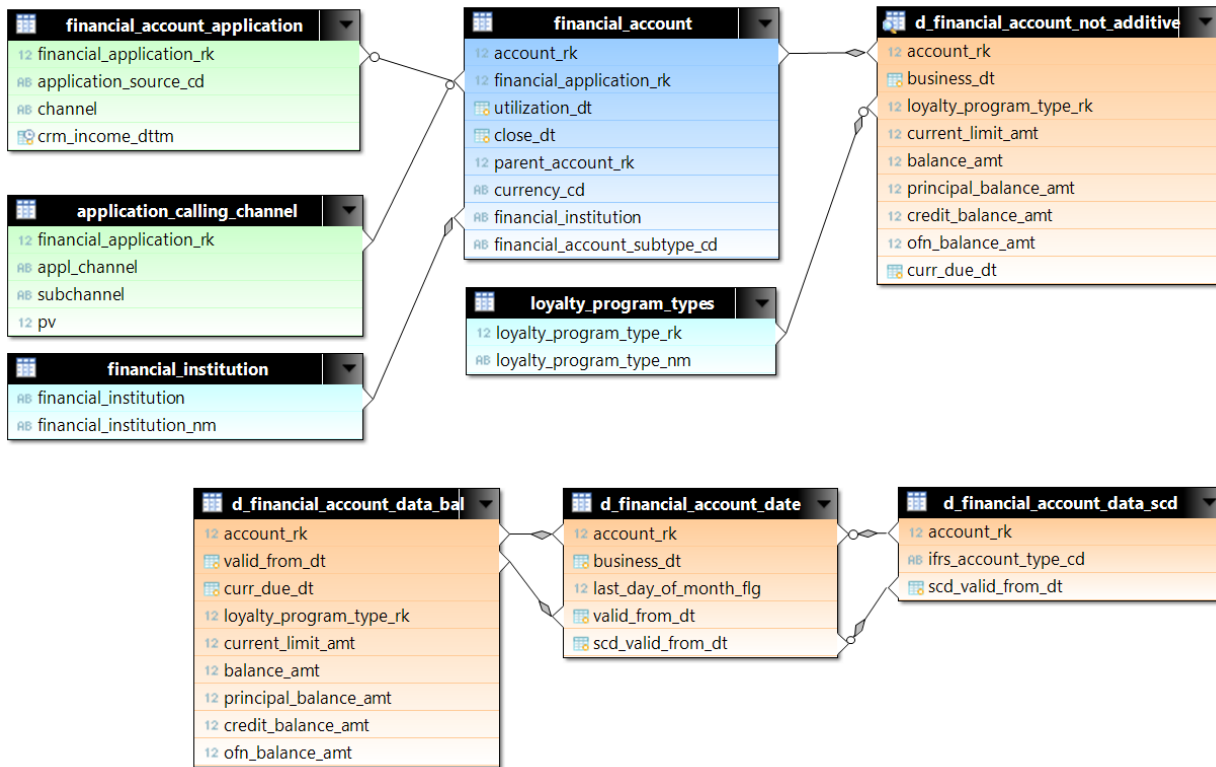


Описание и текст тестовых запросов, использованных в тестировании Exasol, SAP HANA, Greenplum, MemSQL, Impala, ClickHouse в банке «Тинькофф»



Для тестирования были подобраны запросы с `d_financial_account_not_additive`.

`d_financial_account_not_additive` — это представление (view) с данными по каждому счету на каждый день. View сделано на основе трех таблиц с целью оптимизации места на диске, и, соответственно, чтения с диска. Для тестирования была выбрана часть данных по первому миллиону счетов с начала 2015 года. Это чуть больше 522 миллионов строк. К `not_additive` мы присоединяем данные по счетам (`financial_account`) и по заявкам (`financial_account_application` и `application_calling_channel`). В Greenplum (для примера) для таблиц заданы ключи распределения по сегментам: для счетов это `account_rk`, для заявок – `financial_application_rk`. В запросах `join`-ы между основными таблицами происходят по равенству, поэтому мы можем ожидать `hash join`, без `nested loop`, когда нужно сравнить построчно большое количество строк из разных таблиц.

Общий объём данных составил около 200 Гб в несжатом виде (рассчитываем, что весь этот объём с небольшим запасом влезает в память).

Число строк в используемых таблицах:

Таблица	Число строк
<code>d_financial_account_date</code>	522726636

d_financial_account_data_bal	229255701
financial_account_application	52118559
application_calling_channel	28158924
d_financial_account_data_scd	3494716
financial_account	2930425
currency_rates	3948
dds_calendar_date	731
loyalty_program_types	35
financial_institution	5

Запрос N1

```
SELECT date_trunc('year', d_financial_account_not_additive.business_dt) || '-' ||
date_trunc('month',d_financial_account_not_additive.business_dt) as yymm,
d_financial_account_not_additive.business_dt,
financial_account.financial_account_subtype_cd, case when
d_financial_account_not_additive.ofn_balance_amt <0 then 1 else 0 end,
loyalty_program_by_day.loyalty_program_type_nm, financial_account.currency_cd,
sum(d_financial_account_not_additive.balance_amt*Table__14.rate),
sum(d_financial_account_not_additive.balance_amt) FROM
prod_emart.loyalty_program_types loyalty_program_by_day RIGHT OUTER JOIN
prod_emart.d_financial_account_not_additive d_financial_account_not_additive ON
(d_financial_account_not_additive.loyalty_program_type_rk=loyalty_program_by_day.l
oyalty_program_type_rk AND loyalty_program_by_day.valid_to_dttm > now()) INNER
JOIN prod_emart.financial_account financial_account ON
(financial_account.account_rk=d_financial_account_not_additive.account_rk) INNER
JOIN ( SELECT r.currency_from_cd, r.currency_to_cd, r.rate, r.rate_dt FROM
prod_emart.currency_rates r WHERE ( r.currency_to_cd='RUR' ) union all SELECT
'RUB', 'RUR', 1, d.calendar_dt FROM prod_emart.dds_calendar_date d ) Table__14 ON
(Table__14.currency_from_cd=financial_account.currency_cd) WHERE (
Table__14.rate_dt=d_financial_account_not_additive.business_dt ) AND (
d_financial_account_not_additive.business_dt >= to_date(( 2016 - 2)::character
varying || '-01-01', 'YYYY-MM-DD') AND d_financial_account_not_additive.business_dt
<= (current_date-1) AND financial_account.financial_account_subtype_cd IN (
'DEP', 'SAV', 'SVN', 'LEG', 'CUR' ) ) GROUP BY date_trunc('year',
d_financial_account_not_additive.business_dt) || '-' ||
date_trunc('month',d_financial_account_not_additive.business_dt),
d_financial_account_not_additive.business_dt,
financial_account.financial_account_subtype_cd, case when
d_financial_account_not_additive.ofn_balance_amt <0 then 1 else 0 end,
loyalty_program_by_day.loyalty_program_type_nm, financial_account.currency_cd
```

Запрос N2

```
select count(*) from (SELECT date_trunc('year',
d_financial_account_not_additive.business_dt) || '-' ||
date_trunc('month',d_financial_account_not_additive.business_dt) as yymm,
d_financial_account_not_additive.business_dt,
financial_account.financial_account_subtype_cd,
loyalty_program_by_day.loyalty_program_type_nm,
application_calling_channel.appl_channel, case when (
financial_account_application.application_product_type )='010222' then 'Y' else
'N' end , case when ( financial_account_application.application_product_type
)='020202' then 'Y' else 'N' end, case when financial_account.parent_account_rk is
null then 'N' else 'Y' end,
prod_emart.financial_institution.financial_institution_nm,
sum(d_financial_account_not_additive.principal_balance_amt),
sum(d_financial_account_not_additive.interest_balance_amt),
sum(d_financial_account_not_additive.f2g_balance_amt),
sum(d_financial_account_not_additive.f2n_balance_amt),
sum(d_financial_account_not_additive.overdue_fee_balance_amt),
sum(d_financial_account_not_additive.pastdue_gvt_balance_amt),
sum(d_financial_account_not_additive.annual_fee_balance_amt),
sum(d_financial_account_not_additive.sim_kke_balance_amt) FROM
prod_emart.loyalty_program_types loyalty_program_by_day RIGHT OUTER JOIN
prod_emart.d_financial_account_not_additive d_financial_account_not_additive ON
(d_financial_account_not_additive.loyalty_program_type_rk=loyalty_program_by_day.l
oyalty_program_type_rk AND loyalty_program_by_day.valid_to_dttm > now()) INNER
JOIN prod_emart.financial_account financial_account ON
(financial_account.account_rk=d_financial_account_not_additive.account_rk) LEFT
OUTER JOIN prod_emart.financial_account_application ON
financial_account.financial_application_rk=financial_account_application.financial
_application_rk LEFT OUTER JOIN prod_emart.application_calling_channel on
financial_account.financial_application_rk=application_calling_channel.financial_a
pplication_rk LEFT OUTER JOIN prod_emart.financial_account parent_account ON
(financial_account.parent_account_rk=parent_account.account_rk) LEFT OUTER JOIN
prod_emart.financial_institution ON
(financial_account.financial_institution=financial_institution.financial_instituti
on) WHERE ( d_financial_account_not_additive.business_dt >= to_date('2014-01-01',
'YYYY-MM-DD') AND d_financial_account_not_additive.business_dt <= (current_date-1)
AND ( financial_account.financial_account_subtype_cd IN (
'CCR','INS','CLN','VKR','EIC' ) OR (
financial_account.financial_account_subtype_cd IN ( 'PHX' ) AND (
parent_account.financial_account_subtype_cd Is Null OR
parent_account.financial_account_subtype_cd NOT IN ( 'IFS' ) ) ) ) ) GROUP BY
date_trunc('year', d_financial_account_not_additive.business_dt) || '-' ||
date_trunc('month',d_financial_account_not_additive.business_dt),
d_financial_account_not_additive.business_dt,
financial_account.financial_account_subtype_cd,
loyalty_program_by_day.loyalty_program_type_nm,
application_calling_channel.appl_channel, case when (
financial_account_application.application_product_type )='010222' then 'Y' else
'N' end , case when ( financial_account_application.application_product_type
)='020202' then 'Y' else 'N' end, case when financial_account.parent_account_rk is
null then 'N' else 'Y' end, financial_institution.financial_institution_nm) sq
```

Запрос N3

```
SELECT date_trunc('year', d_financial_account_not_additive.business_dt) || '-' ||
date_trunc('month',d_financial_account_not_additive.business_dt) as yymm,
d_financial_account_not_additive.business_dt,
financial_account.financial_account_subtype_cd, case when (
prod_emart.financial_account_application.application_product_type )='010222' then
'Y' else 'N' end , d_financial_account_not_additive.risk_status_cd, case when
financial_account.utilization_dt<=d_financial_account_not_additive.business_dt
then 1 else 0 end, case when ( d_financial_account_not_additive.current_limit_amt)
> 0 then 1 else 0 end, prod_emart.financial_institution.financial_institution_nm,
--sum(d_financial_account_not_additive.credit_balance_amt),
sum(d_financial_account_not_additive.principal_balance_amt),
sum(d_financial_account_not_additive.current_limit_amt),
count(d_financial_account_not_additive.account_rk), sum(case when
d_financial_account_not_additive.current_limit_amt > 0 then
d_financial_account_not_additive.principal_balance_amt /
d_financial_account_not_additive.current_limit_amt end) FROM
prod_emart.d_financial_account_not_additive INNER JOIN
prod_emart.financial_account financial_account ON
(financial_account.account_rk=d_financial_account_not_additive.account_rk) LEFT
OUTER JOIN prod_emart.financial_account_application on
financial_account.financial_application_rk=prod_emart.financial_account_applicatio
n.financial_application_rk LEFT OUTER JOIN prod_emart.financial_institution ON
(financial_account.financial_institution=prod_emart.financial_institution.financia
l_institution) WHERE ( d_financial_account_not_additive.business_dt >= to_date((
2016 - 2)::character varying || '-01-01', 'YYYY-MM-DD') AND
d_financial_account_not_additive.business_dt <= (current_date-1) AND
financial_account.financial_account_subtype_cd IN ( 'CCR','CLN','VKR','INS','EIC'
) AND case when
financial_account.close_dt<=d_financial_account_not_additive.business_dt then 1
else 0 end IN ( 0 ) ) GROUP BY date_trunc('year',
d_financial_account_not_additive.business_dt) || '-' ||
date_trunc('month',d_financial_account_not_additive.business_dt),
d_financial_account_not_additive.business_dt,
financial_account.financial_account_subtype_cd, case when (
prod_emart.financial_account_application.application_product_type )='010222' then
'Y' else 'N' end , d_financial_account_not_additive.risk_status_cd, case when
financial_account.utilization_dt<=d_financial_account_not_additive.business_dt
then 1 else 0 end, case when ( d_financial_account_not_additive.current_limit_amt)
> 0 then 1 else 0 end, prod_emart.financial_institution.financial_institution_nm
```

Запрос T1

```
SELECT count(*) FROM ( SELECT * FROM prod_emart.d_financial_account_data_bal ) ALL  
INNER JOIN ( SELECT * FROM prod_emart.d_financial_account_date ) USING account_rk,  
valid_from_dt
```

Запрос T2

```
SELECT count(*) FROM prod_emart.d_financial_account_data_bal a JOIN  
prod_emart.d_financial_account_date b ON a.account_rk = b.account_rk AND  
a.valid_from_dt = b.valid_from_dt LEFT JOIN  
prod_emart.d_financial_account_data_scd sc ON a.account_rk = sc.account_rk AND  
b.scd_valid_from_dt = sc.scd_valid_from_dt;
```

Запрос D1 (Декартово произведение одной колонки)

```
-- В таблице - 291 157 926 408 строк select count(*) from (SELECT * FROM  
WRK.D_FINANCIAL_ACCOUNT_DATE) t1 INNER JOIN (SELECT * FROM  
WRK.D_FINANCIAL_ACCOUNT_DATE) t2 on t1.account_rk = t2.account_rk;
```

Запрос D2 (Декартово произведение нескольких колонок)

```
select count(*) ,sum(t1.last_day_of_month_flg - t2.last_day_of_month_flg) as  
sum_flg ,sum(t1.business_dt - t2.valid_from_dt) as blv2 ,sum(t1.valid_from_dt -  
coalesce(t2.scd_valid_from_dt,current_date)) as vls2  
,sum(coalesce(t1.scd_valid_from_dt,current_date) - t1.business_dt) as slb2 from  
prod_emart.D_FINANCIAL_ACCOUNT_DATE t1 INNER JOIN  
prod_emart.D_FINANCIAL_ACCOUNT_DATE t2 on t1.account_rk = t2.account_rk;
```

Как протестировать Exasol самостоятельно?

Убедиться в простоте настройки и производительности Exasol Вы можете на основе:

1. Бесплатной версии, [Exasol Community Edition](https://www.exasol.com/en/download/community-edition-download/). Это полнофункциональная версия аналитической базы данных, ограниченная 200 Гигабайтами сырых данных. Она идеально подходит для тестирования с последующим переходом на платную версию Exasol, по мере роста данных.
Скачать Exasol Community Edition: <https://www.exasol.com/en/download/community-edition-download/>
2. Триал облачной версии Exasol на собственном облаке ExaCloud, первые 30 дней бесплатно.
Получить доступ к ExaCloud: <https://www.exasol.com/en/download/>
3. Триал Exasol на облаке AWS, первые 30 дней бесплатно, 200\$ на аккаунт AWS для тестирования.
Получить доступ к Exasol на AWS: <https://www.exasol.com/en/insights/cloud/aws-30-day-free-trial/>
4. Тестовая лицензия Exasol с любыми параметрами. Если Вам нужно протестировать Exasol на своем кластере, с более чем 200 Гб сырых данных, обратитесь за тестовой лицензией в Консультационную Группу ATK, официального партнера Exasol в России.
Связаться с нами: <http://www.atkcg.ru/>

Консультационная Группа ATK

Телефон: +7 (495) 937 16 50

E-mail: consult@atkcg.ru

Адрес: 121087, Россия, Москва, ул. Баркляя, д. 6, стр. 3, БЦ «Барклай-Парк», оф. 402