SOCIAL PLATFORM ANALYSIS

Project for SQL module

By Abhishek Jagtap

Project Description:

The "Social Platform Analysis" project aims to provide comprehensive insights into user behaviour, engagement, and content interaction on a social media platform. By analysing various aspects such as user demographics, content popularity, reaction types, and session activities, this project seeks to uncover valuable patterns and trends to inform decision-making and enhance the user experience.

Tables:

Content Table: Contains information about the content shared on the social platform, including content ID, category, user ID, URL, and type (e.g., photo, video, text).

Users Table: Stores user profiles with details such as user ID, username and email.

Reactions Table: Records user reactions to content, including the user ID, content ID, reaction type, and datetime (on which date and timing it was posted).

Reactiontypes Table: Defines the types of reactions available on the platform, such as sentiment (positive and negative), Type of reaction along with their corresponding scores or weights.

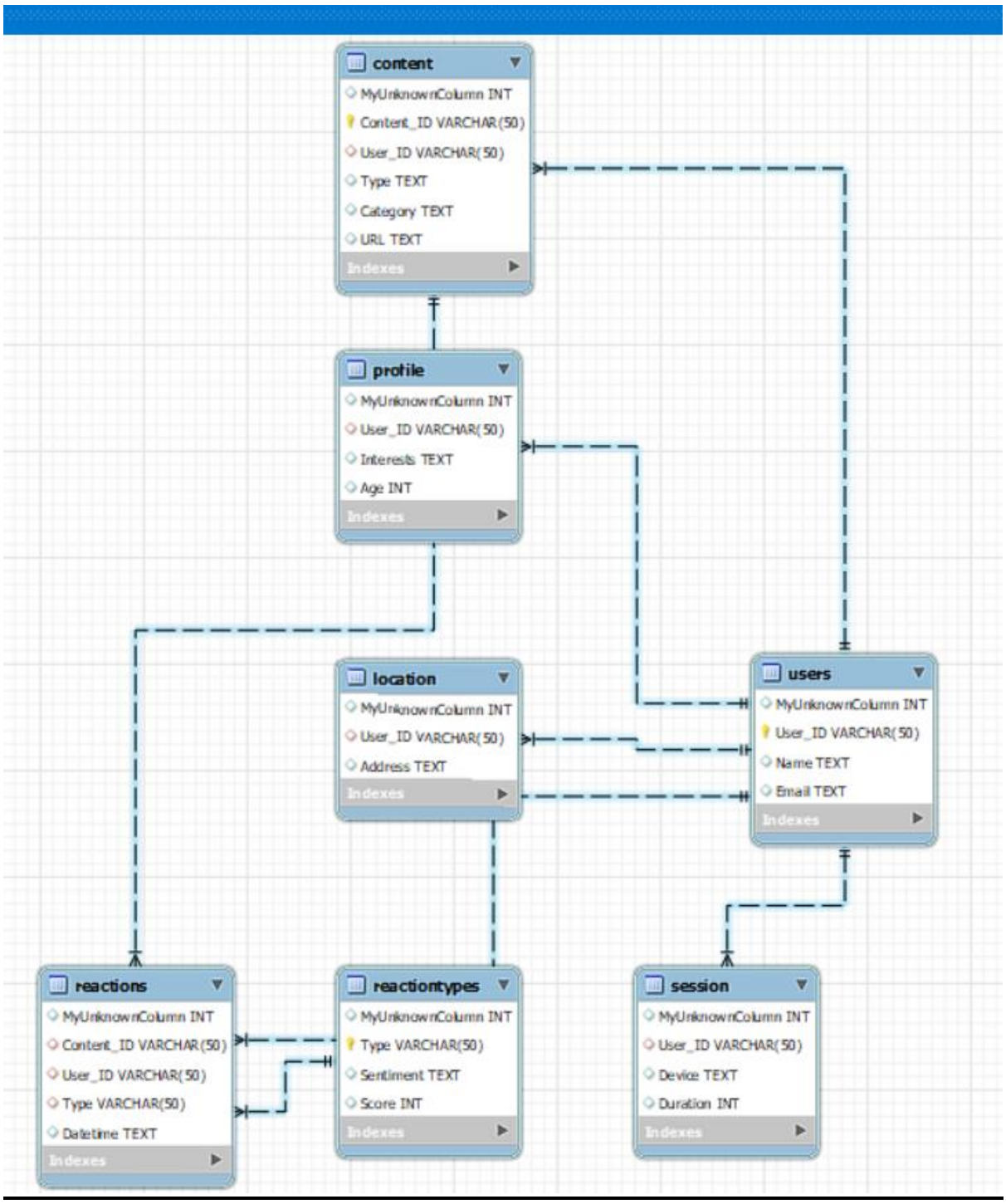
Profile Table: Contains additional profile information for users, such as age, interests and User id.

Session Table: Tracks user sessions on the platform, including user ID, user device and session duration.

Location Table: Stores geographic location data, such as,city, country, and any other relevant location details.

EER diagram / Table schema:

This shows how the tables relate to primary and foreign keys based on the common columns / attributes.



Commands:

Tables are imported through import wizard in MySQL workbench.

Following commands are used to define the table structure:

- Assign Primary key constraints -

  -- assiging primary key in table users

  alter table users modify User_ID varchar(50)primary key;

  -- assiging primary key in table content

  alter table content modify Content_ID varchar(50)primary key;

- Modifying the data type –

  -- modify the datatype from text to varchar for table content
  alter table content modify User_ID varchar(50);

  -- modify the datatype from text to varchar for table profile
  alter table profile modify User_ID varchar(50) ;

  -- modify the datatype from text to varchar for table session
  alter table session modify User_ID varchar(50) ;

  -- modify the datatype from text to varchar for table reactions
  alter table reactions modify User_ID varchar(50) ;

  -- modify the datatype from text to varchar for table location
  alter table location modify User_ID varchar(50) ;

  -- modify the datatype from text to varchar for table reactiontypes
  alter table reactiontypes modify Type varchar(50)primary key;

  -- modify the datatype from text to varchar for table reactions
  alter table reactions modify Content_ID varchar(50) ;

  -- modify the datatype from text to varchar for table reactions
  alter table reactions modify Type varchar(50);

- Assign foreign key to relate the tables –

```
-- connecting users to content on user_id
alter table content add constraint contents foreign key (User_ID)
references users(User_ID);

 -- connecting users to location on user_id
alter table location add constraint locations foreign key (User_ID)
references users(User_ID);

-- connecting users to profile on user_id
alter table profile add constraint profiles foreign key (User_ID) references
users(User_ID);

-- connecting users to session on user_id
alter table session add constraint sessions foreign key (User_ID)
references users(User_ID);

-- connecting reactions to content on Content_ID
alter table reactions add constraint reactionss foreign key (Content_ID)
references content(Content_ID);

-- connecting reactions to reactiontypes on Type
alter table reactions add constraint reactionsss foreign key (Type)
references reactiontypes(Type);
```

- Following command disables checking for foreign key constraints in MySQL –

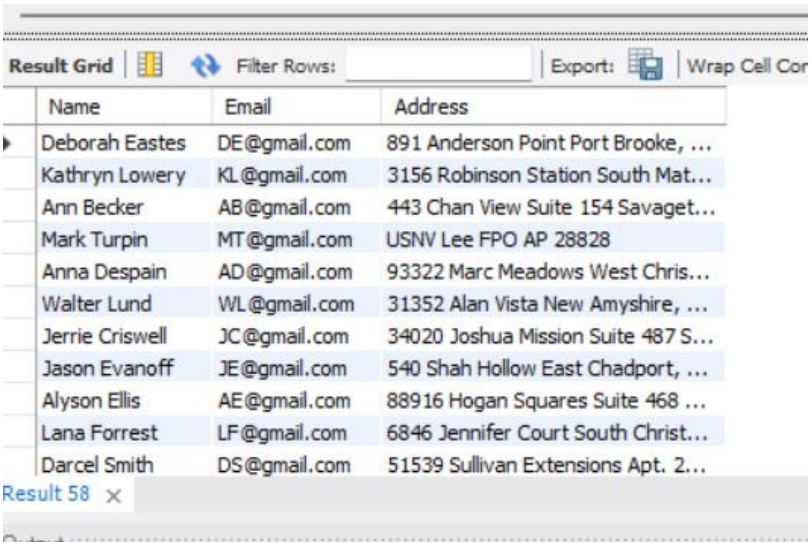
```
SET FOREIGN_KEY_CHECKS=0;
```

Operations Performed:

1. Identifying the users location –
   Query –

```
SELECT
   users.Name, users.Email, location.Address
FROM
   users
      LEFT JOIN
   location ON users.User_ID = location.User_ID;
```

Results-



Result Grid | Filter Rows: | Export: | Wrap Cell Cor

Name	Email	Address
Deborah Eastes	DE@gmail.com	891 Anderson Point Port Brooke, ...
Kathryn Lowery	KL@gmail.com	3156 Robinson Station South Mat...
Ann Becker	AB@gmail.com	443 Chan View Suite 154 Savaget...
Mark Turpin	MT@gmail.com	USNV Lee FPO AP 28828
Anna Despain	AD@gmail.com	93322 Marc Meadows West Chris...
Walter Lund	WL@gmail.com	31352 Alan Vista New Amyshire, ...
Jerrie Criswell	JC@gmail.com	34020 Joshua Mission Suite 487 S...
Jason Evanoff	JE@gmail.com	540 Shah Hollow East Chadport, ...
Alyson Ellis	AE@gmail.com	88916 Hogan Squares Suite 468 ...
Lana Forrest	LF@gmail.com	6846 Jennifer Court South Christ...
Darcel Smith	DS@gmail.com	51539 Sullivan Extensions Apt. 2...

Result 58

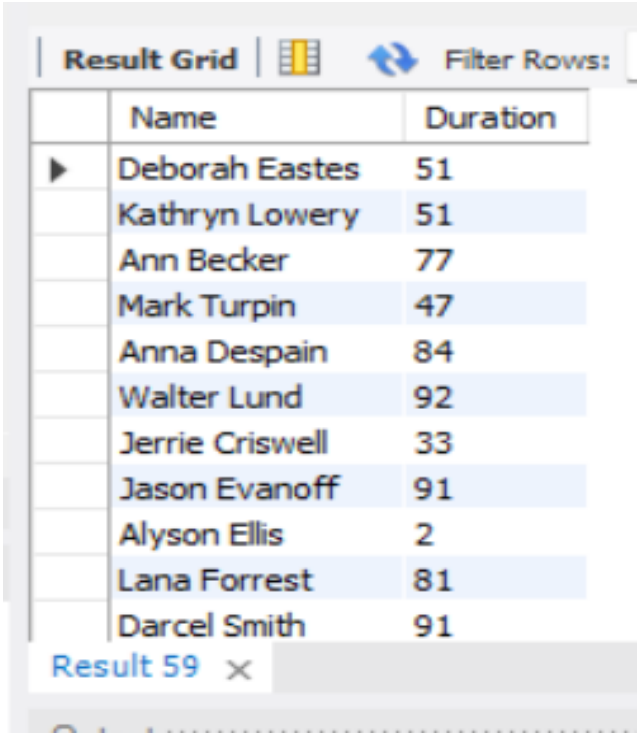
2. Time spent by users –

   Query-

```
SELECT
   users.Name, session.Duration
FROM
   users
      INNER JOIN
```

session ON users.User_ID = session.User_ID;

Results-



Result Grid | Filter Rows:

	Name	Duration
▶	Deborah Eastes	51
	Kathryn Lowery	51
	Ann Becker	77
	Mark Turpin	47
	Anna Despain	84
	Walter Lund	92
	Jerrie Criswell	33
	Jason Evanoff	91
	Alyson Ellis	2
	Lana Forrest	81
	Darcel Smith	91

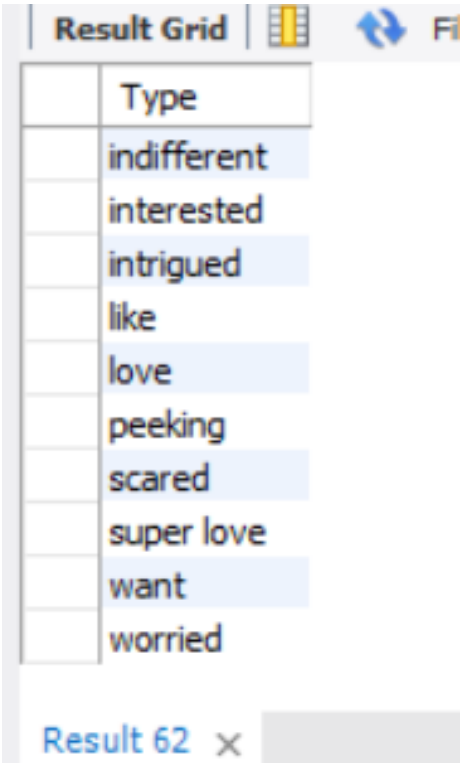
Result 59

3. Types of user reactions-

Query-

```
SELECT DISTINCT
    (reactiontypes.Type)
FROM
    reactions
        LEFT JOIN
    reactiontypes ON reactions.Type = reactiontypes.Type;
```

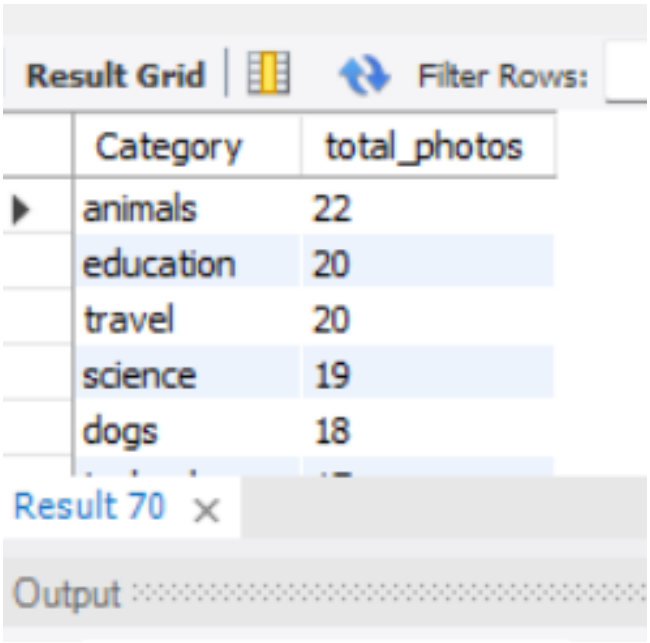
Results –



4. Highest category posted for content type PHOTOS –
Query –

```
SELECT
   Category, COUNT(Type) AS total_photos
FROM
   content
WHERE
   Type IN ('photo')
GROUP BY Category
ORDER BY 2 DESC;
```

Results-

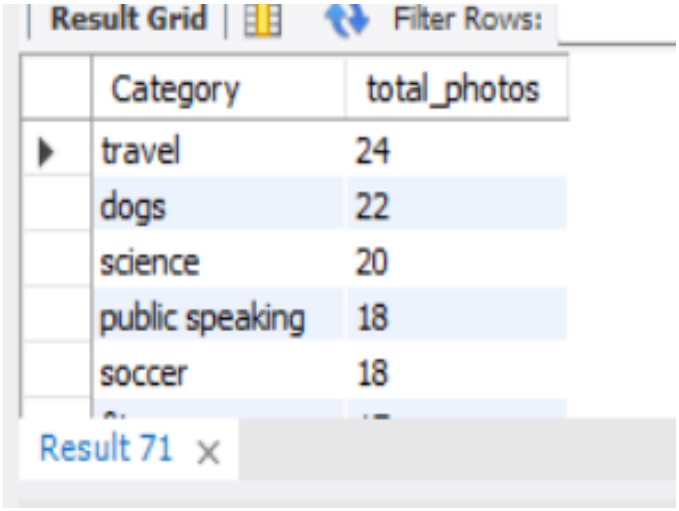


5. Highest category posted for content type VIDEOS –

Query-

```
SELECT
    Category, COUNT(Type) AS total_photos
FROM
    content
WHERE
    Type IN ('Video')
GROUP BY Category
ORDER BY 2 DESC;
```

Results –

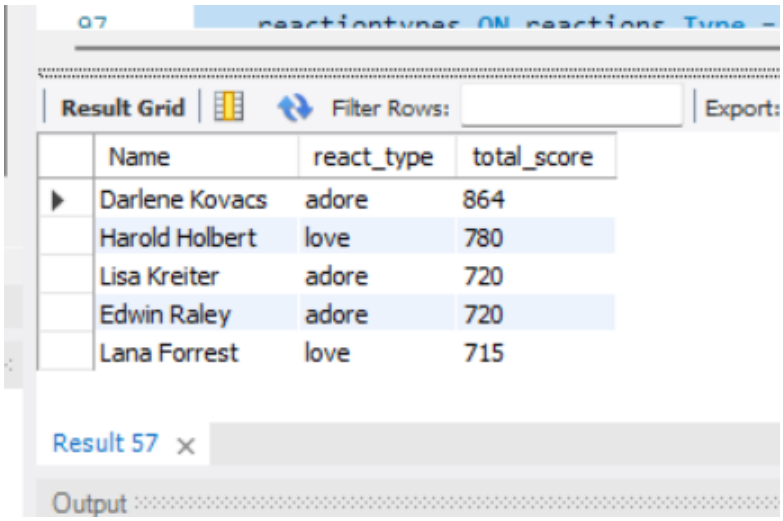


6. Maximum reaction score for a user –

Query -

```
SELECT
    users.Name,
    reactiontypes.Type AS react_type,
    SUM(reactiontypes.score) AS total_score
FROM
    users
        INNER JOIN
    reactions ON users.User_ID = reactions.User_ID
        INNER JOIN
    reactiontypes ON reactions.Type = reactiontypes.Type
GROUP BY users.Name , reactiontypes.Type
order by total_score desc limit 5;
```

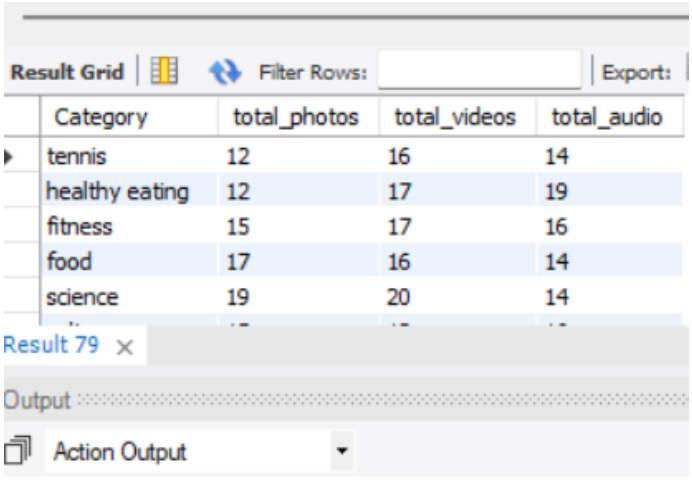
Results –



7. Types and category wise distribution-
Query-

```
SELECT
    Category,
    SUM(Type = 'photo') AS total_photos,
    SUM(Type = 'video') AS total_videos,
    SUM(Type = 'audio') AS total_audio
FROM
    content
GROUP BY Category
```

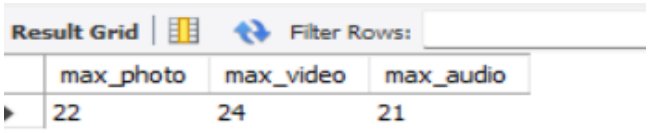
Results-



Query for max Value for each type –

```
SELECT
   MAX(total_photos) AS max_photo,
   MAX(total_videos) AS max_video,
   MAX(total_audio) AS max_audio
FROM
   (SELECT
      Category,
         SUM(Type = 'photo') AS total_photos,
         SUM(Type = 'video') AS total_videos,
         SUM(Type = 'audio') AS total_audio
   FROM
      content
   GROUP BY Category) AS max_value_each;
```

Results –



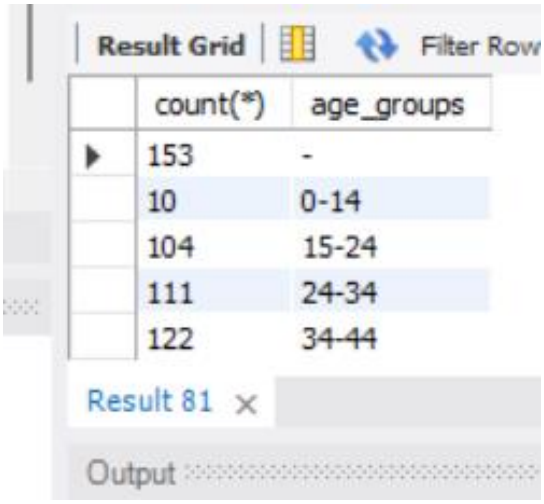
8. Finding what age group people are on this platform-

Query-

```
SELECT
   COUNT(*),
   CASE
      WHEN age = 0 AND age <= 14 THEN '0-14'
      WHEN age > 14 AND age <= 24 THEN '15-24'
      WHEN age > 24 AND age <= 34 THEN '24-34'
      WHEN age > 34 AND age <= 44 THEN '34-44'
```

```
        ELSE '-'
    END AS age_groups
FROM
    profile
GROUP BY 2
ORDER BY 2;
```

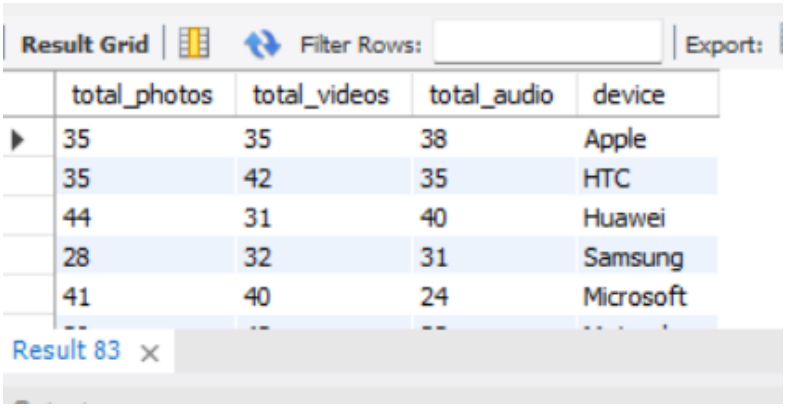
Results-



9. Classifying total values of content types for each device - Query-

```
SELECT
    SUM(Type = 'photo') AS total_photos,
    SUM(Type = 'video') AS total_videos,
    SUM(Type = 'audio') AS total_audio,
    session.device
FROM
    content
        INNER JOIN
    session ON content.User_ID = session.User_ID
GROUP BY session.device;
```

Results -



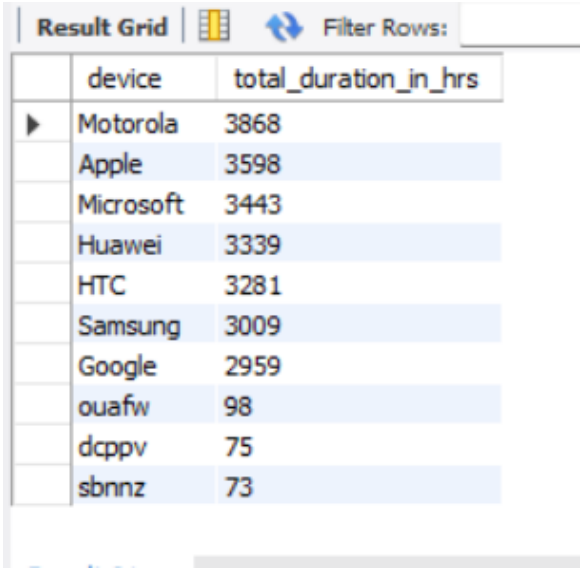
total_photos	total_videos	total_audio	device
35	35	38	Apple
35	42	35	HTC
44	31	40	Huawei
28	32	31	Samsung
41	40	24	Microsoft

10. Top 10 devices with respect to duration use-

Query-

```
SELECT
   device, SUM(duration) AS total_duration_in_hrs
FROM
   session
GROUP BY 1
ORDER BY total_duration_in_hrs DESC
LIMIT 10;
```

Results-



Result Grid | Filter Rows:

device	total_duration_in_hrs
Motorola	3868
Apple	3598
Microsoft	3443
Huawei	3339
HTC	3281
Samsung	3009
Google	2959
ouafw	98
dcppv	75
sbnnz	73

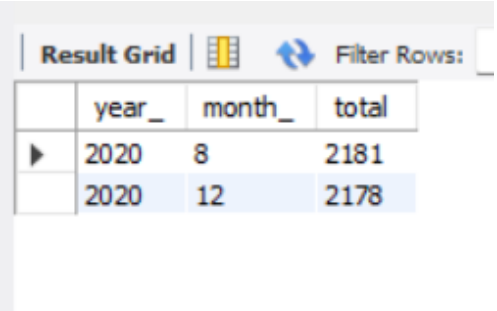
11. At what time of the year there is more posting / traffic on platform:
Query for all year present in the data:

```
SELECT
   YEAR(datetime) AS year_,
   MONTH(datetime) AS month_,
   COUNT(*) AS total
FROM
   reactions
GROUP BY 1 , 2
ORDER BY 3 DESC;
```

Query for the year 2020 –

```
SELECT
   YEAR(datetime) AS year_,
   MONTH(datetime) AS month_,
   COUNT(*) AS total
FROM
   reactions
WHERE
   YEAR(datetime) = 2020
GROUP BY 1 , 2
ORDER BY 3 DESC
LIMIT 2;
```

Results-

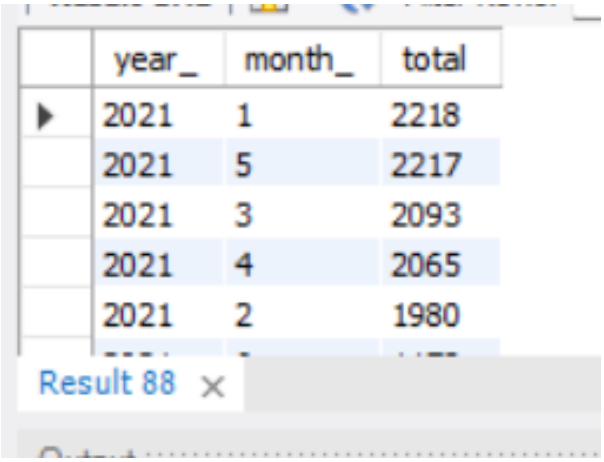


year_	month_	total
2020	8	2181
2020	12	2178

Query for the year 2021 –

```
SELECT
   YEAR(datetime) AS year_,
   MONTH(datetime) AS month_,
   COUNT(*) AS total
FROM
   reactions
WHERE
   YEAR(datetime) = 2021
GROUP BY 1 , 2
ORDER BY 3 DESC;
```

Results-



year_	month_	total
2021	1	2218
2021	5	2217
2021	3	2093
2021	4	2065
2021	2	1980

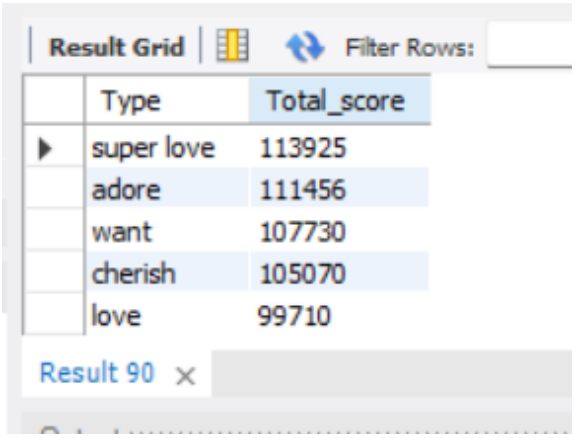
Result 88

12. Identifying top 5 categories/Reaction Types –

Query –

```
SELECT
   Type, MAX(total_type * score) AS Total_score
FROM
   (SELECT
      COUNT(*) AS total_type,
         reactions.Type,
         reactiontypes.score AS score
   FROM
      reactions
   INNER JOIN reactiontypes ON reactions.Type = reactiontypes.Type
   GROUP BY reactions.Type) AS total
GROUP BY 1
ORDER BY 2 DESC
LIMIT 5;
```

Results-



CONSLUSION-

The analysis of user behaviour and content interaction on the social platform provides valuable insights for enhancing user experience and platform performance. By examining user location data, we can better understand where our users are located, allowing us to tailor content and features to their preferences. Understanding the time users spend on the platform helps us gauge engagement levels and optimize platform usability. Identifying the types of reactions users have towards content gives us insight into user sentiment and preferences. Additionally, analysing the most popular categories for photo and video content helps us pinpoint trending topics and themes. By determining the maximum reaction score for a user, we can highlight the most positively received interactions. Examining the distribution of reactions and content categories provides a comprehensive overview of user engagement patterns. Classifying users into age groups reveals the platform's demographic composition, informing targeted marketing strategies. Understanding content consumption across different devices aids in optimizing platform compatibility and user experience. Identifying the devices with the longest usage durations informs device-specific optimization strategies. Analysing posting and traffic trends throughout the year helps us identify peak activity periods and optimize content scheduling. Overall, these insights enable us to make informed decisions to enhance user satisfaction, increase engagement, and drive the sustainable growth of the platform.