date_time_types_formats

```sql
-- Count requests created on January 31, 2017
SELECT count(*)
  FROM evanston311
 WHERE date_created::date = '2017-01-31';
```

Explanation:

- This SQL query counts the number of records in the evanston311 table where the date_created field is equal to January 31st, 2017. The ::date casts the date_created field (presumably a timestamp or datetime type) to a date type for comparison with the provided date string.

```sql
-- Count requests created on February 29, 2016
SELECT count(*)
  FROM evanston311
 WHERE date_created >= '2016-02-29'
   AND date_created < '2016-03-01';
```

Explanation:

- This SQL query counts the number of records in the evanston311 table where the date_created field falls on February 29, 2016. It uses a date range to ensure only entries from that specific date are included. Note that 2016 was a leap year, so February 29th did exist that year.

```sql
-- Count requests created on March 13, 2017
SELECT count(*)
  FROM evanston311
 WHERE date_created >= '2017-03-13'
   AND date_created < '2017-03-13'::date + 1;
```

Explanation:

- This SQL query counts the number of records in the evanston311 table where the date_created field falls on March 13, 2017. It uses a date range to ensure only entries from that specific day are included. The '2017-03-13'::date + 1 part adds one day to the date, effectively creating an upper bound for the date range that excludes March 14th.

```sql
-- Subtract the min date_created from the max
SELECT MAX(date_created) - MIN(date_created)
  FROM evanston311;
```

Explanation:

- This SQL query calculates the difference between the maximum and minimum values in the date_created column of the evanston311 table. The result shows the total duration (in days if the datatype is DATE, or potentially in some other time unit depending on the exact date_created data type) between the earliest and latest recorded dates.

```
-- How old is the most recent request?
SELECT now() - MAX(date_created)
 FROM evanston311;
```

Explanation:

- This SQL query calculates the age of the most recent request in the evanston311 table. It finds the maximum date_created value (the most recent date) and subtracts it from the current time (now()), giving the time elapsed since the most recent request was created. The result is a time interval.

```
-- Add 100 days to the current timestamp
SELECT now() + '100 days'::interval;
```

Explanation:

- This SQL query adds 100 days to the current timestamp. now() gets the current timestamp. '100 days'::interval casts the string ‘100 days’ into an interval data type, allowing it to be added to the timestamp. The result is a timestamp 100 days in the future.

```
-- Select the current timestamp,
-- and the current timestamp + 5 minutes
SELECT now(),
    now() + '5 minutes'::interval;
```

Explanation:

- This SQL query retrieves the current timestamp from the database system using the now() function. It then adds 5 minutes to the current timestamp by using the + operator with a string literal ‘5 minutes’ explicitly cast as an interval data type. The result shows two timestamps: the current time and the time 5 minutes into the future.

```
-- Select the category and the average completion time by category
SELECT category,
    AVG(date_completed - date_created) AS completion_time
 FROM evanston311
```

```sql
 GROUP BY category
-- Order the results
 ORDER BY completion_time DESC;
```

Explanation:

- This SQL query calculates the average time it takes to complete requests for each category in the evanston311 table. It subtracts the date_created from date_completed to find the completion time for each request, then averages these times for each category using AVG(). Finally, it orders the results from longest average completion time to shortest.