

Лабораторная работа №8

Цель задания: попрактиковаться с таблицами в GreenPlum и со схемой виртуализацией данных.

а) Используя HDFS файл `/user/yakupov/data/recommendations.csv` постройте интеграцию Greenplum -> HDFS, используя PXF подход и EXTERNAL таблицу. Назовите вашу EXTERNAL таблицу, используя шаблон именования “lab9_recom_фамилия”

```
CREATE EXTERNAL TABLE lab9_recom_pavlov (  
    app_id int8,  
    helpful int4,  
    funny int4,  
    date date,  
    is_recommended bool,  
    hours numeric,  
    user_id int8,  
    review_id int8  
)  
LOCATION (  
    'pxf://user/yakupov/data/recommendations.csv?PROFILE=hdfs:csv&SERVER=hadoop'  
)  
FORMAT 'CSV' ( delimiter ',' null '' escape '"' quote '"' header );
```

б) Создайте физическую таблицу в GreenPlum, используя шаблон именования “lab9_recom_new_фамилия” с моделью данных и типами абсолютно такими же как и у “lab9_recom_фамилия”. Выберите ключ дистрибуции, какой по вашему мнению необходим для предотвращения SKEW аномалии (самый простой - DISTRIBUTED RANDOMLY).

```
CREATE TABLE lab9_recom_new_pavlov (  
    app_id int8,  
    helpful int4,  
    funny int4,  
    date date,  
    is_recommended bool,  
    hours numeric,  
    user_id int8,  
    review_id int8  
)  
DISTRIBUTED RANDOMLY;
```

Заполните вашу физическую таблицу “lab9_recom_new_фамилия” 5000 новыми строками, НО со значением поля модели данных “date” за 2023 год (чтобы данные из предоставленного файла и вновь сгенерированные не пересекались)

```

INSERT INTO lab9_recom_new_pavlov (app_id, helpful, funny, date, is_recommended,
hours, user_id, review_id)
SELECT
  (random()*10000)::int8,
  (random()*10)::int4,
  (random()*5)::int4,
  date '2023-01-01' + (random()*364)::int4,
  (random() > 0.5),
  (random()*100)::numeric(5,2),
  (random()*10000)::int8,
  (random()*100000)::int8
FROM generate_series(1,5000);

```

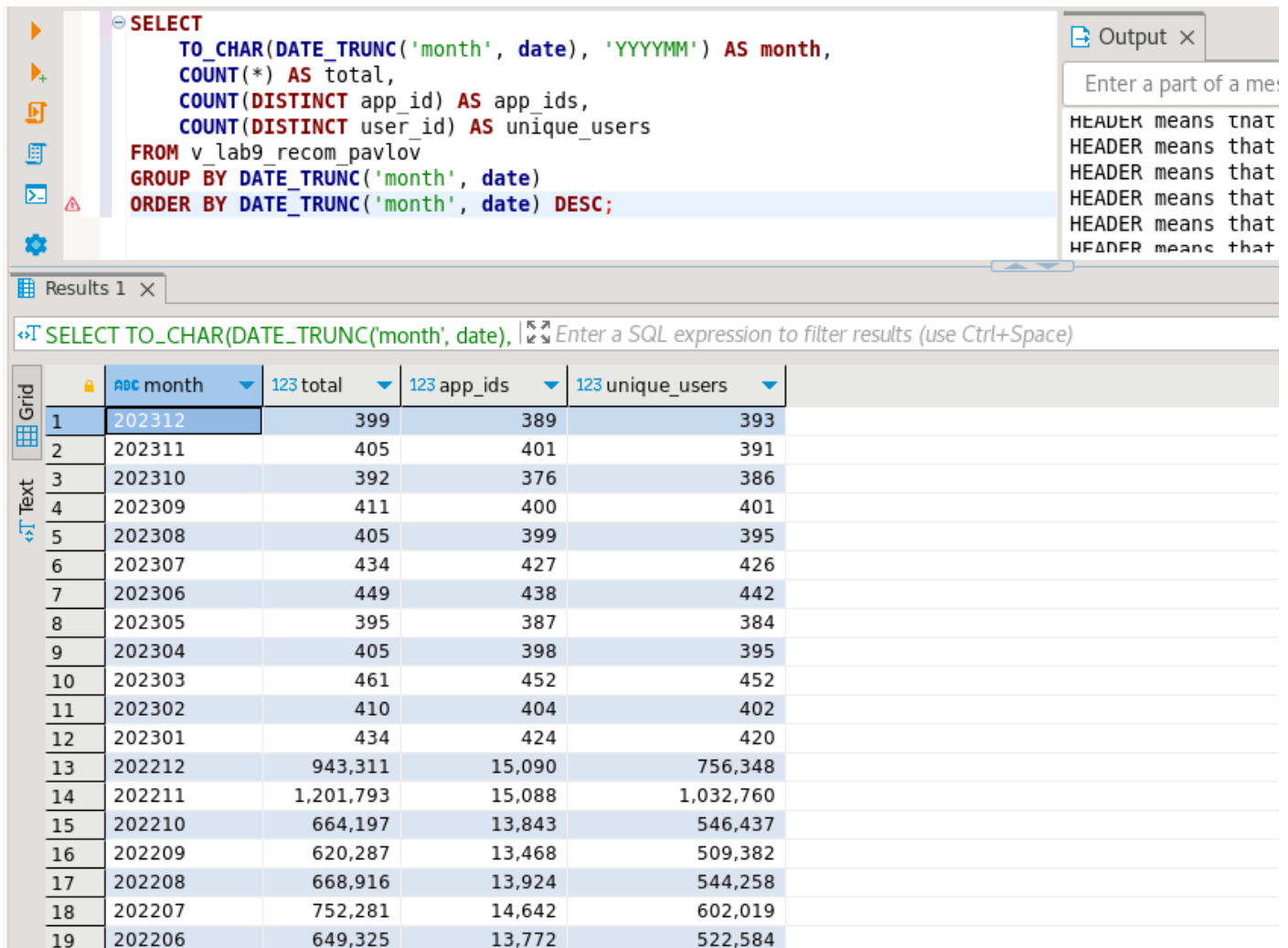
с) Создайте виртуальную таблицу VIEW с шаблоном именования “v_lab9_recom_фамилия”, которая будет содержать UNION операцию для двух таблиц и пунктов а) и б)

```

CREATE VIEW v_lab9_recom_pavlov AS
SELECT * FROM lab9_recom_pavlov
UNION
SELECT * FROM lab9_recom_new_pavlov;

```

Результат работы выюхи:



SELECT

```

TO_CHAR(DATE_TRUNC('month', date), 'YYYYMM') AS month,
COUNT(*) AS total,
COUNT(DISTINCT app_id) AS app_ids,
COUNT(DISTINCT user_id) AS unique_users
FROM v_lab9_recom_pavlov
GROUP BY DATE_TRUNC('month', date)
ORDER BY DATE_TRUNC('month', date) DESC;

```

Output x

Enter a part of a me:

HEADER means that
HEADER means that
HEADER means that
HEADER means that
HEADER means that

Results 1 x

SELECT TO_CHAR(DATE_TRUNC('month', date), 'YYYYMM') AS month, COUNT(*) AS total, COUNT(DISTINCT app_id) AS app_ids, COUNT(DISTINCT user_id) AS unique_users FROM v_lab9_recom_pavlov GROUP BY DATE_TRUNC('month', date) ORDER BY DATE_TRUNC('month', date) DESC;

	month	total	app_ids	unique_users
1	202312	399	389	393
2	202311	405	401	391
3	202310	392	376	386
4	202309	411	400	401
5	202308	405	399	395
6	202307	434	427	426
7	202306	449	438	442
8	202305	395	387	384
9	202304	405	398	395
10	202303	461	452	452
11	202302	410	404	402
12	202301	434	424	420
13	202212	943,311	15,090	756,348
14	202211	1,201,793	15,088	1,032,760
15	202210	664,197	13,843	546,437
16	202209	620,287	13,468	509,382
17	202208	668,916	13,924	544,258
18	202207	752,281	14,642	602,019
19	202206	649,325	13,772	522,584