

AD-HOC ANALYSIS



FOR ATLIQ HARDWARE

AD-HOC REQUEST 1 –

CROMA INDIA PRODUCT SALES SUMMARY (FY 2021)

As a **Product Owner**, I need a **monthly summary report** showing **sales figures for each product (based on product code)** for the **Croma India** customer during **FY 2021**.

SQL Query:

```
select
    s.date,s.product_code,p.product,
    p.variant,s.sold_quantity, gross_price ,round(g.gross_price*sold_quantity,2) as gross_price_total
from fact_sales_monthly s
join dim_product p
on
    p.product_code=s.product_code
join fact_gross_price g
on
    g.product_code = s.product_code and
    s.fiscal_year = g.fiscal_year
where
    customer_code=90002002 and
    s.fiscal_year = 2021
    order by date asc;
```

Result:

date	product_code	product	variant	sold_quantity	gross_price	gross_price_total
2021-01-01	A0118150102	AQ Dracula HDD – 3.5 I...	Plus	121	21.4565	2596.24
2021-01-01	A0118150103	AQ Dracula HDD – 3.5 I...	Premium	142	21.7795	3092.69
2021-01-01	A0118150104	AQ Dracula HDD – 3.5 I...	Premium Plus	35	22.9729	804.05
2021-01-01	A0219150201	AQ WereWolf NAS Intern...	Standard	161	23.6987	3815.49
2021-01-01	A0219150202	AQ WereWolf NAS Intern...	Plus	241	24.7312	5960.22
2021-01-01	A0220150203	AQ WereWolf NAS Intern...	Premium	41	23.6154	968.23
2021-01-01	A0320150301	AQ Zion Saga	Standard	121	23.7223	2870.40
2021-01-01	A0321150302	AQ Zion Saga	Plus	164	27.1027	4444.84
2021-01-01	A0321150303	AQ Zion Saga	Premium	172	28.0059	4817.01
2021-01-01	A0418150103	AQ Mforce Gen X	Standard 3	133	19.5235	2596.63
2021-01-01	A0418150104	AQ Mforce Gen X	Plus 1	155	19.9239	3088.20
2021-01-01	A0418150105	AQ Mforce Gen X	Plus 2	176	20.0766	3533.48
2021-01-01	A0418150106	AQ Mforce Gen X	Plus 3	121	19.9365	2412.32
2021-01-01	A0519150201	AQ Mforce Gen Y	Standard 1	81	22.3984	1814.27
2021-01-01	A0519150202	AQ Mforce Gen Y	Standard 2	65	24.9298	1620.44
2021-01-01	A0519150203	AQ Mforce Gen Y	Standard 3	144	26.5871	3828.54
2021-01-01	A0519150204	AQ Mforce Gen Y	Plus 1	114	26.1081	2976.32
2021-01-01	A0519150205	AQ Mforce Gen Y	Plus 2	166	29.7008	4930.33
2021-01-01	A0519150206	AQ Mforce Gen Y	Plus 3	120	31.2439	3749.27
2021-01-01	A0519150207	AQ Mforce Gen Y	Premium 1	20	32.4427	648.85
2021-01-01	A0519150208	AQ Mforce Gen Y	Premium 2	72	30.5816	2201.88
2021-01-01	A0619150301	AQ Mforce Gen Z	Standard 1	123	30.4696	3747.76

AD-HOC REQUEST 2 –

GROSS MONTHLY TOTAL SALES REPORT FOR CROMA

As a **Product Owner**, I need an **aggregate monthly gross sales report** for **Croma India** so that I can track how much sales this particular customer is generating for **AtliQ Hardwares** and manage our relationship accordingly.

SQL Query:

```
select date_format(s.date, '%b %y') as month,
round(sum(g.gross_price*s.sold_quantity), 2) as gross_total
from fact_sales_monthly s
join fact_gross_price g
on
g.product_code = s.product_code and
g.fiscal_year = s.fiscal_year
where customer_code = 90002002
group by s.date
order by s.date asc
```

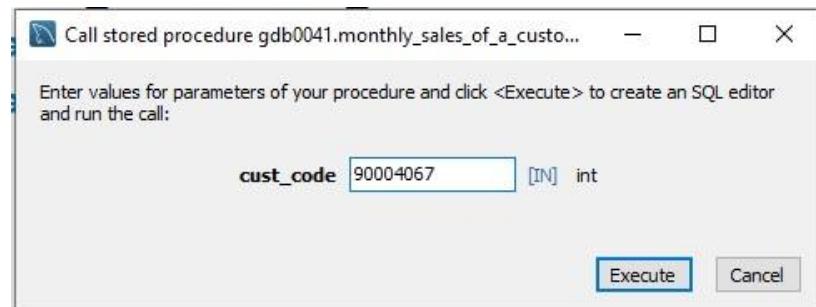
Result:

	month	gross_price_total
▶	Sep 17	122407.56
	Oct 17	162687.57
	Dec 17	245673.80
	Jan 18	127574.74
	Feb 18	144799.52
	Apr 18	130643.90
	May 18	139165.10
	Jun 18	125735.38
	Aug 18	125409.88
	Sep 18	343337.17
	Oct 18	440562.08
	Dec 18	653944.75
	Jan 19	359025.02
	Feb 19	356607.17
	Apr 19	379549.69
	May 19	340152.23
	Jun 19	343792.04
	Aug 19	338108.88
	Sep 19	808250.44
	Oct 19	1092622.20
	Dec 19	1488174.02
	Jan 20	812929.75

Stored Procedure Query:

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `monthly_sales_of_a_customer_for_a_fiscal_year`(cust_code int)
BEGIN
    select date_format(s.date,"%b %y") as month, round(sum(g.gross_price * s.sold_quantity),2) as gross_sales
    from fact_sales_monthly s
    join fact_gross_price g
    on s.product_code = g.product_code and s.fiscal_year = g.fiscal_year
    where s.customer_code = cust_code
    group by s.date
    order by s.date asc;
END
```

Result:



	month	gross_sales
▶	Sep 19	83112.57
	Nov 19	159651.84
	Dec 19	159487.86
	Jan 20	88891.61
	Mar 20	12957.59
	Apr 20	91281.06
	May 20	139147.16
	Jul 20	168967.54
	Aug 20	81040.17
	Sep 20	243117.55
	Nov 20	437960.20
	Dec 20	445361.45
	Jan 21	282077.10
	Mar 21	241123.47
	Apr 21	248101.43
	May 21	263468.63
	Jul 21	257409.74
	Aug 21	251201.08
	Sep 21	1245380.78
	Nov 21	2126850.42
	Dec 21	2183085.96

AD-HOC REQUEST 3 – MARKET BADGE STORED PROCEDURE

As a **Product Owner**, I want to create a **stored procedure** that determines the **market badge** based on the **total sold quantity**.

If the total sold quantity exceeds **5 million units**, the market is considered **Gold**; otherwise, it is **Silver**.

SQL Query:

```
CREATE DEFINER='root'@'localhost' PROCEDURE `market_badge`(in country varchar(45),in fy year,out badge varchar(45))
BEGIN
    declare qty int default 0;

    if country = '' then set country = 'india';
    end if;

    select sum(sold_quantity) into qty
    from fact_sales_monthly
    join dim_customer using (customer_code)
    where market = country and fiscal_year = fy
    group by fiscal_year;

    if qty>5000000 then set badge = 'Gold';
    else set badge = 'Silver';
    end if;
END
```

Result:



	market	total_qty_sold	badge
▶	India	13751429	Gold

AD-HOC REQUEST 4 –

TOP MARKETS, PRODUCTS, AND CUSTOMERS BY NET SALES

As a **Product Owner**, I want a **report** showing the **top markets, products, and customers by net sales** for a given **financial year**.

SQL Query to create Net Sales table:

```

with gs as
  (select s.date, s.fiscal_year, s.customer_code, s.product_code, s.sold_quantity,
    g.gross_price as gross_price_per_item, g.gross_price * s.sold_quantity as gross_total, pre.pre_invoice_discount_pct
  from fact_sales_monthly s
  join fact_gross_price g
    on s.product_code = g.product_code and s.fiscal_year = g.fiscal_year
  join fact_pre_invoice_deductions pre
    on s.customer_code = pre.customer_code and s.fiscal_year = pre.fiscal_year
  ),
  nis as
  (select s.* , ((1 - s.pre_invoice_discount_pct) * gross_total) as net_invoice_sales,
    (po.discounts_pct + po.other_deductions_pct) as post_invoice_deductions_pct
  from gs s
  join fact_post_invoice_deductions po
    on s.date = po.date and
      s.product_code = po.product_code and
      s.customer_code = po.customer_code
  )
  select s.* , ((1 - s.post_invoice_deductions_pct) * s.net_invoice_sales) as net_sales
  FROM nis s
  
```

Result:

The screenshot shows a database management tool window. On the left, there's a 'Navigator' pane with a 'SCHEMAS' section containing tables like dim_date, dim_product, fact_act_est, fact_sales_monthly, fact_fairight_cost, fact_gross_sales, fact_manufacturing_cost, fact_post_invoice_deductions, fact_pre_invoice_deductions, fact_sales_monthly, gross_sales_table, net_invoice_sales_table, and net_sales_table. Below this is a 'Views' section with gross_sales_table, net_invoice_sales_table, and net_sales_table. Under 'Stored Procedures', there's a single entry: market_hanline. At the bottom of the sidebar, it says 'No object selected'. The main workspace shows the query '1. SELECT * FROM gdb0041.net_sales_table;' at the top, followed by a result grid. The grid has columns: date, fiscal_year, customer_code, customer, market, product_code, product, variant, sold_quantity, gross_price_per_item, and gross_total. The data consists of multiple rows of sales records for different dates, customers, and products across Brazil and Canada.

date	fiscal_year	customer_code	customer	market	product_code	product	variant	sold_quantity	gross_price_per_item	gross_total
2017-09-01	2018	90027207	Amazon	Brazil	A0118150101	AQ Dracula HDD - 3.5 I...	Standard	4	15.3952	61.5808
2017-11-01	2018	90027207	Amazon	Brazil	A0118150101	AQ Dracula HDD - 3.5 I...	Standard	16	15.3952	246.3232
2017-12-01	2018	90027207	Amazon	Brazil	A0118150101	AQ Dracula HDD - 3.5 I...	Standard	4	15.3952	61.5808
2018-01-01	2018	90027207	Amazon	Brazil	A0118150101	AQ Dracula HDD - 3.5 I...	Standard	6	15.3952	92.3712
2018-03-01	2018	90027207	Amazon	Brazil	A0118150101	AQ Dracula HDD - 3.5 I...	Standard	9	15.3952	138.5568
2018-04-01	2018	90027207	Amazon	Brazil	A0118150101	AQ Dracula HDD - 3.5 I...	Standard	6	15.3952	92.3712
2018-05-01	2018	90027207	Amazon	Brazil	A0118150101	AQ Dracula HDD - 3.5 I...	Standard	7	15.3952	107.7664
2018-07-01	2018	90027207	Amazon	Brazil	A0118150101	AQ Dracula HDD - 3.5 I...	Standard	10	15.3952	153.9520
2018-08-01	2018	90027207	Amazon	Brazil	A0118150101	AQ Dracula HDD - 3.5 I...	Standard	6	15.3952	92.3712
2017-09-01	2018	90023030	Amazon	Canada	A0118150101	AQ Dracula HDD - 3.5 I...	Standard	4	15.3952	61.5808
2017-10-01	2018	90023030	Amazon	Canada	A0118150101	AQ Dracula HDD - 3.5 I...	Standard	2	15.3952	30.7904
2017-12-01	2018	90023030	Amazon	Canada	A0118150101	AQ Dracula HDD - 3.5 I...	Standard	3	15.3952	46.1856
2018-01-01	2018	90023030	Amazon	Canada	A0118150101	AQ Dracula HDD - 3.5 I...	Standard	5	15.3952	76.9760
2018-02-01	2018	90023030	Amazon	Canada	A0118150101	AQ Dracula HDD - 3.5 I...	Standard	1	15.3952	15.3952
2018-04-01	2018	90023030	Amazon	Canada	A0118150101	AQ Dracula HDD - 3.5 I...	Standard	1	15.3952	15.3952
2018-05-01	2018	90023030	Amazon	Canada	A0118150101	AQ Dracula HDD - 3.5 I...	Standard	5	15.3952	76.9760
2018-06-01	2018	90023030	Amazon	Canada	A0118150101	AQ Dracula HDD - 3.5 I...	Standard	1	15.3952	15.3952
2018-08-01	2018	90023030	Amazon	Canada	A0118150101	AQ Dracula HDD - 3.5 I...	Standard	1	15.3952	15.3952
2017-09-01	2018	90023029	Staples	Canada	A0118150101	AQ Dracula HDD - 3.5 I...	Standard	2	15.3952	30.7904
2017-10-01	2018	90023029	Staples	Canada	A0118150101	AQ Dracula HDD - 3.5 I...	Standard	4	15.3952	61.5808
2017-11-01	2018	90023029	Staples	Canada	A0118150101	AQ Dracula HDD - 3.5 I...	Standard	3	15.3952	46.1856

Stored Procedure Query for Top n Market:

```
CREATE DEFINER='root'@'localhost' PROCEDURE `top_n_market`(fy int, top_n int)
BEGIN
SELECT market, round(SUM(net_sales)/1000000,2) as total_net_sales
FROM net_sales_table
WHERE fiscal_year = fy
GROUP BY market
order by total_net_sales desc limit top_n;
END
```

Result:

The screenshot shows the MySQL Workbench interface with a call dialog for the stored procedure. The parameters are set to `in_fiscal_year`: 2021 and `in_top_n`: 3. The result table displays the top three markets by net sales in millions for the fiscal year 2021.

market	net_sales_mln
India	210.67
USA	132.05
South Korea	64.01

Stored Procedure Query for Top n Product:

```
CREATE DEFINER='root'@'localhost' PROCEDURE `top_n_product`(fy int, country text, top_n int)
BEGIN
SELECT product, round(SUM(net_sales)/1000000,2) as total_net_sales
FROM net_sales_table
WHERE fiscal_year = fy and market = country
GROUP BY product
order by total_net_sales desc limit top_n;
END
```

Result:

The screenshot shows the MySQL Workbench interface with a call dialog for the stored procedure. The parameters are set to `in_fiscal_year`: 2021 and `in_top_n`: 5. The result table displays the top five products by net sales in millions for the fiscal year 2021.

product	net_sales_mln
AQ BZ Allin1	33.75
AQ Qwerty	27.84
AQ Trigger	26.95
AQ Gen Y	23.58
AQ Maxima	22.32

Stored Procedure Query for Top n Customer:

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `top_n_customers`(fy int, top_n int)
BEGIN
SELECT customer, round(SUM(net_sales)/1000000,2) as total_net_sales
FROM net_sales_table
WHERE fiscal_year = fy
GROUP BY customer
order by total_net_sales desc limit top_n;
END
```

Result:

Call stored procedure gdb0041.get_top_n_customers_by_...

Enter values for parameters of your procedure and click <Execute> to create an SQL editor and run the call:

in_market	india	[IN] varchar(45)
in_fiscal_year	2021	[IN] int
in_top_n	5	[IN] int

Execute Cancel

	customer	net_sales_mln
▶	Amazon	30.00
	Atliq Exclusive	23.98
	Flipkart	12.96
	Electricalsociety	12.31
	Propel	11.86

AD-HOC REQUEST 5 –

NET SALES % SHARE (GLOBAL)

As a Product Owner, I want to see a **bar chart report** for FY 2021 showing the **top 10 markets by % Net Sales**.

SQL Query:

```
> with cte1 as (
  SELECT c.customer, round(SUM(s.net_sales)/1000000,2) as total_net_sales
  FROM ns s
  join dim_customer c
  on c.customer_code = s.customer_code
  WHERE s.fiscal_year = 2021
  GROUP BY c.customer
)
select *, round(total_net_sales * 100 / sum(total_net_sales) over(),2) as pct
from cte1
order by total_net_sales desc
```

Result:

customer	total_net_sales	pct
Amazon	109.03	13.23
Atliq Exclusive	79.92	9.70
Atliq e Store	70.31	8.53
Sage	27.07	3.29
Flipkart	25.25	3.06
Leader	24.52	2.98
Neptune	21.01	2.55
Ebay	19.88	2.41
Electricalsociety	16.25	1.97
Synthetic	16.10	1.95
Electricalslyti...	15.64	1.90
Acclaimed St...	14.32	1.74
Propel	14.14	1.72
Novus	12.91	1.57
Expression	12.90	1.57
Reliance Digital	12.75	1.55
walmart	12.63	1.53
Costco	12.19	1.48
Staples	11.49	1.39
Girias	11.30	1.37
Vijay Sales	11.27	1.37
Path	11.02	1.34
Lotus	10.53	1.28



AD-HOC REQUEST 6 –

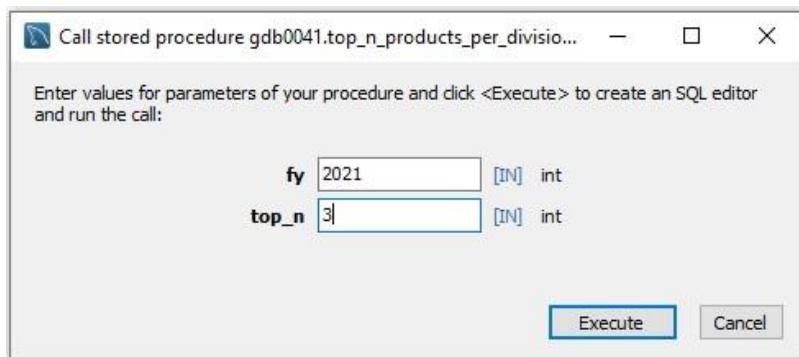
TOP N PRODUCTS BY QUANTITY SOLD (PER DIVISION)

As a **Product Owner**, I want to get the **Top N products** in each **division** based on their **quantity sold** for a given **financial year**.

SQL Query:

```
CREATE DEFINER='root'@'localhost' PROCEDURE `top_n_products_per_division_by_sold_qnt_using_drnk`  
(fy int, top_n int)  
BEGIN  
    with cte1 as (  
        SELECT p.division, p.product, sum(s.sold_quantity) as sold_quantity  
        FROM fact_sales_monthly s  
        join dim_product p  
        on p.product_code = s.product_code  
        where fiscal_year = fy  
        GROUP BY p.division, p.product  
        order by sold_quantity desc  
    ),  
    cte2 as (select *,  
        dense_rank() over( partition by division order by sold_quantity desc) as drnk  
    from cte1)  
    select * from cte2  
    where drnk <= top_n;  
END
```

Result:



	division	product	sold_quantity	drnk
▶	N & S	AQ Pen Drive DRC	2034569	1
	N & S	AQ Digit SSD	1240149	2
	N & S	AQ Clx1	1238683	3
	P & A	AQ Gamers Ms	2477098	1
	P & A	AQ Maxima Ms	2461991	2
	P & A	AQ Master wireless x1 Ms	2448784	3
	PC	AQ Digit	135092	1
	PC	AQ Gen Y	135031	2
	PC	AQ Elite	134431	3

AD-HOC REQUEST 7 –

FORECAST ACCURACY FOR ALL CUSTOMERS

As a **Product Owner**, I need an **aggregate forecast accuracy report** for all **customers** for a given **fiscal year**, so that I can track how accurate our forecasts are compared to actual sales.

SQL Query to create fact_act_est table:

```
SELECT s.*, f.forecast_quantity
  FROM fact_sales_monthly s
LEFT JOIN fact_forecast_monthly f
    USING (date, cust_id, prod_id)
```

```
UNION ALL
```

```
SELECT s.*, f.forecast_quantity
  FROM fact_sales_monthly s
RIGHT JOIN fact_forecast_monthly f
    USING (date, cust_id, prod_id)
 WHERE s.date IS NULL
```

Result:

	date	fiscal_year	customer_code	product_code	sold_quantity	forecast_quantity
▶	2017-09-01	2018	70002017	A0118150101	51	18
	2017-09-01	2018	70002018	A0118150101	77	11
	2017-09-01	2018	70003181	A0118150101	17	9
	2017-09-01	2018	70003182	A0118150101	6	6
	2017-09-01	2018	70006157	A0118150101	5	5
	2017-09-01	2018	70006158	A0118150101	7	6
	2017-09-01	2018	70007198	A0118150101	29	4
	2017-09-01	2018	70007199	A0118150101	34	7
	2017-09-01	2018	70008169	A0118150101	22	7
	2017-09-01	2018	70008170	A0118150101	5	8
	2017-09-01	2018	70011193	A0118150101	10	5
	2017-09-01	2018	70011194	A0118150101	4	7
	2017-09-01	2018	70013125	A0118150101	1	2
	2017-09-01	2018	70013126	A0118150101	1	2
	2017-09-01	2018	70022085	A0118150101	20	12
	2017-09-01	2018	70023031	A0118150101	4	1
	2017-09-01	2018	80001019	A0118150101	10	7
	2017-09-01	2018	80006154	A0118150101	10	21
	2017-09-01	2018	80006155	A0118150101	28	21
	2017-09-01	2018	80007195	A0118150101	80	62
	2017-09-01	2018	90001021	A0118150101	1	2
	2017-09-01	2018	90002001	A0118150101	42	52
	2017-09-01	2018	90002002	A0118150101	40	47

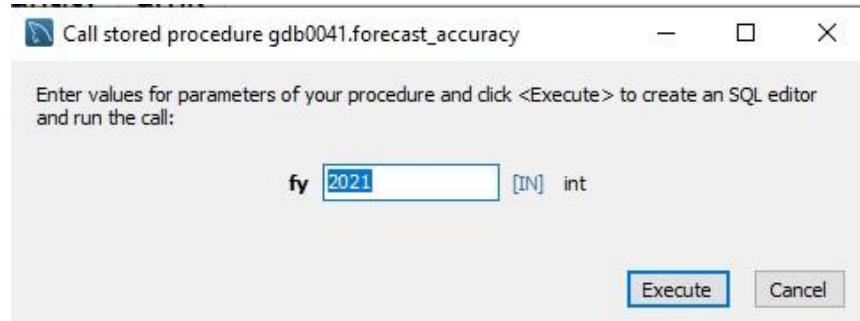
Stored Procedure Query:

```

CREATE DEFINER='root'@'localhost' PROCEDURE `forecast_accuracy`(fy int)
BEGIN
  with cte1 as (
    select *, forecast_quantity - sold_quantity as net_error, abs(forecast_quantity - sold_quantity) as abs_net_error
    from fact_act_est
  )
  select s.customer_code, c.customer, c.market, sum(s.sold_quantity) as total_sold_qnt,
    sum(s.forecast_quantity) as total_forecast_qnt, sum(net_error) as net_error,
    round((sum(net_error) * 100) / sum(forecast_quantity),2) as net_error_per, sum(abs_net_error) as abs_net_error,
    round((sum(abs_net_error)* 100) / sum(forecast_quantity),2) as abs_net_error_per,
    if ((sum(abs_net_error)* 100) / sum(forecast_quantity) > 100, 0,
        round(100 - (sum(abs_net_error)* 100) / sum(forecast_quantity),2)) as forecast_accuracy
  from cte1 s
  join dim_customer c
  on s.customer_code = c.customer_code
  where fiscal_year = fy
  group by s.customer_code
  order by forecast_accuracy ;
END

```

Result:



	customer_code	customer	market	total_sold_qnt	total_forecast_qnt	net_error	net_error_per	abs_net_error	abs_net_error_per	forecast_ac
>	90014138	Mbit	Netherlands	79249	43755	-35284	-80.64	45612	104.24	0
	90014135	Electricals lance Stores	Netherlands	82731	46698	-35871	-76.81	48267	103.36	0
	90014137	Media Markt	Netherlands	77931	44783	-32922	-73.51	45522	101.65	0
	70014143	Atliq e Store	Netherlands	81942	45314	-36432	-80.40	48348	106.70	0
	90014141	Amazon	Netherlands	77462	43887	-33412	-76.13	44852	102.20	0
	70014142	Atliq Exclusive	Netherlands	81324	44704	-36439	-81.51	47723	106.75	0
	90014140	Radio Popular	Netherlands	85581	47564	-37842	-79.56	49586	104.25	0
	90014139	Elkjøp	Netherlands	83198	46220	-36790	-79.60	48242	104.37	0
	90014136	Reliance Digital	Netherlands	84808	48415	-36201	-74.77	48253	99.67	0.33
	90007197	Amazon	South Korea	344240	226857	-117368	-51.74	191032	84.21	15.79
	70007198	Atliq Exclusive	South Korea	345667	228102	-117542	-51.53	188560	82.66	17.34
	70007199	Atliq e Store	South Korea	358064	236637	-121411	-51.31	194937	82.38	17.62
	80007196	Leader	South Korea	1456255	986574	-469681	-47.61	807109	81.81	18.19
	90019202	Argos (Sainsbury's)	Sweden	26581	18075	-8158	-45.13	14782	81.78	18.22
	90019203	Amazon	Sweden	27550	18912	-8313	-43.96	15455	81.72	18.28
	90012036	Billa	Germany	118242	80367	-37793	-47.03	65585	81.61	18.39
	90012034	Otto	Germany	112107	76219	-35821	-47.00	62141	81.53	18.47
	90012035	Notebillig	Germany	116882	80119	-36690	-45.79	64924	81.03	18.97
	90012037	Saturn	Germany	110414	76997	-33344	-43.31	62164	80.74	19.26
	80007195	Sage	South Korea	1443568	976823	-466745	-47.78	786971	80.56	19.44
	70012043	Atliq e Store	Germany	114116	80261	-33798	-42.11	64318	80.14	19.86
	90012040	Fnac-Darty	Germany	116151	84035	-32044	-38.13	65598	78.06	21.94
	90012039	Currys (Dixons Carph...	Germany	117037	81844	-35134	-42.93	63616	77.73	22.27