

Финальный проект по учебному курсу «SQL для аналитиков»

(Международная школа профессий, преподаватель Павел Долганов).

Задача: спроектировать, создать и наполнить значениями БД, создать аналитические запросы к ней, выгрузить данные в Google таблицу и провести когортный анализ.

Этапы создания проекта:

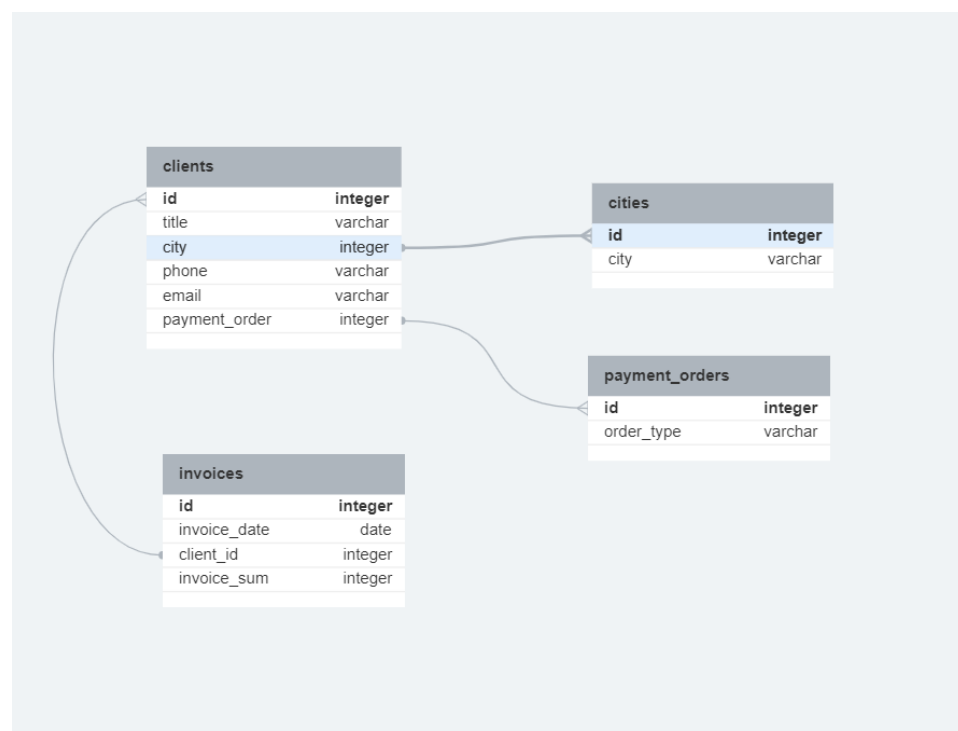
1. Выбор предметной области и назначения БД.
2. Проектирование БД в dbdesigner.net.
3. Создание БД в Oracle Live SQL.
4. Заполнение БД значениями.
5. Написание запросов для анализа данных.
6. Выгрузка данных в Google таблицу и когортный анализ.

1. Выбор предметной области и назначения БД

В качестве предметной области выбраны адаптированные для целей настоящего проекта и частично обезличенные данные об оптовых продажах сувенирной продукции в Беларуси за 2018 – 2019 год (в период ведения автором проекта индивидуальной предпринимательской деятельности).

Назначение БД – хранение и систематизация данных об отгрузках товара оптовым покупателям.

2. Проектирование БД в dbdesigner.net



3. Создание БД в Oracle Live SQL

Из dbdesigner.net был выгружен файл кодом [wholesale-sale-of-souvenirs.sql](#), для удобства дальнейшего использования в коде удалены все кавычки:

```
CREATE TABLE payment_orders (  
    id INT NOT NULL,  
    order_type VARCHAR2(70) NOT NULL,  
    constraint PAYMENT_ORDERS_PK PRIMARY KEY (id));  
  
CREATE sequence PAYMENT_ORDERS_ID_SEQ;  
  
CREATE trigger BI_PAYMENT_ORDERS_ID  
    before insert on payment_orders  
    for each row  
  
begin  
    select PAYMENT_ORDERS_ID_SEQ.nextval into :NEW.id from dual;  
end;  
  
/  
  
CREATE TABLE cities (  
    id INT NOT NULL,  
    city VARCHAR2(30) NOT NULL,  
    constraint CITIES_PK PRIMARY KEY (id));  
  
CREATE sequence CITIES_ID_SEQ;  
  
CREATE trigger BI_CITIES_ID  
    before insert on cities  
    for each row  
  
begin  
    select CITIES_ID_SEQ.nextval into :NEW.id from dual;  
end;  
  
/  
  
CREATE TABLE clients (  
    id INT NOT NULL,  
    title VARCHAR2(70) NOT NULL,  
    city INT NOT NULL,  
    phone VARCHAR2(20) NOT NULL,  
    email VARCHAR2(30) NOT NULL,  
    payment_order INT NOT NULL,  
    constraint CLIENTS_PK PRIMARY KEY (id));  
  
CREATE sequence CLIENTS_ID_SEQ;  
  
CREATE trigger BI_CLIENTS_ID  
    before insert on clients  
    for each row  
  
begin  
    select CLIENTS_ID_SEQ.nextval into :NEW.id from dual;  
end;  
  
/  
  
CREATE TABLE invoices (  
    id INT NOT NULL,  
    invoice_date DATE NOT NULL,  
    client_id INT NOT NULL,
```

```

    invoice_sum INT NOT NULL,
    constraint INVOICES_PK PRIMARY KEY (id));

CREATE sequence INVOICES_ID_SEQ;

CREATE trigger BI_INVOICES_ID
    before insert on invoices
    for each row
begin
    select INVOICES_ID_SEQ.nextval into :NEW.id from dual;
end;
/

ALTER TABLE clients ADD CONSTRAINT clients_fk0 FOREIGN KEY (city) REFERENCES cities(id);
ALTER TABLE clients ADD CONSTRAINT clients_fk1 FOREIGN KEY (payment_order) REFERENCES payment_orders(id);
ALTER TABLE invoices ADD CONSTRAINT invoices_fk0 FOREIGN KEY (client_id) REFERENCES clients(id);

```

4. Заполнение БД значениями

Для заполнения БД значениями был подготовлен файл **Опт-сувениры.xlsx**, для подготовки запросов использованы такие формулы:

=insert into payment_orders (id, order_type) values "&D3&";

=insert into cities (id, city) values "&D10&";

=insert into clients (id, title, city, phone, email, payment_order) values ("&G19&");

=("&A3&", to_date("&C3&", 'yyyy-mm-dd'), "&D3&", "&E3&")

=insert into invoices (id, invoice_date, client_id, invoice_sum) values "&G3&";

и в результате созданы следующие запросы (привожу частично, см. файл **inserting.sql**):

```

insert into cities (id, city) values (1, 'Минск');
insert into cities (id, city) values (2, 'Брест');
.....
insert into payment_orders (id, order_type) values (1, 'Оплата по мере
реализации');
insert into payment_orders (id, order_type) values (2, 'Срок оплаты 30 дней');
.....
insert into clients (id, title, city, phone, email, payment_order) values (1,
'Брестский универмаг', 2, '+375xxxxxxxx1', '1xxx@gmail.com', 1);
insert into clients (id, title, city, phone, email, payment_order) values (2,
'Брестская союзпечать', 2, '+375xxxxxxxx2', '2xxx@gmail.com', 1);
.....
insert into invoices (id, invoice_date, client_id, invoice_sum) values (1801,
to_date('2018-01-02', 'yyyy-mm-dd'), 12, 1170);
insert into invoices (id, invoice_date, client_id, invoice_sum) values (1802,
to_date('2018-01-17', 'yyyy-mm-dd'), 1, 930);
.....

```

5. Написание запросов для анализа данных

Были созданы запросы с целью подготовить данные для последующего когортного анализа (файл **queries.sql**). Первый запрос создал выборку покупателей, которые за два года сделали не более 5-и заказов или общая сумма продаж которым не превысила 4 000 белорусских рублей:

```

create view other_clients as
select cl.title, c.city, p.order_type, sum(i.invoice_sum) as INV_SUM, count(i.id)
as INV_COUNT
from invoices i
INNER JOIN clients cl ON i.client_id = cl.id
INNER JOIN cities c ON cl.city = c.id
INNER JOIN payment_orders p ON cl.payment_order = p.id
group by cl.title, c.city, p.order_type having count(i.id) < 6 or
sum(i.invoice_sum) < 4001
order by INV_SUM desc;

select * from other_clients;

```

TITLE	CITY	ORDER_TYPE	INV_SUM	INV_COUNT
ЧП Славутасть	Минск	Оплата по мере реализации	2270	4
Магазин Негоциант	Витебск	Срок оплаты 30 дней	1340	4
УП БытТоргПродукт	Ивацевичи	Срок оплаты 10 дней	1090	2
Могилевский краеведческий музей	Могилев	Оплата по мере реализации	615	2
ФК Динамо-Брест	Брест	Предоплата	360	1

Вторым запросом было создано представление с выборкой всех необходимых данных отдельно за 2018 и 2019 год по «хорошим покупателям» для последующего когортного анализа:

```

create view good_clients as
select cl.title, c.city, p.order_type,
sum(case when i.invoice_date between to_date('2017-12-31', 'yyyy-mm-dd') and
to_date('2019-01-01', 'yyyy-mm-dd')
then i.invoice_sum else 0 end) as INV_SUM_18,
count(case when i.invoice_date between to_date('2017-12-31', 'yyyy-mm-dd') and
to_date('2019-01-01', 'yyyy-mm-dd')
then i.id else null end) as INV_COUNT_18,
sum(case when i.invoice_date between to_date('2018-12-31', 'yyyy-mm-dd') and
to_date('2020-01-01', 'yyyy-mm-dd')
then i.invoice_sum else 0 end) as INV_SUM_19,
count(case when i.invoice_date between to_date('2018-12-31', 'yyyy-mm-dd') and
to_date('2020-01-01', 'yyyy-mm-dd')
then i.id else null end) as INV_COUNT_19
from invoices i
INNER JOIN clients cl ON i.client_id = cl.id
INNER JOIN cities c ON cl.city = c.id
INNER JOIN payment_orders p ON cl.payment_order = p.id

```

```
group by cl.title, c.city, p.order_type having count(i.id) > 5 and
sum(i.invoice_sum) > 4000
order by INV_SUM_18 desc;

select * from good_clients;
```

TITLE	CITY	ORDER_TYPE	INV_SUM_18	INV_COUNT_18	INV_SUM_19	INV_COUNT_19
Брестский универмаг	Брест	Оплата по мере реализации	15325	15	19090	17
Музей Несвижский замок	Несвиж	Оплата по мере реализации	14755	12	16075	12
Полоцкий музей	Полоцк	Оплата по мере реализации	8470	9	8550	9
Брестская союзпечать	Брест	Оплата по мере реализации	8355	10	9760	11
ИП Бартошевич	Витебск	Срок оплаты 30 дней	5340	7	5565	7
Белпочта	Минск	Оплата по мере реализации	3910	7	3945	7
ИП Буденко	Брест	Срок оплаты 30 дней	3620	3	4070	3
Магазин Аркада	Могилев	Срок оплаты 30 дней	2865	4	2920	4
ООО Нико-Вест	Брест	Срок оплаты 30 дней	2840	5	3475	6
Витебский универмаг	Витебск	Оплата по мере реализации	2100	4	2230	4

6. Выгрузка данных в Google таблицу и когортный анализ

Представление `good_clients` было скачано в формате файла csv, преобразовано в файл `table.xls`, данные разбиты по столбцам и скопированы в Google таблицу.
https://docs.google.com/spreadsheets/d/1pkex_5FJCJxegPoRysJyKpOxIIrVuMiIbENPHr1BraQ/edit#gid=741241786

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	Таблица 1 ОБЩИЕ СУММЫ ПРОДАЖ ПО ГОРОДАМ								Таблица 2 МАКСИМАЛЬНАЯ И СРЕДНЯЯ СУММЫ ПРОДАЖ ПО ГОРОДАМ					
2	ГОРОД	ПОКУПАТЕ ЛЕЙ	Продажи 2018	Продажи 2018 (%)	Продажи 2019	Продажи 2019 (%)	Рост продаж в 2019 году		CITY	ПОКУП АТЕЛЕЙ	МАХ на 1 покупателя 2018	AVG на 1 покупателя 2018	МАХ на 1 покупателя 2019	AVG на 1 покупателя 2019
3	Брест	4	30 140,00	44,60%	36 395,00	48,09%	124,57%		Несвиж	1	14 755,00	14 755,00	16 075,00	16 075,00
4	Несвиж	1	14 755,00	21,83%	16 075,00	21,24%	108,95%		Брест	4	15 325,00	7 535,00	19 090,00	9 098,75
5	Полоцк	1	8 470,00	12,53%	8 550,00	11,30%	100,94%		Полоцк	1	8 470,00	8 470,00	8 550,00	8 550,00
6	Витебск	2	7 440,00	11,01%	7 795,00	10,30%	104,21%		Минск	1	3 910,00	3 910,00	3 945,00	3 945,00
7	Минск	1	3 910,00	5,79%	3 945,00	5,21%	100,90%		Витебск	2	5 340,00	3 720,00	5 565,00	3 897,50
8	Могилев	1	2 865,00	4,24%	2 920,00	3,86%	101,92%		Могилев	1	2 865,00	2 865,00	2 920,00	2 920,00
9	Итого	10	67 580,00	100,00%	75 680,00	100,00%	124,57%							
10									* Исходя из Таблицы 1 следует определить Брест и Несвиж как города с максимальными суммами продаж, они же показывают и наилучший рост по отношению к предыдущему году. Также хорошие результаты мы видим в Полоцке и Витебске. Поставкам в эти 4 города следует отдать приоритет. * Таблица 2 показывает, что не следует использовать для сравнительного анализа такой показатель, как средняя сумма продаж на одного покупателя в разных городах, если мы имеем небольшое количество покупателей, особенно если в некоторых городах по 1-2 покупателя. А вот максимальная сумма продаж на одного покупателя в городе может быть полезной метрикой. Анализ этого показателя подтверждает выводы Таблицы 1. * В Таблице 3 выведены показатели продаж покупателям, сгруппированным по порядку оплаты отгруженной продукции. Как общая сумма продаж, так и средние продажи на покупателя гораздо выше по тем покупателям, которые оплачивают продукцию по мере реализации. Это можно объяснить тем, что такие покупатели не опасаются иметь более широкий ассортимент продукции и большие товарные запасы, так как им нужно оплачивать только реализованный товар.					
11	Таблица 3 ОБЩИЕ И СРЕДНИЕ СУММЫ ПРОДАЖ ПО ПОРЯДКУ ОПЛАТЫ													
12	Порядок оплаты	ПОКУПАТЕ ЛЕЙ	Продажи 2018	Продажи 2018 на покупателя	Продажи 2019	Продажи 2019 на покупател								
13	Оплата по мере реализации	6	52 915,00	8 819,17	59 650,00	9 941,67								
14	Срок оплаты 30 дней	4	14 665,00	3 666,25	16 030,00	4 007,50								
15														
16														

* Исходя из Таблицы 1 следует определить Брест и Несвиж как города с максимальными суммами продаж, они же показывают и наилучший рост по отношению к предыдущему году. Также хорошие результаты мы видим в Полоцке и Витебске. Поставкам в эти 4 города следует отдать приоритет.

* Таблица 2 показывает, что не следует использовать для сравнительного анализа такой показатель, как средняя сумма продаж на одного покупателя в разных городах, если мы

имеем небольшое количество покупателей, особенно если в некоторых городах по 1-2 покупателя. А вот максимальная сумма продаж на одного покупателя в городе может быть полезной метрикой. Анализ этого показателя подтверждает выводы Таблицы 1.

* В Таблице 3 выведены показатели продаж покупателям, сгруппированным по порядку оплаты отгруженной продукции. Как общая сумма продаж, так и средние продажи на покупателя гораздо выше по тем покупателям, которые оплачивают продукцию по мере реализации. Это можно объяснить тем, что такие покупатели не опасаются иметь более широкий ассортимент продукции и большие товарные запасы, так как им нужно оплачивать только реализованный товар.

ПРИМЕЧАНИЕ: все файлы проекта - <https://github.com/LeuKhaak/final-project-sql>