Исследование данных торговой компании с помощью SQL

Постановка задачи

Задача проекта — на основе предоставленных первичных данных с помощью SQL запросов подготовить аналитическую справку по деятельности торговой компании «СтройСнаб».

Решение задачи проводилось поэтапно:
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 1. Исследование полученных данных

На данном этапе было проведено первичное исследование полученных данных.

```
1  SELECT TOP 5 * FROM  invoice_order_operations
```

dt	tm	order_number	order_type	product_category	product	manufacturer	cnt	price	selling_price
2000-07-30	09:32:00.0000000	1	purchase order	Dry building mixes	fugen	Knauf	5400	3.2	*null*
2000-07-30	09:55:00.0000000	1	sales invoice	Dry building mixes	fugen	Knauf	2870	3.2	3.33
2000-08-04	10:10:00.0000000	2	sales invoice	Dry building mixes	fugen	Knauf	2336	3.2	3.48
2000-08-04	11:04:00.0000000	3	sales invoice	Dry building mixes	fugen	Knauf	26	3.2	3.58
2000-08-04	11:21:00.0000000	4	sales invoice	Dry building mixes	fugen	Knauf	9	3.2	3.28

```
1  SELECT
2  column_name,
3  data_type,
4  CHARACTER_MAXIMUM_LENGTH
5  FROM INFORMATION_SCHEMA.COLUMNS
6  WHERE table_name = 'invoice_order_operations'
```

column_name	data_type	CHARACTER_MAXIMUM_LENGTH
dt	date	*null*
tm	time	*null*
order_number	int	*null*
order_type	varchar	50
product_category	varchar	250
product	varchar	250
manufacturer	varchar	250
cnt	int	*null*
price	float	*null*
selling_price	float	*null*

Ссылка на среду выполнения

Было выяснено, что структура таблицы invoice_order_operations состоит из 10 колонок и содержит 368 строк данных.

```
1  SELECT COUNT( * ) FROM  invoice_order_operations
```

(No column name)
368

```
1  SELECT table_name,
2    row_number() over(order by table_name) AS number,
3    column_name, data_type FROM information_schema.columns
4
```

table_name	number	column_name	data_type
invoice_order_operations	1	dt	date
invoice_order_operations	2	tm	time
invoice_order_operations	3	order_number	int
invoice_order_operations	4	order_type	varchar
invoice_order_operations	5	product_category	varchar
invoice_order_operations	6	product	varchar
invoice_order_operations	7	manufacturer	varchar
invoice_order_operations	8	cnt	int
invoice_order_operations	9	price	float
invoice_order_operations	10	selling_price	float

Таблица 1 - Структура таблицы данных, список столбцов, типы данных

№	Название колонки	Тип данных
1	dt	date
2	tm	time
3	order_number	int
4	order_type	varchar(50)
5	product_category	varchar(250)
6	product	varchar(250)
7	manufacturer	varchar(250)
8	cnt	int
9	price	float
10	selling_price	float

Далее был проведен анализ содержимого полей таблицы с целью подготовки ко второму этапу проекта - «Исправление ошибок в данных»

```sql
SELECT MIN(order_number), MAX(order_number)
FROM invoice_order_operations
```

(No column name)	(No column name)
1	102

```sql
SELECT MIN(cnt), MAX(cnt)
FROM invoice_order_operations
```

(No column name)	(No column name)
1	999999999

```sql
SELECT MIN(price), MAX(price), AVG(price)
FROM invoice_order_operations
```

(No column name)	(No column name)	(No column name)
-90	90	15.2807608695652

```sql
SELECT MIN(selling_price), MAX(selling_price), AVG(selling_price)
FROM invoice_order_operations
WHERE order_type != 'purchase order'
```

(No column name)	(No column name)	(No column name)
3.28	97.69	18.8518885448916

```sql
SELECT DISTINCT manufacturer
FROM invoice_order_operations
```

manufacturer
Ceresit
Knauf
Makroflex
Rockwool
Tikkurila
Tikurila

Таблица 2 – Данные полей

№	Название колонки	Характеристика данных	Назначение
1	dt	значения от 1877-09-08 до 2001-08-16, есть аномалии	содержит дату заказа
2	tm	значения от 09:17:00 до 17:37:00	содержит время заказа
3	order_number	значения от 1 до 102	содержит номер заказа
4	order_type	значения либо закупка (purchase order) либо продажа товара (sales invoice)	указывает тип заказа
5	product_category	5 категорий товаров, есть дубли в названии insulation material	содержит категорию товара
6	product	22 наименования товара	указывает название товара
7	manufacturer	5 поставщиков, есть дубли	указывает производителя товара
8	cnt	значения от 1 до 999 999 999, есть аномалии в количестве 7 со значением 999 999 999	указывает количество товара
9	price	значения от -90 до 90 со средним 15.28, есть аномалии (отрицательные) цены	указывает цену закупки товара
10	selling_price	значения от 3.28 до 97.69 со средним 18.85, для операции закупки цена не указывается (null)	указывает цену продажи товара

Этап 2. Исправление ошибок в данных

Таблица 3 – Исправление ошибок в данных

№	Ошибки в данных	Исправление
1	Колонка dt содержит 16 записей с аномальными значения года совершения сделок	DELETE FROM invoice_order_operations WHERE dt < '2000-01-01'
2	Колонка cnt имеет аномалии в количестве 7 со значением 999 999 999	DELETE FROM invoice_order_operations WHERE cnt = 999999999
3	Колонка price имеет отрицательные значения цены в количестве 16 записей	DELETE FROM invoice_order_operations WHERE price <= 0
4	Колонка product_category содержит дубли в названии insulation material	UPDATE invoice_order_operations SET product_category = REPLACE(product_category, 'insulation Material', 'Insulation materials') WHERE product_category LIKE 'insulation Material'
5	Колонка manufacturer содержит дубли в наименовании производителя краски Тиккурила (Tikkurila)	UPDATE invoice_order_operations SET manufacturer = REPLACE(manufacturer, 'Tikurila', 'Tikkurila') WHERE manufacturer LIKE 'Tikurila'

Таким образом, выявленные на первом этапе анализа ошибки в данных были исправлены. Далее был проведен расчет характеристик исправленной таблицы invoice_order_operations

Таблица 4 – Коды для расчета характеристик данных invoice_order_operations

№	Название характеристики	Код SQL
1	Количество строк таблицы	SELECT COUNT(*) FROM invoice_order_operations
2	Количество строк в распределении по каждой категории товара	SELECT product_category, COUNT(*) FROM invoice_order_operations GROUP BY product_category
3	Количество строк в распределении по каждому производителю	SELECT manufacturer, COUNT(*) FROM invoice_order_operations GROUP BY manufacturer
4	Минимальное, максимальное и среднее значения по колонке cnt	SELECT MIN(cnt) min, MAX(cnt) max, ROUND(AVG(CONVERT(FLOAT, cnt)), 2) avg FROM invoice_order_operations
5	Минимальное, максимальное и среднее значения по колонке price	SELECT MIN(price) min, MAX(price) max, ROUND(AVG(price),2) avg FROM invoice_order_operations

В результате проведенной очистки данных было потеряно 10,6 % первоначального объема данных (количество строк в таблице invoice_order_operations сократилось с 368 до 329 строк)

```
1  SELECT 100-CONVERT(FLOAT, COUNT(*))/368*100 percentage FROM invoice_order_operations
```

percentage
10.5978260869565

Таблица 5 – Характеристики исправленной таблицы invoice_order_operations

№	Название характеристики	Результат
1	Количество строк таблицы	329
2	Количество строк в распределении по каждой категории товара	5
3	Количество строк в распределении по каждому производителю	5
4	Минимальное, максимальное и среднее значения по колонке cnt	min: 1, max: 5800, avg: 751.97
5	Минимальное, максимальное и среднее значения по колонке price	min: 3.2, max: 90, avg: 16.86

product_category	(No column name)
Construction chemistry	48
Dry building mixes	89
Insulation materials	79
Two-component mortar	11
Varnishes and paints	102

manufacturer	(No column name)
Ceresit	41
Knauf	64
Makroflex	43
Rockwool	79
Tikkurila	102

Этап 3. Анализ финансовых показателей

На данном этапе была произведена оценка динамики закупок по месяцам

Таблица 6 – Расчет динамики закупок

Период	Количество накладных	Сумма закупки
2000-07	1	17 280
2000-08	2	41 956
2000-09	5	314 940
2000-10	1	29 952
2000-11	7	169 990
2000-12	7	335 630
2001-01	6	145 008
2001-04	2	508 640
2001-06	5	145 866
2001-07	3	198 126
2001-08	3	188 352

```
SELECT FORMAT(dt,'yyyy-MM') period,
COUNT(order_number) orders,
SUM(cnt*price) orders_cost
FROM invoice_order_operations WHERE order_type = 'purchase order’
GROUP BY FORMAT(dt,'yyyy-MM’) ORDER BY FORMAT(dt,'yyyy-MM')
```

Оценка динамики продаж по месяцам

Таблица 7 – Расчет динамики продаж

Период	Количество	Выручка	Прибыль
2000-07	2 870	9 557,1	373,1
2000-08	2 899	15 313,36	1 052,81
2000-09	20 801	311 582,09	18 543,82
2000-10	761	5 770,83	548,75
2000-11	19 134	258 877,57	17 299,64
2000-12	17 267	311 646,44	16 676,02
2001-01	10 848	109 103	7 724,37
2001-03	11 816	102 495,34	7 410,74
2001-04	4 397	412 078,08	21 952,56
2001-06	20 753	286 415,39	26 162,96
2001-07	5 726	215 813,01	16 418,96
2001-08	5 326	193 321,95	18 751,57

```
SELECT FORMAT(dt,'yyyy-MM') period, SUM(cnt) cnt,
ROUND(SUM(cnt*selling_price),2) revenue,
ROUND(SUM(cnt*selling_price) - SUM(cnt*price),2) sales_profit
FROM invoice_order_operations
WHERE order_type = 'sales invoice'
GROUP BY FORMAT(dt,'yyyy-MM’)
ORDER BY FORMAT(dt,'yyyy-MM')
```

Расчет помесячно суммы, на которые был закуплен товар, и суммы, на которые был продан товар в одной таблице с помощью функции COALESCE

```
SELECT COALESCE (t1.period, t2.period) period, orders_cost, revenue
FROM (SELECT FORMAT(dt,'yyyy-MM') period, SUM(cnt*price) orders_cost FROM invoice_order_operations
WHERE order_type = 'purchase order’ GROUP BY FORMAT(dt,'yyyy-MM’) ) t1
FULL JOIN (
SELECT FORMAT(dt,'yyyy-MM') period, ROUND(SUM(cnt*selling_price),2) revenue FROM invoice_order_operations
WHERE order_type = 'sales invoice’ GROUP BY FORMAT(dt,'yyyy-MM’) ) t2
ON t1.period = t2.period
```

period	orders_cost	revenue
2000-07	17280	9557.1
2000-08	41956	15313.36
2000-09	314940	311582.09
2000-10	29952	5770.83
2000-11	169990	258877.57
2000-12	335630	311646.44
2001-01	145008	109103
2001-03	*null*	102495.34
2001-04	508640	412078.08
2001-06	145866	286415.39
2001-07	198126	215813.01
2001-08	188352	193321.95

Расчет прибыли компании за весь период (как сумма разниц стоимости продажи и закупки каждого проданного товара)

Таблица 8 – Закупки и продажи товаров

Период	Закупка	Продажа
2000-07	17 280	9 557,1
2000-08	41 956	15 313,36
2000-09	314 940	311 582,09
2000-10	29 952	5 770,83
2000-11	169 990	258 877,57
2000-12	335 630	311 646,44
2001-01	145 008	109 103
2001-03	0	102 495,34
2001-04	508 640	412 078,08
2001-06	145 866	286 415,39
2001-07	198 126	215 813,01
2001-08	188 352	193 321,95

```
1 SELECT SUM((selling_price - price)*cnt)
2 FROM invoice_order_operations
3 WHERE order_type = 'sales invoice'
```

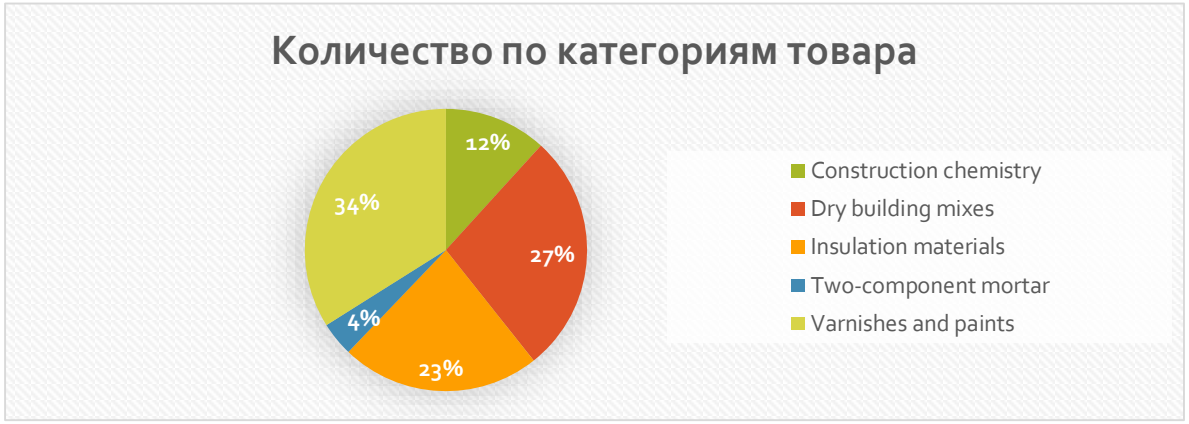
(No column name)
152915.3

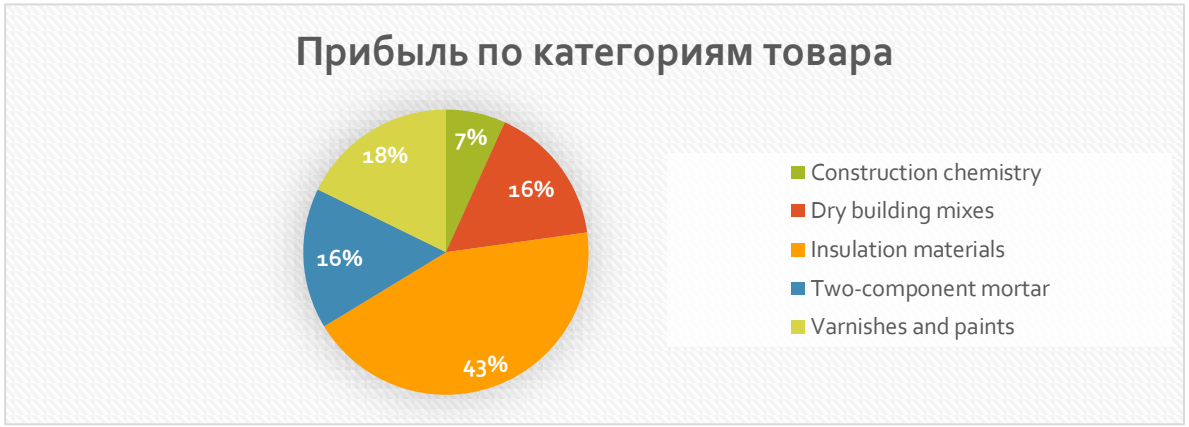
Таким образом, сумма валовóй прибыли продаж за весь анализируемый период составила 152 915,30 денежных единиц

Расчет проданного количества товара, выручки и прибыли за весь период по категориям товара

Таблица 9 – Анализ продаж по категориям товара

Категория товара	Количество	Выручка	Прибыль
Construction chemistry	14 400	63974,95	10354,95
Dry building mixes	33 800	333886,58	24486,58
Insulation materials	27 998	962087,9	66525,04
Two-component mortar	4 800	456399,36	24399,36
Varnishes and paints	41 600	415625,37	27149,37






```
SELECT product_category, SUM(cnt) total_cnt, ROUND(SUM(cnt*selling_price),2) revenue,
        ROUND(SUM(cnt*(selling_price-price)),2) profit
FROM invoice_order_operations WHERE order_type = 'sales invoice' GROUP BY product_category
```

Этап 4. Модернизация структуры БД

На данном этапе были созданы и заполнены таблицы-справочники: product, product_category, order_type, manufacturer

product_id	product
1	epoxy grouts
2	everal aqua 10
3	everal aqua 10 interior
4	everal aqua 40
5	fugen
6	glue cm 11
7	glue cm 17
8	grout
9	helmi 10
10	helmi 30
11	helmi primer
12	kiva 10
13	kiva 30
14	kiva 70
15	partial fill cavity slab 100
16	partial fill cavity slab 50
17	partial fill cavity slab 80
18	polyurethane foam
19	polyurethane foam premium of winter
20	rockclose insulated dpc 20
21	rotband
22	uniflott

CREATE TABLE product (product_id int IDENTITY(1, 1), product varchar(250));
INSERT INTO product(product)
SELECT DISTINCT product
FROM invoice_order_operations
ORDER BY product

product_category_id	product_category
1	Construction chemistry
2	Dry building mixes
3	Insulation materials
4	Two-component mortar
5	Varnishes and paints

CREATE TABLE product_category (product_category_id int IDENTITY(1, 1), product_category varchar(250));
INSERT INTO product_category(product_category) SELECT DISTINCT product_category
FROM invoice_order_operations
ORDER BY product_category

manufacturer_id	manufacturer
1	Ceresit
2	Knauf
3	Makroflex
4	Rockwool
5	Tikkurila

order_type_id	order_type
1	purchase order
2	sales invoice

Также была создана новая таблица данных operations_data и заполнена из таблицы invoice_order_operations и всех таблиц-справочников

```
CREATE TABLE operations_data(dt date, tm time, order_number int, order_type_id int,
product_category_id int,  product_id int, manufacturer_id int, cnt int, price float, selling_price float);
INSERT INTO operations_data
SELECT dt,tm,order_number,  o.order_type_id, pc.product_category_id,  p.product_id,
m.manufacturer_id,   cnt, price, selling_price
FROM invoice_order_operations a
LEFT JOIN order_type o ON a.order_type=o.order_type
LEFT JOIN product_category pc ON a.product_category=pc.product_category
LEFT JOIN product p ON a.product=p.product
LEFT JOIN manufacturer m ON a.manufacturer=m.manufacturer
```

```
1  SELECT TOP 5 * FROM operations_data
```

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	*null*
2000-07-30	09:55:00.0000000	1	2	2	5	2	2870	3.2	3.33
2000-08-04	10:10:00.0000000	2	2	2	5	2	2336	3.2	3.48
2000-08-04	11:04:00.0000000	3	2	2	5	2	26	3.2	3.58
2000-08-04	11:21:00.0000000	4	2	2	5	2	9	3.2	3.28

К новой таблице operations_data были сделаны запросы характеристик данных.

Таблица 10 – Характеристики таблицы operations_data

№	Название характеристики	Результат
1	Количество строк таблицы	329
2	Количество строк в распределении по каждой категории товара	1 48 2 89 3 79 4 11 5 102
3	Количество строк в распределении по каждому производителю	1 41 2 64 3 43 4 79 5 102
4	Минимальное, максимальное и среднее значения по колонке cnt	Min 1; Max 5800; Avg 751
5	Минимальное, максимальное и среднее значения по колонке price	Min 3,2; Max 90; Avg 16,86

Характеристики созданной таблицы operations_data полностью совпадают с характеристиками исправленной таблицы invoice_order_operations (Таблица 5)

Этап 5. Расчёт остатков на складе

product	balance_cnt
epoxy grouts	0
everal aqua 10	0
everal aqua 10 interior	0
everal aqua 40	0
fugen	0
glue cm 11	0
glue cm 17	0
grout	0
helmi 10	0
helmi 30	0
helmi primer	0
kiva 10	0
kiva 30	0
kiva 70	0
partial fill cavity slab 100	0
partial fill cavity slab 50	0
partial fill cavity slab 80	2
polyurethane foam	0
polyurethane foam premium of winter	0
rockclose insulated dpc 20	0
rotband	0
uniflott	2200

На заключительном этапе были рассчитаны остатки на складе по каждой товарной позиции

```sql
SELECT product, balance_cnt FROM product a
LEFT JOIN (SELECT product_id, SUM(
  cnt*CASE WHEN order_type_id =1 THEN 1 ELSE 0 END -
  cnt*CASE WHEN order_type_id =2 THEN 1 ELSE 0 END
  ) balance_cnt
FROM operations_data
GROUP BY product_id) b
ON a.product_id = b.product_id
```

```sql
SELECT * FROM invoice_order_operations WHERE product = 'uniflott'
```

dt	tm	order_number	order_type	product_category	product	manufacturer	cnt	price	selling_price
2001-08-16	15:02:00.0000000	42	purchase order	Dry building mixes	uniflott	Knauf	2200	7.55	*null*

Заключение

Таким образом, поэтапное решение поставленной задачи успешно реализовано с помощью SQL запросов:

- ✓ Первичные данные исследованы;
- ✓ Ошибки в данных были найдены и исправлены;
- ✓ На основе первичных данных рассчитаны финансовые показатели;
- ✓ Проведена модернизация структуры БД;
- ✓ Осуществлен расчет остатков товаров на складе

СПАСИБО ЗА ВНИМАНИЕ!