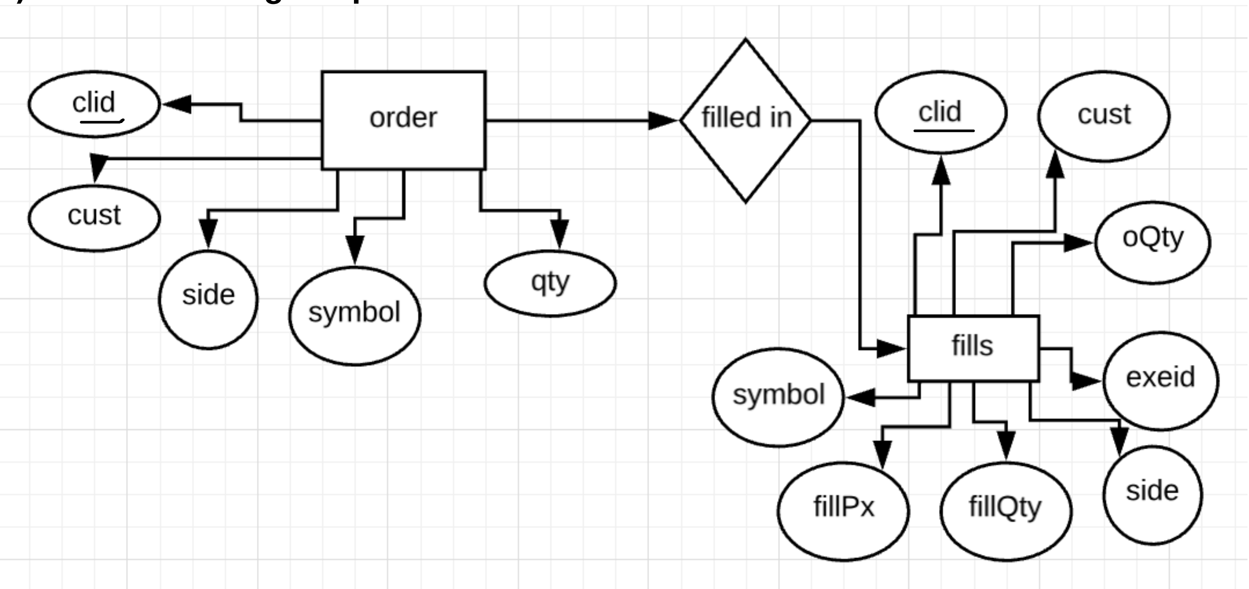
Topgyal Gurung
Discussed with Wei and Mohammed
CSCI 43500-02
October 30, 2018

Q) Customers place orders to buy or sell a certain number of shares in the market.Each order also has their clid unique to each order, across all the customers. These Orders are filled in the market and fills arrive as shown in Fills.Each fill is whole or partial of the order, filled at a price. Each fill may beat a different price (fillPx). fillQty is the number of shares filled at that price.
Each Fill record consists of clid, cust, symbol, slide, oQty,fillQty,fillPx and an execid.

Side= BUY/ SELL

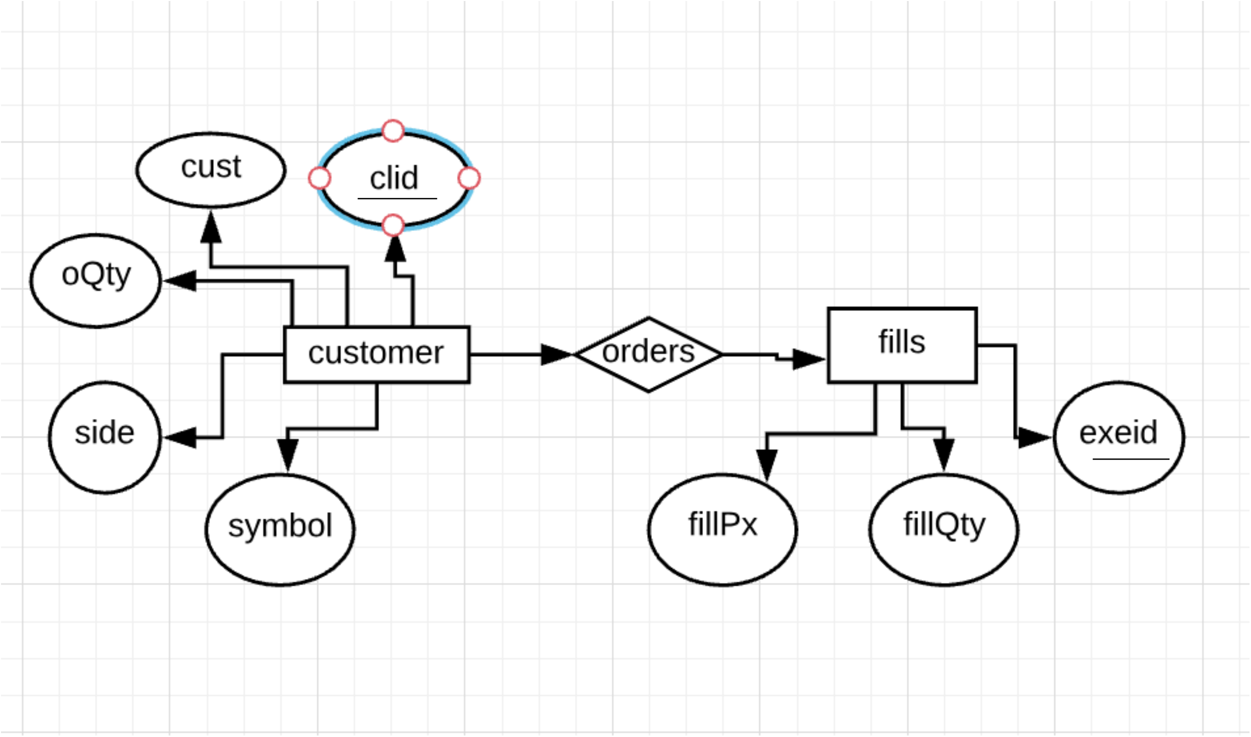
1) Draw a E-R diagram pre-normalization.



In what ways will you optimize it?

We can optimize this by performing normalization since there are redundant data and we don’t need order table. We can just have fills table for this problem.

2) Perform Normalization and draw a E-R diagram and explain why you are doing those normalization steps.



Normalization involves dividing an entity table into two or more tables and defining relationship between the tables. Normalization is performed for eliminating anomaly and minimizing redundancy, and ensure data dependencies make sense. Eliminating anomalies is to remove insert, delete and update anomalies.

First Normal Form (1NF) –

all fields/ attributes are single valued i.e atomic or no repeating groups.

Since customer (order) and fills are have relation we can have clid, cust, oQty, side and symbol for just customer.

Second Normal Form (2NF) –

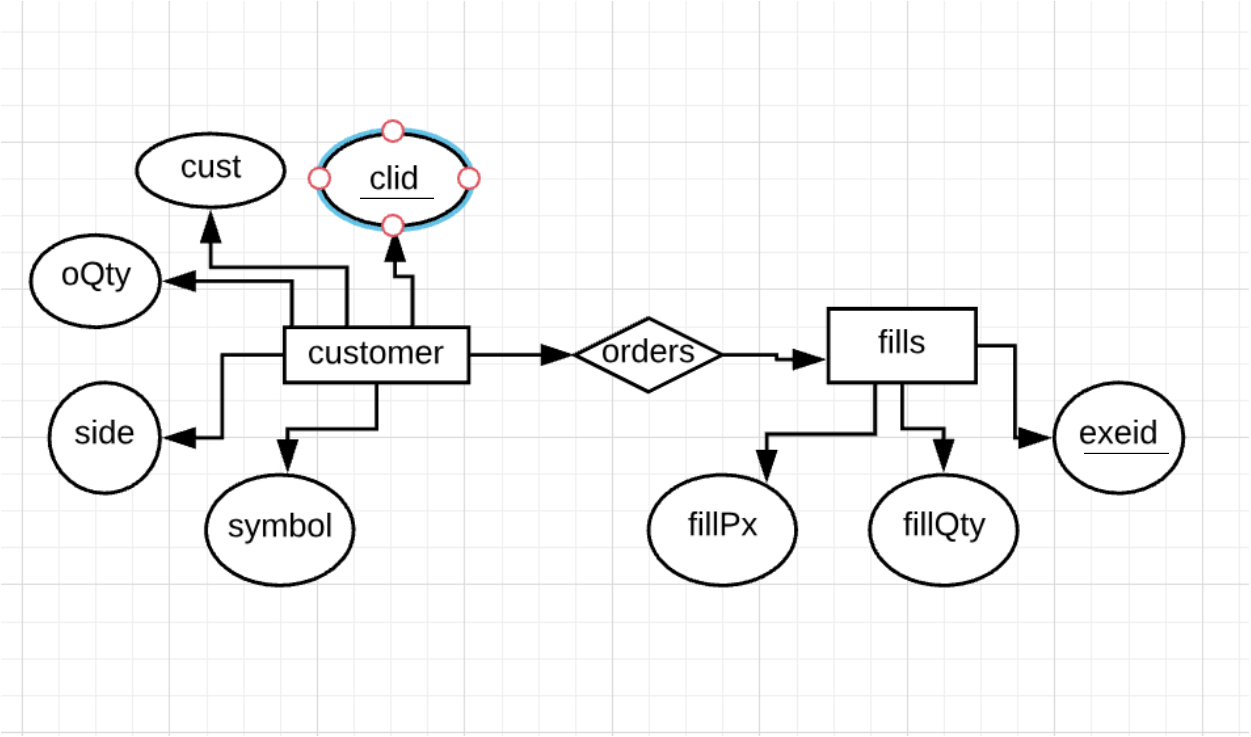
be in 1NF and all non-key attribute dependent on primary key. Every non-key depends on primary key.

Primary key : clid, cust, oQty, exeid, side, fillQty, fillPx, symbol
Candidate key: clid, exeid
Superkey: clid for customer and exeid for fills

Third Normal Form (3NF) - no transitive dependency.

There is no transitive dependency.

3) Draw an E-R diagram post-normalization.



SQL PART:

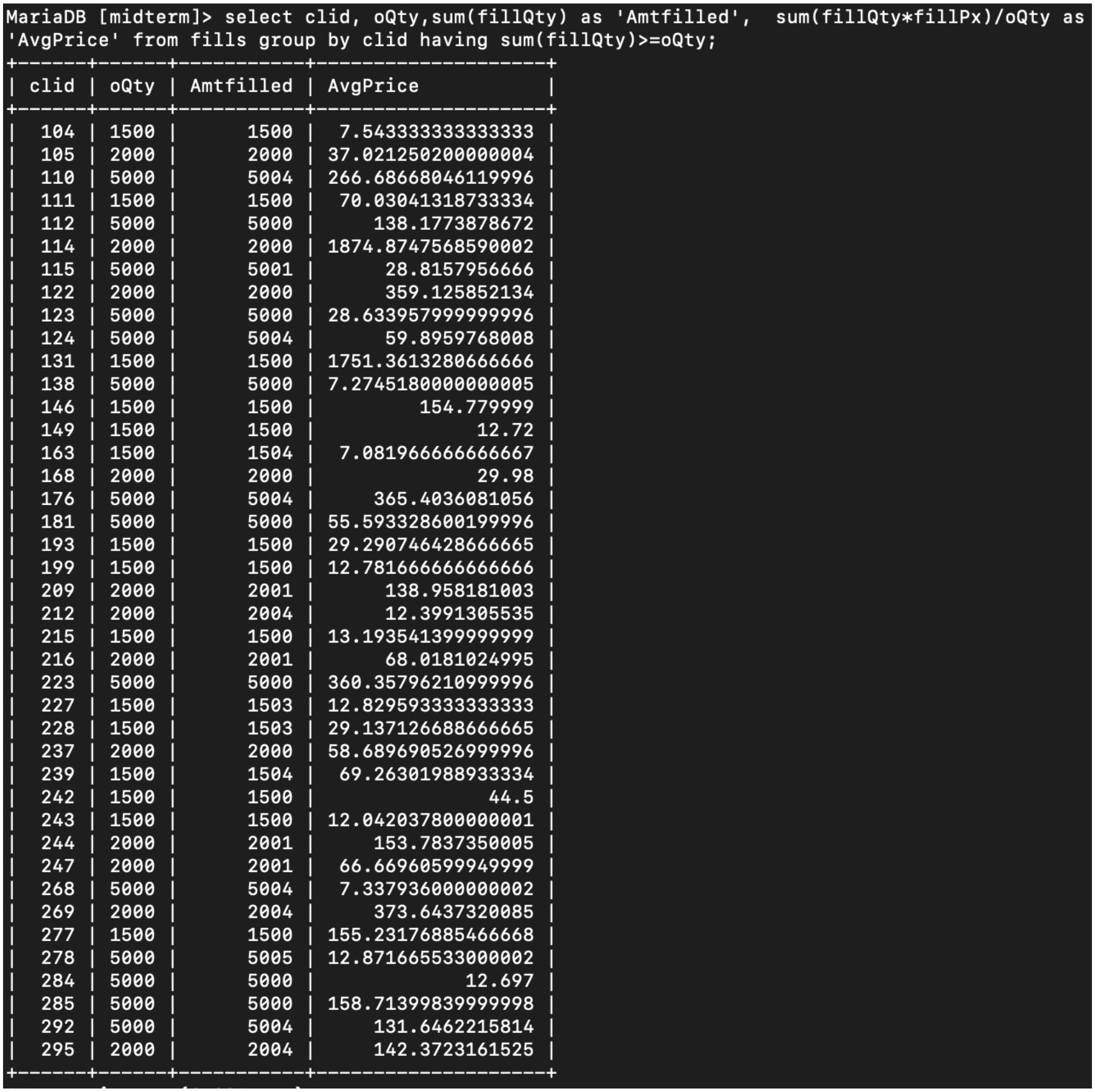
```
$ mysql -u root
> create database midterm;
> use midterm;
> create table fills( clid int primary key, cust varchar(3),symbol
varchar(5), side varchar(4),oQty int, fillQty int, fillPx double,
execid int);
> load data local infile '/Users/topgyaltsering/Desktop/fills16.csv'
into table fills fields terminated by ',';
```

4) **Generate a report for each customer order, if it has been filled and the average price at which it has been filled.**

if oQty=sum(partialFillQty) order is filled
and
AvgPrice= sum(fillQty* fillPx) / oQty

I am just selecting clid, oQty and amtfilled and avgprice created to print those.
sum(FillQty) is to compare with the oQty.

The specified condition having is used to compare if sum(fillQty>=oQty) if greater or equal than it is filled. The statement only prints the one with greater or equal. We used greater since some data tends to surpass the oQty.



```
MariaDB [midterm]> select clid, oQty,sum(fillQty) as 'Amtfilled',  sum(fillQty*fillPx)/oQty as
'AvgPrice' from fills group by clid having sum(fillQty)>=oQty;
+------+------+-----------+--------------------+
| clid | oQty | Amtfilled | AvgPrice           |
+------+------+-----------+--------------------+
|  104 | 1500 |      1500 |  7.543333333333333 |
|  105 | 2000 |      2000 | 37.021250200000004 |
|  110 | 5000 |      5004 | 266.68668046119996 |
|  111 | 1500 |      1500 |  70.03041318733334 |
|  112 | 5000 |      5000 |     138.1773878672 |
|  114 | 2000 |      2000 | 1874.8747568590002 |
|  115 | 5000 |      5001 |      28.8157956666 |
|  122 | 2000 |      2000 |      359.125852134 |
|  123 | 5000 |      5000 | 28.633957999999996 |
|  124 | 5000 |      5004 |      59.8959768008 |
|  131 | 1500 |      1500 | 1751.3613280666666 |
|  138 | 5000 |      5000 | 7.2745180000000005 |
|  146 | 1500 |      1500 |         154.779999 |
|  149 | 1500 |      1500 |              12.72 |
|  163 | 1500 |      1504 |  7.081966666666667 |
|  168 | 2000 |      2000 |              29.98 |
|  176 | 5000 |      5004 |     365.4036081056 |
|  181 | 5000 |      5000 | 55.593328600199996 |
|  193 | 1500 |      1500 | 29.290746428666665 |
|  199 | 1500 |      1500 | 12.781666666666666 |
|  209 | 2000 |      2001 |      138.958181003 |
|  212 | 2000 |      2004 |      12.3991305535 |
|  215 | 1500 |      1500 | 13.193541399999999 |
|  216 | 2000 |      2001 |      68.0181024995 |
|  223 | 5000 |      5000 | 360.35796210999996 |
|  227 | 1500 |      1503 | 12.829593333333333 |
|  228 | 1500 |      1503 | 29.137126688666665 |
|  237 | 2000 |      2000 | 58.689690526999996 |
|  239 | 1500 |      1504 |  69.26301988933334 |
|  242 | 1500 |      1500 |               44.5 |
|  243 | 1500 |      1500 | 12.042037800000001 |
|  244 | 2000 |      2001 |     153.7837350005 |
|  247 | 2000 |      2001 |  66.66960599949999 |
|  268 | 5000 |      5004 |  7.337936000000002 |
|  269 | 2000 |      2004 |     373.6437320085 |
|  277 | 1500 |      1500 | 155.23176885466668 |
|  278 | 5000 |      5005 | 12.871665533000002 |
|  284 | 5000 |      5000 |             12.697 |
|  285 | 5000 |      5000 | 158.71399839999998 |
|  292 | 5000 |      5004 |     131.6462215814 |
|  295 | 2000 |      2004 |     142.3723161525 |
+------+------+-----------+--------------------+
```

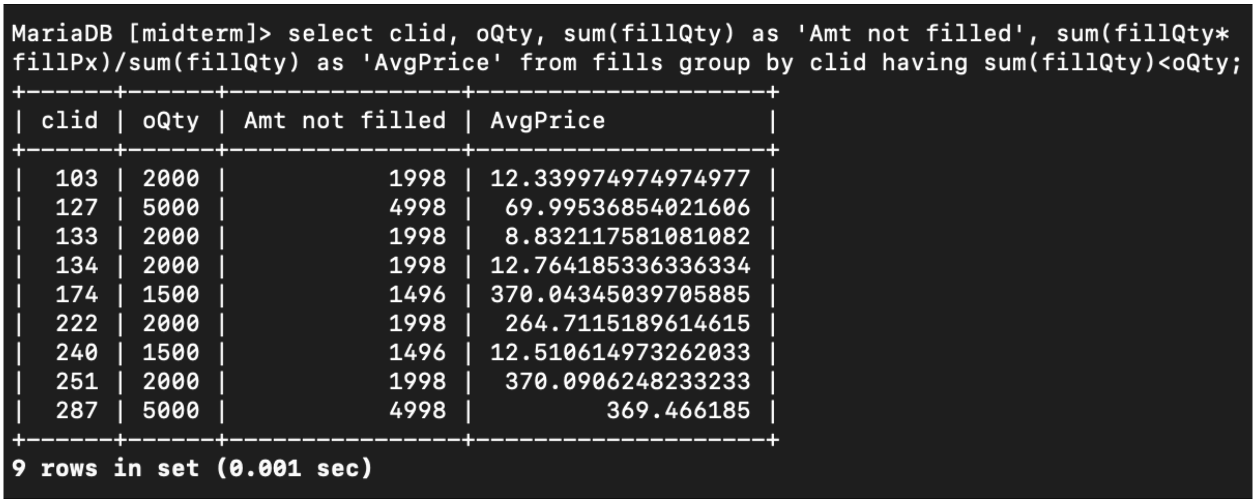
5) **Generate a report of all orders which have NOT been filled SUM(partial_fill_qty) < OrderQty.**

When OrderQty<SUM(partialFillQty) --- order is unFilled

the average price of the customer order = SUM (partial_fill_qty * partial_fill_px)/SUM(partial_fill_qty)

avgPrice is sum(fillQty*fillPx)/sum(fillQty)

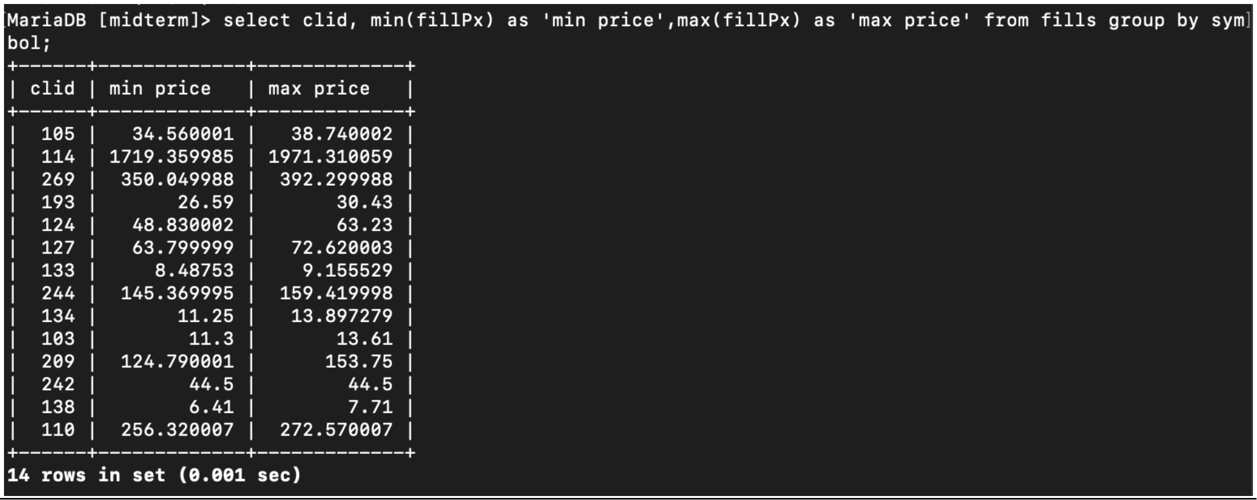
For unfilled we have condition having sum(fillQty)<oQty. Since such orders are unfilled.



```
MariaDB [midterm]> select clid, oQty, sum(fillQty) as 'Amt not filled', sum(fillQty*
fillPx)/sum(fillQty) as 'AvgPrice' from fills group by clid having sum(fillQty)<oQty;
+------+------+----------------+--------------------+
| clid | oQty | Amt not filled | AvgPrice           |
+------+------+----------------+--------------------+
|  103 | 2000 |           1998 | 12.339974974974977 |
|  127 | 5000 |           4998 |  69.99536854021606 |
|  133 | 2000 |           1998 |  8.832117581081082 |
|  134 | 2000 |           1998 | 12.764185336336334 |
|  174 | 1500 |           1496 | 370.04345039705885 |
|  222 | 2000 |           1998 |  264.7115189614615 |
|  240 | 1500 |           1496 | 12.510614973262033 |
|  251 | 2000 |           1998 |  370.0906248233233 |
|  287 | 5000 |           4998 |         369.466185 |
+------+------+----------------+--------------------+
9 rows in set (0.001 sec)
```

6) **For each order, find the min(fillPx) and max(fillPx) for the partialFills.**

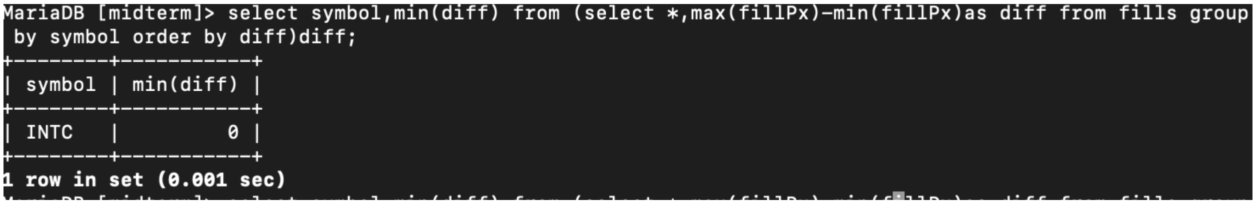
Min(fillPx) will print all min partialfills and max(fillPx) will print all max partial fills.



```
MariaDB [midterm]> select clid, min(fillPx) as 'min price',max(fillPx) as 'max price' from fills group by sym
bol;
+------+-------------+-------------+
| clid | min price   | max price   |
+------+-------------+-------------+
|  105 |   34.560001 |   38.740002 |
|  114 | 1719.359985 | 1971.310059 |
|  269 |  350.049988 |  392.299988 |
|  193 |       26.59 |       30.43 |
|  124 |   48.830002 |       63.23 |
|  127 |   63.799999 |   72.620003 |
|  133 |     8.48753 |    9.155529 |
|  244 |  145.369995 |  159.419998 |
|  134 |       11.25 |   13.897279 |
|  103 |        11.3 |       13.61 |
|  209 |  124.790001 |      153.75 |
|  242 |        44.5 |        44.5 |
|  138 |        6.41 |        7.71 |
|  110 |  256.320007 |  272.570007 |
+------+-------------+-------------+
14 rows in set (0.001 sec)
```

7**) Find the symbol for which the difference between min(fillPx) and max(fillPx) is smallest?**

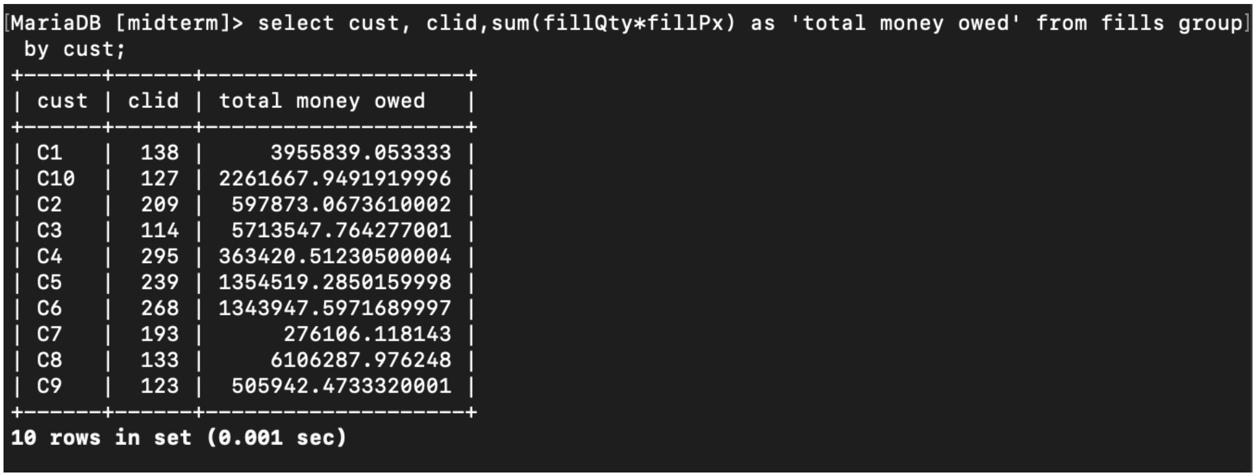
Using subquery, max(fillPx)-min(fillPx) will output the minimum and select and group by symbol since we need symbol.



```
MariaDB [midterm]> select symbol,min(diff) from (select *,max(fillPx)-min(fillPx)as diff from fills group
 by symbol order by diff)diff;
+--------+-----------+
| symbol | min(diff) |
+--------+-----------+
| INTC   |         0 |
+--------+-----------+
1 row in set (0.001 sec)
```

8) **For each customer, generate the total money owed SUM (partial_fill_qty * partial_fill_px)**

Sum(fillQty* fillPx) will give total money owed and we group by cust.



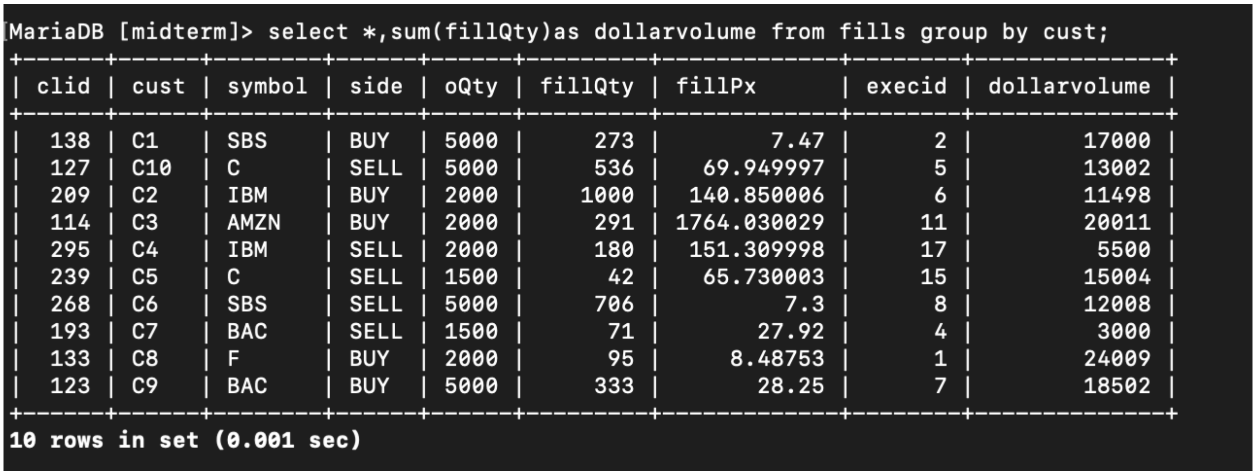
```
MariaDB [midterm]> select cust, clid,sum(fillQty*fillPx) as 'total money owed' from fills group
 by cust;
+------+------+--------------------+
| cust | clid | total money owed   |
+------+------+--------------------+
| C1   |  138 |     3955839.053333 |
| C10  |  127 | 2261667.9491919996 |
| C2   |  209 |  597873.0673610002 |
| C3   |  114 |  5713547.764277001 |
| C4   |  295 | 363420.51230500004 |
| C5   |  239 | 1354519.2850159998 |
| C6   |  268 | 1343947.5971689997 |
| C7   |  193 |      276106.118143 |
| C8   |  133 |     6106287.976248 |
| C9   |  123 |  505942.4733320001 |
+------+------+--------------------+
10 rows in set (0.001 sec)
```

9) **List the customers who transacted the most volume (bot or sold the most number of shares, sum(fillqty) for all orders submitted by each customer, and then find the customer with the maximum.**

Add all oQty for each customer and find the max.

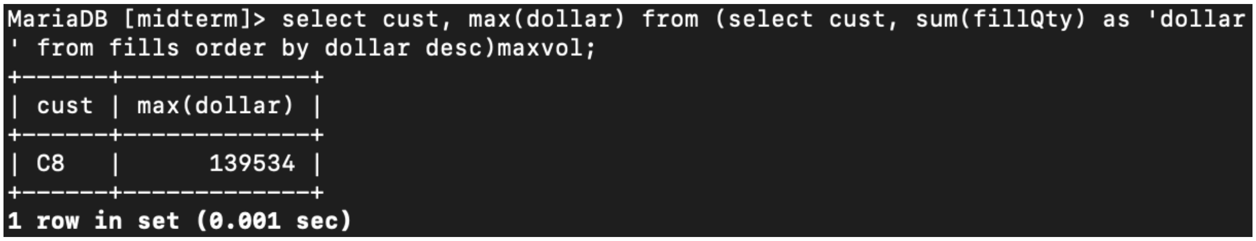
Using subquery:

First output the sum(fillQty) for all cust as dollarvolume.



```
[MariaDB [midterm]> select *,sum(fillQty)as dollarvolume from fills group by cust;
+------+------+--------+------+------+---------+-------------+--------+--------------+
| clid | cust | symbol | side | oQty | fillQty | fillPx      | execid | dollarvolume |
+------+------+--------+------+------+---------+-------------+--------+--------------+
|  138 | C1   | SBS    | BUY  | 5000 |     273 |        7.47 |      2 |        17000 |
|  127 | C10  | C      | SELL | 5000 |     536 |   69.949997 |      5 |        13002 |
|  209 | C2   | IBM    | BUY  | 2000 |    1000 |  140.850006 |      6 |        11498 |
|  114 | C3   | AMZN   | BUY  | 2000 |     291 | 1764.030029 |     11 |        20011 |
|  295 | C4   | IBM    | SELL | 2000 |     180 |  151.309998 |     17 |         5500 |
|  239 | C5   | C      | SELL | 1500 |      42 |   65.730003 |     15 |        15004 |
|  268 | C6   | SBS    | SELL | 5000 |     706 |         7.3 |      8 |        12008 |
|  193 | C7   | BAC    | SELL | 1500 |      71 |       27.92 |      4 |         3000 |
|  133 | C8   | F      | BUY  | 2000 |      95 |     8.48753 |      1 |        24009 |
|  123 | C9   | BAC    | BUY  | 5000 |     333 |       28.25 |      7 |        18502 |
+------+------+--------+------+------+---------+-------------+--------+--------------+
10 rows in set (0.001 sec)
```

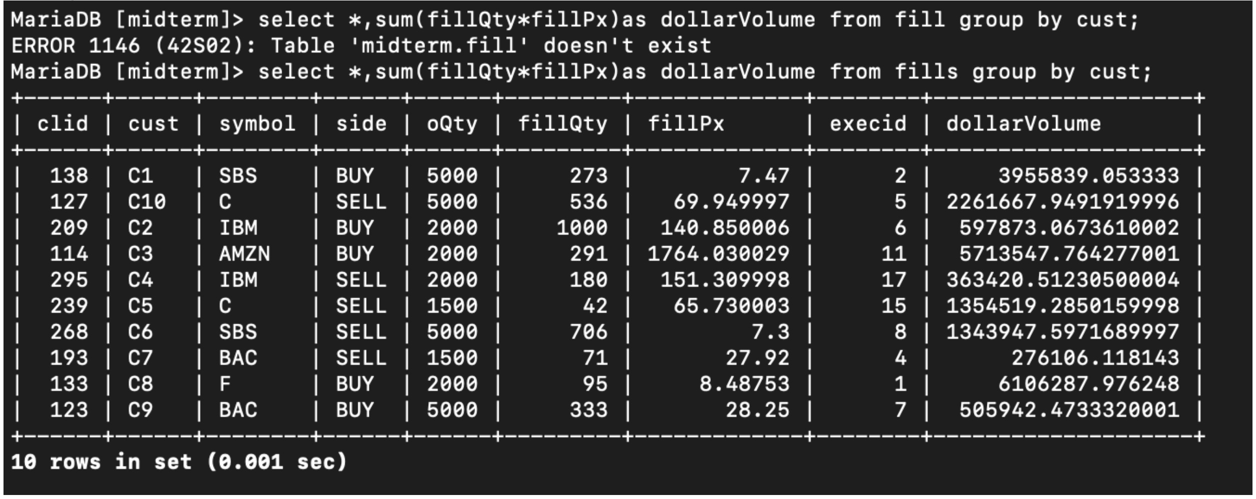
Then we find the customer with the maximum:



```
MariaDB [midterm]> select cust, max(dollar) from (select cust, sum(fillQty) as 'dollar
' from fills order by dollar desc)maxvol;
+------+-------------+
| cust | max(dollar) |
+------+-------------+
| C8   |      139534 |
+------+-------------+
1 row in set (0.001 sec)
```

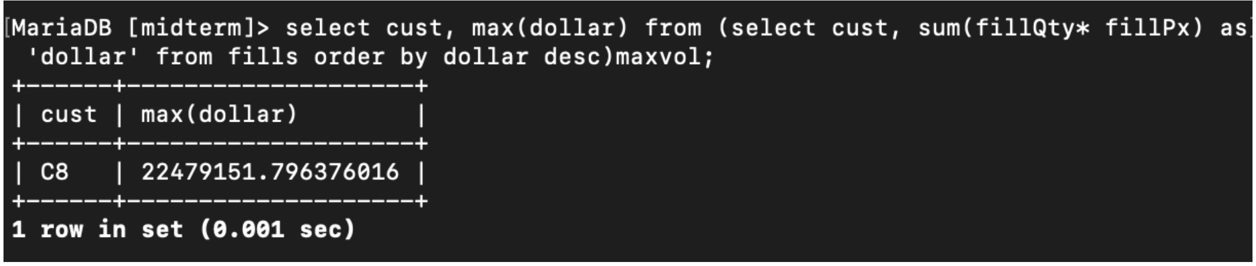
10) **List the customers who transacted the most in dollar amount (bot or sold,sum (fillqty* fillpx) and find the customer(s) with the highest)**

First we output the customer who transacted the most in dollar amount sum(fillQty* fillPx)



```
MariaDB [midterm]> select *,sum(fillQty*fillPx)as dollarVolume from fill group by cust;
ERROR 1146 (42S02): Table 'midterm.fill' doesn't exist
MariaDB [midterm]> select *,sum(fillQty*fillPx)as dollarVolume from fills group by cust;
+------+------+--------+------+------+---------+-------------+--------+--------------------+
| clid | cust | symbol | side | oQty | fillQty | fillPx      | execid | dollarVolume       |
+------+------+--------+------+------+---------+-------------+--------+--------------------+
|  138 | C1   | SBS    | BUY  | 5000 |     273 |        7.47 |      2 |     3955839.053333 |
|  127 | C10  | C      | SELL | 5000 |     536 |   69.949997 |      5 | 2261667.9491919996 |
|  209 | C2   | IBM    | BUY  | 2000 |    1000 |  140.850006 |      6 |  597873.0673610002 |
|  114 | C3   | AMZN   | BUY  | 2000 |     291 | 1764.030029 |     11 |  5713547.764277001 |
|  295 | C4   | IBM    | SELL | 2000 |     180 |  151.309998 |     17 | 363420.51230500004 |
|  239 | C5   | C      | SELL | 1500 |      42 |   65.730003 |     15 | 1354519.2850159998 |
|  268 | C6   | SBS    | SELL | 5000 |     706 |         7.3 |      8 | 1343947.5971689997 |
|  193 | C7   | BAC    | SELL | 1500 |      71 |       27.92 |      4 |      276106.118143 |
|  133 | C8   | F      | BUY  | 2000 |      95 |     8.48753 |      1 |     6106287.976248 |
|  123 | C9   | BAC    | BUY  | 5000 |     333 |       28.25 |      7 |  505942.4733320001 |
+------+------+--------+------+------+---------+-------------+--------+--------------------+
10 rows in set (0.001 sec)
```

Then we output the customer with the highest by using subquery max(dollar) where dollar is sum(fillQty*fillPx)



```
[MariaDB [midterm]> select cust, max(dollar) from (select cust, sum(fillQty* fillPx) as]
 'dollar' from fills order by dollar desc)maxvol;
+------+-------------------+
| cust | max(dollar)       |
+------+-------------------+
| C8   | 22479151.796376016 |
+------+-------------------+
1 row in set (0.001 sec)
```
