

Исследование данных торговой компании с помощью SQL

Этап 1. Исследование полученных данных

Этап 2. Исправление ошибок в данных

Этап 3. Анализ финансовых показателей

Этап 4. Модернизация структуры БД

Этап 5. Расчет остатков на складе

Этап 6. Импорт дополнительной порции данных

Этап 1

Исследуйте структуру таблицы с данными, определите список столбцов, типы данных.

№	название колонки	Тип данных
1	dt	date
2	tm	time
3	order_number	int
4	order_type	varchar
5	product_category	varchar
6	product	varchar
7	manufacturer	varchar
8	cnt	int
9	price	float
10	selling_price	float

Исследуйте данные каждого поля. Результат оформите в виде таблицы:

№	Название колонки	Характеристика данных	Назначение
1	dt	Временной период 30.07.2000 – 16.08.2001*	Судя по массиву данных, вероятнее всего даты в данном столбце означают дату продажи товара
2	tm	Время сделок между 09:17 и 17:37	вероятнее всего время в данном столбце означает время регистрации сделки по продаже товара. Все сделки регистрировались в рабочее время.
3	order_number	Имеет значение от 1 до 102, количество 102	Судя по данным значения представляют порядковые номера сделок, часть номеров повторяется, всего 102 ордера
4	order_type	2 вида значений	Вид документа, по которому проводятся продажи
5	product_category	6	категорий видов товара
6	product	22	вида продуктов
7	manufacturer	6	Количество производителей
8	cnt	**Min = 1, Max = 5800, Avg = 747	Количество одного вида проданного товара
9	price	*** Min=-90, Max=90	Цена товара
10	selling_price	****Min=3.28, Max=97.69	Цена продаваемого товара

*В датах, очевидно, присутствуют ошибки, обнаружены явно некорректные даты (е.g. 08.09.1877), в ячейке представлены даты после исключения из анализа не валидных дат. Для более детальной информации по анализу не валидных данных см Задание 4 пункт g.

**Предположительно в данных присутствует ошибка, цифра максимального количества товара выглядит недостоверной 999999999, строк с такими данными – 7, все относятся к разным сделкам, совершены в разные даты. Данные в ячейке подсчитаны после исключения из анализа недостоверных цифр.

*** В столбце с ценой товара есть значения с отрицательной стоимостью.

****В цене продаваемого товара есть значение null, данные в ячейке представлены после исключения этого значения из анализа.

Этап 2

Найдите и исправьте все ошибки в данных. Варианты исправления перечислены в рекомендациях к данному этапу.

После исправления ошибок определите некоторые характеристики данных

№	Название характеристики	результат
1	Количество строк в таблице	329
2	Количество строк в распределении по каждой категории товара	Construction chemistry = 48, Dry building mixes = 89, Insulation materials = 79, Two-component mortar = 11, Varnishes and paints = 102
3	Количество строк в распределении по каждому производителю	Ceresit = 41, Knauf = 64, Makroflex = 43, Rockwool = 79, Tikkurila = 102
8	Минимальное, максимальное и среднее значения по колонке cnt	**Min = 1, Max = 5800, Avg = 751
9	Минимальное, максимальное и среднее значения по колонке price	*** Min=-3,2, Max=90, Avg = 16,86

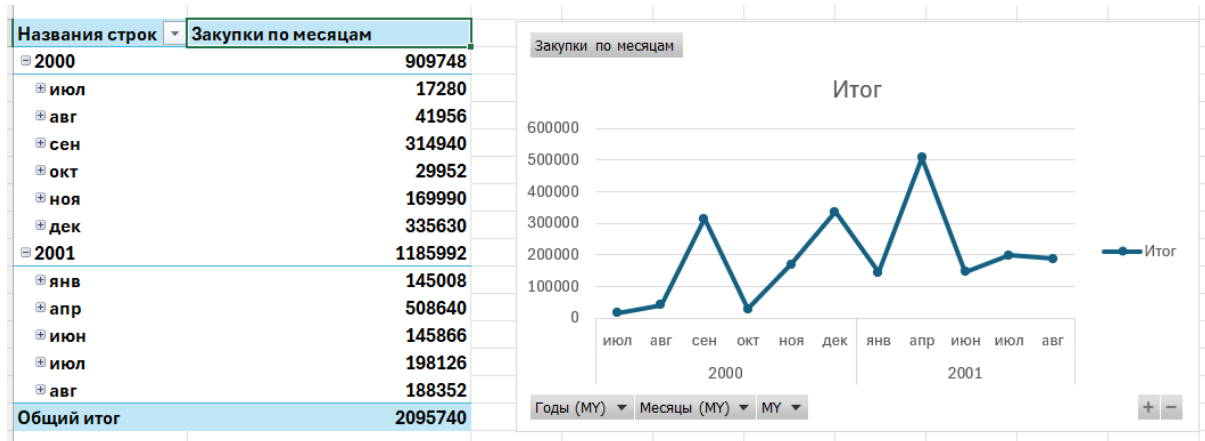
Этап 3

Для оценки динамики закупок рассчитайте показатели:

Количество накладных. Общая сумма закупленного товара

период	количество накладных	сумма закупки
07-2000	1	17280
08-2000	2	41956
09-2000	5	314940
10-2000	1	29952
11-2000	7	169990
12-2000	7	335630
01-2001	6	145008
04-2001	2	508640

06-2001	5	145866
07-2001	3	198126
08-2001	3	188352

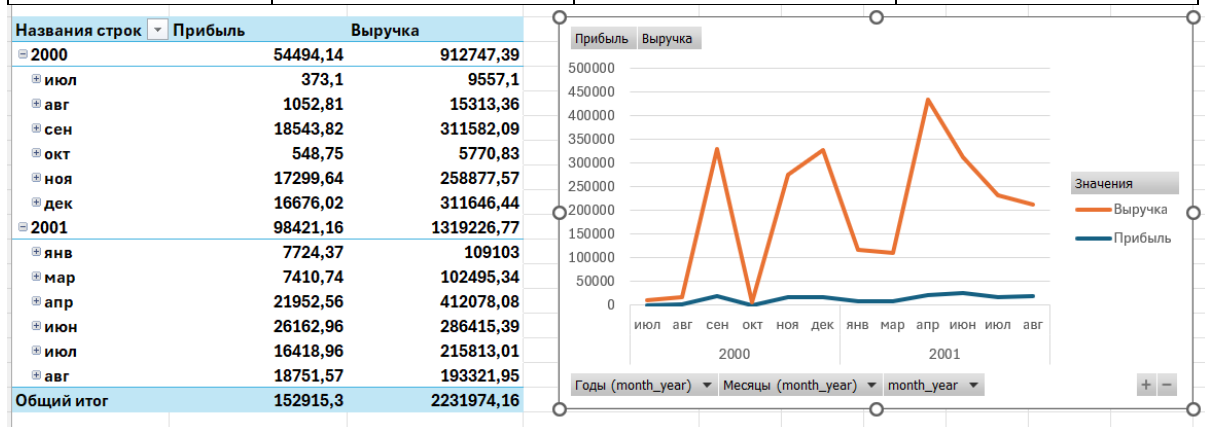


Посчитайте прибыль компании за весь период. - Прибыль за весь период - 159915,3

Для оценки динамики рассчитайте по месяцам следующие показатели:

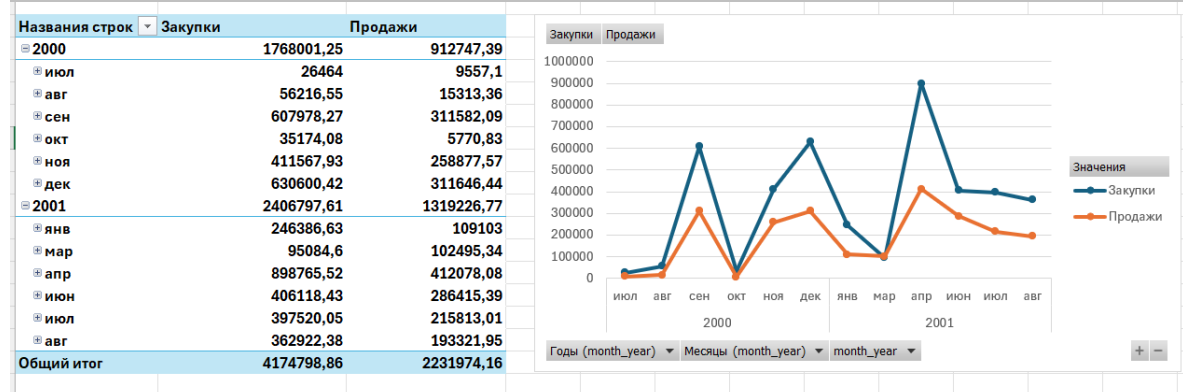
- Количество проданных товаров.
- Сумма выручки проданного товара.
- Прибыль

Период	Количество	Выручка	Прибыль
07-2000	2870	9557,1	373,1
08-2000	2899	15313,36	1052,81
09-2000	20801	311582,09	18543,82
10-2000	761	5770,83	548,75
11-2000	19134	258877,57	17299,64
12-2000	17267	311646,44	16676,02
01-2001	10848	109103	7724,37
03-2001	11816	102495,34	7410,74
04-2001	4397	412078,08	21952,56
06-2001	20753	286415,39	26162,96
07-2001	5726	215813,01	16418,96
08-2001	5326	193321,95	18751,57



Рассчитайте ежемесячно суммы, на которые был закуплен товар, и суммы, на которые был продан товар.

Период	Закупки	Продажи
07-2000	26464	9557,1
08-2000	56216,55	15313,36
09-2000	607978,27	311582,09
10-2000	35174,08	5770,83
11-2000	411567,93	258877,57
12-2000	630600,42	311646,44
01-2001	246386,63	109103
03-2001	95084,6	102495,34
04-2001	898765,52	412078,08
06-2001	406118,43	286415,39
07-2001	397520,05	215813,01
08-2001	362922,38	193321,95

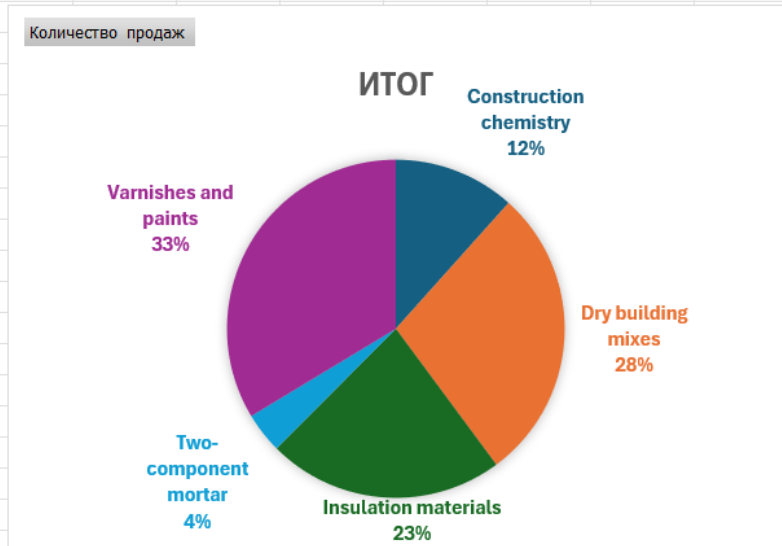


За весь период рассчитайте по каждой категории товара следующие параметры:

- **Количество проданных товаров.**
- **Сумма выручки проданного товара.**
- **Прибыль**

Категория товара	Количество	Выручка	Прибыль
Construction chemistry	28800	63974,95	10354,95
Dry building mixes	69800	333886,58	24486,58
Insulation materials	55998	962087,9	66525,04
Two-component mortar	9600	456399,36	24399,36
Varnishes and paints	83200	415625,37	27149,37

Названия строк	Количество продаж
Construction chemistry	28800
Dry building mixes	69800
Insulation materials	55998
Two-component mortar	9600
Varnishes and paints	83200
Общий итог	247398



Этап 4

Создайте следующие таблицы-справочники

Заполните созданные таблицы-справочники данными из соответствующих столбцов таблицы `invoice_order_operations`

Создайте таблицу для данных

Заполните таблицу `operations_data` из таблицы `invoice_order_operations` данными из таблицы и всех таблиц-справочников

Выполните к новой таблице `operations_data` запросы, определяющие некоторые характеристики данных

```
1 select top 3 * from invoice_order_operations
```

tm	order_number	order_type	product_category	product	manufacturer	cnt	price	selling_price
0-07-30	09:32:00.0000000	1 purchase order	Dry building mixes	fugen	Knauf	5400	3.2	null
0-07-30	09:55:00.0000000	1 sales invoice	Dry building mixes	fugen	Knauf	2870	3.2	3.33
0-08-04	10:10:00.0000000	2 sales invoice	Dry building mixes	fugen	Knauf	2336	3.2	3.48

```
1 CREATE TABLE order_type
2 (
3     order_type_id int NOT NULL IDENTITY,
4     order_type varchar (280) DEFAULT NULL,
5     PRIMARY KEY (order_type_id)
6 )
```

```
1 insert into order_type (order_type)
2 select distinct order_type from invoice_order_operations order by order_type
```

2 rows affected

```
1 select top 2 * from order_type
```

order_type_id	order_type
1	purchase order
2	sales invoice

```
1 CREATE TABLE product_category
2 (
3     product_category_id int NOT NULL IDENTITY,
4     product_category varchar (280) DEFAULT NULL,
5     PRIMARY KEY (product_category_id)
6 )
```

```
1 insert into product_category (product_category)
2 select distinct product_category from invoice_order_operations order by product_category
```

5 rows affected

```
1 CREATE TABLE product
2 (
3     product_id int NOT NULL IDENTITY,
4     product varchar (280) DEFAULT NULL,
```

<pre>1 CREATE TABLE manufacturer 2 (3 manufacturer_id int NOT NULL IDENTITY, 4 manufacturer varchar (280) DEFAULT NULL, 5 PRIMARY KEY (manufacturer_id) 6)</pre>																																									
<pre>1 insert into manufacturer (manufacturer) 2 select distinct manufacturer from invoice_order_operations order by manufacturer</pre>	5 rows affected																																								
<pre>1 create table operations_data (2 dt date, 3 tm time, 4 order_number int, 5 order_type_id int, 6 product_category_id int, 7 product_id int, 8 manufacturer_id int, 9 cnt int, 10 price float, 11 selling_price float) 12</pre>																																									
<pre>1 select * from operations_data</pre>	<table><tr><th>dt</th><th>tm</th><th>order_number</th><th>order_type_id</th><th>product_category_id</th><th>product_id</th><th>manufacturer_id</th><th>cnt</th><th>price</th><th>selling_price</th></tr></table>	dt	tm	order_number	order_type_id	product_category_id	product_id	manufacturer_id	cnt	price	selling_price																														
dt	tm	order_number	order_type_id	product_category_id	product_id	manufacturer_id	cnt	price	selling_price																																
<pre>1 select top 3 dt, tm, order_number, 2 ot.order_type_id, 3 pc.product_category_id, 4 p.product_id, 5 m.manufacturer_id, 6 cnt, price, selling_price 7 from invoice_order_operations od 8 left join order_type ot on ot.order_type=od.order_type 9 left join product_category pc on pc.product_category=od.product_category 10 left join product p on p.product=od.product 11 left join manufacturer m on m.manufacturer=od.manufacturer</pre>	<table><tr><th>dt</th><th>tm</th><th>order_number</th><th>order_type_id</th><th>product_category_id</th><th>product_id</th><th>manufacturer_id</th><th>cnt</th><th>price</th><th>selling_price</th></tr><tr><td>2000-07-30</td><td>09:32:00.0000000</td><td>1</td><td>1</td><td>2</td><td>5</td><td>2</td><td>5400</td><td>3.2</td><td></td></tr><tr><td>2000-07-30</td><td>09:55:00.0000000</td><td>1</td><td>2</td><td>2</td><td>5</td><td>2</td><td>2870</td><td>3.2</td><td></td></tr><tr><td>2000-08-04</td><td>10:10:00.0000000</td><td>2</td><td>2</td><td>2</td><td>5</td><td>2</td><td>2336</td><td>3.2</td><td></td></tr></table>	dt	tm	order_number	order_type_id	product_category_id	product_id	manufacturer_id	cnt	price	selling_price	2000-07-30	09:32:00.0000000	1	1	2	5	2	5400	3.2		2000-07-30	09:55:00.0000000	1	2	2	5	2	2870	3.2		2000-08-04	10:10:00.0000000	2	2	2	5	2	2336	3.2	
dt	tm	order_number	order_type_id	product_category_id	product_id	manufacturer_id	cnt	price	selling_price																																
2000-07-30	09:32:00.0000000	1	1	2	5	2	5400	3.2																																	
2000-07-30	09:55:00.0000000	1	2	2	5	2	2870	3.2																																	
2000-08-04	10:10:00.0000000	2	2	2	5	2	2336	3.2																																	
<pre>1 insert into operations_data 2 select dt, tm, order_number, 3 ot.order_type_id, 4 pc.product_category_id, 5 p.product_id, 6 m.manufacturer_id, 7 cnt, price, selling_price 8 from invoice_order_operations od 9 left join order_type ot on ot.order_type=od.order_type 10 left join product_category pc on pc.product_category=od.product_category 11 left join product p on p.product=od.product 12 left join manufacturer m on m.manufacturer=od.manufacturer</pre>	329 rows affected																																								
<pre>1 select top 3 * from operations_data</pre>	<table><tr><th>dt</th><th>tm</th><th>order_number</th><th>order_type_id</th><th>product_category_id</th><th>product_id</th><th>manufacturer_id</th><th>cnt</th><th>price</th><th>selling_price</th></tr><tr><td>2000-07-30</td><td>09:32:00.0000000</td><td>1</td><td>1</td><td>2</td><td>5</td><td>2</td><td>5400</td><td>3.2</td><td></td></tr><tr><td>2000-07-30</td><td>09:55:00.0000000</td><td>1</td><td>2</td><td>2</td><td>5</td><td>2</td><td>2870</td><td>3.2</td><td></td></tr><tr><td>2000-08-04</td><td>10:10:00.0000000</td><td>2</td><td>2</td><td>2</td><td>5</td><td>2</td><td>2336</td><td>3.2</td><td></td></tr></table>	dt	tm	order_number	order_type_id	product_category_id	product_id	manufacturer_id	cnt	price	selling_price	2000-07-30	09:32:00.0000000	1	1	2	5	2	5400	3.2		2000-07-30	09:55:00.0000000	1	2	2	5	2	2870	3.2		2000-08-04	10:10:00.0000000	2	2	2	5	2	2336	3.2	
dt	tm	order_number	order_type_id	product_category_id	product_id	manufacturer_id	cnt	price	selling_price																																
2000-07-30	09:32:00.0000000	1	1	2	5	2	5400	3.2																																	
2000-07-30	09:55:00.0000000	1	2	2	5	2	2870	3.2																																	
2000-08-04	10:10:00.0000000	2	2	2	5	2	2336	3.2																																	
<pre>1 select count(*) lines from operations_data</pre>	<table><tr><th>lines</th></tr><tr><td>329</td></tr></table>	lines	329																																						
lines																																									
329																																									
<pre>1 select count(distinct(product_category_id)) quantity_product_category_id from operations_data</pre>	<table><tr><th>quantity_product_category_id</th></tr><tr><td>5</td></tr></table>	quantity_product_category_id	5																																						
quantity_product_category_id																																									
5																																									
<pre>1 select count(distinct(manufacturer_id)) quantity_manufacturer_id from operations_data</pre>	<table><tr><th>quantity_manufacturer_id</th></tr><tr><td>5</td></tr></table>	quantity_manufacturer_id	5																																						
quantity_manufacturer_id																																									
5																																									
<pre>1 select min(cnt) min_cnt, max(cnt) max_cnt, avg(cnt) avg_cnt from operations_data</pre>	<table><tr><th>min_cnt</th><th>max_cnt</th><th>avg_cnt</th></tr><tr><td>1</td><td>5800</td><td>751</td></tr></table>	min_cnt	max_cnt	avg_cnt	1	5800	751																																		
min_cnt	max_cnt	avg_cnt																																							
1	5800	751																																							
<pre>1 select min(price) min_price, max(price) max_price, round(avg(price),2) avg_price 2 from operations_data</pre>	<table><tr><th>min_price</th><th>max_price</th><th>avg_price</th></tr><tr><td>3.2</td><td>90</td><td>16.86</td></tr></table>	min_price	max_price	avg_price	3.2	90	16.86																																		
min_price	max_price	avg_price																																							
3.2	90	16.86																																							

Этап 5

Рассчитать текущие остатки товара на складе торговой компании

Проанализируйте движение товара за весь период и рассчитайте текущее состояние склада по каждой товарной позиции

```
1 select pr.product product, g.cnt_purchase cnt_purchase, g.sum_purchase sum_purchase,
2 g.cnt_selling cnt_selling, g.sum_selling sum_selling,
3 g.cnt_purchase-g.cnt_selling stoke,
4 g.sum_selling-sum_purchase profit
5 from depot_analysis_general g
6 left join product pr on g.product_id=pr.product_id
```

product	cnt_purchase	sum_purchase	cnt_selling	sum_selling	stoke	profit
epoxy grouts	4800	432000	4800	456399.36	0	24399.36
everal aqua 10	3800	37544	3800	40903.44	0	3359.44
everal aqua 10 interior	8400	82992	8400	88123.77	0	5131.76999999998
everal aqua 40	6000	59280	6000	62781.22	0	3501.21999999999
fugen	19000	60800	19000	66955.58	0	6155.57999999997
glue cm 11	5000	34950	5000	38012.43	0	3062.42999999999
glue cm 17	3000	50790	3000	55840.52	0	5050.52
grout	400	6596	400	6997.36	0	401.360000000001
helimi 10	3200	29952	3200	32331.99	0	2379.99
helimi 30	3800	35568	3800	38905.92	0	3337.92
helimi primer	5400	45900	5400	48411.92	0	2511.92
kiva 10	1200	10608	1200	11228.62	0	620.620000000001
kiva 30	5000	44200	5000	46879.44	0	2679.44000000001
kiva 70	4800	42432	4800	46059.05	0	3627.05
partial fill cavity slab 100	7600	337972	7600	359426.97	0	21454.97
partial fill cavity slab 50	3200	71072	3200	77110.22	0	6038.22
partial fill cavity slab 80	9000	320130	8998	349281.06	2	29151.0600000001
polyurethane foam	4400	16016	4400	18838.18	0	2822.18
polyurethane foam premium of winter	9600	31008	9600	38139.41	0	7131.41
rockclose insulated dpc 20	8200	166460	8200	176269.65	0	9809.64999999999
rotband	6800	162860	6800	173078.05	0	10218.05
uniflott	2200	16610	0	0	2200	-16610

```
1 select sum(sum_selling-sum_purchase) current_general_profit from depot_analysis_general
```

current_general_profit
136234.16