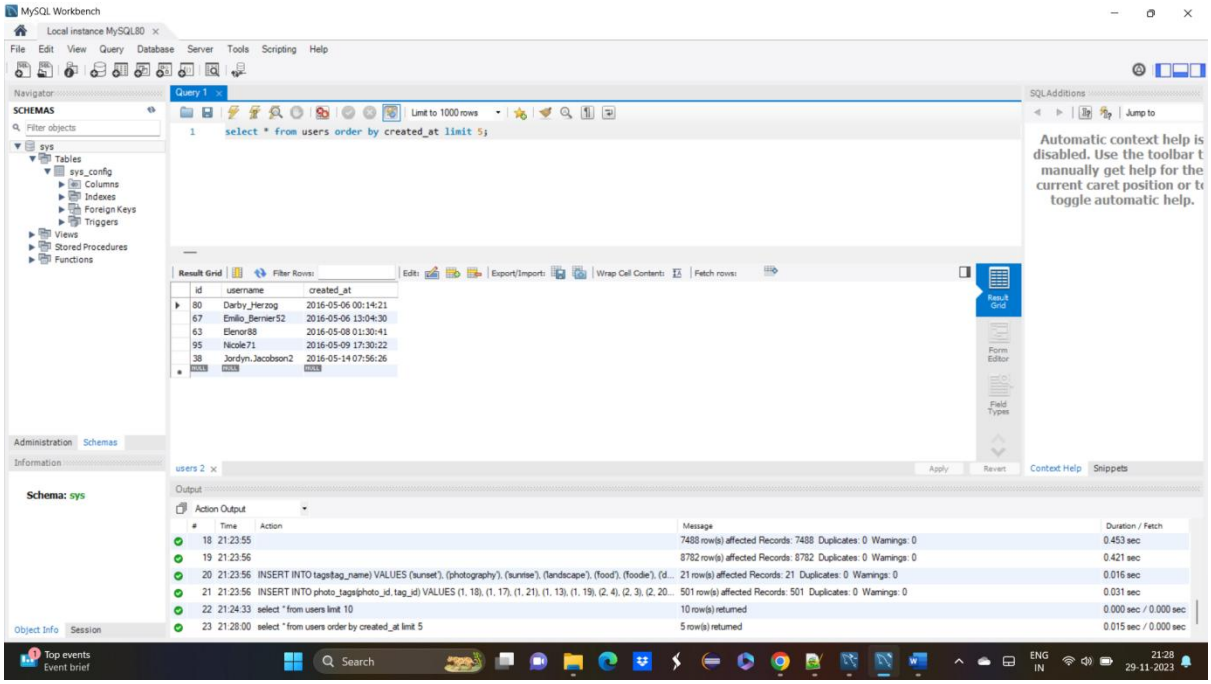
Instagram User Analytics

A) Marketing Analysis:

1.Loyal User Reward: Identify the five oldest users on Instagram from the provided database.



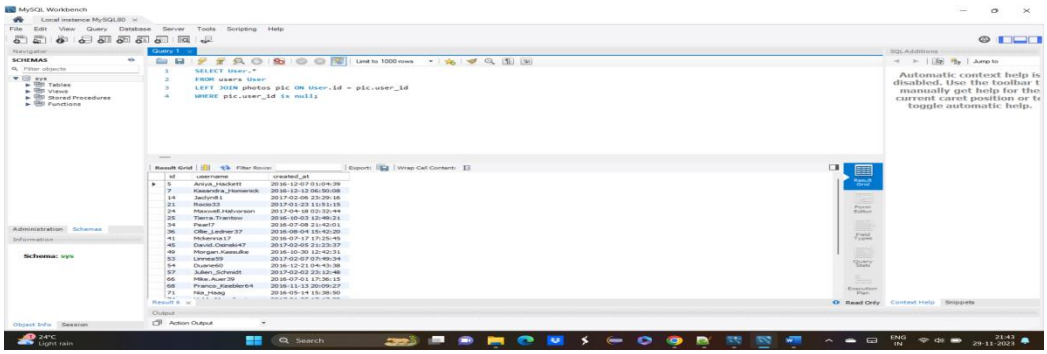
Query used

select * from users order by created_at limit 5;

Explanation

This query is used to get the users data of 5 oldest customers in users table with the use of column value created_at having inculcated order by function (if not specified, its ascending order – which means from oldest to latest date). The limit by 5 is used to get the 5 customers alone in this situation.

2. Inactive User Engagement: Identify users who have never posted a single photo on Instagram.



Query used

SELECT User.*

FROM users User

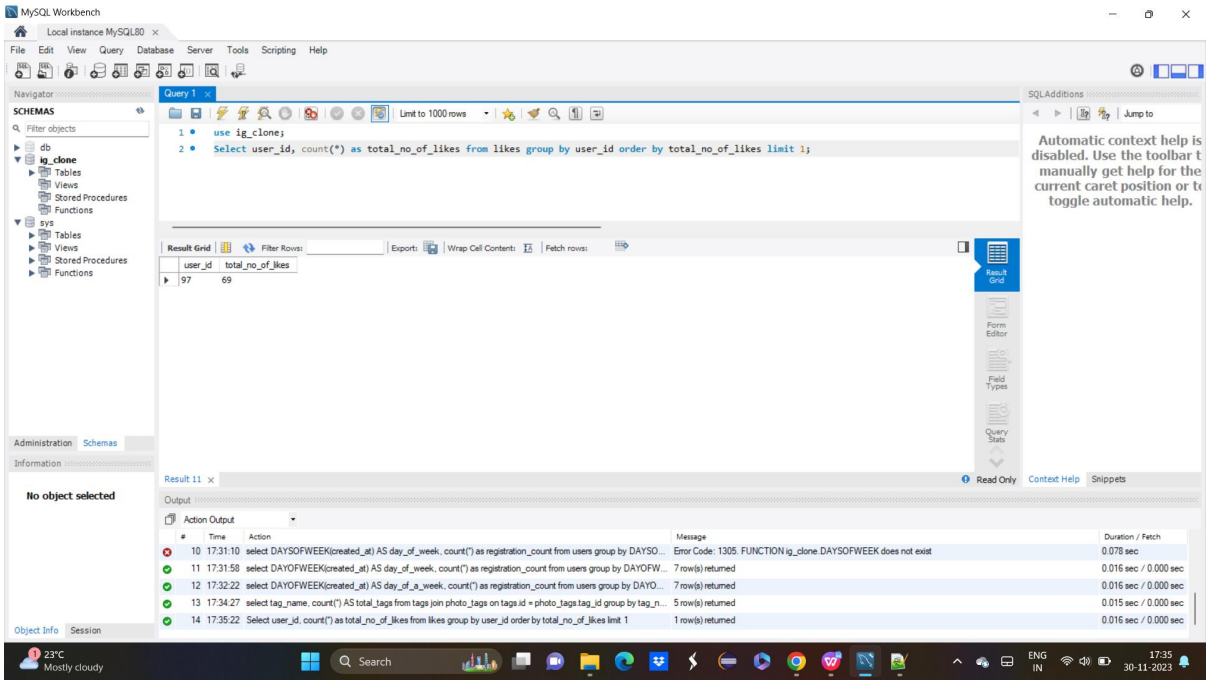
LEFT JOIN photos pic ON User.id = pic.user_id

WHERE pic.user_id is null;

Explanation

This query is used to get the users data compared with photos table to get the results of people in the database who have not posted a single photo on Instagram. The join query is used to integrate the user table and photos and the comparison factor which is used to check the above required condition is – first, user id of user table is equal to the user id of photo table and the second being photo's user id being null. This query will give us 26 rows under this condition.

3.Contest Winner Declaration: Determine the winner of the contest and provide their details to the team.



Query used

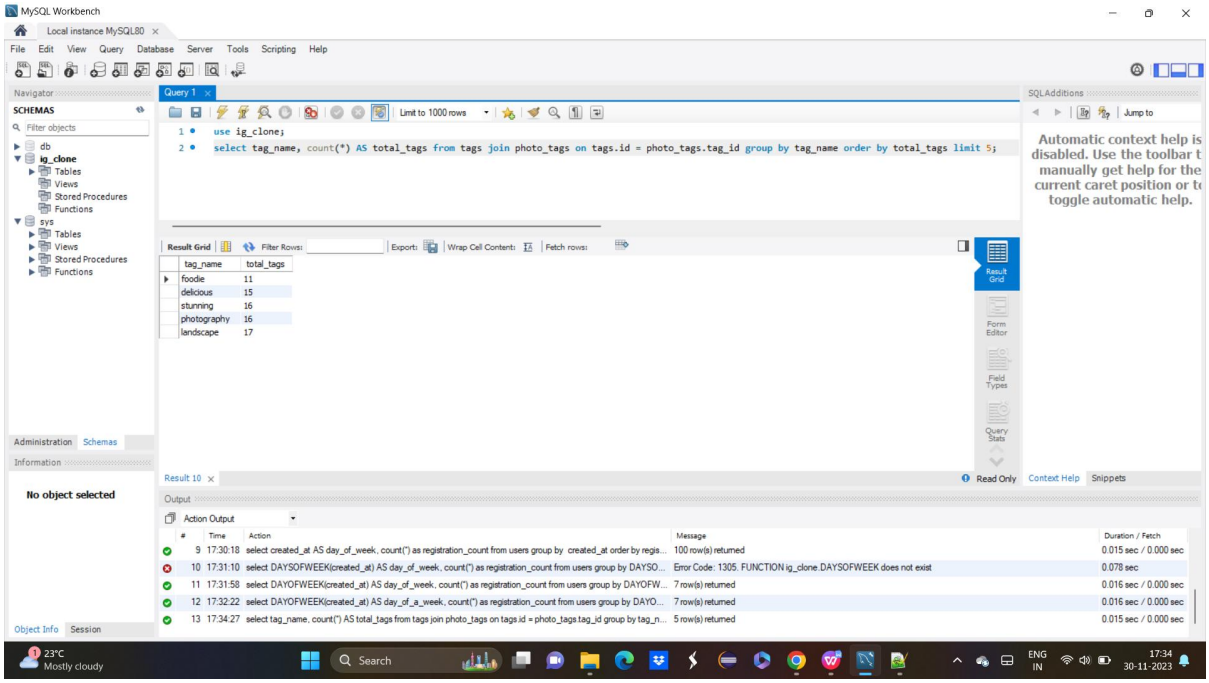
Select user_id, count(*) as total_no_of_likes from likes group by user_id order by total_no_of_likes limit 1;

Explanation

This query is used to get the count of likes from the likes table with an alias name total_likes alongside user_id from user table to identify which user id has got the maximum likes. This query uses count function to count the no of rows in likes table, group by function is to group

the count of the likes table with user id and order by is used to order the alias name total_likes in the descending order and finally limit function to limit the no of occurrences of a output.

4.Hashtag Research: Identify and suggest the top five most commonly used hashtags on the platform.



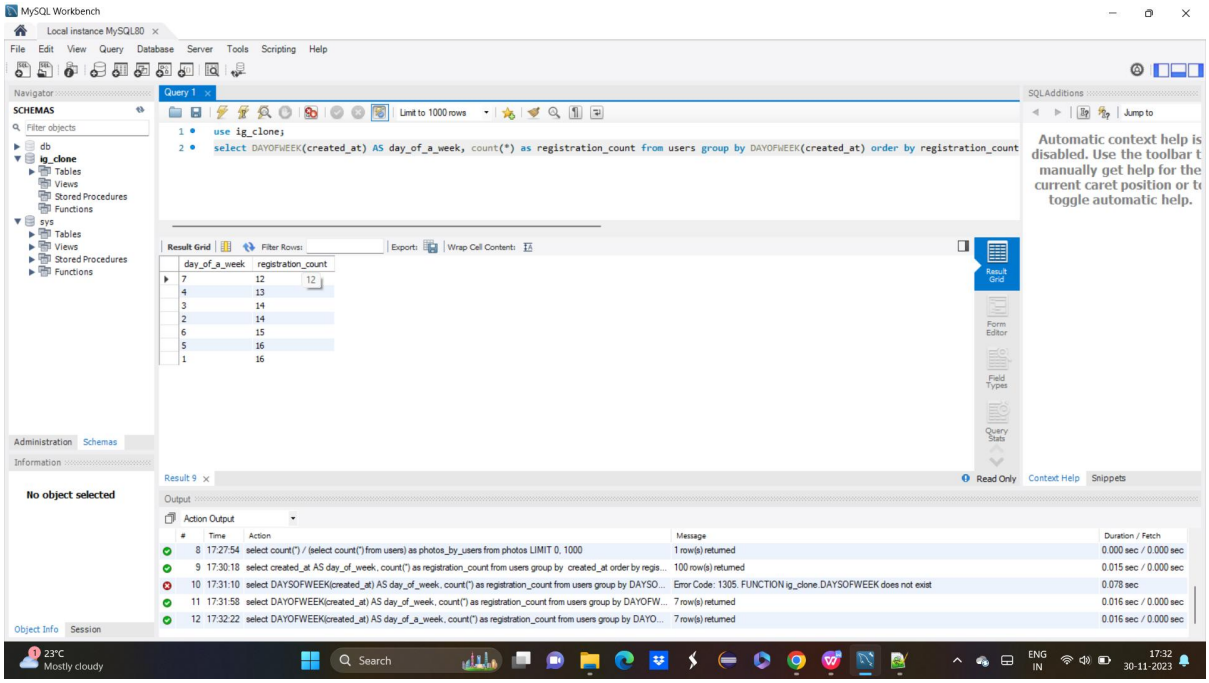
Query used

select tag_name, count(*) AS total_tags from tags join photo_tags on tags.id = photo_tags.tag_id group by tag_name order by total_tags limit 5;

Explanation

This query is used to get the tag name, count the no of rows with alias name tag_count from the tag table which is to be merged with photo_tags table by scrutinizing the output by comparing the tag id of tag table and tag_id from photo_tags table which then is grouped by tag_name and the order is defined by the count in descending order to limit only the top 5 users. The count function is to get the total tag count from the tag table, join query to combine it with photo_tags table, group by to group the answer of the query to tag_name, order by tag_count to find out each no of tag_names used represented in descending order and finally limit the output to 5 since the top 5 is the requirement.

5. Ad Campaign Launch: Determine the day of the week when most users register on Instagram. Provide insights on when to schedule an ad campaign.



Query used

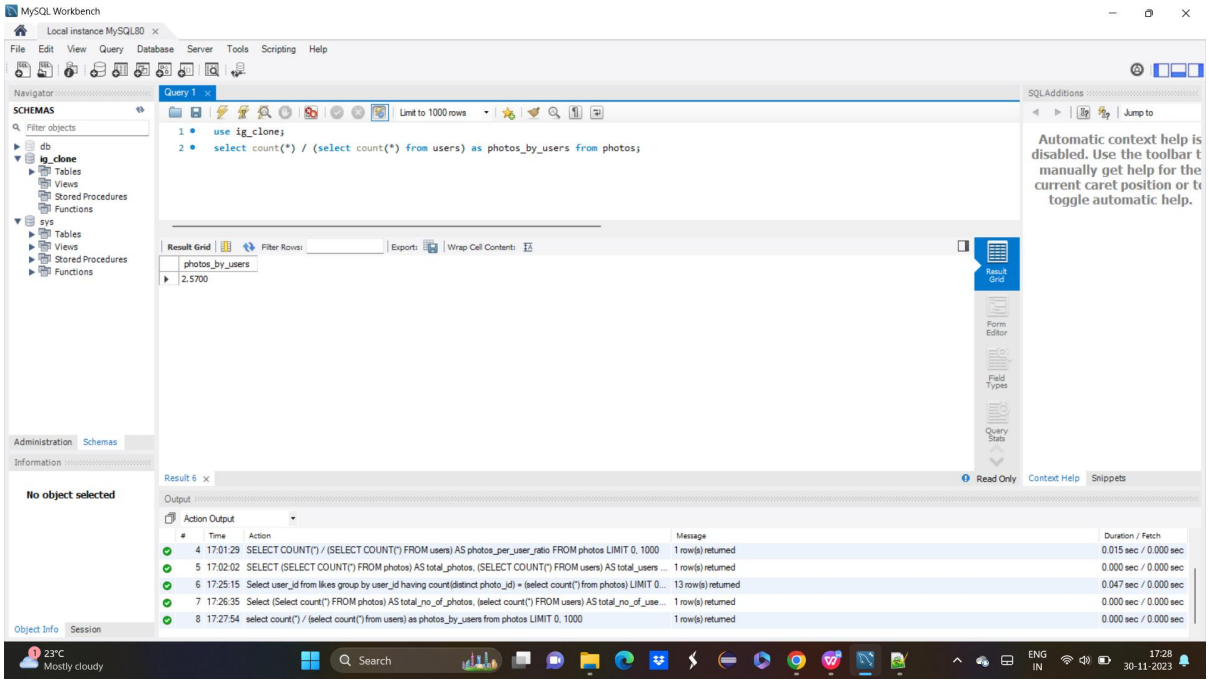
select created_at AS day_of_a_week, count(*) as registration_count from users group by created_at order by registration_count;

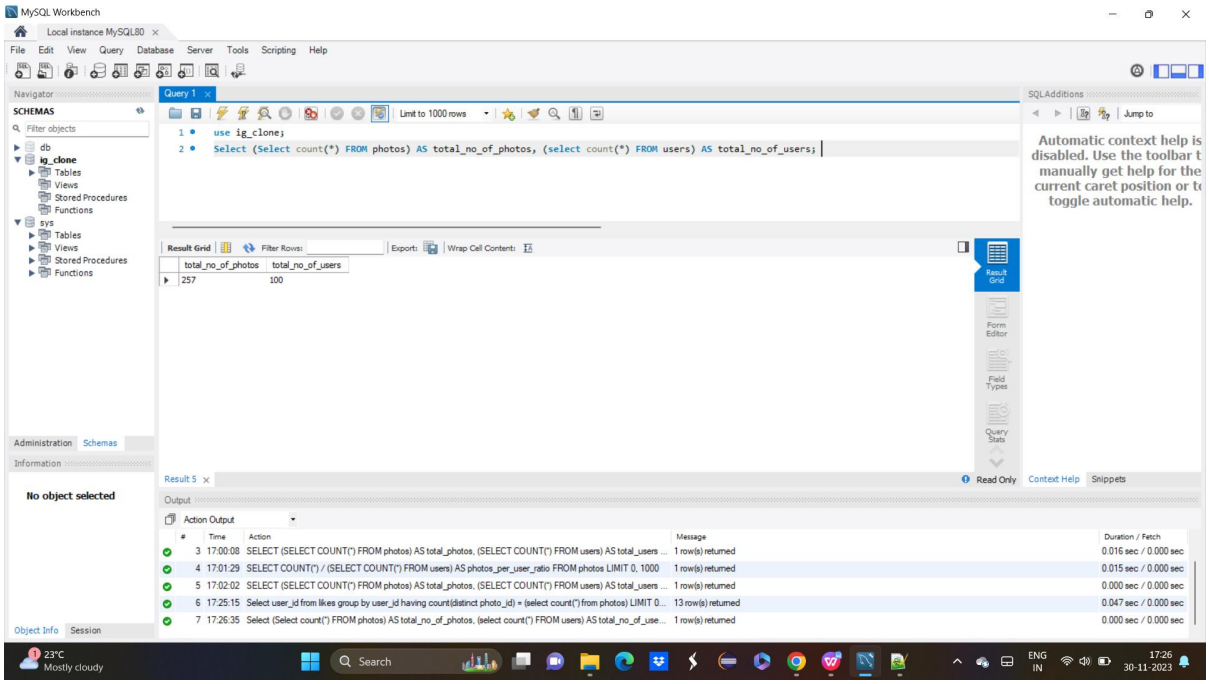
Explanation

This query is used to get the days of the week in which the registration count is maximum from the users table with grouped by days of week and ordered according to the registration count in descending order. This query displayes created at column with an alias name as days_of_week takes the total number of registration through count function on registration_count column from the users table with grouping the output to days_of_week and ordering the final output in the structure of registration count in descending format.

B) Investor Metrics:

1. User Engagement: Calculate the average number of posts per user on Instagram. Also, provide the total number of photos on Instagram divided by the total number of users.





Query used

Select (Select count(*) FROM photos) AS total_no_of_photos, (select count(*) FROM users) AS total_no_of_users;

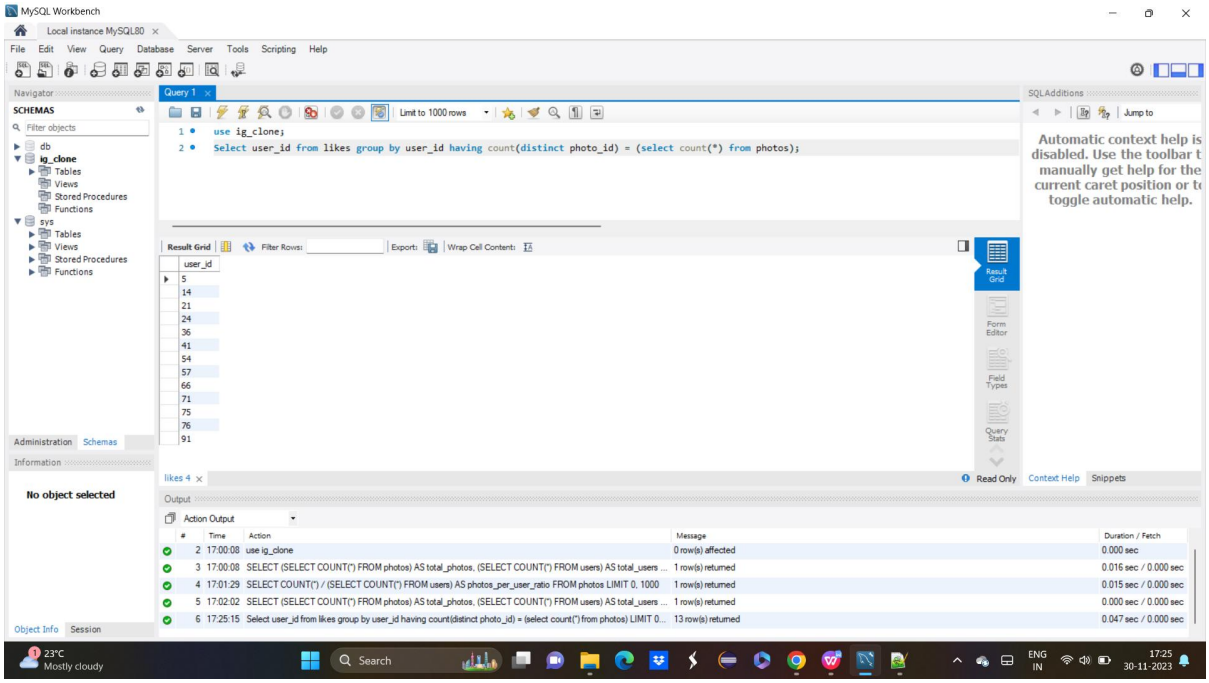
select count(*) / (select count(*) from users) as photos_by_users from photos;

Explanation

This query is used to display the average no of photos each user post on Instagram and also even displays the total number of photos in Instagram divided by users. Both of the query uses sub queries as if refers to different tables within the same query. The first query is used to first get the count of photos in the sub query enclosed within brackets as an alias name total_photos from the photos table and the second half of the first query is used to get the total count of users as total users from the users table with an alias name total_users.

The second query is used to divide the total count of the photos from photos table and the total number of users from the users table with sub query being used in the second half for referring the users table and both combined output is given the alias name as photos_per_user_ratio.

2.Bots & Fake Accounts: Identify users (potential bots) who have liked every single photo on the site, as this is not typically possible for a normal user.



Query used

Select user_id from likes group by user_id having count(distinct photo_id) = (select count(*) from photos);

Explanation

This query is used to get the person or a bot who has liked all of the photos on the site which is practically not possible for a human being with combination of photos and likes table. This query is first used to collect the user_id from the the likes table and group them separately inside a table. This is in turn then used for comparison with the photos table with taking the distinct or unique count of photo ids from the photo table. This query uses having term which is used to compare the

grouped column I.e; user_id with another column from the another table i.e; unique photos_id from photos table.

Project Description

This project is to gain a thorough knowledge on the extensive use of the SQL queries. Its main topics like sub query and join are extensively used in the project questions. Understanding of different types of command like ddl, dml, dql etc and use of them has been experienced through this project. This project provides an overview of collection of various kinds of data related to a day-to-day usage scenario - Instagram database like collection of oldest customers, contest and winners and various other features like photos, likes and comments have been used as table and combined together to get the required details. I have analyzed the table structures, columns and studied the various insert commands to see what data is inside each table.

Approach

The step by step approach for finding the answers to the above said scenario are as follows:-

1. First, I analyzed what all tables are given, what's each table purpose is, what column values each table hold and corresponding purpose and use, what primary and foreign keys are used in table.

2. Second, in order to combine the table I have analyzed how the primary and foreign key are used in each table and how to do comparison in order to obtain the desired result.

3. Thirdly, analyzing the question what data it is requiring of, what all table values it actually wants to display in the final output and how efficiently we can combine table through join or use sub queries to get the desired output.

4. Finally, using different approaches and seeing what results it is giving and trimming the queries to get the exact requirement satisfied.

Tech-Stack Used

The software used for this project is MySQL Workbench and the version is 8.0.35. The main aim of using this software for this project is due to the project requirement being implementation of various scenarios from the database through the use of SQL queries. This software is a open-source provided by Oracle which can be easily used to perform the above said requirement.

Insights

The following are the insights gained from the above project:-

1. In order to combine two tables, we should use join query and that query requires a primary key and foreign key comparison in the first and second table respectively with any of the conditions like equal, greater, lesser etc.

2. The sub query is used when we want to gather multiple information from more than one table and need it to be displayed in the final result as well.

3. In order to classify the data after grouping a particular column, having by clause is used to scrutinize the data with certain conditions placed on it and similarly to get the unique elements distinct command is used.

4. In order to get the days of week there is a function called dayofweek to be used on the date column to get the same and similarly to get the oldest customers the dates must be ordered in ascending order.

Result

Through this project I was able to understand how various tables are combined using different commands to get a desired pattern of data. This project also taught us the resilience and consistency of the data as the roles of the database administrator or data analyst in analyzing and extracting the useful data for classification or further use. I have also understood how a raw data looks like to a data analyst and upon applying certain principals or demands, how analyst can gather the required data and present it in turn to others in a more effective and efficient manner. Through this project, it was understood that by using SQL, a data pattern and insights can be gathered by putting up some basic requirements like the above stated scenarios.