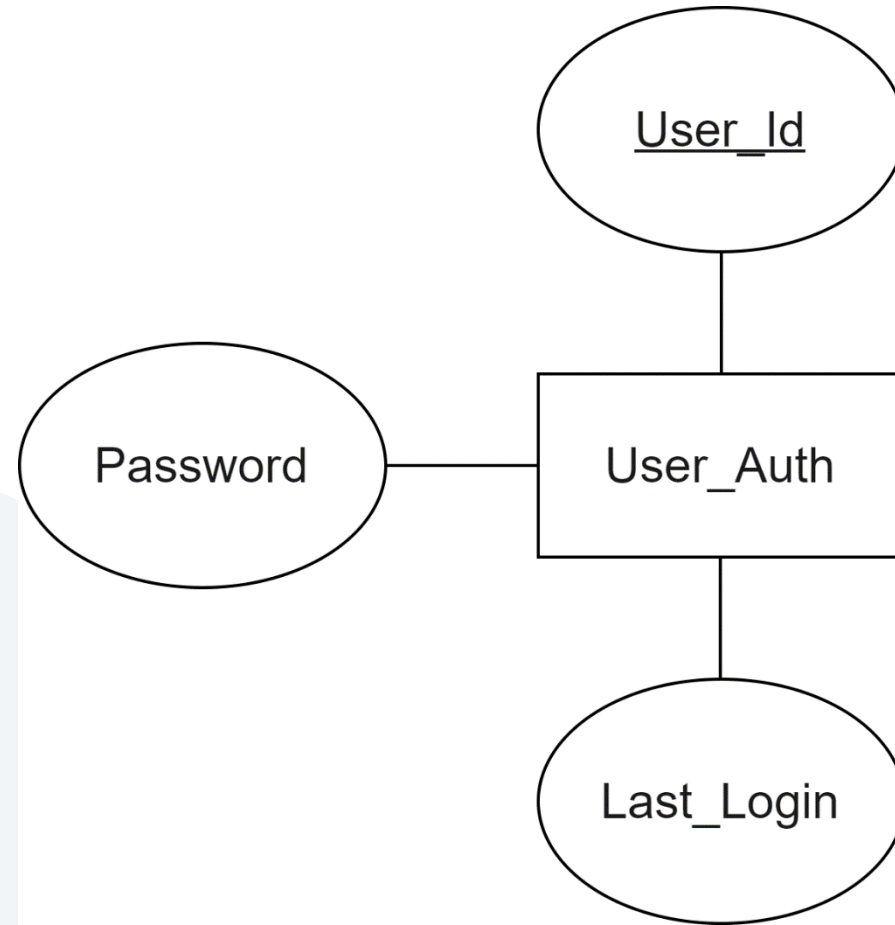
ER Diagram (完整版在github)



標籤

Category1_Id	父標籤名稱	Category2_Id	子標籤名稱
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
92	債券型	1107	全球債
92	債券型	1105	公債
92	債券型	1116	半年配
92	債券	1115	季配

ER Diagram



ER Diagram

各資料表之間的實體關係

ETF_HistoryPrice 的外來鍵 (FK) ETF_Id 參照 ETF 的主鍵 (PK) ETF_Id
一個 ETF 可以有多筆歷史價格記錄

ETF_Category 的外來鍵 Category2_Id 參照 Category_Level2 的主鍵 Category2_Id
一個 ETF 可以對應多個二級分類

ETF_Category 的外來鍵 ETF_Id 參照 ETF 的主鍵 ETF_Id
一個 ETF 可以對應多個二級分類

Category_Level2 的外來鍵 Category1_Id 參照 Category_Level1 的主鍵 Category1_Id
一個一級分類可以包含多個二級分類

Portfolio 的外來鍵 User_Id 參照 Users 的主鍵 User_Id
一個使用者可以有多個ETF

ER Diagram

各資料表之間的實體關係

Portfolio 的外來鍵 ETF_Id 參照 ETF 的主鍵 ETF_Id。
一個 ETF 可以被持有多組

Transaction 的外來鍵 User_Id 參照 Users 的主鍵 User_Id
一個使用者可以有多筆交易紀錄

Transaction 的外來鍵 ETF_Id 參照 ETF 的主鍵 ETF_Id
一個 ETF 可以有多筆交易紀錄

User_Auth 的外來鍵 User_Id 參照 Users 的主鍵 User_Id
一個使用者對應一筆認證資訊

ETF_Holdings 的外來鍵 ETF_Id 參照 ETF 的主鍵 ETF_Id
一個 ETF 能有一筆成分股

ETF_Holdings 的外來鍵 Ticker_Symbol 參照 ETF 的主鍵 Ticker_Symbol
一個 ETF 能有多筆成分股

完整性限制

ETF(基本資料表)

欄位名稱	欄位說明	資料型態	是否可為空	值域
ETF_Id(PK)	ETF代號	VARCHAR(10)	N	數字+英文組成的字串
ETF_Name	ETF名稱	VARCHAR(100)	N	長度 1~100 的文字
Holders	持有人數	INT	N	大於或等於 0 的整數
IndexName	追蹤指數	VARCHAR(50)	N	長度 1~50 的文字
Scale	規模(億)	INT	N	大於或等於 0 的整數
ETF_Created_At	創立時間	TIMESTAMP	N	時間格式：YYYY-MM-DD

完整性限制

```
-- 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');
```


完整性限制

Stock_list(成分股清單)				
欄位名稱	欄位說明	資料型態	是否可為空	值域
Ticker_Symbol (PK)	成分股代號	VARCHAR(10)	N	長度 1~10 的文字
Stock_Name	成分股名稱	VARCHAR(100)	N	長度 1~100 的文字
Sector	所屬產業類別	VARCHAR(50)	Y	長度 1~50 的文字

完整性限制

```
-- 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');
```

完整性限制

ETF(基本資料表)

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

完整性限制

```
-- 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);
```

完整性限制

Category_Level1(第一分類)				
欄位名稱	欄位說明	資料型態	是否可為空	值域
Category1_Id(PK)	第一分類代號	INT	N	從 1 開始遞增的整數
Category1_Name	第一分類名稱	VARCHAR(20)	N	長度 1~20 的文字

完整性限制

```
-- 建立第一分類表
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：新增第一分類「股票型」
INSERT INTO Category_Level1 (Category1_Name) VALUES ('股票型');
```

完整性限制

Category_Level2(第二分類)				
欄位名稱	欄位說明	資料型態	是否可為空	值域
Category2_Id(PK)	第二分類代號	INT	N	從 1 開始遞增的整數
Category1_Id(FK)	第一分類代號	INT	N	參考 Category_Level1.Category1_Id
Category2_Name	第二分類名稱	VARCHAR(20)	N	長度 1~20 的文字

完整性限制

```
-- 建立第二分類表
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：新增第二分類「大型權值」屬於第一分類1
INSERT INTO Category_Level2 (Category1_Id, Category2_Name) VALUES (1, '大型權值');
```


完整性限制

ETF_Category(紀錄分類)				
欄位名稱	欄位說明	資料型態	是否可為空	值域
Category_Id(PK)	紀錄分類代號	INT	N	從 1 開始遞增的整數
ETF_Id(FK)	ETF代號	INT	N	參考 ETF 資料表的 etf_id
Category2_Id(FK)	第二分類代號	INT	N	參考 Category_Level2.Category2_Id

完整性限制

```
-- 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歸類至第二分類1 (大型權值)
INSERT INTO ETF_Category (ETF_Id, Category2_Id) VALUES ('0050', 1);
```

完整性限制

Transaction(交易紀錄表)				
欄位名稱	欄位說明	資料型態	是否可為空	值域
Transaction_Id(PK)	交易代號	INT	N	大於或等於 1 的整數
User_Id(FK)	使用者代號	VARCHAR(50)	N	參考 Users.User_Id
ETF_Id(FK)	ETF代號	VARCHAR(10)	N	參考 ETF.ETF_Id
Transaction_Types	交易類型	ENUM('Buy', 'Sell')	N	僅可為 'Buy' 或 'Sell'，表示買進或賣出
Shares	買賣股數	INT	N	大於 0 的整數
Price	交易價格	DECIMAL(10,2)	N	大於或等於 0 的數值，最多小數第 2 位
Transaction_Date	交易時間	TIMESTAMP	N	時間格式：YYYY-MM-DD HH:MM:SS

完整性限制

```
-- 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');
```

完整性限制

Portfolio(持倉資料表)				
欄位名稱	欄位說明	資料型態	是否可為空	值域
Portfolio_Id(PK)	交易代號	INT	N	大於或等於 1 的整數
User_Id(FK)	使用者代號	VARCHAR(50)	N	參考 Users.User_Id
ETF_Id(FK)	ETF代號	VARCHAR(10)	N	參考 ETF.ETF_Id
Shares_Held	持有股數	INT	N	大於或等於 0 的整數
Average_Cost	平均成本	DECIMAL(10,2)	N	大於或等於 0 的數值， 最多小數第 2 位
Last_Updated	最後更新日期	TIMESTAMP	N	時間格式：YYYY-MM-DD HH:MM:SS

完整性限制

```
-- 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 ('U000001', '0050', 100, 167.80);
```

完整性限制

ETF_HistoryPrice(ETF歷史價格)				
欄位名稱	欄位說明	資料型態	是否可為空	值域
PriceRecord_Id(PK)	價格紀錄代號	INT	N	從 1 開始遞增的整數
ETF_Id(FK)	ETF代號	VARCHAR(10)	N	參考 ETF 資料表的etf_id
Open_Price	開盤價	DECIMAL(10,2)	N	大於或等於 0 的數值，最多小數第 2 位
Close_Price	收盤價	DECIMAL(10,2)	N	大於或等於 0 的數值，最多小數第 2 位
High_Price	最高價	DECIMAL(10,2)	N	大於或等於 0 的數值，最多小數第 2 位
Low_Price	最低價	DECIMAL(10,2)	N	大於或等於 0 的數值，最多小數第 2 位
Volume	交易量	BIGINT	N	大於或等於 0 的整數
History_Date	日期	Date	N	時間格式：YYYY-MM-DD

完整性限制

```
-- 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_D
VALUES ('0050', 167.15, 167.80, 168.00, 166.50, 10830, '2025-04-28');
```


完整性限制

Users(使用者基本資料)				
欄位名稱	欄位說明	資料型態	是否可為空	值域
<u>User_Id</u> (PK)	使用者代號	INT	N	從 1 開始遞增的整數
User_Name	使用者名稱	VARCHAR(50)	N	長度 1~50 的文字
Full_Name	全名	VARCHAR(100)	N	長度 1~100 的文字
Email	電子郵件	VARCHAR(100)	N	Email 格式
Phone_Number	電話號碼	VARCHAR(10)	N	長度固定為 10 碼
Role	權限	ENUM('user', 'admin')	N	僅限 'user' 或 'admin'
Max_Amount	當日最大交易量	INT	N	大於或等於 0 的整數
Users_Creat_At	帳號創建日期	TIMESTAMP	N	時間格式：YYYY-MM-DD HH:MM:SS

完整性限制

```
-- 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);
```

完整性限制

Auth(使用者密碼) (在其他資料庫)

欄位名稱	欄位說明	資料型態	是否可為空	值域
<u>User_Id</u> (PK)	使用者代號	VARCHAR(50)	N	參考 Users.User_Id
Password	使用者密碼	VARCHAR(255)	N	長度 1~255 的文字
Last_Login	最近登入	TIMESTAMP	N	時間格式：YYYY-MM-DD HH:MM:SS

完整性限制

```
-- 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');
```

使用者權限設定-使用者

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

使用者權限設定-使用者

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;
```

使用者權限設定-管理者

AUTH_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

資料表	權限	說明
Auth	SELECT	比對密碼輸入是否正確

使用者權限設定-管理者

SQL語法

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


使用者權限設定-資料庫管理者

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	讀寫密碼及更新最後登入

使用者權限設定-資料庫管理者

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;
```

View (使用者查詢ETF標籤層次結構)

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;
```



View

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

View (使用者依分類查詢ETF產品)

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';
```



View

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	1381	4	2015-09-10	股票型	大型權值

View (使用者透過名稱搜尋ETF)

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%';
```



View

(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&P 500
00647L	元大 S&P 500 正 2
00648R	元大 S&P 500 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)

View (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;
```

View

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

View (使用者管理投資組合持倉)

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)

-- 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';
```

(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 (I
        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';
```

View

(使用者查看ETF K線
資料與每日變動)

(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	1008	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	21060	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	100.05	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	9327948	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

```
-- 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;
```



View

(使用者查詢某股票
包含在哪些 ETF 裡)

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;
```



(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	富邦台灣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

View

🔗 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;
```



(使用者查詢某股票
包含在哪些 ETF 裡)

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';
```



(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

View (管理員查看用戶投資組合持股明細)

管理員查看用戶投資組合持股明細

```
-- 7.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;
```



View (管理員查看用戶投資組合持股明細)

使用方式

```
-- 7.1查看所有用戶的投資組合詳細  
SELECT * FROM vw_portfolio_detail;
```



```
-- 7.2查看特定用戶的投資組合詳細  
SELECT * FROM vw_portfolio_detail  
WHERE User_Id = 'user001';
```



```
-- 7.3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;
```



(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	103.90	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

View (管理員查看近期交易記錄與統計分析)

管理員查看近期交易記錄與統計分析

```
-- 8.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;
```



View (管理員查看近期交易記錄與統計分析)

使用方式

```
-- 8.1查看最新 10 筆交易記錄  
SELECT * FROM vw_recent_transactions LIMIT 10;
```



```
-- 8.2查看今日所有交易  
SELECT * FROM vw_recent_transactions  
WHERE DATE(Transaction_Date) = CURDATE();
```



```
-- 8.3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;
```



View

(管理員查看近期交易記錄與統計分析)

-- 8.4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;
```



(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	富邦台股	Sell	25	5950	661500.00
0051	元大中國100	Buy	3	2001	359679.00
0057	富邦台股	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	富邦台股	Buy	2	1200	113900.00
006208	富邦台股	Sell	1	500	51950.00
00679B	元大美國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	元大美國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	元大美國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)

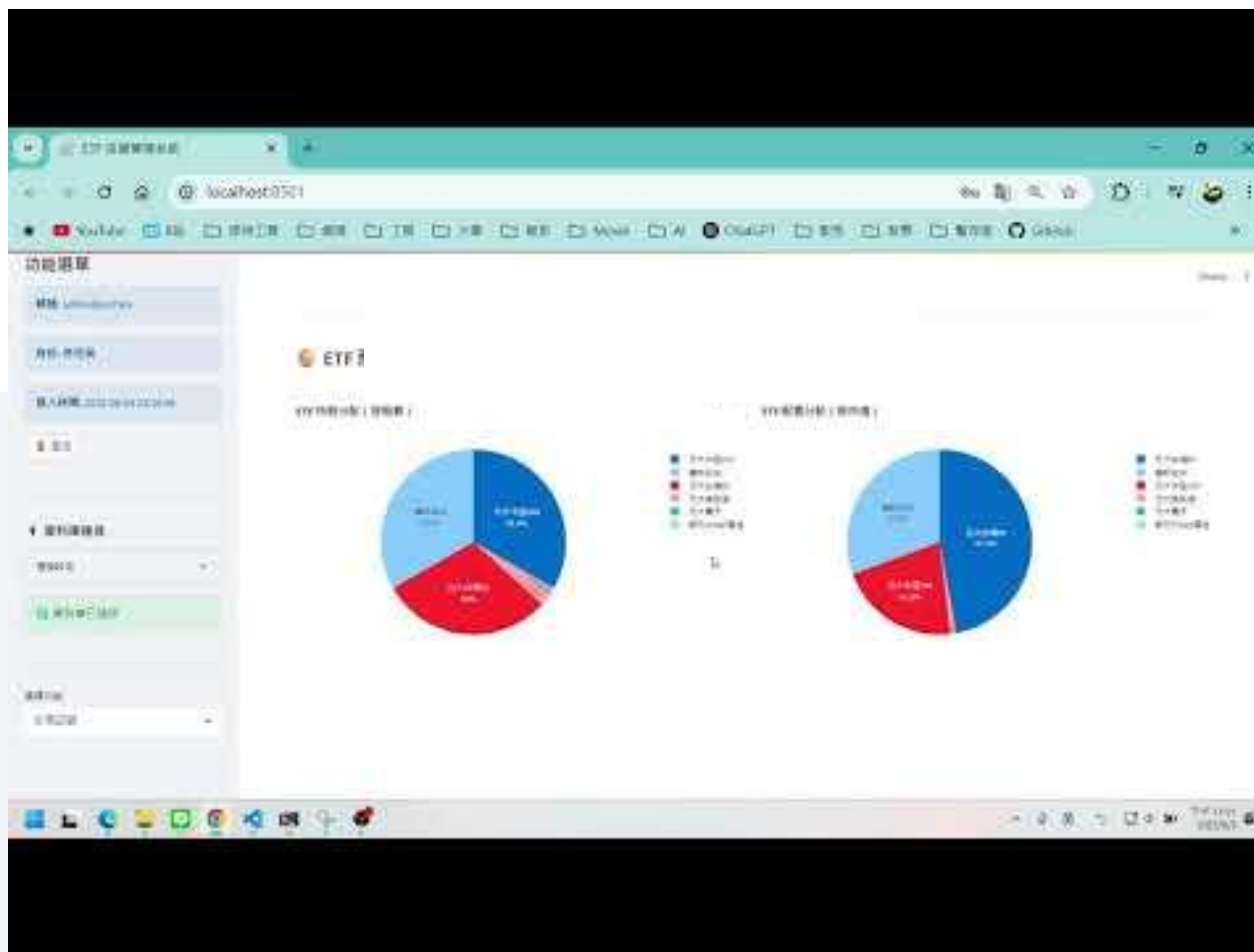
User_Id	Full_Name	Transaction_Count	Transaction_Amount
user001	王小明	52	1193548.00
adminbenchen	陳峻宇	22	1726169.50
user010	林怡君	2	472800.00
user003	陳柏宇	2	31650.00
user002	林小美	1	7360.00

系統資料筆數

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

```
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;
```

DEMO



工作分配

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

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)、第一分類表 (Category_Level1)、第二分類表 (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>

參考資料/處理方式

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

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、Plotly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(只包含部分ETF的資料):https://mopsov.twse.com.tw/mops/web/t78sb03_q2

心得

41143210 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。

心得

41143219 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。

心得

41143238 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。

心得

41143239 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。



THE END