```sql
DROP TABLE dmart_trn_nch.organizations_y;
DROP TABLE dmart_trn_nch.people_y;
DROP TABLE dmart_trn_nch.customers_y;

-- Создание таблицы organizations
CREATE TABLE dmart_trn_nch.organizations_y (
    index INT,
    organization_id STRING,
    name STRING,
    website STRING,
    country STRING,
    description STRING,
    founded INT,
    industry STRING,
    number_of_employees INT,
    n_group INT
)
CLUSTERED BY (n_group) INTO 4 BUCKETS
ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
STORED AS TEXTFILE;


-- Создание таблицы people
CREATE TABLE dmart_trn_nch.people_y (
    index INT,
    user_id STRING,
    first_name STRING,
    last_name STRING,
    sex STRING,
    email STRING,
    phone STRING,
    date_of_birth DATE,
    job_title STRING,
    n_group INT
)
CLUSTERED BY (n_group) INTO 4 BUCKETS
ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
STORED AS TEXTFILE;

-- Создание таблицы customers
CREATE TABLE dmart_trn_nch.customers_y (
    index INT,
    customer_id STRING,
    first_name STRING,
    last_name STRING,
    company STRING,
    city STRING,
    country STRING,
    phone_1 STRING,
    phone_2 STRING,
    email STRING,
    subscription_date DATE,
    website STRING,
    n_group INT
)
PARTITIONED BY (subscription_year INT)
CLUSTERED BY (n_group) INTO 4 BUCKETS
ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
STORED AS TEXTFILE;

-- Загрузка данных в таблицу dmart_trn_nch.customers_y
LOAD DATA INPATH '/user/yakovlev/customers_new.csv'
 INTO TABLE dmart_trn_nch.customers_y ;

-- Загрузка данных в таблицу dmart_trn_nch.organizations_y
LOAD DATA INPATH '/user/yakovlev/organizations_new.csv'
 INTO TABLE dmart_trn_nch.organizations_y ;

-- Загрузка данных в таблицу dmart_trn_nch.people_y
LOAD DATA INPATH '/user/yakovlev/people_new.csv'
 INTO TABLE dmart_trn_nch.people_y ;




-- запрос: на уровне каждого года получить целевую возрастную
группу подписчиков — то есть,
--возрастную группу, представители которой чаще всего совершали
подписку именно в текущий год на текущую компанию

with people_age as (select index, extract(year from CURRENT_DATE)
- extract(year from date_of_birth)
as age from dmart_trn_nch.people_y),
people_group as (select index, case when age <= 18 then '0 - 18'
when age > 18 and age <= 25 then '19 - 25'
when age > 25 and age <= 45 then '26 - 45'
when age > 45 and age <= 60 then '46 - 60'
else '60 - '
end as age_group from people_age),
stat_table as (select company, n_group as n_year, age_group,
count(*) as n_row from dmart_trn_nch.customers_y as cus
join people_group as pg on cus.index = pg.index
group by company, n_group, age_group),
table_range as (select company, n_year, age_group, ROW_NUMBER()
over(partition by company, n_year order by n_row desc)
as n_range from stat_table)
select  company, n_year, age_group from table_range
where n_range = 1

;
```

```
План запросов:
Plan optimized by CBO.

Vertex dependency in root stage
Map 2 <- Map 1 (BROADCAST_EDGE)
Reducer 3 <- Map 2 (SIMPLE_EDGE)
Reducer 4 <- Reducer 3 (SIMPLE_EDGE)

Stage-0
  Fetch Operator
    limit:-1
    Stage-1
      Reducer 4
      File Output Operator [FS_22]
        Select Operator [SEL_17] (rows=12499 width=281)
          Output:["_col0","_col1","_col2"]
          Filter Operator [FIL_25] (rows=12499 width=289)
            predicate:(ROW_NUMBER_window_0 = 1)
            PTF Operator [PTF_16] (rows=24999 width=290)
              Function definitions:[{},
{"name:":"windowingtablefunction","order by:":"_col3 DESC NULLS
LAST","partition by:":"_col0, _col1"}]
              Select Operator [SEL_15] (rows=24999 width=290)
                Output:["_col0","_col1","_col2","_col3"]
              <-Reducer 3 [SIMPLE_EDGE] vectorized
                SHUFFLE [RS_49]
                  PartitionCols:_col0, _col1
                  Group By Operator [GBY_48] (rows=24999
width=290)
                    Output:
["_col0","_col1","_col2","_col3"],aggregations:
["count(VALUE._col0)"],keys:KEY._col0, KEY._col1, KEY._col2
                  <-Map 2 [SIMPLE_EDGE] vectorized
                    SHUFFLE [RS_47]
                      PartitionCols:_col0, _col1, _col2
                      Group By Operator [GBY_46] (rows=49999
width=290)
                        Output:
["_col0","_col1","_col2","_col3"],aggregations:
["count()"],keys:_col1, _col2, _col4
                        Map Join Operator [MAPJOIN_45] (rows=99999
width=282)

Conds:RS_42._col0=SEL_44._col0(Inner),Output:
["_col1","_col2","_col4"]
                        <-Map 1 [BROADCAST_EDGE] vectorized
                          BROADCAST [RS_42]
                            PartitionCols:_col0
                            Select Operator [SEL_41] (rows=100000
width=102)
                              Output:["_col0","_col1","_col2"]
                              Filter Operator [FIL_40]
(rows=100000 width=102)
                                predicate:index is not null
                                TableScan [TS_0] (rows=100001
width=102)

dmart_trn_nch@customers_y,cus,Tbl:COMPLETE,Col:COMPLETE,Output:
["index","company","n_group"]
                        <-Select Operator [SEL_44] (rows=100000
width=188)
                            Output:["_col0","_col1"]
                            Filter Operator [FIL_43] (rows=100000
width=60)
                              predicate:index is not null
                              TableScan [TS_3] (rows=100001
width=60)

dmart_trn_nch@people_y,people_y,Tbl:COMPLETE,Col:COMPLETE,Output:
["index","date_of_birth"]
```