```sql
create database SocialBuzz;

use SocialBuzz;

select * from Reactions;
select * from Content;
select * from ReactionTypes;

drop table Reactions

/*Data Cleaning*/
select * from Reactions;

/*removing unnecessary columns from Reactions*/
alter table Reactions
drop column column1, User_ID;

/*checking null values in columns*/
select * FROM Reactions
where Content_ID IS NULL OR Type IS NULL OR Datetime IS NULL;

/*delete the rows having null values*/
delete from Reactions
where Content_ID IS NULL OR Type IS NULL OR Datetime IS NULL;

-----------------------------------------------------------------

/*removing unnecessary columns from Content*/
alter table Content
drop column column1, User_ID, URL;

/*checking null values in columns*/
select * FROM Content
where Content_ID IS NULL OR Type IS NULL OR Datetime IS NULL;

/*Updating Category column with lowercase values*/
update Content
set Category=LOWER(Category);

/*Checking Category column*/
select * from Content where Category LIKE '%"%';

/*Updating Category column with correct values*/
update Content
set Category=REPLACE(Category,'"','')
where Category LIKE '%"%';

/*checking null values in columns*/
select * FROM Content
```

```sql
where Content_ID IS NULL OR Type IS NULL OR Category IS NULL;

-----------------------------------------------------------------------

/*removing unnecessary columns from ReactionType*/
alter table ReactionTypes
drop column column1;

/*checking null values in columns*/
select * FROM ReactionTypes
where Type IS NULL OR Sentiment IS NULL OR Score IS NULL;

-----------------------------------------------------------------------

/*Creating View*/
create VIEW Dataset AS
select r.Content_ID, r.Type AS Reaction_Type, c.Type AS Content_Type,
c.Category, t.Sentiment, t.Score, CAST(r.Datetime AS DATE) AS Date,
    CONVERT(TIME(0), r.Datetime) AS Time  from Reactions r
LEFT JOIN Content c
ON c.Content_ID=r.Content_ID
LEFT JOIN ReactionTypes t
ON t.Type=r.Type;

/*Selecting data from View*/
select * from Dataset;

-------------------------------------------------------------------------
/*SQL Queries for Report*/
/*KPI's*/
--Unique Categories
select COUNT(DISTINCT Category) AS [Unique Categories] from Dataset;

--Total Reactions
select SUM(Score) AS [Total Score] from Dataset;

--Total Reactions
select COUNT(Reaction_Type) AS [Total Reactions] from Dataset;

/*Visuals*/
/*Total Posts by Month and Year*/
select COUNT(Content_ID) AS [Total Posts], DATENAME(MONTH,Date) AS Month,
DATEPART(YEAR, Date) AS Year from Dataset
group by DATENAME(MONTH,Date), DATEPART(YEAR, Date), MONTH(Date)
order by MONTH(Date); /*ordering by Month number*/

/*Pivoting the Total Posts by Month and Year*/
select Year,
    COALESCE(January, '') AS January,
```

```sql
    COALESCE(February, '') AS February,
    COALESCE(March, '') AS March,
    COALESCE(April, '') AS April,
    COALESCE(May, '') AS May,
    COALESCE(June, '') AS June,
    COALESCE(July, '') AS July,
    COALESCE(August, '') AS August,
    COALESCE(September, '') AS September,
    COALESCE(October, '') AS October,
    COALESCE(November, '') AS November,
    COALESCE(December, '') AS December
    from
    (
        select COUNT(Content_ID) AS Total_Posts, DATENAME(MONTH,Date) AS Month,
        DATEPART(YEAR, Date) AS Year from Dataset
        group by DATENAME(MONTH,Date), DATEPART(YEAR, Date)
    ) AS PostsDataset
PIVOT
(
    SUM(Total_Posts) FOR Month IN ([January], [February], [March],
    [April], [May], [June], [July], [August], [September], [October],
      [November], [December])
) AS PivotDataset
order by Year;

/*Top 5 Categories by Reaction Type*/
select TOP 5 WITH TIES
    Category,
    Reaction_Type_Count,
    CONCAT(ROUND(CAST(Reaction_Type_Count AS FLOAT) / SUM(Reaction_Type_Count)
      OVER() * 100, 2), '%') AS [Percentage of Reaction_Type_Count]
from
(
    select
        Category,
        COUNT(Reaction_Type) AS Reaction_Type_Count,
        DENSE_RANK() OVER (ORDER BY COUNT(Reaction_Type) DESC) AS Rank
    from Dataset
    group by Category
) AS RankedCategories
where RankedCategories.Rank <= 5 --filters the top 5 categories from
  RankedCategories.
order by Reaction_Type_Count DESC;

/*Top 5 Categories by Total Score*/
select TOP 5 WITH TIES Category, SUM(Score) AS Total_Score from Dataset
group by Category
order by Total_Score DESC;

```

```sql
/*Total Score by Sentiment*/
select Sentiment, SUM(Score) AS Total_Score from Dataset
group by Sentiment
order by Total_Score DESC;
```