A
Project Report
on

TrackMyStock

Developed by

RUSHI RAVAL(IT-135) – Department of IT, DD University

Guided By
Internal Guide:
Prof. Archana N. Vyas
Department of Information Technology
Faculty of Technology
DD University

Department of Information Technology
Faculty of Technology, Dharmsinh Desai University
College Road, Nadiad-387001
October-2021

DHARMSINH DESAI UNIVERSITY
NADIAD-387001, GUJARAT

CERTIFICATE

This is to certify that the project entitled "**TrackMyStock**" is a bonafied report of the work carried out by

1) **Mr. RUSHI RAVAL** , Student ID No : **20ITUOD008**

of Department of Information Technology, semester V, under the guidance and supervision for the subject Database Management System. He was involved in Project training during academic year 2021-2022.

Prof. Archana N. Vyas
(Project Guide)
Department of Information Technology,
Faculty of Technology,
Dharmsinh Desai University, Nadiad
Date:

Prof. Vipul Dabhi
Head , Department of Information Technology,
Faculty of Technology,
Dharmsinh Desai University, Nadiad
Date:

ACKNOWLEDGEMENT

We would like to give our sincere acknowledgement to everybody responsible for the successful completion of our project “ TrackMyStock ”.

The success and final outcome of this project required a lot of guidance and assistance from many people and we are extremely privileged to have got this all along the completion of this project.

We owe our deep gratitude to our project guide Prof. Archana N. Vyas, who took been interest on our project work and guided us all along till the completion of our project work by providing all the necessary help for developing a good Database System.

We would also like to thank all our lecturers.

Finally we convoy our acknowledgement to all our friends and family members who directly or indirectly associated with us in the successful completion of the project. We thank one and all.

TABLE OF CONTENTS

1. SYSTEM OVERVIEW

1.1 CURRENT SYSTEM

In the current system, new user find very complex to understand the logic of it like there are various share market terminologies that a newbie cant understand, our project is a straight forward easy to use system that is used to track your stock.

Indians are very fond of share market. India’s NSE and BSE has the largest market cap in among various countries over the world. Every Indian in his/her life thinks about investing in sharemarket but due to the high level terminology used in they feel like stereotype, so our goal is to provide them a simple and sober tracking system.

1.2 OBJECTIVES OF THE PROPOSED SYSTEM

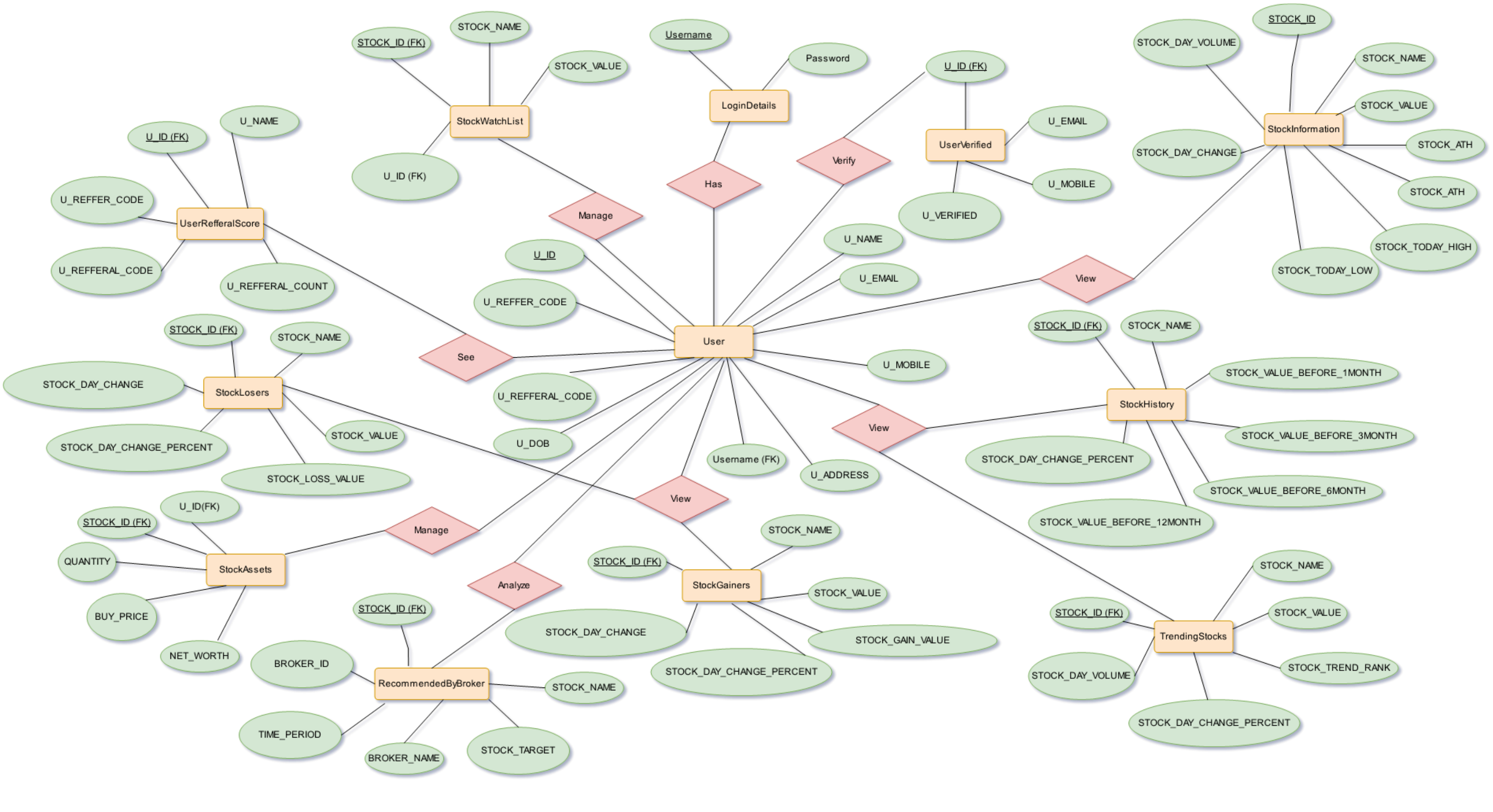
- Login and create their user profile.
- Add their bought shares in the SHAREASSSETS with respective data table so he/she can track their portfolio.
- Create a watchlist to see their favorite shares in another page rather than searching it.
- View the trending stocks, 24 hour gainers and 24 hour losers.
- We also implemented a referral system so we can reward the user who reffered someone else.

1.3 ADVANTAGES OF THE PROPOSED) SYSTEM

Through this system we try to achieve the below objectives

- To make user comfortable through simple terminologies.
- To let users track their watchlist rather than searching throut whole market.
- To allow users to refer someone else and reward them.
- To show users recommended stock which are recommended by various brokers.
- To provide various information about the share they search like their price, ATH(All Time High), ATL(All Time Low), today’s high, today’s low, day change percent, day volume.

2.E-R DIAGRAM

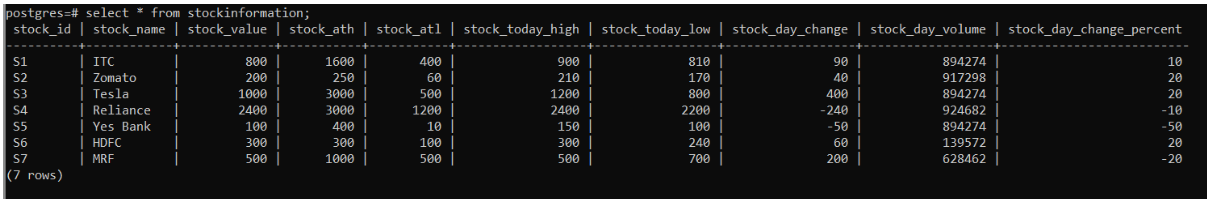


TrackMyStock - ER Diagram

Viewer does not support full SVG 1.1

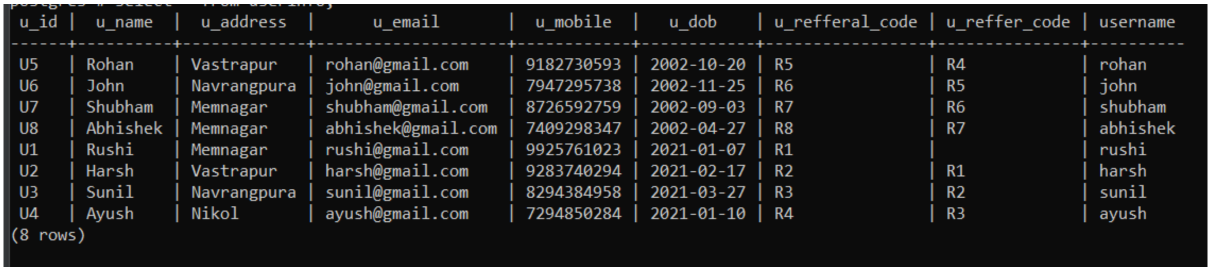
3. DATA DICTIONARY

3.1 StockInformation:



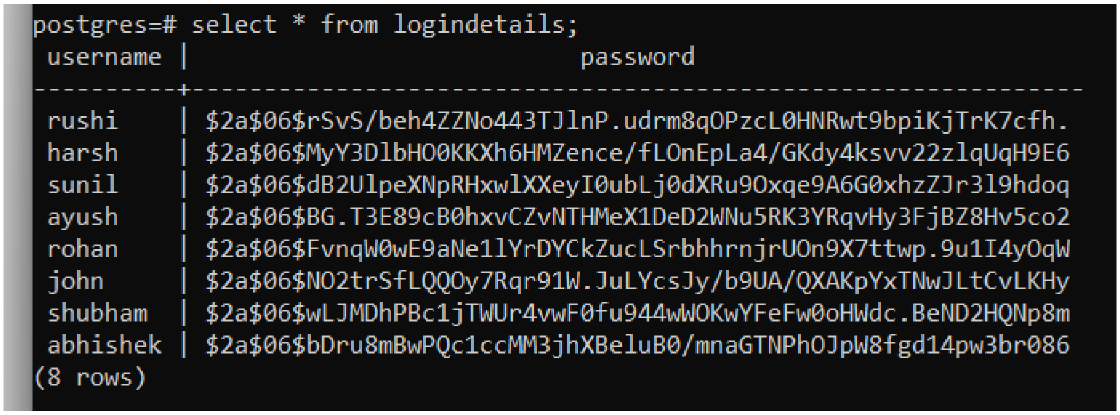
```
postgres=# select * from stockinformation;
 stock_id | stock_name | stock_value | stock_ath | stock_atl | stock_today_high | stock_today_low | stock_day_change | stock_day_volume | stock_day_change_percent
----------+------------+-------------+-----------+-----------+------------------+-----------------+------------------+------------------+--------------------------
 S1       | ITC        |         800 |      1600 |       400 |              900 |             810 |               90 |           894274 |                       10
 S2       | Zomato     |         200 |       250 |        60 |              210 |             170 |               40 |           917298 |                       20
 S3       | Tesla      |        1000 |      3000 |       500 |             1200 |             800 |              400 |           894274 |                       20
 S4       | Reliance   |        2400 |      3000 |      1200 |             2400 |            2200 |             -240 |           924682 |                      -10
 S5       | Yes Bank   |         100 |       400 |        10 |              150 |             100 |              -50 |           894274 |                      -50
 S6       | HDFC       |         300 |       300 |       100 |              300 |             240 |               60 |           139572 |                       20
 S7       | MRF        |         500 |      1000 |       500 |              500 |             700 |              200 |           628462 |                      -20
(7 rows)
```

3.2 UserInfo:



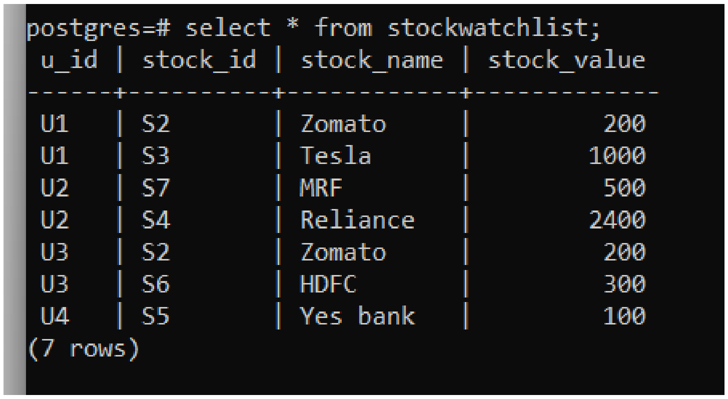
```
postgres=# select * from userinfo;
 u_id |  u_name  |  u_address  |      u_email       |  u_mobile  |   u_dob    | u_refferal_code | u_reffer_code | username
------+----------+-------------+--------------------+------------+------------+-----------------+---------------+----------
 U5   | Rohan    | Vastrapur   | rohan@gmail.com    | 9182730593 | 2002-10-20 | R5              | R4            | rohan
 U6   | John     | Navrangpura | john@gmail.com     | 7947295738 | 2002-11-25 | R6              | R5            | john
 U7   | Shubham  | Memnagar    | shubham@gmail.com  | 8726592759 | 2002-09-03 | R7              | R6            | shubham
 U8   | Abhishek | Memnagar    | abhishek@gmail.com | 7409298347 | 2002-04-27 | R8              | R7            | abhishek
 U1   | Rushi    | Memnagar    | rushi@gmail.com    | 9925761023 | 2021-01-07 | R1              |               | rushi
 U2   | Harsh    | Vastrapur   | harsh@gmail.com    | 9283740294 | 2021-02-17 | R2              | R1            | harsh
 U3   | Sunil    | Navrangpura | sunil@gmail.com    | 8294384958 | 2021-03-27 | R3              | R2            | sunil
 U4   | Ayush    | Nikol       | ayush@gmail.com    | 7294850284 | 2021-01-10 | R4              | R3            | ayush
(8 rows)
```

3.3 LoginDetails:



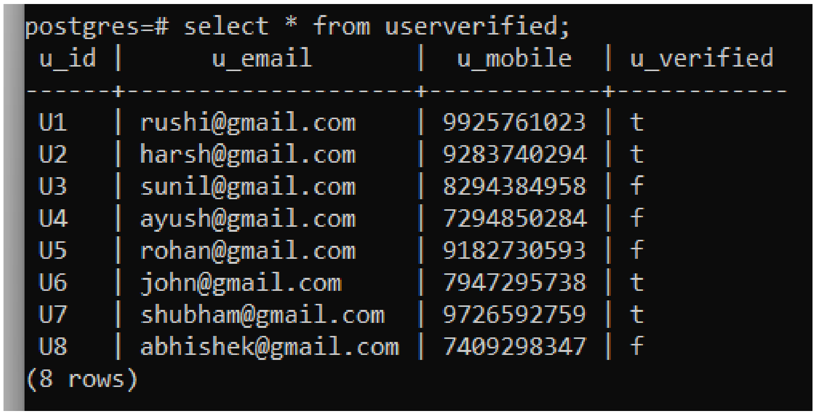
```
postgres=# select * from logindetails;
 username |                           password
----------+--------------------------------------------------------------
 rushi    | $2a$06$rSvS/beh4ZZNo443TJlnP.udrm8qOPzcL0HNRwt9bpiKjTrK7cfh.
 harsh    | $2a$06$MyY3DlbHO0KKXh6HMZence/fLOnEpLa4/GKdy4ksvv22zlqUqH9E6
 sunil    | $2a$06$dB2UlpeXNpRHxwlXXeyI0ubLj0dXRu9Oxqe9A6G0xhzZJr3l9hdoq
 ayush    | $2a$06$BG.T3E89cB0hxvCZvNTHMeX1DeD2WNu5RK3YRqvHy3FjBZ8Hv5co2
 rohan    | $2a$06$FvnqW0wE9aNe1lYrDYCkZucLSrbhhrnjrUOn9X7ttwp.9u1I4yOqW
 john     | $2a$06$NO2trSfLQQOy7Rqr91W.JuLYcsJy/b9UA/QXAKpYxTNwJLtCvLKHy
 shubham  | $2a$06$wLJMDhPBc1jTWUr4vwF0fu944wWOKwYFeFw0oHWdc.BeND2HQNp8m
 abhishek | $2a$06$bDru8mBwPQc1ccMM3jhXBeluB0/mnaGTNPhOJpW8fgd14pw3br086
(8 rows)
```

3.4 StockWatchlist:



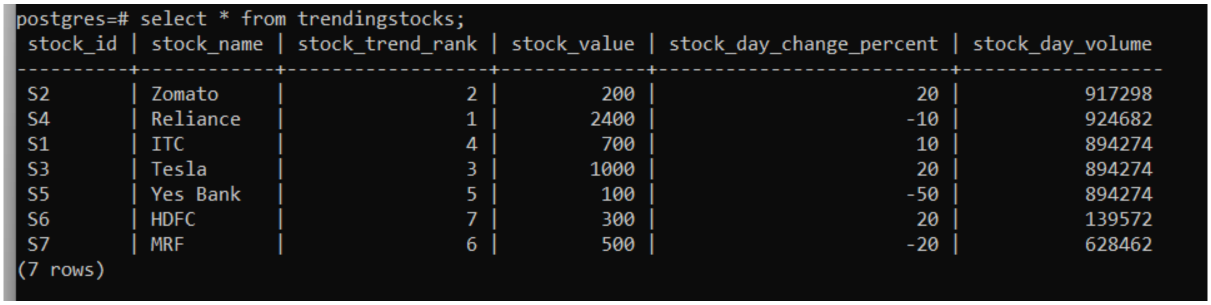
```
postgres=# select * from stockwatchlist;
 u_id | stock_id | stock_name | stock_value
------+----------+------------+-------------
 U1   | S2       | Zomato     |         200
 U1   | S3       | Tesla      |        1000
 U2   | S7       | MRF        |         500
 U2   | S4       | Reliance   |        2400
 U3   | S2       | Zomato     |         200
 U3   | S6       | HDFC       |         300
 U4   | S5       | Yes bank   |         100
(7 rows)
```

3.5 UserVerified:



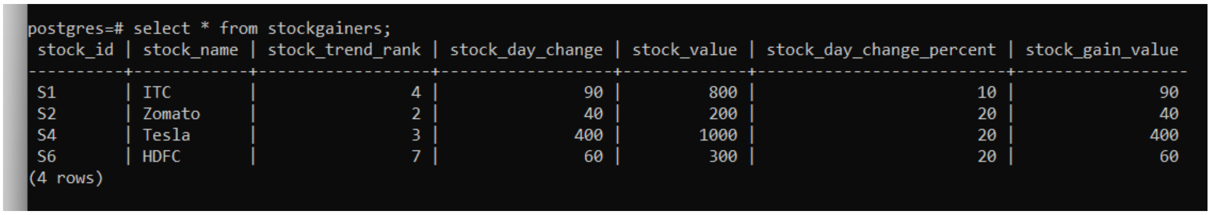
```
postgres=# select * from userverified;
 u_id |      u_email       |  u_mobile  | u_verified
------+--------------------+------------+------------
 U1   | rushi@gmail.com    | 9925761023 | t
 U2   | harsh@gmail.com    | 9283740294 | t
 U3   | sunil@gmail.com    | 8294384958 | f
 U4   | ayush@gmail.com    | 7294850284 | f
 U5   | rohan@gmail.com    | 9182730593 | f
 U6   | john@gmail.com     | 7947295738 | t
 U7   | shubham@gmail.com  | 9726592759 | t
 U8   | abhishek@gmail.com | 7409298347 | f
(8 rows)
```

3.6 TrendingStocks:



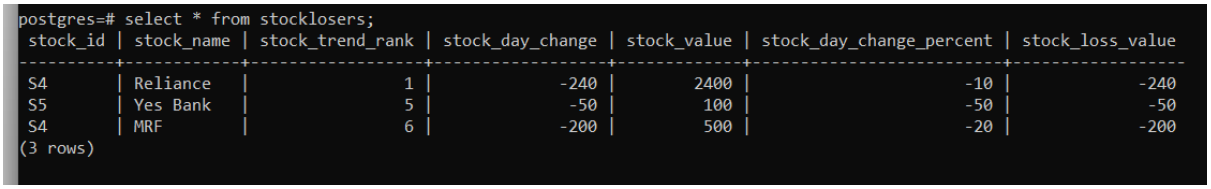
```
postgres=# select * from trendingstocks;
 stock_id | stock_name | stock_trend_rank | stock_value | stock_day_change_percent | stock_day_volume
----------+------------+------------------+-------------+--------------------------+------------------
 S2       | Zomato     |                2 |         200 |                       20 |           917298
 S4       | Reliance   |                1 |        2400 |                      -10 |           924682
 S1       | ITC        |                4 |         700 |                       10 |           894274
 S3       | Tesla      |                3 |        1000 |                       20 |           894274
 S5       | Yes Bank   |                5 |         100 |                      -50 |           894274
 S6       | HDFC       |                7 |         300 |                       20 |           139572
 S7       | MRF        |                6 |         500 |                      -20 |           628462
(7 rows)
```

3.7 StockGainers :



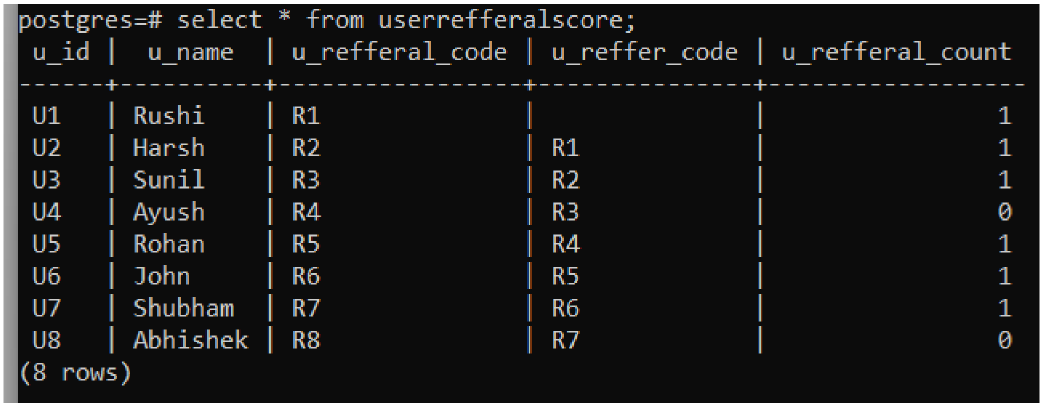
```
postgres=# select * from stockgainers;
 stock_id | stock_name | stock_trend_rank | stock_day_change | stock_value | stock_day_change_percent | stock_gain_value
----------+------------+------------------+------------------+-------------+--------------------------+------------------
 S1       | ITC        |                4 |               90 |         800 |                       10 |               90
 S2       | Zomato     |                2 |               40 |         200 |                       20 |               40
 S4       | Tesla      |                3 |              400 |        1000 |                       20 |              400
 S6       | HDFC       |                7 |               60 |         300 |                       20 |               60
(4 rows)
```

3.8 StockLosers :



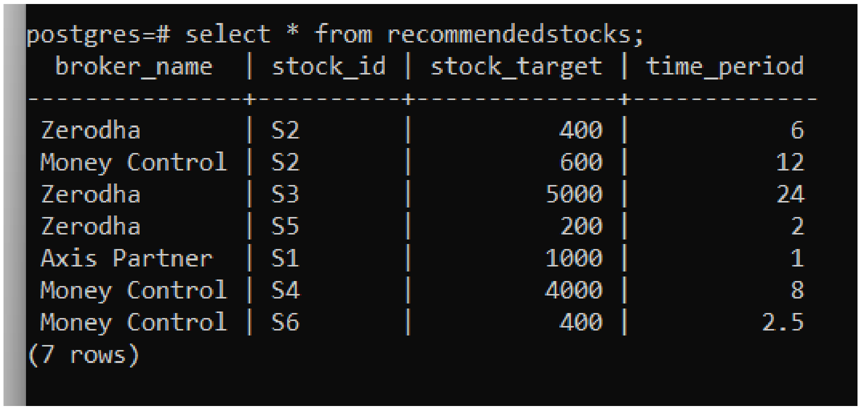
```
postgres=# select * from stocklosers;
 stock_id | stock_name | stock_trend_rank | stock_day_change | stock_value | stock_day_change_percent | stock_loss_value
----------+------------+------------------+------------------+-------------+--------------------------+------------------
 S4       | Reliance   |                1 |             -240 |        2400 |                      -10 |             -240
 S5       | Yes Bank   |                5 |              -50 |         100 |                      -50 |              -50
 S4       | MRF        |                6 |             -200 |         500 |                      -20 |             -200
(3 rows)
```

3.9 UserRefferalScore :



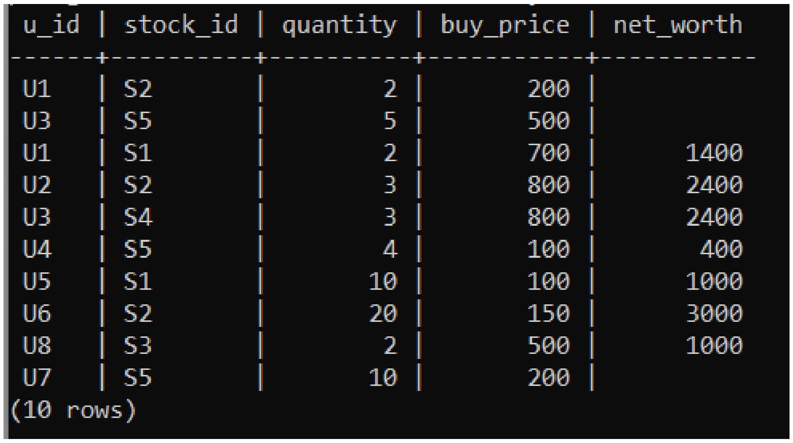
```
postgres=# select * from userrefferalscore;
 u_id |  u_name   | u_refferal_code | u_reffer_code | u_refferal_count
------+-----------+-----------------+---------------+------------------
 U1   | Rushi     | R1              |               |                1
 U2   | Harsh     | R2              | R1            |                1
 U3   | Sunil     | R3              | R2            |                1
 U4   | Ayush     | R4              | R3            |                0
 U5   | Rohan     | R5              | R4            |                1
 U6   | John      | R6              | R5            |                1
 U7   | Shubham   | R7              | R6            |                1
 U8   | Abhishek  | R8              | R7            |                0
(8 rows)
```

3.10 RecommenedStocks :



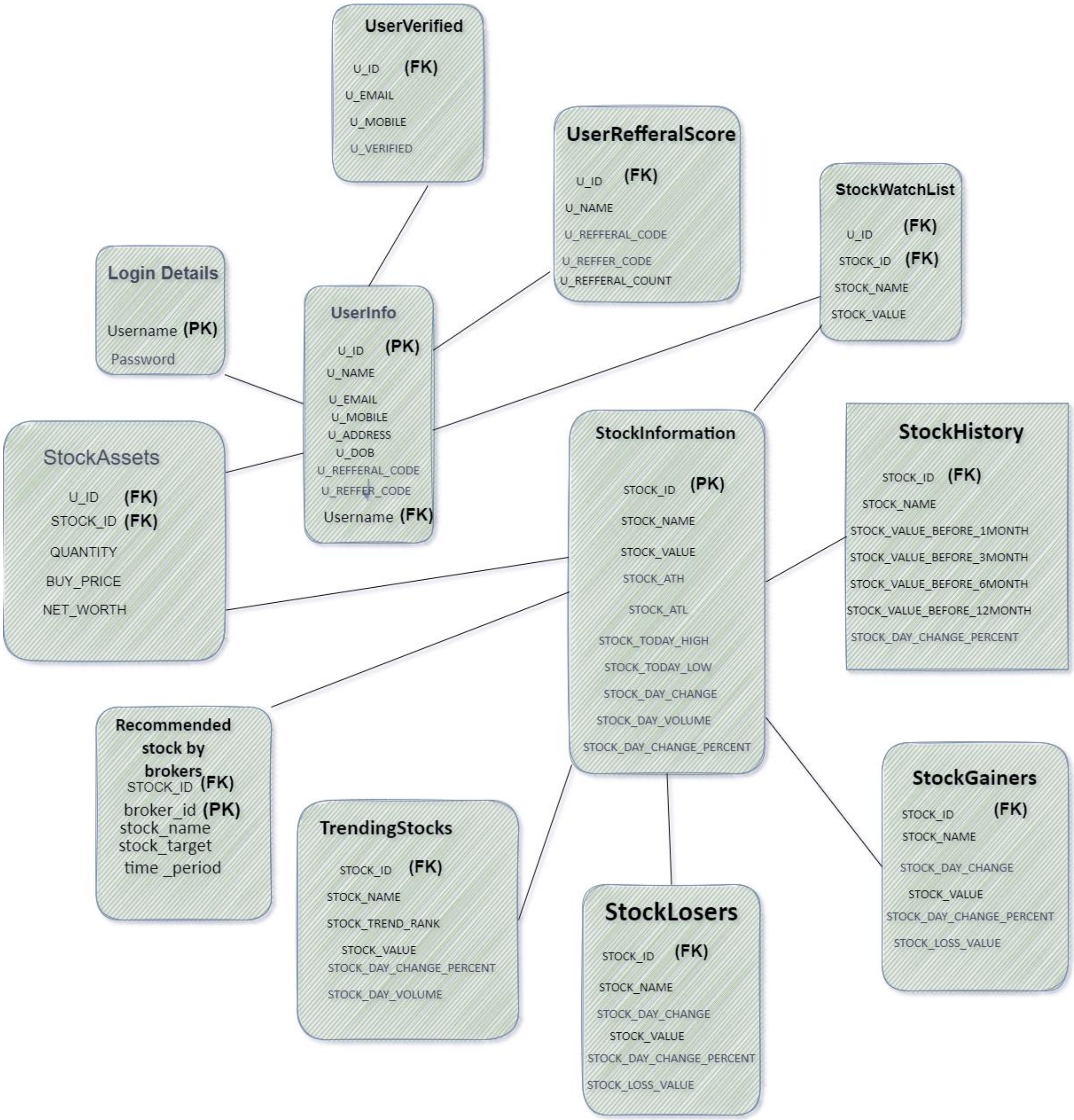
```
postgres=# select * from recommendedstocks;
  broker_name  | stock_id | stock_target | time_period
---------------+----------+--------------+-------------
 Zerodha       | S2       |          400 |           6
 Money Control | S2       |          600 |          12
 Zerodha       | S3       |         5000 |          24
 Zerodha       | S5       |          200 |           2
 Axis Partner  | S1       |         1000 |           1
 Money Control | S4       |         4000 |           8
 Money Control | S6       |          400 |         2.5
(7 rows)
```

3.11 stockassets:



```
 u_id | stock_id | quantity | buy_price | net_worth
------+----------+----------+-----------+-----------
 U1   | S2       |        2 |       200 |
 U3   | S5       |        5 |       500 |
 U1   | S1       |        2 |       700 |      1400
 U2   | S2       |        3 |       800 |      2400
 U3   | S4       |        3 |       800 |      2400
 U4   | S5       |        4 |       100 |       400
 U5   | S1       |       10 |       100 |      1000
 U6   | S2       |       20 |       150 |      3000
 U8   | S3       |        2 |       500 |      1000
 U7   | S5       |       10 |       200 |
(10 rows)
```

4. SCHEMA DIAGRAM



5. DATABASE IMPLEMENTATION

5.1 CREATE SCHEMA

5.1.1 USERINFO

```
CREATE TABLE USERINFO(
U_ID VARCHAR,
U_NAME VARCHAR,
U_ADDRESS VARCHAR,
U_EMAIL VARCHAR,
U_MOBILE NUMERIC,
U_DOB DATE,
U_REFFERAL_CODE VARCHAR,
U_REFFER_CODE VARCHAR,
USERNAME VARCHAR,
CONSTRAINT U_FK FOREIGN KEY(USERNAME) REFERENCES
LOGINDETAILS(USERNAME),
CONSTRAINT U_PK PRIMARY KEY(U_ID));
```

5.1.2 LOGINDETAILS

```
CREATE TABLE LOGINDETAILS(
USERNAME VARCHAR,
PASSWORD VARCHAR,
CONSTRAINT L_PK PRIMARY KEY(USERNAME));
```

5.1.3 STOCKINFORMATION

```
CREATE TABLE STOCKINFORMATION(
STOCK_ID VARCHAR,
STOCK_NAME VARCHAR,
STOCK_VALUE NUMERIC,
STOCK_ATH NUMERIC,
STOCK_ATL NUMERIC,
STOCK_TODAY_HIGH NUMERIC,
STOCK_TODAY_LOW NUMERIC,
STOCK_DAY_CHANGE NUMERIC,
STOCK_DAY_VOLUME NUMERIC,
STOCK_DAY_CHANGE_PERCENT NUMERIC,
CONSTRAINT S_PK PRIMARY KEY(STOCK_ID));
```

5.1.4 STOCKASSETS

```
CREATE TABLE STOCKASSETS(
  u_id VARCHAR(40) ,
  STOCK_ID VARCHAR(40) NOT NULL,
  quantity INT NOT NULL,
  buy_price INT NOT NULL,
  net_worth INT,
  CONSTRAINT SI_FK FOREIGN KEY(STOCK_ID) REFERENCES
STOCKINFORMATION(STOCK_ID),
  CONSTRAINT UI_FK FOREIGN KEY(U_ID) REFERENCES
USERINFO(U_ID));
```

5.1.5 STOCKWATCHLIST

```
CREATE TABLE STOCKWATCHLIST(
U_ID VARCHAR,
STOCK_ID VARCHAR,
STOCK_NAME VARCHAR,
STOCK_VALUE NUMERIC,
CONSTRAINT U_FK FOREIGN KEY (U_ID) REFERENCES
USERINFO(U_ID),
CONSTRAINT S_FK FOREIGN KEY (STOCK_ID) REFERENCES
STOCKINFORMATION(STOCK_ID));
```

5.1.6 STOCKHISTORY

```
CREATE TABLE STOCKHISTORY(
STOCK_ID VARCHAR,
STOCK_NAME VARCHAR,
STOCK_VALUE_BEFORE_1MONTH NUMERIC,
STOCK_VALUE_BEFORE_3MONTH NUMERIC,
STOCK_VALUE_BEFORE_6MONTH NUMERIC,
STOCK_VALUE_BEFORE_12MONTH NUMERIC,
STOCK_DAY_CHANGE_PERCENT NUMERIC,
CONSTRAINT S_FK FOREIGN KEY(STOCK_ID) REFERENCES
STOCKINFORMATION(STOCK_ID));
```

5.1.7 USERVERIFIED

```
CREATE TABLE USERVERIFIED(
U_ID VARCHAR,
U_EMAIL VARCHAR,
U_MOBILE NUMERIC,
U_VERIFIED BOOLEAN,
CONSTRAINT U_FK FOREIGN KEY(U_ID) REFERENCES
USERINFO(U_ID));
```

5.1.8 TRENDINGSTOCKS

```
CREATE TABLE TRENDINGSTOCKS(
STOCK_ID VARCHAR,
STOCK_NAME VARCHAR,
STOCK_TREND_RANK NUMERIC UNIQUE,
STOCK_VALUE NUMERIC,
STOCK_DAY_CHANGE_PERCENT NUMERIC,
STOCK_DAY_VOLUME NUMERIC,
CONSTRAINT S_FK FOREIGN KEY(STOCK_ID) REFERENCES
STOCKINFORMATION(STOCK_ID));
```

5.1.9 STOCKGAINERS

```
CREATE TABLE STOCKGAINERS(
STOCK_ID VARCHAR,
STOCK_NAME VARCHAR,
STOCK_TREND_RANK NUMERIC UNIQUE,
STOCK_DAY_CHANGE NUMERIC,
STOCK_VALUE NUMERIC,
STOCK_DAY_CHANGE_PERCENT NUMERIC,
STOCK_GAIN_VALUE NUMERIC,
CONSTRAINT S_FK FOREIGN KEY(STOCK_ID) REFERENCES
STOCKINFORMATION(STOCK_ID));
```

5.1.10 STOCKLOSERS

```
CREATE TABLE STOCKLOSERS(STOCK_ID VARCHAR,
STOCK_NAME VARCHAR,
STOCK_TREND_RANK NUMERIC UNIQUE,
STOCK_DAY_CHANGE NUMERIC,
STOCK_VALUE NUMERIC,
STOCK_DAY_CHANGE_PERCENT NUMERIC,
STOCK_LOSS_VALUE NUMERIC,
CONSTRAINT S_FK FOREIGN KEY(STOCK_ID) REFERENCES
STOCKINFORMATION(STOCK_ID));
```

5.1.11 USERREFFERALSCORE

```
CREATE TABLE USERREFFERALSCORE(
U_ID VARCHAR,
U_NAME VARCHAR,
U_REFFERAL_CODE VARCHAR,
U_REFFER_CODE  VARCHAR,
U_REFFERAL_COUNT NUMERIC DEFAULT 0,
CONSTRAINT U_FK FOREIGN KEY(U_ID) REFERENCES
USERINFO(U_ID),
CONSTRAINT U_R_FK FOREIGN KEY(U_REFFERAL_CODE)
REFERENCES USERINFO(U_REFFERAL_CODE),
CONSTRAINT U_RR_FK FOREIGN KEY(U_REFFER_CODE) REFERENCES
USERINFO(U_REFFER_CODE));
```

5.1.12 RECOMMENDEDSTOCKS

```
CREATE TABLE RECOMMENDEDSTOCKS(
BROKER_NAME VARCHAR,
STOCK_ID VARCHAR,
STOCK_TARGET NUMERIC,
TIME_PERIOD NUMERIC,
CONSTRAINT S_FK FOREIGN KEY(STOCK_ID) REFERENCES
STOCKINFORMATION(STOCK_ID));
```

5.2 INSERT DATA VALUE

5.2.1 USERINFO

insert into userinfo values('U1','Rushi','Memnagar','rushi@gmail.com',9925761023,'07-JAN-2021','R1',NULL,’rushi’);

insert into userinfo values('U2','Harsh','Vastrapur','harsh@gmail.com',9283740294,'17-FEB-2021','R2','R1',’harsh’);

insert into userinfo values('U3','Sunil','Navrangpura','sunil@gmail.com',8294384958,'27-MAR-2021','R3','R2',’sunil’);

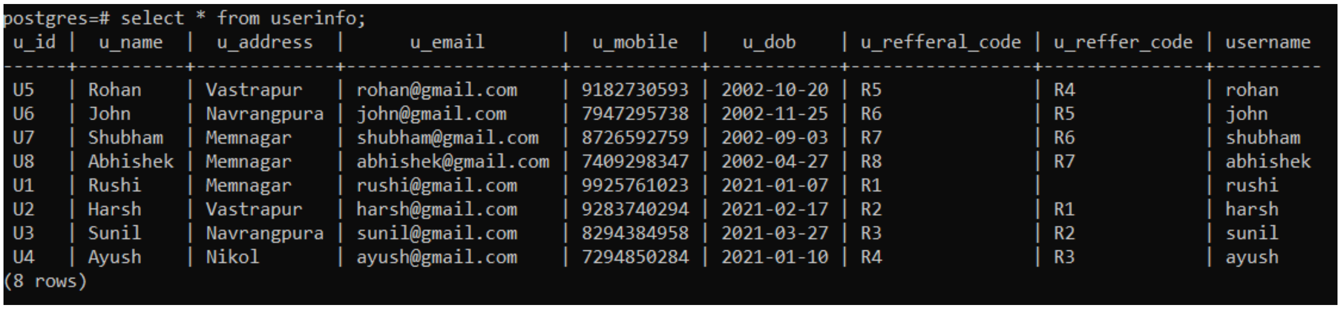
insert into userinfo values('U4','Ayush','Nikol','ayush@gmail.com',7294850284,'10-JAN-2021','R4','R3',’ayush’);

insert into userinfo values('U5','Rohan',Vastrapur',' rohan@gmail.com', 9182730593,' 20-10-2002','R5','R4',’rohan’);

insert into userinfo values('U6','John','Navrangpura','john@gmail.com', 7947295738,'25-11-2002','R6','R5',’john’);

insert into userinfo values('U7','Shubham','Memnagar','shubham@gmail.com', 8726592759,'03-09-2002','R7','R6',’shubham’);

insert into userinfo values('U8','Abhishek','Nikol','abhishek@gmail.com', 7409298347,'27-04-2002','R8','R7',’abhishek’);



```
postgres=# select * from userinfo;
 u_id |  u_name  |  u_address  |       u_email        |  u_mobile  |   u_dob    | u_refferal_code | u_reffer_code | username
------+----------+-------------+----------------------+------------+------------+-----------------+---------------+----------
 U5   | Rohan    | Vastrapur   | rohan@gmail.com      | 9182730593 | 2002-10-20 | R5              | R4            | rohan
 U6   | John     | Navrangpura | john@gmail.com       | 7947295738 | 2002-11-25 | R6              | R5            | john
 U7   | Shubham  | Memnagar    | shubham@gmail.com    | 8726592759 | 2002-09-03 | R7              | R6            | shubham
 U8   | Abhishek | Memnagar    | abhishek@gmail.com   | 7409298347 | 2002-04-27 | R8              | R7            | abhishek
 U1   | Rushi    | Memnagar    | rushi@gmail.com      | 9925761023 | 2021-01-07 | R1              |               | rushi
 U2   | Harsh    | Vastrapur   | harsh@gmail.com      | 9283740294 | 2021-02-17 | R2              | R1            | harsh
 U3   | Sunil    | Navrangpura | sunil@gmail.com      | 8294384958 | 2021-03-27 | R3              | R2            | sunil
 U4   | Ayush    | Nikol       | ayush@gmail.com      | 7294850284 | 2021-01-10 | R4              | R3            | ayush
(8 rows)
```

5.2.2 STOCKINFORMATION

INSERT INTO STOCKINFORMATION
VALUES('S1','ITC','800',1600,400,900,810,90,894274,10);

INSERT INTO STOCKINFORMATION
VALUES('S2','Zomato','200',250,60,210,170,40,917298,20);

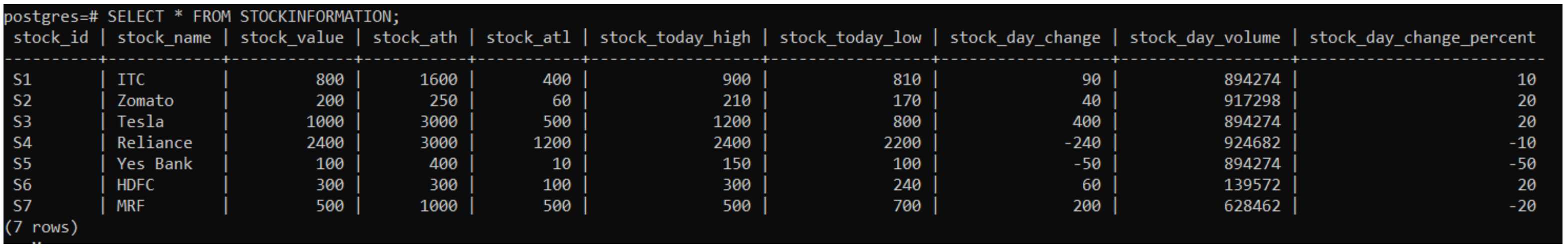
INSERT INTO STOCKINFORMATION
VALUES('S3','Tesla','1000',3000,500,1200,800,400,894274,20);

INSERT INTO STOCKINFORMATION
VALUES('S4','Reliance','2400',3000,1200,2400,2200,-240,924682,-10);

INSERT INTO STOCKINFORMATION VALUES('S5','Yes Bank','100',400,10,150,100,-50,894274,-50);

INSERT INTO STOCKINFORMATION
VALUES('S6','HDFC','300',300,100,300,240,60,139572,20);

INSERT INTO STOCKINFORMATION
VALUES('S7','MRF','500',1000,500,500,700,200,628462,-20);



```
postgres=# SELECT * FROM STOCKINFORMATION;
 stock_id | stock_name | stock_value | stock_ath | stock_atl | stock_today_high | stock_today_low | stock_day_change | stock_day_volume | stock_day_change_percent
----------+------------+-------------+-----------+-----------+------------------+-----------------+------------------+------------------+--------------------------
 S1       | ITC        |         800 |      1600 |       400 |              900 |             810 |               90 |           894274 |                       10
 S2       | Zomato     |         200 |       250 |        60 |              210 |             170 |               40 |           917298 |                       20
 S3       | Tesla      |        1000 |      3000 |       500 |             1200 |             800 |              400 |           894274 |                       20
 S4       | Reliance   |        2400 |      3000 |      1200 |             2400 |            2200 |             -240 |           924682 |                      -10
 S5       | Yes Bank   |         100 |       400 |        10 |              150 |             100 |              -50 |           894274 |                      -50
 S6       | HDFC       |         300 |       300 |       100 |              300 |             240 |               60 |           139572 |                       20
 S7       | MRF        |         500 |      1000 |       500 |              500 |             700 |              200 |           628462 |                      -20
(7 rows)
```

5.2.3 STOCKWATCHLIST

INSERT INTO STOCKWATCHLIST VALUES('U1','S2','Zomato','200');

INSERT INTO STOCKWATCHLIST VALUES('U1','S3','Tesla','1000');

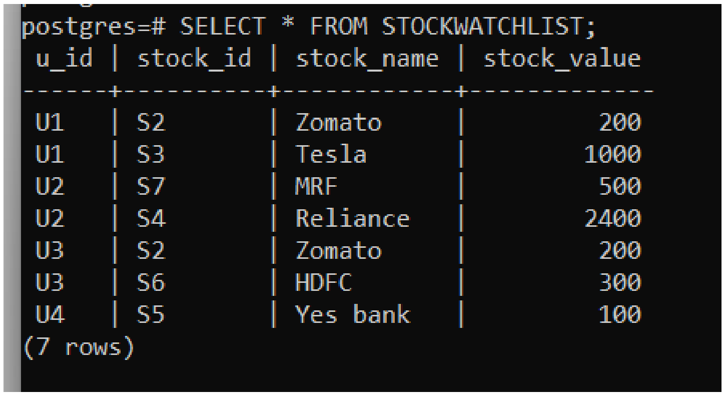
INSERT INTO STOCKWATCHLIST VALUES('U2','S7','MRF','500');

INSERT INTO STOCKWATCHLIST VALUES('U2','S4','Reliance','2400');

INSERT INTO STOCKWATCHLIST VALUES('U3','S2','Zomato','200');

INSERT INTO STOCKWATCHLIST VALUES('U3','S6','HDFC','300');

INSERT INTO STOCKWATCHLIST VALUES('U4','S5','Yes bank','100');



```
postgres=# SELECT * FROM STOCKWATCHLIST;
 u_id | stock_id | stock_name | stock_value
------+----------+------------+-------------
 U1   | S2       | Zomato     |         200
 U1   | S3       | Tesla      |        1000
 U2   | S7       | MRF        |         500
 U2   | S4       | Reliance   |        2400
 U3   | S2       | Zomato     |         200
 U3   | S6       | HDFC       |         300
 U4   | S5       | Yes bank   |         100
(7 rows)
```

5.2.4 STOCKHISTORY

INSERT INTO STOCKHISTORY VALUES('S1','ITC',700,650,1000,1200,10);

INSERT INTO STOCKHISTORY VALUES('S2','Zomato',150,100,NULL,NULL,20);

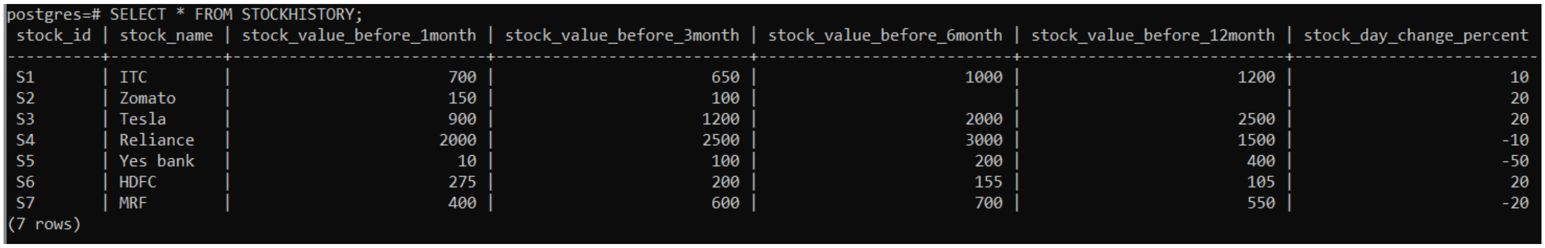
INSERT INTO STOCKHISTORY VALUES('S3','Tesla',900,1200,2000,2500,20);

INSERT INTO STOCKHISTORY VALUES('S4','Reliance',2000,2500,3000,1500,-10);

INSERT INTO STOCKHISTORY VALUES('S5','Yes bank',10,100,200,400,-50);

INSERT INTO STOCKHISTORY VALUES('S6','HDFC',275,200,155,105,20);

INSERT INTO STOCKHISTORY VALUES('S7','MRF',400,600,700,550,-20);



```
postgres=# SELECT * FROM STOCKHISTORY;
 stock_id | stock_name | stock_value_before_1month | stock_value_before_3month | stock_value_before_6month | stock_value_before_12month | stock_day_change_percent
----------+------------+---------------------------+---------------------------+---------------------------+----------------------------+--------------------------
 S1       | ITC        |                       700 |                       650 |                      1000 |                       1200 |                       10
 S2       | Zomato     |                       150 |                       100 |                           |                            |                       20
 S3       | Tesla      |                       900 |                      1200 |                      2000 |                       2500 |                       20
 S4       | Reliance   |                      2000 |                      2500 |                      3000 |                       1500 |                      -10
 S5       | Yes bank   |                        10 |                       100 |                       200 |                        400 |                      -50
 S6       | HDFC       |                       275 |                       200 |                       155 |                        105 |                       20
 S7       | MRF        |                       400 |                       600 |                       700 |                        550 |                      -20
(7 rows)
```

5.2.5 TRENDINGSTOCKS

INSERT INTO TRENDINGSTOCKS VALUES('S1','ITC',4,700,10,894274);

INSERT INTO TRENDINGSTOCKS VALUES('S2','Zomato',2,200,20,917298);

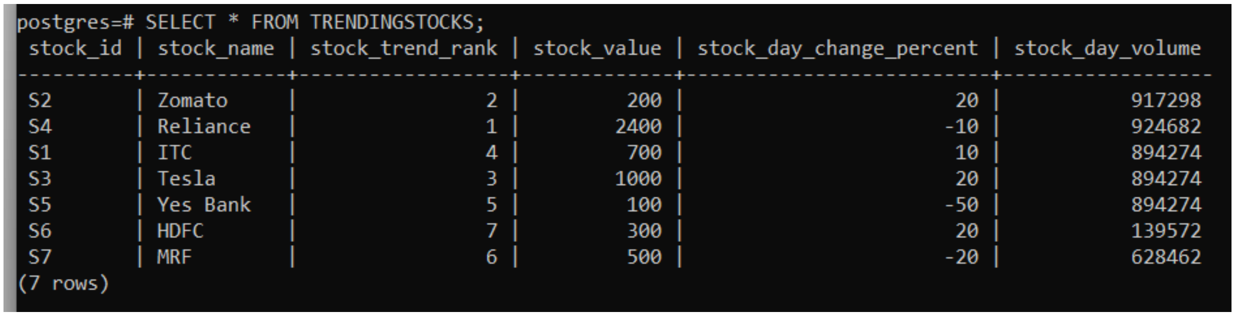
INSERT INTO TRENDINGSTOCKS VALUES('S3','Tesla',3,1000,20,894274);

INSERT INTO TRENDINGSTOCKS VALUES('S4','Reliance',1,2400,-10,924682);

INSERT INTO TRENDINGSTOCKS VALUES('S5','Yes Bank',5,100,-50,894274);

INSERT INTO TRENDINGSTOCKS VALUES('S6','HDFC',7,300,20,139572);

INSERT INTO TRENDINGSTOCKS VALUES('S7','MRF',6,500,-20,628462);



```
postgres=# SELECT * FROM TRENDINGSTOCKS;
 stock_id | stock_name | stock_trend_rank | stock_value | stock_day_change_percent | stock_day_volume
----------+------------+------------------+-------------+--------------------------+------------------
 S2       | Zomato     |                2 |         200 |                       20 |           917298
 S4       | Reliance   |                1 |        2400 |                      -10 |           924682
 S1       | ITC        |                4 |         700 |                       10 |           894274
 S3       | Tesla      |                3 |        1000 |                       20 |           894274
 S5       | Yes Bank   |                5 |         100 |                      -50 |           894274
 S6       | HDFC       |                7 |         300 |                       20 |           139572
 S7       | MRF        |                6 |         500 |                      -20 |           628462
(7 rows)
```

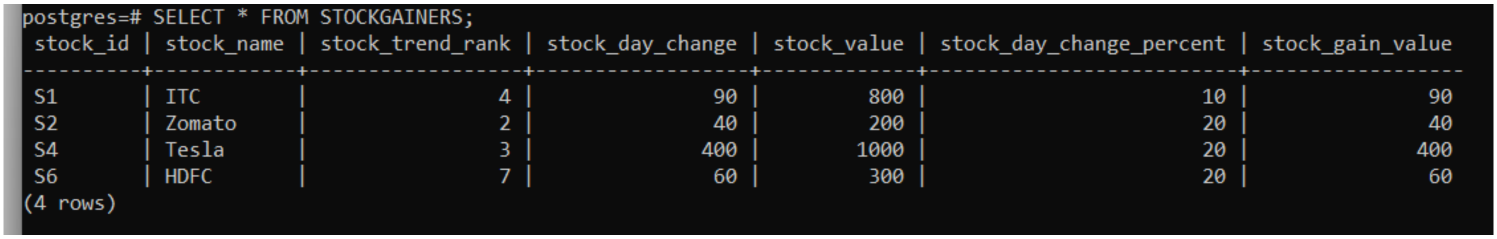
5.2.6 STOCKGAINERS

INSERT INTO STOCKGAINERS VALUES('S1','ITC',4,90,800,10,90);

INSERT INTO STOCKGAINERS VALUES('S2','Zomato',2,40,200,20,40);

INSERT INTO STOCKGAINERS VALUES('S4','Tesla',3,400,1000,20,400);

INSERT INTO STOCKGAINERS VALUES('S6','HDFC',7,60,300,20,60);



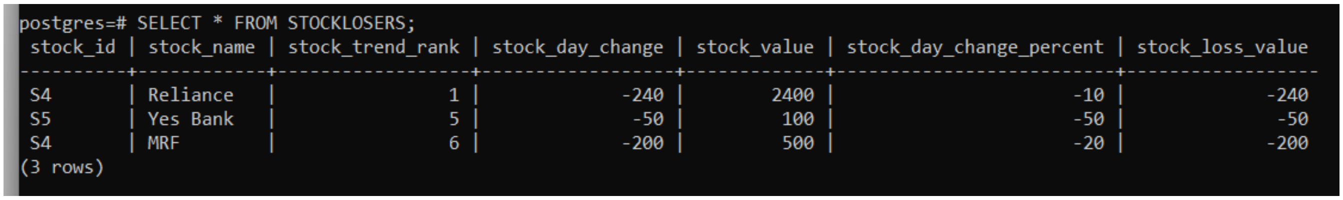
```
postgres=# SELECT * FROM STOCKGAINERS;
 stock_id | stock_name | stock_trend_rank | stock_day_change | stock_value | stock_day_change_percent | stock_gain_value
----------+------------+------------------+------------------+-------------+--------------------------+------------------
 S1       | ITC        |                4 |               90 |         800 |                       10 |               90
 S2       | Zomato     |                2 |               40 |         200 |                       20 |               40
 S4       | Tesla      |                3 |              400 |        1000 |                       20 |              400
 S6       | HDFC       |                7 |               60 |         300 |                       20 |               60
(4 rows)
```

5.2.7 STOCKLOSERS

INSERT INTO STOCKLOSERS VALUES('S4','Reliance',1,-240,2400,-10,-240);

INSERT INTO STOCKLOSERS VALUES('S5','Yes Bank',5,-50,100,-50,-50);

INSERT INTO STOCKLOSERS VALUES('S4','MRF',6,-200,500,-20,-200);



```
postgres=# SELECT * FROM STOCKLOSERS;
 stock_id | stock_name | stock_trend_rank | stock_day_change | stock_value | stock_day_change_percent | stock_loss_value
----------+------------+------------------+------------------+-------------+--------------------------+------------------
 S4       | Reliance   |                1 |             -240 |        2400 |                      -10 |             -240
 S5       | Yes Bank   |                5 |              -50 |         100 |                      -50 |              -50
 S4       | MRF        |                6 |             -200 |         500 |                      -20 |             -200
(3 rows)
```

5.2.8 USERREFFERALSCORE

INSERT INTO USERREFFERALSCORE VALUES('U1','Rushi','R1',NULL,1);

INSERT INTO USERREFFERALSCORE VALUES('U2','Harsh','R2','R1',1);

INSERT INTO USERREFFERALSCORE VALUES('U3','Sunil','R3','R2',1);

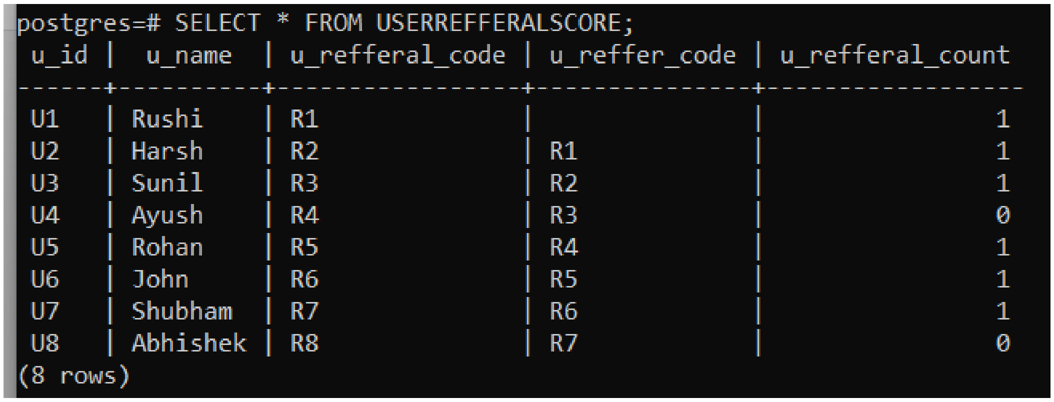
INSERT INTO USERREFFERALSCORE VALUES('U4','Ayush','R4','R3',1);

INSERT INTO USERREFFERALSCORE VALUES('U5','Rohan','R5','R4',1);

INSERT INTO USERREFFERALSCORE VALUES('U6','John','R6','R5',1);

INSERT INTO USERREFFERALSCORE VALUES('U7','Shubham','R7','R6',1);

INSERT INTO USERREFFERALSCORE VALUES('U8','Abhishek','R8','R7',0);



```
postgres=# SELECT * FROM USERREFFERALSCORE;
 u_id |  u_name  | u_refferal_code | u_reffer_code | u_refferal_count
------+----------+-----------------+---------------+------------------
 U1   | Rushi    | R1              |               |                1
 U2   | Harsh    | R2              | R1            |                1
 U3   | Sunil    | R3              | R2            |                1
 U4   | Ayush    | R4              | R3            |                0
 U5   | Rohan    | R5              | R4            |                1
 U6   | John     | R6              | R5            |                1
 U7   | Shubham  | R7              | R6            |                1
 U8   | Abhishek | R8              | R7            |                0
(8 rows)
```

5.2.9 RECOMMENDEDSTOCKS

INSERT INTO RECOMMENDEDSTOCKS VALUES('Zerodha','S2',400,6);

INSERT INTO RECOMMENDEDSTOCKS VALUES('Money Control','S2',600,12);

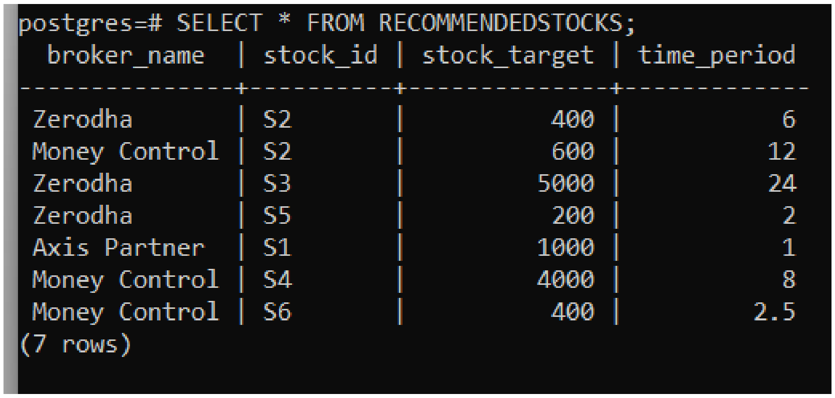
INSERT INTO RECOMMENDEDSTOCKS VALUES('Zerodha','S3',5000,24);

INSERT INTO RECOMMENDEDSTOCKS VALUES('Zerodha','S5',200,2);

INSERT INTO RECOMMENDEDSTOCKS VALUES('Axis Partner','S1',1000,1);

INSERT INTO RECOMMENDEDSTOCKS VALUES('Money Control','S4',4000,8);

INSERT INTO RECOMMENDEDSTOCKS VALUES('Money Control','S6',400,2.5);



```
postgres=# SELECT * FROM RECOMMENDEDSTOCKS;
  broker_name  | stock_id | stock_target | time_period
---------------+----------+--------------+-------------
 Zerodha       | S2       |          400 |           6
 Money Control | S2       |          600 |          12
 Zerodha       | S3       |         5000 |          24
 Zerodha       | S5       |          200 |           2
 Axis Partner  | S1       |         1000 |           1
 Money Control | S4       |         4000 |           8
 Money Control | S6       |          400 |         2.5
(7 rows)
```

5.2.10 USERVERIFIED

INSERT INTO USERVERIFIED VALUES('U1','rushi@gmail.com',9925761023,true);

INSERT INTO USERVERIFIED VALUES('U2','harsh@gmail.com',9283740294,true);

INSERT INTO USERVERIFIED VALUES('U3','sunil@gmail.com',8294384958,false);

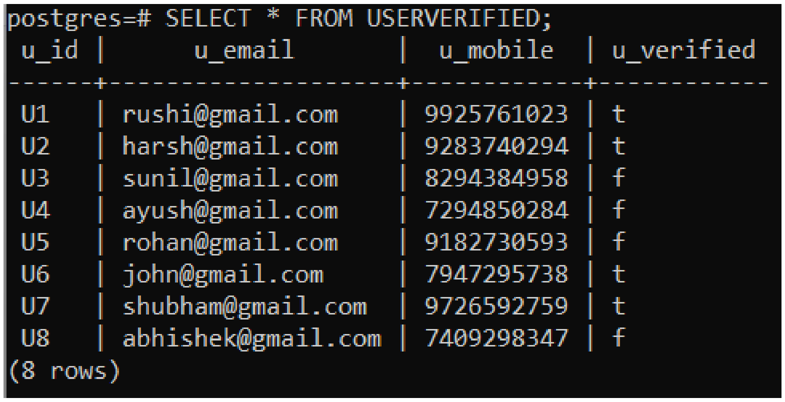
INSERT INTO USERVERIFIED VALUES('U4','ayush@gmail.com',7294850284,false);

INSERT INTO USERVERIFIED VALUES('U5','rohan@gmail.com',9182730593,false);

INSERT INTO USERVERIFIED VALUES('U6','john@gmail.com',7947295738,true);

INSERT INTO USERVERIFIED VALUES('U7','shubham@gmail.com',9726592759,true);

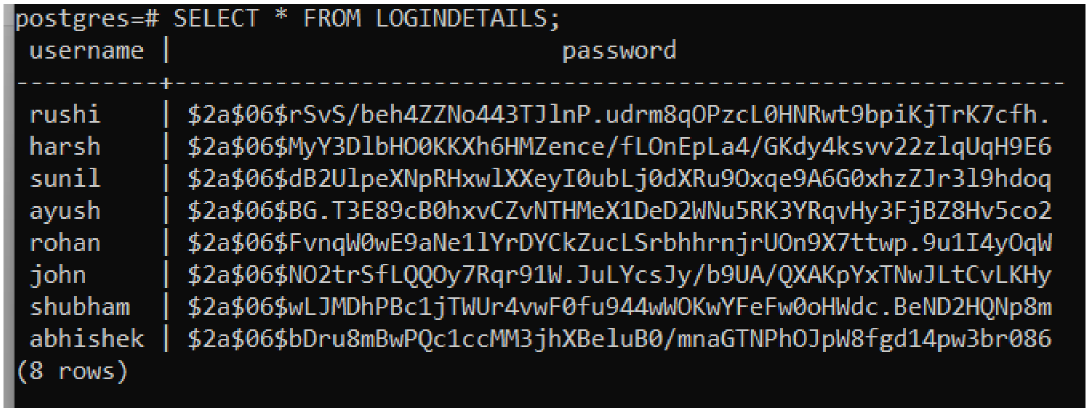
INSERT INTO USERVERIFIED VALUES('U8','abhishek@gmail.com',7409298347,false);



```
postgres=# SELECT * FROM USERVERIFIED;
 u_id |      u_email       |  u_mobile  | u_verified
------+--------------------+------------+------------
 U1   | rushi@gmail.com    | 9925761023 | t
 U2   | harsh@gmail.com    | 9283740294 | t
 U3   | sunil@gmail.com    | 8294384958 | f
 U4   | ayush@gmail.com    | 7294850284 | f
 U5   | rohan@gmail.com    | 9182730593 | f
 U6   | john@gmail.com     | 7947295738 | t
 U7   | shubham@gmail.com  | 9726592759 | t
 U8   | abhishek@gmail.com | 7409298347 | f
(8 rows)
```

5.2.11 LOGINDETAILS

```
CREATE EXTENSION pgcrypto;

INSERT INTO LOGINDETAILS
VALUES('rushi',crypt('rushi@123', gen_salt('bf')));

INSERT INTO LOGINDETAILS
VALUES('harsh',crypt('harsh@321', gen_salt('bf')));

INSERT INTO LOGINDETAILS
VALUES('sunil',crypt('sunil@435', gen_salt('bf')));

INSERT INTO LOGINDETAILS
VALUES('ayush',crypt('ayush@534', gen_salt('bf')));

INSERT INTO LOGINDETAILS
VALUES('rohan',crypt('rohan@12', gen_salt('bf')));

INSERT INTO LOGINDETAILS
VALUES('john',crypt('john@92', gen_salt('bf')));

INSERT INTO LOGINDETAILS
VALUES('shubham',crypt('shubham@34', gen_salt('bf')));

INSERT INTO LOGINDETAILS
VALUES('abhishek',crypt('abhishek@97', gen_salt('bf')));
```



```
postgres=# SELECT * FROM LOGINDETAILS;
 username |                           password
----------+-------------------------------------------------------------
 rushi    | $2a$06$rSvS/beh4ZZNo443TJlnP.udrm8qOPzcL0HNRwt9bpiKjTrK7cfh.
 harsh    | $2a$06$MyY3DlbHO0KKXh6HMZence/fLOnEpLa4/GKdy4ksvv22zlqUqH9E6
 sunil    | $2a$06$dB2UlpeXNpRHxwlXXeyI0ubLj0dXRu9Oxqe9A6G0xhzZJr3l9hdoq
 ayush    | $2a$06$BG.T3E89cB0hxvCZvNTHMeX1DeD2WNu5RK3YRqvHy3FjBZ8Hv5co2
 rohan    | $2a$06$FvnqW0wE9aNe1lYrDYCkZucLSrbhhrnjrUOn9X7ttwp.9u1I4yOqW
 john     | $2a$06$NO2trSfLQQOy7Rqr91W.JuLYcsJy/b9UA/QXAKpYxTNwJLtCvLKHy
 shubham  | $2a$06$wLJMDhPBc1jTWUr4vwF0fu944wWOKwYFeFw0oHWdc.BeND2HQNp8m
 abhishek | $2a$06$bDru8mBwPQc1ccMM3jhXBeluB0/mnaGTNPhOJpW8fgd14pw3br086
(8 rows)
```

5.2.12 STOCKASSETS

INSERT INTO STOCKASSETS(U_ID,STOCK_ID,QUANTITY,BUY_PRICE)
VALUES('U1','S2',2,200);

INSERT INTO STOCKASSETS(U_ID,STOCK_ID,QUANTITY,BUY_PRICE)
VALUES('U3','S5',5,500);

INSERT INTO STOCKASSETS(U_ID,STOCK_ID,QUANTITY,BUY_PRICE)
VALUES('U1','S5',5,500);

INSERT INTO STOCKASSETS(U_ID,STOCK_ID,QUANTITY,BUY_PRICE)
VALUES('U1','S1',2,700);

INSERT INTO STOCKASSETS(U_ID,STOCK_ID,QUANTITY,BUY_PRICE)
VALUES('U2','S2',3,800);

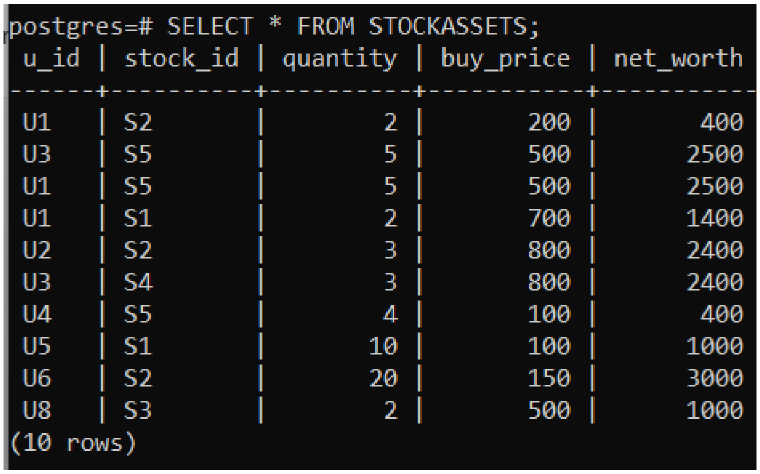
INSERT INTO STOCKASSETS(U_ID,STOCK_ID,QUANTITY,BUY_PRICE)
VALUES('U3','S4',3,800);

INSERT INTO STOCKASSETS(U_ID,STOCK_ID,QUANTITY,BUY_PRICE)
VALUES('U4','S5',4,100);

INSERT INTO STOCKASSETS(U_ID,STOCK_ID,QUANTITY,BUY_PRICE)
VALUES('U5','S1',10,100);

INSERT INTO STOCKASSETS(U_ID,STOCK_ID,QUANTITY,BUY_PRICE)
VALUES('U6','S2',20,150);

INSERT INTO STOCKASSETS(U_ID,STOCK_ID,QUANTITY,BUY_PRICE)
VALUES('U8','S3',2,500);



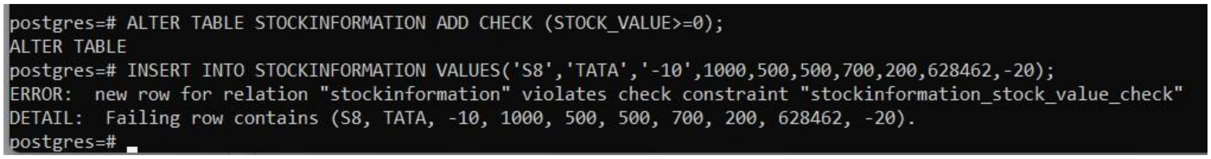
```
postgres=# SELECT * FROM STOCKASSETS;
 u_id | stock_id | quantity | buy_price | net_worth
------+----------+----------+-----------+-----------
 U1   | S2       |        2 |       200 |       400
 U3   | S5       |        5 |       500 |      2500
 U1   | S5       |        5 |       500 |      2500
 U1   | S1       |        2 |       700 |      1400
 U2   | S2       |        3 |       800 |      2400
 U3   | S4       |        3 |       800 |      2400
 U4   | S5       |        4 |       100 |       400
 U5   | S1       |       10 |       100 |      1000
 U6   | S2       |       20 |       150 |      3000
 U8   | S3       |        2 |       500 |      1000
(10 rows)
```

5.3 QUERIES

5.3.1. Check constraint query.

- ALTER TABLE STOCKINFORMATION ADD CHECK (STOCK_VALUE>=0);
- INSERT INTO STOCKINFORMATION VALUES('S8','TATA','-10',1000,500,500,700,200,628462,-20);

Output:

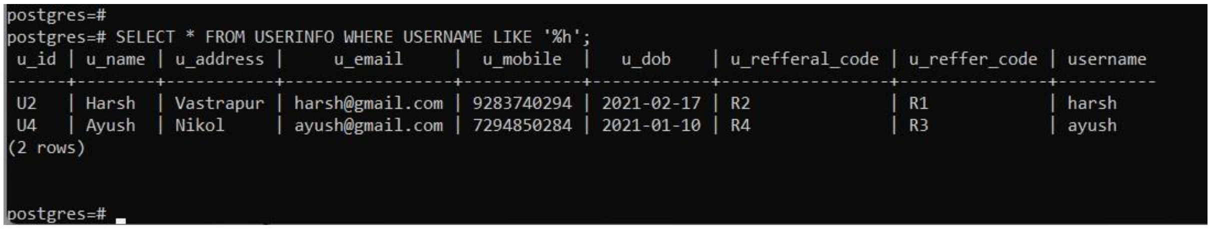


```
postgres=# ALTER TABLE STOCKINFORMATION ADD CHECK (STOCK_VALUE>=0);
ALTER TABLE
postgres=# INSERT INTO STOCKINFORMATION VALUES('S8','TATA','-10',1000,500,500,700,200,628462,-20);
ERROR:  new row for relation "stockinformation" violates check constraint "stockinformation_stock_value_check"
DETAIL:  Failing row contains (S8, TATA, -10, 1000, 500, 500, 700, 200, 628462, -20).
postgres=#
```

5.3.2. Display data of username whose name ends with ‘h’ from userinfo.

- SELECT * FROM USERINFO WHERE USERNAME LIKE '%h';

Output:

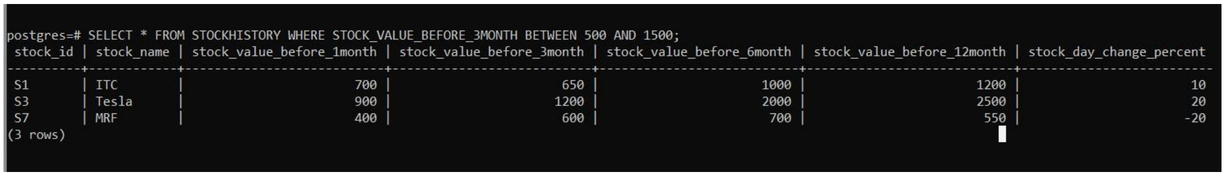


```
postgres=#
postgres=# SELECT * FROM USERINFO WHERE USERNAME LIKE '%h';
 u_id | u_name | u_address |     u_email     |  u_mobile  |   u_dob    | u_refferal_code | u_reffer_code | username
------+--------+-----------+-----------------+------------+------------+-----------------+---------------+----------
 U2   | Harsh  | Vastrapur | harsh@gmail.com | 9283740294 | 2021-02-17 | R2              | R1            | harsh
 U4   | Ayush  | Nikol     | ayush@gmail.com | 7294850284 | 2021-01-10 | R4              | R3            | ayush
(2 rows)


postgres=#
```

5.3.3. Display data of stock whose stock_value_before_3months is between 500 and 1500 from STOCKHISTORY.

- SELECT * FROM STOCKHISTORY WHERE STOCK_VALUE_BEFORE_3MONTH BETWEEN 500 AND 1500;

Output:

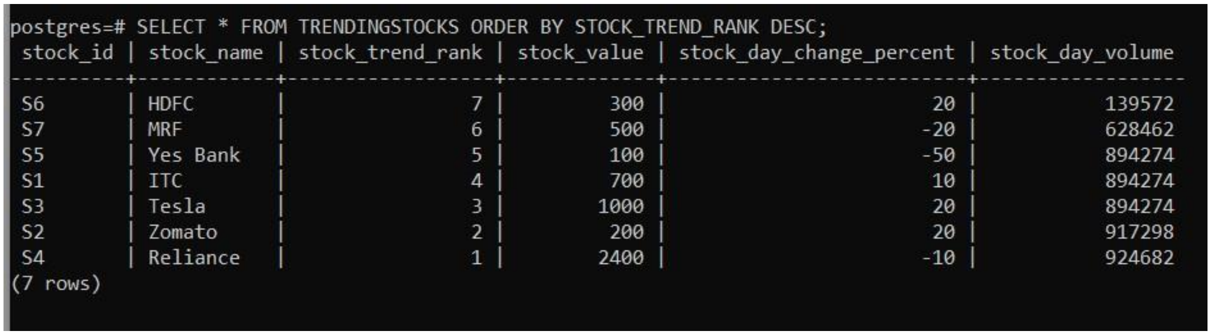


```
postgres=# SELECT * FROM STOCKHISTORY WHERE STOCK_VALUE_BEFORE_3MONTH BETWEEN 500 AND 1500;
 stock_id | stock_name | stock_value_before_1month | stock_value_before_3month | stock_value_before_6month | stock_value_before_12month | stock_day_change_percent
----------+------------+---------------------------+---------------------------+---------------------------+----------------------------+--------------------------
 S1       | ITC        |                       700 |                       650 |                      1000 |                       1200 |                       10
 S3       | Tesla      |                       900 |                      1200 |                      2000 |                       2500 |                       20
 S7       | MRF        |                       400 |                       600 |                       700 |                        550 |                      -20
(3 rows)
```

5.3.4 Display data of stock in the order of worst to best order from TRENDINDINGSTOCKS.

- SELECT * FROM TRENDINGSTOCKS ORDER BY STOCK_TREND_RANK DESC;

Output:

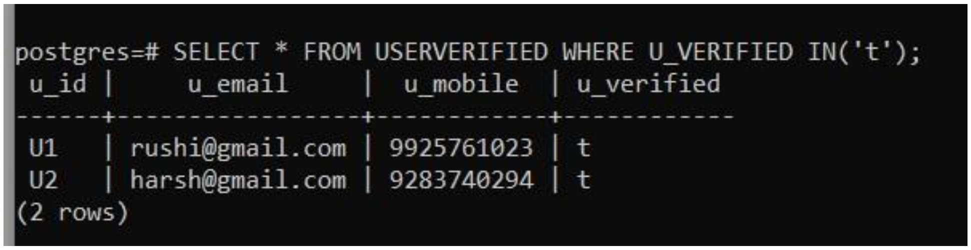


```
postgres=# SELECT * FROM TRENDINGSTOCKS ORDER BY STOCK_TREND_RANK DESC;
 stock_id | stock_name | stock_trend_rank | stock_value | stock_day_change_percent | stock_day_volume
----------+------------+------------------+-------------+--------------------------+------------------
 S6       | HDFC       |                7 |         300 |                       20 |           139572
 S7       | MRF        |                6 |         500 |                      -20 |           628462
 S5       | Yes Bank   |                5 |         100 |                      -50 |           894274
 S1       | ITC        |                4 |         700 |                       10 |           894274
 S3       | Tesla      |                3 |        1000 |                       20 |           894274
 S2       | Zomato     |                2 |         200 |                       20 |           917298
 S4       | Reliance   |                1 |        2400 |                      -10 |           924682
(7 rows)
```

5.3.5. Display user who are verified from userverified.

- SELECT * FROM USERVERIFIED WHERE U_VERIFIED IN('t');

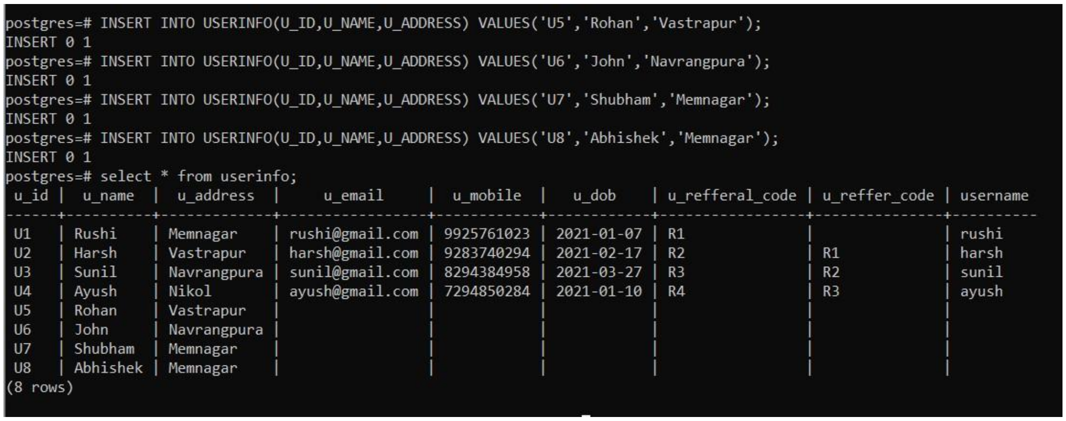
Output:



```
postgres=# SELECT * FROM USERVERIFIED WHERE U_VERIFIED IN('t');
 u_id |     u_email      |  u_mobile  | u_verified
------+-----------------+------------+------------
 U1   | rushi@gmail.com | 9925761023 | t
 U2   | harsh@gmail.com | 9283740294 | t
(2 rows)
```

5.3.6. Display count of users who are from same address in userinfo.

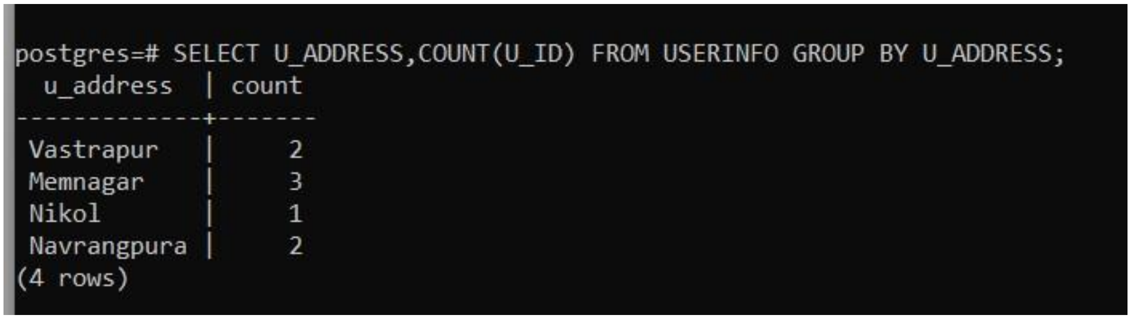
- To perform this query, I had to insert some dummy data into our userinfo table.
- Dummy Data:



```
postgres=# INSERT INTO USERINFO(U_ID,U_NAME,U_ADDRESS) VALUES('U5','Rohan','Vastrapur');
INSERT 0 1
postgres=# INSERT INTO USERINFO(U_ID,U_NAME,U_ADDRESS) VALUES('U6','John','Navrangpura');
INSERT 0 1
postgres=# INSERT INTO USERINFO(U_ID,U_NAME,U_ADDRESS) VALUES('U7','Shubham','Memnagar');
INSERT 0 1
postgres=# INSERT INTO USERINFO(U_ID,U_NAME,U_ADDRESS) VALUES('U8','Abhishek','Memnagar');
INSERT 0 1
postgres=# select * from userinfo;
 u_id |  u_name  |  u_address  |     u_email     |  u_mobile  |   u_dob    | u_refferal_code | u_reffer_code | username
------+----------+-------------+-----------------+------------+------------+-----------------+---------------+----------
 U1   | Rushi    | Memnagar    | rushi@gmail.com | 9925761023 | 2021-01-07 | R1              |               | rushi
 U2   | Harsh    | Vastrapur   | harsh@gmail.com | 9283740294 | 2021-02-17 | R2              | R1            | harsh
 U3   | Sunil    | Navrangpura | sunil@gmail.com | 8294384958 | 2021-03-27 | R3              | R2            | sunil
 U4   | Ayush    | Nikol       | ayush@gmail.com | 7294850284 | 2021-01-10 | R4              | R3            | ayush
 U5   | Rohan    | Vastrapur   |                 |            |            |                 |               |
 U6   | John     | Navrangpura |                 |            |            |                 |               |
 U7   | Shubham  | Memnagar    |                 |            |            |                 |               |
 U8   | Abhishek | Memnagar    |                 |            |            |                 |               |
(8 rows)
```

- **Query:**
- SELECT U_ADDRESS,COUNT(U_ID) FROM USERINFO GROUP BY U_ADDRESS;

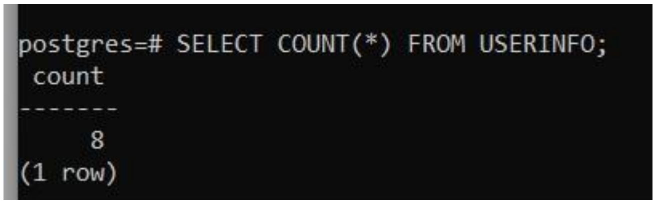
Output:



```
postgres=# SELECT U_ADDRESS,COUNT(U_ID) FROM USERINFO GROUP BY U_ADDRESS;
  u_address  | count
-------------+-------
 Vastrapur   |     2
 Memnagar    |     3
 Nikol       |     1
 Navrangpura |     2
(4 rows)
```

5.3.7. Display number of users in our database(userinfo).

- SELECT COUNT(*) FROM USERINFO;
- **Output:**

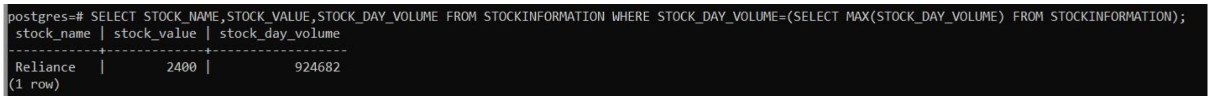


```
postgres=# SELECT COUNT(*) FROM USERINFO;
 count
-------
     8
(1 row)
```

5.3.8. Display the stock who has maximum volume in stock information.

- SELECT STOCK_NAME,STOCK_VALUE,STOCK_DAY_VOLUME FROM STOCKINFORMATION WHERE STOCK_DAY_VOLUME=(SELECT MAX(STOCK_DAY_VOLUME) FROM STOCKINFORMATION);

Output:

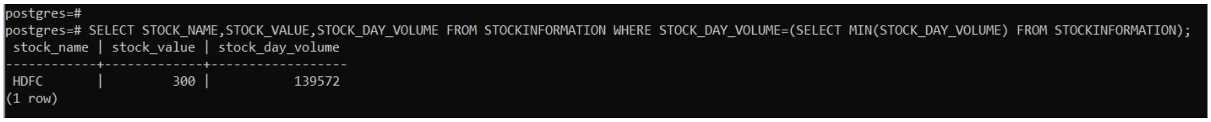


```
postgres=# SELECT STOCK_NAME,STOCK_VALUE,STOCK_DAY_VOLUME FROM STOCKINFORMATION WHERE STOCK_DAY_VOLUME=(SELECT MAX(STOCK_DAY_VOLUME) FROM STOCKINFORMATION);
 stock_name | stock_value | stock_day_volume
------------+-------------+------------------
 Reliance   |        2400 |           924682
(1 row)
```

5.3.9. Display the stock who has minimum volume in stock information.

- SELECT STOCK_NAME,STOCK_VALUE FROM STOCKINFORMATION WHERE STOCK_DAY_VOLUME=(SELECT MIN(STOCK_DAY_VOLUME) FROM STOCKINFORMATION);

Output:

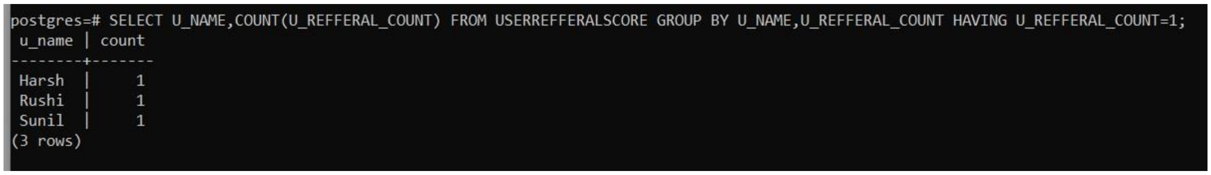


```
postgres=#
postgres=# SELECT STOCK_NAME,STOCK_VALUE,STOCK_DAY_VOLUME FROM STOCKINFORMATION WHERE STOCK_DAY_VOLUME=(SELECT MIN(STOCK_DAY_VOLUME) FROM STOCKINFORMATION);
 stock_name | stock_value | stock_day_volume
------------+-------------+------------------
 HDFC       |         300 |           139572
(1 row)
```

5.3.10. Display users who has exactly 1 refferal count from userrrefferalscore.

- SELECT U_NAME,COUNT(U_REFFERAL_COUNT) FROM USERREFFERALSCORE GROUP BY U_NAME,U_REFFERAL_COUNT HAVING U_REFFERAL_COUNT=1;

Output:



```
postgres=# SELECT U_NAME,COUNT(U_REFFERAL_COUNT) FROM USERREFFERALSCORE GROUP BY U_NAME,U_REFFERAL_COUNT HAVING U_REFFERAL_COUNT=1;
 u_name | count
--------+-------
 Harsh  |     1
 Rushi  |     1
 Sunil  |     1
(3 rows)
```

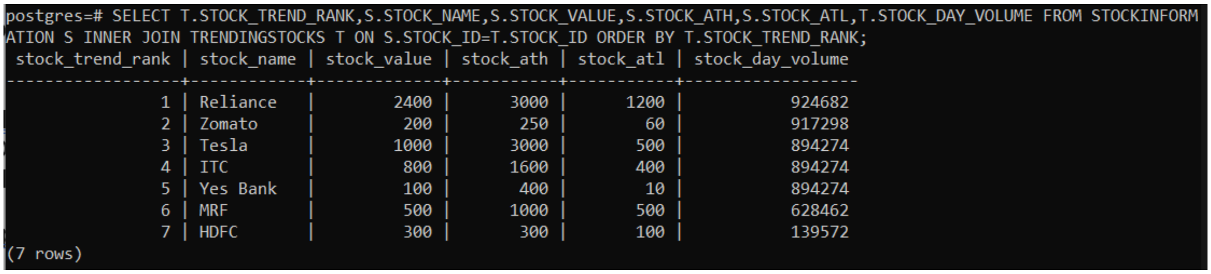
5.4 JOIN QUERIES

5.4.1 Display Information of trending stock and display them in order.

- SELECT T.STOCK_TREND_RANK,S.STOCK_NAME,S.STOCK_VALUE,

S.STOCK_ATH,S.STOCK_ATL,T.STOCK_DAY_VOLUME FROM STOCKINFORMATION S INNER JOIN TRENDINGSTOCKS T ON S.STOCK_ID=T.STOCK_ID ORDER BY T.STOCK_TREND_RANK;

Output :

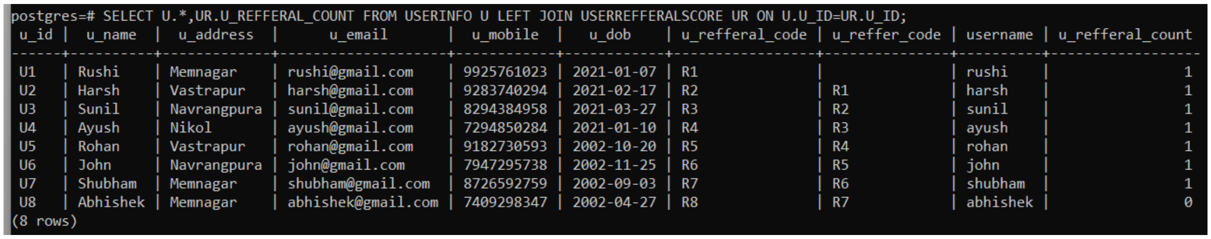


```
postgres=# SELECT T.STOCK_TREND_RANK,S.STOCK_NAME,S.STOCK_VALUE,S.STOCK_ATH,S.STOCK_ATL,T.STOCK_DAY_VOLUME FROM STOCKINFORM
ATION S INNER JOIN TRENDINGSTOCKS T ON S.STOCK_ID=T.STOCK_ID ORDER BY T.STOCK_TREND_RANK;
 stock_trend_rank | stock_name | stock_value | stock_ath | stock_atl | stock_day_volume
------------------+------------+-------------+-----------+-----------+------------------
                1 | Reliance   |        2400 |      3000 |      1200 |           924682
                2 | Zomato     |         200 |       250 |        60 |           917298
                3 | Tesla      |        1000 |      3000 |       500 |           894274
                4 | ITC        |         800 |      1600 |       400 |           894274
                5 | Yes Bank   |         100 |       400 |        10 |           894274
                6 | MRF        |         500 |      1000 |       500 |           628462
                7 | HDFC       |         300 |       300 |       100 |           139572
(7 rows)
```

5.4.2 Display referral count of user including their information using left join.

- SELECT U.*,UR.U_REFFERAL_COUNT FROM USERINFO U LEFT JOIN USERREFFERALSCORE UR ON U.U_ID=UR.U_ID;

Output :

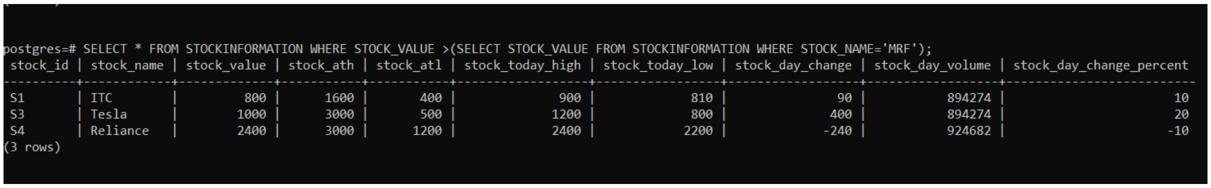


```
postgres=# SELECT U.*,UR.U_REFFERAL_COUNT FROM USERINFO U LEFT JOIN USERREFFERALSCORE UR ON U.U_ID=UR.U_ID;
 u_id |  u_name  |  u_address  |      u_email       |  u_mobile  |   u_dob    | u_refferal_code | u_reffer_code | username | u_refferal_count
------+----------+-------------+--------------------+------------+------------+-----------------+---------------+----------+------------------
 U1   | Rushi    | Memnagar    | rushi@gmail.com    | 9925761023 | 2021-01-07 | R1              |               | rushi    |                1
 U2   | Harsh    | Vastrapur   | harsh@gmail.com    | 9283740294 | 2021-02-17 | R2              | R1            | harsh    |                1
 U3   | Sunil    | Navrangpura | sunil@gmail.com    | 8294384958 | 2021-03-27 | R3              | R2            | sunil    |                1
 U4   | Ayush    | Nikol       | ayush@gmail.com    | 7294850284 | 2021-01-10 | R4              | R3            | ayush    |                1
 U5   | Rohan    | Vastrapur   | rohan@gmail.com    | 9182730593 | 2002-10-20 | R5              | R4            | rohan    |                1
 U6   | John     | Navrangpura | john@gmail.com     | 7947295738 | 2002-11-25 | R6              | R5            | john     |                1
 U7   | Shubham  | Memnagar    | shubham@gmail.com  | 8726592759 | 2002-09-03 | R7              | R6            | shubham  |                1
 U8   | Abhishek | Memnagar    | abhishek@gmail.com | 7409298347 | 2002-04-27 | R8              | R7            | abhishek |                0
(8 rows)
```

5.4.3 Display stock information of stock whose value is more than MRF’s stock value using sub query.

- SELECT * FROM STOCKINFORMATION WHERE STOCK_VALUE >(SELECT STOCK_VALUE FROM STOCKINFORMATION WHERE STOCK_NAME='MRF');

Output :



```
postgres=# SELECT * FROM STOCKINFORMATION WHERE STOCK_VALUE >(SELECT STOCK_VALUE FROM STOCKINFORMATION WHERE STOCK_NAME='MRF');
 stock_id | stock_name | stock_value | stock_ath | stock_atl | stock_today_high | stock_today_low | stock_day_change | stock_day_volume | stock_day_change_percent
----------+------------+-------------+-----------+-----------+------------------+-----------------+------------------+------------------+--------------------------
 S1       | ITC        |         800 |      1600 |       400 |              900 |             810 |               90 |           894274 |                       10
 S3       | Tesla      |        1000 |      3000 |       500 |             1200 |             800 |              400 |           894274 |                       20
 S4       | Reliance   |        2400 |      3000 |      1200 |             2400 |            2200 |             -240 |           924682 |                      -10
(3 rows)
```

5.4.4 Display users who has stock in watchlist which gained more than or equal to 10%.

- SELECT U.U_ID,U.U_NAME,S.STOCK_NAME

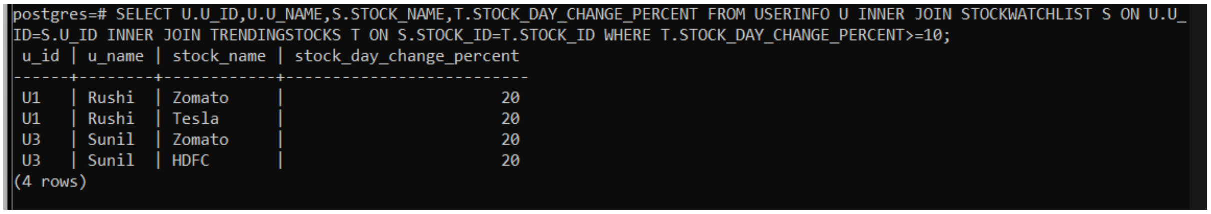
,T.STOCK_DAY_CHANGE_PERCENT FROM USERINFO U

INNER JOIN STOCKWATCHLIST S ON U.U_ID=S.U_ID

INNER JOIN TRENDINGSTOCKS T ON

S.STOCK_ID=T.STOCK_ID WHERE T.STOCK_DAY_CHANGE_PERCENT>=10;

Output :



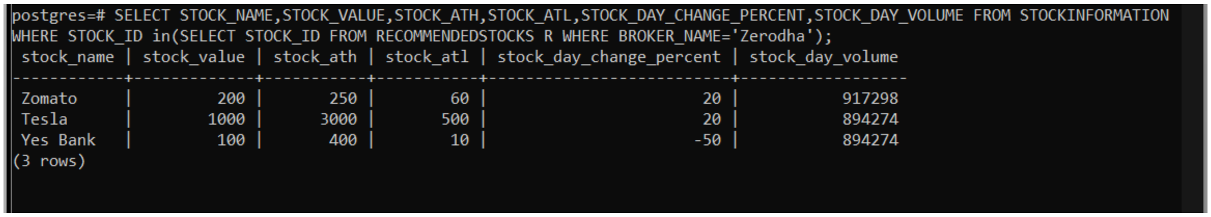
```
postgres=# SELECT U.U_ID,U.U_NAME,S.STOCK_NAME,T.STOCK_DAY_CHANGE_PERCENT FROM USERINFO U INNER JOIN STOCKWATCHLIST S ON U.U_
ID=S.U_ID INNER JOIN TRENDINGSTOCKS T ON S.STOCK_ID=T.STOCK_ID WHERE T.STOCK_DAY_CHANGE_PERCENT>=10;
 u_id | u_name | stock_name | stock_day_change_percent
------+--------+------------+--------------------------
 U1   | Rushi  | Zomato     |                       20
 U1   | Rushi  | Tesla      |                       20
 U3   | Sunil  | Zomato     |                       20
 U3   | Sunil  | HDFC       |                       20
(4 rows)
```

5.4.5 Display stock information of stock who is recommended by Zerodha.

- SELECT STOCK_NAME,STOCK_VALUE,STOCK_ATH,STOCK_ATL,

STOCK_DAY_CHANGE_PERCENT,STOCK_DAY_VOLUME FROM STOCKINFORMATION WHERE STOCK_ID in(SELECT STOCK_ID FROM RECOMMENDEDSTOCKS R WHERE BROKER_NAME='Zerodha');

Output :

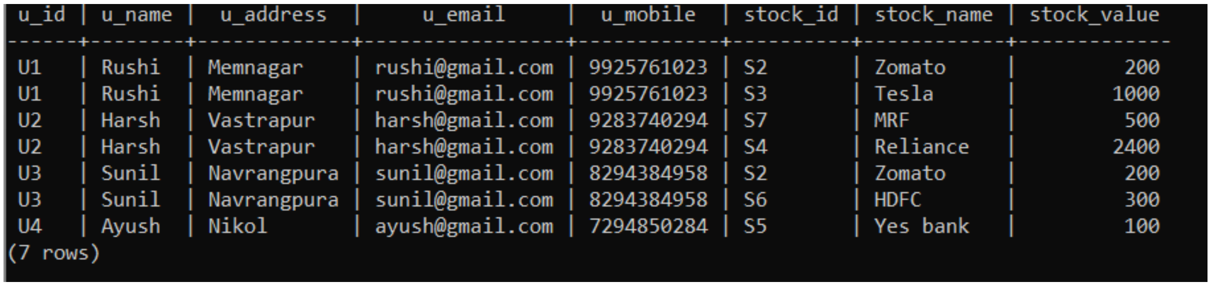


```
postgres=# SELECT STOCK_NAME,STOCK_VALUE,STOCK_ATH,STOCK_ATL,STOCK_DAY_CHANGE_PERCENT,STOCK_DAY_VOLUME FROM STOCKINFORMATION
WHERE STOCK_ID in(SELECT STOCK_ID FROM RECOMMENDEDSTOCKS R WHERE BROKER_NAME='Zerodha');
 stock_name | stock_value | stock_ath | stock_atl | stock_day_change_percent | stock_day_volume
------------+-------------+-----------+-----------+--------------------------+------------------
 Zomato     |         200 |       250 |        60 |                       20 |           917298
 Tesla      |        1000 |      3000 |       500 |                       20 |           894274
 Yes Bank   |         100 |       400 |        10 |                      -50 |           894274
(3 rows)
```

5.4.6 Display user info and their stock watch list.

- Select u.u_id,u.u_name,u.u_address,u.u_email,u.u_mobile,s.stock_id,s.stock_name,s.stock_value from userinfo u ,StockWatchList s where u.u_id=s.u_id;

Output :



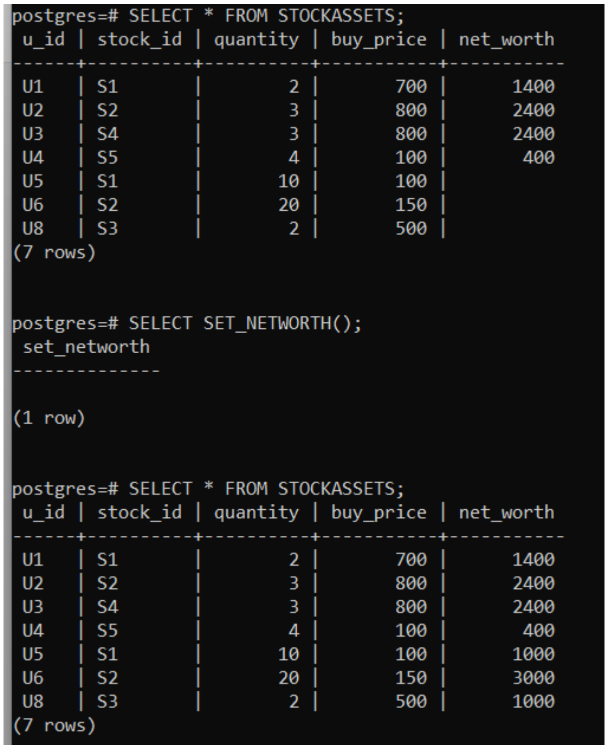
```
 u_id | u_name |  u_address  |     u_email      |  u_mobile  | stock_id | stock_name | stock_value
------+--------+-------------+------------------+------------+----------+------------+-------------
 U1   | Rushi  | Memnagar    | rushi@gmail.com  | 9925761023 | S2       | Zomato     |         200
 U1   | Rushi  | Memnagar    | rushi@gmail.com  | 9925761023 | S3       | Tesla      |        1000
 U2   | Harsh  | Vastrapur   | harsh@gmail.com  | 9283740294 | S7       | MRF        |         500
 U2   | Harsh  | Vastrapur   | harsh@gmail.com  | 9283740294 | S4       | Reliance   |        2400
 U3   | Sunil  | Navrangpura | sunil@gmail.com  | 8294384958 | S2       | Zomato     |         200
 U3   | Sunil  | Navrangpura | sunil@gmail.com  | 8294384958 | S6       | HDFC       |         300
 U4   | Ayush  | Nikol       | ayush@gmail.com  | 7294850284 | S5       | Yes bank   |         100
(7 rows)
```

5.5 FUNCTIONS

5.5.1 Create a function to calculate the net worth of user holding stocks in stock asset and also insert calculated net worth in respective column.

```
CREATE OR REPLACE FUNCTION SET_NETWORTH() RETURNS VOID
LANGUAGE PLPGSQL
AS $$
DECLARE
BEGIN
UPDATE STOCKASSETS SET NET_WORTH=QUANTITY*BUY_PRICE;
UPDATE BACKUP_STOCK_ASSETS SET
NET_WORTH=QUANTITY*BUY_PRICE;
END $$
```

Output :



```
postgres=# SELECT * FROM STOCKASSETS;
 u_id | stock_id | quantity | buy_price | net_worth
------+----------+----------+-----------+-----------
 U1   | S1       |        2 |       700 |      1400
 U2   | S2       |        3 |       800 |      2400
 U3   | S4       |        3 |       800 |      2400
 U4   | S5       |        4 |       100 |       400
 U5   | S1       |       10 |       100 |
 U6   | S2       |       20 |       150 |
 U8   | S3       |        2 |       500 |
(7 rows)


postgres=# SELECT SET_NETWORTH();
 set_networth
--------------

(1 row)


postgres=# SELECT * FROM STOCKASSETS;
 u_id | stock_id | quantity | buy_price | net_worth
------+----------+----------+-----------+-----------
 U1   | S1       |        2 |       700 |      1400
 U2   | S2       |        3 |       800 |      2400
 U3   | S4       |        3 |       800 |      2400
 U4   | S5       |        4 |       100 |       400
 U5   | S1       |       10 |       100 |      1000
 U6   | S2       |       20 |       150 |      3000
 U8   | S3       |        2 |       500 |      1000
(7 rows)
```

5.6 TRIGGERS

5.6.1 Create a trigger to make a backup of stockassets table where a data is inserted whenever new data is inserted in original table.

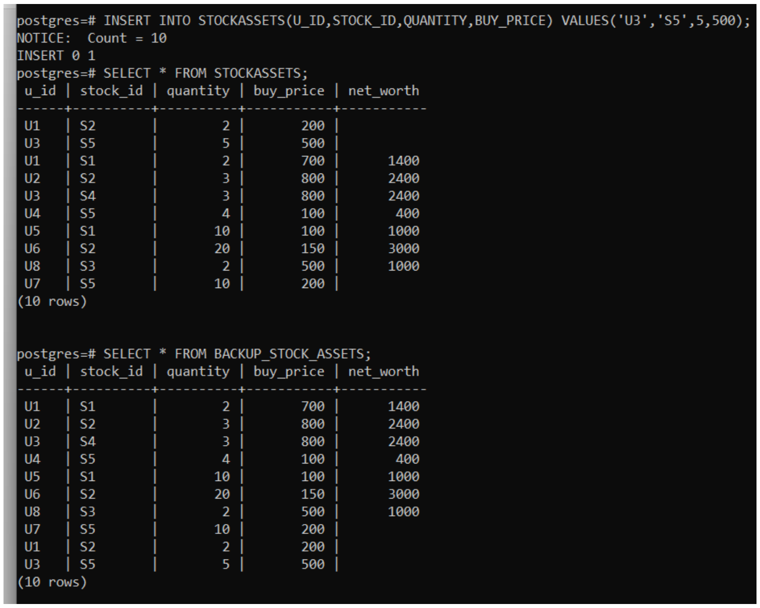
- Function for trigger:

```
CREATE FUNCTION MAKE_STOCKASSETS_BACKUP()
RETURNS TRIGGER
LANGUAGE PLPGSQL
AS
$$
BEGIN
    INSERT INTO
        BACKUP_STOCK_ASSETS(U_ID,STOCK_ID,QUANTITY,BUY_PRICE)
        VALUES
        (NEW.U_ID,NEW.STOCK_ID,NEW.QUANTITY,NEW.BUY_PRICE);
        RETURN NEW;
END;
$$ ;
```

- Trigger:

```
CREATE TRIGGER BACKUP_STOCKASSETS_DATA
AFTER INSERT ON STOCKASSETS
FOR EACH ROW EXECUTE PROCEDURE
    MAKE_STOCKASSETS_BACKUP();
```

Output :



```
postgres=# INSERT INTO STOCKASSETS(U_ID,STOCK_ID,QUANTITY,BUY_PRICE) VALUES('U3','S5',5,500);
NOTICE:  Count = 10
INSERT 0 1
postgres=# SELECT * FROM STOCKASSETS;
 u_id | stock_id | quantity | buy_price | net_worth
------+----------+----------+-----------+-----------
 U1   | S2       |        2 |       200 |
 U3   | S5       |        5 |       500 |
 U1   | S1       |        2 |       700 |      1400
 U2   | S2       |        3 |       800 |      2400
 U3   | S4       |        3 |       800 |      2400
 U4   | S5       |        4 |       100 |       400
 U5   | S1       |       10 |       100 |      1000
 U6   | S2       |       20 |       150 |      3000
 U8   | S3       |        2 |       500 |      1000
 U7   | S5       |       10 |       200 |
(10 rows)


postgres=# SELECT * FROM BACKUP_STOCK_ASSETS;
 u_id | stock_id | quantity | buy_price | net_worth
------+----------+----------+-----------+-----------
 U1   | S1       |        2 |       700 |      1400
 U2   | S2       |        3 |       800 |      2400
 U3   | S4       |        3 |       800 |      2400
 U4   | S5       |        4 |       100 |       400
 U5   | S1       |       10 |       100 |      1000
 U6   | S2       |       20 |       150 |      3000
 U8   | S3       |        2 |       500 |      1000
 U7   | S5       |       10 |       200 |
 U1   | S2       |        2 |       200 |
 U3   | S5       |        5 |       500 |
(10 rows)
```

5.6.2 Create a trigger to count number of records everytime you insert a record in stockassets table.

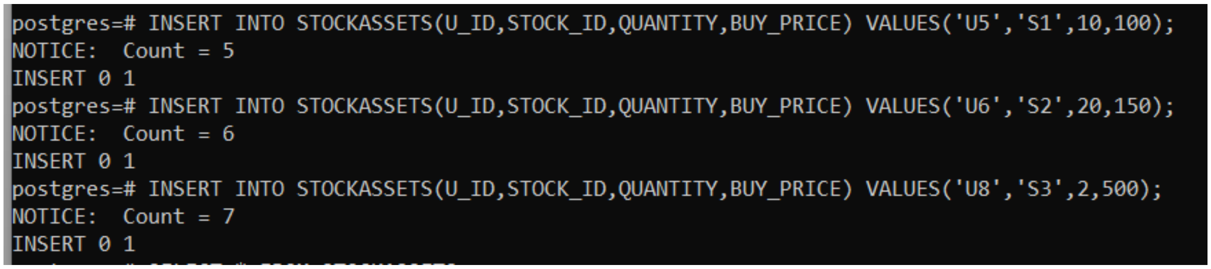
- Function for trigger:

```
CREATE FUNCTION COUNTLOG()
  RETURNS TRIGGER
  LANGUAGE PLPGSQL
  AS
$$
DECLARE TCOUNT INT;
BEGIN
        SELECT COUNT(*) INTO TCOUNT FROM STOCKASSETS;

        RAISE NOTICE 'COUNT = %',TCOUNT;
END;
$$;
```

- Trigger:

```
CREATE TRIGGER COUNTTRIGGER
  AFTER INSERT
  ON STOCKASSETS
  FOR EACH ROW
  EXECUTE PROCEDURE COUNTLOG();
```

Output :



```
postgres=# INSERT INTO STOCKASSETS(U_ID,STOCK_ID,QUANTITY,BUY_PRICE) VALUES('U5','S1',10,100);
NOTICE:  Count = 5
INSERT 0 1
postgres=# INSERT INTO STOCKASSETS(U_ID,STOCK_ID,QUANTITY,BUY_PRICE) VALUES('U6','S2',20,150);
NOTICE:  Count = 6
INSERT 0 1
postgres=# INSERT INTO STOCKASSETS(U_ID,STOCK_ID,QUANTITY,BUY_PRICE) VALUES('U8','S3',2,500);
NOTICE:  Count = 7
INSERT 0 1
```

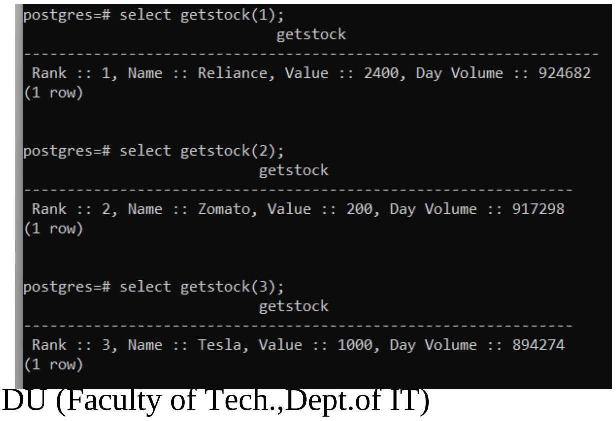
5.7 CURSORS

5.7.1 Display stock information according to its rank using cursor from trending stock table.

- Cursor

```
CREATE OR REPLACE FUNCTION GETSTOCK(STR NUMERIC)
RETURNS TEXT AS $$
DECLARE
MSG TEXT;
STOCKREC RECORD;
STOCKCUR CURSOR(STR NUMERIC)
FOR SELECT
STOCK_TREND_RANK,STOCK_NAME,STOCK_VALUE,STOCK_DAY_V
OLUME FROM TRENDINGSTOCKS WHERE STOCK_TREND_RANK =
STR;
BEGIN
OPEN STOCKCUR(STR);
LOOP
FETCH STOCKCUR INTO STOCKREC;
IF STOCKREC.STOCK_TREND_RANK=STR THEN
MSG :='Rank :: ' || STOCKREC.STOCK_TREND_RANK || ', Name :: ' ||
STOCKREC.STOCK_NAME || ', Value :: ' || STOCKREC.STOCK_VALUE ||
', Day Volume :: ' ||
STOCKREC.STOCK_DAY_VOLUME;
END IF;
EXIT WHEN NOT FOUND;
END LOOP;
CLOSE STOCKCUR;
RETURN MSG;
END;
$$
LANGUAGE PLPGSQL;
```

Output :



```
postgres=# select getstock(1);
                             getstock
--------------------------------------------------------------------
 Rank :: 1, Name :: Reliance, Value :: 2400, Day Volume :: 924682
(1 row)


postgres=# select getstock(2);
                           getstock
-----------------------------------------------------------------
 Rank :: 2, Name :: Zomato, Value :: 200, Day Volume :: 917298
(1 row)


postgres=# select getstock(3);
                           getstock
-----------------------------------------------------------------
 Rank :: 3, Name :: Tesla, Value :: 1000, Day Volume :: 894274
(1 row)
```

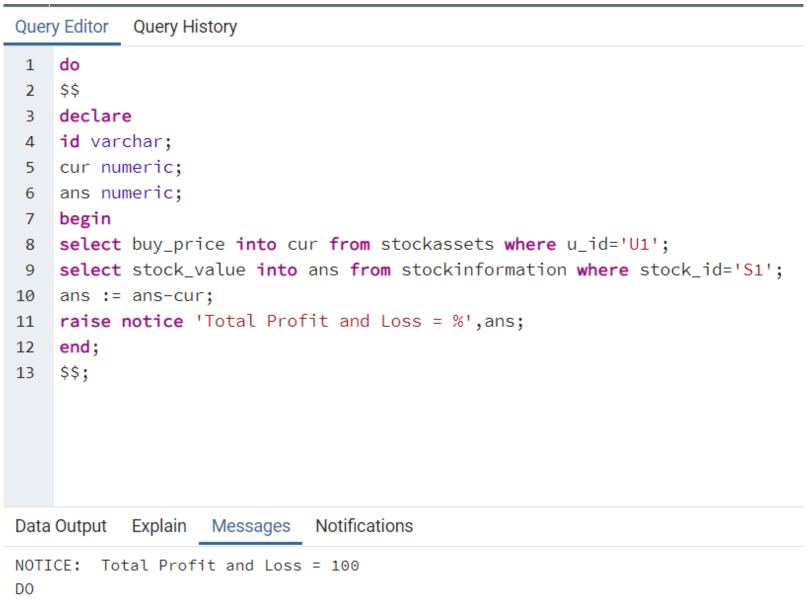
5.8 PLSQL

5.8.1 Display Profit/Loss of first record in stockassets using PLSQL.

- Block :

```
do
$$
declare
id varchar;
cur numeric;
ans numeric;
begin
select buy_price into cur from stockassets where u_id='U1';
select stock_value into ans from stockinformation where stock_id='S1';
ans := ans-cur;
raise notice 'Total Profit and Loss = %',ans;
end;
$$;
```

Output :



```
do
$$
declare
id varchar;
cur numeric;
ans numeric;
begin
select buy_price into cur from stockassets where u_id='U1';
select stock_value into ans from stockinformation where stock_id='S1';
ans := ans-cur;
raise notice 'Total Profit and Loss = %',ans;
end;
$$;
```

```
NOTICE:  Total Profit and Loss = 100
DO
```

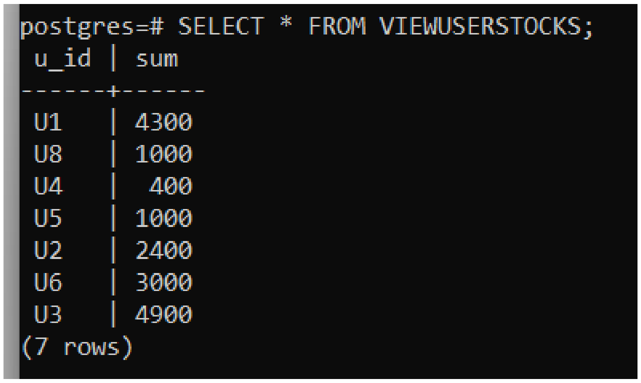
5.9 Views

5.9.1 Create a view to show sum of stocks(networth) held by users.

- View

```
CREATE VIEW VIEWUSERSTOCKS AS
SELECT U_ID,SUM(NET_WORTH) FROM STOCKASSETS GROUP BY
U_ID;
```

Output :



```
postgres=# SELECT * FROM VIEWUSERSTOCKS;
 u_id | sum
------+------
 U1   | 4300
 U8   | 1000
 U4   |  400
 U5   | 1000
 U2   | 2400
 U6   | 3000
 U3   | 4900
(7 rows)
```

6. FUTURE ENHANCEMENTS OF THE SYSTEM

- We will design Front-end Design in HTML , CSS , React, JavaScript and Develop Bank- end in Nodejs.
- For security purpose New Registration is done using OTP.
- We will make database more consistent and We are making this database efficient and easy to implement with huge data capacity.
- Methods and user data input will be lot easy after the implement of GUI.
- We will also add some extra features so that the users can share their portfolio with their friends.

7. BIBLIOGRAPHY

- For the successful implementation of this project we referred to many websites and books.
- The schema was designed by taking ideas from website of moneycontrol.com tradingview.com .
- We created the ER Diagram and Schema Diagram on “draw.io”.
- Mostly we referred the online material for syntax of procedures, triggers, Exception and cursors.

Reference book:
Data Base System Concepts
-Henry F. Korth & A. Silberschatz 2nd Ed. McGraw-Hill 1991

Reference Websites:

- https://www.moneycontrol.com/
- https://www.w3school.com/
- https://www.tutorialspoint.com/
- http://www.mysqltutorial.org/
- http://in.tradingview.com/