Предметная область: Транзакции по банковским картам

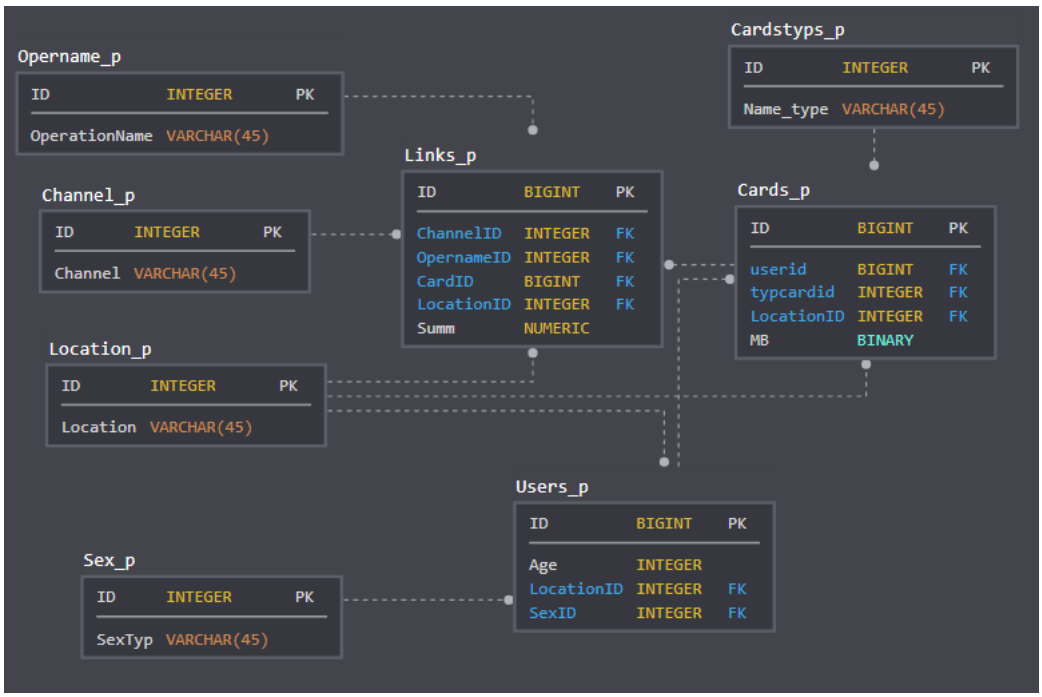
Структура базы:

1. Тип карт (Cardstyps_p)
    1.1. Описание:
        1.1.1. Таблица «Cardstyps_p» содержит информацию о возможном типе карт. Карта может быть дебетовой и кредитной.
    1.2. Поля:
        1.2.1. ID-integer
        1.2.2. Name_type-varchar
2. Перечень территорий(Location-p)
    2.1. Описание:
        2.1.1. Таблица «Location-p» содержит информацию о территории. В данном примере – это справочник городов.
    2.2. Поля:
        2.2.1. ID-integer
        2.2.2. Location-varchar
3. Перечень пользователей(Users_p)
    3.1. Описание:
        3.1.1. Таблица содержит информацию о пользователе. В таблице отсутствуют поля с именами т.к. подразумевается, что поле ID будет являться некоторым универсальным идентификатором, например, СНИЛС. Поле «LocationID» содержит место постоянного пребывания пользователя. Поле «Age» содержит возраст пользователя. В поле Sexid указывается пол пользователя.
    3.2. Поля
        3.2.1. ID-integer
        3.2.2. LocationID-integer
        3.2.3. Age-integer
        3.2.4. Sexid-integer
4. Тип мест совершения операций
    4.1. Описание:
        4.1.1. Справочник типов возможных мест совершения операции. Например, пользователь может пополнить номер телефона через WEB приложение или через кассу офиса.
    4.2. Поля:
        4.2.1. ID-integer
        4.2.2. Channel-varchar
5. Тип операций
    5.1. Описание:
        5.1.1. Справочник типов операций. Например, пополнение счета, платеж, перевод и.д.р.
    5.2. Поля:
        5.2.1. ID-integer
        5.2.2. OperationName-varchar
6. Перечень карт
    6.1. ID-integer
    6.2. Userid-integer
    6.3. Location_card-integer
    6.4. Typcard-integer
    6.5. MB- boolean
7. Перечень операций
    7.1. ID-integer

7.2. ChannelID-integer
7.3. OpernameID-integer
7.4. CardID-integer
7.5. LocationID-integer
7.6. Summ-numeric

8. Sex_p
8.1. ID-integer
8.2. SexTyp-varchar

Схема базы:



Загрузка данных:

С Google drive

idimameza@instance-1:~$ python download_google_drive/download_gdrive.py 11_9pX7MGXOxnaF73hptQOxb699dqJ_g6 data_dp.zip

```
psql --host $APP_POSTGRES_HOST  -U postgres -c \
   "DROP TABLE IF EXISTS Links_p"
psql --host $APP_POSTGRES_HOST  -U postgres -c \
   "DROP TABLE IF EXISTS Cards_p"
psql --host $APP_POSTGRES_HOST  -U postgres -c \
   "DROP TABLE IF EXISTS Users_p"
psql --host $APP_POSTGRES_HOST  -U postgres -c \
   "DROP TABLE IF EXISTS Opername_p"
psql --host $APP_POSTGRES_HOST  -U postgres -c \
   "DROP TABLE IF EXISTS Channel_p"
psql --host $APP_POSTGRES_HOST  -U postgres -c \
   "DROP TABLE IF EXISTS Cardstyps_p"
psql --host $APP_POSTGRES_HOST  -U postgres -c \
   "DROP TABLE IF EXISTS Location_p"
psql --host $APP_POSTGRES_HOST  -U postgres -c \
   "DROP TABLE IF EXISTS Sex_p"

echo "Загружаем Opername_p.csv..."
psql --host $APP_POSTGRES_HOST -U postgres -c '
  CREATE TABLE IF NOT EXISTS Opername_p (
   id integer PRIMARY KEY,
   OperationName varchar(20)
   );'
psql --host $APP_POSTGRES_HOST -U postgres -c \
   "\\copy Opername_p FROM '/data/Data_pd/Opername_p.csv' DELIMITER ',' CSV HEADER"
```

```
echo " Загружаем Channel_p.csv..."
psql --host $APP_POSTGRES_HOST -U postgres -c '
  CREATE TABLE IF NOT EXISTS Channel_p (
    id integer PRIMARY KEY,
    Channel varchar(20)
    );'
psql --host $APP_POSTGRES_HOST -U postgres -c \
    "\\copy Channel_p FROM '/data/Data_pd/Channel_p.csv' DELIMITER ',' CSV HEADER"



echo " Загружаем  Cardstyps_p.csv..."
psql --host $APP_POSTGRES_HOST -U postgres -c '
  CREATE TABLE IF NOT EXISTS Cardstyps_p (
    id integer PRIMARY KEY,
    Name_type varchar(20)
    );'
psql --host $APP_POSTGRES_HOST -U postgres -c \
    "\\copy Cardstyps_p FROM '/data/Data_pd/Cardstyps_p.csv' DELIMITER ',' CSV HEADER"



echo " Загружаем Location_p.csv..."
psql --host $APP_POSTGRES_HOST -U postgres -c '
  CREATE TABLE IF NOT EXISTS Location_p (
    id integer PRIMARY KEY,
    Location varchar(60)
    );'
psql --host $APP_POSTGRES_HOST -U postgres -c \
    "\\copy Location_p FROM '/data/Data_pd/Location_p.csv' DELIMITER ',' CSV HEADER"

echo " Загружаем sex_p.csv..."
```

```
psql --host $APP_POSTGRES_HOST -U postgres -c '
  CREATE TABLE IF NOT EXISTS Sex_p (
    id integer PRIMARY KEY,
    SexTyp varchar(20)
    );'
psql --host $APP_POSTGRES_HOST -U postgres -c \
    "\\copy Sex_p FROM '/data/Data_pd/Sex_p.csv' DELIMITER ',' CSV HEADER"




echo " Загружаем Users_p.csv..."
psql --host $APP_POSTGRES_HOST -U postgres -c '
  CREATE TABLE IF NOT EXISTS Users_p (
    id bigint PRIMARY KEY,
    LocationID integer,
    Age integer,
  SexID integer REFERENCES Sex_p(ID)
    );'

psql --host $APP_POSTGRES_HOST -U postgres -c \
    "\\copy Users_p FROM '/data/Data_pd/Users_p.csv' DELIMITER ',' CSV HEADER"

echo " Загружаем Cards_p.csv..."
psql --host $APP_POSTGRES_HOST -U postgres -c '
  CREATE TABLE IF NOT EXISTS Cards_p (
    id bigint PRIMARY KEY,
  Userid bigint REFERENCES Users_p(ID),
  LocationID integer REFERENCES Location_p(ID),
    typcardid integer REFERENCES Cardstyps_p(ID),
  MB boolean
  );'
```

```
psql --host $APP_POSTGRES_HOST -U postgres -c \
   "\\copy Cards_p FROM '/data/Data_pd/Cards_p.csv' DELIMITER ',' CSV HEADER"


echo " Загружаем links_p.csv..."
psql --host $APP_POSTGRES_HOST -U postgres -c '
  CREATE TABLE IF NOT EXISTS links_p (
    id bigint PRIMARY KEY,
    CardID bigint REFERENCES cards_p(ID),
    LocationID integer REFERENCES Location_p(ID),
    ChannelId integer REFERENCES Channel_p(ID),
  OpernameID integer REFERENCES Opername_p(ID),
  Summ numeric
  );'


psql --host $APP_POSTGRES_HOST  -U postgres -c \
   "\\copy links_p FROM '/data/Data_pd/links_p.csv' DELIMITER ',' CSV HEADER"
```

Запросы:

1. Вывести по убыванию средний чек операции «Payment» в топ 10 городах и имена городов, где возраст пользователя меньше 25 лет.
2. Вывести 10 пользователей с максимальным кол-вом карт.
3. Вывести ТОП 10 пользователей с наибольшим средним чеком операции «Transfer» у кого место регистрации не совпадало с местом проведения транзакции
4. Вывести список мест совершения операций и названия операций с максимальной суммой операций
5. Вывести топ пользователей и количество транзакций у которых были транзакции в разных городах.
6. Вывести топ городов где пользователи предпочитают совершать операции по кредитным картам. Кол-во операций по кредиткам.
7. Вывести топ городов, где средний чек по операции «Output» выше среднего по стране на 1%. (Данный процент взят для текущих данных. В реальной жизни рассматриваются более существенные отклонения)
8. Вывести город, где разница количества операций «Output» проведенных в vsp и US максимальная.
9. Вывести кол-во транзакций и среднюю сумму транзакции по типам операции
10. Вывести топ 10 пользователей с самым высоким оборотом.

----------1----------

With aaa as (select id from users_p where age<25 limit 3), bbb as (select id from cards_p where userid in (select * from aaa)),ccc as (select locationid, avg(summ) as avg from links_p where cardid in (select * from bbb) and opernameid in (select id from opername_p where OperationName='Payment') group by locationid) select location,avg from location_p inner join ccc on (location_p.id=ccc.locationid) order by avg DESC;

----------2----------

Select users_p.id,count(cards_p.id) as countt from users_p inner join cards_p on (users_p.id= cards_p.userid) group by users_p.id order by countt DESC limit 10;

----------3----------

With aaa as (select users_p.id, users_p.locationid, cards_p.id as card from users_p inner join cards_p on (users_p.id= cards_p.userid)) select aaa.id, avg(summ) from aaa inner join links_p on (aaa.card=links_p.cardid) where aaa.locationid<>links_p.locationid and opernameid in (select id from opername_p where OperationName='Transfer') group by aaa.id order by avg(summ) DESC limit 10;

----------4----------

With mmm as (Select cardid,channelid, opernameid, summ,max(summ) over (partition by opernameid) as maxx from links_p),ttt as (select channelid, opernameid, avg(maxx) as maxsumm from mmm where summ=maxx group by channelid, opernameid), ppp as (select channelid, operationname, maxsumm from ttt inner join opername_p on (ttt.opernameid=opername_p.id)) select channel, operationname,maxsumm from ppp inner join channel_p on (ppp.channelid= channel_p.id) order by channel, operationname ;

----------5----------

With bbb as (Select users_p.id as idname, users_p.locationid as locationname,cards_p.id as cc from users_p inner join cards_p on (users_p.id=cards_p.userid)) select bbb.idname, count(links_p.id) from bbb inner join links_p on (bbb.cc=links_p.cardid) where bbb.locationname<>locationid group by bbb.idname order by count(links_p.id) desc limit 10;

----------6----------

Select locationid, count(id) from links_p where cardid in (Select id from cards_p where typcardid=2) group by locationid order by count(id) desc limit 10;

----------7----------

With bbb as (select id from opername_p where OperationName='Output') ,r1 as (Select avg(summ)*1.01 as t1 from links_p where opernameid in (select * from bbb) group by opernameid), qq as(Select locationid,avg(summ) as r2 from links_p where opernameid in (select * from bbb) group by locationid) select r2 from qq cross join r1 where r2>t1;

----------8----------

With g1 as (select id from opername_p where OperationName='Output'), g2 as (select id from Channel_p where Channel='VSP'), g3 as (select id from Channel_p where Channel='US'), aaa as (Select locationid, count(id) as c1 from links_p where opernameid in (select * from g1) and channelid in (select * from g2) group by locationid), bbb as (Select locationid, count(id) as c2 from links_p where opernameid in (select * from g1) and channelid in (select * from g3) group by locationid), ttt as (select

aaa.locationid as l1,aaa.c1,bbb.c2 from aaa inner join bbb on (aaa.locationid=bbb.locationid)),rez as (select l1 from ttt order by (c1+c2) desc limit 1) select location from location_p where id in (select * from rez);

----------9----------

Select operationname, count(links_p.id), avg(summ) from links_p inner join opername_p on (links_p.opernameid= opername_p.id) group by operationname order by count(links_p.id) desc ;

----------10----------

With T2 as (Select cardid, sum(summ) as ss from links_p group by cardid), T1 as (Select users_p.id as us, cards_p.id as ca from users_p inner join cards_p on (users_p.id=cards_p.userid)) select us, ss from t1 inner join t2 on (t1.ca=t2.cardid) order by ss desc limit 10;