

Казакова Анастасия Сергеевна

**Задание выполнялось для sql oracle. Дата хранится в формате месяц/день/год. Также в дополнении к данному файлу прикреплен скрипт с этими тремя запросами!**

## Задача 1

### Задача 1

Table: Transactions

Client_id	Report_date	Txn_amount
123	2017.01.01	50000

Table: Rates

Report_date	Ccy_code	CCy_rate
2016.12.30	840	60,58
2017.01.09	840	61,01

Задание: Напишите sql запрос, который будет переводить сумму транзакций в usd (ccy\_code = 840) с учетом того, что в таблице rates данные только за рабочие дни. Транзакции, совершенные в выходные, пересчитываются по курсу последнего рабочего дня перед праздником/выходным.  
Результат: Клиент, дата, сумма операций в usd.

**Ответ:** В строчках 6-7 выводятся максимальная и минимальная дата, написанная в таблице с курсом (Rates) для того, чтобы восстановить недостающие (выходные, праздничные) дни. В блоке, состоящем в строчках 4-7 создается столбец all\_days, который содержит каждый день без пробелов. В 8 строчке происходит объединение полученной таблицы с изначальной таблицей, где код транзакции является 840 (usd). Объединение происходит слева поскольку нам нужно сохранить все даты и “выкинуть” ненужные коды (все, кроме 840). Присоединение таблиц происходит как раз по полям, отвечающих за дни, что можно заметить в строчке 9. Для красоты и удобства в строчке 10 производится сортировка вывода по дате. В блоке со 2 по 10 строчку выводятся все дни и курс валюты, а также на место пустых значений дублируется курс с прошлого (рабочего) дня. Полученная таблица присоединяется к таблице с данными о транзакциях (Transactions). Происходит это по дате, потому что нам важен лишь курс валюты, а для всех клиентов он одинаковый. В итоге, в строчке 1 мы выводим окончательный результат. На выходе получаем таблицу, в которой для каждого клиента ежедневно написан перевод, но уже в необходимой валюте.

**Скриншот:**

```

1  select Client_id, REPORT_DATE, TXN_AMOUNT*CCY_RATE_WITH_WEEKEND as sum_of_operations
2  from (select all_days, NVL(ccy_rate, lag(ccy_rate,1) ignore nulls over (order by all_days)) as ccy_rate_with_weekend
3  from
4  (select distinct (max_date - level + 1) as all_days
5  from(
6  select max(REPORT_DATE) as max_date, min(REPORT_DATE) as min_date
7  from Rates) connect by level <=(max_date - min_date +1)) t1
8  left join (select * from Rates where ccy_code = 840) t2
9  on t1.all_days = t2.REPORT_DATE
10 order by all_days ) t3
11 join Transactions
12 on t3.all_days = Transactions.Report_date

```

### Скрипт, который используется (ответ) в виде текста:

```

select Client_id, REPORT_DATE, TXN_AMOUNT*CCY_RATE_WITH_WEEKEND as
sum_of_operations
from (select all_days, NVL(ccy_rate, lag(ccy_rate,1) ignore nulls over (order by all_days)) as
ccy_rate_with_weekend
from
(select distinct (max_date - level + 1) as all_days
from(
select max(REPORT_DATE) as max_date, min(REPORT_DATE) as min_date
from Rates) connect by level <=(max_date - min_date +1)) t1
left join (select * from Rates where ccy_code = 840) t2
on t1.all_days = t2.REPORT_DATE
order by all_days ) t3
join Transactions
on t3.all_days = Transactions.Report_date

```

### Мои игрушечные данные:

CLIENT_ID	REPORT_DATE	TXN_AMOUNT
1	05/02/2017	10000
1	05/12/2017	100
1	01/01/2017	5000
2	12/30/2016	600
2	02/02/2017	10000

REPORT_DATE	CCY_CODE	CCY_RATE
05/13/2017	840	-
12/30/2016	840	60.58
01/09/2017	840	61.01
04/02/2017	840	61.52
02/02/2017	840	60.1
02/02/2017	850	100.3

### Результат на моих игрушечных данных:

CLIENT_ID	REPORT_DATE	SUM_OF_OPERATIONS
1	05/02/2017	615200
1	05/12/2017	6152
1	01/01/2017	302900
2	12/30/2016	36348
2	02/02/2017	601000

## Задача 2.1

### Задача 2

Table: VSP\_oper\_data

Client_id	Report_date	VSP_Number	Txn_type	Txn_amount
1233455	2017.05.02	1234/0123	debit	10000

В таблице VSP\_oper\_data txn\_type принимает значения debit, credit

Задание: напишите sql запрос, который для каждого клиента выводит сумму debit, credit операций и последнее посещенное VSP по месяцам. Результат представьте в виде:

Client_id	Report_date	Debit_amount	Credit_amount	Last_VSP

**Ответ:** Я поняла данное задание так, что для каждого клиента нужно вывести сколько в деньгах он выполнил debit операций за месяц, credit операций за месяц, а также последнее посещенное VSP за месяц. Причем под месяцем я подразумеваю пару месяц-год (январь 2017 и январь 2018 года это разные месяцы). В блоках 3-5 и 7-9 происходит одно и то же действие, только для двух разных результатов: блок 3-5 для получения суммы в деньгах (если Txn\_amount это деньги) для операции debit, а 7-9 для операции credit. Выводится id клиента, дата (в формате месяц.год, поскольку я считаю, что так удобнее, чем видеть одну и ту же информацию, если мы добавим день), а также считается сумма переводов/оплаты с помощью оконных функций (причем параметр Txn\_type из partition by можно убрать, поскольку он вынесен в отдельное условие where (строка 5 и 9)). Две полученные таблицы необходимо объединить full join, чтобы не терять людей, которые платили только по credit или только по debit, склеивание происходит по id клиента и дате соответственно. Как видно в строке 1, все пустые значения, которые образуются при соединении таблиц мы заменяем. Если id или даты нет в таблице с credit, то берем ее из debit, если значение суммы где-то отсутствует, то происходит замена на 0. Далее осталось выписать последнее посещенное за месяц VSP и также присоединить к полученной таблице. Для этого в строках 14-15 выписывается необходимая для соединения таблиц информация (id клиента и дата), а также создается две новые колонки, которые содержат в себе: пару месяц-год (из-за поставленной задачи) и пару (день-месяц) чтобы пойти последний день, в котором было посещение в месяце. Как раз для нахождения последнего дня (он же не обязательно будет последним календарным

днем месяца, например человек последний раз пришел 20 числа, а не 30 или 31) используется строка 13, которая выводит “заново соединенную из двух столбцов” дату (с помощью конкатенации последнего дня посещения и поскольку, если конкатенировать напрямую, то месяц повториться дважды, используется регулярное выражение для удаления повтора), чтобы в будущем ее сравнить с исходной датой для поиска номера подразделения VSP в строке 16. Таким образом, через данное сравнение блок 12-16 выдает id клиента, дату (также в виде месяц-год) и номер VSP. Осталось только все соединить так, чтобы или таблица credit или таблица debit совпадала и по id клиента и по дате. Для красоты в конце я сортирую по id.

## Скриншот:

```

1 select distinct NVL(credit_date.Client_id, debit_date.Client_id) as Client_id, NVL(credit_date.Report_date, debit_date.Report_date) as Report_date, NVL(CREDIT_AMOUNT, 0) as CREDIT_AMOUNT, NVL(DEBIT_AMOUNT, 0) as DEBIT_AMOUNT, Last_VSP
2 from
3 (select Client_id, sum(Txn_amount) over(partition by Txn_type, to_char(Report_date,'MM.YYYY'), Client_id) as Debit_amount, to_char(Report_date,'MM.YYYY') as Report_date
4 from VSP_oper_data
5 where Txn_type = 'debit') debit_date
6 full join
7 (select Client_id, sum(Txn_amount) over(partition by Txn_type, to_char(Report_date,'MM.YYYY'), Client_id) as Credit_amount, to_char(Report_date,'MM.YYYY') as Report_date
8 from VSP_oper_data
9 where Txn_type = 'credit') credit_date
10 on credit_date.Report_date = debit_date.Report_date and credit_date.Client_id = debit_date.Client_id
11 inner join
12 (select distinct Client_id, to_char(Report_date,'MM.YYYY') as Report_date, VSP_NUMBER as Last_VSP
13 from ( Select Client_id, VSP_Number, Report_date, month_year, day_month, to_date(CONCAT(max(day_month) over (partition by Client_id, month_year), REGEXP_REPLACE(month_year, '^d\d', '')), 'DD.MM.YYYY') as max_day_in_mothh
14 from ( Select Client_id, VSP_Number, Report_date, to_char(Report_date,'MM.YYYY') as month_year, to_char(Report_date,'DD.MM') as day_month
15 from VSP_oper_data))
16 where REPORT_DATE = MAX_DAY_IN_MOHTH) last_vsp_date
17 on credit_date.Report_date = last_vsp_date.Report_date and credit_date.Client_id=last_vsp_date.Client_id
18 or debit_date.Report_date = last_vsp_date.Report_date and debit_date.Client_id=last_vsp_date.Client_id
19 order by Client_id
20

```

## Скрипт, который используется (ответ) в виде текста:

```

select distinct NVL(credit_date.Client_id, debit_date.Client_id) as Client_id,
NVL(credit_date.Report_date, debit_date.Report_date) as Report_date,
NVL(CREDIT_AMOUNT, 0) as CREDIT_AMOUNT, NVL(DEBIT_AMOUNT, 0) as
DEBIT_AMOUNT, Last_VSP
from
(select Client_id, sum(Txn_amount) over(partition by Txn_type,
to_char(Report_date,'MM.YYYY'), Client_id) as Debit_amount,
to_char(Report_date,'MM.YYYY') as Report_date
from VSP_oper_data
where Txn_type = 'debit') debit_date
full join
(select Client_id, sum(Txn_amount) over(partition by Txn_type,
to_char(Report_date,'MM.YYYY'), Client_id) as Credit_amount,
to_char(Report_date,'MM.YYYY') as Report_date
from VSP_oper_data
where Txn_type = 'credit') credit_date
on credit_date.Report_date = debit_date.Report_date and credit_date.Client_id =
debit_date.Client_id
inner join
(select distinct Client_id, to_char(Report_date,'MM.YYYY') as Report_date, VSP_NUMBER
as Last_VSP
from( Select Client_id, VSP_Number, Report_date,month_year, day_month,
to_date(CONCAT(max(day_month) over (partition by Client_id, month_year),
REGEXP_REPLACE(month_year, '^d\d', '')), 'DD.MM.YYYY') as max_day_in_mothh
from ( Select Client_id, VSP_Number, Report_date,to_char(Report_date,'MM.YYYY') as
month_year, to_char(Report_date,'DD.MM') as day_month
from VSP_oper_data))
where REPORT_DATE = MAX_DAY_IN_MOHTH) last_vsp_date

```

on credit\_date.Report\_date = last\_vsp\_date.Report\_date and  
 credit\_date.Client\_id=last\_vsp\_date.Client\_id  
 or debit\_date.Report\_date = last\_vsp\_date.Report\_date and  
 debit\_date.Client\_id=last\_vsp\_date.Client\_id  
 order by Client\_id

### Мои игрушечные данные:

CLIENT_ID	REPORT_DATE	VSP_NUMBER	TXN_TYPE	TXN_AMOUNT
1	05/02/2017	1234/0123	debit	10000
2	03/02/2017	1234/1234	debit	5000
1	04/02/2017	1234/2345	credit	1000
2	03/02/2017	1234/3456	credit	2000
1	03/02/2017	1234/0123	credit	10000
1	03/12/2017	0000/2112	debit	3456
3	05/05/2018	0123/4321	debit	100
1	03/02/2017	1234/0123	debit	10000
2	05/05/2017	1234/9876	credit	20000

9 rows returned in 0.02 seconds [Download](#)

### Результат на моих игрушечных данных:

CLIENT_ID	REPORT_DATE	CREDIT_AMOUNT	DEBIT_AMOUNT	LAST_VSP
1	03.2017	10000	13456	0000/2112
1	04.2017	1000	0	1234/2345
1	05.2017	0	10000	1234/0123
2	03.2017	2000	5000	1234/1234
2	03.2017	2000	5000	1234/3456
2	05.2017	20000	0	1234/9876
3	05.2018	0	100	0123/4321

### Задача 2.2

## Задача 2

Table: VSP\_oper\_data

Client_id	Report_date	VSP_Number	Txn_type	Txn_amount
1233455	2017.05.02	1234/0123	debit	10000

В таблице VSP\_oper\_data txn\_type принимает значения debit, credit

Задание: напишите sql запрос, который для каждого клиента выведет долю debit операций клиента к debit операциям всех клиентов по месяцам. Результат в виде таблицы:

Client_id	Report_date	Ratio

**Ответ:** Как я поняла данное задание, “по месяцам” имеется в виду по паре месяц-год (то есть январь 2017 и январь 2018 являются разным выводом). В блоке строчек 2-6 выводится табличка, которая состоит из id клиента, суммы, которую клиент перевел/оплатил в каком-то месяце в сочетании с годом (Debit\_each\_client, данный параметр легко находится через оконные функции), аналогично можно найти сумму, которую перевели/оплатили все клиенты в данном месяце (Debit\_all\_clients), а также не забываем дату (в данном случае я оставила от даты пару месяц-год, потому что считаю, что так красивее и дни не нужны). Все данные, перечисленные выше считаются только для debit, что показано в строчке 6. Теперь, из полученных данных осталось вывести необходимые по заданию столбцы: id клиента, дату (в своем формате, надеюсь это не страшно), а также отношение денег клиента ко всем деньгам за месяц.

## Скриншот:

```
1 select Client_id, Report_date, Debit_each_client/Debit_all_clients as Ratio
2 from (select distinct Client_id,
3 sum(Txn_amount) over(partition by Client_id, to_char(Report_date,'MM.YYYY')) as Debit_each_client, to_char(Report_date,'MM.YYYY') as Report_date,
4 sum(Txn_amount) over(partition by to_char(Report_date,'MM.YYYY')) as Debit_all_clients
5 from VSP_oper_data
6 where Txn_type = 'debit')
7
```

## Скрипт, который используется (ответ) в виде текста:

```
select Client_id, Report_date, Debit_each_client/Debit_all_clients as Ratio
from (Select distinct Client_id,
sum(Txn_amount) over(partition by Client_id, to_char(Report_date,'MM.YYYY')) as
Debit_each_client, to_char(Report_date,'MM.YYYY') as Report_date,
sum(Txn_amount) over(partition by to_char(Report_date,'MM.YYYY')) as Debit_all_clients
from VSP_oper_data
where Txn_type = 'debit')
```

## Мои игрушечные данные:

CLIENT_ID	REPORT_DATE	VSP_NUMBER	TXN_TYPE	TXN_AMOUNT
1	05/02/2017	1234/0123	debit	10000
2	03/02/2017	1234/1234	debit	5000
1	04/02/2017	1234/2345	credit	1000
2	03/02/2017	1234/3456	credit	2000
1	03/02/2017	1234/0123	credit	10000
1	03/12/2017	0000/2112	debit	3456
3	05/05/2018	0123/4321	debit	100
1	03/02/2017	1234/0123	debit	10000
2	05/05/2017	1234/9876	credit	20000
9 rows returned in 0.02 seconds <a href="#">Download</a>				

Результат на моих игрушечных данных:

CLIENT_ID	REPORT_DATE	RATIO
1	03.2017	.729085392284351972258344169917641959254
3	05.2018	1
2	03.2017	.270914607715648027741655830082358040746
1	05.2017	1