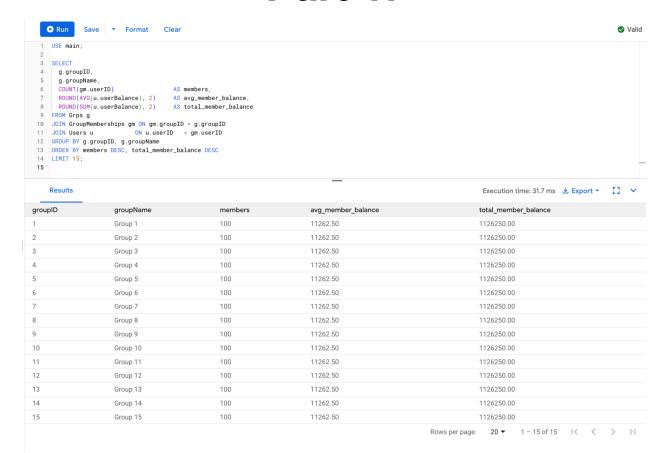
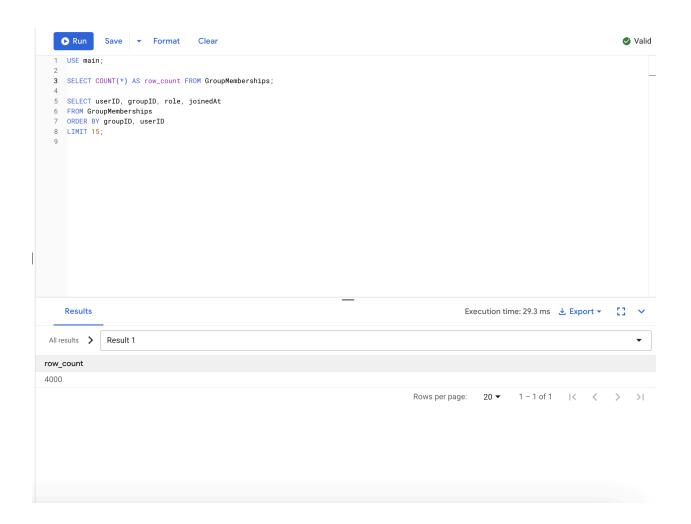
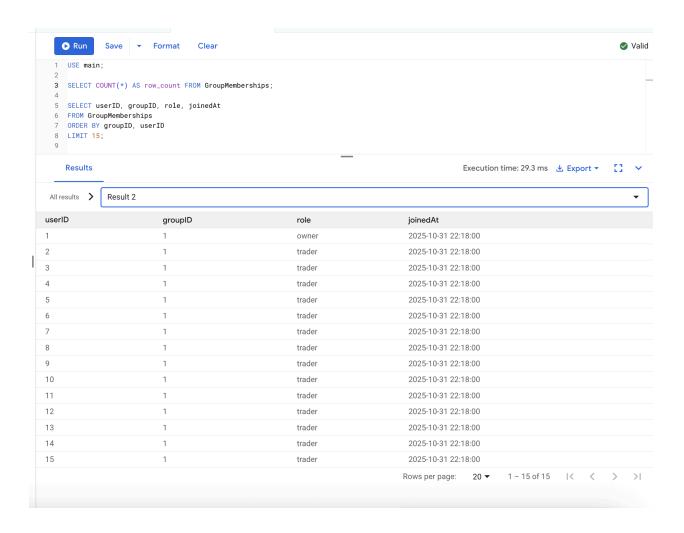
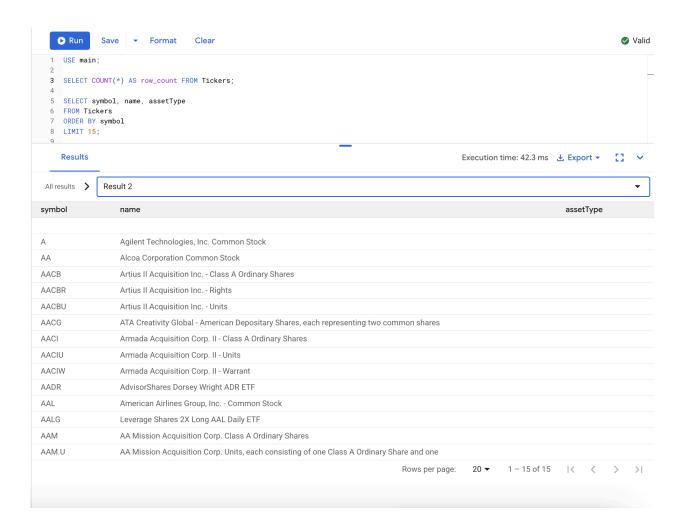
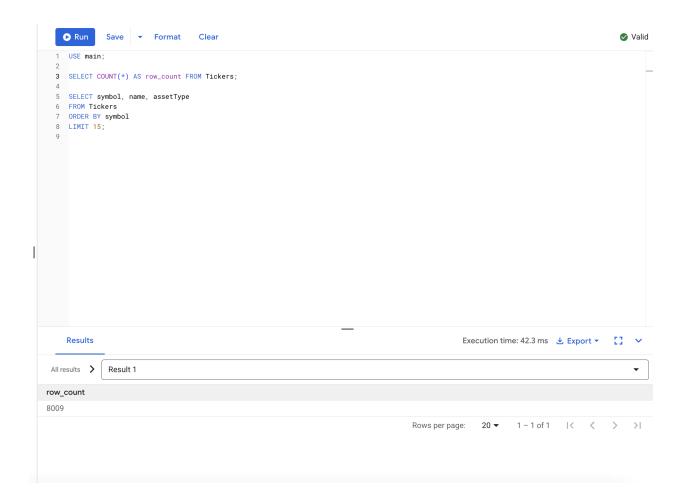
Part 1:

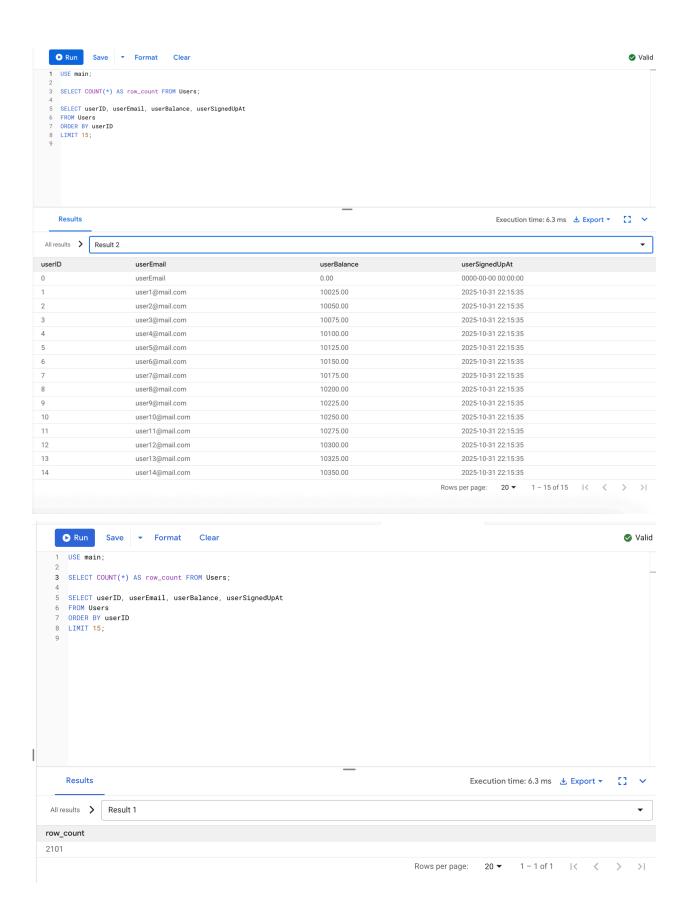












Part 2:

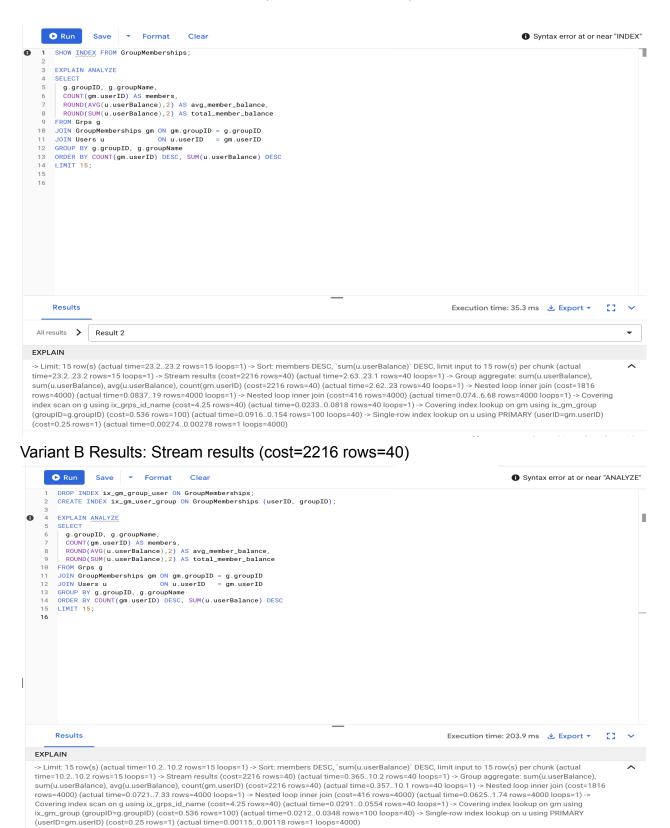
Query 1:

Variant A reduced plan cost from 4216 to 2216 by aligning the join and GROUP BY with groupID, enabling range scans over GroupMemberships. Variant B did not help because the leading column (userID) did not match the group-driven access pattern; the optimizer continued to choose the groupID path. Adding covering indexes on Users and Grps (Variant C) did not further reduce cost on our dataset. Users is small and remain cached, so index-only reads brought negligible savings. We select Variant A as the final design. Trade-off: slightly higher write cost and index storage, acceptable for our read-heavy analytics.

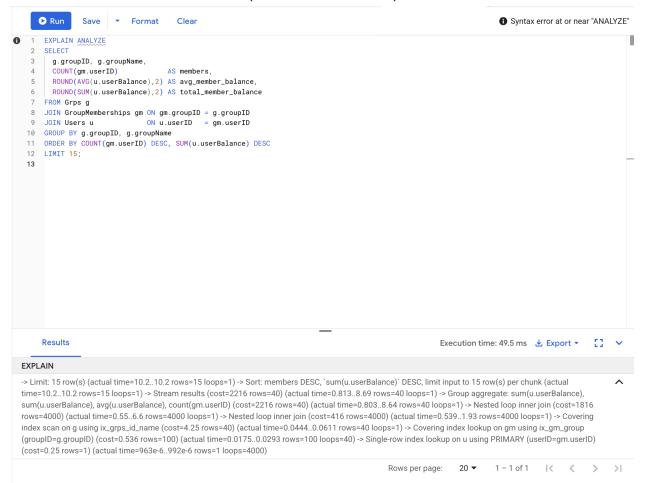
Baseline Results: Cost = 4216



Variant A Results: Stream results (cost=2216 rows=40)



Variant C Results: Stream results (cost=2216 rows=40)



Query 2:

Based on the three variants, all decreased the total cost, although to varying degrees. When we did index on close and volume (A and B), both had a similar decrease to the cost, which makes sense as both are used in aggregation (close being in the having clause in subquery, and volume being returned in the outer query). Variant C had the largest impact, which is reasonable since it combined volume (which was related to the aggregation) and time which was related to grouping.

Baseline cost: 61204



Variant A: Cost: 60982



Variant B: Cost: 60971



>> Sort: 'month', Tickers.'name' (actual time=1405..1405 rows=22 loops=1) >> Table scan on <temporary> (actual time=1405..1405 rows=22 loops=1) >> Aggregate using temporary table (actual time=1405..1405 rows=22 loops=1) >> Nested loop inner join (cost=60971 rows=522621) (actual time=1399..1404 rows=600 loops=1) >> Filter. ((Tickers.symbol in (META;'AAPL';'AMZN',NVDA;'GOOG',TSLA;'UBER;'MSFT';'ABNB;'AVGO',ORCL')) or

'in_optimizer-(Tickers.symbol, Tickers.symbol in (select #2))) (cost=916 rows=8759) (actual time=1399..1404 rows=11 loops=1) >> Table scan on Tickers (cost=916 rows=8759) (actual time=0.751..3.25 rows=8009) loops=1) >> Select #2 (subquey in condition; run only note) -> Filter. ((Tickers.symbol in rows=0.25 rows=0.951) -> Tickers.symbol in (select #2))) (cost=9858.9.8855 rows=1) (actual time=1398..1398 rows=0 loops=1) >> Index lookup on <materialized_subquery> using <auto_distinct_key> (ticker=Tickers.symbol) (actual time=1398..1398 rows=0 loops=1) >> Materialize with deduplication (cost=98858.9.8855 rows=8010) (actual time=1398..1398 rows=0 loops=1) >> Filter. (avg(PriceBars.'close') >> 135) (cost=98057 rows=8010) (actual time=1398..1398 rows=0 loops=1) >> Index lookup on PriceBars sclose') (cost=98057 rows=8010) (actual time=3.9..1289 rows=8009 loops=1) >> Index lookup on PriceBars using PRIMARY (cost=50264 rows=477931) (actual time=3.9..1289 rows=480541 loops=1) >> Index lookup on PriceBars using PRIMARY (cost=50264 rows=477931) (actual time=3.9..1289 rows=480541 loops=1) >> Index lookup on PriceBars using PRIMARY (cost=50264 rows=477931) (actual time=3.9..1289 rows=480541 loops=1) > Index lookup on PriceBars using PRIMARY (cost=50264 rows=477931) (actual time=3.9..1289 rows=480541 loops=1) > Index lookup on PriceBars using PRIMARY (cost=50264 rows=477931) (actual time=3.9..1289 rows=480541 loops=1) > Index lookup on PriceBars using PRIMARY (cost=50264 rows=477931) (actual time=3.9..1289 rows=480541 loops=1) > Index lookup on PriceBars using PRIMARY (cost=50264 rows=477931) (actual time=3.

Rows per page: 20 ▼ 1 − 1 of 1 | ⟨ ⟨ > >|

Variant C: Cost: 60810 © Run Save ▼ Format Clear

© Syntax error at or near "ANALYZE"

| #CREATE INDEX pb_ticker_time_volume ON PriceBars(ticker, time, volume);
| #EXPLAIN ANALYZE | SELECT name, MONTH(time) as month, avg(volume) as volume | FRON Tickers JOIN PriceBars ON (Tickers.symbol = PriceBars.ticker) | FRON PriceBars ON (Tickers.symbol = PriceBars.ticker) | FRON PriceBars | FRON PriceBars | FRON PriceBars | GROUP BY ticker | Ticker | FRON PriceBars | GROUP BY ticker | Ticke

EXPLAIN

Rows per page: 20 ▼ 1 − 1 of 1 |< < > >|