ETF Holdings & Sector Allocation Database

1. Description

This database is designed to store and manage structured information about popular U.S. stock exchange-traded funds (stock ETFs), including their sector allocations, stock holdings, and market index memberships. The goal is to enable efficient querying of an ETF's sector distribution and individual stock holdings for financial analysis and portfolio management.

The database consists of 5 entities and their relationships. The entities include ETFs, Stocks, Sectors, Market Index, and Fund Families. Relationships are structured to represent how ETFs invest in stocks and sectors, and how ETFs and stocks relate to broader market indexes.

2. Entities and relationship
   a. **ETF**: stores basic information of ETF

Attributes	Description	constraint
etf_ticker	The unique ID of ETF	Primary Key
name	ETF names	Not null
1,3,5 annualized return	Capture the annualized return of the underlying ETF	
{category}	ETF types on investment strategy perspective	Have a own table
inception_date	Establish date	
aum	Asset under management	Check (aum >=0)
Management fee	Management fee of the underlying ETF	
Number of stock included	How many different stocks does this ETF hold	Check (number >=0)

   b. **Stock**: stores information about all individual stocks that may be in ETFs

Attributes	Description	constraint
stock_ticker	The unique ID of the stock	Primary key
name	The name of the company	Not null

Founded date	Which sector does the stock belong to	Foreign key
IPO date	The date of IPO	
Stock price	Current stock price (dynamic)	Not sure whether to inclued
Daily change	Daily price percentage change (dynamic)	
Shares outstanding	How many shares	

c. **Sector**: stores different industry categories

Attributes	Description	constraint
Sector_id	**The unique id of the sector**	**Primary key**
sector _name	The name of the sector	Unique not null
1, 3, 5 annualized return	**Capture the annualized return of the underlying sector**	
Number of stocks included	**How many different stocks does this sector hold**	Check (number >=0)

d. **Market_index:** The market index (S&P 500, Nasdaq 100)

Attributes	Description	constraint
index_ticker	The unique ID of the market index	Primary key
index_name	The name of the index	Unique not null
category	The kind of index	
Number_of_c ontributes	How many stocks included in the market index	
Launch_date	The date the index launched	
base value	It is the reference point when the	Not null

	index is created and is used to calculate the rise and fall of the index.	Check (base >=0)

e. **fund family**: the founders of ETF

Attributes	Description	constraint
Fundfamily_id	The unique ID of the fund family	Primary key
Fund family name	The name of the fund family	Not null
Headquarter	The headquarter location	
aum	Asset under management	Check (aum >= 0)
Number of funds	The number of funds that the fund family managed	

Relationship

relationship	Related entities	types	description
stocks_in_ETF	ETF↔Stock	many-to-many	Connecting ETF and component stocks, storing the stock weight
stocks_in_sector	stock↔ Sector	Many-to-one Every stock only belongs to one sector	Connecting stock and the sector, storing which sector the underlying stock belongs to.
Stock_in_index	Stock ↔ index	many-to-many	Connecting stock and index, what stocks the underlying index included
ETF_in_index	ETF ↔ Market_index	many-to-many	Connecting ETF to the market index
ETF_has_sector	ETF ↔ sector	many-to-many	Connecting ETF to sector
ETF_belongs_to_f unnfamily	ETF ↔ fund family	many-to-one	Connecting ETF to the founder company

3. **Data plan Summary**

   To populate the database, we will collect ETF composition data from publicly available sources such as ETF providers (e.g., Vanguard, iShares), financial data platforms (e.g., Yahoo Finance, Bloomberg), and regulatory filings (e.g., SEC EDGAR reports). This data will include ETF stock holdings, sector allocations, and market index memberships, ensuring accurate representation of ETF structures. Stock industry classifications will also be sourced from standard sector categorization systems GICS (Global Industry Classification Standard).

4. **Part 3 plans (UI design)**

   Users can use this as a basic ETF information searching system. People can search by ETF ticker, stock ticker, sector name, and market index ticker to get the information they want.

   Also, they can enter the ETF they hold or the ETF they are interested in. By recommendation algorithm (ETFs have the same investment category and the same largest weight sector), the system can recommend some ETF for the user.

   Searching system:

input	Page 1 Basic information	Page 2 Included stock	Page 3 compare
enter the ETF ticker	Return the basic information of the ETF (name, categories, Inception_date, Aum, Number of stock included, sector, management fee, market_index, fund family)	Return all the stock included in the ETF (Each stock is one row, including the stock basic information) by weight decending.	Compare the 1,2,3 year annualized return to the sector belonging to

Input	Page 1: basic information	Page 2 : The ETFs that contain this stock	Page 3 Return information
Enter the stock ticker	Stock name, sector, founded date, IPO date	All the ETFs that contain this stock (return their ticker and name)	Stock price (dynamic) Daily change (dynamic) Shares outstanding (dynamic) (do we need to include the history change？)

Input	Page1: basic information	Page 2: ETF related to this sector	Page 3: Stocks included in this sector
Enter the sector name	Sector name 1，3，5 annualized return The number of stocks include	The ETF relates to this sector and the weight of this sector in the ETF	All the stocks included in the sector (stock name and ticker

Input	Page 1: basic information	Page 2: ETF related to the market index	Page 3: stock related to the market index
Enter the market index ticker	Index_name Ticker category Number_of_contribut es Launch_date base value	The ETF relates to this index (ETF name and ticker)	All the stocks related to the market index(stock name and ticker