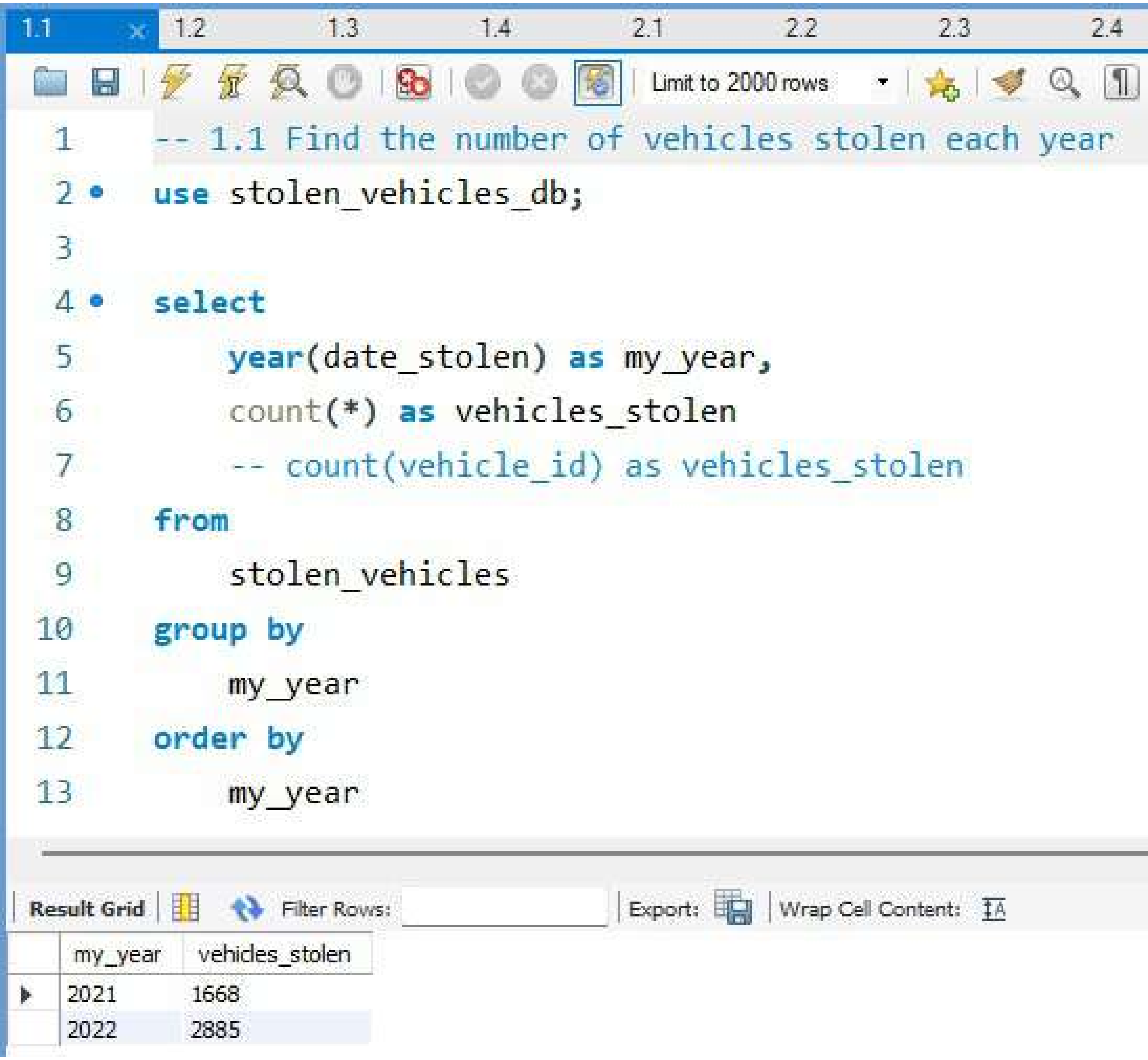
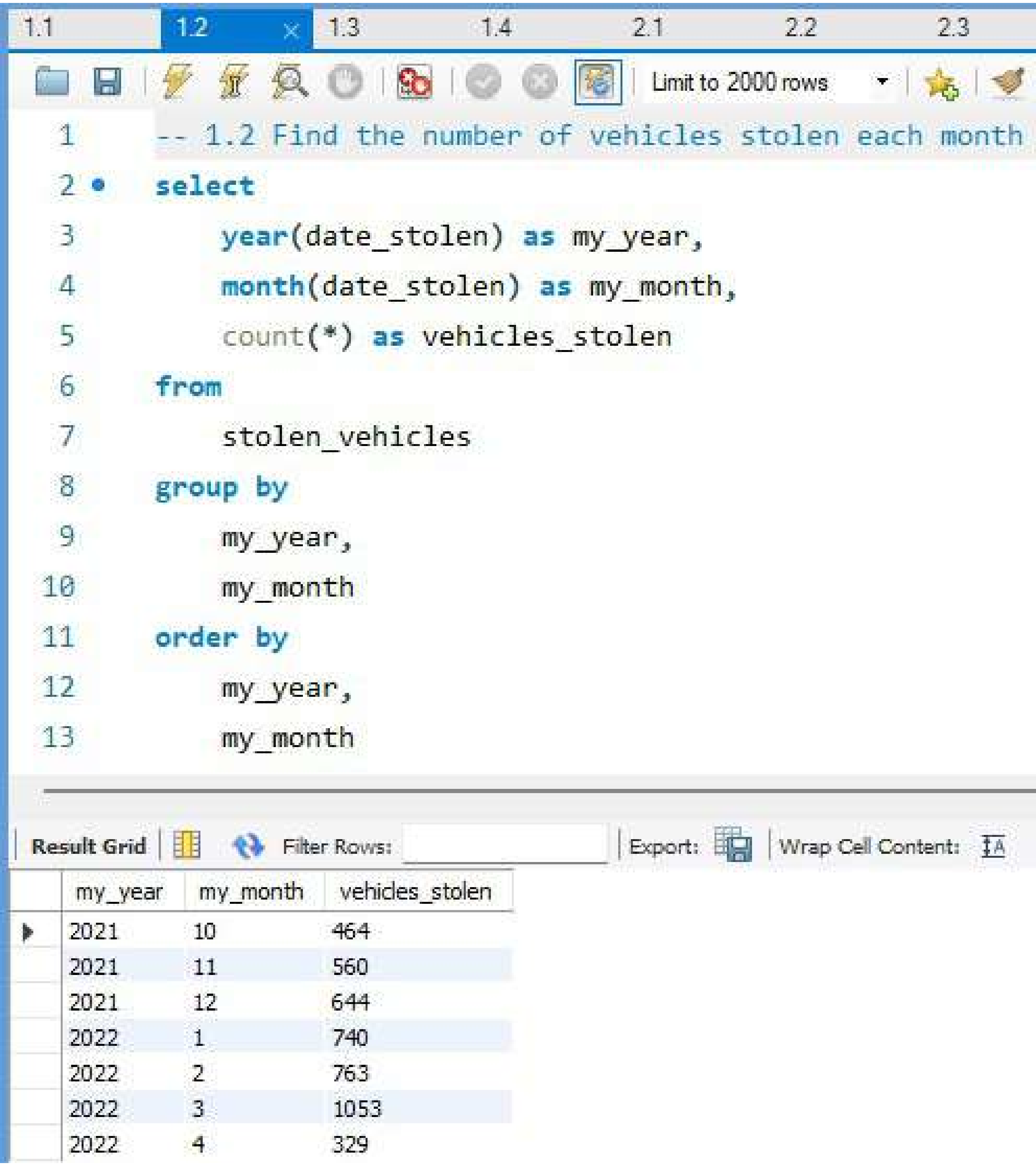


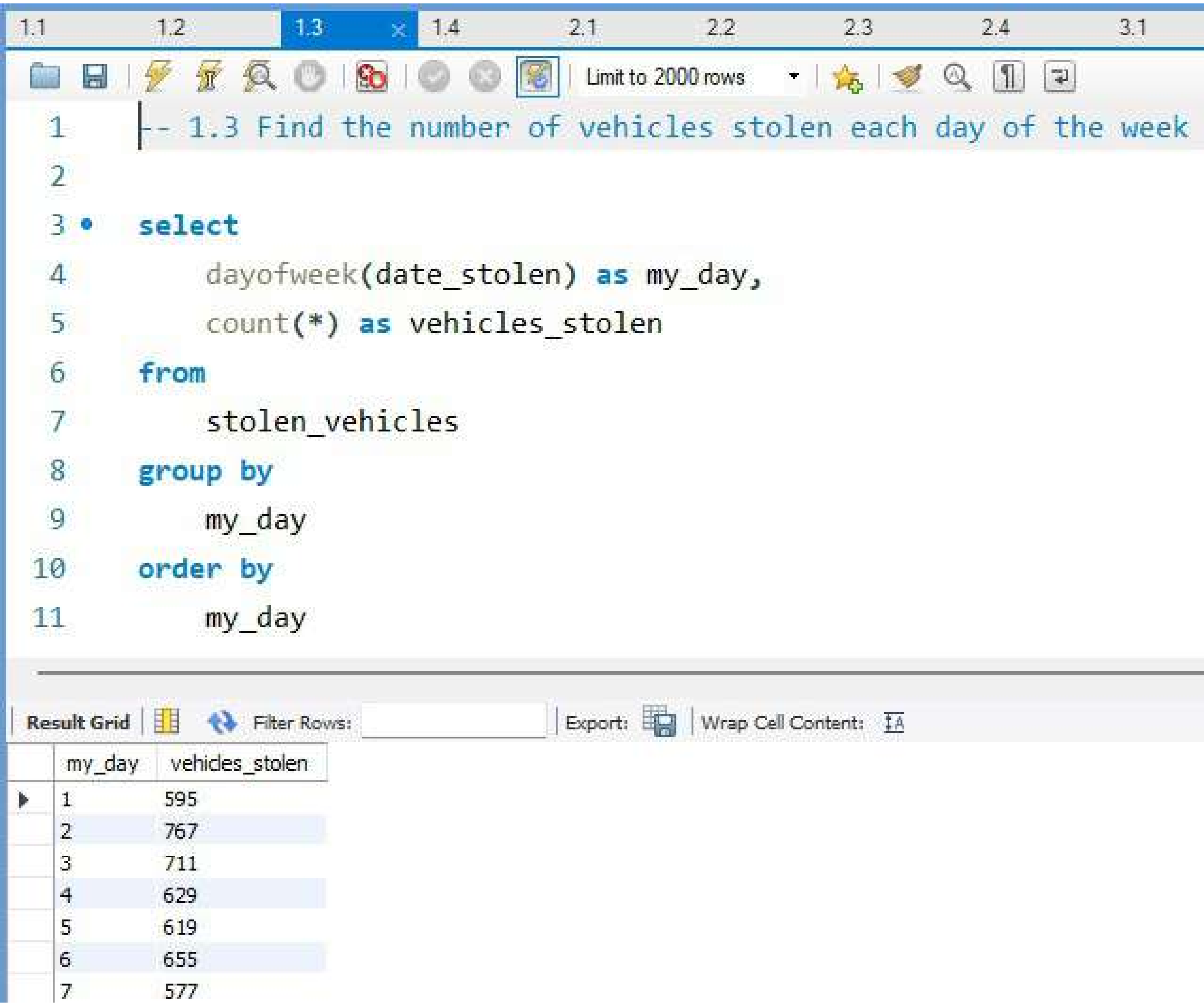
```
-- 1.1 Find the number of vehicles stolen each year
use stolen_vehicles_db;

select
    year(date_stolen) as my_year,
    count(*) as vehicles_stolen
    -- count(vehicle_id) as vehicles_stolen
from
    stolen_vehicles
group by
    my_year
order by
    my_year
```

my_year	vehicles_stolen
2021	1668
2022	2885



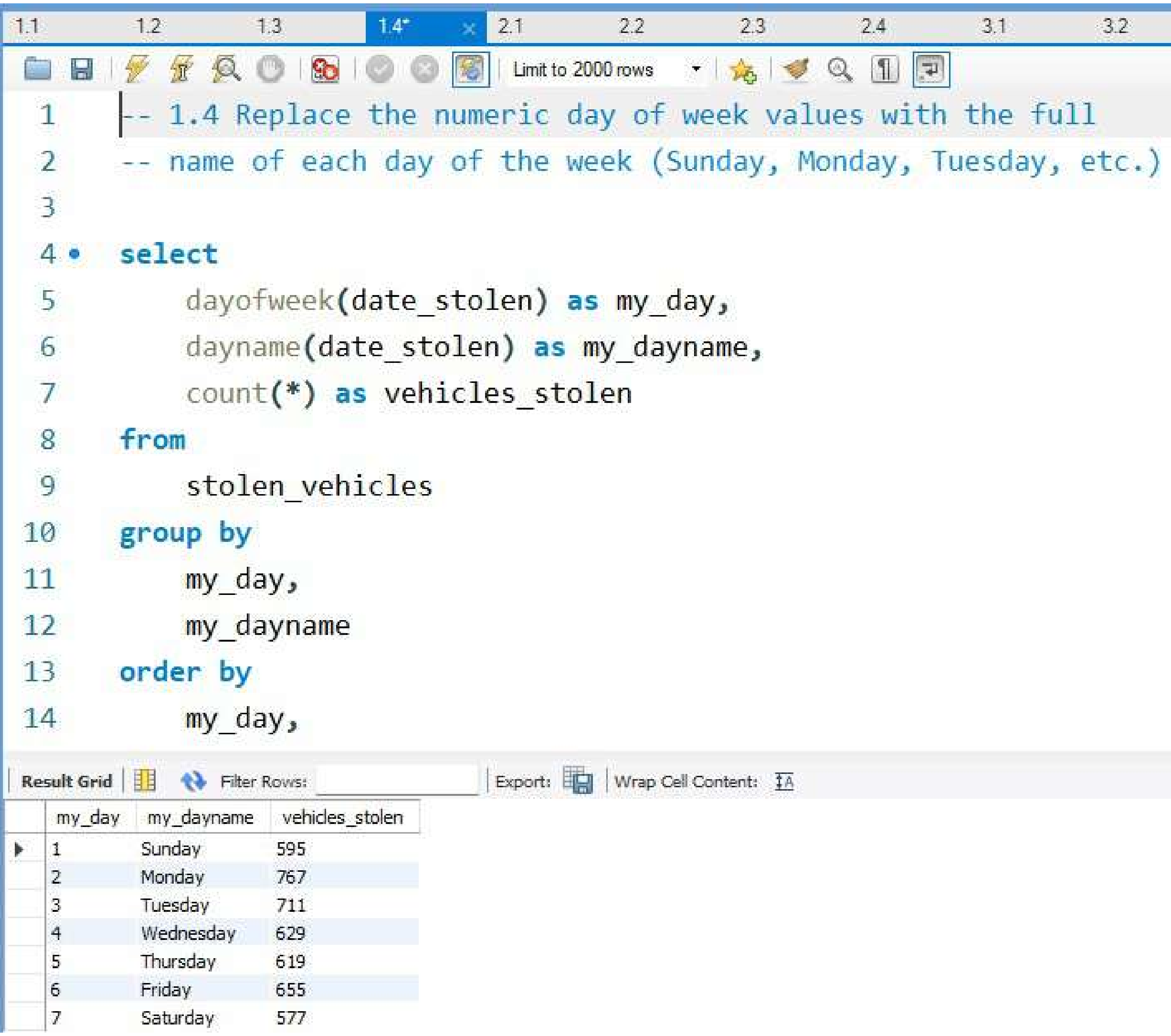
```
-- 1.2 Find the number of vehicles stolen each month
select
    year(date_stolen) as my_year,
    month(date_stolen) as my_month,
    count(*) as vehicles_stolen
from
    stolen_vehicles
group by
    my_year,
    my_month
order by
    my_year,
    my_month
```

my_year	my_month	vehicles_stolen
2021	10	464
2021	11	560
2021	12	644
2022	1	740
2022	2	763
2022	3	1053
2022	4	329



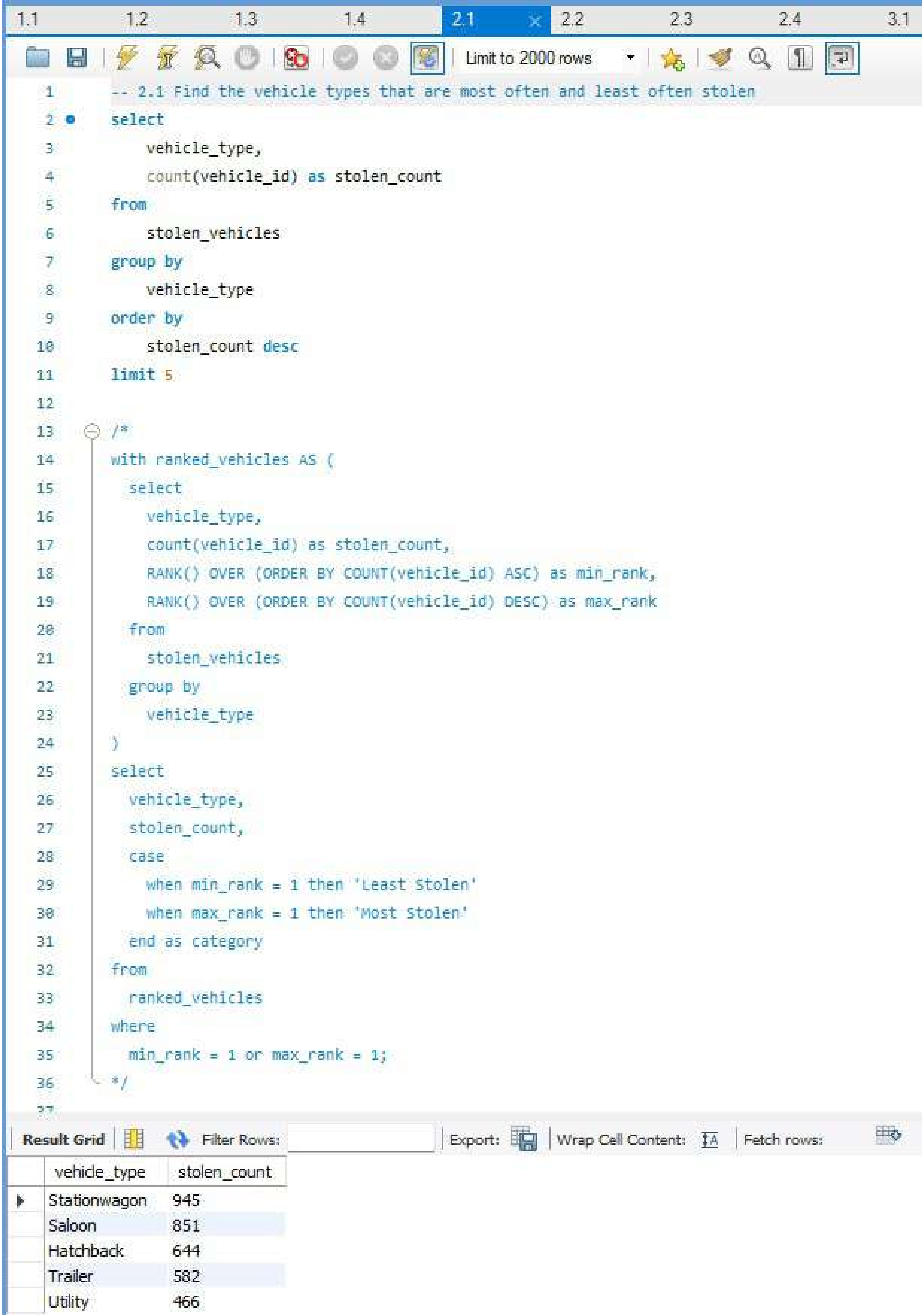
```sql
-- 1.3 Find the number of vehicles stolen each day of the week

select
    dayofweek(date_stolen) as my_day,
    count(*) as vehicles_stolen
from
    stolen_vehicles
group by
    my_day
order by
    my_day
```

my_day	vehicles_stolen
1	595
2	767
3	711
4	629
5	619
6	655
7	577



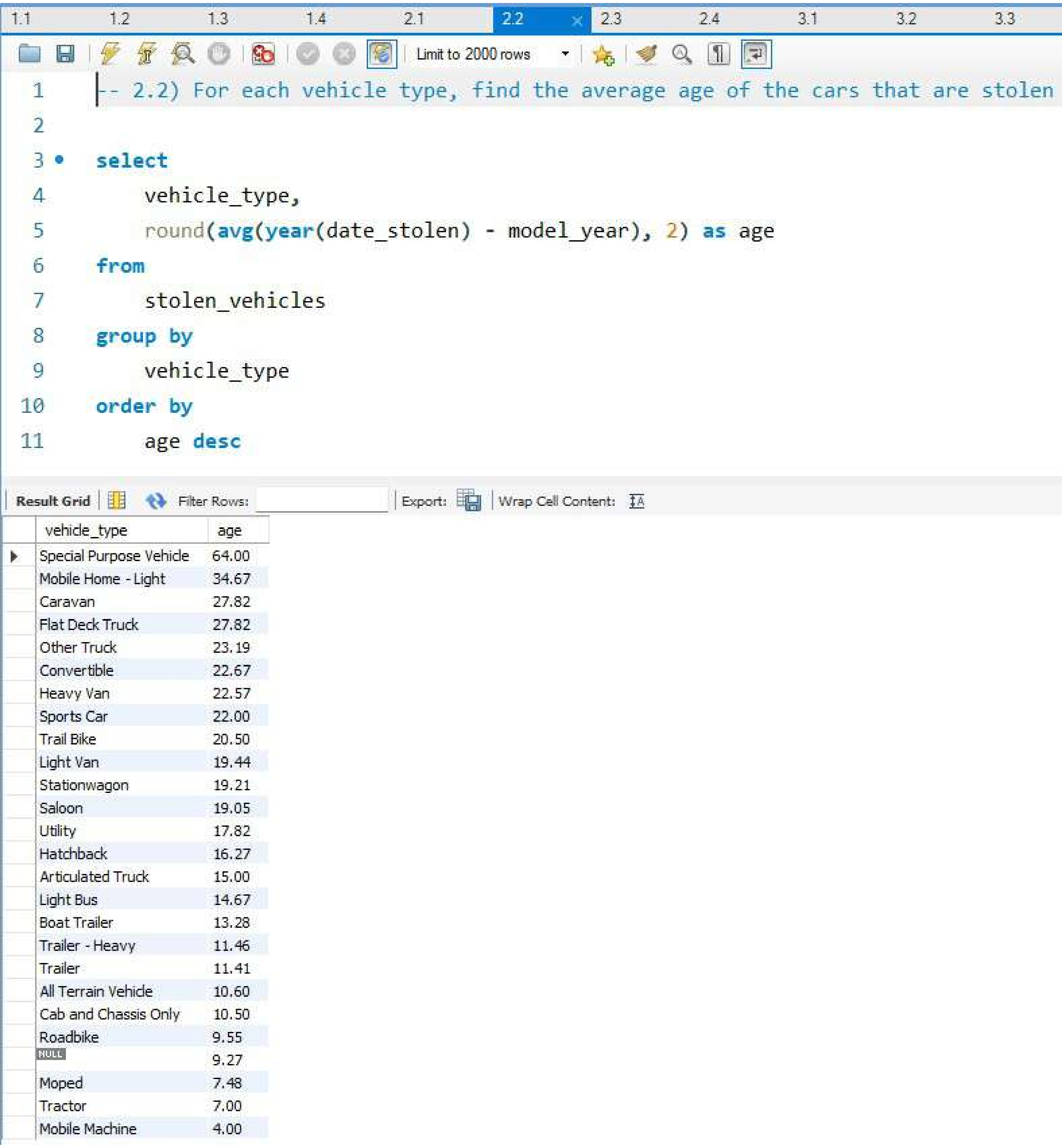
```
-- 1.4 Replace the numeric day of week values with the full
-- name of each day of the week (Sunday, Monday, Tuesday, etc.)

select
    dayofweek(date_stolen) as my_day,
    dayname(date_stolen) as my_dayname,
    count(*) as vehicles_stolen
from
    stolen_vehicles
group by
    my_day,
    my_dayname
order by
    my_day,
```

my_day	my_dayname	vehicles_stolen
1	Sunday	595
2	Monday	767
3	Tuesday	711
4	Wednesday	629
5	Thursday	619
6	Friday	655
7	Saturday	577



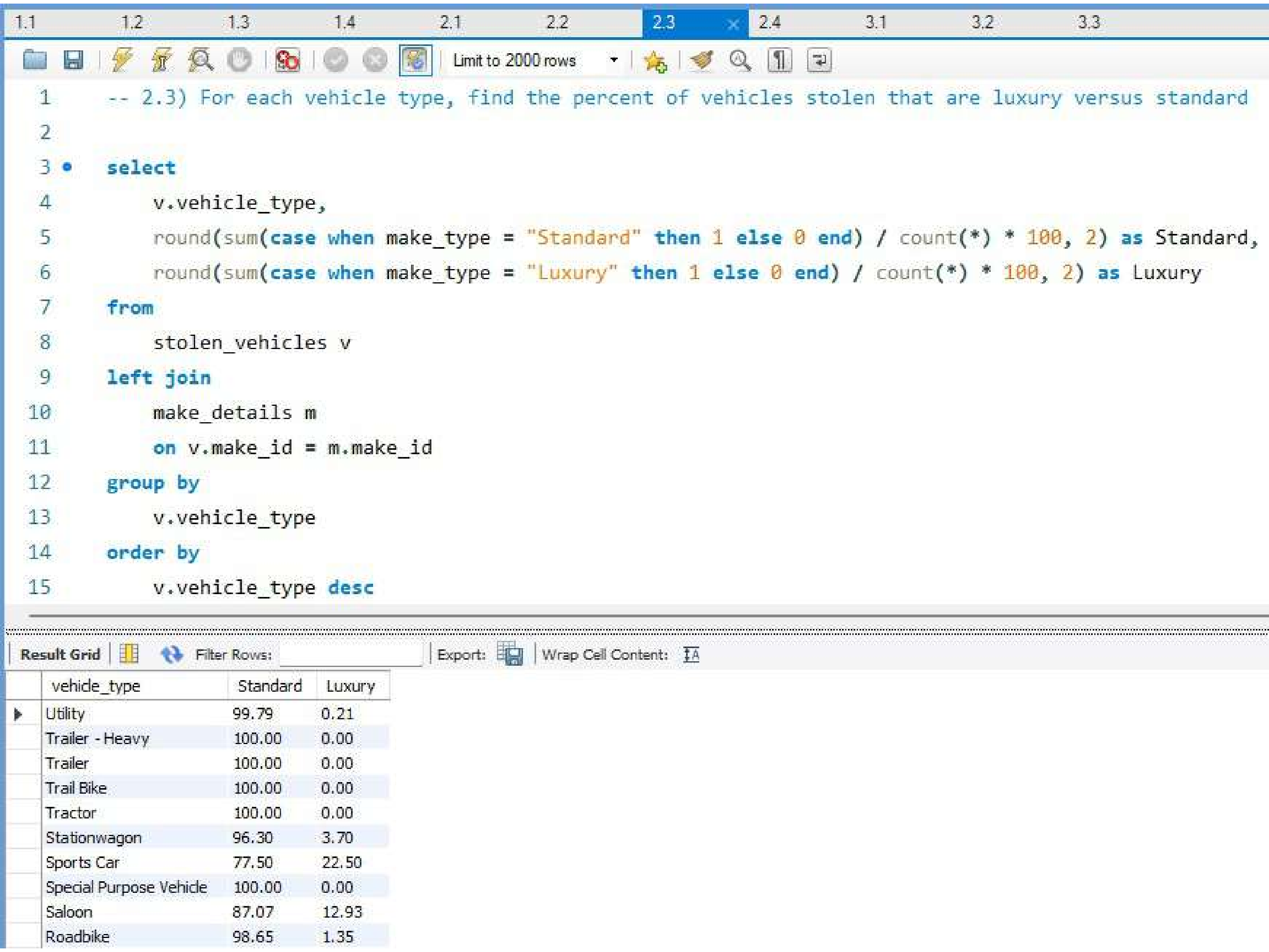
```sql
-- 2.1 Find the vehicle types that are most often and least often stolen
select
    vehicle_type,
    count(vehicle_id) as stolen_count
from
    stolen_vehicles
group by
    vehicle_type
order by
    stolen_count desc
limit 5

/*
with ranked_vehicles AS (
  select
    vehicle_type,
    count(vehicle_id) as stolen_count,
    RANK() OVER (ORDER BY COUNT(vehicle_id) ASC) as min_rank,
    RANK() OVER (ORDER BY COUNT(vehicle_id) DESC) as max_rank
  from
    stolen_vehicles
  group by
    vehicle_type
)
select
  vehicle_type,
  stolen_count,
  case
    when min_rank = 1 then 'Least Stolen'
    when max_rank = 1 then 'Most Stolen'
  end as category
from
  ranked_vehicles
where
  min_rank = 1 or max_rank = 1;
*/
```

vehicle_type	stolen_count
Stationwagon	945
Saloon	851
Hatchback	644
Trailer	582
Utility	466



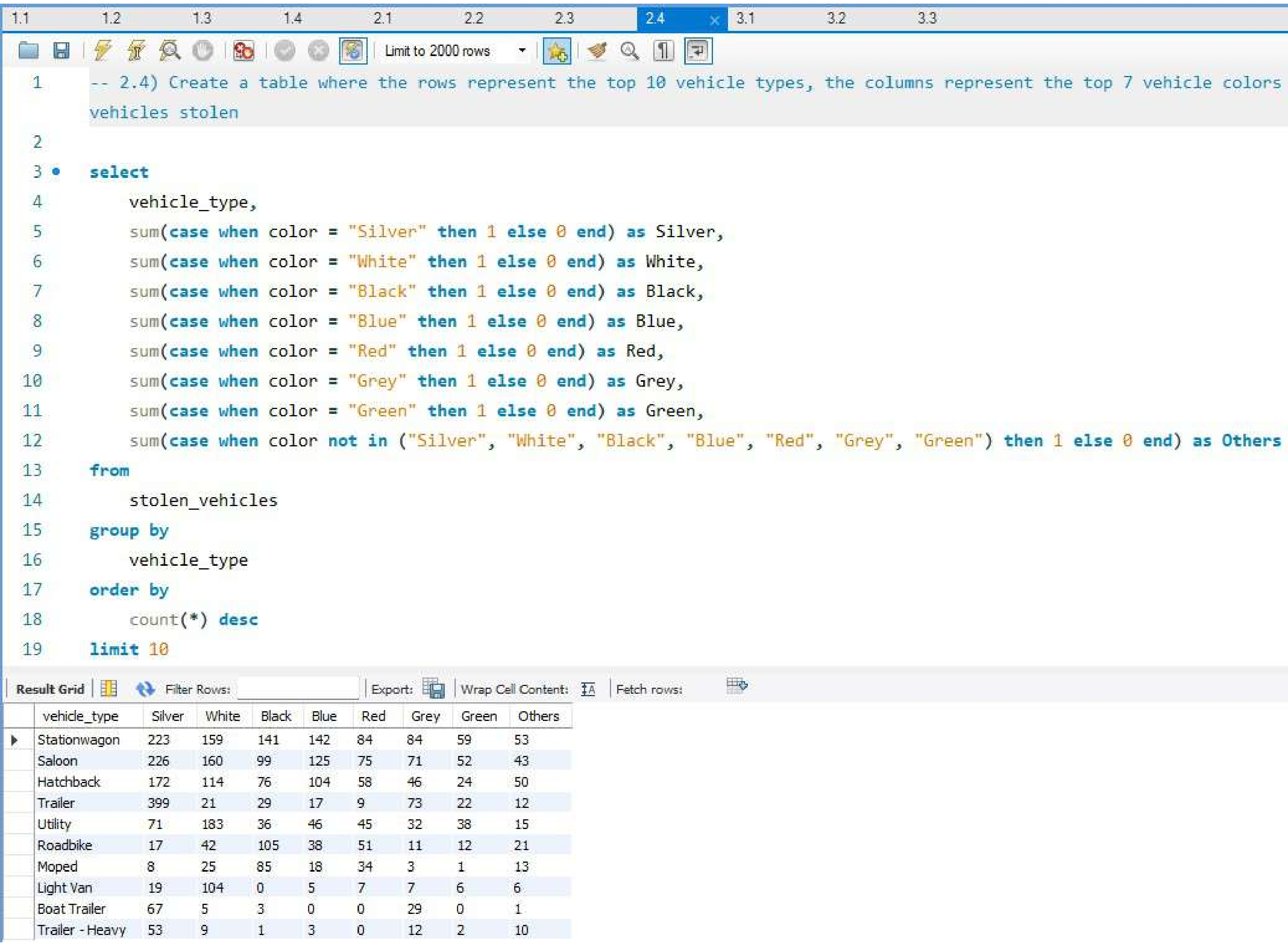
```sql
-- 2.2) For each vehicle type, find the average age of the cars that are stolen

select
    vehicle_type,
    round(avg(year(date_stolen) - model_year), 2) as age
from
    stolen_vehicles
group by
    vehicle_type
order by
    age desc
```

vehicle_type	age
Special Purpose Vehicle	64.00
Mobile Home - Light	34.67
Caravan	27.82
Flat Deck Truck	27.82
Other Truck	23.19
Convertible	22.67
Heavy Van	22.57
Sports Car	22.00
Trail Bike	20.50
Light Van	19.44
Stationwagon	19.21
Saloon	19.05
Utility	17.82
Hatchback	16.27
Articulated Truck	15.00
Light Bus	14.67
Boat Trailer	13.28
Trailer - Heavy	11.46
Trailer	11.41
All Terrain Vehicle	10.60
Cab and Chassis Only	10.50
Roadbike	9.55
NULL	9.27
Moped	7.48
Tractor	7.00
Mobile Machine	4.00



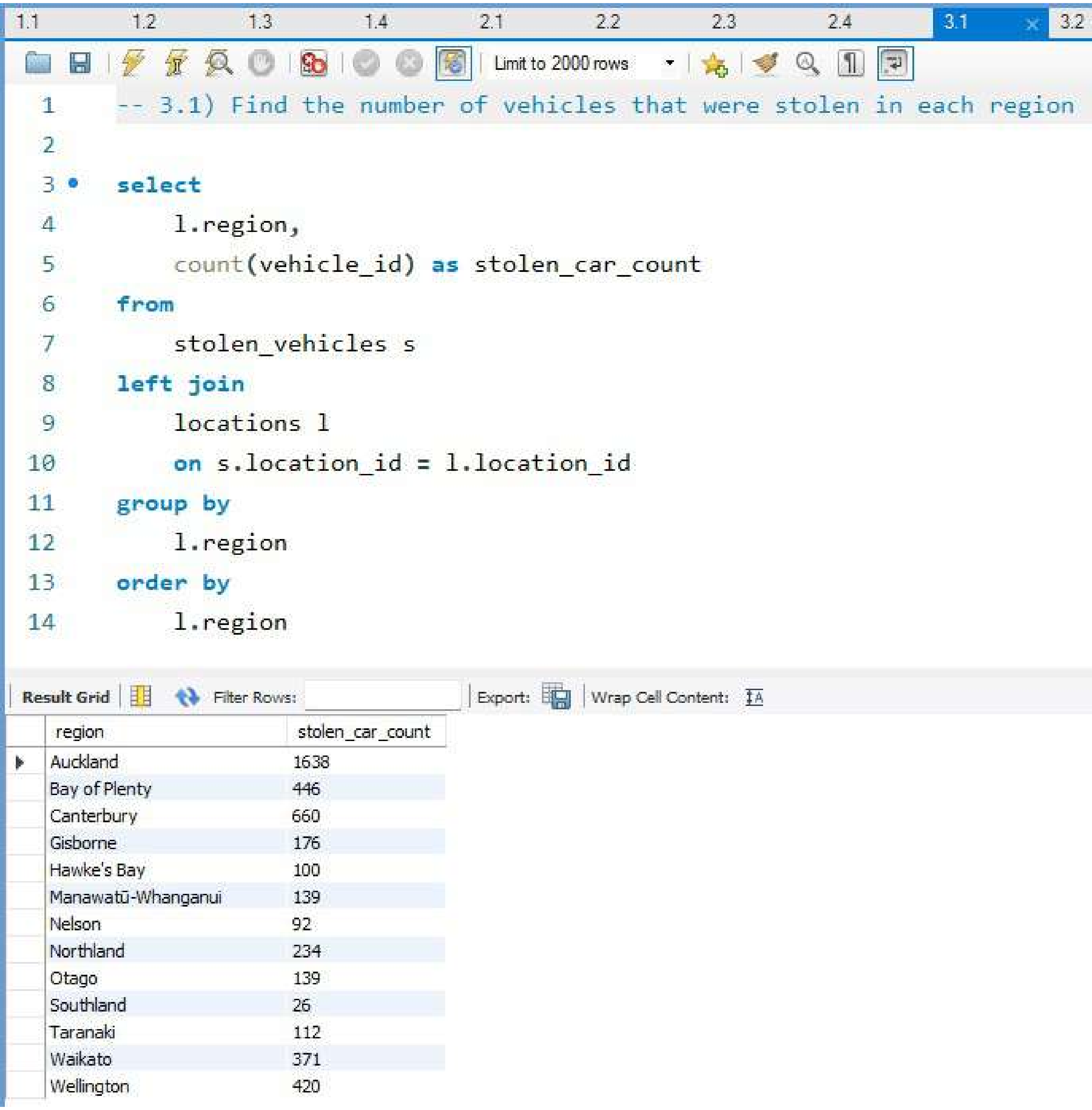
```
-- 2.3) For each vehicle type, find the percent of vehicles stolen that are luxury versus standard

select
    v.vehicle_type,
    round(sum(case when make_type = "Standard" then 1 else 0 end) / count(*) * 100, 2) as Standard,
    round(sum(case when make_type = "Luxury" then 1 else 0 end) / count(*) * 100, 2) as Luxury
from
    stolen_vehicles v
left join
    make_details m
    on v.make_id = m.make_id
group by
    v.vehicle_type
order by
    v.vehicle_type desc
```

vehicle_type	Standard	Luxury
Utility	99.79	0.21
Trailer - Heavy	100.00	0.00
Trailer	100.00	0.00
Trail Bike	100.00	0.00
Tractor	100.00	0.00
Stationwagon	96.30	3.70
Sports Car	77.50	22.50
Special Purpose Vehicle	100.00	0.00
Saloon	87.07	12.93
Roadbike	98.65	1.35



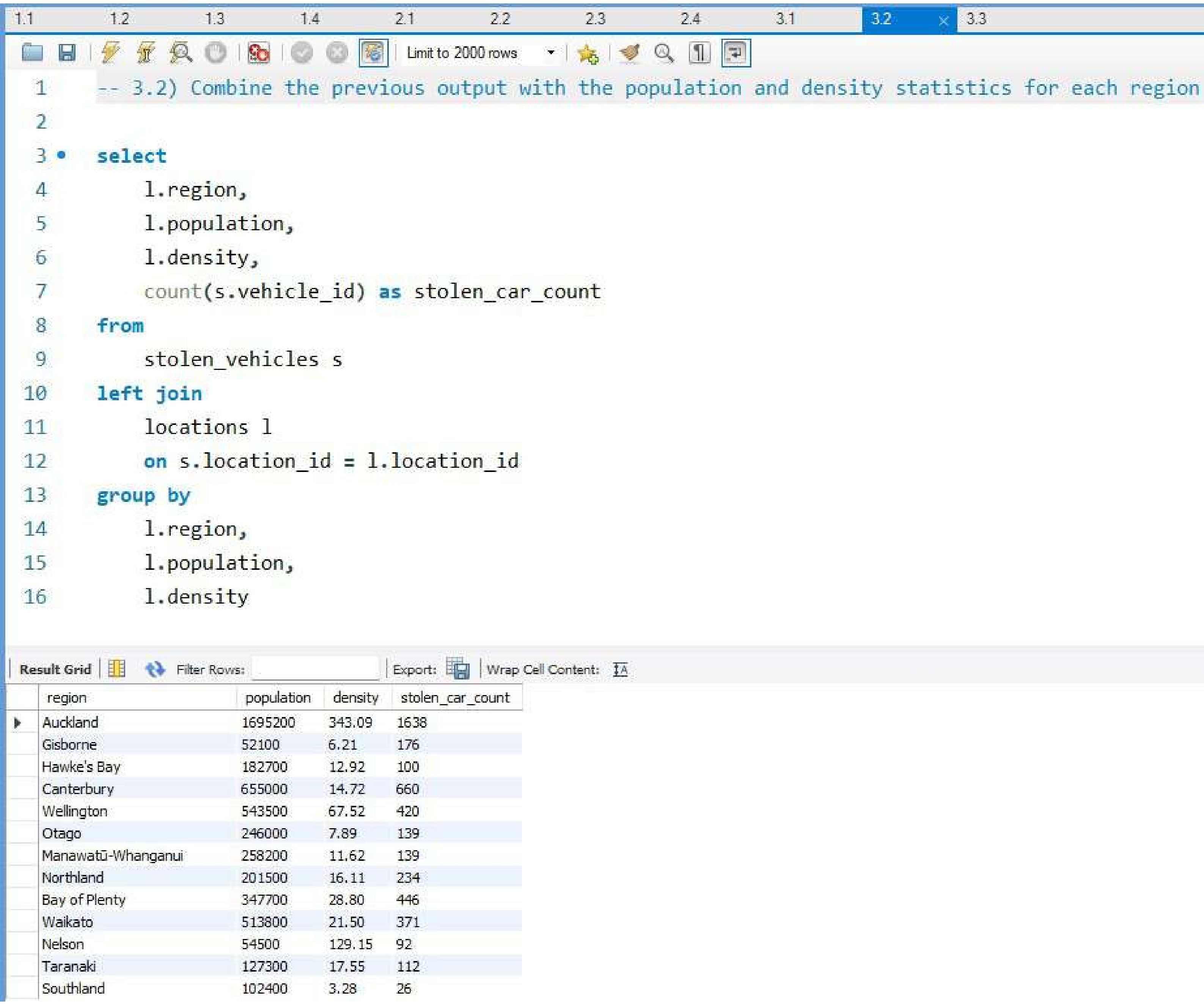
```
-- 2.4) Create a table where the rows represent the top 10 vehicle types, the columns represent the top 7 vehicle colors
vehicles stolen

select
    vehicle_type,
    sum(case when color = "Silver" then 1 else 0 end) as Silver,
    sum(case when color = "White" then 1 else 0 end) as White,
    sum(case when color = "Black" then 1 else 0 end) as Black,
    sum(case when color = "Blue" then 1 else 0 end) as Blue,
    sum(case when color = "Red" then 1 else 0 end) as Red,
    sum(case when color = "Grey" then 1 else 0 end) as Grey,
    sum(case when color = "Green" then 1 else 0 end) as Green,
    sum(case when color not in ("Silver", "White", "Black", "Blue", "Red", "Grey", "Green") then 1 else 0 end) as Others
from
    stolen_vehicles
group by
    vehicle_type
order by
    count(*) desc
limit 10
```

vehicle_type	Silver	White	Black	Blue	Red	Grey	Green	Others
Stationwagon	223	159	141	142	84	84	59	53
Saloon	226	160	99	125	75	71	52	43
Hatchback	172	114	76	104	58	46	24	50
Trailer	399	21	29	17	9	73	22	12
Utility	71	183	36	46	45	32	38	15
Roadbike	17	42	105	38	51	11	12	21
Moped	8	25	85	18	34	3	1	13
Light Van	19	104	0	5	7	7	6	6
Boat Trailer	67	5	3	0	0	29	0	1
Trailer - Heavy	53	9	1	3	0	12	2	10



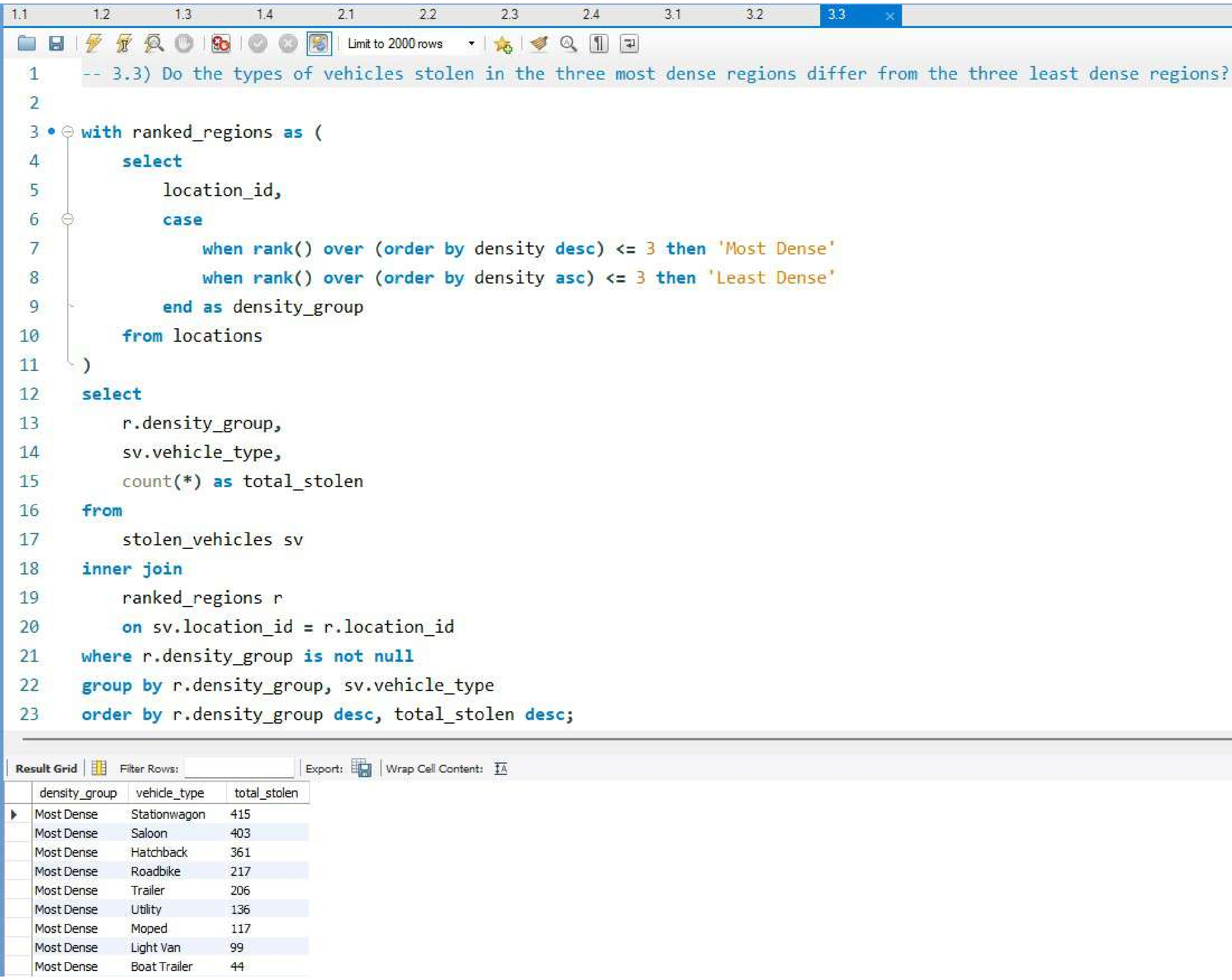
```sql
-- 3.1) Find the number of vehicles that were stolen in each region

select
    l.region,
    count(vehicle_id) as stolen_car_count
from
    stolen_vehicles s
left join
    locations l
    on s.location_id = l.location_id
group by
    l.region
order by
    l.region
```

region	stolen_car_count
Auckland	1638
Bay of Plenty	446
Canterbury	660
Gisborne	176
Hawke's Bay	100
Manawatū-Whanganui	139
Nelson	92
Northland	234
Otago	139
Southland	26
Taranaki	112
Waikato	371
Wellington	420



```
-- 3.2) Combine the previous output with the population and density statistics for each region

select
    l.region,
    l.population,
    l.density,
    count(s.vehicle_id) as stolen_car_count
from
    stolen_vehicles s
left join
    locations l
    on s.location_id = l.location_id
group by
    l.region,
    l.population,
    l.density
```

region	population	density	stolen_car_count
Auckland	1695200	343.09	1638
Gisborne	52100	6.21	176
Hawke's Bay	182700	12.92	100
Canterbury	655000	14.72	660
Wellington	543500	67.52	420
Otago	246000	7.89	139
Manawatū-Whanganui	258200	11.62	139
Northland	201500	16.11	234
Bay of Plenty	347700	28.80	446
Waikato	513800	21.50	371
Nelson	54500	129.15	92
Taranaki	127300	17.55	112
Southland	102400	3.28	26



```
-- 3.3) Do the types of vehicles stolen in the three most dense regions differ from the three least dense regions?

with ranked_regions as (
    select
        location_id,
        case
            when rank() over (order by density desc) <= 3 then 'Most Dense'
            when rank() over (order by density asc) <= 3 then 'Least Dense'
        end as density_group
    from locations
)
select
    r.density_group,
    sv.vehicle_type,
    count(*) as total_stolen
from
    stolen_vehicles sv
inner join
    ranked_regions r
    on sv.location_id = r.location_id
where r.density_group is not null
group by r.density_group, sv.vehicle_type
order by r.density_group desc, total_stolen desc;
```

density_group	vehicle_type	total_stolen
Most Dense	Stationwagon	415
Most Dense	Saloon	403
Most Dense	Hatchback	361
Most Dense	Roadbike	217
Most Dense	Trailer	206
Most Dense	Utility	136
Most Dense	Moped	117
Most Dense	Light Van	99
Most Dense	Boat Trailer	44