



BUSINESS REPORT

ANALYTICAL SQL
CASE STUDY

ITI

By: Ibrahim Elsayed Elsadek

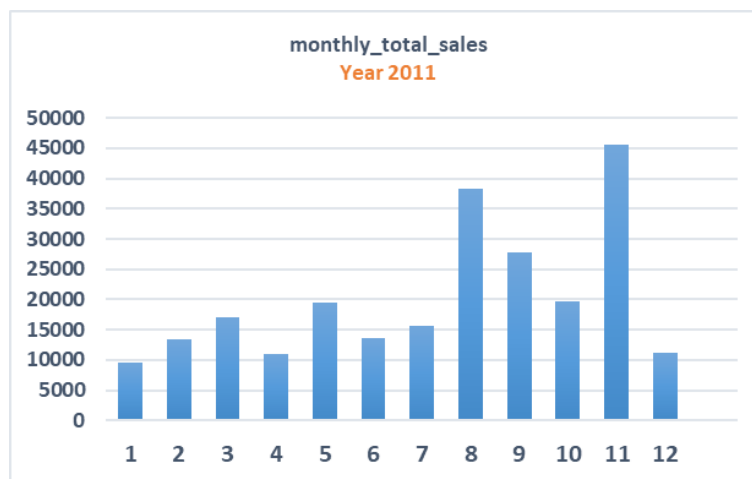
Query (1-1): Sales Report

First of all we need to see our annually sales report to identify what happened in the previous year, so as got from result we notice: *(12-2010 was excluded from result set in chart)*

- The highest monthly sales happened in November and the lowest was in April

year	month	monthly_total_sales	Quarterly_total_sales	yearly_total_sales
2010	12	13422.96	13422.96	13422.96
2011	1	9541.29	39916.14	242295.42
2011	2	13336.84	39916.14	242295.42
2011	3	17038.01	39916.14	242295.42
2011	4	10980.51	43993.7	242295.42
2011	5	19496.18	43993.7	242295.42
2011	6	13517.01	43993.7	242295.42
2011	7	15664.54	81893	242295.42
2011	8	38374.64	81893	242295.42
2011	9	27853.82	81893	242295.42
2011	10	19735.07	76492.58	242295.42
2011	11	45633.38	76492.58	242295.42
2011	12	11124.13	76492.58	242295.42

Monthly Sales (query result set)



Query (1-2): Most Purchased product in invoices

Now let's get some statistics about retail sales:

- **How many times a product included in an invoice?**
So, most requested product in our retail was (84879) with 59 invoice included, also we could see top 10 products requested in orders

StockCode	num_of_invoices
84879	59
22086	56
85099B	53
22197	49
22457	49
85123A	48
47566	47
23298	47
20725	46
21034	43

Query Result Set

Query (1-3): Discounts Offering decision

If we defined loyalty based on the customer with the highest total payments and number of orders, it may be beneficial to offer them additional discounts. To determine if the number of orders is related to the total amount paid, we should conduct further analysis by count number of orders per customer and get total payments of the customer.

Interestingly, the top two ranks are closely aligned, suggesting that offering more discounts to these customers could further increase their loyalty.

customer_id	num_of_orders	num_of_orders_rank	total_paid	total_paid_rank
12931	15	6	42055.96	1
12748	210	1	33719.73	2
12901	28	4	17654.54	3
12921	37	3	16587.09	4
12939	8	11	11581.8	5
12830	6	21	6814.64	6
12839	14	7	5591.42	7
12971	45	2	5190.74	8
12955	11	9	4757.16	9
12747	11	9	4196.01	10

Query Result Set

Query (1-4): Customers churn rate

Let's see percentage of churned customers per month

- Using 3 values (new customers per month – number of total customers at the beginning – number of customers at the end of month) and fourmula

$$(period_end_total_customers - new_customers) / period_start_total_customers * 100$$

YEAR	MONTH	PERIOD_END_TOTAL_CUSTOMERS	PERIOD_START_TOTAL_CUSTOMERS	NEW_CUSTOMERS	CHURN_RATE
2010	12	24	24	24	0
2011	01	22	24	11	45.83
2011	02	21	22	11	45.45
2011	03	31	21	15	76.19
2011	04	15	31	3	38.71
2011	05	34	15	4	200
2011	06	25	34	5	58.82
2011	07	24	25	3	84
2011	08	20	24	6	58.33
2011	09	37	20	13	120

Query (1-5): average discount percentage for product

As noticed from data: there are some products has different prices for different invoices which is referring to there are discounts on these products.

As per query for example:

```
select * from tableretail
where stockcode = '84879';
```

	INVOICE	STOCKCODE	QUANTITY	INVOICEDATE	PRICE	CUSTOMER_ID	COUNTRY
▶	562439	84879	2880	8/4/2011 18:06	1.45	12931	United Kingdom
	563074	84879	968	8/11/2011 15:58	1.45	12931	United Kingdom
	575335	84879	1200	11/9/2011 13:56	1.45	12931	United Kingdom
	560483	84879	32	7/19/2011 10:49	1.69	12955	United Kingdom
	566023	84879	48	9/8/2011 12:48	1.69	12955	United Kingdom
	578118	84879	16	11/23/2011 10:08	1.69	12955	United Kingdom

Same Product with different prices

So let's get the average discount rate for each product, we have got every product with total sales with respect to the maximum price (we considered that max price of the product is the price before discount) and actual total sales, and with formula: $(\text{total-actual} / \text{total}) * 100$ we have got average discount percentage

Note: the product with stockcode 'M' seems like not a real one or a random entry so we will exclude it

	PRO...	TOTAL_SOLD_QUANTITY	ACTUAL_PRICE	TOTAL_SALES_AFTER_DISCOUNT	TOTAL_SALES_BF_DISCOUNT	Avg_discount_percentage%
	22095	22	1.25	12.02	27.5	56.3
	22645	50	1.45	32.22	72.5	55.6
	21111	25	2.95	32.95	73.75	55.3
	35471D	10	1.25	5.62	12.5	55
	21179	5	1.25	2.81	6.25	55
	21843	154	10.95	768.9	1686.3	54.4
	84341B	37	0.85	14.53	31.45	53.8
	47504K	42	1.95	37.82	81.9	53.8
	21993	15	2.95	20.45	44.25	53.8
	22402	27	1.25	15.69	33.75	53.5
	20996	128	0.42	25.01	53.76	53.5
▶	21109	16	6.75	50.4	108	53.3

Sample from result set of query

Query (1-6): Most frequent combination of products

To build good recommendation system for our store based on customer's behavior, we need to know which combination of products is being repeated over time.

PRODUCT	BOUGHT_WITH	TIMES_BOUGHT_TOGETHER
22355	20724	23
20724	22355	23
20728	20725	22
20725	20728	22
82482	82494L	21
20725	22384	21
82494L	82482	21
20719	22355	21

As we see product 22355 with 20724 is the most repeated combination so we have to recommend these combinations in our stores

Query (2-1):

As required to implement a Monetary model for customers behavior for product purchasing and segment each customer based on the below groups

Champions - Loyal Customers - Potential Loyalists – Recent Customers – Promising - Customers Needing Attention - At Risk - Cant Lose Them – Hibernating – Lost

Assumptions:

- Monetary has been calculated with $(\text{sum}(\text{price} * \text{quantity}))$ for every customer
- FM_score taken as $(\text{f_score} + \text{m_score} / 2)$
- adding Rule For (2-1) customer segment

CUSTOMER_ID	RECENCY	FREQUENCY	MONETARY	R_SCORE	F_SCORE	M_SCORE	FM_SCORE	CUST_SEGMENT
12747	2	11	4196.01	5	5	5	5	Champions
12748	0	210	33719.73	5	5	5	5	Champions
12749	3	5	4090.88	5	4	5	5	Champions
12820	3	4	942.34	5	4	3	4	Champions
12821	214	1	92.72	1	1	1	1	Lost
12822	70	2	948.88	3	2	3	3	Potential Loyalists
12823	74	5	1759.5	2	4	4	4	At Risk
12824	59	1	397.12	3	1	2	2	Customers Needing attention
12826	2	7	1474.72	5	5	4	5	Champions
12827	5	3	430.15	5	3	2	3	Loyal Customers
12828	2	6	1018.71	5	4	3	4	Champions
12829	336	2	293	1	3	1	2	Hibernating

Sample from result set

Query (3-1):

As per required the maximum number of consecutive days a customer made purchases

	CUST_ID	MAX_CONSECUTIVE_DAYS
▶	100010376	5
	100011085	10
	100014033	46
	100018482	3
	100020880	46
	100035887	13
	100054374	8
	100070652	1
	100077596	2
	100087785	61
	100105254	10
	100135808	15

Query (3-2):

As per required the average days/transactions it take a customer to reach a spent threshold of 250 L.E

Assumptions:

- interval calculated based on the period between first transaction and the day customer reach 250 or more

	AVG_THRESHOLD_DAYS
▶	12