Campaign Results

Overview

- Used PostgreSQL to query the following 3 tasks
- Followed by query are the outputs for each task
- Additional notes are included in the 'Notes' section
- Screenshots of query and outputs are included in the Appendix

Task 1 - Write a query in SQL to find the 2nd highest stoppage time which had been added in 2nd half of play

Query:

```
SELECT stop2_sec FROM Scores s1
WHERE 1 = (SELECT COUNT(DISTINCT stop2_sec)
        FROM Scores s2
        WHERE s2.stop2_sec > s1.stop2_sec);
```

Output:

374

Task 2a - Which action had more points earned per month?

Query:

```
SELECT actions, total_points, creation_month
FROM
        (SELECT
            rank() OVER (PARTITION BY to_char(creation_date, 'Mon')
            ORDER BY SUM(points_earned) DESC) AS rank, action_ AS actions,
            SUM(points_earned) AS total_points
            ,to_char(creation_date, 'Mon') AS creation_month
            FROM campaigns
            GROUP BY to_char(creation_date, 'Mon'), action_) AS sq
WHERE rank = 1;
```

Task 2a - Which action had more points earned per month?

Output:

Month	Action	Points earned
January	Link New Card	2334
February	Link New Card	1878
March	Complete Survey	11

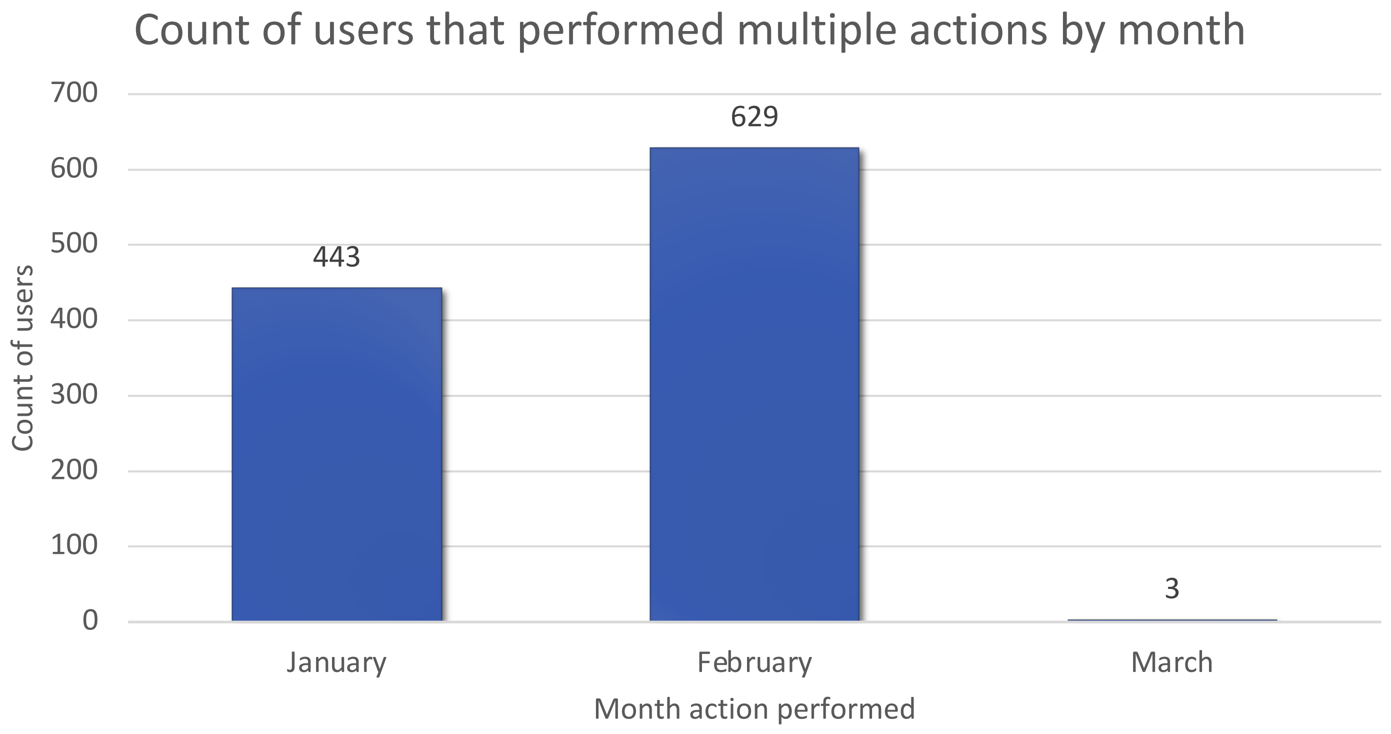
Task 2b - How many users performed multiple actions per month?

Query:

```
SELECT creation_month, COUNT (userid) AS user_count
FROM
    (SELECT
    to_char(creation_date, 'Mon') AS creation_month, COUNT(action_) AS actions, userid
    FROM campaigns
    GROUP BY to_char(creation_date, 'Mon'), userid
    HAVING COUNT (action_) > 1) AS sq
GROUP BY creation_month;
```

Task 2b - How many users performed multiple actions per month?

Output:



Task 2c - How many users from campaign 8 also participated in campaign 9?

Query:

```
SELECT COUNT(DISTINCT userid) AS user_count
FROM campaigns
WHERE userid IN (SELECT userid FROM
    campaigns WHERE campaign_id = 8)
    AND userid IN (SELECT userid FROM
    campaigns WHERE campaign_id = 9);
```

Output:

54

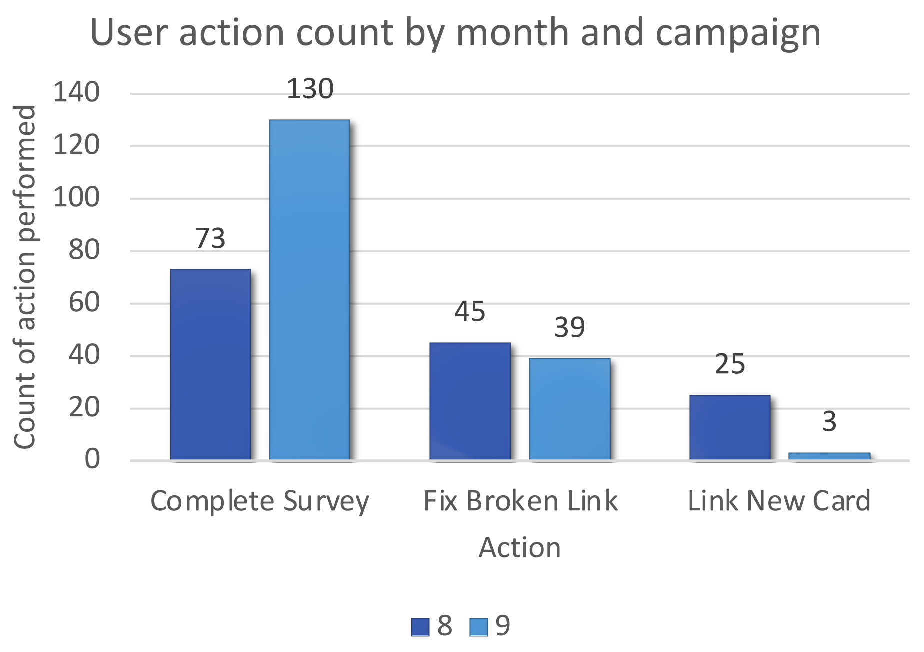
Task 2d - From question 3 users, was there any change in participation from campaign 8 to campaign 9?

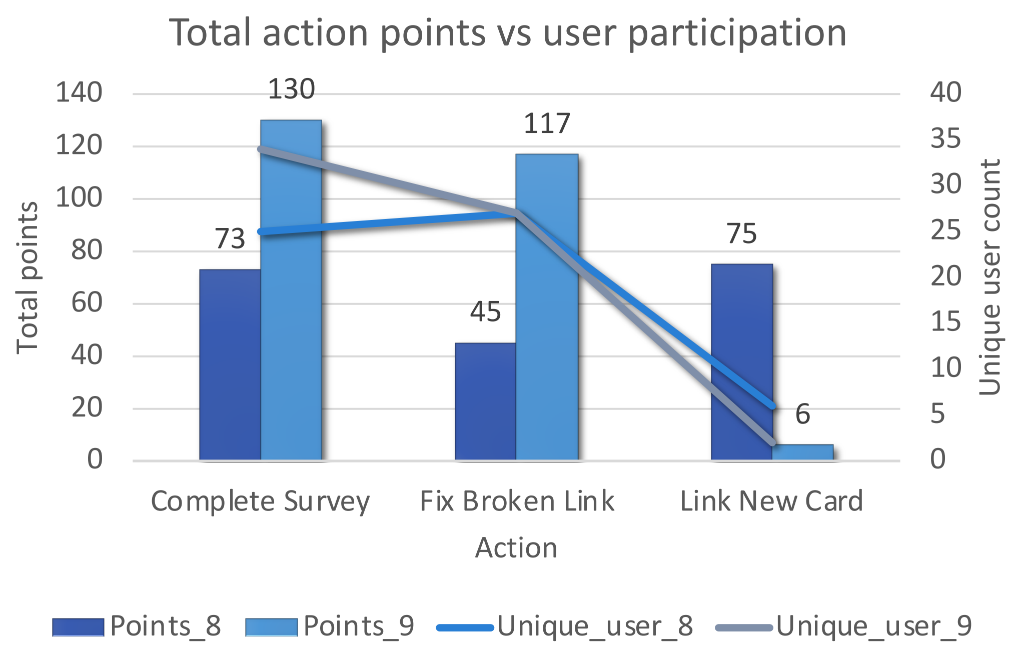
Query:

```
 SELECT campaign_id, action_ AS actions, SUM(points_earned) AS total_points,
                COUNT(action_) AS total_actions, COUNT(DISTINCT userid) AS distinct_user
FROM campaigns
WHERE userid IN (SELECT userid FROM campaigns WHERE campaign_id = 8)
                AND userid IN (SELECT userid FROM campaigns WHERE campaign_id = 9)
GROUP BY campaign_id, action_
ORDER BY action_, campaign_id;
```

Task 2d - From question 3 users, was there any change in participation from campaign 8 to campaign 9?

Output:





Task 2e - Which action was more attractive to users?

Query:

```
SELECT action_, COUNT(userid) AS user_count, COUNT(DISTINCT userid) AS distinct_count
FROM campaigns
GROUP BY action_
ORDER BY COUNT(*) DESC;
```

Task 2e - Which action was more attractive to users?

Output:

- **Action ‘Complete_Survey’ is most attractive to users**
- It was performed 2,328 times in total
- 656 unique users performed this action

Task 3 - Write SQL query to aggregate the above data into one query after converting operators into Sql Expression

Query:

```
SELECT  s.store_id, s.store_name,
    replace(
      string_agg(((CAST(CASE WHEN left_id IS NOT NULL THEN left_operator
      ELSE ('') END AS VARCHAR) || replace(replace(mrc.sql_expression, 'A', mr.field), 'B', mr.value_))), '*'
      ORDER BY COALESCE(mr.left_id, -100)), '*', ' ') AS consolidatedview
FROM stores AS s
LEFT OUTER JOIN store_matching_rules AS smr
   ON s.store_id = smr.store_id
LEFT OUTER JOIN matching_rules AS mr
   ON smr.matching_rule_id = mr.matching_rule_id
LEFT OUTER JOIN matching_rule_conversion AS mrc
   ON UPPER(mr.operator_) = UPPER(mrc.operator_)
GROUP BY s.store_id, s.store_name;
```

Task 3 - Write SQL query to aggregate the above data into one query after converting operators into Sql Expression

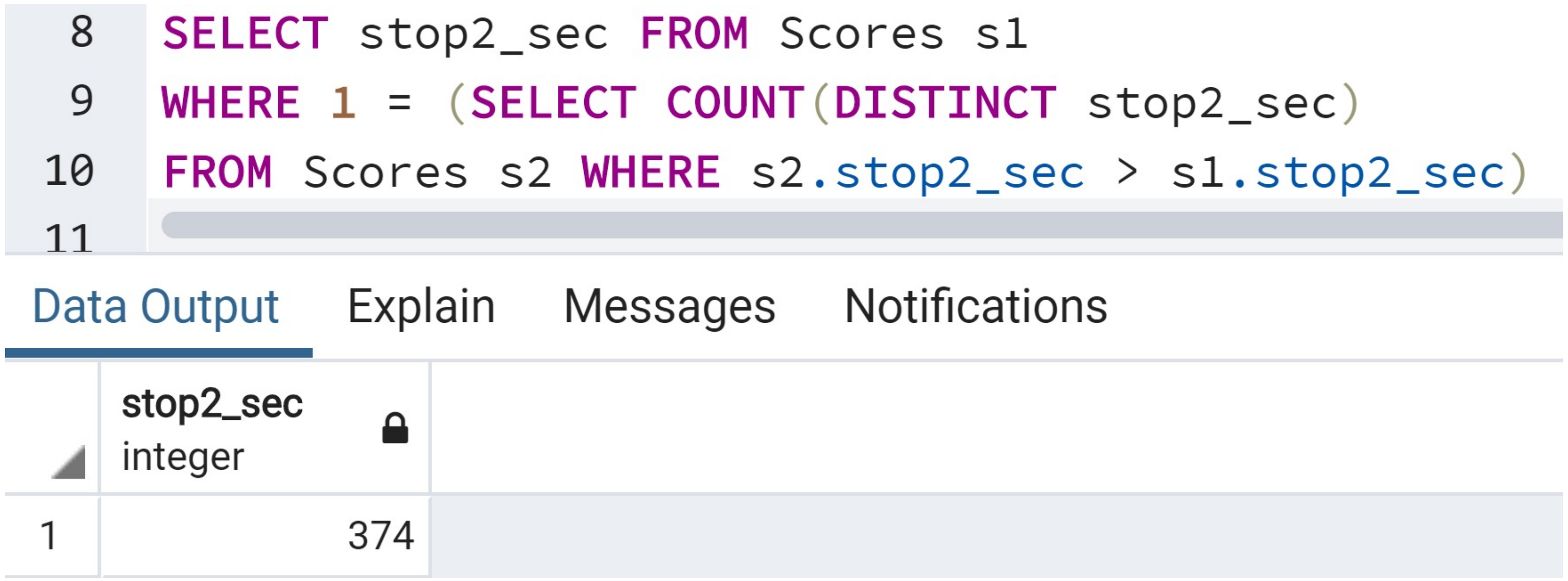
Output:

Store ID	Store name	Consolidatedview
3	Lexar Pharma	descname ILIKE "%Lexar Pharma%" ordescname ILIKE "%Lexar Pharmacy%" ordescname ~ '\yLexar PHARMACY #8164\y'
7	Domino Shoes	descname ILIKE %Domino # 1%" or descname ~ '\yDominoSHOES.COM\y'
17	COMBOS Tea	descname ILIKE %COMBOSTEA%" and descname NOT ILIKE "%HOUSE%" or descname ILIKE "%COMBOS TEA%"

Appendix

Task 1

- Screenshot of PostgreSQL query and output



```sql
SELECT stop2_sec FROM Scores s1
WHERE 1 = (SELECT COUNT(DISTINCT stop2_sec)
FROM Scores s2 WHERE s2.stop2_sec > s1.stop2_sec)
```

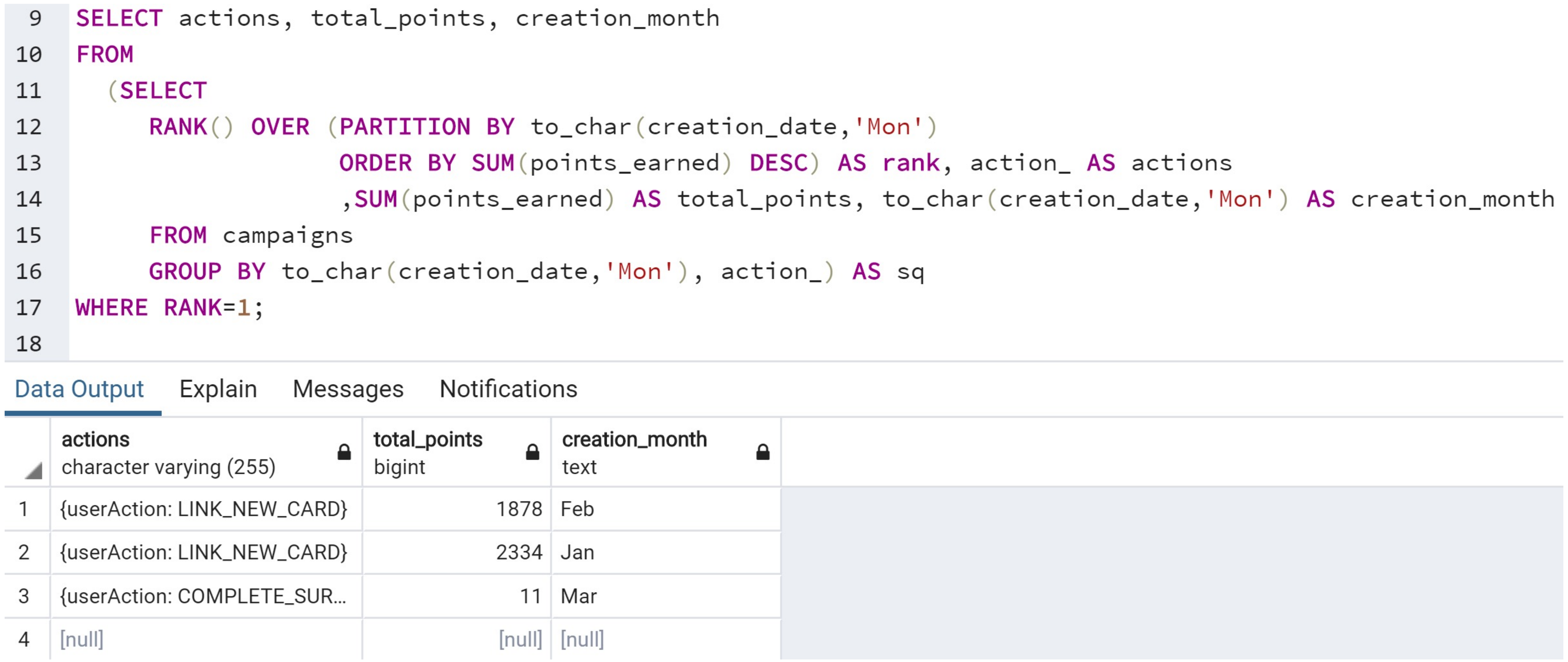
```
   stop2_sec
   integer
1        374
```

Task 2

- Changed the column names to following:
  - points earned → points_earned
  - action → action_
  - creation date → creation_date

Task 2a

- Screenshot of PostgreSQL query and output

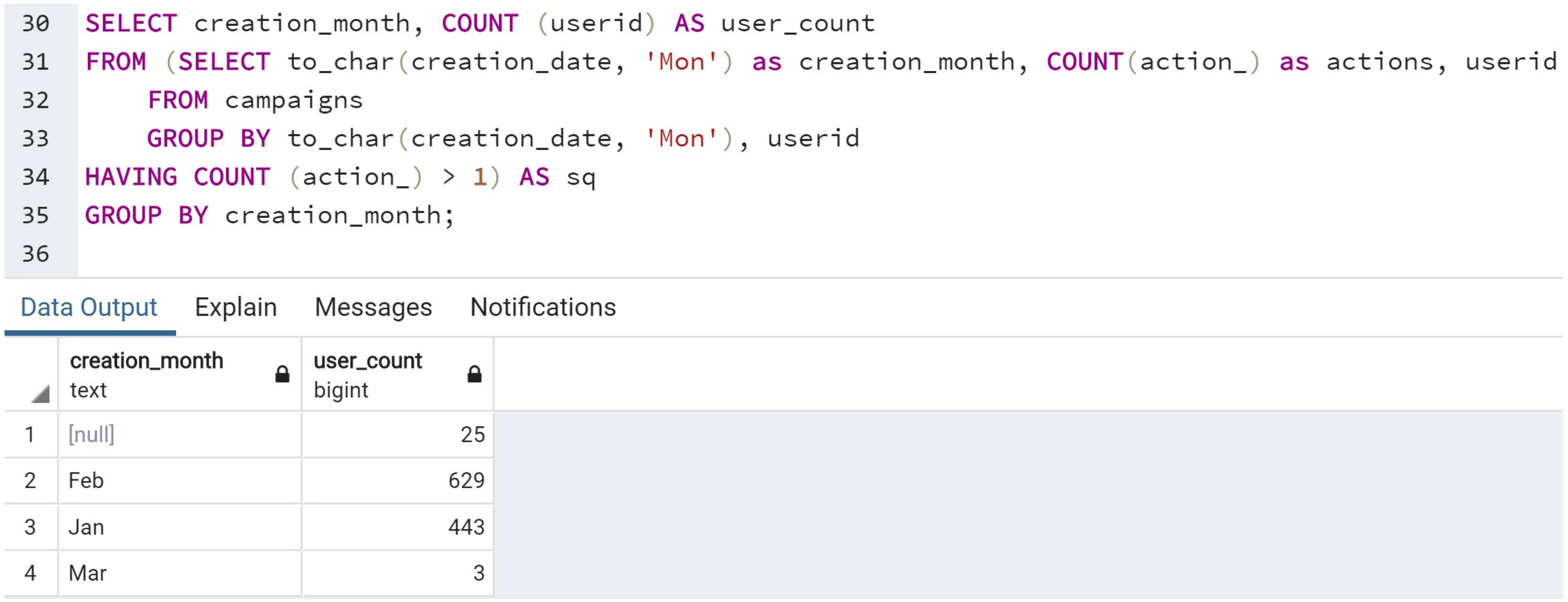


```
SELECT actions, total_points, creation_month
FROM
  (SELECT
     RANK() OVER (PARTITION BY to_char(creation_date,'Mon')
                  ORDER BY SUM(points_earned) DESC) AS rank, action_ AS actions
                  ,SUM(points_earned) AS total_points, to_char(creation_date,'Mon') AS creation_month
     FROM campaigns
     GROUP BY to_char(creation_date,'Mon'), action_) AS sq
WHERE RANK=1;
```

	actions character varying (255)	total_points bigint	creation_month text
1	{userAction: LINK_NEW_CARD}	1878	Feb
2	{userAction: LINK_NEW_CARD}	2334	Jan
3	{userAction: COMPLETE_SUR...	11	Mar
4	[null]	[null]	[null]

Task 2b

- Screenshot of PostgreSQL query and output



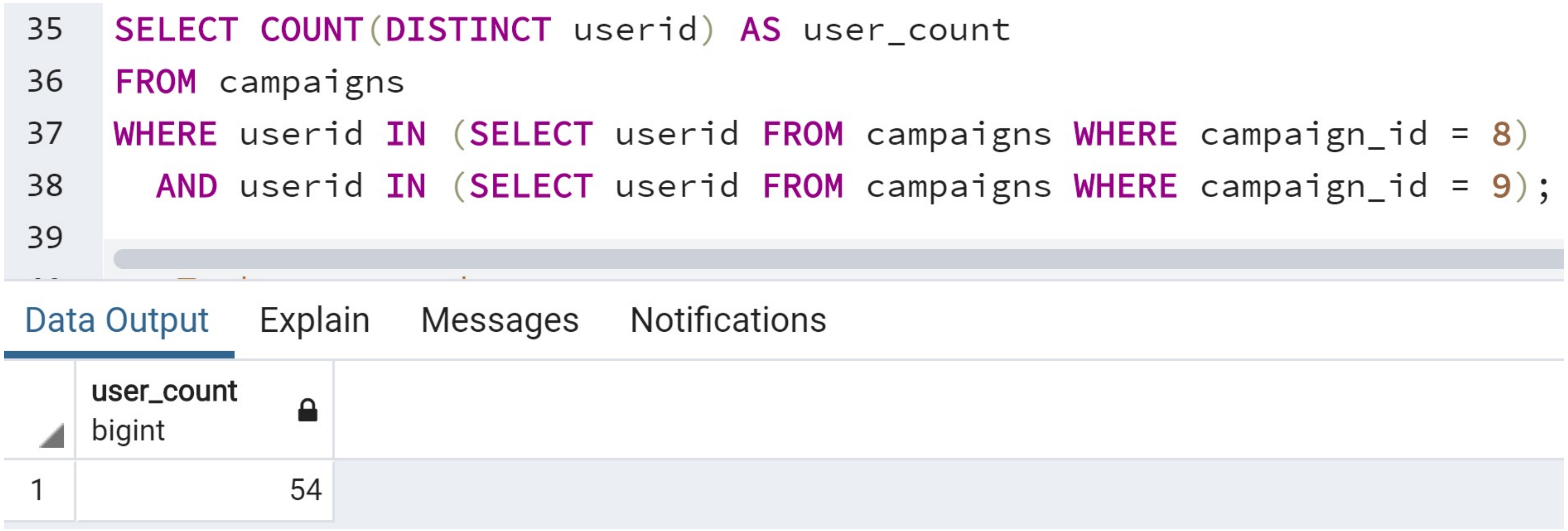
```sql
SELECT creation_month, COUNT (userid) AS user_count
FROM (SELECT to_char(creation_date, 'Mon') as creation_month, COUNT(action_) as actions, userid
    FROM campaigns
    GROUP BY to_char(creation_date, 'Mon'), userid
HAVING COUNT (action_) > 1) AS sq
GROUP BY creation_month;
```

Data Output | Explain | Messages | Notifications

	creation_month text	user_count bigint
1	[null]	25
2	Feb	629
3	Jan	443
4	Mar	3

Task 2c

- Screenshot of PostgreSQL query and output

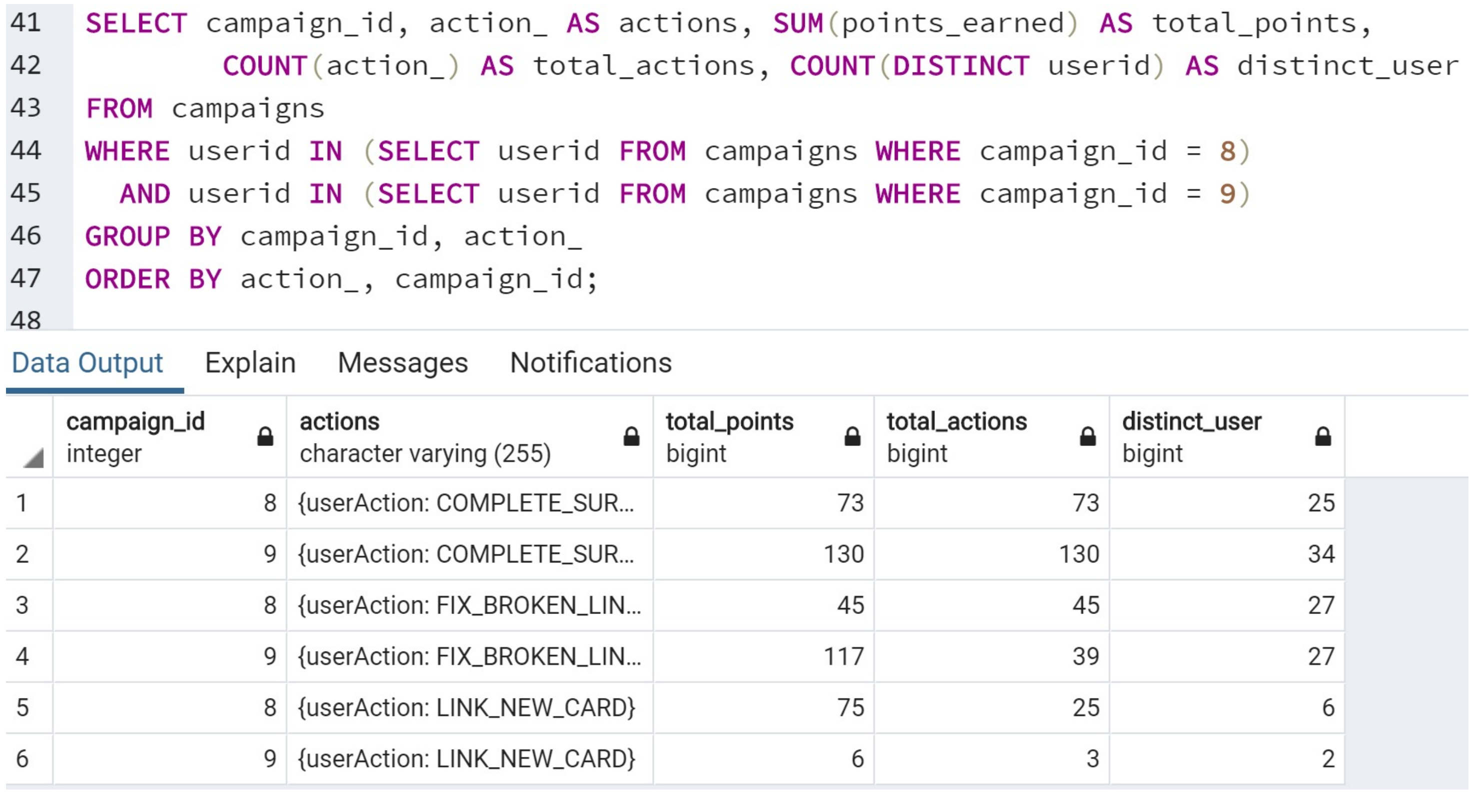


```
SELECT COUNT(DISTINCT userid) AS user_count
FROM campaigns
WHERE userid IN (SELECT userid FROM campaigns WHERE campaign_id = 8)
  AND userid IN (SELECT userid FROM campaigns WHERE campaign_id = 9);
```

```
   user_count
   bigint
1          54
```

Task 2d

- Screenshot of PostgreSQL query and output



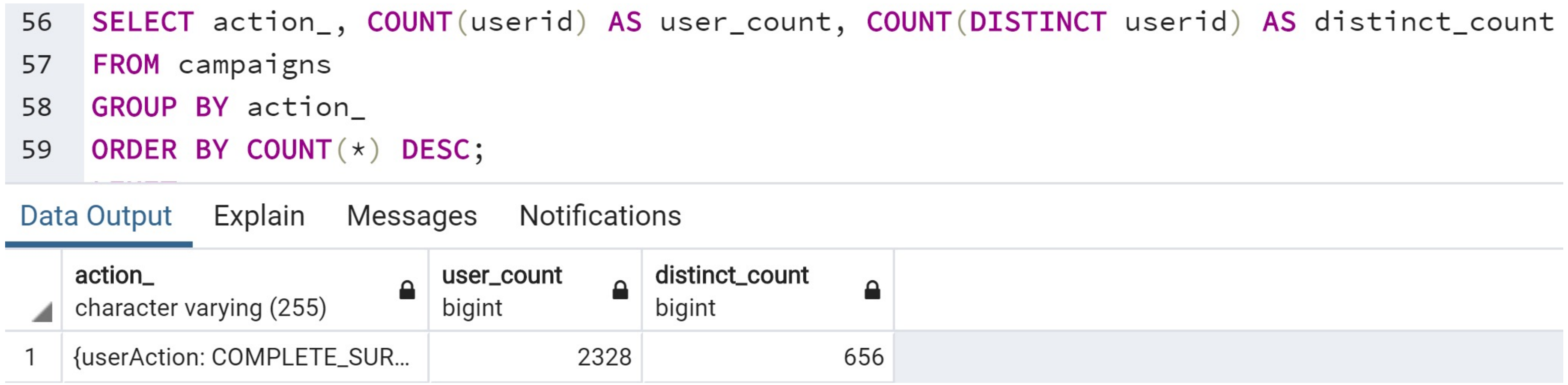
```
SELECT campaign_id, action_ AS actions, SUM(points_earned) AS total_points,
        COUNT(action_) AS total_actions, COUNT(DISTINCT userid) AS distinct_user
FROM campaigns
WHERE userid IN (SELECT userid FROM campaigns WHERE campaign_id = 8)
  AND userid IN (SELECT userid FROM campaigns WHERE campaign_id = 9)
GROUP BY campaign_id, action_
ORDER BY action_, campaign_id;
```

Data Output | Explain | Messages | Notifications

	campaign_id integer	actions character varying (255)	total_points bigint	total_actions bigint	distinct_user bigint
1	8	{userAction: COMPLETE_SUR...	73	73	25
2	9	{userAction: COMPLETE_SUR...	130	130	34
3	8	{userAction: FIX_BROKEN_LIN...	45	45	27
4	9	{userAction: FIX_BROKEN_LIN...	117	39	27
5	8	{userAction: LINK_NEW_CARD}	75	25	6
6	9	{userAction: LINK_NEW_CARD}	6	3	2

Task 2e

- Screenshot of PostgreSQL query and output



```
SELECT action_, COUNT(userid) AS user_count, COUNT(DISTINCT userid) AS distinct_count
FROM campaigns
GROUP BY action_
ORDER BY COUNT(*) DESC;
```

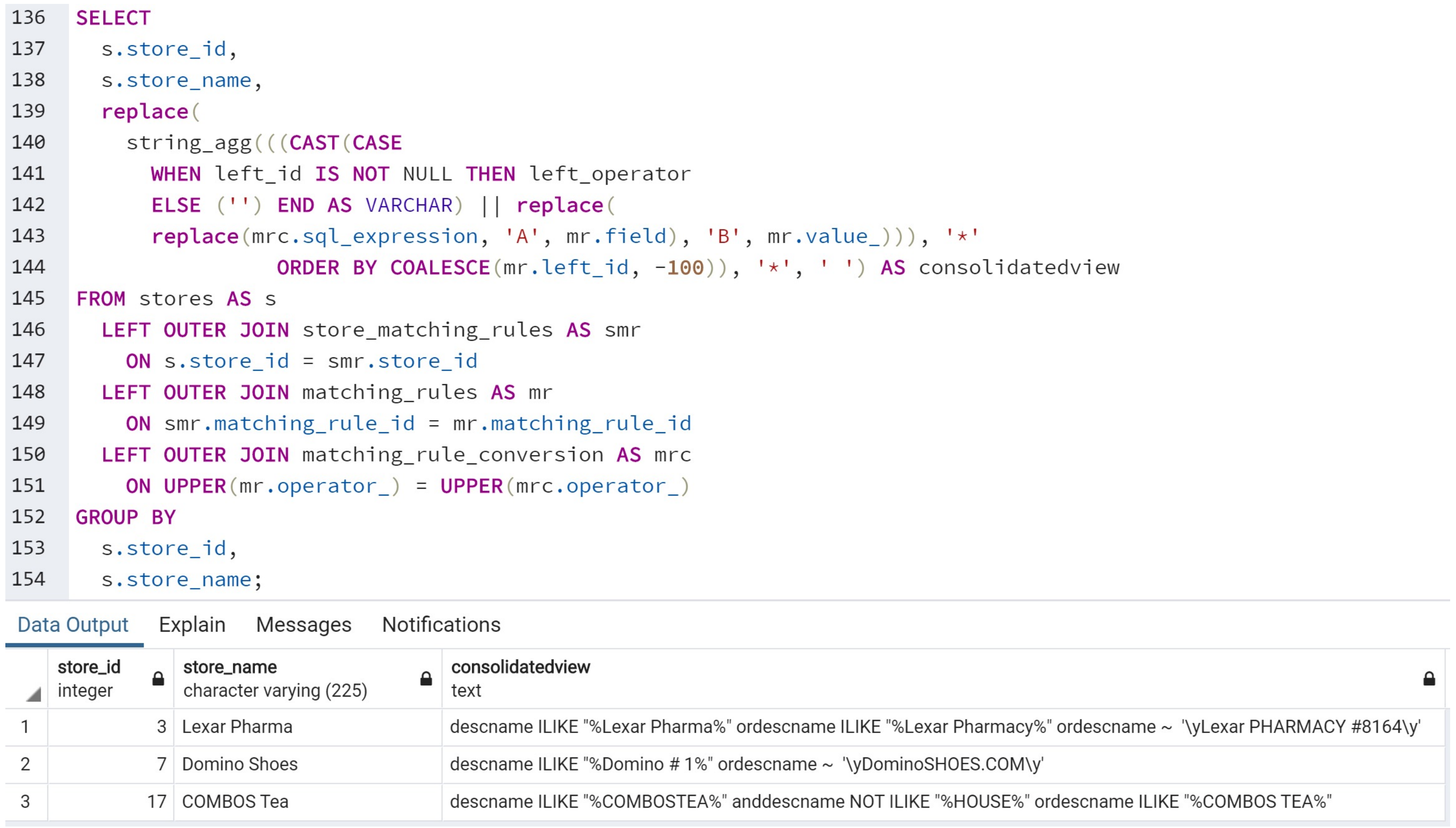
	action_ character varying (255)	user_count bigint	distinct_count bigint
1	{userAction: COMPLETE_SUR...	2328	656

Task 3

- Changed the column names in table Matching_Rules to following:
  - operator → operator_
  - value → value_

Task 3

- Screenshot of PostgreSQL query and output



```sql
SELECT
  s.store_id,
  s.store_name,
  replace(
    string_agg(((CAST(CASE
      WHEN left_id IS NOT NULL THEN left_operator
      ELSE ('') END AS VARCHAR) || replace(
      replace(mrc.sql_expression, 'A', mr.field), 'B', mr.value_))), '*'
                ORDER BY COALESCE(mr.left_id, -100)), '*', ' ') AS consolidatedview
FROM stores AS s
  LEFT OUTER JOIN store_matching_rules AS smr
    ON s.store_id = smr.store_id
  LEFT OUTER JOIN matching_rules AS mr
    ON smr.matching_rule_id = mr.matching_rule_id
  LEFT OUTER JOIN matching_rule_conversion AS mrc
    ON UPPER(mr.operator_) = UPPER(mrc.operator_)
GROUP BY
  s.store_id,
  s.store_name;
```

Data Output | Explain | Messages | Notifications

	store_id integer	store_name character varying (225)	consolidatedview text
1	3	Lexar Pharma	descname ILIKE "%Lexar Pharma%" ordescname ILIKE "%Lexar Pharmacy%" ordescname ~ '\yLexar PHARMACY #8164\y'
2	7	Domino Shoes	descname ILIKE "%Domino # 1%" ordescname ~ '\yDominoSHOES.COM\y'
3	17	COMBOS Tea	descname ILIKE "%COMBOSTEA%" anddescname NOT ILIKE "%HOUSE%" ordescname ILIKE "%COMBOS TEA%"