Cryptocurrency_MultiCoinTracker - Documentation

https://github.com/Js24zz/Cryptocurrency_MultiCoinTracker.git

---

Introduction

This project implements an automated Cryptocurrency Multi-Coin Tracker that periodically collects live market data for 10 major cryptocurrencies, stores historical snapshots in a normalized MySQL database, and automatically produces PNG graphs using GNUPlot. The solution is implemented primarily in Bash, with curl + jq for extraction/cleaning, MySQL for storage, GNUPlot for visualization, and cron for automation.

- Unix Script for Data Collection (curl + jq + MySQL insert).
- Unix Script for Plotting (GNUPlot called from .sh functions).
- Use of Git for Version Control (commit history evidence).
- MySQL for Data Storage (normalized schema, ERD, and dump file).

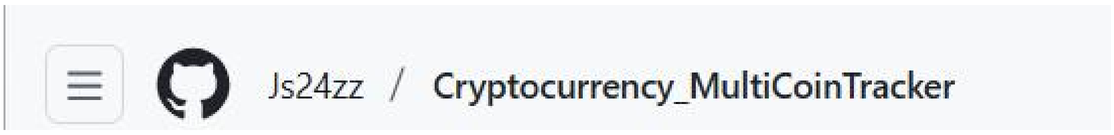
Environment & Tools (WSL-first workflow)

1) Windows + WSL (Ubuntu) + VS Code (Remote – WSL) for a stable Linux environment.

2) MySQL Server runs inside WSL; command-line MySQL is used in scripts.

3) Core dependencies: curl, jq, mysql-client, gnuplot, cron.

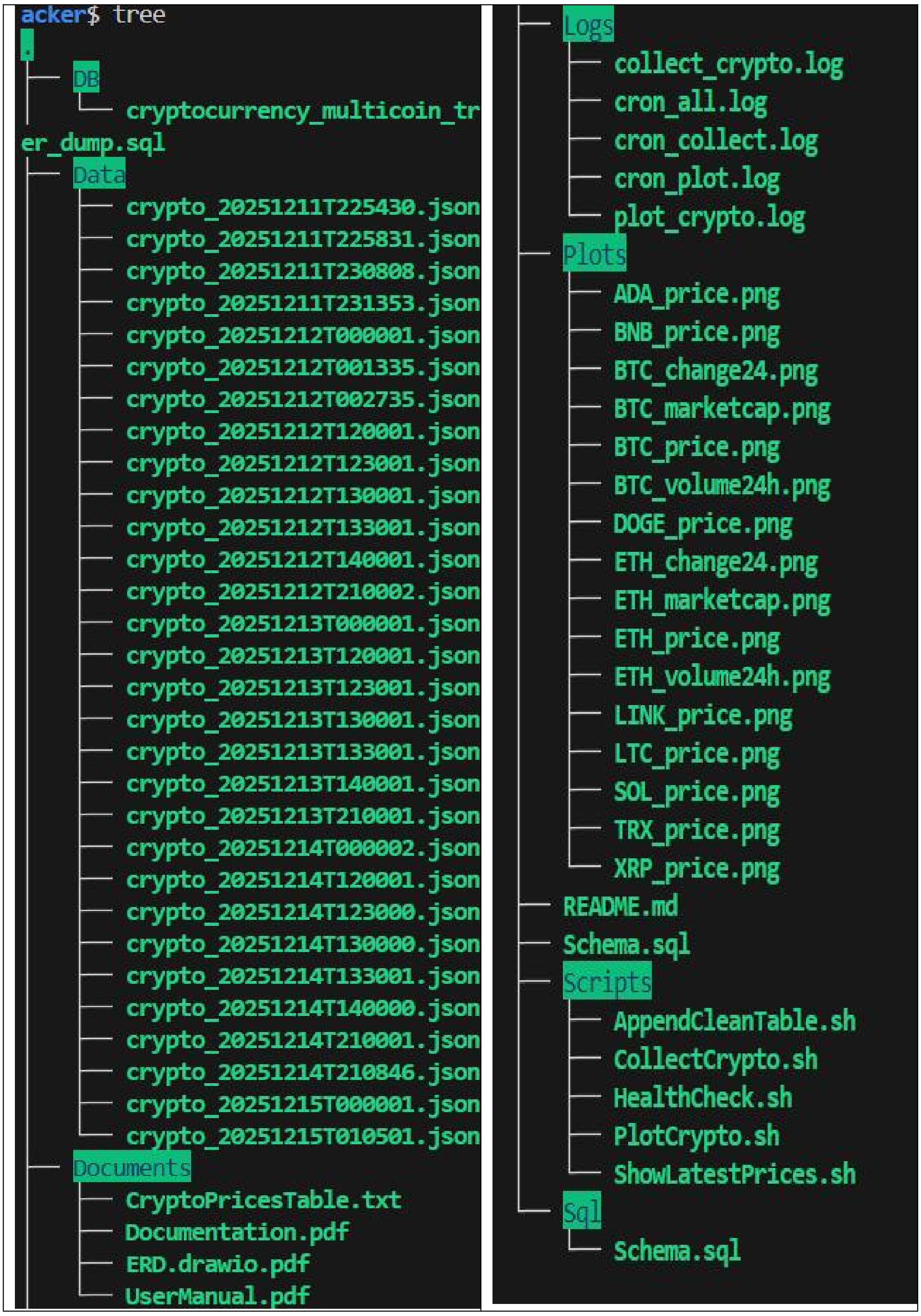
Creating A GitHub Repository


Js24zz / Cryptocurrency_MultiCoinTracker

Successfully created repository
https://github.com/Js24zz/Cryptocurrency_MultiCoinTracker.git

(Welcome to my GitHub)

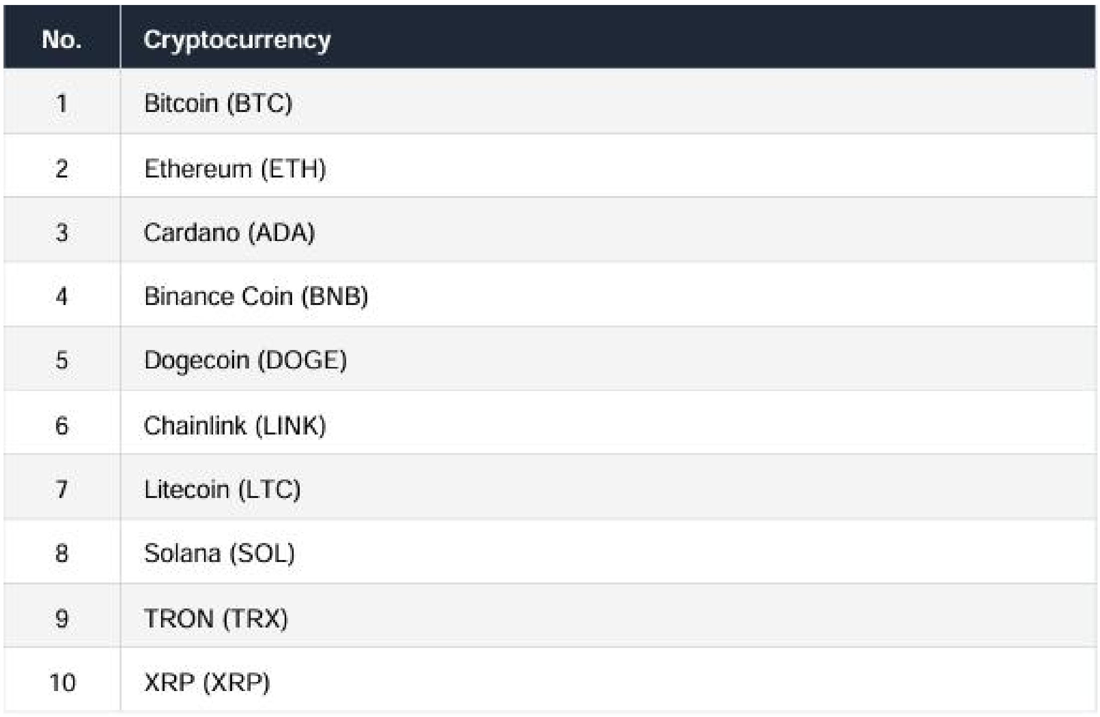
Project Structure (VSC)



```
acker$ tree
.
├── DB
│   └── cryptocurrency_multicoin_tr
er_dump.sql
├── Data
│   ├── crypto_20251211T225430.json
│   ├── crypto_20251211T225831.json
│   ├── crypto_20251211T230808.json
│   ├── crypto_20251211T231353.json
│   ├── crypto_20251212T000001.json
│   ├── crypto_20251212T001335.json
│   ├── crypto_20251212T002735.json
│   ├── crypto_20251212T120001.json
│   ├── crypto_20251212T123001.json
│   ├── crypto_20251212T130001.json
│   ├── crypto_20251212T133001.json
│   ├── crypto_20251212T140001.json
│   ├── crypto_20251212T210002.json
│   ├── crypto_20251213T000001.json
│   ├── crypto_20251213T120001.json
│   ├── crypto_20251213T123001.json
│   ├── crypto_20251213T130001.json
│   ├── crypto_20251213T133001.json
│   ├── crypto_20251213T140001.json
│   ├── crypto_20251213T210001.json
│   ├── crypto_20251214T000002.json
│   ├── crypto_20251214T120001.json
│   ├── crypto_20251214T123000.json
│   ├── crypto_20251214T130000.json
│   ├── crypto_20251214T133001.json
│   ├── crypto_20251214T140000.json
│   ├── crypto_20251214T210001.json
│   ├── crypto_20251214T210846.json
│   ├── crypto_20251215T000001.json
│   └── crypto_20251215T010501.json
├── Documents
│   ├── CryptoPricesTable.txt
│   ├── Documentation.pdf
│   ├── ERD.drawio.pdf
│   └── UserManual.pdf
```

```
├── Logs
│   ├── collect_crypto.log
│   ├── cron_all.log
│   ├── cron_collect.log
│   ├── cron_plot.log
│   └── plot_crypto.log
├── Plots
│   ├── ADA_price.png
│   ├── BNB_price.png
│   ├── BTC_change24.png
│   ├── BTC_marketcap.png
│   ├── BTC_price.png
│   ├── BTC_volume24h.png
│   ├── DOGE_price.png
│   ├── ETH_change24.png
│   ├── ETH_marketcap.png
│   ├── ETH_price.png
│   ├── ETH_volume24h.png
│   ├── LINK_price.png
│   ├── LTC_price.png
│   ├── SOL_price.png
│   ├── TRX_price.png
│   └── XRP_price.png
├── README.md
├── Schema.sql
├── Scripts
│   ├── AppendCleanTable.sh
│   ├── CollectCrypto.sh
│   ├── HealthCheck.sh
│   ├── PlotCrypto.sh
│   └── ShowLatestPrices.sh
└── Sql
    └── Schema.sql
```

Tracked Cryptocurrencies



No.	Cryptocurrency
1	Bitcoin (BTC)
2	Ethereum (ETH)
3	Cardano (ADA)
4	Binance Coin (BNB)
5	Dogecoin (DOGE)
6	Chainlink (LINK)
7	Litecoin (LTC)
8	Solana (SOL)
9	TRON (TRX)
10	XRP (XRP)

Section 1 (Unix Script for Data Collection)

Objective: collect data regularly, perform parsing and manipulation, handle errors, and insert cleaned data into MySQL.

1.1) Data Source Choice (Website/API)

Chosen source: CoinGecko public API. It provides price, market cap, 24h volume, 24h high/low, and 24h % change in one response, which reduces scraping risk and increases reliability compared to HTML scraping.

Endpoint pattern:

https://api.coingecko.com/api/v3/coins/markets?vs_currency=usd&ids=<coin_ids>&order=market_cap_desc&per_page=250&page=1&sparkline=false&price_change_percentage=24h

1.2) JSON Parsing with jq (example)

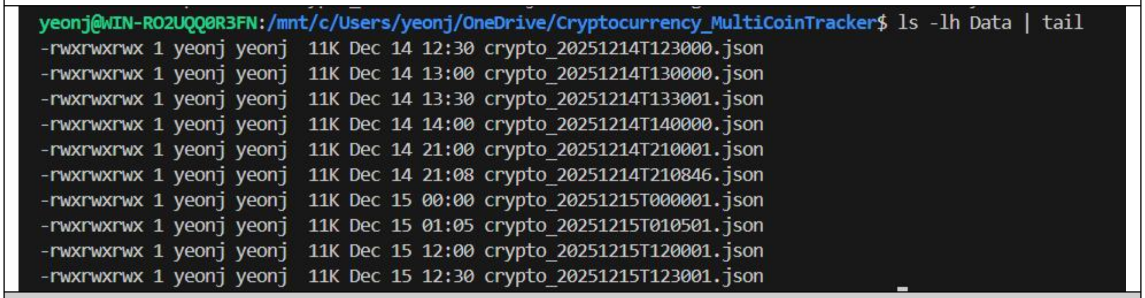
Typical jq extraction pattern: jq -r '.[] |

[.id, .symbol, .current_price, .market_cap, .total_volume, .low_24h, .high_24h, .price_change_percentage_24h] | @tsv'

Evidences

Raw API response exists (JSON file)

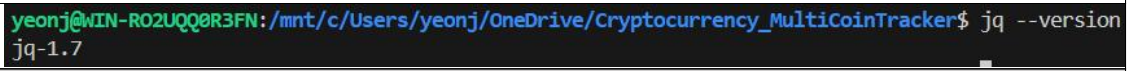
ls -lh Data | tail



```
yeonj@WIN-RO2UQQ0R3FN:/mnt/c/Users/yeonj/OneDrive/Cryptocurrency_MultiCoinTracker$ ls -lh Data | tail
-rwxrwxrwx 1 yeonj yeonj  11K Dec 14 12:30 crypto_20251214T123000.json
-rwxrwxrwx 1 yeonj yeonj  11K Dec 14 13:00 crypto_20251214T130000.json
-rwxrwxrwx 1 yeonj yeonj  11K Dec 14 13:30 crypto_20251214T133001.json
-rwxrwxrwx 1 yeonj yeonj  11K Dec 14 14:00 crypto_20251214T140000.json
-rwxrwxrwx 1 yeonj yeonj  11K Dec 14 21:00 crypto_20251214T210001.json
-rwxrwxrwx 1 yeonj yeonj  11K Dec 14 21:08 crypto_20251214T210846.json
-rwxrwxrwx 1 yeonj yeonj  11K Dec 15 00:00 crypto_20251215T000001.json
-rwxrwxrwx 1 yeonj yeonj  11K Dec 15 01:05 crypto_20251215T010501.json
-rwxrwxrwx 1 yeonj yeonj  11K Dec 15 12:00 crypto_20251215T120001.json
-rwxrwxrwx 1 yeonj yeonj  11K Dec 15 12:30 crypto_20251215T123001.json
```

jq is installed

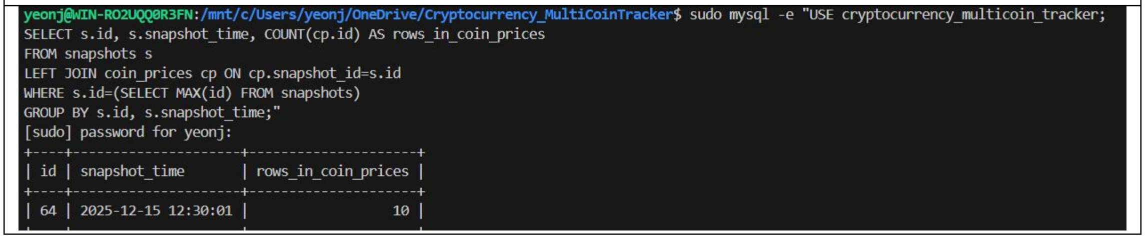
jq --version



```
yeonj@WIN-RO2UQQ0R3FN:/mnt/c/Users/yeonj/OneDrive/Cryptocurrency_MultiCoinTracker$ jq --version
jq-1.7
```

jq data went into MySQL

sudo mysql -e "USE cryptocurrency_multicoin_tracker;
SELECT s.id, s.snapshot_time, COUNT(cp.id) AS rows_in_coin_prices
FROM snapshots s
LEFT JOIN coin_prices cp ON cp.snapshot_id=s.id
WHERE s.id=(SELECT MAX(id) FROM snapshots)
GROUP BY s.id, s.snapshot_time;"



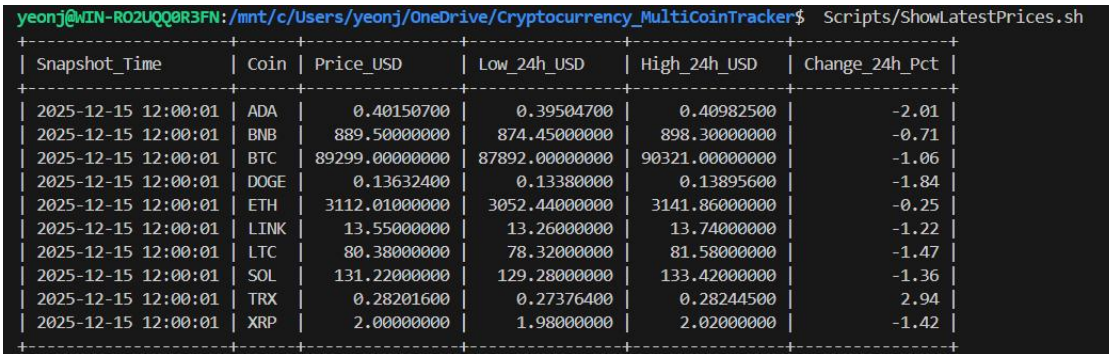
```
yeonj@WIN-RO2UQQ0R3FN:/mnt/c/Users/yeonj/OneDrive/Cryptocurrency_MultiCoinTracker$ sudo mysql -e "USE cryptocurrency_multicoin_tracker;
SELECT s.id, s.snapshot_time, COUNT(cp.id) AS rows_in_coin_prices
FROM snapshots s
LEFT JOIN coin_prices cp ON cp.snapshot_id=s.id
WHERE s.id=(SELECT MAX(id) FROM snapshots)
GROUP BY s.id, s.snapshot_time;"
[sudo] password for yeonj:
+----+---------------------+--------------------+
| id | snapshot_time       | rows_in_coin_prices |
+----+---------------------+--------------------+
| 64 | 2025-12-15 12:30:01 |                 10 |
```

Here, we use jq instead of manual string parsing to guarantee reliable extraction from structured JSON and ensure consistent column mapping for MySQL inserts.

1.3) ShowLatestPrices.sh (manual check)

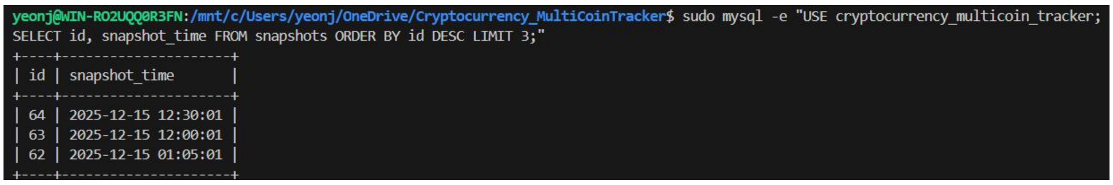
Quick verification: prints the latest snapshot_time and latest values per coin (price, low/high 24h, change 24h) as a formatted table in the terminal.

a) Bash **Scripts/ShowLatestPrices.sh**



```
yeonj@WIN-RO2UQQ0R3FN:/mnt/c/Users/yeonj/OneDrive/Cryptocurrency_MultiCoinTracker$  Scripts/ShowLatestPrices.sh
+---------------------+------+----------------+----------------+----------------+----------------+
| Snapshot_Time       | Coin | Price_USD      | Low_24h_USD    | High_24h_USD   | Change_24h_Pct |
+---------------------+------+----------------+----------------+----------------+----------------+
| 2025-12-15 12:00:01 | ADA  |     0.40150700 |     0.39504700 |     0.40982500 |          -2.01 |
| 2025-12-15 12:00:01 | BNB  |   889.50000000 |   874.45000000 |   898.30000000 |          -0.71 |
| 2025-12-15 12:00:01 | BTC  | 89299.00000000 | 87892.00000000 | 90321.00000000 |          -1.06 |
| 2025-12-15 12:00:01 | DOGE |     0.13632400 |     0.13380000 |     0.13895600 |          -1.84 |
| 2025-12-15 12:00:01 | ETH  |  3112.01000000 |  3052.44000000 |  3141.86000000 |          -0.25 |
| 2025-12-15 12:00:01 | LINK |    13.55000000 |    13.26000000 |    13.74000000 |          -1.22 |
| 2025-12-15 12:00:01 | LTC  |    80.38000000 |    78.32000000 |    81.58000000 |          -1.47 |
| 2025-12-15 12:00:01 | SOL  |   131.22000000 |   129.28000000 |   133.42000000 |          -1.36 |
| 2025-12-15 12:00:01 | TRX  |     0.28201600 |     0.27376400 |     0.28244500 |           2.94 |
| 2025-12-15 12:00:01 | XRP  |     2.00000000 |     1.98000000 |     2.02000000 |          -1.42 |
+---------------------+------+----------------+----------------+----------------+----------------+
```

b) Bash **sudo mysql -e "USE cryptocurrency_multicoin_tracker; SELECT COUNT(*) FROM coin_prices WHERE snapshot_id=(SELECT MAX(id) FROM snapshots);"**



```
yeonj@WIN-RO2UQQ0R3FN:/mnt/c/Users/yeonj/OneDrive/Cryptocurrency_MultiCoinTracker$ sudo mysql -e "USE cryptocurrency_multicoin_tracker;
SELECT id, snapshot_time FROM snapshots ORDER BY id DESC LIMIT 3;"
+----+---------------------+
| id | snapshot_time       |
+----+---------------------+
| 64 | 2025-12-15 12:30:01 |
| 63 | 2025-12-15 12:00:01 |
| 62 | 2025-12-15 01:05:01 |
+----+---------------------+
```

ShowLatestPrices.sh is a manual quality-assurance script used after data collection. It queries the most recent snapshot (latest run) and joins snapshots, coin_prices, and coins to display one row per tracked cryptocurrency.

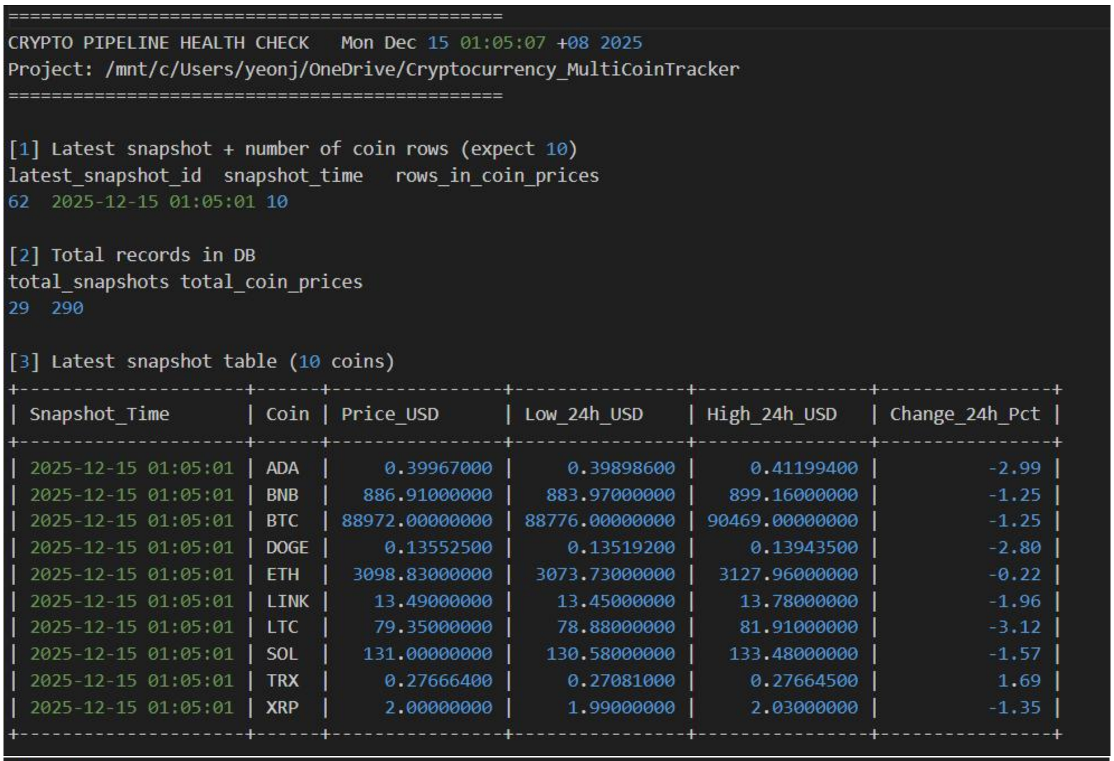
This provides a fast terminal-based validation that the ingestion pipeline succeeded (a new snapshot exists, rows were inserted for all coins, and key metrics such as price, 24h low/high, and 24h % change are present).

It also supports debugging by immediately revealing missing rows, unchanged timestamps, or null metrics without needing a GUI tool.

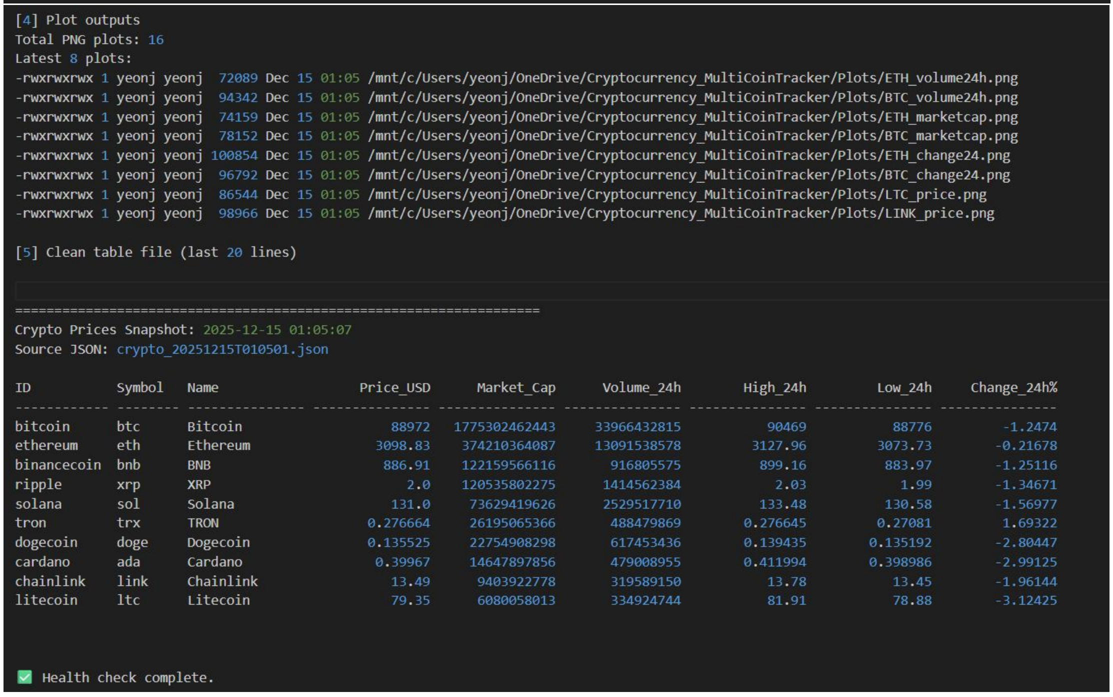
1.4) HealthCheck.sh (manually bash)

Diagnostic report of DB + files + plots Console output

a) In wsl terminal bash Scripts/HealthCheck.sh.



```
==============================================
CRYPTO PIPELINE HEALTH CHECK   Mon Dec 15 01:05:07 +08 2025
Project: /mnt/c/Users/yeonj/OneDrive/Cryptocurrency_MultiCoinTracker
==============================================

[1] Latest snapshot + number of coin rows (expect 10)
latest_snapshot_id  snapshot_time   rows_in_coin_prices
62  2025-12-15 01:05:01 10

[2] Total records in DB
total_snapshots total_coin_prices
29  290

[3] Latest snapshot table (10 coins)
+---------------------+------+----------------+----------------+----------------+----------------+
| Snapshot_Time       | Coin | Price_USD      | Low_24h_USD    | High_24h_USD   | Change_24h_Pct |
+---------------------+------+----------------+----------------+----------------+----------------+
| 2025-12-15 01:05:01 | ADA  |     0.39967000 |     0.39898600 |     0.41199400 |          -2.99 |
| 2025-12-15 01:05:01 | BNB  |   886.91000000 |   883.97000000 |   899.16000000 |          -1.25 |
| 2025-12-15 01:05:01 | BTC  | 88972.00000000 | 88776.00000000 | 90469.00000000 |          -1.25 |
| 2025-12-15 01:05:01 | DOGE |     0.13552500 |     0.13519200 |     0.13943500 |          -2.80 |
| 2025-12-15 01:05:01 | ETH  |  3098.83000000 |  3073.73000000 |  3127.96000000 |          -0.22 |
| 2025-12-15 01:05:01 | LINK |    13.49000000 |    13.45000000 |    13.78000000 |          -1.96 |
| 2025-12-15 01:05:01 | LTC  |    79.35000000 |    78.88000000 |    81.91000000 |          -3.12 |
| 2025-12-15 01:05:01 | SOL  |   131.00000000 |   130.58000000 |   133.48000000 |          -1.57 |
| 2025-12-15 01:05:01 | TRX  |     0.27666400 |     0.27081000 |     0.27664500 |           1.69 |
| 2025-12-15 01:05:01 | XRP  |     2.00000000 |     1.99000000 |     2.03000000 |          -1.35 |
+---------------------+------+----------------+----------------+----------------+----------------+
```



```
[4] Plot outputs
Total PNG plots: 16
Latest 8 plots:
-rwxrwxrwx 1 yeonj yeonj  72089 Dec 15 01:05 /mnt/c/Users/yeonj/OneDrive/Cryptocurrency_MultiCoinTracker/Plots/ETH_volume24h.png
-rwxrwxrwx 1 yeonj yeonj  94342 Dec 15 01:05 /mnt/c/Users/yeonj/OneDrive/Cryptocurrency_MultiCoinTracker/Plots/BTC_volume24h.png
-rwxrwxrwx 1 yeonj yeonj  74159 Dec 15 01:05 /mnt/c/Users/yeonj/OneDrive/Cryptocurrency_MultiCoinTracker/Plots/ETH_marketcap.png
-rwxrwxrwx 1 yeonj yeonj  78152 Dec 15 01:05 /mnt/c/Users/yeonj/OneDrive/Cryptocurrency_MultiCoinTracker/Plots/BTC_marketcap.png
-rwxrwxrwx 1 yeonj yeonj 100854 Dec 15 01:05 /mnt/c/Users/yeonj/OneDrive/Cryptocurrency_MultiCoinTracker/Plots/ETH_change24.png
-rwxrwxrwx 1 yeonj yeonj  96792 Dec 15 01:05 /mnt/c/Users/yeonj/OneDrive/Cryptocurrency_MultiCoinTracker/Plots/BTC_change24.png
-rwxrwxrwx 1 yeonj yeonj  86544 Dec 15 01:05 /mnt/c/Users/yeonj/OneDrive/Cryptocurrency_MultiCoinTracker/Plots/LTC_price.png
-rwxrwxrwx 1 yeonj yeonj  98966 Dec 15 01:05 /mnt/c/Users/yeonj/OneDrive/Cryptocurrency_MultiCoinTracker/Plots/LINK_price.png

[5] Clean table file (last 20 lines)


===================================================================
Crypto Prices Snapshot: 2025-12-15 01:05:07
Source JSON: crypto_20251215T010501.json

ID           Symbol   Name                 Price_USD     Market_Cap     Volume_24h       High_24h        Low_24h   Change_24h%
------------ -------- --------------- -------------- -------------- -------------- -------------- -------------- --------------
bitcoin      btc      Bitcoin                  88972  1775302462443    33966432815          90469          88776        -1.2474
ethereum     eth      Ethereum               3098.83   374210364087    13091538578        3127.96        3073.73       -0.21678
binancecoin  bnb      BNB                     886.91   122159566116      916805575         899.16         883.97       -1.25116
ripple       xrp      XRP                        2.0   120535802275     1414562384           2.03           1.99       -1.34671
solana       sol      Solana                   131.0    73629419626     2529517710         133.48         130.58       -1.56977
tron         trx      TRON                  0.276664    26195065366      488479869       0.276645        0.27081        1.69322
dogecoin     doge     Dogecoin              0.135525    22754908298      617453436       0.139435       0.135192       -2.80447
cardano      ada      Cardano                0.39967    14647897856      479008955       0.411994       0.398986       -2.99125
chainlink    link     Chainlink                13.49     9403922778      319589150          13.78          13.45       -1.96144
litecoin     ltc      Litecoin                 79.35     6080058013      334924744          81.91          78.88       -3.12425


✅ Health check complete.
```

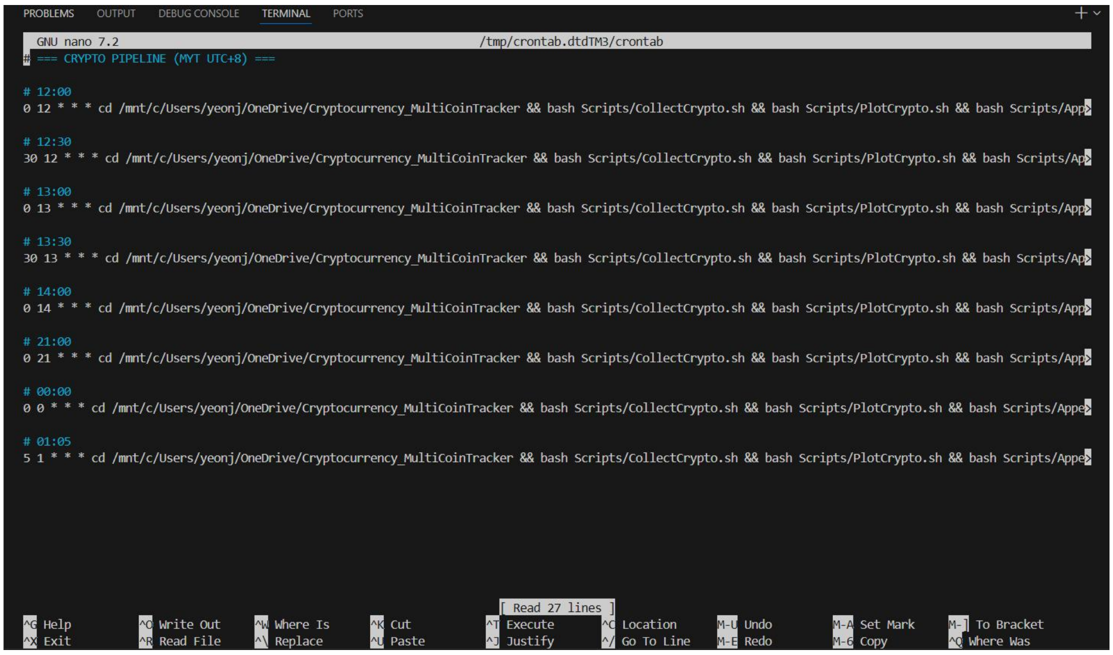
The health check status record will be stored in the cron_all.log file.

1.5) Automation (cron on MYT, UTC+8)

Chosen schedule: 12:00, 12:30, 13:00, 13:30, 14:00, 21:00, 00:00, 01:05 (MYT).

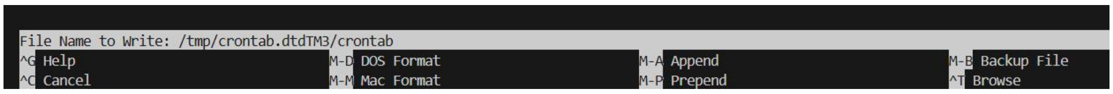
Reason: aligns with common lunch/rest window (12–14), end-of-day or after work or overtime-working (21:00), and a fresh new-day snapshot (00:00)(01:05).

a) Bash sudo crontab -e.
b) Insert the command scheduling in.
c) Press Ctrl O.



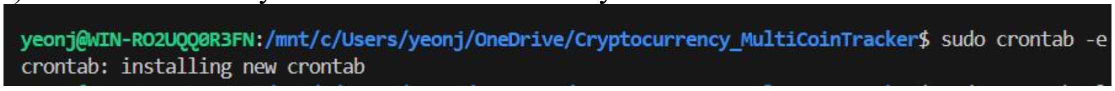
```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

 GNU nano 7.2                                   /tmp/crontab.dtdTM3/crontab
# === CRYPTO PIPELINE (MYT UTC+8) ===

# 12:00
0 12 * * * cd /mnt/c/Users/yeonj/OneDrive/Cryptocurrency_MultiCoinTracker && bash Scripts/CollectCrypto.sh && bash Scripts/PlotCrypto.sh && bash Scripts/App>

# 12:30
30 12 * * * cd /mnt/c/Users/yeonj/OneDrive/Cryptocurrency_MultiCoinTracker && bash Scripts/CollectCrypto.sh && bash Scripts/PlotCrypto.sh && bash Scripts/Ap>

# 13:00
0 13 * * * cd /mnt/c/Users/yeonj/OneDrive/Cryptocurrency_MultiCoinTracker && bash Scripts/CollectCrypto.sh && bash Scripts/PlotCrypto.sh && bash Scripts/App>

# 13:30
30 13 * * * cd /mnt/c/Users/yeonj/OneDrive/Cryptocurrency_MultiCoinTracker && bash Scripts/CollectCrypto.sh && bash Scripts/PlotCrypto.sh && bash Scripts/Ap>

# 14:00
0 14 * * * cd /mnt/c/Users/yeonj/OneDrive/Cryptocurrency_MultiCoinTracker && bash Scripts/CollectCrypto.sh && bash Scripts/PlotCrypto.sh && bash Scripts/App>

# 21:00
0 21 * * * cd /mnt/c/Users/yeonj/OneDrive/Cryptocurrency_MultiCoinTracker && bash Scripts/CollectCrypto.sh && bash Scripts/PlotCrypto.sh && bash Scripts/App>

# 00:00
0 0 * * * cd /mnt/c/Users/yeonj/OneDrive/Cryptocurrency_MultiCoinTracker && bash Scripts/CollectCrypto.sh && bash Scripts/PlotCrypto.sh && bash Scripts/Appe>

# 01:05
5 1 * * * cd /mnt/c/Users/yeonj/OneDrive/Cryptocurrency_MultiCoinTracker && bash Scripts/CollectCrypto.sh && bash Scripts/PlotCrypto.sh && bash Scripts/Appe>

                                              [ Read 27 lines ]
^G Help         ^O Write Out    ^W Where Is     ^K Cut          ^T Execute      ^C Location     M-U Undo        M-A Set Mark    M-] To Bracket
^X Exit         ^R Read File    ^\ Replace      ^U Paste        ^J Justify      ^/ Go To Line   M-E Redo        M-6 Copy        ^Q Where Was
```

d) You will see a file name below.
e) Click Enter(to save) and Ctrl X to exit.



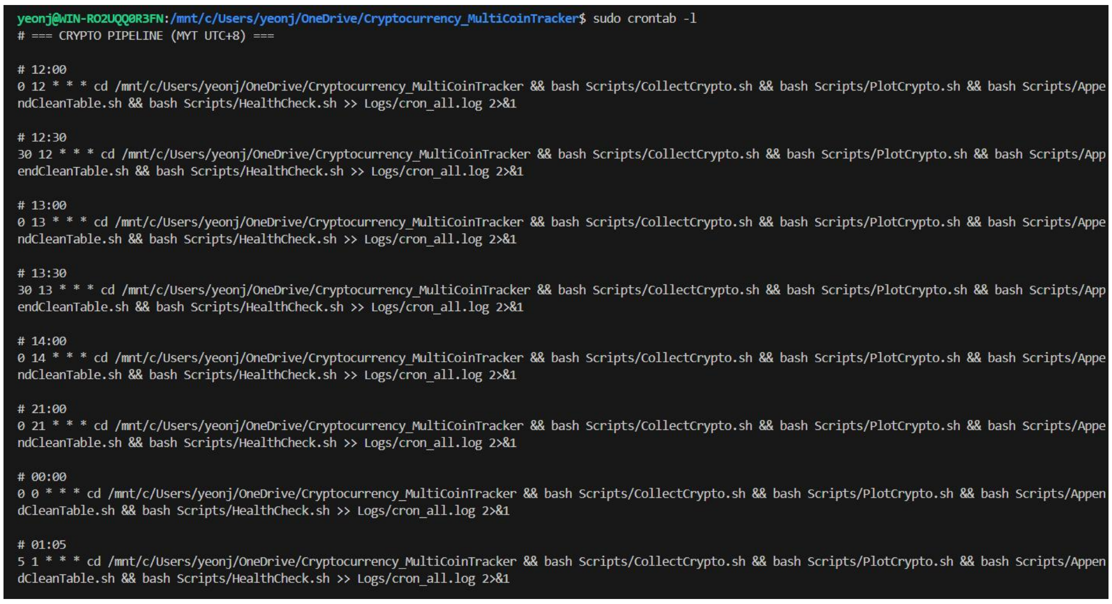
```
File Name to Write: /tmp/crontab.dtdTM3/crontab
^G Help                     M-D DOS Format              M-A Append                  M-B Backup File
^C Cancel                   M-M Mac Format              M-P Prepend                 ^T Browse
```

f) You will see that your crontab is successfully installed.



```
yeonj@WIN-RO2UQQ0R3FN:/mnt/c/Users/yeonj/OneDrive/Cryptocurrency_MultiCoinTracker$ sudo crontab -e
crontab: installing new crontab
```

g) Bash sudo crontab -l to double check.



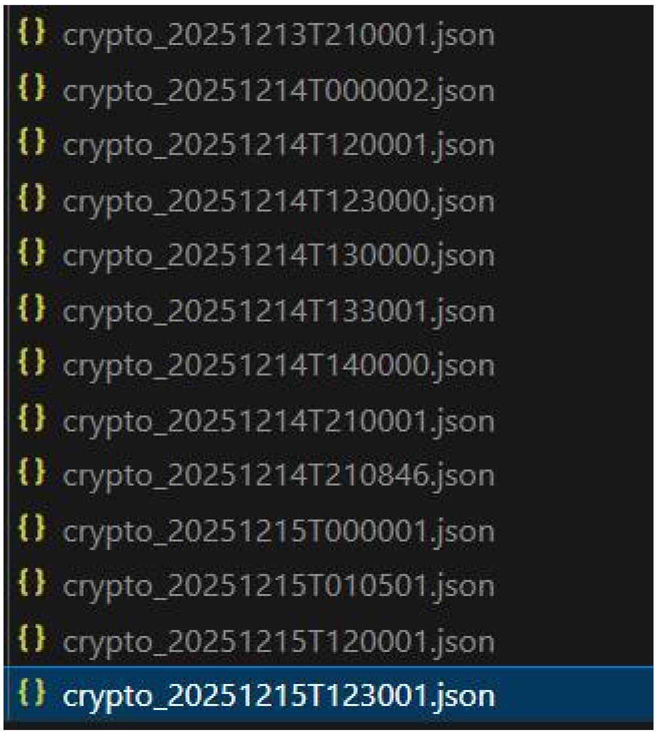
```
yeonj@WIN-RO2UQQ0R3FN:/mnt/c/Users/yeonj/OneDrive/Cryptocurrency_MultiCoinTracker$ sudo crontab -l
# === CRYPTO PIPELINE (MYT UTC+8) ===

# 12:00
0 12 * * * cd /mnt/c/Users/yeonj/OneDrive/Cryptocurrency_MultiCoinTracker && bash Scripts/CollectCrypto.sh && bash Scripts/PlotCrypto.sh && bash Scripts/Appe
ndCleanTable.sh && bash Scripts/HealthCheck.sh >> Logs/cron_all.log 2>&1

# 12:30
30 12 * * * cd /mnt/c/Users/yeonj/OneDrive/Cryptocurrency_MultiCoinTracker && bash Scripts/CollectCrypto.sh && bash Scripts/PlotCrypto.sh && bash Scripts/App
endCleanTable.sh && bash Scripts/HealthCheck.sh >> Logs/cron_all.log 2>&1

# 13:00
0 13 * * * cd /mnt/c/Users/yeonj/OneDrive/Cryptocurrency_MultiCoinTracker && bash Scripts/CollectCrypto.sh && bash Scripts/PlotCrypto.sh && bash Scripts/Appe
ndCleanTable.sh && bash Scripts/HealthCheck.sh >> Logs/cron_all.log 2>&1

# 13:30
30 13 * * * cd /mnt/c/Users/yeonj/OneDrive/Cryptocurrency_MultiCoinTracker && bash Scripts/CollectCrypto.sh && bash Scripts/PlotCrypto.sh && bash Scripts/App
endCleanTable.sh && bash Scripts/HealthCheck.sh >> Logs/cron_all.log 2>&1

# 14:00
0 14 * * * cd /mnt/c/Users/yeonj/OneDrive/Cryptocurrency_MultiCoinTracker && bash Scripts/CollectCrypto.sh && bash Scripts/PlotCrypto.sh && bash Scripts/Appe
ndCleanTable.sh && bash Scripts/HealthCheck.sh >> Logs/cron_all.log 2>&1

# 21:00
0 21 * * * cd /mnt/c/Users/yeonj/OneDrive/Cryptocurrency_MultiCoinTracker && bash Scripts/CollectCrypto.sh && bash Scripts/PlotCrypto.sh && bash Scripts/Appe
ndCleanTable.sh && bash Scripts/HealthCheck.sh >> Logs/cron_all.log 2>&1

# 00:00
0 0 * * * cd /mnt/c/Users/yeonj/OneDrive/Cryptocurrency_MultiCoinTracker && bash Scripts/CollectCrypto.sh && bash Scripts/PlotCrypto.sh && bash Scripts/Appen
dCleanTable.sh && bash Scripts/HealthCheck.sh >> Logs/cron_all.log 2>&1

# 01:05
5 1 * * * cd /mnt/c/Users/yeonj/OneDrive/Cryptocurrency_MultiCoinTracker && bash Scripts/CollectCrypto.sh && bash Scripts/PlotCrypto.sh && bash Scripts/Appen
dCleanTable.sh && bash Scripts/HealthCheck.sh >> Logs/cron_all.log 2>&1
```

Congratulation!Your crontab setup is done!

1.6) AppendCleanTable.sh(auto-cron)

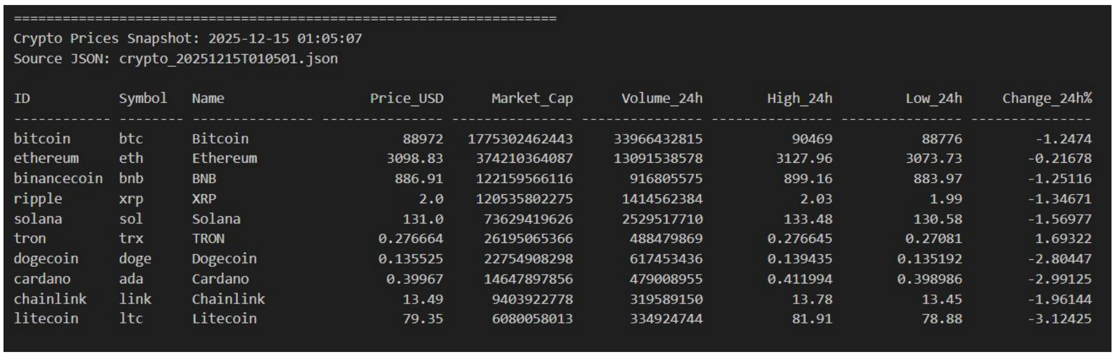
Function as: Extracts latest JSON → formats readable table

The table that will be formated will be extracted from the data json file accordingly everytime the cron runs.



```
{} crypto_20251213T210001.json
{} crypto_20251214T000002.json
{} crypto_20251214T120001.json
{} crypto_20251214T123000.json
{} crypto_20251214T130000.json
{} crypto_20251214T133001.json
{} crypto_20251214T140000.json
{} crypto_20251214T210001.json
{} crypto_20251214T210846.json
{} crypto_20251215T000001.json
{} crypto_20251215T010501.json
{} crypto_20251215T120001.json
{} crypto_20251215T123001.json
```

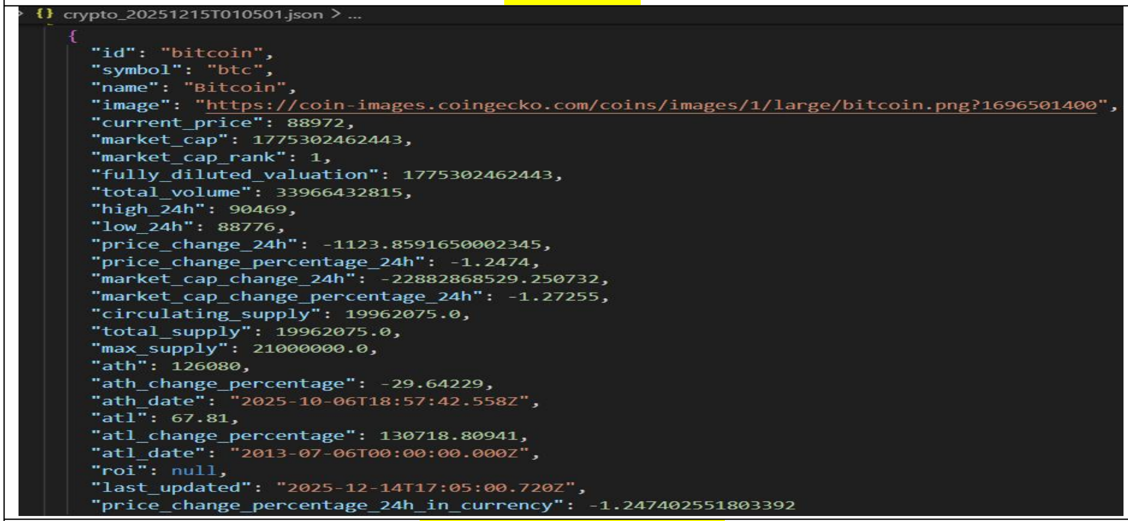
Below is the expected output (Documents/CryptoPricesTable.txt) for a table as user are easy to read manually compared to data json which is not meant for human reading.



```
===================================================================
Crypto Prices Snapshot: 2025-12-15 01:05:07
Source JSON: crypto_20251215T010501.json

ID          Symbol  Name                 Price_USD     Market_Cap     Volume_24h       High_24h        Low_24h    Change_24h%
----------- ------- -------------- -------------- -------------- -------------- -------------- -------------- --------------
bitcoin     btc     Bitcoin                  88972  1775302462443    33966432815          90469          88776        -1.2474
ethereum    eth     Ethereum               3098.83   374210364087    13091538578        3127.96        3073.73       -0.21678
binancecoin bnb     BNB                     886.91   122159566116      916805575         899.16         883.97       -1.25116
ripple      xrp     XRP                        2.0   120535802275     1414562384           2.03           1.99       -1.34671
solana      sol     Solana                   131.0    73629419626     2529517710         133.48         130.58       -1.56977
tron        trx     TRON                  0.276664    26195065366      488479869       0.276645        0.27081        1.69322
dogecoin    doge    Dogecoin              0.135525    22754908298      617453436       0.139435       0.135192       -2.80447
cardano     ada     Cardano                0.39967    14647897856      479008955       0.411994       0.398986       -2.99125
chainlink   link    Chainlink                13.49     9403922778      319589150          13.78          13.45       -1.96144
litecoin    ltc     Litecoin                 79.35     6080058013      334924744          81.91          78.88       -3.12425
```

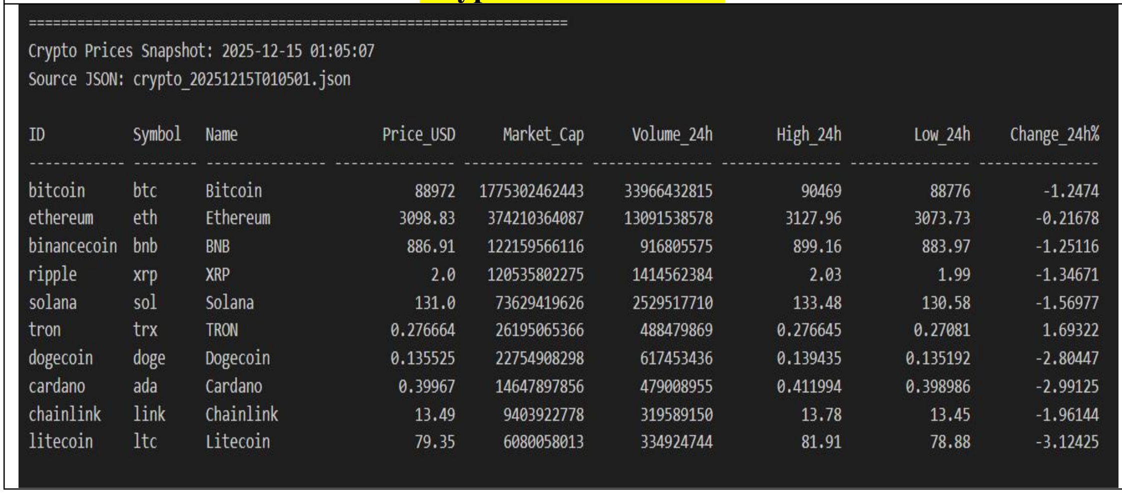
We can compare manually here.

Json Data



```
{} crypto_20251215T010501.json > ...
    {
        "id": "bitcoin",
        "symbol": "btc",
        "name": "Bitcoin",
        "image": "https://coin-images.coingecko.com/coins/images/1/large/bitcoin.png?1696501400",
        "current_price": 88972,
        "market_cap": 1775302462443,
        "market_cap_rank": 1,
        "fully_diluted_valuation": 1775302462443,
        "total_volume": 33966432815,
        "high_24h": 90469,
        "low_24h": 88776,
        "price_change_24h": -1123.8591650002345,
        "price_change_percentage_24h": -1.2474,
        "market_cap_change_24h": -22882868529.250732,
        "market_cap_change_percentage_24h": -1.27255,
        "circulating_supply": 19962075.0,
        "total_supply": 19962075.0,
        "max_supply": 21000000.0,
        "ath": 126080,
        "ath_change_percentage": -29.64229,
        "ath_date": "2025-10-06T18:57:42.558Z",
        "atl": 67.81,
        "atl_change_percentage": 130718.80941,
        "atl_date": "2013-07-06T00:00:00.000Z",
        "roi": null,
        "last_updated": "2025-12-14T17:05:00.720Z",
        "price_change_percentage_24h_in_currency": -1.247402551803392
```

CryptoPricesTable.txt



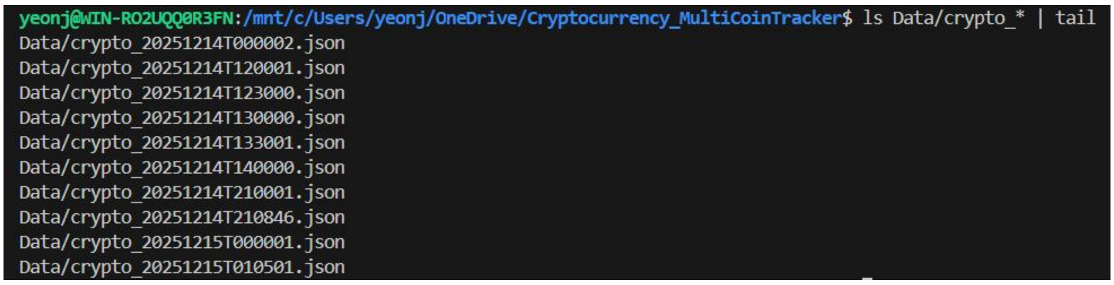
```
===================================================================
Crypto Prices Snapshot: 2025-12-15 01:05:07
Source JSON: crypto_20251215T010501.json

ID          Symbol  Name                 Price_USD     Market_Cap     Volume_24h       High_24h        Low_24h    Change_24h%
----------- ------- -------------- -------------- -------------- -------------- -------------- -------------- --------------
bitcoin     btc     Bitcoin                  88972  1775302462443    33966432815          90469          88776        -1.2474
ethereum    eth     Ethereum               3098.83   374210364087    13091538578        3127.96        3073.73       -0.21678
binancecoin bnb     BNB                     886.91   122159566116      916805575         899.16         883.97       -1.25116
ripple      xrp     XRP                        2.0   120535802275     1414562384           2.03           1.99       -1.34671
solana      sol     Solana                   131.0    73629419626     2529517710         133.48         130.58       -1.56977
tron        trx     TRON                  0.276664    26195065366      488479869       0.276645        0.27081        1.69322
dogecoin    doge    Dogecoin              0.135525    22754908298      617453436       0.139435       0.135192       -2.80447
cardano     ada     Cardano                0.39967    14647897856      479008955       0.411994       0.398986       -2.99125
chainlink   link    Chainlink                13.49     9403922778      319589150          13.78          13.45       -1.96144
litecoin    ltc     Litecoin                 79.35     6080058013      334924744          81.91          78.88       -3.12425
```

1.7) Script Workflow (Scripts/CollectCrypto.sh) (auto cron)

a) This scripts defines environment paths which uses BASE_DIR, DATA_DIR and LOG_DIR.

b) Selects 10 tracked cryptocurrencies.
c) Builds CoinGecko API URL.

d) Download JSON using curl + retry (up to 3 times)
  - Saves clean, formatted JSON (crypto_YYYYMMDDTHHMMSS.json) in Data/.
  - Example filename: crypto_20251214T210846.json



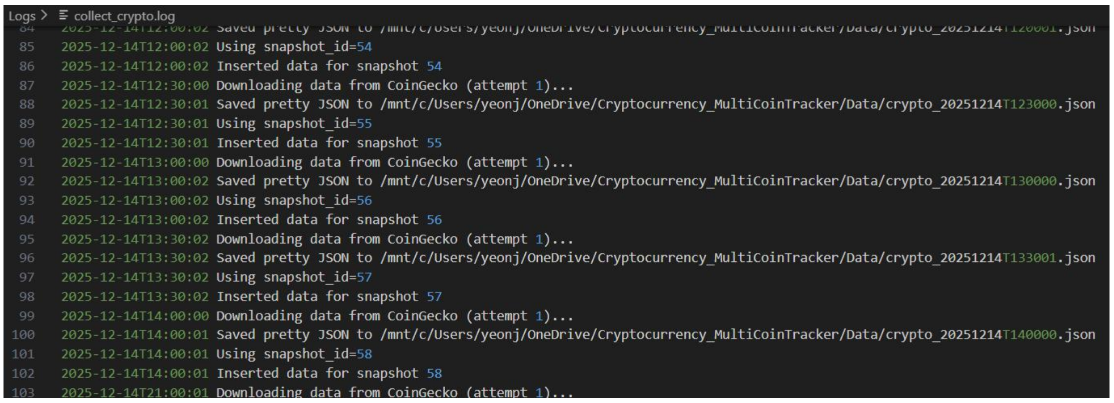
```
yeonj@WIN-RO2UQQ0R3FN:/mnt/c/Users/yeonj/OneDrive/Cryptocurrency_MultiCoinTracker$ ls Data/crypto_* | tail
Data/crypto_20251214T000002.json
Data/crypto_20251214T120001.json
Data/crypto_20251214T123000.json
Data/crypto_20251214T130000.json
Data/crypto_20251214T133001.json
Data/crypto_20251214T140000.json
Data/crypto_20251214T210001.json
Data/crypto_20251214T210846.json
Data/crypto_20251215T000001.json
Data/crypto_20251215T010501.json
```

e) Validates and parses JSON using jq

  Extracts .id, .current_price, .market_cap, .total_volume, .low_24h, .high_24h, .price_change_percentage_24h.

  Converts into tab-separated lines ready for SQL insertion.

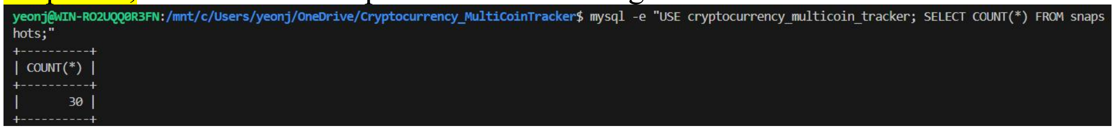
f) Inserts one row into snapshots which records snapshot time + API source URL.
g) Loops through all coins and inserts rows into coin_prices table which auto-links each coin to its id in the coins table.
h) Finally, writes progress logs in Logs/collect_crypto.log.



Logs > collect_crypto.log

```
2025-12-14T12:00:02 Saved pretty JSON to /mnt/c/Users/yeonj/OneDrive/Cryptocurrency_MultiCoinTracker/Data/crypto_20251214T120001.json
2025-12-14T12:00:02 Using snapshot_id=54
2025-12-14T12:00:02 Inserted data for snapshot 54
2025-12-14T12:30:00 Downloading data from CoinGecko (attempt 1)...
2025-12-14T12:30:01 Saved pretty JSON to /mnt/c/Users/yeonj/OneDrive/Cryptocurrency_MultiCoinTracker/Data/crypto_20251214T123000.json
2025-12-14T12:30:01 Using snapshot_id=55
2025-12-14T12:30:01 Inserted data for snapshot 55
2025-12-14T13:00:00 Downloading data from CoinGecko (attempt 1)...
2025-12-14T13:00:02 Saved pretty JSON to /mnt/c/Users/yeonj/OneDrive/Cryptocurrency_MultiCoinTracker/Data/crypto_20251214T130000.json
2025-12-14T13:00:02 Using snapshot_id=56
2025-12-14T13:00:02 Inserted data for snapshot 56
2025-12-14T13:30:02 Downloading data from CoinGecko (attempt 1)...
2025-12-14T13:30:02 Saved pretty JSON to /mnt/c/Users/yeonj/OneDrive/Cryptocurrency_MultiCoinTracker/Data/crypto_20251214T133001.json
2025-12-14T13:30:02 Using snapshot_id=57
2025-12-14T13:30:02 Inserted data for snapshot 57
2025-12-14T14:00:00 Downloading data from CoinGecko (attempt 1)...
2025-12-14T14:00:01 Saved pretty JSON to /mnt/c/Users/yeonj/OneDrive/Cryptocurrency_MultiCoinTracker/Data/crypto_20251214T140000.json
2025-12-14T14:00:01 Using snapshot_id=58
2025-12-14T14:00:01 Inserted data for snapshot 58
2025-12-14T21:00:01 Downloading data from CoinGecko (attempt 1)...
```

Then we bash "USE cryptocurrency_multicoin_tracker; SELECT COUNT(*) FROM snapshots;" it will Show snapshot rows increasing



```
yeonj@WIN-RO2UQQ0R3FN:/mnt/c/Users/yeonj/OneDrive/Cryptocurrency_MultiCoinTracker$ mysql -e "USE cryptocurrency_multicoin_tracker; SELECT COUNT(*) FROM snaps
hots;"
+----------+
| COUNT(*) |
+----------+
|       30 |
+----------+
```

Section 2 (Unix Script for Plotting)

Objective: Generate a minimum of TEN (10) different plots from the stored historical data using a Unix script (functions).

2.1) Plots Delivered

10× price plots: BTC, ETH, ADA, BNB, DOGE, LINK, LTC, SOL, TRX, XRP
Extra plots for 'diverse metrics' (advanced): BTC/ETH 24h change, market cap, and 24h volume.

Chosen BTC and ETH for extra plots as they both are currently the most famous in cryptocurrency.

2.2) How Plotting Works (Scripts/PlotCrypto.sh) (auto cron)

This is the core analytics script that transforms MySQL data into charts.

Function	Purpose	Output
Plot_price()	Line graph of coin price vs time	*_price.png
Plot_change()	24h % change over time (BTC, ETH)	*_change24.png
Plot_marketcap()	Market capitalization	*_marketcap.png
Plot_volume()	24h trading volume	*_volume24h.png

The plotting script queries MySQL for the latest time-series values, writes them into temporary .dat files, and renders PNG outputs into Plots/ using gnuplot. Reusable plot functions keep the script clean and extendable.

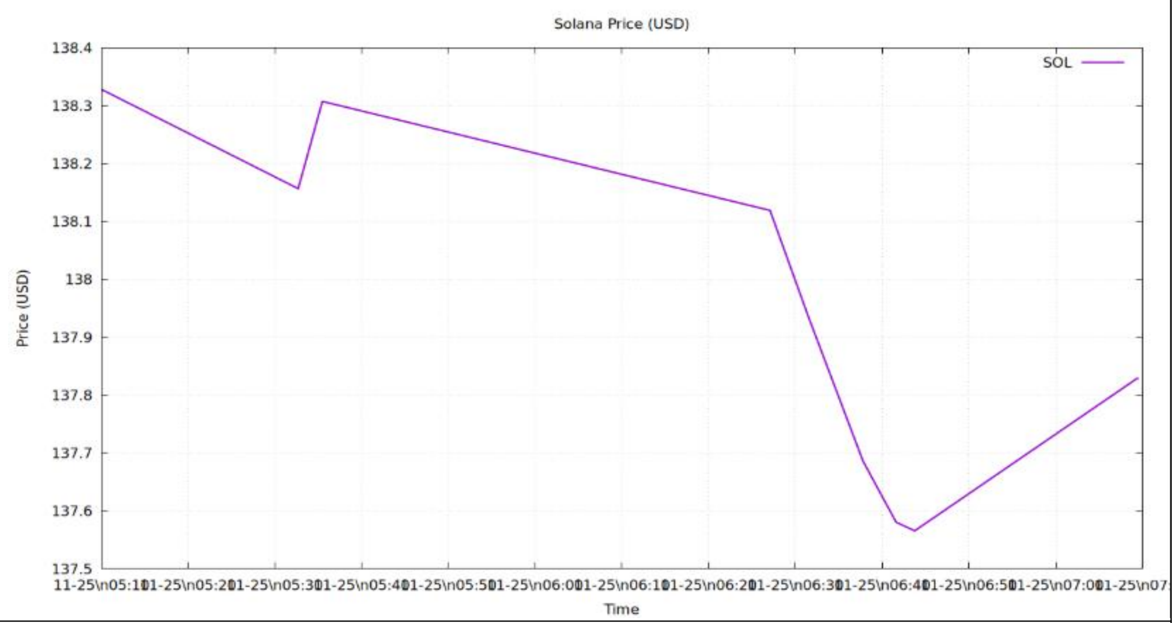
Steps:

a) Queries MySQL using mysql -N -B -e.
b) Formats data as tab-separated values for Gnuplot.
c) Gnuplot plots line charts with time-based X-axis.
d) Produces PNGs in /Plots/

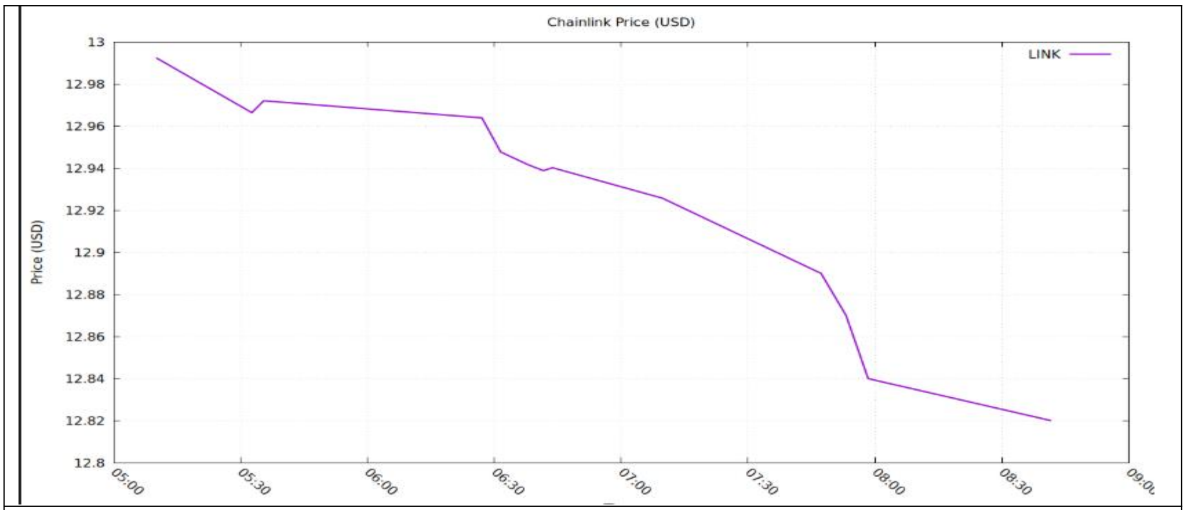
Supported modes:

Bash	What will you get?
bash Scripts/PlotCrypto.sh all	# All plots (default)
bash Scripts/PlotCrypto.sh price	# Only price
bash Scripts/PlotCrypto.sh change	# 24h change for BTC/ETH
bash Scripts/PlotCrypto.sh marketcap	# Market cap (BTC/ETH)
bash Scripts/PlotCrypto.sh volume	# 24h volume (BTC/ETH)

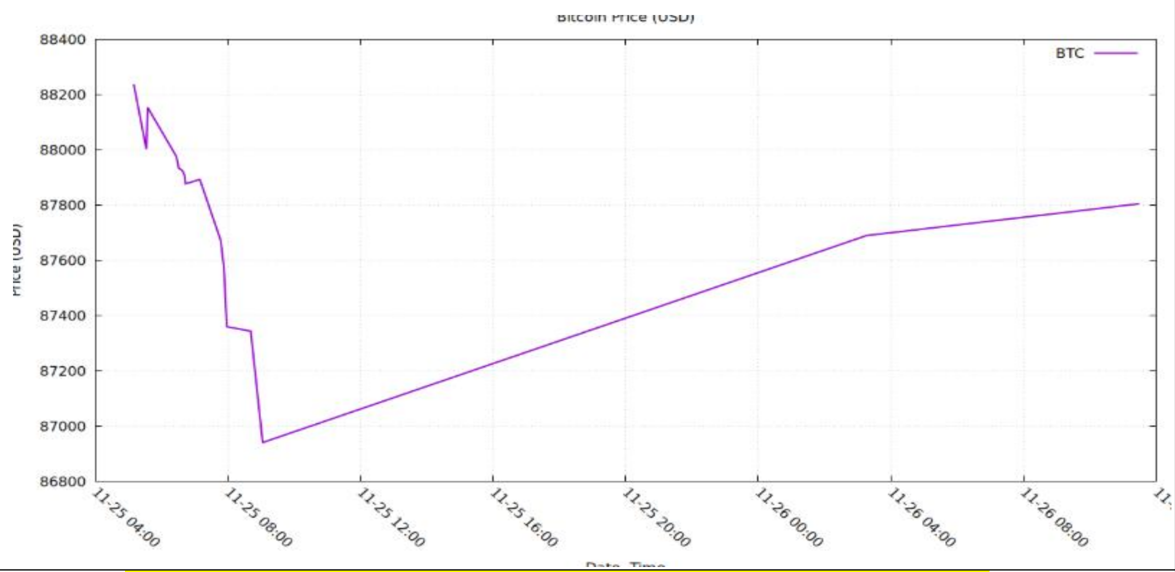
2.3 Important Fixes (what went wrong and how it was solved)



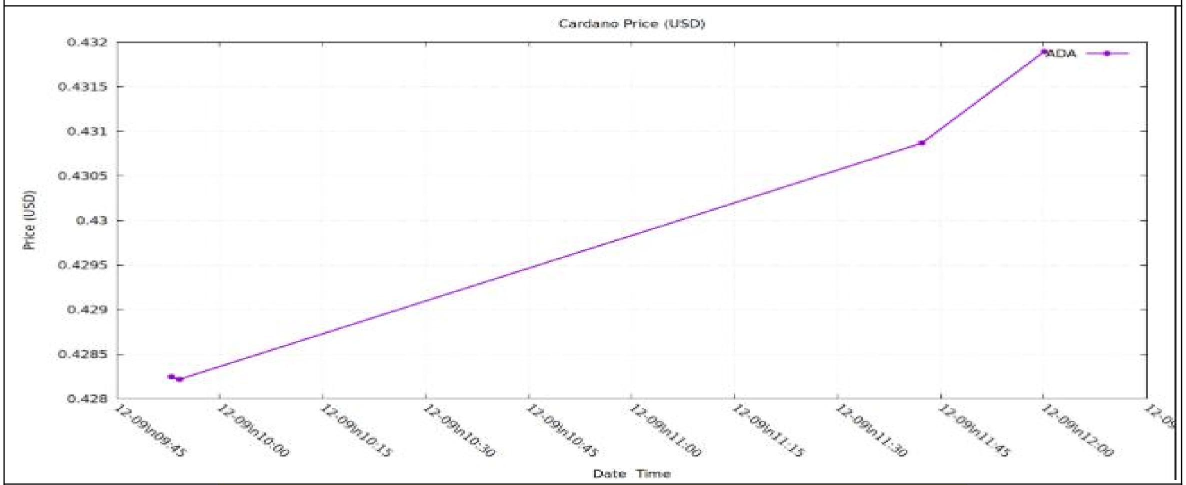
This is the first version of graph. The time and date is mixed together which is difficult for users to read.



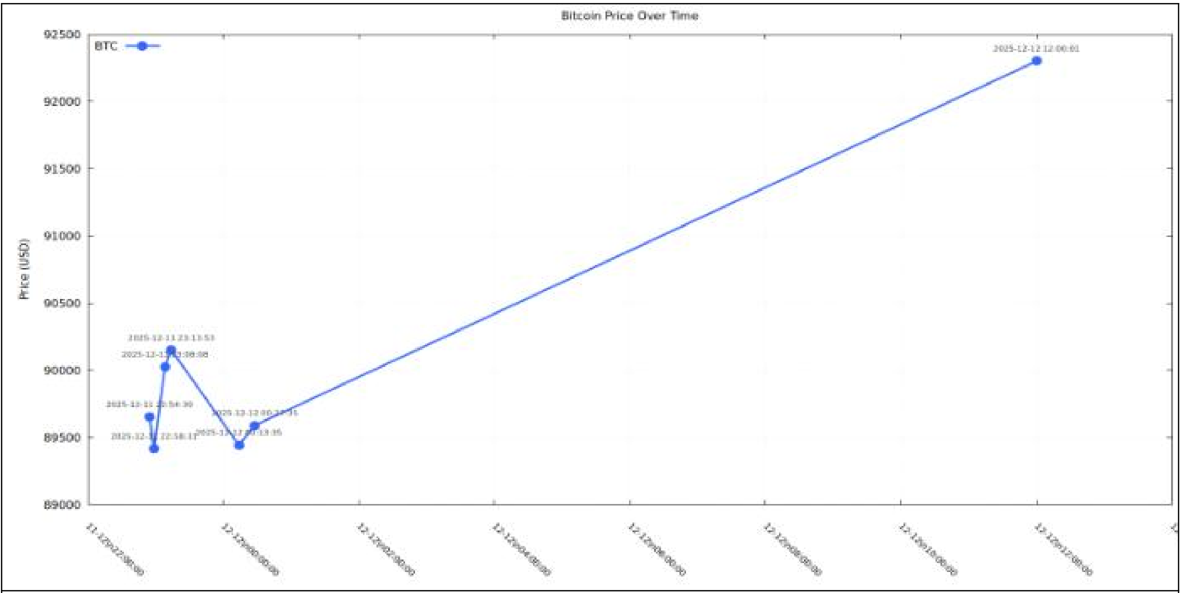
The date has been removed and left time only to remove complexity. But this is not user friendly enough as will make confusion.



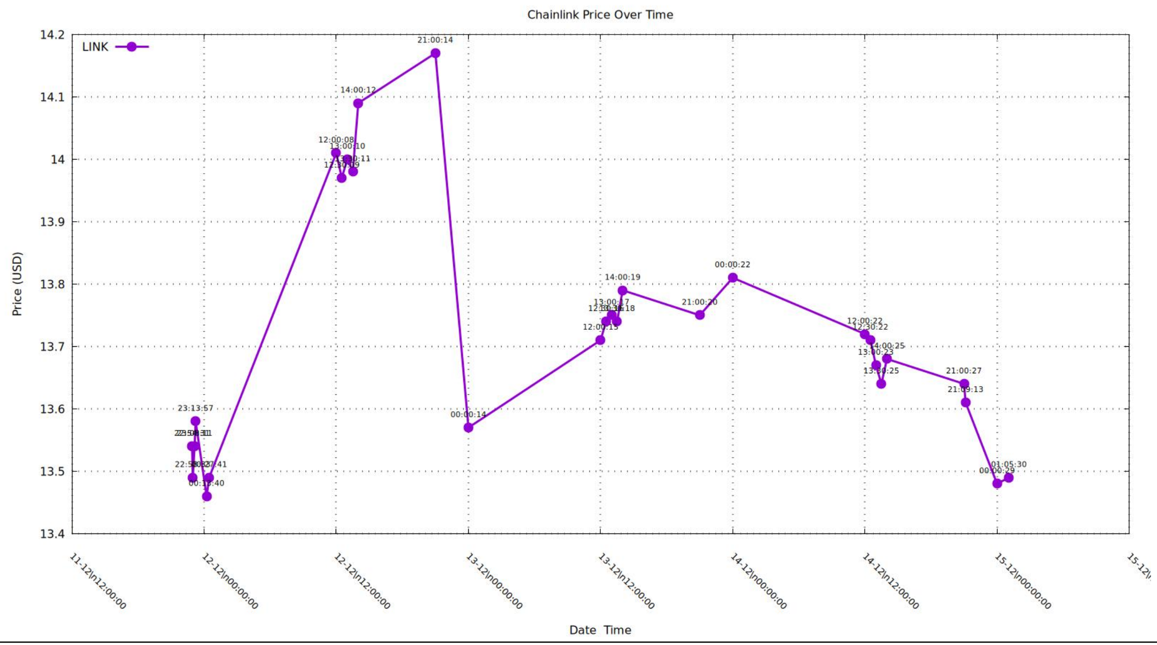
The time and date can be seen correctly but the x plot is not visible.



The x plot is now visible, which easier for user to identify.



This version is better but only the date and time duplicates in the x plot. Hence, the date will be removed and only timing is on the plot.



This will be the final version of graph plotting. Clean and professional with all information.

2.5) Example of expected graph plotting output

Figure 1. BTC Price Over Time (sample output)

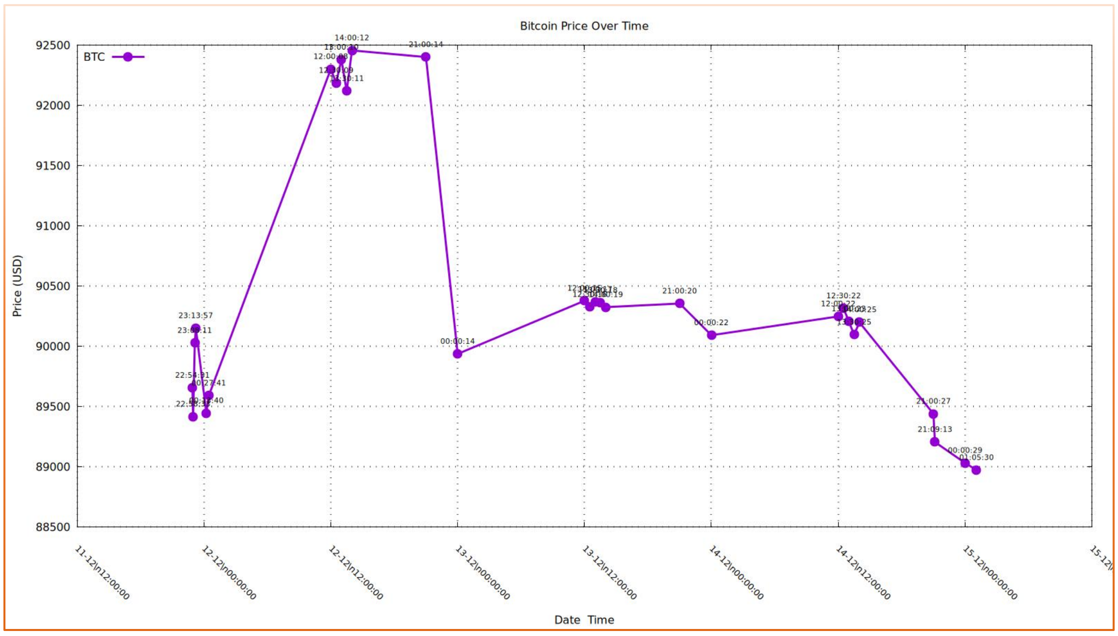


Figure 2. BTC Market Cap Over Time (diverse metric)

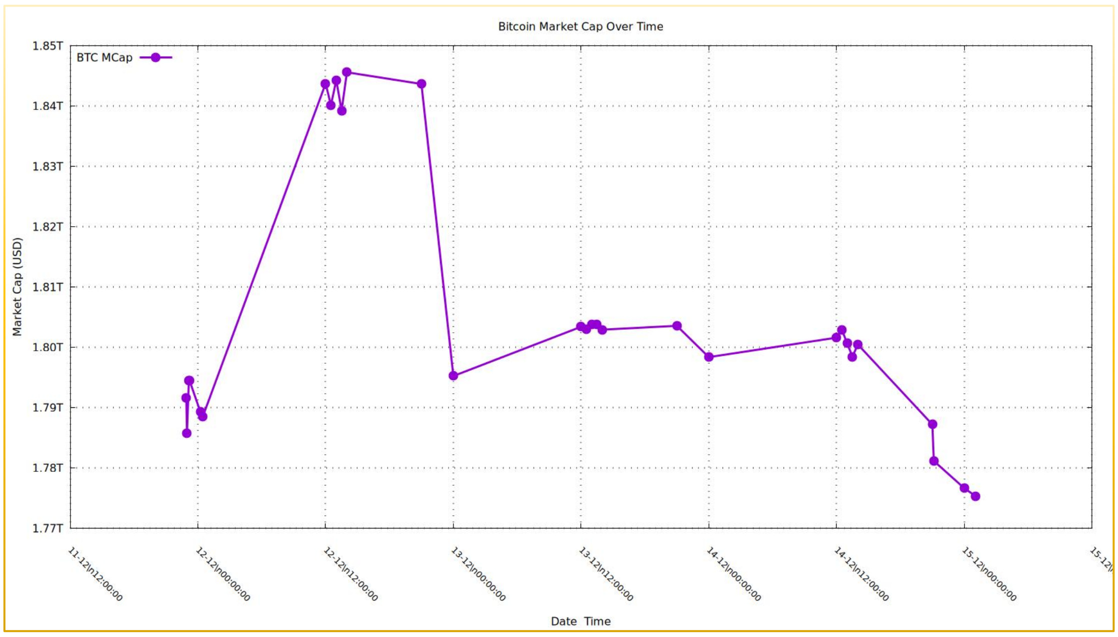


Figure 3. BTC 24h Volume Over Time (diverse metric)

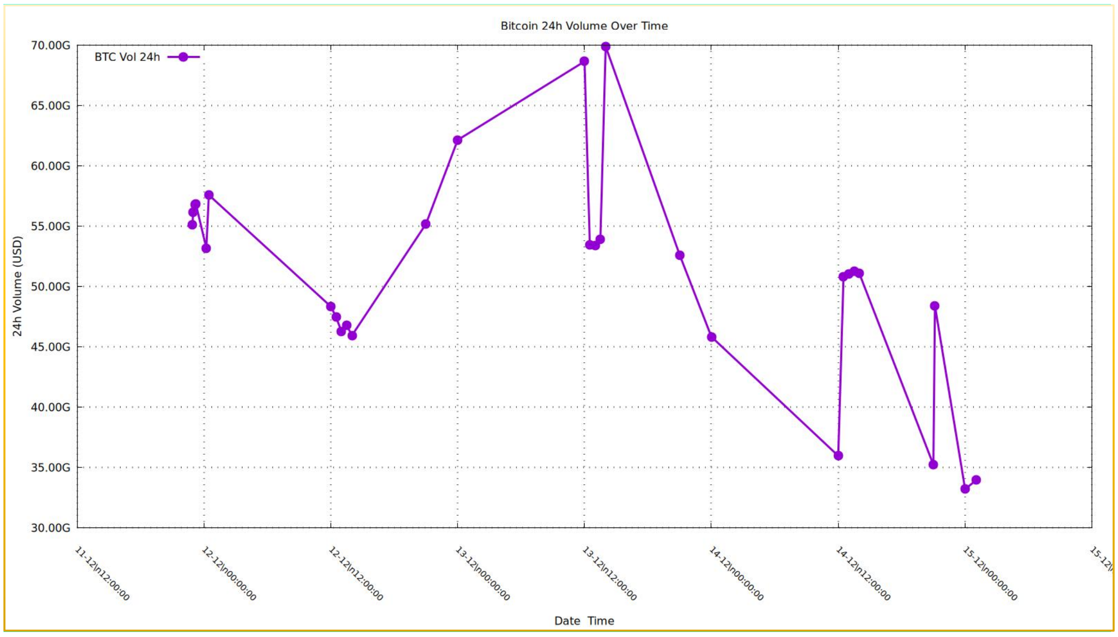
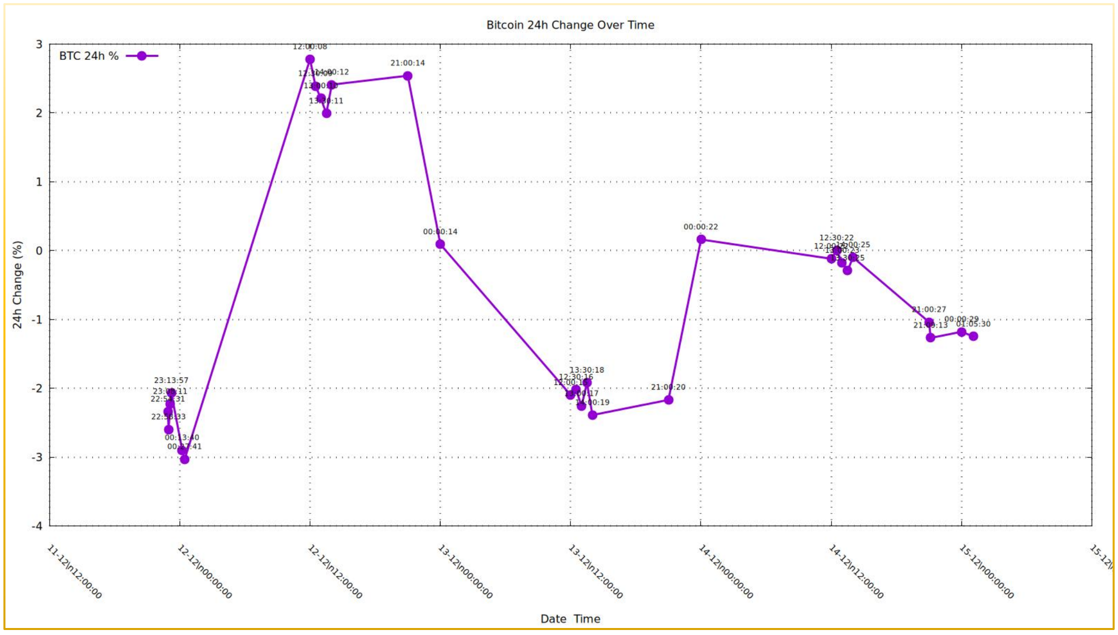


Figure 4. ETH 24h Change Over Time (diverse metric)

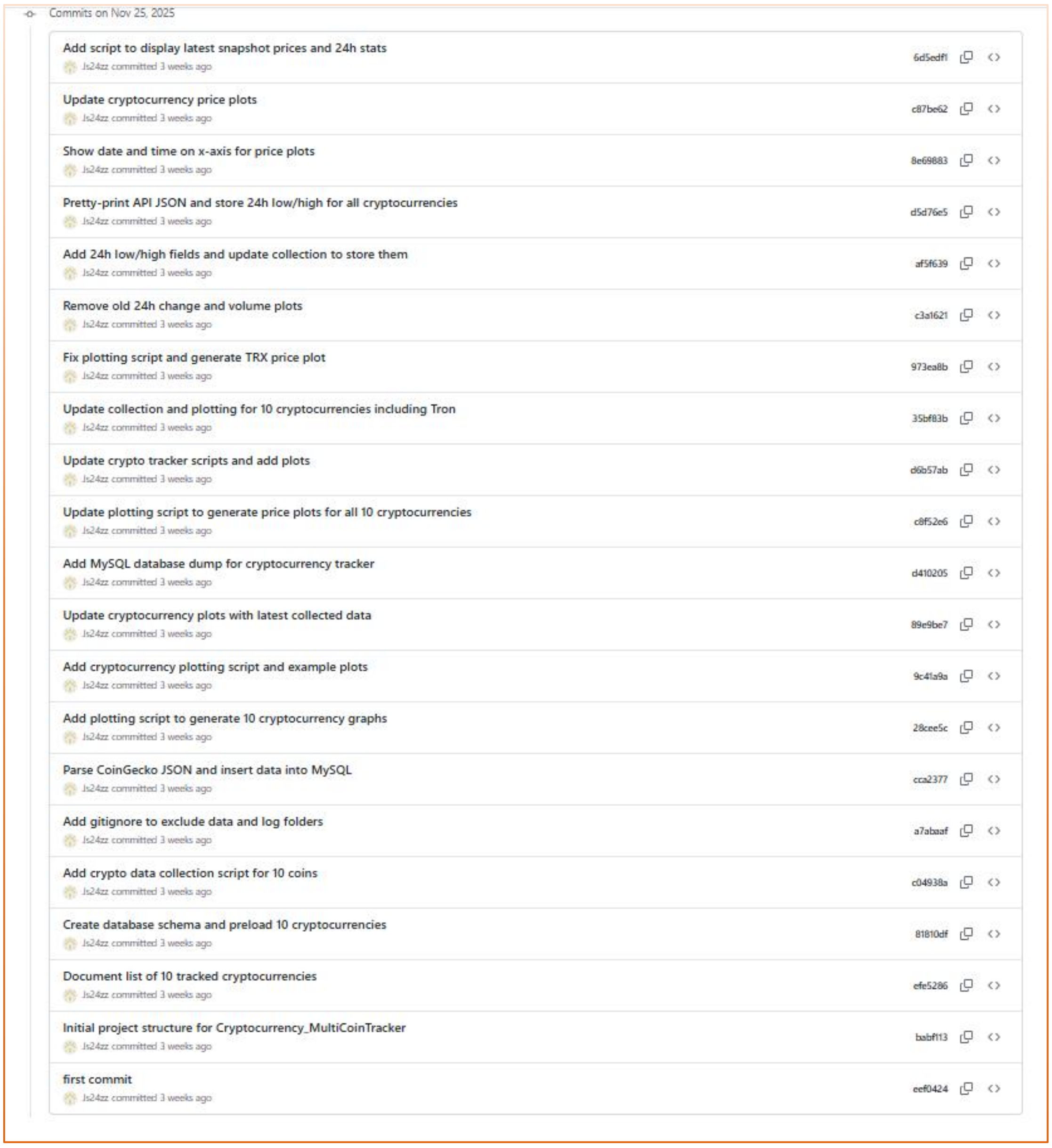


Section 3 (Git Version Control)

https://github.com/Js24zz/Cryptocurrency_MultiCoinTracker.git

Objective: show professional version control: at least 10 commits spread across at least one week, plus GitHub URL evidence.

3.1 First Commit Date



Commits on Nov 25, 2025

Add script to display latest snapshot prices and 24h stats
Js24zz committed 3 weeks ago — 6d5edf1

Update cryptocurrency price plots
Js24zz committed 3 weeks ago — c87be62

Show date and time on x-axis for price plots
Js24zz committed 3 weeks ago — 8e69883

Pretty-print API JSON and store 24h low/high for all cryptocurrencies
Js24zz committed 3 weeks ago — d5d76e5

Add 24h low/high fields and update collection to store them
Js24zz committed 3 weeks ago — af5f639

Remove old 24h change and volume plots
Js24zz committed 3 weeks ago — c3a1621

Fix plotting script and generate TRX price plot
Js24zz committed 3 weeks ago — 973ea8b

Update collection and plotting for 10 cryptocurrencies including Tron
Js24zz committed 3 weeks ago — 35bf83b

Update crypto tracker scripts and add plots
Js24zz committed 3 weeks ago — d6b57ab

Update plotting script to generate price plots for all 10 cryptocurrencies
Js24zz committed 3 weeks ago — c8f52e6

Add MySQL database dump for cryptocurrency tracker
Js24zz committed 3 weeks ago — d410205

Update cryptocurrency plots with latest collected data
Js24zz committed 3 weeks ago — 89e9be7

Add cryptocurrency plotting script and example plots
Js24zz committed 3 weeks ago — 9c41a9a

Add plotting script to generate 10 cryptocurrency graphs
Js24zz committed 3 weeks ago — 28cee5c

Parse CoinGecko JSON and insert data into MySQL
Js24zz committed 3 weeks ago — cca2377

Add gitignore to exclude data and log folders
Js24zz committed 3 weeks ago — a7abaaf

Add crypto data collection script for 10 coins
Js24zz committed 3 weeks ago — c04938a

Create database schema and preload 10 cryptocurrencies
Js24zz committed 3 weeks ago — 81810df

Document list of 10 tracked cryptocurrencies
Js24zz committed 3 weeks ago — efe5286

Initial project structure for Cryptocurrency_MultiCoinTracker
Js24zz committed 3 weeks ago — babf113

first commit
Js24zz committed 3 weeks ago — eef0424

All the commit history can be found in my Github Repository.

Section 4 (MySQL Data Storage)

Objective: store data in multiple related tables and prove correctness with ERD + MySQL dump file for import.

4.1 ERD & Table Relationships

The Entity–Relationship Diagram (ERD) defines how cryptocurrency market data is stored in a structured, queryable relational database. The key design goal is to capture repeated "snapshots" of multiple coins over time without duplicating coin metadata, and to enforce data integrity via primary keys (PK), foreign keys (FK) and a uniqueness rule per (snapshot, coin).

a) Table: coins

The coins table is the master list of tracked cryptocurrencies. It stores:

• **coingecko_id** — the identifier used in the CoinGecko API request
• **symbol** — ticker symbol used in outputs and plots (e.g., BTC)
• **name** — human-readable coin name (e.g., Bitcoin)

It uses id as the PK and enforces UNIQUE(coingecko_id) so each CoinGecko coin appears once.

b) Table: snapshots

The snapshots table records each data collection run. This includes:

• **snapshot_time** (datetime) — when the snapshot was collected (MYT / UTC+8 in your cron schedule)
• **source** (varchar) — the API URL used for traceability (evidence of data provenance)

This design allows time-series queries and simplifies 'latest snapshot' lookups

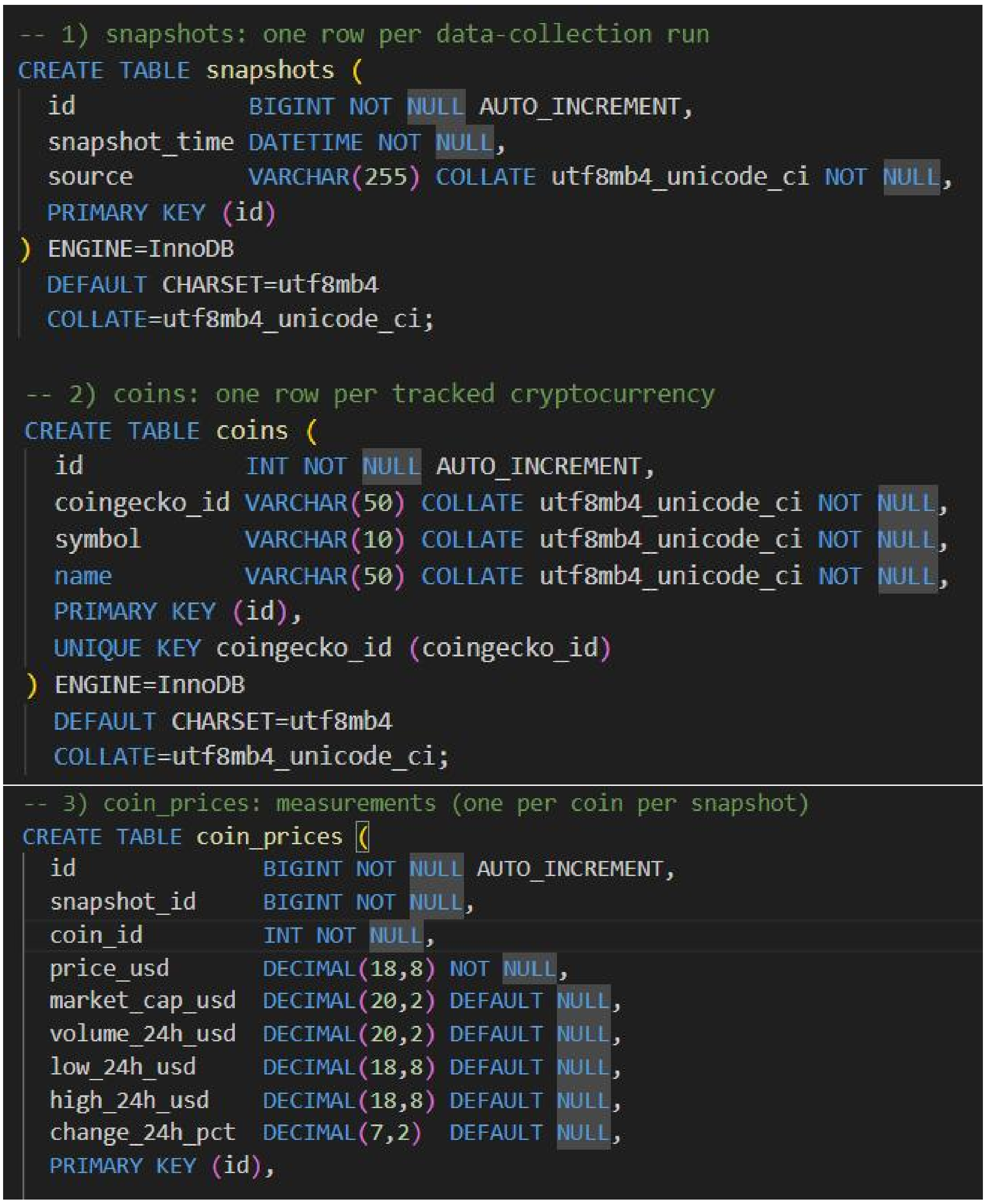
c) Table: coin_prices

The **coin_prices** table stores the market metrics for each coin at each snapshot. It contains FKs to snapshots and coins, plus numeric metrics such as **price_usd, market_cap_usd, volume_24h_usd, low_24h_usd, high_24h_usd**, and **change_24h_pct.**

Critical rule: it enforces **UNIQUE(snapshot_id, coin_id)** so each snapshot contains at most one row per coin. This prevents duplicates if a script is accidentally re-run for the same snapshot

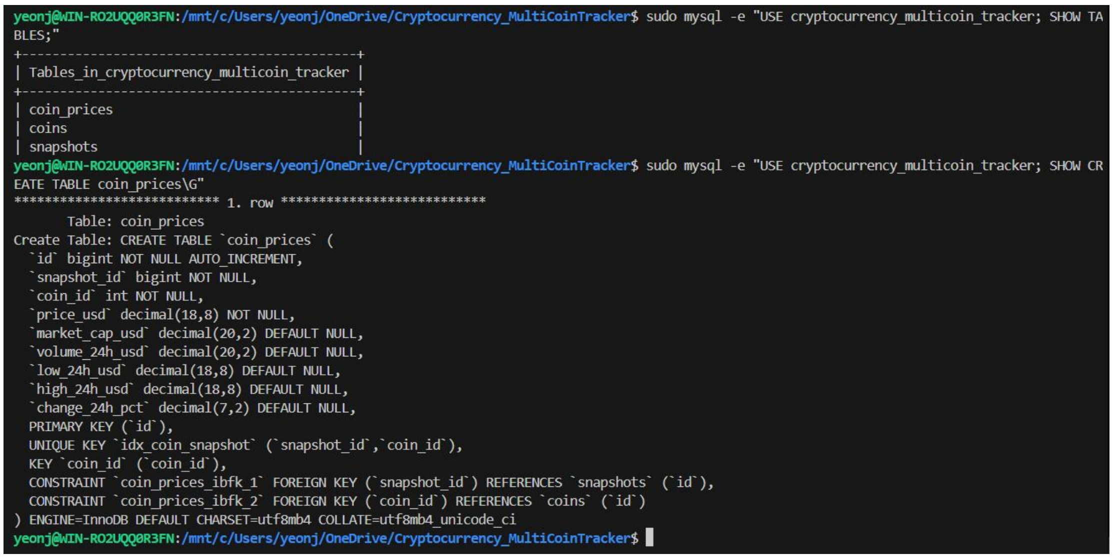
4.2)Keys, Constraints, and Data Types (as implemented in MySQL)

The following DDL is taken from your database dump (cryptocurrency_multicoin_tracker_dump.sql). Including this in your submission helps the marker verify your implementation matches the ERD.



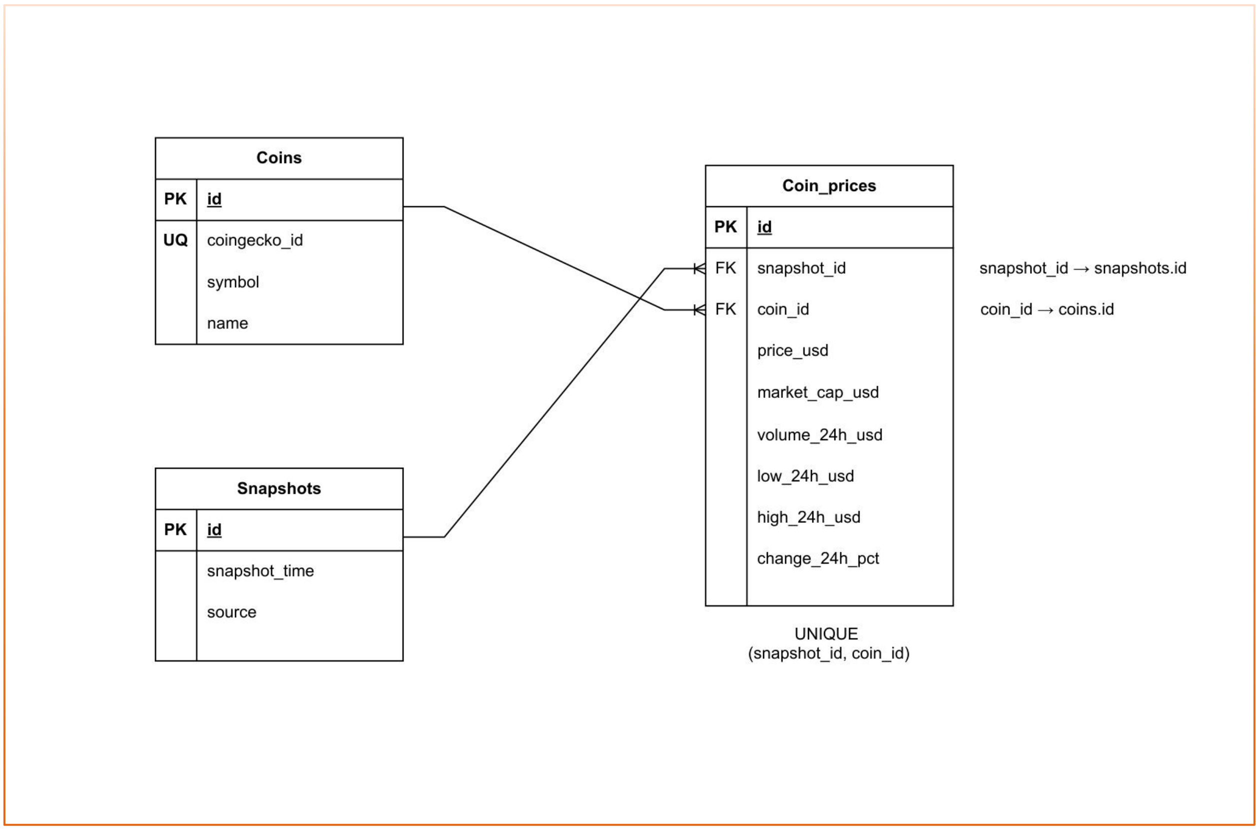
```sql
-- 1) snapshots: one row per data-collection run
CREATE TABLE snapshots (
  id            BIGINT NOT NULL AUTO_INCREMENT,
  snapshot_time DATETIME NOT NULL,
  source        VARCHAR(255) COLLATE utf8mb4_unicode_ci NOT NULL,
  PRIMARY KEY (id)
) ENGINE=InnoDB
  DEFAULT CHARSET=utf8mb4
  COLLATE=utf8mb4_unicode_ci;

-- 2) coins: one row per tracked cryptocurrency
CREATE TABLE coins (
  id           INT NOT NULL AUTO_INCREMENT,
  coingecko_id VARCHAR(50) COLLATE utf8mb4_unicode_ci NOT NULL,
  symbol       VARCHAR(10) COLLATE utf8mb4_unicode_ci NOT NULL,
  name         VARCHAR(50) COLLATE utf8mb4_unicode_ci NOT NULL,
  PRIMARY KEY (id),
  UNIQUE KEY coingecko_id (coingecko_id)
) ENGINE=InnoDB
  DEFAULT CHARSET=utf8mb4
  COLLATE=utf8mb4_unicode_ci;
-- 3) coin_prices: measurements (one per coin per snapshot)
CREATE TABLE coin_prices (
  id             BIGINT NOT NULL AUTO_INCREMENT,
  snapshot_id    BIGINT NOT NULL,
  coin_id        INT NOT NULL,
  price_usd      DECIMAL(18,8) NOT NULL,
  market_cap_usd DECIMAL(20,2) DEFAULT NULL,
  volume_24h_usd DECIMAL(20,2) DEFAULT NULL,
  low_24h_usd    DECIMAL(18,8) DEFAULT NULL,
  high_24h_usd   DECIMAL(18,8) DEFAULT NULL,
  change_24h_pct DECIMAL(7,2)  DEFAULT NULL,
  PRIMARY KEY (id),
```

Data type rationale: DECIMAL is used for financial values to avoid floating-point rounding errors. The chosen precisions support high-value coins (BTC) and low-value coins (DOGE) while keeping accuracy



```
yeonj@WIN-RO2UQQ0R3FN:/mnt/c/Users/yeonj/OneDrive/Cryptocurrency_MultiCoinTracker$ sudo mysql -e "USE cryptocurrency_multicoin_tracker; SHOW TA
BLES;"
+-------------------------------------------+
| Tables_in_cryptocurrency_multicoin_tracker |
+-------------------------------------------+
| coin_prices                                |
| coins                                      |
| snapshots                                  |
yeonj@WIN-RO2UQQ0R3FN:/mnt/c/Users/yeonj/OneDrive/Cryptocurrency_MultiCoinTracker$ sudo mysql -e "USE cryptocurrency_multicoin_tracker; SHOW CR
EATE TABLE coin_prices\G"
*************************** 1. row ***************************
       Table: coin_prices
Create Table: CREATE TABLE `coin_prices` (
  `id` bigint NOT NULL AUTO_INCREMENT,
  `snapshot_id` bigint NOT NULL,
  `coin_id` int NOT NULL,
  `price_usd` decimal(18,8) NOT NULL,
  `market_cap_usd` decimal(20,2) DEFAULT NULL,
  `volume_24h_usd` decimal(20,2) DEFAULT NULL,
  `low_24h_usd` decimal(18,8) DEFAULT NULL,
  `high_24h_usd` decimal(18,8) DEFAULT NULL,
  `change_24h_pct` decimal(7,2) DEFAULT NULL,
  PRIMARY KEY (`id`),
  UNIQUE KEY `idx_coin_snapshot` (`snapshot_id`,`coin_id`),
  KEY `coin_id` (`coin_id`),
  CONSTRAINT `coin_prices_ibfk_1` FOREIGN KEY (`snapshot_id`) REFERENCES `snapshots` (`id`),
  CONSTRAINT `coin_prices_ibfk_2` FOREIGN KEY (`coin_id`) REFERENCES `coins` (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci
yeonj@WIN-RO2UQQ0R3FN:/mnt/c/Users/yeonj/OneDrive/Cryptocurrency_MultiCoinTracker$
```

Figure 5. ERD (exported from draw.io)



4.3) Schema vs Dump (Should you merge?)

We keep them separate for marking clarity:
• Schema.sql = structure only
• cryptocurrency_multicoin_tracker_dump.sql = structure + data

Separate files are clearer and safer.

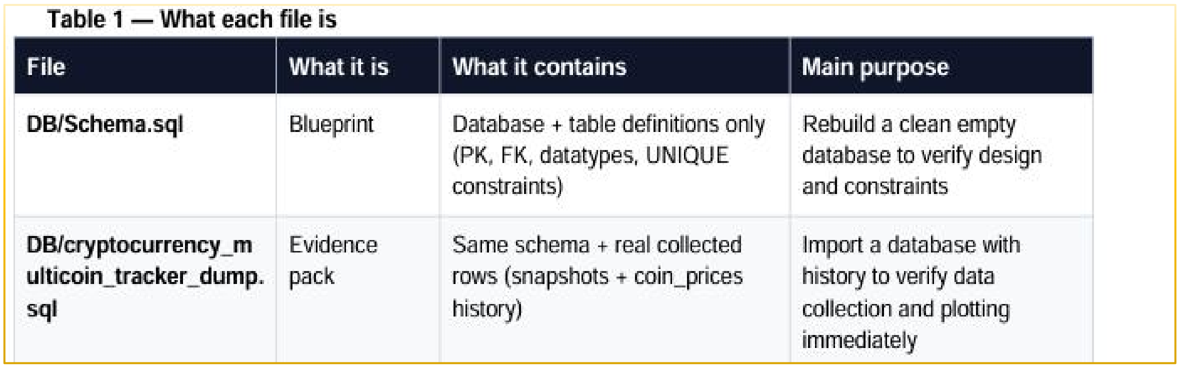


Table 1 — What each file is

File	What it is	What it contains	Main purpose
DB/Schema.sql	Blueprint	Database + table definitions only (PK, FK, datatypes, UNIQUE constraints)	Rebuild a clean empty database to verify design and constraints
DB/cryptocurrency_multicoin_tracker_dump.sql	Evidence pack	Same schema + real collected rows (snapshots + coin_prices history)	Import a database with history to verify data collection and plotting immediately

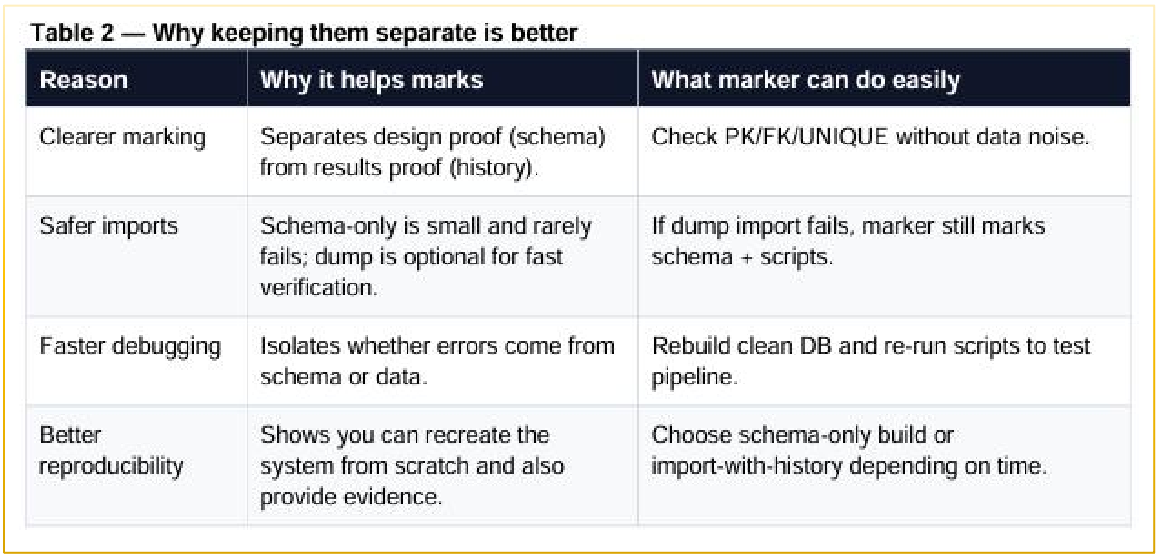


Table 2 — Why keeping them separate is better

Reason	Why it helps marks	What marker can do easily
Clearer marking	Separates design proof (schema) from results proof (history).	Check PK/FK/UNIQUE without data noise.
Safer imports	Schema-only is small and rarely fails; dump is optional for fast verification.	If dump import fails, marker still marks schema + scripts.
Faster debugging	Isolates whether errors come from schema or data.	Rebuild clean DB and re-run scripts to test pipeline.
Better reproducibility	Shows you can recreate the system from scratch and also provide evidence.	Choose schema-only build or import-with-history depending on time.

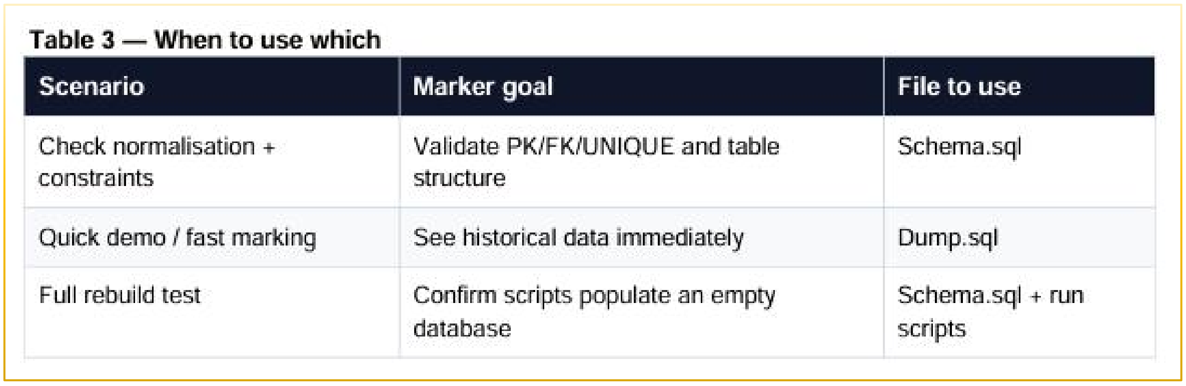


Table 3 — When to use which

Scenario	Marker goal	File to use
Check normalisation + constraints	Validate PK/FK/UNIQUE and table structure	Schema.sql
Quick demo / fast marking	See historical data immediately	Dump.sql
Full rebuild test	Confirm scripts populate an empty database	Schema.sql + run scripts

We keep two SQL files because they prove two different things. Schema.sql is the blueprint: it recreates an empty database with the correct tables, PK/FK and constraints, so the marker can verify the database design from scratch.

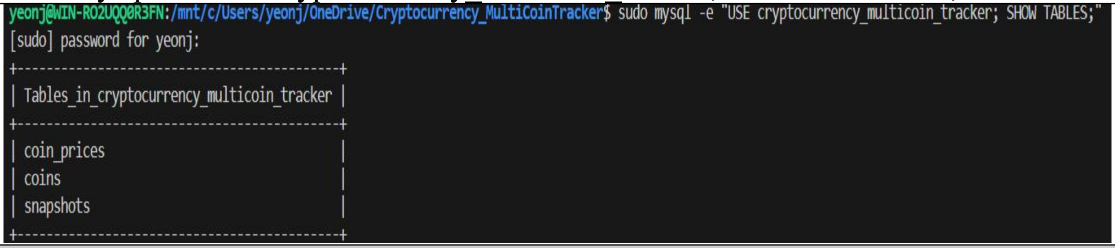
Dump.sql is the evidence pack: it includes the same structure plus real collected history (snapshots + coin_prices), so the marker can import and immediately verify that the system stored data over time and that plots can be generated from historical data.

Keeping them separate is clearer and safer than merging, because one file proves the design and the other proves the results

4.4) MySQL Evidence

Show table exist

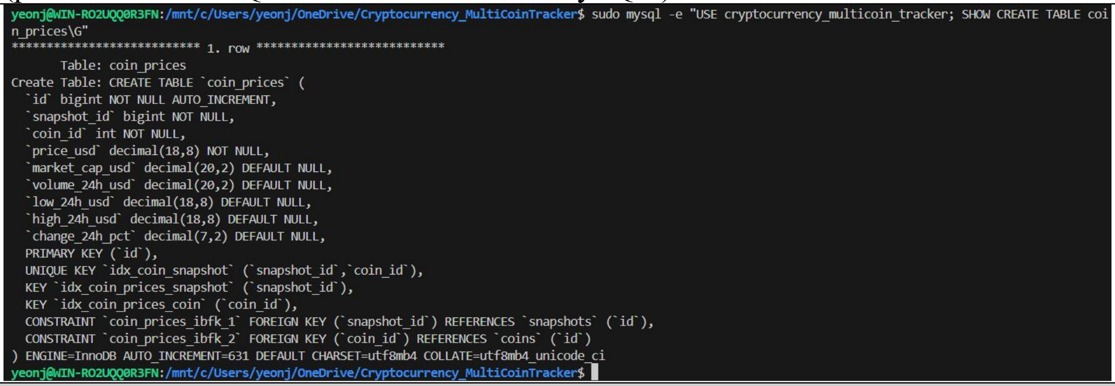
sudo mysql -e "USE cryptocurrency_multicoin_tracker; SHOW TABLES;"



```
yeonj@WIN-RO2UQQ0R3FN:/mnt/c/Users/yeonj/OneDrive/Cryptocurrency_MultiCoinTracker$ sudo mysql -e "USE cryptocurrency_multicoin_tracker; SHOW TABLES;"
[sudo] password for yeonj:
+------------------------------------------+
| Tables_in_cryptocurrency_multicoin_tracker |
+------------------------------------------+
| coin_prices                              |
| coins                                    |
| snapshots                                |
+------------------------------------------+
```

Show structure + constraints

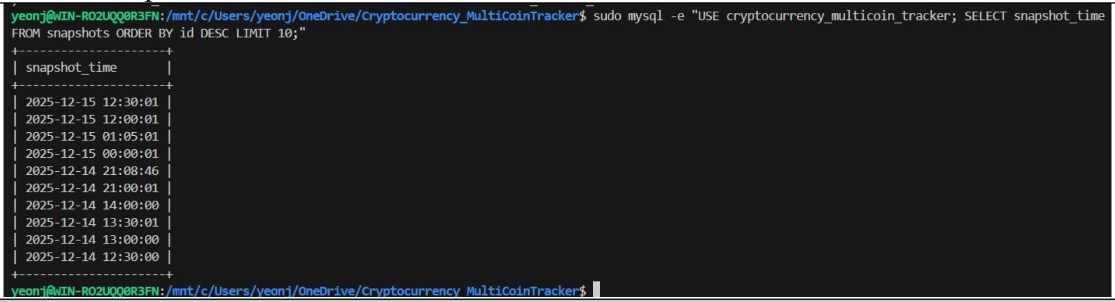
sudo mysql -e "USE cryptocurrency_multicoin_tracker; SHOW CREATE TABLE coin_prices\G"
(proves FK + UNIQUE constraint exist in MySQL.)



```
yeonj@WIN-RO2UQQ0R3FN:/mnt/c/Users/yeonj/OneDrive/Cryptocurrency_MultiCoinTracker$ sudo mysql -e "USE cryptocurrency_multicoin_tracker; SHOW CREATE TABLE coi
n_prices\G"
*************************** 1. row ***************************
       Table: coin_prices
Create Table: CREATE TABLE `coin_prices` (
  `id` bigint NOT NULL AUTO_INCREMENT,
  `snapshot_id` bigint NOT NULL,
  `coin_id` int NOT NULL,
  `price_usd` decimal(18,8) NOT NULL,
  `market_cap_usd` decimal(20,2) DEFAULT NULL,
  `volume_24h_usd` decimal(20,2) DEFAULT NULL,
  `low_24h_usd` decimal(18,8) DEFAULT NULL,
  `high_24h_usd` decimal(18,8) DEFAULT NULL,
  `change_24h_pct` decimal(7,2) DEFAULT NULL,
  PRIMARY KEY (`id`),
  UNIQUE KEY `idx_coin_snapshot` (`snapshot_id`,`coin_id`),
  KEY `idx_coin_prices_snapshot` (`snapshot_id`),
  KEY `idx_coin_prices_coin` (`coin_id`),
  CONSTRAINT `coin_prices_ibfk_1` FOREIGN KEY (`snapshot_id`) REFERENCES `snapshots` (`id`),
  CONSTRAINT `coin_prices_ibfk_2` FOREIGN KEY (`coin_id`) REFERENCES `coins` (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=631 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci
yeonj@WIN-RO2UQQ0R3FN:/mnt/c/Users/yeonj/OneDrive/Cryptocurrency_MultiCoinTracker$
```

Show historical data works

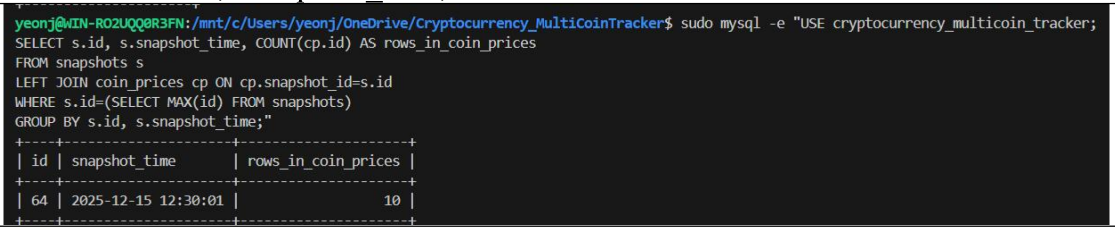
sudo mysql -e "USE cryptocurrency_multicoin_tracker; SELECT snapshot_time FROM snapshots ORDER BY id DESC LIMIT 10;"



```
yeonj@WIN-RO2UQQ0R3FN:/mnt/c/Users/yeonj/OneDrive/Cryptocurrency_MultiCoinTracker$ sudo mysql -e "USE cryptocurrency_multicoin_tracker; SELECT snapshot_time
FROM snapshots ORDER BY id DESC LIMIT 10;"
+---------------------+
| snapshot_time       |
+---------------------+
| 2025-12-15 12:30:01 |
| 2025-12-15 12:00:01 |
| 2025-12-15 01:05:01 |
| 2025-12-15 00:00:01 |
| 2025-12-14 21:08:46 |
| 2025-12-14 21:00:01 |
| 2025-12-14 14:00:00 |
| 2025-12-14 13:30:01 |
| 2025-12-14 13:00:00 |
| 2025-12-14 12:30:00 |
+---------------------+
yeonj@WIN-RO2UQQ0R3FN:/mnt/c/Users/yeonj/OneDrive/Cryptocurrency_MultiCoinTracker$
```

Show 10 rows per latest snapshot

sudo mysql -e "USE cryptocurrency_multicoin_tracker;
SELECT s.id, s.snapshot_time, COUNT(cp.id) AS rows_in_coin_prices
FROM snapshots s
LEFT JOIN coin_prices cp ON cp.snapshot_id=s.id
WHERE s.id=(SELECT MAX(id) FROM snapshots)
GROUP BY s.id, s.snapshot_time;"



```
yeonj@WIN-RO2UQQ0R3FN:/mnt/c/Users/yeonj/OneDrive/Cryptocurrency_MultiCoinTracker$ sudo mysql -e "USE cryptocurrency_multicoin_tracker;
SELECT s.id, s.snapshot_time, COUNT(cp.id) AS rows_in_coin_prices
FROM snapshots s
LEFT JOIN coin_prices cp ON cp.snapshot_id=s.id
WHERE s.id=(SELECT MAX(id) FROM snapshots)
GROUP BY s.id, s.snapshot_time;"
+----+---------------------+---------------------+
| id | snapshot_time       | rows_in_coin_prices |
+----+---------------------+---------------------+
| 64 | 2025-12-15 12:30:01 |                  10 |
+----+---------------------+---------------------+
```