Project SQL: Chicago traffic crashes

- Original query create table

```
DROP TABLE IF EXISTS crashes;
CREATE TABLE crashes (
     crash id int PRIMARY KEY,
     crash date timestamp,
     posted speed limit varchar(2),
     traffic control device varchar(50),
     device condition varchar(50),
     weather condition varchar(25),
     lighting_condition varchar(50),
     first_crash_type varchar(100),
     traffic_way_type varchar(100),
     lane_count varchar(5),
     alignment varchar(25),
     roadway surface condition varchar(25),
     road defect varchar(50),
     report type varchar(50),
     crash_type varchar(100),
     hit_and_run varchar(1),
     damage varchar(25),
     date police notified timestamp,
     primary cause varchar(100),
     secondary cause varchar(100),
     street direction varchar(1),
     street name varchar(50),
     statement taken varchar(1),
     work zone varchar(1),
     work_zone_type varchar(25),
     workers_present varchar(1),
     number unit varchar(2),
     most_severe_injury varchar(50),
     injuries total varchar(2),
     injuries fatal varchar(2),
     injuries incapacitated varchar(2).
     injuries non incapacitated varchar(2),
     injuries_reported_not_evident varchar(2),
     crash_hour varchar(2),
```

```
crash_day_of_week varchar(9),
     crash_month varchar(2),
     latitude varchar(25),
     longitude varchar(25),
     crash location varchar(75)
);

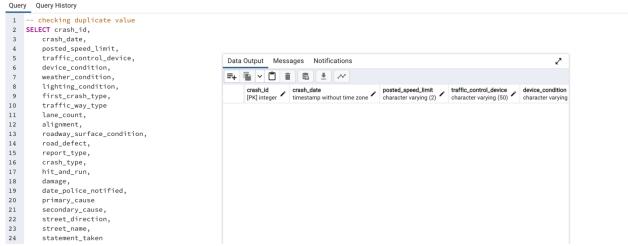
    Create columns in pgadmin4 then import cvs file

-- Create Crashes Table
DROP TABLE IF EXISTS crashes:
CREATE TABLE crashes (
     crash id int PRIMARY KEY,
     crash_date timestamp,
     posted speed limit varchar(2),
     traffic control device varchar(50),
     device_condition varchar(50),
     weather condition varchar(25),
     lighting_condition varchar(50),
     first_crash_type varchar(100),
     traffic way type varchar(100),
     lane count varchar(5),
     alignment varchar(25),
     roadway surface condition varchar(25),
     road defect varchar(50),
     report_type varchar(50),
     crash_type varchar(100),
     hit and run varchar(1),
     damage varchar(25),
     date_police_notified timestamp,
     primary_cause varchar(100),
     secondary_cause varchar(100),
     street direction varchar(1),
     street_name varchar(50),
     number_unit varchar(2),
     most_severe_injury varchar(50),
     injuries_total varchar(2),
     injuries_fatal varchar(2),
     injuries_incapacitated varchar(2),
     injuries non incapacitated varchar(2),
     injuries reported not evident varchar(2),
```

```
crash_hour varchar(2),
      crash_day_of_week varchar(9),
      crash month varchar(2),
      latitude varchar(25),
      longitude varchar(25),
      crash_location varchar(75),
);
COPY crashes
FROM
'/Users/rusamijan/Desktop/Chicago traffic crashes-main/update
csv/CTC cleaned.csv<sup>1</sup>
DELIMITER ',' CSV HEADER;
-- checking null value
SELECT *
FROM crashes
WHERE crash date is NULL;
-- checking null value
SELECT *
 FROM crashes
WHERE crash_date is NULL;
   Data Output Messages Notifications
                                                                         7
   3
                                                   traffic_control_device
                                    posted_speed_limit
                                                                  device_condition
                                    character varying (2)
                 timestamp without time zone
                                                   character varying (50)
                                                                  character varying
-- checking duplicate value
SELECT crash_id,
      crash_date,
      posted speed limit,
      traffic control device,
      device_condition,
      weather_condition,
      lighting_condition,
      first_crash_type,
      traffic_way_type
      lane_count,
      alignment,
      roadway_surface_condition,
```

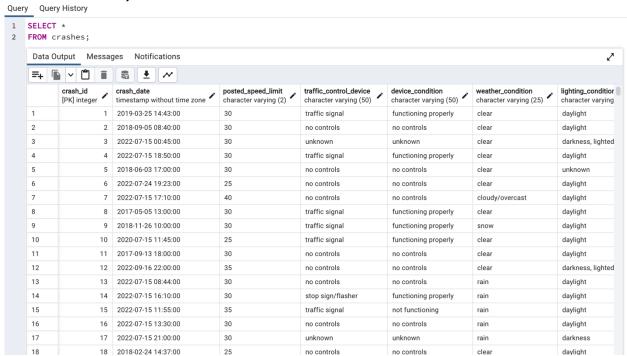
```
road_defect,
     report_type,
     crash_type,
     hit_and_run,
     damage,
     date_police_notified,
     primary_cause
     secondary_cause,
     street_direction,
     street_name,
     statement taken
     work_zone,
     work_zone_type,
     workers present
     number unit,
     most severe injury,
     injuries_total,
     injuries_fatal,
     injuries_incapacitated,
     injuries_non_incapacitated,
     injuries_reported_not_evident,
     crash_hour,
     crash_day_of_week,
     crash_month,
     latitude,
     longitude,
     crash_location,
COUNT(*)
FROM crashes
GROUP BY crash_id,
     crash_date,
     posted_speed_limit,
     traffic_control_device,
     device_condition,
     weather_condition,
     lighting_condition,
     first_crash_type,
     traffic_way_type,
     lane count,
     alignment,
```

roadway_surface_condition, road_defect, report_type, crash_type, hit_and_run, damage, date_police_notified, primary_cause, secondary_cause, street_direction, street_name, statement_taken, work_zone, work_zone_type, workers_present, number_unit, most_severe_injury, injuries_total, injuries_fatal, injuries_incapacitated, injuries_non_incapacitated, injuries_reported_not_evident, crash_hour, crash_day_of_week, crash_month, latitude, longitude, crash_location **HAVING COUNT(*)>1;**



No duplicate data

Checking database infoSELECT *FROM crashes;



--Descriptive statistics (MIN and MAX) for numerical columns for timeline crash

SELECT MIN(crash_date) AS min_crash_date,
MAX(crash_date) AS max_crash_date,
MIN(posted_speed_limit) AS min_speed_limit,
MAX(posted_speed_limit) AS max_speed_limit,
MIN(lane_count) AS min_lane_count,

MAX(lane count) AS max lane count,

MIN(damage) AS min_damage,

MAX(damage) AS max_damage,

MIN(date_police_notified) AS min_date_police_notified,

MAX(date_police_notified) AS max_date_police_notified,

MIN(number_unit) AS min_number_unit,

MAX(number_unit) AS max_number_unit,

MIN(injuries_total) AS min_injuries_total,

MAX(injuries_total) AS max_injuries_total,

MIN(injuries_fatal) AS min_injuries_fatal,

MAX(injuries_fatal) AS max_injuries_fatal,

MIN(injuries_incapacitated) AS min_injuries_incapacitatede,

MAX(injuries_incapacitated) AS max_injuries_incapacitated,

MIN(injuries_reported_not_evident) AS

min_injuries_reported_not_evident,

MAX(injuries_reported_not_evident) AS

max_injuries_reported_not_evident,

MIN(crash_hour) AS min_crash_hour,

MAX(crash_hour) AS max_crash_hour,

MIN(crash_month) AS min_crash_month,

MAX(crash_month) AS max_crash_month,

MIN(latitude) AS min_latitude,

MAX(latitude) AS max_latitude,

MIN(longitude) AS min_longitude,

MAX(longitude) AS max_longitude,

MIN(crash_location) AS min_crash_location,

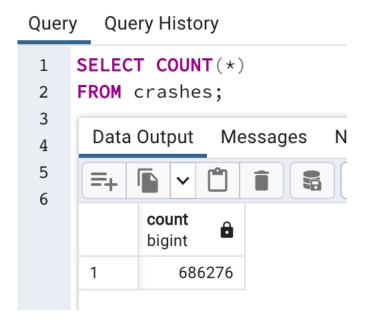
MAX(crash_location) AS max_crash_lo

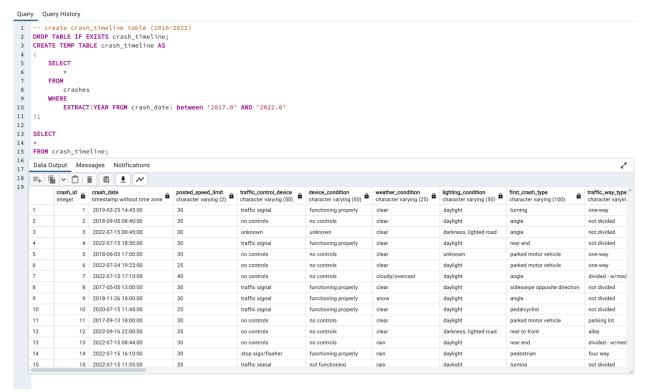


cation

FROM crash_timeline;

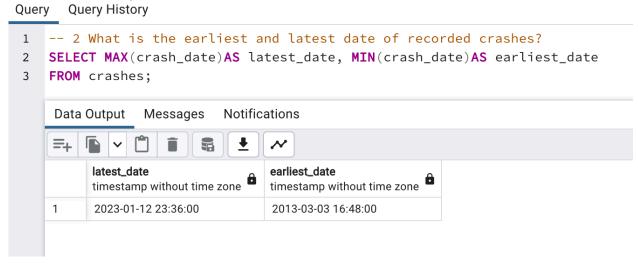
-- What is the total count of recorded crashes in the complete dataset? SELECT COUNT(*) FROM crashes;





Due to fail to import Chicago traffic crash part 2, the total data is not correct. But I will use what I have for analysis.

-- What is the earliest and latest date of recorded crashes? SELECT MAX(crash_date)AS latest_date, MIN(crash_date)AS earliest_date FROM crashes;



-- What is the number of reported crashes per year?
--SELECT EXTRACT('YEAR' FROM DATE));
SELECT

EXTRACT('YEAR' FROM crash_date) AS crash_year, count(*) AS reported_crashes

FROM

crashes

GROUP BY

crash_year

ORDER BY

crash_year;

Query Query History

```
1 -- 3 What is the number of reported crashes per year?
2 --SELECT EXTRACT('YEAR' FROM DATE));
 3 SELECT
        EXTRACT('YEAR' FROM crash_date) AS crash_year,
        count(*) AS reported_crashes
 5
   FROM
 6
7
        crashes
8
   GROUP BY
9
        crash_year
10
   ORDER BY
11
       crash_year;
```

Data Output Messages Notifications		
=+		
	crash_year numeric	reported_crashes bigint
1	2013	2
2	2014	6
3	2015	9828
4	2016	44297
5	2017	83786
6	2018	118950
7	2019	117762
8	2020	92088
9	2021	108756
10	2022	108292
11	2023	2509

Found that the data in 2013 to 2016 are missing or **2017 appears to be the first year with the most complete data** so I decided to create a new table that includes only 2017 - 2022 for my analysis

```
-- Create crash_timeline table (2016-2022)
DROP TABLE IF EXISTS crash_timeline;
CREATE TEMP TABLE crash_timeline AS
(

SELECT

*
FROM

crashes
WHERE

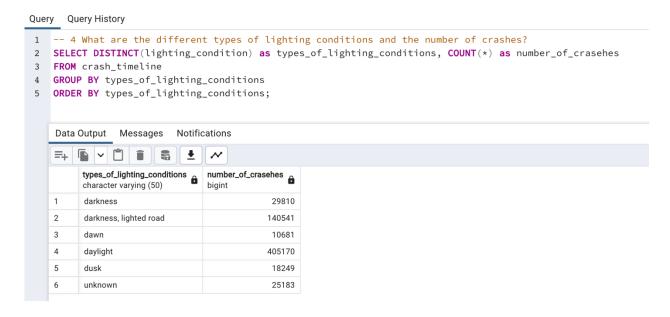
EXTRACT(YEAR FROM crash_date) between '2017.0' AND '2022.0'
);

SELECT

*
FROM crash_timeline;
```

— What are the different types of lighting conditions and the number of crashes?

```
SELECT DISTINCT(lighting_condition) as types_of_lighting_conditions, COUNT(*) as number_of_crasehes FROM crash_timeline GROUP BY types_of_lighting_conditions ORDER BY types_of_lighting_conditions;
```



—What are the different kinds of road conditions and the number of crashes?

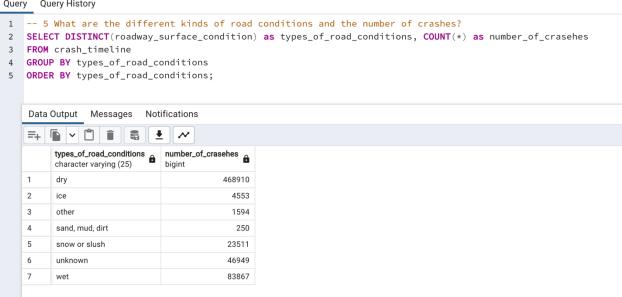
SELECT DISTINCT(roadway_surface_condition) as types_of_road_conditions,

COUNT(*) as number_of_crasehes

FROM crash timeline

GROUP BY types_of_road_conditions

ORDER BY types_of_road_conditions;



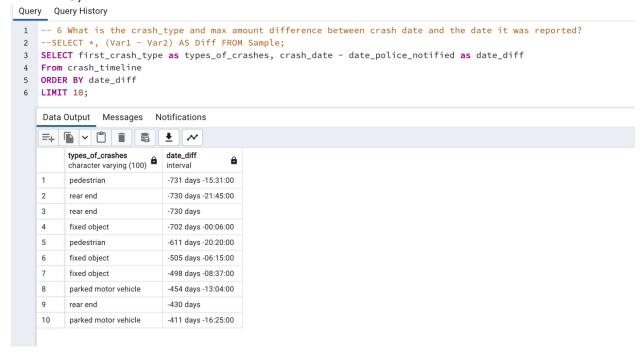
-- What is the crash_type and max amount difference between crash date and the date it was reported?

--SELECT *, (Var1 - Var2) AS Diff FROM Sample;

SELECT first_crash_type as types_of_crashes, crash_date - date_police_notified as date_diff

From crash_timeline ORDER BY date diff

LIMIT 10;



-- What are the top 5 Crash Types?

```
-- 8 What are the top 5 Crash Types?
 1
 2
    -- CTE
 3
    WITH crash_type AS(
         -- inner query
 4
 5
         SELECT
              first_crash_type,
 6
 7
              count(*) AS cnt_crash,
 8
              RANK() OVER (ORDER BY count(*) DESC) AS crash_rank
 9
         FROM crash_timeline
         GROUP BY first_crash_type
10
11
     )
12
          -- outer query
13
     SELECT
14
         first_crash_type as crash_type, cnt_crash
15
     FROM crash_type
16
    WHERE crash_rank <= 5;</pre>
     Data Output
                  Messages
                              Notifications
     =+
           crash_type
                                cnt_crash
           character varying (100)
                                bigint
     1
           parked motor vehicle
                                     147481
     2
           rear end
                                     140314
     3
           sideswipe same direction
                                     93799
     4
           turning
                                     89882
     5
           angle
                                      68282
```

-- How many road defects caused crashes?

FROM road_defects ORDER BY total_road_defect DESC;

```
-- 10 How many road defects caused crashes?
 2
    -- CTE
 3
    WITH road_defects AS(
         SELECT road_defect,
 4
 5
         COUNT(*) AS cnt_road_df
 6
         FROM crash_timeline
         GROUP BY road_defect
 7
 8
 9
    SELECT road_defect AS road_df, cnt_road_df AS total_road_defect
   FROM road_defects
10
    ORDER BY total_road_defect DESC;
11
     Data Output
                 Messages
                             Notifications
         =+
                              total_road_defect
          road_df
          character varying (50)
                              bigint
    1
          no defects
                                        515811
    2
          unknown
                                        101142
    3
          rut, holes
                                          4960
          other
                                          3488
    4
    5
          worn surface
                                          2511
    6
          shoulder defect
                                          1229
```

-- What are the top 10 deadliest streets?

street_name, injuries_fatal, cnt_injuries_fetal

FROM deadliest streets

WHERE dead_street_rank <=10

Cottage Grove Ave

Lake Shore Dr Nb

Lake Shore Dr Sb

Kedzie Ave

8

9

10

1

1

1

1

```
query query mistory
 1 -- 12 What are the top 10 deadliest streets?
 2 -- CTE
 3 WITH deadliest_streets AS (
         SELECT
 4
 5
              street_name,
 6
              injuries_fatal,
              count(*) AS cnt_injuries_fetal,
 7
              {\tt RANK()} \ \ {\tt OVER} \ \ ({\tt ORDER} \ \ {\tt BY} \ \ {\tt count(\star)} \ \ {\tt DESC)} \ \ {\tt AS} \ \ {\tt dead\_street\_rank}
 8
         FROM
 9
10
               crash_timeline
11
         WHERE injuries_fatal<>'0'
12
         GROUP BY
13
              injuries_fatal,street_name
14 )
15 SELECT
16
         street_name, injuries_fatal, cnt_injuries_fetal
17 FROM deadliest_streets
18 WHERE dead_street_rank <=10</pre>
                                                                                                               ~
       Data Output Messages Notifications
       =+ 6 ~ 6 6
                                  ₹ ~
                                                      cnt_injuries_fetal
                                  injuries_fatal
             street_name
            character varying (50)
                                  character varying (2)
                                                      bigint
       1
             Ashland Ave
                                                                    24
      2
             Cicero Ave
                                  1
                                                                    22
       3
             Western Ave
                                  1
                                                                    20
       4
             Halsted St
                                  1
                                                                    19
             Pulaski Rd
       5
                                  1
                                                                    19
       6
             Archer Ave
                                  1
                                                                    13
```

13

13

13

12