character_datatypes_common_issues

```
-- Select the count of each level of priority
SELECT priority, COUNT(*)  -- Counts the occurrences of each priority level
FROM evanston311        -- From the evanston311 table
GROUP BY priority;      -- Groups the results by priority level
```

Explanation:

- This SQL query calculates the number of times each priority level appears in the evanston311 table. It uses GROUP BY to group the rows based on the priority column and COUNT(*) to count the rows within each group. The result shows each unique priority and its corresponding count.

```
-- Find zip codes that appear in at least 100 rows
-- Also get the count of each zip code
SELECT zip, COUNT(*) AS count
  FROM evanston311
 GROUP BY zip
HAVING COUNT(*) >= 100;
```

Explanation:

- This SQL query analyzes the evanston311 table to find zip codes that occur at least 100 times. It groups the rows by zip code, counts the occurrences of each zip code using COUNT(*), and then filters these counts using HAVING to only include those with a count of 100 or more. The result shows each qualifying zip code and its frequency.

```
-- Find values of source that appear in at least 100 rows
-- Also get the count of each value
SELECT source, COUNT(*)
  FROM evanston311
 GROUP BY source
HAVING COUNT(*) >= 100;
```

Explanation:

- This SQL query analyzes the evanston311 table. It groups the rows by the source column, counts the occurrences of each unique source value, and then filters these grouped results to only include those source values that appear in 100 or more rows. The result shows each qualifying source and its corresponding count.

```
-- Find the 5 most common values of street and the count of each
SELECT street, COUNT(*) AS street_count  -- Counts occurrences of each street
FROM evanston311
GROUP BY street
ORDER BY street_count DESC  -- Orders by count in descending order to get the most common
```

```
first
LIMIT 5;
```

Explanation:

- This SQL query identifies the five most frequent street names from the evanston311 table. It groups the data by street name, counts the occurrences of each street using COUNT(*), orders the results by count in descending order (ORDER BY street_count DESC), and finally limits the output to the top 5 using LIMIT 5. The original query used COUNT(DISTINCT street) which is redundant since GROUP BY street already ensures only unique streets are counted. The improved version uses COUNT(*) which is more efficient.

```
-- Find the 5 most common values of street and the count of each
SELECT street, COUNT(*)
 FROM evanston311
GROUP BY street
ORDER BY COUNT(*) DESC
LIMIT 5;
```

Explanation:

- This SQL query finds the five most frequent street names from the evanston311 table. It groups the rows by street name, counts the occurrences of each street using COUNT(*), orders the results in descending order based on the count, and finally limits the output to the top 5 rows.