Top 10 restaurants by good users

```
WITH good_users AS (
   SELECT UserName
   FROM joinedTable
   WHERE UserAverageStars > 3
),

good_reviews AS (
   SELECT BusinessName, ReviewStars, BusinessReviewCount
   FROM joinedTable
   WHERE LOWER(BusinessCategories) LIKE '%restaurant%'
   AND UserName IN (SELECT UserName FROM good_users)
)

SELECT BusinessName, AVG(ReviewStars) as AverageRating, SUM(BusinessReviewCount) as
TotalGoodReviews
FROM good_reviews
GROUP BY BusinessName
ORDER BY AverageRating DESC, TotalGoodReviews DESC
LIMIT 10;
```

Top 10 Businesses with Highest Review Variability by 'Good' Users

```
WITH good_users AS (
   SELECT UserName
   FROM joinedTable
   WHERE UserAverageStars > 3
),

good_reviews AS (
   SELECT BusinessName, split(BusinessCategories, '\\|')[0] as FirstCategory, ReviewStars
   FROM joinedTable
   WHERE UserName IN (SELECT UserName FROM good_users)
),

businesses_with_reviews AS (
   SELECT BusinessName, FirstCategory
   FROM good_reviews
   GROUP BY BusinessName, FirstCategory
   HAVING COUNT(*) > 50
),

businesses_with_variability AS (
   SELECT BusinessName, FirstCategory, stddev(ReviewStars) as Variability
   FROM good_reviews
   WHERE BusinessName IN (SELECT BusinessName FROM businesses_with_reviews)
   GROUP BY BusinessName, FirstCategory
)

SELECT BusinessName, FirstCategory
FROM (
   SELECT BusinessName, FirstCategory
   FROM businesses_with_variability
```

```
    ORDER BY Variability DESC
) t
LIMIT 10;
```

Top 10 Businesses with Highest Review Count and Their First Categories

```
WITH reviews AS (
    SELECT BusinessName, split(BusinessCategories, '\\|')[0] as FirstCategory
    FROM joinedTable
),

businesses_with_reviews AS (
    SELECT BusinessName, FirstCategory, COUNT(*) as ReviewCount
    FROM reviews
    GROUP BY BusinessName, FirstCategory
)

SELECT BusinessName, FirstCategory
FROM (
    SELECT BusinessName, FirstCategory
    FROM businesses_with_reviews
    ORDER BY ReviewCount DESC
) t
LIMIT 10;
```

Top 10 rating business with first category

```
WITH reviews AS (
    SELECT BusinessName, split(BusinessCategories, '\\|')[0] as FirstCategory, ReviewStars
    FROM joinedTable
),

businesses_with_ratings AS (
    SELECT BusinessName, FirstCategory, AVG(ReviewStars) as AvgRating, COUNT(*) as ReviewCount
    FROM reviews
    GROUP BY BusinessName, FirstCategory
)

SELECT BusinessName, FirstCategory
FROM (
    SELECT BusinessName, FirstCategory
    FROM businesses_with_ratings
    ORDER BY AvgRating DESC, ReviewCount DESC
) t
LIMIT 10;
```

The average UserReviewCount of all users
UserAverageStars for all users
Calculate the number of users who joined each year according to UserYelpingSince

```
SELECT AVG(UserReviewCount) as AverageUserReviewCount
FROM joinedTable;
SELECT AVG(UserAverageStars) as AverageUserAverageStars
FROM joinedTable;
```

```
SELECT year(UserYelpingSince) as JoinYear, COUNT(*) as NumberOfUsers
FROM joinedTable
GROUP BY year(UserYelpingSince)
ORDER BY JoinYear;
```

Top 10 restaurants by good users with categories and state

```
WITH good_users AS (
  SELECT UserName
  FROM joinedTable
  WHERE UserAverageStars > 3
),

restaurant_reviews AS (
  SELECT BusinessName, concat_ws('|', split(BusinessCategories, '\\|')[0],
split(BusinessCategories, '\\|')[1]) as FirstTwoCategories, BusinessState, ReviewStars
  FROM joinedTable
  WHERE UserName IN (SELECT UserName FROM good_users) AND
     lower(BusinessCategories) LIKE '%restaurant%'
),

avg_ratings AS (
  SELECT BusinessName, FirstTwoCategories, BusinessState, AVG(ReviewStars) as AvgRating
  FROM restaurant_reviews
  GROUP BY BusinessName, FirstTwoCategories, BusinessState
)

SELECT BusinessName, FirstTwoCategories, BusinessState
FROM (
  SELECT BusinessName, FirstTwoCategories, BusinessState
  FROM avg_ratings
  ORDER BY AvgRating DESC
) t
LIMIT 10;
```

Top 10 restaurants by good users where yelping_since = 2022 or 2021 or 2020

```
WITH good_users AS (
  SELECT UserName
  FROM joinedTable
  WHERE UserAverageStars > 3 AND
     year(UserYelpingSince) IN (2020, 2021, 2022)
),

restaurant_reviews AS (
  SELECT BusinessName, concat_ws('|', split(BusinessCategories, '\\|')[0],
split(BusinessCategories, '\\|')[1]) as FirstTwoCategories, BusinessState, ReviewStars
  FROM joinedTable
  WHERE UserName IN (SELECT UserName FROM good_users) AND
     lower(BusinessCategories) LIKE '%restaurant%'
),

avg_ratings AS (
  SELECT BusinessName, FirstTwoCategories, BusinessState, AVG(ReviewStars) as AvgRating
```

```
   FROM restaurant_reviews
   GROUP BY BusinessName, FirstTwoCategories, BusinessState
)

SELECT BusinessName, FirstTwoCategories, BusinessState
FROM (
   SELECT BusinessName, FirstTwoCategories, BusinessState
   FROM avg_ratings
   ORDER BY AvgRating DESC
) t
LIMIT 10;
```

Top 10 restaurants reviewed by good users and opinion leaders (who has friends more than avg friends_count)

```
WITH avg_friends_count AS (
   SELECT AVG(UserFriendsCount) as AvgFriendsCount
   FROM joinedTable
),

opinion_leaders AS (
   SELECT j.UserName
   FROM joinedTable j, avg_friends_count a
   WHERE j.UserFriendsCount > a.AvgFriendsCount
),

good_users AS (
   SELECT UserName
   FROM joinedTable
   WHERE UserAverageStars > 3
),

good_opinion_leaders AS (
   SELECT UserName
   FROM good_users
   WHERE UserName IN (SELECT UserName FROM opinion_leaders)
),

restaurant_reviews AS (
   SELECT BusinessName, concat_ws('|', split(BusinessCategories, '\\|')[0],
split(BusinessCategories, '\\|')[1]) as FirstTwoCategories, ReviewStars
   FROM joinedTable
   WHERE UserName IN (SELECT UserName FROM good_opinion_leaders) AND
        lower(BusinessCategories) LIKE '%restaurant%'
),

avg_ratings AS (
   SELECT BusinessName, FirstTwoCategories, AVG(ReviewStars) as AvgRating
   FROM restaurant_reviews
   GROUP BY BusinessName, FirstTwoCategories
)
```

```
SELECT BusinessName, FirstTwoCategories
FROM (
   SELECT BusinessName, FirstTwoCategories
   FROM avg_ratings
   ORDER BY AvgRating DESC
) t
LIMIT 10;
```