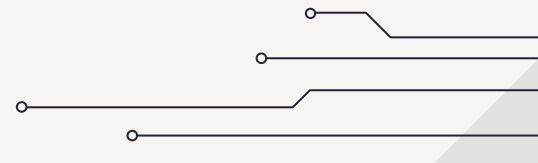




Portfolio SQL DANA Sentiment Analysis

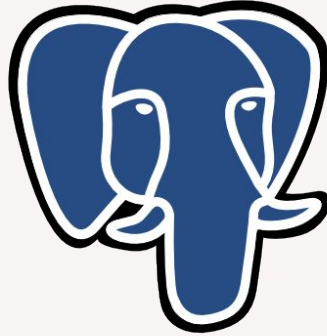
By : Johan Sanjaya



TOOLS



Excel



PostgreSQL



GitHub




SOURCE DATA

kaggleTM



DATA


- The topic to be worked on is “DANA Sentiment Analysis from Playstore Indonesia”
- The dataset is downloaded from web : [LINK](#)

 ALEX MARIO SIMANJUNTAK · UPDATED A MONTH AGO

1

New Notebook

Download (2 MB)



[Data Card](#) [Code \(0\)](#) [Discussion \(0\)](#) [Suggestions \(0\)](#)

About Dataset

50k labeled DANA App Reviews, scrapped from Google Playstore Indonesia.

The data contains 5 columns :

1. UserName
2. score
3. at
4. content
5. sentimen

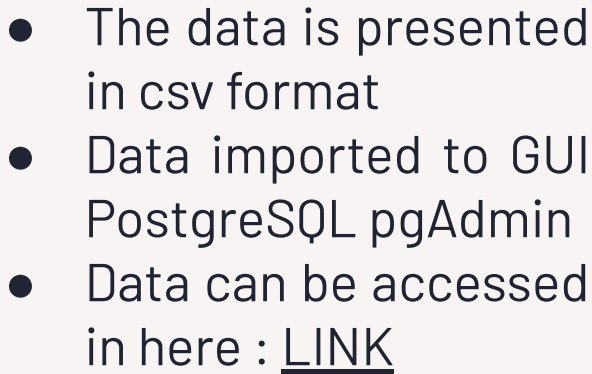
Usability ⓘ
3.53

License
Unknown

Expected update frequency
Not specified

Tags

- The data is presented in csv format
- Data imported to GUI PostgreSQL pgAdmin
- Data can be accessed in here : [LINK](#)



[LINK GITHUB](#)

PGADMIN DISPLAY

pgAdmin 4

File Object Tools Help

Object Explorer

- FTS Dictionaries
- FTS Parsers
- FTS Templates
- Foreign Tables
- Functions
- Materialized Views
- Operators
- Procedures
- Sequences
- Tables (1)
 - review_dana
 - Columns (7)
 - user_name
 - score
 - content
 - sentimen
 - date
 - month_name
 - year
 - Constraints
 - Indexes
 - RLS Policies
 - Rules
 - Triggers
 - Trigger Functions
 - Types
 - Views
 - Subscriptions
 - Login/Group Roles
 - Tablespaces

review_dana/postgres@Local

No limit

Query Query History

```
1 create table review_dana(  
2 User_name CHAR,  
3 Score INT,  
4 Content CHAR,  
5 Sentimen CHAR,  
6 Date date,  
7 Month_name CHAR,  
8 Year INT  
9 );  
10  
11 alter table review_dana alter column user_name type VARCHAR (300);  
12  
13 alter table review_dana alter column content type VARCHAR (1000);  
14  
15 alter table review_dana alter column sentimen type VARCHAR (300);
```

Data Output Messages Notifications

| | user_name character varying (300) | score integer | content character varying (1000) |
|---|--------------------------------------|------------------|--|
| 1 | Elisya Kasni | 5 | Bagus |
| 2 | Rusman Man | 2 | Dana mmg keren mantap. |
| 3 | Qiliw Sadega | 1 | Saya ngajuin upgrade dana premium krna ktp saya buram jdi ga bisa verifikasi .trus coba lewat email di suruh nunggu 3 hri .ini udh 3 hri lebih |
| 4 | Kijutrv2 Kijut | 3 | Kocak mana diskon nya ml malah eror segala kaga ikhlas ngasih diskon nya |
| 5 | Fifi Alfiyah | 1 | Saldo hilang karena no lama Hilang ganti no saldonya gk ada sama sekali dana tidak bertanggung jawab Buktinya saldo saya gk ada !!! Saldo |
| 6 | Kiki57 | 3 | mayan |
| 7 | Parhan Parhan | 1 | Udah gua hapus dana ya. ilang ya udah 1 juta lebih duwit gua.mf dana sayah buka orang kaya itu uang keringat sayah uang halal |
| 8 | Dewi Anggreni | 5 | baik |

Total rows: 1000 of 50000 Query complete 00:00:00.163 Ln 20, Col 1

QUERIES

1. Viewing all data
 - a. This query is to view all data.
 - b. The data contains 50000 rows

The screenshot shows the pgAdmin 4 interface with a query window titled 'review_dana_query.sql'. The query contains two SQL statements:

```
--View all data
select
*
from review_dana;

--Counting how many reviewers in total
select
count(*) as count_user
```

The 'Data Output' tab is active, displaying a table with 14 rows and 3 columns: 'user_name' (character varying (300)), 'score' (integer), and 'content' (character varying (1000)). The table contains user reviews and their scores.

| | user_name character varying (300) | score integer | content character varying (1000) |
|----|--------------------------------------|------------------|---|
| 1 | Elisya Kasni | 5 | Bagus |
| 2 | Rusman Man | 2 | Dana ming keren mantap. |
| 3 | Qiliw Sadega | 1 | Saya ngajuin upgrade dana premium kna ktp saya buram jdi ga bisa verifikasi.. trus coba lewat email di suruh nunggu 3 hri .ini udh 3 hri lebih msih gitu* doang s |
| 4 | Kijutjr2 Kijut | 3 | Kocak mana diskon nya ml malah eror segala kaga ikhlas ngasih diskon nya |
| 5 | Fifi Alfiyah | 1 | Saldo hilang karena no lama Hilang ganti no saldonya gk ada sama sekali dana tidak bertanggung jawab Buktinya saldo saya gk ada !!! Saldo lama saya 1800.00 |
| 6 | Kiki57 | 3 | mayan |
| 7 | Parhan Parhan | 1 | Udah gua hapus dana ya. ilang ya udah 1 juta lebih duwit gua.mf dana sayaah buka orang kaya itu uang keringat sayah uang halal |
| 8 | Dewi Anggreni | 5 | baik |
| 9 | Bang Ewok13 | 1 | TOLONG UNTUK SISTEM KEAMANAN DI PERBAIKI. KALAU 1 ATAU 2 ORNG YG KE HACK WAJAR, MUNGKIN KETELEDORAN PENGGUNA. TAPI KALAU SUDAH B |
| 10 | M Alifian | 5 | mempermudah transfer |
| 11 | DIDIK QU | 5 | baru yobain semoga aman sentosa tidak pembobolan apk sebelah |
| 12 | Liaz Tahajudin | 4 | kok gx bisa keridit |
| 13 | Chinju Nuriya | 5 | sangat puas |
| 14 | Miliah 343 | 5 | sangat memuaskan |

At the bottom of the interface, it shows 'Total rows: 1000 of 50000' and 'Query complete 00:00:00.161'.

QUERIES

2. Counting how many reviewers in total

The screenshot shows a database query editor interface. The top bar includes tabs for 'Dashboard', 'Processes', and 'review_dana_query.sql'. Below the tabs, the connection is 'review_dana/postgres@Local'. The query editor shows a SQL query to count reviewers in total. The query is as follows:

```
--Counting how many reviewers in total
select
count(*) as count_user
from
review_dana;
```

The query is highlighted in blue. Below the query editor, there are tabs for 'Query', 'Query History', 'Data Output', 'Messages', and 'Notifications'. The 'Data Output' tab is selected, showing the results of the query in a table format.

| | count_user bigint |
|---|----------------------|
| 1 | 50000 |

3. Counting how many user_name in every year

The screenshot shows a database query editor interface. The top bar includes tabs for 'Dashboard', 'Processes', and 'review_dana_query.sql'. Below the tabs, the connection is 'review_dana/postgres@Local'. The query editor shows a SQL query to count user_name in every year. The query is as follows:

```
--Counting how many user_name in every year
select
year,
count(*) as count_users
from
review_dana
group by 1;
```

The query is highlighted in blue. Below the query editor, there are tabs for 'Query', 'Query History', 'Data Output', 'Messages', and 'Notifications'. The 'Data Output' tab is selected, showing the results of the query in a table format.

| | year integer | count_users bigint |
|---|-----------------|-----------------------|
| 1 | 2024 | 49775 |
| 2 | 2023 | 225 |

QUERIES

4. Finding for score percentage in 2024

Dashboard × Processes × review_dana_query.sql ×

review_dana/postgres@Local

No limit

Query Query History

```
38 --Finding for score percentage in 2024
39 select
40 score,
41 count(*) as count_users,
42 round(100*count(*)/sum(count(*) over(),2) || '%' as percentage
43 from
44 review_dana
45 where year=2024
46 group by 1
47 order by 1 desc;
48
```

Data Output Messages Notifications

| | score integer | count_users bigint | percentage text |
|---|------------------|-----------------------|--------------------|
| 1 | 5 | 28921 | 58.10% |
| 2 | 4 | 4026 | 8.09% |
| 3 | 3 | 2564 | 5.15% |
| 4 | 2 | 2041 | 4.10% |
| 5 | 1 | 12223 | 24.56% |

5. Finding how many user and percentage in every sentimen and every year

Dashboard × Processes × review_dana_query.sql ×

review_dana/postgres@Local

No limit

Query Query History


```
49 --Finding how many user and percentage in every sentimen and every year
50 select
51 year,
52 sentimen,
53 count(*) as count_users,
54 round(100*count(*)/sum(count(*) over(),2) || '%' as percentage
55 from
56 review_dana
57 group by 1,2
58 order by 1,3 asc;
```

Data Output Messages Notifications

| | year integer | sentimen character varying (300) | count_users bigint | percentage text |
|---|-----------------|-------------------------------------|-----------------------|--------------------|
| 1 | 2023 | NEUTRAL | 32 | 0.06% |
| 2 | 2023 | NEGATIVE | 75 | 0.15% |
| 3 | 2023 | POSITIVE | 118 | 0.24% |
| 4 | 2024 | NEUTRAL | 6340 | 12.68% |
| 5 | 2024 | NEGATIVE | 16998 | 34.00% |
| 6 | 2024 | POSITIVE | 26437 | 52.87% |




QUERIES THAT IMPLEMENTED

1. **SELECT** : retrieve data from table
 2. **COUNT** : aggregate function to get number of rows with specific condition
 3. **ROUND** : round numeric value
 4. **OVER** : execute aggregate inside aggregate queries
 5. **ORDER BY** : sort data ascending/descending
 6. **GROUP BY** : aggregate same values into summary rows
- 



CONCLUSION

1. Data can be analyzed using query SQL, the data contains 50.000 rows
 2. There are 225 reviewers in 2023, 49.775 reviewers in 2024
 3. There are more than half of the total reviewers give 5 score rating (58.1%) in 2024, and overall give positive sentiment (52.87%)
- 

The slide features a light gray background with decorative elements in the corners. The top-left corner has a gray triangle and four horizontal lines of varying lengths, each ending in a small circle. The top-right corner has a blue triangle. The bottom-left corner has a blue triangle and a gray triangle. The text "THANK YOU" is centered in a bold, dark blue font.

THANK YOU