

```

-- ## Требования к выполнению

-- Задания должны быть выполнены на языке запросов SQL и выполняться без ошибок на реальной базе данных PostgreSQL .

-- Для выполнения заданий необходимо установить локальный сервер базы данных PostgreSQL версии 12 или выше. Приложе

-- Решения заданий можно приложить в виде отдельных файлов SQL или вставить в README.md в блоки кода.

-- ### Задание 1

-- Подготовить витрину с данными по контрактам.

-- Данные для витрины хранятся в файлах CSV в директории `data`, которые необходимо загрузить в базу данных.

-- Описание файлов исходных данных для формирования витрины:

-- - test_data_contract.csv - общая информация о контрактах.
--   * customer_id - ID клиента.
--   * contract_id - ID контракта.
--   * contract_code - код контракта в CRM.
--   * marker_delete - пометка на удаление.
--   * issue_dt - дата заключения контракта.
--   * subdivision_id - ID подразделения, заключившего контракт.
--   * renewal_contract_id - ID контракта, который стал причиной переоформления.
-- - test_data_contract_conditions.csv - информация об изменении условий контракта, в том числе перенос срока и про
--   * condition_id - ID изменения условий контракта.
--   * condition_dt - дата изменения условия.
--   * contract_id - ID контракта.
--   * conducted - признак проведения изменения в CRM.
--   * marker_delete - пометка на удаление.
--   * operation_type - тип операции.
--   * condition_type - тип изменения условий.
--   * condition_start_dt - дата начала действия изменения условий. Относится к сроку действия контракта или продле
--   * condition_end_dt - дата окончания действия изменения условий. Относится к сроку действия контракта или продл
--   * days - срок действия изменения условий в днях.
-- - test_data_contract_conditions_payment_plan.csv - план-график выплат по контракту. Генерируется для каждого изм
--   * condition_id - ID изменения условия контракта.
--   * payment_dt - дата платежа.
--   * loan_amount - сумма платежа.
-- - test_data_contract_status.csv - информация об изменении статуса контракта.
--   * contract_id - ID контракта.
--   * status_dt - дата и время статуса контракта.
--   * status_type - вид статуса контракта.

-- Необходимо получить итоговую таблицу (витрину) со следующим набором столбцов:

-- - contract_id - ID контракта.
-- - contract_code - Код контракта.
-- - customer_id - ID клиента.
-- - condition_id - ID документа о заключении контракта.
-- - subdivision_id - ID подразделения, заключившего контракт.
-- - contract_serial_number - Порядковый номер контракта у клиента.
-- - contract_renewal_serial_number - Порядковый номер контракта у клиента без учёта переоформлений. Если контракт
-- - is_renewal - Является ли данный контракт переоформлением (наличие ID в поле renewal_contract_id).
-- - is_installment - Является ли данный контракт долгосрочным (наличие нескольких платежей в плане погашений).
-- - prolong_count - Количество продлений. Контракт может быть продлён неограниченное количество раз (см. test_data
-- - first_issue_dt - Дата первого контракта у клиента.
-- - issue_dt - Дата выдачи займа.
-- - plan_dt - Дата планового погашения займа.
-- - last_plan_dt - Дата планового погашения займа с учётом продлений.
-- - close_dt - Дата фактического погашения займа (дата закрытия). Учитывается только последний статус по контракту
-- - loan_amount - Сумма займа. Суммируются все платежи по графику.
-- - total_loan_amount - Сумма всех предыдущих займов.
-- - min_loan_amount - Минимальная сумма предыдущих займов.
-- - max_loan_amount - Максимальная сумма предыдущих займов.
-- - loan_term - Срок займа в днях.
-- - min_loan_term - Минимальный срок предыдущих займов.
-- - max_loan_term - Максимальный срок предыдущих займов.
-- - is_closed - Является ли контракт закрытым на текущий момент. Закрытым считаем контракт со следующими статусами

-- Условия для сборки итоговой таблицы (витрины):

-- - первичный ключ итоговой таблицы: contract_id.
-- - индексы в итоговой таблице: contract_id, customer_id, condition_id.
-- - типы данных итоговой таблицы необходимо определить на основе исходных данных.
-- - в витрину не должны попасть данные помеченные на удаление (поле `mark_delete`) и непроведённые документы (поле
-- - для извлечения условий договора используются только операции «ЗаключениеДоговора» (см. test_data_contract_cond.

-- Решение:

```

```

--создадим временную таблицу для расчета contract_serial_number
DROP TABLE tmp_contract;
CREATE TABLE tmp_contract AS (
SELECT customer_id, contract_id, contract_code, marker_delete, issue_dt, subdivision_id, renewal_contract_id,
DENSE_RANK() OVER (PARTITION BY customer_id order by issue_dt, contract_id) as contract_serial_number
FROM public.contract_bd
where marker_delete<>TRUE
order by customer_id, issue_dt
);

--создадим таблицу v_contract
DROP TABLE IF EXISTS public.v_contract;

CREATE TABLE IF NOT EXISTS public.v_contract
(
contract_id character varying(36) COLLATE pg_catalog."default",
contract_code character varying(11) COLLATE pg_catalog."default",
customer_id character varying(36) COLLATE pg_catalog."default",
condition_id character varying(36) COLLATE pg_catalog."default",
subdivision_id character varying(36) COLLATE pg_catalog."default",
contract_serial_number integer,
contract_renewal_serial_number integer,
is_renewal integer,
is_installment integer,
prolong_count integer,
first_issue_dt date,
issue_dt date,
plan_dt date,
last_plan_dt date,
close_dt date,
loan_amount real,
total_loan_amount real,
min_loan_amount real,
max_loan_amount real,
loan_term integer,
min_loan_term integer,
max_loan_term integer,
is_closed character varying(6)
)
TABLESPACE pg_default;

ALTER TABLE tmp_contract ADD COLUMN ren_cont_id_dupl text;
ALTER TABLE tmp_contract ADD COLUMN contract_renewal_serial_number bigint;
Update tmp_contract * set ren_cont_id_dupl = renewal_contract_id;
Update tmp_contract * set contract_renewal_serial_number = contract_serial_number;

CREATE OR REPLACE FUNCTION renumber_renewal() RETURNS integer AS
$BODY$
DECLARE rfound INT;
BEGIN
RAISE NOTICE 'Запускаем нумерацию контрактов исключая переоформления';
LOOP
select COUNT(*) INTO STRICT rfound
from tmp_contract as l
JOIN
(select contract_id from tmp_contract) as r
ON l.ren_cont_id_dupl = r.contract_id;

RAISE NOTICE 'Осталось [%]', rfound;

IF rfound = 0 THEN
EXIT; -- ВЫХОД ИЗ ЦИКЛА
END IF;

UPDATE tmp_contract
SET contract_renewal_serial_number = new_n.new_num, ren_cont_id_dupl = new_n.new_renewal_contract_id
FROM
(select distinct
l.contract_id,
l.ren_cont_id_dupl,
r.new_renewal_contract_id,
r.contract_serial_number as new_num
from tmp_contract as l
JOIN
(select
contract_id,
renewal_contract_id as new_renewal_contract_id,
contract_serial_number from tmp_contract) as r
ON l.ren_cont_id_dupl = r.contract_id
)
as new_n

```

```

where tmp_contract.contract_id = new_n.contract_id and tmp_contract.ren_cont_id_dupl IS NOT NULL;

END LOOP;
RAISE NOTICE 'Нумерация завершена';
RETURN 0;

END;
$BODY$
LANGUAGE plpgsql;

--запускаем процедуру нумерации
SELECT * FROM renumber_renewal();

ALTER TABLE tmp_contract drop column ren_cont_id_dupl;

--ранки со старыми парент ид перенумеруем
update tmp_contract * set contract_renewal_serial_number=new_sn_r.dense_rank
FROM
(select * ,
dense_rank() over (partition by customer_id order by contract_renewal_serial_number)
from tmp_contract as t
order by issue_dt)
as new_sn_r
where tmp_contract.contract_id = new_sn_r.contract_id;
;

--собираем все данные для витрины
with

clear_contract_conditions as
(select *
from contract_conditions_bd
where marker_delete<>True and conducted <>False and operation_type = 'ЗаключениеДоговора' ),

is_instament as
(select cc.contract_id, cc.condition_id, cc.condition_dt,
case when cnt>=2 then 1 else 0 end as is_instament
from (select condition_id, count(payment_dt) as cnt
from contract_conditions_payment_plan_bd
group by condition_id
order by condition_id) as ccpp_cnt
join clear_contract_conditions as cc
on ccpp_cnt.condition_id=cc.condition_id
order by is_instament),

prolong as
(select customer_id, contract_renewal_serial_number, count(condition_type) as prolong_count
from (select customer_id, nc.contract_id, issue_dt, nc.renewal_contract_id, operation_type,
condition_type, nc.contract_renewal_serial_number
from tmp_contract as nc left join contract_conditions_bd as cc on nc.contract_id=cc.contract_id
order by issue_dt) as t
group by customer_id, contract_renewal_serial_number, condition_type
having condition_type = 'Продление'),

first_issue_dt as
(select customer_id,
min(issue_dt) as first_issue_dt
from tmp_contract
group by customer_id),

plan_dt_loan_amount as
(select condition_id, min(payment_dt), max(payment_dt) as plan_dt, sum(loan_amount) as loan_amount
from contract_conditions_payment_plan_bd as ccpp
group by condition_id),

last_plan_dt as
(select tc.contract_id, tc.contract_renewal_serial_number, max(ccpp.payment_dt) as last_plan_dt
from tmp_contract as tc left join clear_contract_conditions as cc on tc.contract_id = cc.contract_id
left join contract_conditions_payment_plan_bd as ccpp on cc.condition_id = ccpp.condition_id
group by tc.contract_id, tc.contract_renewal_serial_number
order by tc.contract_id),

close_dt as
(select cc.condition_id, max(cc.condition_dt) as close_dt
from clear_contract_conditions as cc
left join contract_conditions_payment_plan_bd as ccpp
on cc.condition_id = ccpp.condition_id
group by cc.condition_id),

total_loan_amount as
(select
nc.customer_id,
sum(pd.loan_amount) as total_loan_amount,

```

```

max(pd.loan_amount) as max_loan_amount, min (loan_amount) as min_loan_amount,
max(cc.condition_end_dt- cc.condition_start_dt ) as min_loan_term,
min(cc.condition_end_dt- cc.condition_start_dt ) as max_loan_term
from tmp_contract as nc
join clear_contract_conditions as cc on nc.contract_id = cc.contract_id
left join plan_dt_loan_amount as pd on cc.condition_id = pd.condition_id
group by nc.customer_id
order by nc.customer_id
),

loan_term as
(select
cc.contract_id, cc.condition_id, loan_amount, cc.operation_type, cc.condition_start_dt,
cc.condition_end_dt, cc.condition_end_dt- cc.condition_start_dt as loan_term
from clear_contract_conditions as cc
left join plan_dt_loan_amount as pd on cc.condition_id = pd.condition_id
where operation_type = 'ЗаклучениеДоговора'
order by contract_id),

is_closed as
(select contract_id, is_closed
from (select contract_id,
case when status_type in ('Закрйт', 'Договор закрыт с переплатой', 'Переоформлен')
then 'Закрйт'
end as is_closed
from contract_status_bd) as close
where is_closed='Закрйт'),

showcase as
(select tc.contract_id, tc.contract_code, tc.customer_id, cc.condition_id, tc.subdivision_id,
tc.contract_serial_number, tc.contract_renewal_serial_number,
CASE WHEN renewal_contract_id is not null then 1 else 0 END as is_renewal,
ii.is_instament,
p.prolong_count, f.first_issue_dt, tc.issue_dt, pd.plan_dt, lpd.last_plan_dt, date_trunc('day',cd.close_dt) as
pd.loan_amount, tla.total_loan_amount, tla.min_loan_amount, tla.max_loan_amount,
lt.loan_term, tla.min_loan_term, tla.max_loan_term, ic.is_closed
from tmp_contract as tc
join clear_contract_conditions as cc on tc.contract_id = cc.contract_id
join is_instament as ii on (tc.contract_id=ii.contract_id and cc.condition_id=ii.condition_id)
left join prolong as p on (tc.customer_id=p.customer_id and tc.contract_renewal_serial_number = p.contract_renewal_
join first_issue_dt as f on tc.customer_id=f.customer_id
left join plan_dt_loan_amount as pd on (cc.condition_id = pd.condition_id)
left join last_plan_dt as lpd on (tc.contract_id = lpd.contract_id)
join close_dt as cd on (cc.condition_id = cd.condition_id)
left join total_loan_amount as tla on (tc.customer_id = tla.customer_id)
left join loan_term as lt on (tc.contract_id = lt.contract_id and cc.condition_id = lt.condition_id)
left join is_closed as ic on (tc.contract_id = ic.contract_id)
order by issue_dt)

--вставляем собранный итог в таблицу v_contract
INSERT INTO v_contract (contract_id, contract_code, customer_id,
condition_id, subdivision_id, contract_serial_number,
contract_renewal_serial_number, is_renewal, is_installment,
prolong_count, first_issue_dt, issue_dt, plan_dt, last_plan_dt,
close_dt, loan_amount, total_loan_amount, min_loan_amount,
max_loan_amount, loan_term, min_loan_term, max_loan_term, is_closed)

select distinct * from showcase

-- создаем индексы
CREATE UNIQUE INDEX on v_contract (contract_id);
CREATE INDEX on v_contract (contract_id, customer_id, condition_id);

```