

GROWTHGENIE HACKATHON

Team Name	Twin Titans
Team Leader	<u>Ankit Aabad</u>
Team Members	Hitesh Kumawat, Ankit Aabad

Summary

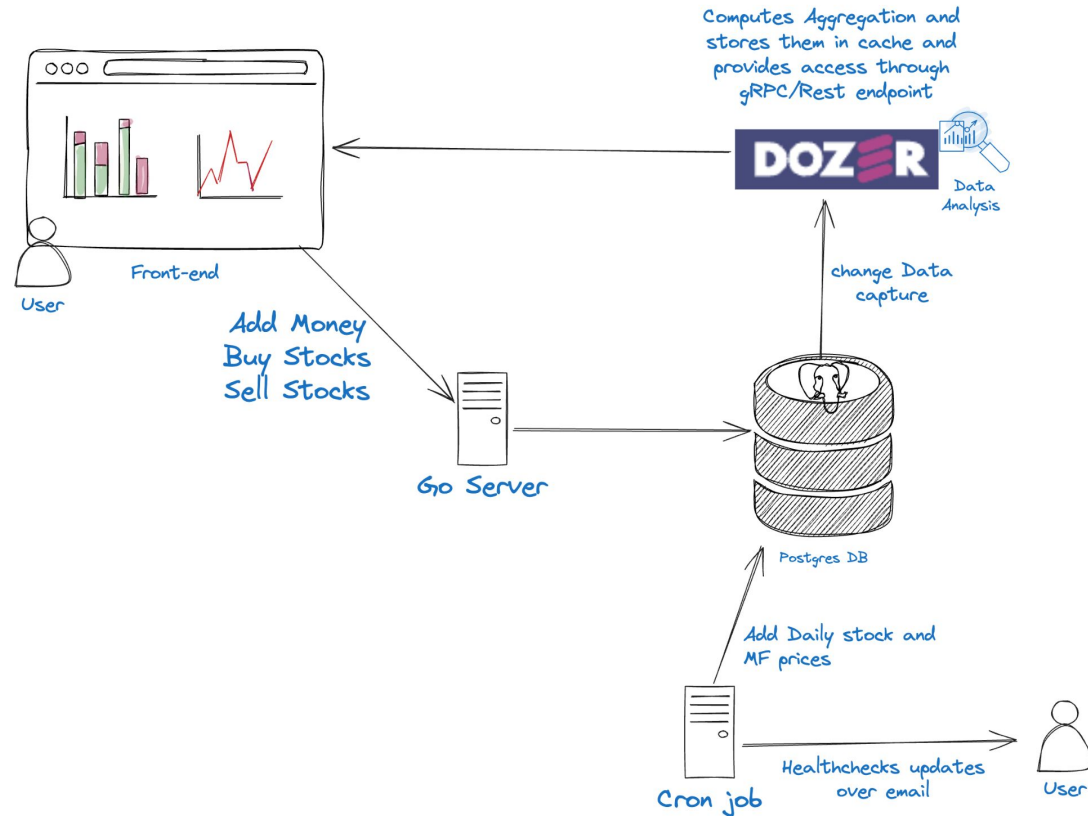
- I will be working on the Theme - 2 (Data Analytics Dashboards)



Solution

- The idea is to use an Open Source Solution called Dozer. We connect the postgres db to Dozer and dozer gets the recent data via Change Data Capture(CDC). Dozer incrementally aggregates and computes the queries and stores them in low latency LMDB cache and provides gRPC apis to access the result.
- A Golang server will provides 3 apis
 - Add Money
 - Buy Stocks
 - Sell Stocks
- Svelte will be used to create frontend and provide a dashboards to users and management.

High Level Design Diagram



Tech Stack

- Golang (Server)
- Postgres (Database)
- [Healthcheck](#) (for cron monitoring)
- [Caprover](#) (for deployment)
- [Dozer](#)
- [Svelte](#) (Frontend)
- [Webscraper.io](#)
- Docker

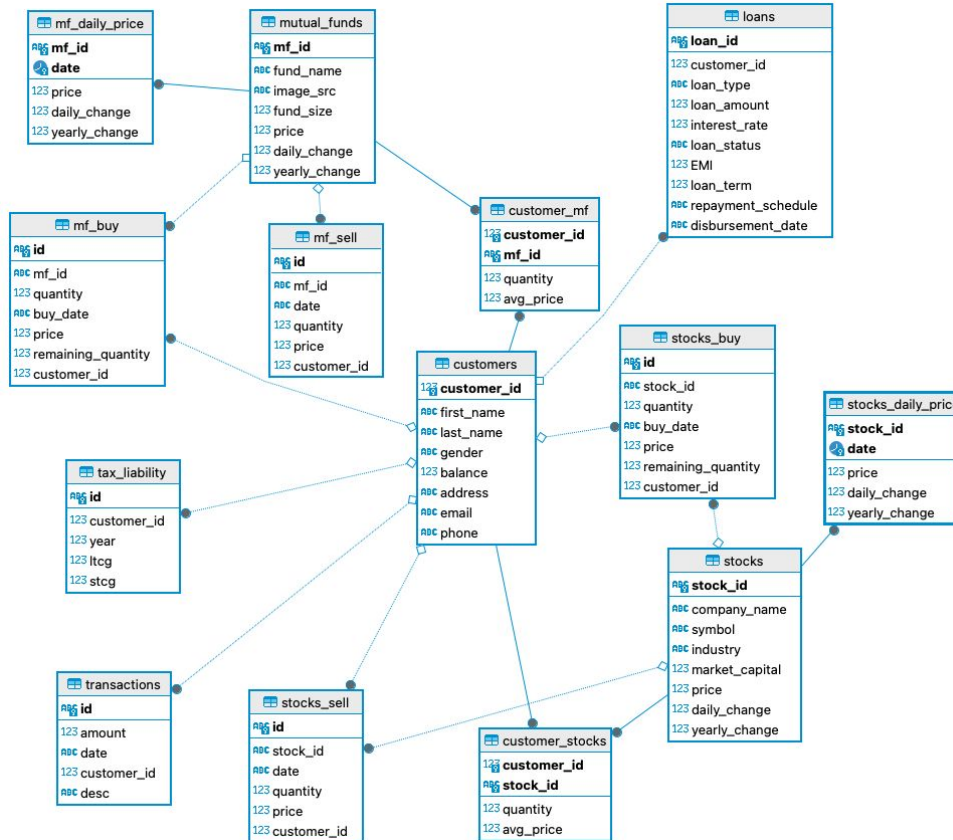


Tech Stack

- Golang (Server)
- Postgres (Database)
- [Healthcheck](#) (for cron monitoring)
- [Caprover](#) (for deployment)
- [Dozer](#)
- [Svelte](#) (Frontend)
- [Atlas](#) + HCL (Database schema management)
- Docker



Database Schema



Code Samples



Buy Stocks

```
func (c *DbConn) BuyStocks(input models.BuyStocksModel) error {
    //ctx := context.Background()
    today := time.Now().Format("2006-01-02") // today date to sql date
    todayIso := time.Now().Format(time.RFC3339)
    transactionId := ksuid.New() // a time sorted uuid
    stockBuyId := ksuid.New()
    desc := fmt.Sprintf("Bought %d stocks of %s ", input.Quantity, input.CompanyName)
    result, err := c.pool.Exec(context.Background(), "with sp as ( select price from stocks_daily_price sdp where sdp.
stock_id = $1 and date = $2),\nbalance as ( select (select balance from customers where customer_id = $3) > sp.
price* $4 as enough_balance, sp.price*$4 as amount,price from sp),\ncu as (update customers set balance = balance
- sp.price*$4 from sp where customer_id = $3 and (select enough_balance from balance) = true),\nit as (INSERT
INTO transactions (id, amount, \"date\", customer_id, \"desc\") select $5, amount, $8, $3, $6 from balance where
enough_balance = true),\nisbuy as (INSERT INTO stocks_buy (id, stock_id, quantity, buy_date, price,
remaining_quantity,customer_id) select $7, $1, $4, $2, price, $4,$3 from balance where enough_balance = true)
\nINSERT INTO customer_stocks as cs (customer_id, stock_id, quantity, avg_price) select $3, $1, $4, price from
balance where enough_balance = true ON CONFLICT (customer_id,stock_id) DO update set quantity = cs.quantity + $4,
avg_price = (cs.quantity*cs.avg_price + $4* (select price from balance))/(cs.quantity+ $4) where cs.stock_id = $1
and cs.customer_id= $3 and (select enough_balance from balance) = true;", input.StockId, today, input.CustomerId,
input.Quantity, transactionId, desc, stockBuyId, todayIso)
    fmt.Println("printing result ", result)

    return err
}
```

Team Snippet

Ankit Aabad: I am a software engineer working in Identity and Access Management space. I am currently working at Loginradius. My skills include Golang, NodeJS, AWS, OpenTelemetry, Postgres, MongoDB, Serverless etc.

These days I spend my free time in learning 3d printing and custom ergonomic keyboards.

Team Snippet

Hitesh Kumawat: I am a Product Designer working in Identity and Access Management space. I am currently working at Loginradius. My skills include HTML, CSS, JavaScript, React, Adobe Software, Figma, etc.

I enjoy developing free materials that help creative people in their process or journey.

Additional Points

- The docker compose up will set up the initial db with given data in excel sheets.
- There will be no apis to insert data. One can directly connect with postgres using visual client like dBeaver/pgAdmin to insert/update data and see how the analytics change.



Thank
you !