

DBMS Assignment 3 Report

Team Details :

Name : Abhishek Aditya BS
SRN : PES1UG19CS019

Name : Adithya M S
SRN : PES1UG19CS027

Name : Abhiram Puranik
SRN : PES1UG19CS018

Simple Queries :

<1> List of Transactions made by an investor.

```
Select dateOfTransaction,transaction_id,name_i,amount,modeOfTransaction
FROM transactions NATURAL JOIN investorsAndTraders
ORDER BY dateOfTransaction;
```

stockexchange=# Select dateOfTransaction,transaction_id,name_i,amount,modeOfTransaction

```
FROM transactions NATURAL JOIN investorsAndTraders
ORDER BY dateOfTransaction;
```

dateoftransaction	transaction_id	name_i	amount	modeoftransaction
2018-09-05	179	Abhiram Puranik	88000.00	Debit Card
2019-01-24	251	Adithya	16000.00	Credit Card
2019-12-18	306	Abhishek	52000.00	Bank Transfer
2019-12-21	353	Chinmay Choudary	45600.00	UPI Transfer
2020-01-06	421	Riya Bannerjee	123000.00	Bank Transfer
2020-02-05	486	Prajwal Desai	8500.00	Credit Card
2020-04-23	542	Supriya Agarwal	76434.33	Bank Transfer
2020-12-04	621	Sai Chaithanya	62307.01	Bank Transfer

(8 rows)

<2> List of stocks held by all investors along with the value of each of them.

```
Select name_i,companyName,currentValue
From holdsStocks hs,investorsAndTraders i,stocks s
Where hs.i_id=i.i_id AND hs.s_id=s.s_id;
```

```
stockexchange=# Select name_i,companyName,currentValue
From holdsStocks hs,investorsAndTraders i,stocks s
Where hs.i_id=i.i_id AND hs.s_id=s.s_id;
      name_i      |      companyname      |      currentvalue
-----+-----+-----
Adithya           | Cred Inc.              |          1784.89
Abhiram Puranik   | JK Lakshmi Cement      |           976.66
Sai Chaithanya    | TCS                    |         9653.67
Prajwal Desai     | Hindustan Uniliver     |         2459.00
Supriya Agarwal   | Relaince               |         2620.00
Chinmay Choudary  | Britania               |         3734.42
Riya Bannerjee    | Sun Pharma             |           825.10
(7 rows)
```

<3> List of brokers associated with an investor.

```
Select name as brokerName,name_i
From investorsAndTraders i,brokers b
Where i.broker_id=b.broker_id;
```

```
stockexchange=# Select name as brokerName,name_i
From investorsAndTraders i,brokers b
Where i.broker_id=b.broker_id;
      brokername      |      name_i
-----+-----
Zerodha              | Abhishek
Upstox               | Adithya
Groww                | Abhiram Puranik
Angel Broking        | Sai Chaithanya
ICICI Direct         | Prajwal Desai
HDFC Securities      | Supriya Agarwal
Kotak Securities     | Chinmay Choudary
Motilal Oswal        | Riya Bannerjee
(8 rows)
```

<4> Stocks handled by a broker “Groww”.

```
Select companyName,currentValue
From brokers b, handlesStocks hs,stocks s
Where b.broker_id=hs.broker_id AND hs.s_id=s.s_id AND b.name='Groww';
```

```
stockexchange=# Select companyName,currentValue
```

```
From brokers b, handlesStocks hs,stocks s
```

```
Where b.broker_id=hs.broker_id AND hs.s_id=s.s_id AND b.name='Groww';
```

companyname	currentvalue
JK Lakshmi Cement	976.66
Cred Inc.	1784.89
Relaince	2620.00
Hindustan Uniliver	2459.00
Britania	3734.42
Sun Pharma	825.10
TCS	9653.67

(7 rows)

<5> Bank account information of an investor.

```
Select name_i,bankName,accountNumber,ifsc,branch
From investorsAndTraders i,banks b
Where b.i_id=i.i_id AND i.name_i='Abhishek';
```

```
stockexchange=# Select name_i,bankName,accountNumber,ifsc,branch
```

```
From investorsAndTraders i,banks b
```

```
Where b.i_id=i.i_id AND i.name_i='Abhishek';
```

name_i	bankname	accountnumber	ifsc	branch
Abhishek	Kotak Bank	8816228345	KB01248234	Basavangudi

(1 row)

Nested queries:

<1> Stocks that constitute a mutual fund “Bajaj Alliance growth Fund”

```
Select companyName,currentValue,totalQuantity
```

```
From stocks s
```

```
Where s_id in (
```

```
    select s_id
```

```
    from mutualFunds mf,isCollectionsOf ico
```

```
    where mf.m_id=ico.m_id and mf.name='Bajaj Alliance growth Fund'
```

```
);
```

```
stockexchange=# Select companyName,currentValue,totalQuantity
```

```
From stocks s
```

```
Where s_id in (
```

```
    select s_id
```

```
    from mutualFunds mf,isCollectionsOf ico
```

```
    where mf.m_id=ico.m_id and mf.name='Bajaj Alliance growth Fund'
```

```
);
```

companyname	currentvalue	totalquantity
Cred Inc.	3000.00	897000
TCS	9653.67	861002
JK Lakshmi Cement	976.66	250000

(3 rows)

QUERY PLAN

```
Nested Loop (cost=51.62..56.64 rows=13 width=212)
-> HashAggregate (cost=51.48..51.61 rows=13 width=4)
    Group Key: ico.s_id
    -> Nested Loop (cost=2.91..51.44 rows=13 width=4)
        -> Seq Scan on mutualfunds mf (cost=0.00..16.75 rows=3 width=4)
            Filter: ((name)::text = 'Bajaj Alliance growth Fund'::text)
        -> Bitmap Heap Scan on iscollectionsof ico (cost=2.91..11.45 rows=11 width=8)
            Recheck Cond: (m_id = mf.m_id)
            -> Bitmap Index Scan on iscollectionsof_pkey (cost=0.00..2.90 rows=11 width=0)
                Index Cond: (m_id = mf.m_id)
    -> Index Scan using stocks_pkey on stocks s (cost=0.15..0.41 rows=1 width=216)
        Index Cond: (s_id = ico.s_id)
(12 rows)
```


<3> Display the updated margin after adding the amount deposited to the wallet.

```
SELECT name_i,marginAvailable + amount as updatedMargin
FROM investorsAndTraders NATURAL JOIN transactions
WHERE i_id in (Select i_id from transactions WHERE
typeOfTransaction='Deposit');

stockexchange=# SELECT name_i,marginAvailable + amount as updatedMargin
FROM investorsAndTraders NATURAL JOIN transactions
WHERE i_id in (Select i_id from transactions WHERE typeOfTransaction='Deposit');

 name_i | updatedmargin 
-----+-----
Abhiram Puranik | 98000.00
Adithya | 26000.00
Chinmay Choudary | 55600.00
Prajwal Desai | 18500.00
(4 rows)
```

```
QUERY PLAN
-----
Nested Loop (cost=14.94..31.23 rows=4 width=110)
-> Hash Join (cost=14.80..30.06 rows=4 width=24)
    Hash Cond: (transactions.i_id = transactions_1.i_id)
    -> Seq Scan on transactions (cost=0.00..13.80 rows=380 width=20)
    -> Hash (cost=14.77..14.77 rows=2 width=4)
        -> Unique (cost=14.76..14.77 rows=2 width=4)
            -> Sort (cost=14.76..14.77 rows=2 width=4)
                Sort Key: transactions_1.i_id
                -> Seq Scan on transactions transactions_1 (cost=0.00..14.75 rows=2 width=4)
                    Filter: ((typeof(transaction)::text = 'Deposit'::text))
    -> Index Scan using investorsandtraders_pkey on investorsandtraders (cost=0.14..0.29 rows=1 width=98)
        Index Cond: (i_id = transactions.i_id)
(12 rows)
```

<4> Find Stock Name, quantity and current value of a stock with value >2000 and quantity >400000.

```
SELECT companyName,totalQuantity,currentValue
FROM stocks
WHERE currentValue>'2000' INTERSECT (
    SELECT companyName,totalQuantity,currentValue
    FROM stocks
    WHERE totalQuantity>'400000'
);
```

```
stockexchange=# SELECT companyName,totalQuantity,currentValue
FROM stocks
WHERE currentValue>'2000' INTERSECT (
    SELECT companyName,totalQuantity,currentValue
    FROM stocks
    WHERE totalQuantity>'400000'
);
```

companyname	totalquantity	currentvalue
Cred Inc.	897000	3000.00
Hindustan Uniliver	452881	2459.00
TCS	861002	9653.67
Britania	765921	3000.00

(4 rows)

```
QUERY PLAN
-----
HashSetOp Intersect (cost=0.00..31.18 rows=86 width=216)
-> Append (cost=0.00..29.79 rows=186 width=216)
    -> Subquery Scan on "*SELECT* 1" (cost=0.00..14.43 rows=93 width=216)
        -> Seq Scan on stocks (cost=0.00..13.50 rows=93 width=212)
            Filter: (currentvalue > '2000'::numeric)
    -> Subquery Scan on "*SELECT* 2" (cost=0.00..14.43 rows=93 width=216)
        -> Seq Scan on stocks stocks_1 (cost=0.00..13.50 rows=93 width=212)
            Filter: ((totalquantity)::text > '400000'::text)
(8 rows)
```

<5> Display the list of stocks not held by investor 'Abhiram Puranik'.

```
SELECT companyName,currentValue
FROM Stocks
WHERE s_id in (
    select s_id
    from holdsStocks EXCEPT select s_id
    from holdsStocks where i_id in (
        select i_id from investorsAndTraders where name_i = 'Abhiram Puranik')
);
```

```

stockexchange=# SELECT companyName,currentValue
FROM Stocks
WHERE s_id in (
    select s_id
    from holdsStocks EXCEPT select s_id
    from holdsStocks where i_id in (
        select i_id from investorsAndTraders where name_i = 'Abhiram Puranik')
);

```

companyname	currentvalue
Sun Pharma	825.10
TCS	9653.67
Britania	3000.00
Relaince	2620.00
Cred Inc.	3000.00
Hindustan Uniliver	2459.00

(6 rows)

```

-----
QUERY PLAN
-----
Hash Join (cost=16.30..112.26 rows=200 width=134)
  Hash Cond: ("ANY_subquery".s_id = stocks.s_id)
    -> Subquery Scan on "ANY_subquery" (cost=0.00..95.43 rows=200 width=4)
      -> HashSetOp Except (cost=0.00..93.43 rows=200 width=8)
        -> Append (cost=0.00..88.30 rows=2051 width=8)
          -> Subquery Scan on "+SELECT* 1" (cost=0.00..50.80 rows=2040 width=8)
            -> Seq Scan on holdsstocks (cost=0.00..30.40 rows=2040 width=4)
          -> Subquery Scan on "+SELECT* 2" (cost=4.23..27.25 rows=11 width=8)
            -> Nested Loop (cost=4.23..27.14 rows=11 width=4)
              -> Seq Scan on investorsandtraders (cost=0.00..12.25 rows=1 width=4)
                Filter: ((name_i)::text = 'Abhiram Puranik'::text)
              -> Bitmap Heap Scan on holdsstocks holdsstocks_1 (cost=4.23..14.79 rows=10 width=8)
                Recheck Cond: (i_id = investorsandtraders.i_id)
                -> Bitmap Index Scan on holdsstocks_pkey (cost=0.00..4.23 rows=10 width=0)
                  Index Cond: (i_id = investorsandtraders.i_id)
            -> Hash (cost=12.80..12.80 rows=280 width=138)
              -> Seq Scan on stocks (cost=0.00..12.80 rows=280 width=138)

```

(17 rows)

Complex Queries:

<1> Display the list of all holdings of an investor 'Adithya'.

```
SELECT i.i_id, c.name as CommodityName, s.companyName as
StockName, bond.name as BondName
FROM investorsAndTraders i, holdsCommodities hc, commodities c, holdsStocks
hs, stocks s, holdsBonds hb, bonds bond
WHERE i.name_i='Adithya' AND i.i_id=hc.i_id AND hc.c_id=c.c_id AND
i.i_id=hs.i_id AND hs.s_id=s.s_id AND i.i_id=hb.i_id AND
hb.b_id=bond.b_id;
```

```
FROM investorsAndTraders i, holdsCommodities hc, commodities c, holdsStock
s hs, stocks s, holdsBonds hb, bonds bond

WHERE i.name_i='Adithya' AND i.i_id=hc.i_id AND hc.c_id=c.c_id AND i.i_
id=hs.i_id AND hs.s_id=s.s_id AND i.i_id=hb.i_id AND hb.b_id=bond.b_id;
```

i_id	commodityname	stockname	bondname
2	GOLD	Cred Inc.	REC CAPITAL GAIN BOND

(1 row)

<2> Buy functionality

Drive Link to the 'buyDDL' file: [BuyDDL.sql](#)

Initial holdings of investors:

```
stockexchange=# TABLE holdsstocks;
 i_id | s_id | quantity
-----+-----+-----
    1 |   10 |       102
    2 |   30 |        38
    3 |   20 |        77
    4 |   80 |        45
    5 |   50 |        90
    6 |   40 |        32
    7 |   60 |        65
    8 |   70 |        62
(8 rows)
```

Buy(buyer_id, stock_id, Buy price, Buy quantity)

```
stockexchange=# select buy(1,70, 82, 10);
```

```
buy
```

```
-----
```

```
2
```

```
(1 row)
```

```
stockexchange=# TABLE buy_table;
```

```
buyer_id | stock_id | buy_price | buy_quantity
```

```
-----+-----+-----+-----
```

```
1 | 70 | 82.00 | 10
```

```
(1 row)
```

```
stockexchange=# TABLE sell_table;
```

```
seller_id | stock_id | sell_price | sell_quantity
```

```
-----+-----+-----+-----
```

```
(0 rows)
```

Buy order is taken and put into the waiting list in the buy_table as there is no one selling that stock at that price point.

Conditions handled by Buy functionality:

1. Buyer must hold enough margin to buy number of particular stock which is calculated from the stocks table.
2. Seller must be present for the particular stock to be bought with the same selling price as of buying price, or else the buy order will go to waiting.
3. Buyer cannot sell stocks to himself.
4. Sell-buy quantity conditions:

If sell quantity is same as buy quantity, waiting is completely removed, seller's and buyer's holdings are updated.

If sell quantity is greater than buy quantity, remaining sell stocks are left in waiting, seller's and buyer's holdings are updated.

If sell quantity is less than buy quantity, remaining buy stocks are left waiting, seller's waiting is completely removed, seller's and buyer's holdings are updated.

<3>Sell functionality

Drive Link to the 'sellDDL' file: [sellDDL.sql](#)

Sell(Seller_id, stock_id, Sell price, Sell quantity)

```
stockexchange=# select sell(8,70, 82, 15);
NOTICE:  Value: true
 sell
-----
      1
(1 row)
```

```
stockexchange=# TABLE buy_table;
 buyer_id | stock_id | buy_price | buy_quantity
-----+-----+-----+-----
(0 rows)

stockexchange=# TABLE sell_table;
 seller_id | stock_id | sell_price | sell_quantity
-----+-----+-----+-----
          8 |         70 |        82.00 |          5
(1 row)
```

Sell order executed. As there user 8 was willing to sell 15 stocks while user 1 was willing to buy 10 stocks, program handles buy transaction of 10 stocks only and the rest 5 stocks are put into the Sell table for waiting.

```
stockexchange=# TABLE holdsstocks;
```

i_id	s_id	quantity
1	10	102
2	30	38
3	20	77
4	80	45
5	50	90
6	40	32
7	60	65
1	70	10
8	70	52

(9 rows)

User 1 has 10 stocks of stock id 70, while user 8 has 10 stocks deducted from his holdings

Conditions handled by sell functionality:

1. Sellers can only sell if he/she holds the stock and the sell quantity should be less than or equal to their holdings.
2. Buyer must be present for the transaction to occur, if no buyer, then the sell order will be put on the waiting list in the sell_table.
3. If the buyer's quantity of the particular stock with same buy price as of the sell price, is equal to seller's quantity, the buyer's waiting will be removed, seller's and buyer's holdings are updated.
4. If the buyer's quantity is greater than seller's quantity, then the transaction occurs with the buyer's remaining stocks still waiting to be sold, seller's and buyer's holdings are updated.
5. If the buyer's quantity is less than the seller's quantity, the remaining sell stocks are put on waiting, seller's and buyer's holdings are updated.

<4> Display the list of all entities handled by the broker 'Zerodha'.

```
SELECT b.broker_id, c.name as CommodityName,mf.name as
MutualFundName,s.companyName as StockName,bond.name as BondName
FROM brokers b,handlesCommodities hc,commodities c,handlesMF
hmf,mutualFunds mf,handlesStocks hs,stocks s,handlesBonds hb,bonds bond
WHERE b.name='Zerodha' AND b.broker_id=hc.broker_id AND hc.c_id=c.c_id AND
b.broker_id=hmf.broker_id AND hmf.m_id=mf.m_id AND
b.broker_id=hs.broker_id AND hs.s_id=s.s_id AND b.broker_id=hb.broker_id
AND hb.b_id=bond.b_id;
```

broker_id	commodityname	mutualfundname	stockname	bondname
55	GOLD	Axis Bank Prduential Fund	JK Lakshmi Cement	REC CAPITAL GAIN BOND
55	GOLD	Axis Bank Prduential Fund	Cred Inc.	REC CAPITAL GAIN BOND
55	GOLD	Axis Bank Prduential Fund	Relaince	REC CAPITAL GAIN BOND
55	GOLD	Axis Bank Prduential Fund	Hindustan Uniliver	REC CAPITAL GAIN BOND
55	GOLD	Axis Bank Prduential Fund	Britania	REC CAPITAL GAIN BOND
55	GOLD	Axis Bank Prduential Fund	Sun Pharma	REC CAPITAL GAIN BOND
55	GOLD	Axis Bank Prduential Fund	TCS	REC CAPITAL GAIN BOND
55	GOLD	High Growth Momentum Fund	JK Lakshmi Cement	REC CAPITAL GAIN BOND
55	GOLD	High Growth Momentum Fund	Cred Inc.	REC CAPITAL GAIN BOND
55	GOLD	High Growth Momentum Fund	Relaince	REC CAPITAL GAIN BOND
55	GOLD	High Growth Momentum Fund	Hindustan Uniliver	REC CAPITAL GAIN BOND
55	GOLD	High Growth Momentum Fund	Britania	REC CAPITAL GAIN BOND
55	GOLD	High Growth Momentum Fund	Sun Pharma	REC CAPITAL GAIN BOND
55	GOLD	High Growth Momentum Fund	TCS	REC CAPITAL GAIN BOND
55	GOLD	Bajaj Alliance growth Fund	JK Lakshmi Cement	REC CAPITAL GAIN BOND
55	GOLD	Bajaj Alliance growth Fund	Cred Inc.	REC CAPITAL GAIN BOND
55	GOLD	Bajaj Alliance growth Fund	Relaince	REC CAPITAL GAIN BOND
55	GOLD	Bajaj Alliance growth Fund	Hindustan Uniliver	REC CAPITAL GAIN BOND
55	GOLD	Bajaj Alliance growth Fund	Britania	REC CAPITAL GAIN BOND
55	GOLD	Bajaj Alliance growth Fund	Sun Pharma	REC CAPITAL GAIN BOND
55	GOLD	Bajaj Alliance growth Fund	TCS	REC CAPITAL GAIN BOND
55	GOLD	Kotak Savings Fund	JK Lakshmi Cement	REC CAPITAL GAIN BOND

<5> Insert a new stock called "PESU" to stocks, add this stock as a constituent of Mutual fund with id 1024 and also add this stock to the holdings of an investor with id 1.

```
WITH Y as (
    INSERT INTO stocks
    VALUES (90,'PESU',100,50000,'Small Cap')
    returning s_id
), X as (
    INSERT INTO isCollectionsOf
    SELECT 1024,s_id
    FROM Y)
INSERT INTO holdsStocks
SELECT 1,s_id,10
FROM Y;
```

```

stockexchange=# WITH Y as (
    INSERT INTO stocks
    VALUES (90,'PESU',100,50000,'Small Cap')
    returning s_id
), X as (
    INSERT INTO isCollectionsOf
    SELECT 1024,s_id
    FROM Y)
INSERT INTO holdsStocks
SELECT 1,s_id,10
FROM Y;
INSERT 0 1

```

Multiple Users with different access privileges:

```

CREATE USER admin with password '1234' createdb;
grant ALL PRIVILEGES ON ALL TABLES IN SCHEMA public TO admin;

CREATE USER database_supervisor with password '1234' createdb;
grant select,insert,update,delete on ALL TABLES IN SCHEMA public TO
database_supervisor;

CREATE USER investor with password '1234' createdb;
grant select on ALL TABLES IN SCHEMA public TO investor;
grant insert,update,delete on
transactions,holdsCommodities,holdsBonds,holdsStocks,holdsMF TO investor;

CREATE USER broker with password '1234' createdb;
grant select on ALL TABLES IN SCHEMA public TO broker;
grant insert,update,delete on
phoneNumber,handlesBonds,handlesCommodities,handlesMF,handlesStocks TO
broker;

```

Transactions:

<1> Insert 3 records into transactions and then create a savepoint, insert 2 other records, then roll back to the savepoint and finally commit the transaction.

```
begin;
insert into transactions values(
    654,
    9430.00,
    '2021-01-09',
    'Debit Card',
    'Deposit',
    4
);
insert into transactions values(
    689,
    4320.08,
    '2021-02-21',
    'Bank Transfer',
    'Withdraw',
    8
);
insert into transactions values(
    703,
    13087.76,
    '2021-02-25',
    'UPI Transfer',
    'Deposit',
    8
);
select * from transactions;
savepoint s1;
insert into transactions values(
    708,
    98711.00,
    '2021-03-01',
    'XXXXXX XXXXX',
    'Deposit',
    8
);
```

```
);
insert into transactions values(
    709,
    16723.00,
    '2021-03-01',
    'XXXXXX XXXXX',
    'Deposit',
    8
);
rollback to savepoint s1;
END;
select * from transactions;
```

```
BEGIN
INSERT 0 1
INSERT 0 1
INSERT 0 1
transaction_id | amount | dateoftransaction | modeoftransaction | typeoftransaction | i_id
-----+-----+-----+-----+-----+-----
179 | 88000.00 | 2018-09-05 | Debit Card | Deposit | 3
251 | 16000.00 | 2019-01-24 | Credit Card | Deposit | 2
306 | 52000.00 | 2019-12-18 | Bank Transfer | Withdraw | 1
353 | 45600.00 | 2019-12-21 | UPI Transfer | Deposit | 7
421 | 123000.00 | 2020-01-06 | Bank Transfer | Withdraw | 8
486 | 8500.00 | 2020-02-05 | Credit Card | Deposit | 5
542 | 76434.33 | 2020-04-23 | Bank Transfer | Withdraw | 6
621 | 62307.01 | 2020-12-04 | Bank Transfer | withdraw | 4
654 | 9430.00 | 2021-01-09 | Debit Card | Deposit | 4
689 | 4320.08 | 2021-02-21 | Bank Transfer | Withdraw | 8
703 | 13087.76 | 2021-02-25 | UPI Transfer | Deposit | 8
```

(11 rows)

```
SAVEPOINT
INSERT 0 1
INSERT 0 1
ROLLBACK
COMMIT
```

```
transaction_id | amount | dateoftransaction | modeoftransaction | typeoftransaction | i_id
-----+-----+-----+-----+-----+-----
179 | 88000.00 | 2018-09-05 | Debit Card | Deposit | 3
251 | 16000.00 | 2019-01-24 | Credit Card | Deposit | 2
306 | 52000.00 | 2019-12-18 | Bank Transfer | Withdraw | 1
353 | 45600.00 | 2019-12-21 | UPI Transfer | Deposit | 7
421 | 123000.00 | 2020-01-06 | Bank Transfer | Withdraw | 8
486 | 8500.00 | 2020-02-05 | Credit Card | Deposit | 5
542 | 76434.33 | 2020-04-23 | Bank Transfer | Withdraw | 6
621 | 62307.01 | 2020-12-04 | Bank Transfer | withdraw | 4
654 | 9430.00 | 2021-01-09 | Debit Card | Deposit | 4
689 | 4320.08 | 2021-02-21 | Bank Transfer | Withdraw | 8
703 | 13087.76 | 2021-02-25 | UPI Transfer | Deposit | 8
```

(11 rows)

<2> Insert 2 phone numbers and then create a savepoint, then insert a duplicate entry and then rollback

```
begin;
insert into phoneNumber values(
    66,
    '+91 8762378178'
);
insert into phoneNumber values(
    222,
    '+91 8972837634'
);
savepoint s1;
insert into phoneNumber values(
    66,
    '+91 8762378178'
);
rollback to savepoint s1;
insert into phoneNumber values(
    111,
    '+91 8423452578'
);
insert into phoneNumber values(
    88,
    '+91 8726387645'
);
savepoint s2;
insert into phoneNumber values(
    111,
    '+91 8423452578'
);
insert into phoneNumber values(
    88,
    '+91 8726387645'
);
rollback to savepoint s2;
end;
select * from phoneNumber;
```

Login to user “broker”

```
(base) abhishek_aditya_bs@pop-os:~/Desktop/DBMS Assignment 3$ psql -h localhost -d stockexchange -U broker -p 5432
Password for user broker:
psql (13.4 (Ubuntu 13.4-0ubuntu0.21.04.1))
SSL connection (protocol: TLSv1.3, cipher: TLS_AES_256_GCM_SHA384, bits: 256, compression: off)
Type "help" for help.
```

```
BEGIN
INSERT 0 1
INSERT 0 1
SAVEPOINT
ERROR:  duplicate key value violates unique constraint "phonenum_pkey"
DETAIL:  Key (broker_id, p_no)=(66, +91 8762378178) already exists.
ROLLBACK
INSERT 0 1
INSERT 0 1
SAVEPOINT
ERROR:  duplicate key value violates unique constraint "phonenum_pkey"
DETAIL:  Key (broker_id, p_no)=(111, +91 8423452578) already exists.
ERROR:  current transaction is aborted, commands ignored until end of transaction block
ROLLBACK
COMMIT
```

broker_id	p_no
55	+91 8627386451
55	080 26765180
66	+91 7722871925
66	080 46681627
77	+91 8819263371
77	080 61128638
88	+91 9982739201
88	080 22346718
99	+91 8262328721
99	080 736349211
111	+91 887392910371
111	080 278382911
222	+91 88273829122
222	080 22638627
333	+91 9872918211
333	080 44829716
66	+91 8762378178
222	+91 8972837634
111	+91 8423452578
88	+91 8726387645

(20 rows)

Login to user “investor”:

```
(base) abhishek_aditya_bs@pop-os:~/Desktop/DBMS Assignment 3$ psql -h localhost -d stockexchange -U investor -p 5432
Password for user investor:
psql (13.4 (Ubuntu 13.4-0ubuntu0.21.04.1))
SSL connection (protocol: TLSv1.3, cipher: TLS_AES_256_GCM_SHA384, bits: 256, compression: off)
Type "help" for help.
```

```
BEGIN
ERROR: permission denied for table phonenumber
ERROR: current transaction is aborted, commands ignored until end of transaction block
ERROR: current transaction is aborted, commands ignored until end of transaction block
ERROR: current transaction is aborted, commands ignored until end of transaction block
ERROR: savepoint "s1" does not exist
ERROR: current transaction is aborted, commands ignored until end of transaction block
ERROR: current transaction is aborted, commands ignored until end of transaction block
ERROR: current transaction is aborted, commands ignored until end of transaction block
ERROR: current transaction is aborted, commands ignored until end of transaction block
ERROR: current transaction is aborted, commands ignored until end of transaction block
ERROR: savepoint "s2" does not exist
ROLLBACK
 broker_id |      p_no
-----+-----
      55 | +91 8627386451
      55 | 080 26765180
      66 | +91 7722871925
      66 | 080 46681627
      77 | +91 8819263371
      77 | 080 61128638
      88 | +91 9982739201
      88 | 080 22346718
      99 | +91 8262328721
      99 | 080 736349211
     111 | +91 887392910371
     111 | 080 278382911
     222 | +91 88273829122
     222 | 080 22638627
     333 | +91 9872918211
     333 | 080 44829716
       66 | +91 8762378178
     222 | +91 8972837634
     111 | +91 8423452578
       88 | +91 8726387645
(20 rows)
```

User “investor” does not have access to the ‘phoneNumber’ table which results in permission denied and the operation is aborted.

Postgres code folder:

- [Drive Link for the postgres code](#)

Contribution and Time Spent:

Abhiram Puranik - Worked on 'sell' and 'buy' complex functionalities and handled multiple edge cases.

(10 hrs)

Abhishek Aditya BS - Worked on nested and complex queries.

(6 hrs)

Adithya M S - Worked on logic of all the queries and report writeup.

(6 hrs)