UE18CS252

Database Management System

PROJECT REPORT

Prof. Priya Badrinath

Section F

B.tech CSE

STOCK MARKET MANAGEMENT

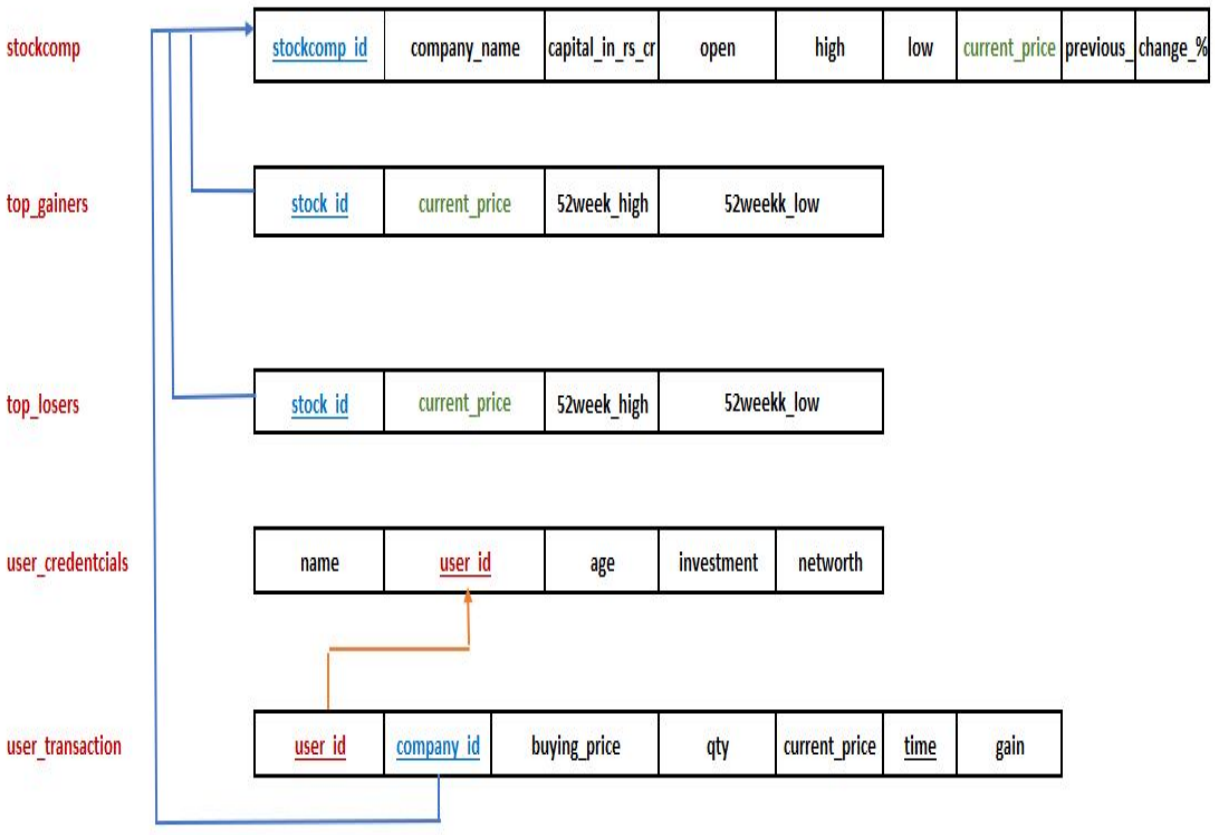
PES1201801127 G JAHNAVI REDDY

Table of contents

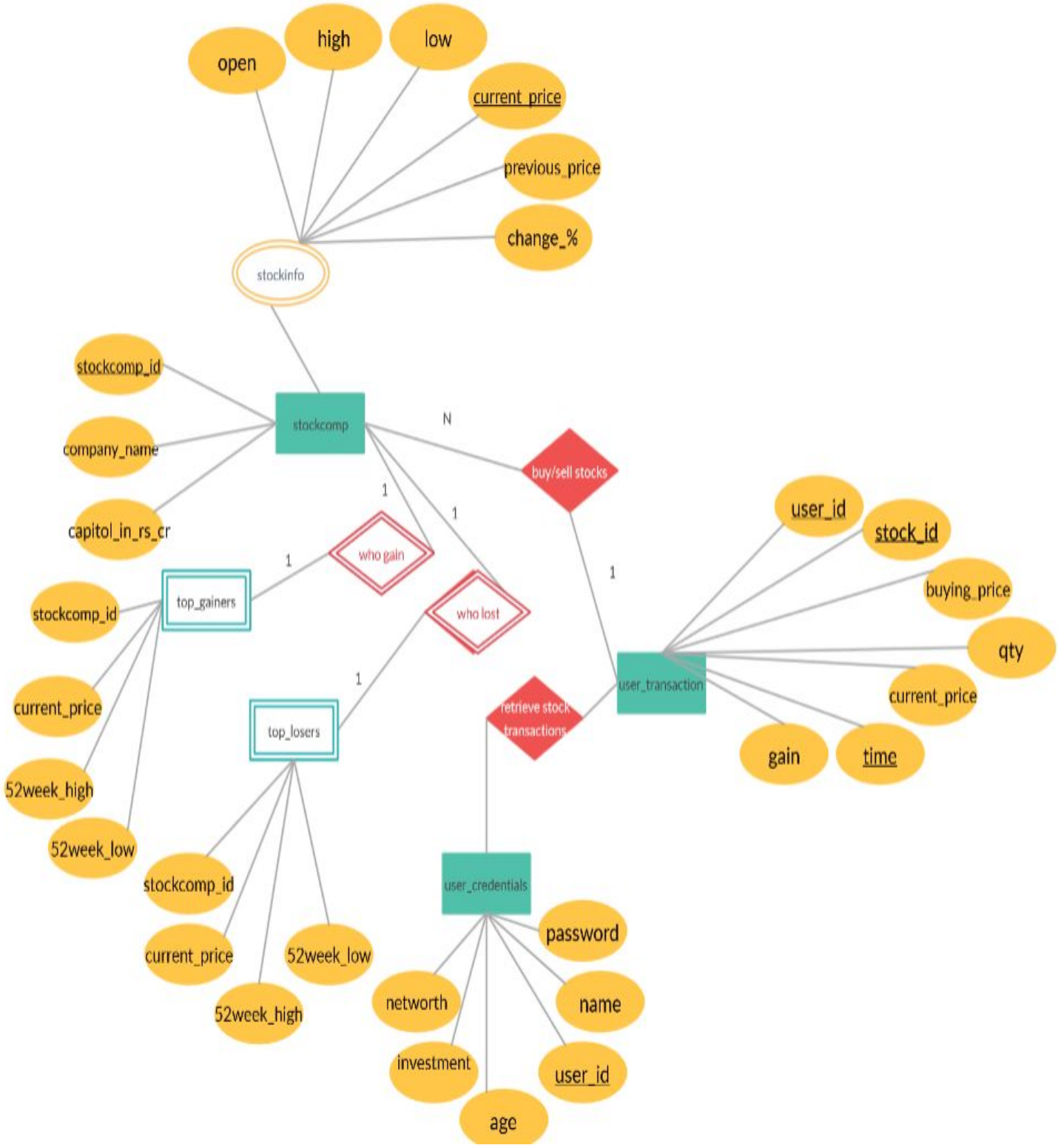
Introduction

Stocks deal with huge volumes of data which is accessed very often for the users to trade. This database management system is intended to provide users with easier access to the stock prices and other information required to trade.

Stock Market Database Schema



ER Diagram


open
high
low
current_price
previous_price
change_%
stockinfo
stockcomp_id
company_name
capitol_in_rs_cr
stockcomp
N
buy/sell stocks
1
1
who gain
1
top_gainers
stockcomp_id
current_price
52week_high
52week_low
who lost
1
top_losers
stockcomp_id
current_price
52week_high
52week_low
1
retrieve stock transactions
user_transaction
user_id
stock_id
buying_price
qty
current_price
gain
time
user_credentials
password
name
user_id
networth
investment
age

Functional Dependencies

1) {stockcomp_id} →{ Company_name ,

   capitol_in_rs_cr,open,high,low,current_price,previous_price,change_percentage}

2) {user_id}→{name,age,investment,networth,password}

3) {user_id,company_id,time)→{buying_price,qty,current_price,gain}

Candidate Keys

To find the candidate key, the attribute closure of all the subsets of the relation is looked at. From this, the super key is determined. Super keys are those where the attribute closure contains all the attributes of the relation. Candidate keys are then identified by checking if none of the subsets of the super key is a super key by itself.

By following the above-mentioned procedure, we get, the keys as follows:

1) stockcomp : stockcomp_id
2) user_credentials: user_id
3) user_transaction: user_id,company_id,time

Normalization

First normal form (1NF)

As per the rule of first normal form, an attribute (column) of a table cannot hold multiple values. It should hold only atomic values.

user_id	company_id	Buying price	qty	current_price	time	gain
101	Reliance	1230	1000	1561.8	2020-03-29 10:23:02	26.95
102	Reliance HUL	1400 2213	1300 150	1561.8 2089.45	2020-02-04 9:31:32 2020-03-24 12:33:33	11.55 -5.58
103	Reliance	1200	1300	1561.8	2020-03-29 09:17:09	30.15

Use_id 102 has bought stocks in more than one company.

This table is not in 1NF as the rule says “each attribute of a table must have atomic (single) values”.

To make the table complies with 1NF we should have the data like this.

user_id	company_id	Buying price	qty	current_price	time	gain
101	Reliance	1230	1000	1561.8	2020-03-29 10:23:02	26.95
102	Reliance	1400	1300	1561.8	2020-02-04 9:31:33	11.55
102	HUL	2213	150	2089.45	2020-09-24 12:33:33	-5.58
103	Reliance	1200	1300	1561.8	2020-03-29	30.15

Second normal form (2NF)

A table is said to be in 2NF if both the following conditions hold:

- Table is in 1NF (First normal form)
- No non-prime attribute is dependent on the proper subset of any candidate key of table.

An attribute that is not part of any candidate key is known as non-prime attribute.

stock_id	company_name	capitol	open	high	Low	current_price	previous_price	change %
Reliance	Reliance Industries Ltd	990088	1545	1579.9	1537.1	1561.8	1506.95	3.64
HUL	Hindustan Uniliver Ltd	490561	2074.95	2098	2036	2089.45	1992.05	4.89

To make the table complies with 2NF we can break it in two tables like this:

stock_id	company_name	capitol
Reliance	Reliance Industries Ltd	990088
HUL	Hindustan Uniliver Ltd	490561

stock_id	open	high	Low	current_price	previous_price	change%
Reliance	1545	1579.9	1537.1	1561.8	1506.95	3.64
HUL	2074.95	2098	2036	2089.45	1992.05	4.89

Third Normal form (3NF)

A table design is said to be in 3NF if both the following conditions hold:

- Table must be in 2NF
- Transitive functional dependency of non-prime attribute on any super key should be removed.

An attribute that is not part of any candidate key is known as non-prime attribute.

In other words 3NF can be explained like this: A table is in 3NF if it is in 2NF and for each functional dependency X-> Y at least one of the following conditions hold:

- X is a super key of table
- Y is a prime attribute of table

An attribute that is a part of one of the candidate keys is known as a prime attribute.

Boyce Codd normal form (BCNF)

It is an advanced version of 3NF that's why it is also referred as 3.5NF. BCNF is stricter than 3NF. A table complies with BCNF if it is in 3NF and for every functional dependency X->Y, X should be the super key of the table.

TESTING LOSSLESS JOIN PROPERTY

Decomposition of a relation is done when a relation in relational model is not in appropriate normal form. Relation R is decomposed into two or more relations if decomposition is lossless join as well as dependency preserving.

Lossless Join Decomposition

If we decompose a relation R into relations R1 and R2,

- Decomposition is lossy if $R1 \bowtie R2 \supset R$
- Decomposition is lossless if $R1 \bowtie R2 = R$

To check for lossless join decomposition using FD set, following conditions must hold:

1. Union of Attributes of R1 and R2 must be equal to attribute of R. Each attribute of R must be either in R1 or in R2.
   $Att(R1) \cup Att(R2) = Att(R)$

2. Intersection of Attributes of R1 and R2 must not be NULL.
   $Att(R1) \cap Att(R2) \neq \Phi$

3. Common attribute must be a key for at least one relation (R1 or R2)
   $Att(R1) \cap Att(R2) \rightarrow Att(R1)$ or $Att(R1) \cap Att(R2) \rightarrow Att(R2)$

Checking lossless join property on stock market database system, we can conclude that relation has lossy decomposition

Screenshots of tables

```
mysql> select *from stockcomp;
```

stockcomp_id	company_name	capitol_in_rs_cr	open	high	low	current_price	previous_price	change_percentage
Asianpaint	Asian Paints Ltd	151351	1611.8	1638	1570	1578.4	1594.3	-1
AXISBANK	Axis Bank Ltd	107868	403.5	409.45	379.65	382.05	387.35	-3.85
Cadilahc	Cadila Healthcare Ltd	32857	323.5	327.45	318	321.25	321.2	0.02
DRREDDY	Dr Reddys Laboratories Ltd	66193	3900.1	4132.2	3900.1	3984	3837.65	3.81
HDFCBANK	HDFC Bank Ltd	509430	942	943.95	925.2	929.05	925	0.44
HUL	Hindustan Uniliver Ltd	490651	2074.95	2098	2036	2089.45	1992.05	4.89
Infy	Infosys Ltd	287332	672.25	680.9	668.5	674.2	664.95	1.39
IOC	Indian Oil Corporation Ltd	69994	76.1	77	74.05	74.35	75.75	-1.85
ITC	ITC Ltd	194586	163.95	164.15	157.1	158.25	161	-1.71
LT	Larsen & Toubro Ltd	114564	1605.25	1614	1570.1	1584.65	1593.25	-0.54
Marico	Marico Ltd	38731	302.9	304.9	298.25	300.25	298.95	0.43
NESTLEIND	Nestle India Ltd	171601	17350	17820	17201.5	17802.95	17142.45	3.85
Ongc	Oil and Natural Gas Corporation Ltd	95547	76.4	77.2	74.75	76	75.6	0.53
Reliance	Reliance Industries Ltd	990088	1545	1579.9	1537.1	1561.8	1506.95	3.64
SBIN	State Bank of India	148773	172.45	173.8	166.1	166.65	170.75	-2.4
Sunpharma	Sun Pharmacetucial Industries Ltd	112528	460	471	454.05	469	452.2	3.72
TATACONSUM	TATA Consumer Products Ltd	32139	347	355.5	344.1	348.75	341.3	2.18
TATAMOTORS	Tata Motors Ltd	25036	84.5	84.95	80.55	81.05	82.5	-1.76
UPL	UPL Ltd	27864	373.45	375.8	363	364.6	368.4	-1.03
Wipro	Wipro Ltd	105127	185.2	186	182.55	184	181.1	-0.05

```
20 rows in set (0.52 sec)
```

```
mysql> select *from user_Credentials;
+--------------+---------+-----+------------+----------+----------+
| name         | user_id | age | investment | networth | password |
+--------------+---------+-----+------------+----------+----------+
| Anil k       |     101 |  42 |    1561800 |  1230000 | abz      |
| Meghana Rai  |     102 |  33 |    2684990 |  2505600 | abx      |
| Shikha Jain  |     103 |  42 |    1133900 |  1112500 | aby      |
| Sharat Kumar |     104 |  50 |    2731060 |  2465000 | xxx      |
| Malini       |     105 |  47 |   588797.4 |   595200 | yyy      |
+--------------+---------+-----+------------+----------+----------+
5 rows in set (0.00 sec)

mysql> select *from user_transaction;
+---------+------------+--------------+------+---------------+------------+---------------------+
| user_id | company_id | buying_price | qty  | current_price | time       | gain                |
+---------+------------+--------------+------+---------------+------------+---------------------+
|     101 | Reliance   |         1230 | 1000 |        1561.8 | 2020-03-02 |  26.975609756097562 |
|     102 | Asianpaint |         1620 |  150 |        1578.4 | 2020-02-01 |  -2.567901234567901 |
|     102 | HUL        |         2213 |  200 |       2089.45 | 2020-04-12 |  -5.582919114324459 |
|     102 | Reliance   |         1400 | 1300 |        1561.8 | 2020-03-01 |  11.557142857142857 |
|     103 | DRREDDY    |         3750 |  200 |          3984 | 2020-03-22 |                6.24 |
|     103 | Infy       |          725 |  500 |         674.2 | 2020-03-23 |  -7.006896551724138 |
|     104 | Ognc       |           85 | 1000 |            76 | 2020-04-01 | -10.588235294117647 |
|     104 | Reliance   |         1400 | 1700 |        1561.8 | 2020-02-24 |  11.557142857142857 |
|     105 | SBIN       |         1200 |  156 |        166.65 | 2020-04-12 |             -86.1125 |
|     105 | Sunpharma  |          340 | 1200 |           469 | 2020-04-11 |   37.94117647058823 |
+---------+------------+--------------+------+---------------+------------+---------------------+
10 rows in set (0.02 sec)
```

```
mysql> select *from top_gainers;
+------------+---------------+-------------+------------+
| stock_id   | current_price | 52week_high | 52week_low |
+------------+---------------+-------------+------------+
| Asianpaint |        1578.4 |        1915 |    1291.25 |
| DRREDDY    |          3984 |        4099 |     2352.2 |
| HUL        |       2089.45 |        2614 |       1660 |
| Ongc       |            76 |      178.95 |       51.8 |
| Reliance   |        1561.8 |     1602.55 |     867.45 |
+------------+---------------+-------------+------------+
5 rows in set (0.01 sec)

mysql> select *from top_losers;
+------------+---------------+-------------+------------+
| stock_id   | current_price | 52week_high | 52week_low |
+------------+---------------+-------------+------------+
| Infy       |         674.2 |       847.4 |      511.1 |
| IOC        |         74.35 |       170.4 |      71.15 |
| Marico     |        300.25 |       403.7 |      233.8 |
| TATAMOTORS |         81.05 |       201.8 |       63.6 |
| Wipro      |           184 |      301.55 |      159.6 |
+------------+---------------+-------------+------------+
5 rows in set (0.00 sec)
```

Creating Triggers

There are three triggers in the database.

1. Trigger gainper - Calculates the percentage gain of individual stock bought by an individual user. This percentage gain statistic can help users to decide to buy or sell stocks for their own profits
2. Trigger inv -Calculates the total investment made by the user buying all the stocks and updates whenever the user buys new stocks
3. Trigger net- Calculates the total net worth of the users based on the current price of all the stocks bought by him/her.
4.

```
create trigger gainper before insert on user_transaction for each row
 set new.gain=((new.current_price * new.qty)-(new.buying_price * new.qty))*100/(new.buying_price *new.qty);

create trigger net after insert on user_transaction for each row update
user_credentials set networth=networth+(new.buying_price * new.qty) where user_id =new.user_id;

 create trigger inv after insert on user_transaction for each row update
 user_credentials set investment=investment+(new.current_price * new.qty) where user_id =new.user_id;
```

Complex SQL queries

Display the name of the users who have invested in 'Reliance Industries Ltd.

```
mysql> select name from user_credentials as c, user_transaction as i where i.company_id='Reliance' and i.user_id=c.user_id;
+--------------+
| name         |
+--------------+
| Anil k       |
| Meghana Rai  |
| Sharat Kumar |
+--------------+
3 rows in set (0.01 sec)
```

Display the company name which has brought maximum profit to users who bought the stock individually.

```
mysql> select company_name from stockcomp as s, user_transaction as t where t.gain=(select max(gain) from user_transaction) and s.stockcomp_id=t.company_id;
+--------------------+
| company_name       |
+--------------------+
| State Bank of India |
+--------------------+
1 row in set (0.00 sec)

mysql>
```

Display the company name which has brought minimum profit to users who bought the stock individually.

```
mysql> select company_name from stockcomp as s, user_transaction as t where t.gain=(select min(gain) from user_transaction) and s.stockcomp_id=t.company_id;
+--------------+
| company_name |
+--------------+
| ITC Ltd      |
+--------------+
1 row in set (0.00 sec)

mysql>
```