

AI 投資分析儀 (Gemini Quant) — 資料庫規格書

文件版本：v9.4 (黃金匯率增強版)

最後更新：2025 年 11 月 22 日

專案代號：Gemini Quant

部署環境：QNAP TS-h973AX (Supabase Self-Hosted via Docker)

1.0 系統願景與儲存架構

1.1 核心目標

建立一個本地優先 (Local-First) 的量化投資資料庫，整合台股、美股、黃金與外匯數據。利用 QNAP NAS 的高效能 NVMe SSD 進行回測運算，並將脫敏數據發送至 Google Gemini 進行決策輔助。

1.2 物理儲存分層策略 (Physical Storage Tiering)

本系統利用 PostgreSQL 的 TABLESPACE 功能，根據資料熱度分配硬體資源。

層級 (Tier)	表空間名稱	對應硬體路徑	物理裝置	適用資料類型
Tier 0 (Hot)	pg_default	/var/lib/postgresql/data	U.2 NVMe SSD (RAID 1)	核心交易表、索引、WAL Logs、近期股價
Tier 0 (Hot)	ts_hot_nvme	/mnt/nvme_data	U.2 NVMe SSD	歷史 K 線 (Partitioned)、量化因子、AI 報告

層級 (Tier)	表空間名稱	對應硬體路徑	物理裝置	適用資料類型
			(RAID 1)	
Tier 2 (Cold)	ts_cold_hd d	/mnt/hdd_archive	HDD (RAID 1)	原始新聞 HTML、系統日誌備份、舊財報 PDF

2.0 命名與設計規範

- 語言：美式英語 (en_US)。
- 格式：Snake Case (e.g., daily_prices, security_id)。
- 主鍵 (PK)：
 - 實體表使用 UUID (gen_random_uuid())。
 - 時序表使用複合主鍵 (security_id + date)。
- 時間格式：統一使用 TIMESTAMPTZ (UTC)，前端顯示時轉換時區。
- 貨幣單位：金額欄位使用 NUMERIC (高精度)，不使用 FLOAT。

3.0 邏輯資料模型 (Schema Design)

3.1 Master Data (證券主檔)

Table: securities_master

系統的核心清單，定義所有追蹤的資產。

- Storage:** pg_default (NVMe)

欄位名稱	資料型態	限制 (Constraints)	描述
id	UUID	PK, Default: gen_random_uuid()	唯一識別碼

欄位名稱	資料型態	限制 (Constraints)	描述
ticker	VARCHAR(20)	Unique, Not Null	代碼 (e.g., 2330.TW, AAPL, XAUUSD, USDTWD)
exchange	VARCHAR(20)	Check: TWSE, TPEX, NYSE, NASDAQ, FOREX, COMMODITY	交易所/資產類別
name	VARCHAR(100)	Not Null	資產名稱
sector	VARCHAR(50)		產業板塊 (e.g., Semiconductor , Precious Metal)
currency	CHAR(3)	Default: USD	計價幣別 (TWD, USD)
is_active	BOOLEAN	Default: true	是否監控中
created_at	TIMESTAMPTZ	Default: now()	建立時間

3.2 Market Data (市場數據)

Table: daily_prices

每日 OHLCV 數據。此表涵蓋股票、黃金與匯率。

- **Storage:** ts_hot_nvme (NVMe)
- **Partitioning:** 按年分區 (Range Partition by date)。

欄位名稱	資料型態	描述	備註
security_id	UUID	FK -> securities_master(id)	複合主鍵 Part 1
date	DATE	交易日期	複合主鍵 Part 2
open	NUMERIC(12,4)	開盤價	匯率/黃金亦同
high	NUMERIC(12,4)	最高價	
low	NUMERIC(12,4)	最低價	
close	NUMERIC(12,4)	收盤價	
adj_close	NUMERIC(12,4)	還原收盤價	僅股票有效，其餘同 close
volume	BIGINT	成交量	匯率通常為 0
source	VARCHAR(20)	資料來源標記	Tiingo, Fugle, Yahoo

Table: shareholder_dispersion (台股專用)

集保中心大戶籌碼數據。

- **Storage:** ts_hot_nvme (NVMe)

欄位名稱	資料型態	描述
security_id	UUID	FK -> securities_master
date	DATE	資料日期 (每週五)
level_15_plus_shares	BIGINT	千張以上大戶持股數

欄位名稱	資料型態	描述
total_shares	BIGINT	總發行股數
large_holders_pct	NUMERIC(5,2)	大戶持股比例 (%)

3.3 Quant & AI (量化與智慧層)

Table: quant_scores

六大因子模型計算結果。

- **Storage:** ts_hot_nvme (NVMe)

欄位名稱	資料型態	描述
security_id	UUID	FK
date	DATE	計算日期
total_score	NUMERIC(5,2)	0-100 總分
value_score	NUMERIC(5,2)	價值因子 (P/E, P/B...)
momentum_score	NUMERIC(5,2)	動能因子 (RSI, Rel Str...)
quality_score	NUMERIC(5,2)	品質因子 (ROE, Margin...)
volatility_score	NUMERIC(5,2)	波動因子 (StdDev...)

Table: ai_reports

Gemini 生成的分析報告存檔。

- **Storage:** ts_hot_nvme (NVMe)

欄位名稱	資料型態	描述
id	UUID	PK
security_id	UUID	FK (Null 若為大盤報告)

欄位名稱	資料型態	描述
report_type	VARCHAR(20)	DAILY_STRATEGY, STOCK_ANALYSIS
content	TEXT	Markdown 格式報告內容
sentiment	VARCHAR(20)	Bullish, Bearish, Neutral
created_at	TIMESTAMPTZ	生成時間

3.4 User & Portfolio (用戶資產層)

Table: user_portfolios

用戶當前持倉快照。必須啟用 RLS (Row Level Security)。

- **Storage:** pg_default (NVMe)

欄位名稱	資料型態	描述
id	UUID	PK
user_id	UUID	FK -> auth.users (Supabase Auth)
security_id	UUID	FK -> securities_master
quantity	NUMERIC(12,4)	持有數量 (股數/盎司/美金)
average_cost	NUMERIC(12,4)	平均成本

Table: transaction_logs

交易流水帳，包含換匯紀錄。

- **Storage:** pg_default (NVMe)

欄位名稱	資料型態	描述
id	UUID	PK
user_id	UUID	FK -> auth.users

欄位名稱	資料型態	描述
security_id	UUID	FK -> securities_master
action	VARCHAR(10)	BUY, SELL, DIVIDEND, EXCHANGE
price	NUMERIC(12,4)	成交價格/匯率
quantity	NUMERIC(12,4)	數量
fee	NUMERIC(10,2)	手續費
date	TIMESTAMPTZ	交易時間
note	TEXT	交易筆記/策略標籤

4.0 數據源策略 (Data Sources - v9.4)

4.1 數據源配置表

資產類別	主要來源 (Primary)	備援來源 (Secondary)	更新頻率
台股 (TW Stocks)	Fugle (富果)	Yahoo Finance, TWSE	即時/盤後
美股 (US Stocks)	Tiingo	Finnhub	收盤後 (EOD)
黃金 (Gold/XAU)	Tiingo (XAUUSD)	Yahoo Finance (GC=F)	每日
匯率 (Forex)	Tiingo (USDTWD)	Yahoo Finance (TWD=X)	每日
大戶籌碼	TDCC (集保)	N/A	每週六
宏觀數據	FRED API	World Bank	每月/每季

4.2 N8N 自動化任務規劃

1. **Daily Market Sync**: 每日 06:00 AM (UTC+8) 抓取 Tiingo (美股/黃金/匯率) 寫入 daily_prices。
 2. **TW Market Sync**: 每日 14:30 PM (UTC+8) 抓取 Fugle/TWSE 寫入 daily_prices。
 3. **Chips Analysis**: 每週六 10:00 AM 抓取 TDCC 數據寫入 shareholder_dispersion。
 4. **AI Reporting**: 當 daily_prices 更新後，觸發 Webhook 呼叫 Gemini 生成報告寫入 ai_reports。
-

5.0 SQL 建置腳本範例 (Quick Start)

以下 SQL 可直接在 Supabase SQL Editor 中執行，以建立核心表格與權限。

SQL

-- 1. 啟用必要的擴充套件

```
CREATE EXTENSION IF NOT EXISTS "uuid-ossp";
```

-- 2. 建立證券主檔 (Securities Master)

```
CREATE TABLE public.securities_master (
    id UUID DEFAULT gen_random_uuid() PRIMARY KEY,
    ticker VARCHAR(20) NOT NULL UNIQUE,
    name VARCHAR(100) NOT NULL,
    exchange VARCHAR(20) CHECK (exchange IN ('TWSE', 'TPEX',
    'NYSE', 'NASDAQ', 'FOREX', 'COMMODITY', 'MACRO')),
    sector VARCHAR(50),
    currency CHAR(3) DEFAULT 'USD',
    is_active BOOLEAN DEFAULT true,
    created_at TIMESTAMPTZ DEFAULT now(),
    updated_at TIMESTAMPTZ DEFAULT now()
); -- 如有設定 TableSpace 可在此加入 TABLESPACE pg_default;
```

-- 建立索引

```
CREATE INDEX idx_securities_ticker ON
```

```
public.securities_master(ticker);
```

```
CREATE INDEX idx_securities_exchange ON
```

```
public.securities_master(exchange);
```

-- 3. 建立每日價格表 (Daily Prices) - 使用 Partitioning
CREATE TABLE public.daily_prices (

```
    security_id UUID NOT NULL REFERENCES
public.securities_master(id),
    date DATE NOT NULL,
    open NUMERIC(12,4),
    high NUMERIC(12,4),
    low NUMERIC(12,4),
    close NUMERIC(12,4),
    adj_close NUMERIC(12,4),
    volume BIGINT,
    source VARCHAR(20),
    PRIMARY KEY (security_id, date)
) PARTITION BY RANGE (date);
```

-- 建立 2024-2026 年的分區表範例

```
CREATE TABLE daily_prices_2024 PARTITION OF daily_prices FOR
VALUES FROM ('2024-01-01') TO ('2025-01-01');
CREATE TABLE daily_prices_2025 PARTITION OF daily_prices FOR
VALUES FROM ('2025-01-01') TO ('2026-01-01');
CREATE TABLE daily_prices_2026 PARTITION OF daily_prices FOR
VALUES FROM ('2026-01-01') TO ('2027-01-01');
```

-- 4. 啟用 RLS 安全性 (以 transaction_logs 為例)

```
CREATE TABLE public.transaction_logs (
    id UUID DEFAULT gen_random_uuid() PRIMARY KEY,
    user_id UUID REFERENCES auth.users NOT NULL,
    security_id UUID REFERENCES public.securities_master(id),
    action VARCHAR(10) NOT NULL,
    price NUMERIC(12,4) NOT NULL,
    quantity NUMERIC(12,4) NOT NULL,
    fee NUMERIC(10,2) DEFAULT 0,
    date TIMESTAMPTZ DEFAULT now()
);
```

```
ALTER TABLE public.transaction_logs ENABLE ROW LEVEL
SECURITY;
```

```
-- 設定 RLS 政策：用戶只能看到自己的交易紀錄
CREATE POLICY "Users can only access own logs"
ON public.transaction_logs
FOR ALL
USING (auth.uid() = user_id);
```

6.0 附錄：初始化資料種子 (Seeds)

建議預先插入以下黃金與匯率的基本定義：

SQL

```
INSERT INTO public.securities_master (ticker, name, exchange, sector,
currency) VALUES
('XAUUSD', 'Spot Gold (現貨黃金)', 'COMMODITY', 'Precious Metal',
'USD'),
('USDTWD', 'USD/TWD Exchange Rate', 'FOREX', 'Currency', 'TWD'),
('2330.TW', 'TSMC (台積電)', 'TWSE', 'Semiconductor', 'TWD'),
('NVDA', 'NVIDIA Corp', 'NASDAQ', 'Semiconductor', 'USD');
```