

NYC MV Collisions: Business Questions

Metrics:

1. Number of collision

```
SELECT COUNT(*) as total_collisions
FROM FCT_Collisions;
```



The screenshot shows a SQL Server Enterprise Manager interface. The top pane displays a query window with the following SQL code:

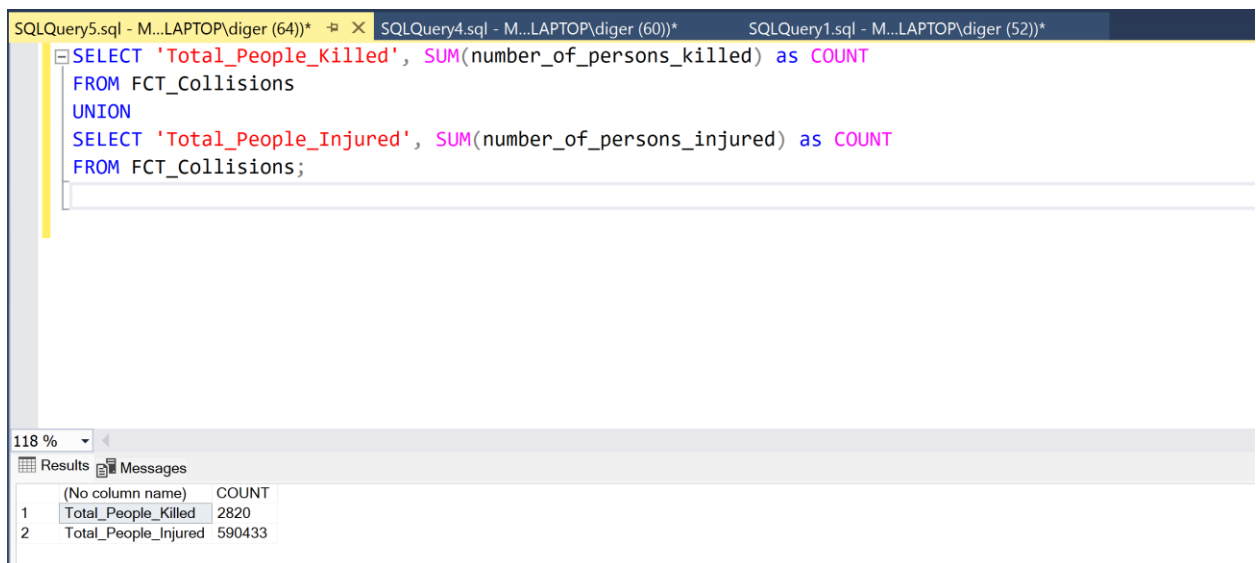
```
SELECT COUNT(*) as total_collisions
FROM FCT_Collisions;
```

The bottom pane shows the results of the query. The results are displayed in a table with one column, 'total_collisions', and one row with the value 1983443.

total_collisions
1983443

2. Number of people injured or died

```
SELECT 'Total_People_Killed', SUM(number_of_persons_killed) as COUNT
FROM FCT_Collisions
UNION
SELECT 'Total_People_Injured', SUM(number_of_persons_injured) as COUNT
FROM FCT_Collisions;
```



The screenshot shows a SQL Server Enterprise Manager interface. The top pane displays a query window with the following SQL code:

```
SELECT 'Total_People_Killed', SUM(number_of_persons_killed) as COUNT
FROM FCT_Collisions
UNION
SELECT 'Total_People_Injured', SUM(number_of_persons_injured) as COUNT
FROM FCT_Collisions;
```

The bottom pane shows the results of the query. The results are displayed in a table with two columns, 'COUNT' and 'COUNT', and two rows. The first row shows 'Total_People_Killed' with a value of 2820. The second row shows 'Total_People_Injured' with a value of 590433.

(No column name)	COUNT
Total_People_Killed	2820
Total_People_Injured	590433

3. Number of people by role, such as pedestrian, injured or died

```
SELECT 'Total_People_Injured', SUM(number_of_persons_injured) as COUNT
FROM FCT_Collisions
UNION
SELECT 'Total_People_Killed', SUM(number_of_persons_killed) as COUNT
FROM FCT_Collisions
UNION
select 'Total_Pedestrians_Injured', sum(number_of_pedestrians_injured) as COUNT
from FCT_Collisions
UNION
select 'Total_Pedestrians_Killed', sum(number_of_pedestrians_killed) as COUNT
from FCT_Collisions
UNION
select 'Total_Motorist_Injured', sum(number_of_motorist_injured) as COUNT
from FCT_Collisions
UNION
select 'Total_Motorist_Killed', sum(number_of_motorist_killed) as COUNT
from FCT_Collisions
union
select 'Total_Cyclist_Injured', sum(number_of_cyclists_injured) as COUNT
from FCT_Collisions
union
select 'Total_Cyclist_Killed', sum(number_of_cyclists_killed) as COUNT
from FCT_Collisions;
```

The screenshot shows a SQL Server Enterprise Manager interface. The top pane displays a SQL query using UNION to aggregate counts from the FCT_Collisions table. The bottom pane shows the results of the query, which consists of 8 rows of data. The status bar at the bottom indicates the query was executed successfully.

```

SELECT 'Total_People_Injured', SUM(number_of_persons_injured) as COUNT
FROM FCT_Collisions
UNION
SELECT 'Total_People_Killed', SUM(number_of_persons_killed) as COUNT
FROM FCT_Collisions
UNION
select 'Total_Pedestrians_Injured', sum(number_of_pedestrians_injured) as COUNT
from FCT_Collisions
UNION
select 'Total_Pedestrians_Killed', sum(number_of_pedestrians_killed) as COUNT
from FCT_Collisions
UNION
select 'Total_Motorist_Injured', sum(number_of_motorist_injured) as COUNT
from FCT_Collisions
UNION
select 'Total_Motorist_Killed', sum(number_of_motorist_killed) as COUNT
from FCT_Collisions
union
select 'Total_Cyclist_Injured', sum(number_of_cyclists_injured) as COUNT
from FCT_Collisions

```

	(No column name)	COUNT
1	Total_People_Injured	590433
2	Total_People_Killed	2820
3	Total_Pedestrians_Injured	108521
4	Total_Pedestrians_Killed	1431
5	Total_Motorist_Injured	425971
6	Total_Motorist_Killed	1138
7	Total_Cyclist_Killed	215
8	Total_Cyclist_Injured	50661

Query executed successfully. MYLAPTOP (15.0 RTM) MYLAPTOP\diger (64) NYC_MV_Collision 00:00:00 8 rows

4. Number of vehicles involved

```

select count(a.Vehicle_SK) from dbo.Dim_Vehicle a, dbo.fct_person_vehicle b,
dbo.FCT_Collisions c
where a.Vehicle_SK = b.vehicle_sk
and c.Collision_SK = b.collision_sk

```

Dimensions:

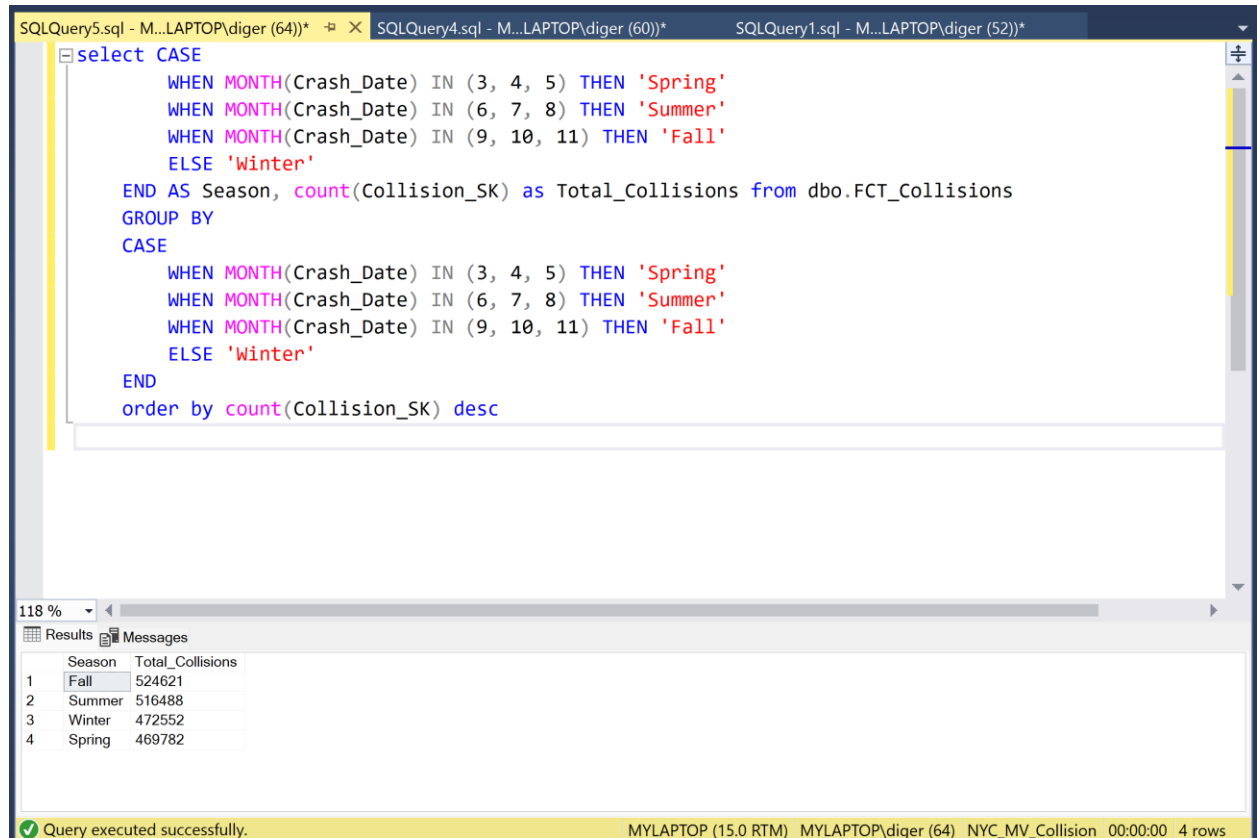
1. Trend (granularity month, year), also seasonality, i.e., Spring, Summer, Fall and Winter

```

select CASE
    WHEN MONTH(Crash_Date) IN (3, 4, 5) THEN 'Spring'
    WHEN MONTH(Crash_Date) IN (6, 7, 8) THEN 'Summer'
    WHEN MONTH(Crash_Date) IN (9, 10, 11) THEN 'Fall'
    ELSE 'Winter'
END AS Season, count(Collision_SK) as Total_Collisions from
dbo.FCT_Collisions
GROUP BY
CASE
    WHEN MONTH(Crash_Date) IN (3, 4, 5) THEN 'Spring'
    WHEN MONTH(Crash_Date) IN (6, 7, 8) THEN 'Summer'
    WHEN MONTH(Crash_Date) IN (9, 10, 11) THEN 'Fall'

```

```
ELSE 'Winter'  
END  
order by count(Collision_SK) desc
```



The screenshot shows a SQL Server Enterprise Manager window with three tabs: SQLQuery5.sql, SQLQuery4.sql, and SQLQuery1.sql. The active tab, SQLQuery5.sql, contains the following SQL query:

```
select CASE  
    WHEN MONTH(Crash_Date) IN (3, 4, 5) THEN 'Spring'  
    WHEN MONTH(Crash_Date) IN (6, 7, 8) THEN 'Summer'  
    WHEN MONTH(Crash_Date) IN (9, 10, 11) THEN 'Fall'  
    ELSE 'Winter'  
END AS Season, count(Collision_SK) as Total_Collisions from dbo.FCT_Collisions  
GROUP BY  
CASE  
    WHEN MONTH(Crash_Date) IN (3, 4, 5) THEN 'Spring'  
    WHEN MONTH(Crash_Date) IN (6, 7, 8) THEN 'Summer'  
    WHEN MONTH(Crash_Date) IN (9, 10, 11) THEN 'Fall'  
    ELSE 'Winter'  
END  
order by count(Collision_SK) desc
```

Below the query editor, the 'Results' pane shows the output of the query. The results are displayed in a table with two columns: 'Season' and 'Total_Collisions'. The table contains four rows of data:

	Season	Total_Collisions
1	Fall	524621
2	Summer	516488
3	Winter	472552
4	Spring	469782

The status bar at the bottom of the window indicates that the query was executed successfully. The status bar also shows the server name 'MYLAPTOP (15.0 RTM)', the database name 'MYLAPTOP\diqer (64)', the table name 'NYC_MV_Collision', the execution time '00:00:00', and the number of rows returned '4 rows'.

2. Annual statistics

```
select year(Crash_Date) as Year, count(Collision_SK) as Total_Collisions from  
dbo.FCT_Collisions  
group by year(Crash_Date)  
order by year(Crash_Date)
```

SQLQuery5.sql - M...LAPTOP\diger (64)* SQLQuery4.sql - M...LAPTOP\diger (60)* SQLQuery1.sql - M...LAPTOP\diger (52)*

```

select year(Crash_Date) as Year, count(Collision_SK) as Total_Collisions from dbo.FCT_Collisions
group by year(Crash_Date)
order by year(Crash_Date)

```

118 %

Results Messages

	Year	Total_Collisions
1	2012	100545
2	2013	203734
3	2014	206033
4	2015	217694
5	2016	229831
6	2017	231007
7	2018	231564
8	2019	211486
9	2020	112916
10	2021	110546
11	2022	103780
12	2023	24307

Query executed successfully. MYLAPTOP (15.0 RTM) MYLAPTOP\diger (64) NYC_MV_Collision 00:00:00 12 rows

3. Time of day (hour granularity)

```

select CASE
    WHEN convert(int, substring(Crash_Time, 1, 2)) >= 0 AND convert(int,
substring(Crash_Time, 1, 2)) < 6 THEN 'Late Night'
    WHEN convert(int, substring(Crash_Time, 1, 2)) >= 6 AND convert(int,
substring(Crash_Time, 1, 2)) < 12 THEN 'Morning'
    ELSE 'Winter'
    END AS Time_Of_Day, count(Collision_SK) as Total_Collisions from
FCT_Collisions
group by CASE
    WHEN convert(int, substring(Crash_Time, 1, 2)) >= 0 AND convert(int,
substring(Crash_Time, 1, 2)) < 6 THEN 'Late Night'
    WHEN convert(int, substring(Crash_Time, 1, 2)) >= 6 AND convert(int,
substring(Crash_Time, 1, 2)) < 12 THEN 'Morning'
    ELSE 'Winter'
    END
order by count(Collision_SK) desc

```

SQLQuery5.sql - M...LAPTOP\diger (64))* SQLQuery4.sql - M...LAPTOP\diger (60))* SQLQuery1.sql - M...LAPTOP\diger (52))*

```

select CASE
    WHEN convert(int, substring(Crash_Time, 1, 2)) >= 0 AND convert(int, substring(Crash_Time, 1, 2)) < 6 AND convert(int, substring(Crash_Time, 1, 2)) < 6 AND convert(int, substring(Crash_Time, 1, 2)) < 6
    WHEN convert(int, substring(Crash_Time, 1, 2)) >= 6 AND convert(int, substring(Crash_Time, 1, 2)) < 6 AND convert(int, substring(Crash_Time, 1, 2)) < 6
    ELSE 'Winter'
END AS Time_Of_Day, count(Collision_SK) as Total_Collisions from FCT_Collisions
group by CASE
    WHEN convert(int, substring(Crash_Time, 1, 2)) >= 0 AND convert(int, substring(Crash_Time, 1, 2)) < 6 AND convert(int, substring(Crash_Time, 1, 2)) < 6
    WHEN convert(int, substring(Crash_Time, 1, 2)) >= 6 AND convert(int, substring(Crash_Time, 1, 2)) < 6 AND convert(int, substring(Crash_Time, 1, 2)) < 6
    ELSE 'Winter'
END
order by count(Collision_SK) desc

```

118 %

Results Messages

	Time_Of_Day	Total_Collisions
1	Winter	1264888
2	Morning	520073
3	Late Night	198482

Query executed successfully. MYLAPTOP (15.0 RTM) MYLAPTOP\diger (64) NYC_MV_Collision 00:00:00 3 rows

4. Vehicle Types

```

select c.Vehicle_Type, count(a.Collision_SK) as Total_Collisions from
FCT_Collisions a, fct_person_vehicle b, Dim_Vehicle c
where a.Collision_SK = b.Collision_SK
and b.Vehicle_SK = c.Vehicle_SK
order by Total_Collisions desc

```

SQLQuery4.sql - DE...P-CJ6OCRS\Tj (65))* SQLQuery3.sql - DE...P-CJ6OCRS\Tj (52))* SQLQuery2.sql - DE...P-CJ6OCRS\Tj (67))* SQLQuery1.sql - DE...P-CJ6OCRS\Tj (64))*

```

select c.Vehicle_Type, count(a.Collision_SK) as Total_Collisions from FCT_Collisions a, fct_person_vehicle b, Dim_Vehicle
where a.Collision_SK = b.Collision_SK
and b.Vehicle_SK = c.Vehicle_SK
group by c.Vehicle_Type
order by Total_Collisions desc

```

119 %

Results Messages

	Vehicle_Type	Total_Collisions
1	Sedan	1682757
2	Station Wagon/Sport Utility Vehicle	1393856
3	UNKNOWN	283674
4	TAXI	178611
5	4 dr sedan	151554
6	PICK-UP TRUCK	126235
7	Box Truck	96819
8	PASSENGER VEHICLE	95587
9	BUS	95270
10	Bike	84916
11	SPORT UTILITY / STATION WAGON	41978
12	VAN	40973
13	Tractor Truck Diesel	37060
14	MOTORCYCLE	26591
15	Dump	15196
16	BICYCLE	14794
17	Construction Vehicle	10603

Query executed successfully. DESKTOP-CJ6OCRS (15.0 RTM) DESKTOP-CJ6OCRS\Tj (65) NYC_MV_COLLISIONS 00:00:01 1,073 rows

5. Collision Causes

```

select a.Contributing_Factor_1 as Collision_Causes, count(b.Collision_SK) as
Total_Collisions from Dim_Collision_Contributing_Factors a, FCT_Collisions b
where a.Collision_Contributing_Factor_SK = b.Collision_Contributing_Factor_SK
group by a.Contributing_Factor_1
order by Total_Collisions desc

```

The screenshot shows a SQL Server Enterprise Manager window with three tabs: SQLQuery5.sql, SQLQuery4.sql, and SQLQuery1.sql. The active tab is SQLQuery5.sql, which contains the following SQL query:

```
select a.Contributing_Factor_1 as Collision_Causes, count(b.Collision_SK) as Total_Collisions from Di
where a.Collision_Contributing_Factor_SK = b.Collision_Contributing_Factor_SK
group by a.Contributing_Factor_1
order by Total_Collisions desc
```

Below the query window, the Results pane shows a table with two columns: Collision_Causes and Total_Collisions. The table contains 14 rows of data, sorted in descending order of Total_Collisions.

Collision_Causes	Total_Collisions
Unspecified	690726
Driver Inattention/Distracted	392450
Failure to Yield Right-of-Way	116956
Following Too Closely	105156
Backing Unsafely	73987
Other Vehicular	61623
Passing or Lane Usage Improper	53882
Turning Improperly	49114
Passing Too Closely	48691
Fatigued/Drowsy	47314
Unsafe Lane Changing	38978
Traffic Control Disregarded	34137
Driver Inexperience	30464
Unsafe Speed	25480

At the bottom of the window, a status bar indicates: "Query executed successfully. MYLAPTOP (15.0 RTM) MYLAPTOP\diger (64) NYC_MV_Collision 00:00:00 59 rows".

Business Questions:

1. How Many Car Accidents Are There in NYC Every Year?

Select YEAR(crash_date) as Year, count(Collision_SK) as Total_Collisions from
 dbo.FCT_Collisions
 group by YEAR(crash_date)
 order by YEAR(crash_date)

The screenshot shows a SQL Server Enterprise Manager window with three tabs: 'SQLQuery5.sql - M...LAPTOP\diger (64))*', 'SQLQuery4.sql - M...LAPTOP\diger (60))*', and 'SQLQuery1.sql - M...LAPTOP\diger (52))*'. The active tab displays the following SQL query:

```
Select YEAR(crash_date) as Year, count(Collision_SK) as Total_Collisions from dbo.FCT_Collisions  
group by YEAR(crash_date)  
order by YEAR(crash_date)
```

Below the query editor, the 'Results' tab is selected, showing a table with 12 rows and 2 columns: 'Year' and 'Total_Collisions'. The data is as follows:

	Year	Total_Collisions
1	2012	100545
2	2013	203734
3	2014	206033
4	2015	217694
5	2016	229831
6	2017	231007
7	2018	231564
8	2019	211486
9	2020	112916
10	2021	110546
11	2022	103780
12	2023	24307

At the bottom of the window, a status bar indicates: 'Query executed successfully. MYLAPTOP (15.0 RTM) MYLAPTOP\diger (64) NYC_MV_Collision 00:00:00 12 rows'.

2. Which Boroughs in New York City Have the Most Accidents?

```
SELECT borough, count(Collision_SK) as Total_Collisions  
FROM Dim_Accident_Location a, dbo.FCT_Collisions b  
WHERE a.Accident_Location_SK = b.Accident_Location_SK  
GROUP BY borough  
ORDER BY Total_Collisions DESC;
```

The screenshot shows a SQL Server Enterprise Manager window with three tabs: 'SQLQuery5.sql - M...LAPTOP\diger (64))*', 'SQLQuery4.sql - M...LAPTOP\diger (60))*', and 'SQLQuery1.sql - M...LAPTOP\diger (52))*'. The active tab 'SQLQuery5.sql' contains the following SQL query:

```
SELECT borough, count(Collision_SK) as Total_Collisions
FROM Dim_Accident_Location a, dbo.FCT_Collisions b
WHERE a.Accident_Location_SK = b.Accident_Location_SK
GROUP BY borough
ORDER BY Total_Collisions DESC;
```

Below the query editor, the 'Results' pane shows the output of the query. The zoom level is set to 118%. The results are displayed in a table with two columns: 'borough' and 'Total_Collisions'. The data is as follows:

borough	Total_Collisions
N/A	612074
BROOKLYN	432359
QUEENS	365961
MANHATTAN	308432
BRONX	201228
STATEN ISLAND	57288

At the bottom of the window, a status bar indicates 'Query executed successfully.' and 'MYLAPTOP (15.0 RTM) MYLAPTOP\diger (64) NYC_MV_Collision 00:00:00 6 rows'.

3. How Many NYC Car Accidents Result in an Injury?

```
SELECT COUNT(Collision_SK) as Total_Injuries
FROM FCT_Collisions
WHERE Number_Of_Cyclists_Injured > 0
OR number_of_motorist_injured > 0
OR number_of_pedestrians_injured > 0
OR number_of_persons_injured > 0
```

The screenshot shows a SQL Server Enterprise Manager window with three tabs: 'SQLQuery5.sql - M...LAPTOP\diger (64))*', 'SQLQuery4.sql - M...LAPTOP\diger (60))*', and 'SQLQuery1.sql - M...LAPTOP\diger (52))*'. The active tab displays the following SQL query:

```
SELECT COUNT(Collision_SK) as Total_Injuries
FROM FCT_Collisions
WHERE Number_Of_Cyclists_Injured > 0
OR number_of_motorist_injured > 0
OR number_of_pedestrians_injured > 0
OR number_of_persons_injured > 0
```

Below the query editor, the 'Results' pane shows a single row of data:

	Total_Injuries
1	435470

The status bar at the bottom indicates 'Query executed successfully.' and 'MYLAPTOP (15.0 RTM) MYLAPTOP\diger (64) NYC_MV_Collision 00:00:00 1 rows'.

4. Which NYC Borough Has the Most Fatal Car Accidents?

```
SELECT COUNT(Collision_SK) as Total_Fatalities
FROM FCT_Collisions
WHERE number_of_cyclists_killed > 0
OR number_of_motorist_killed > 0
OR number_of_pedestrians_killed > 0
OR number_of_persons_killed > 0
```

The screenshot shows a SQL Server Enterprise Manager window with three tabs: 'SQLQuery5.sql - M...LAPTOP\diger (64))*', 'SQLQuery4.sql - M...LAPTOP\diger (60))*', and 'SQLQuery1.sql - M...LAPTOP\diger (52))*'. The active tab displays the following SQL query:

```
SELECT COUNT(Collision_SK) as Total_Fatalities
FROM FCT_Collisions
WHERE number_of_cyclists_killed > 0
OR number_of_motorist_killed > 0
OR number_of_pedestrians_killed > 0
OR number_of_persons_killed > 0
```

Below the query editor, the 'Results' pane shows a single row of data:

	Total_Fatalities
1	2717

The status bar at the bottom indicates: 'Query executed successfully. MYLAPTOP (15.0 RTM) MYLAPTOP\diger (64) NYC_MV_Collision 00:00:00 1 rows'.

5. When Do Most New York City Car Accidents Happen?

```
select month(Crash_Date) as MONTH, count(Collision_SK) as Total_Collisions from
dbo.FCT_Collisions
group by month(Crash_Date)
order by month(Crash_Date);
```

SQLQuery5.sql - M...LAPTOP\diger (64))* SQLQuery4.sql - M...LAPTOP\diger (60))* SQLQuery1.sql - M...LAPTOP\diger (52))*

```
select month(Crash_Date) as MONTH, count(Collision_SK) as Total_Collisions from dbo.FCT_Collisions
group by month(Crash_Date)
order by month(Crash_Date);
```

118 %

Results Messages

	MONTH	Total_Collisions
1	1	156531
2	2	144640
3	3	161663
4	4	144543
5	5	163576
6	6	165560
7	7	176638
8	8	174290
9	9	174108
10	10	179838
11	11	170675
12	12	171381

Query executed successfully. MYLAPTOP (15.0 RTM) MYLAPTOP\diger (64) NYC_MV_Collision 00:00:00 12 rows

```
select Crash_Time, count(Collision_SK) as Total_Collisions from FCT_Collisions
group by Crash_Time
order by Crash_Time;
```

SQLQuery5.sql - M...LAPTOP\diger (64))* SQLQuery4.sql - M...LAPTOP\diger (60))* SQLQuery1.sql - M...LAPTOP\diger (52))*

```

select Crash_Time, count(Collision_SK) as Total_Collisions from FCT_Collisions
group by Crash_Time
order by Crash_Time;

```

118 %

Results Messages

	Crash_Time	Total_Collisions
1	00:00	19190
2	00:01	2473
3	00:02	519
4	00:03	335
5	00:04	330
6	00:05	2971
7	00:06	308
8	00:07	301
9	00:08	351
10	00:09	313
11	00:10	2804
12	00:11	264
13	00:12	387
14	00:13	306
15	00:14	262
16	00:15	3246
17	00:16	281
18	00:17	333
19	00:18	292
20	00:19	283
21	00:20	2689
22	00:21	284

Query executed successfully. MYLAPTOP (15.0 RTM) MYLAPTOP\diger (64) NYC_MV_Collision 00:00:00 1,440 rows

6. How Common Are Bicycle Accidents in NYC?

```

SELECT COUNT(Collision_SK) as Total_Bicycle_accidents
FROM FCT_Collisions a, Dim_Collision_Vehicle_Types b
WHERE a.Collision_Vehicle_Types_SK = b.Collision_Vehicle_Types_SK
and (b.Vehicle_Type_Code_1 = 'Bicycle' OR b.Vehicle_Type_Code_2 = 'Bicycle' OR
b.Vehicle_Type_Code_3 = 'Bicycle' OR b.Vehicle_Type_Code_4 = 'Bicycle' OR
b.Vehicle_Type_Code_5 = 'Bicycle')

```

The screenshot shows a SQL query window with the following text:

```

SELECT COUNT(Collision_SK) as Total_Bicycle_accidents
FROM FCT_Collisions a, Dim_Collision_Vehicle_Types b
WHERE a.Collision_Vehicle_Types_SK = b.Collision_Vehicle_Types_SK
and (b.Vehicle_Type_Code_1 = 'Bicycle' OR b.Vehicle_Type_Code_2 = 'Bicycle' OR b.Vehicle_Type_Code_3

```

A tooltip for the column `Vehicle_Type_Code_1` shows its data type as `varchar, null`.

The Results pane shows a single row with the value 19256 for the column `Total_Bicycle_accidents`.

The status bar at the bottom indicates: "Query executed successfully. MYLAPTOP (15.0 RTM) MYLAPTOP\diger (64) NYC_MV_Collision 00:00:00 1 rows".

7. How Often Are Pedestrians Involved New York Traffic Accidents?

```

SELECT COUNT(a.Collision_SK) as Collisions_Involving_Pedestrians_Count
FROM FCT_Collisions a, fct_person_vehicle b, Dim_Person_Type c
WHERE a.Collision_SK = b.Collision_SK
and b.Person_Type_SK = c.Person_Type_SK
and c.Person_Type = 'Pedestrian'

```

The screenshot shows a SQL query window with the following text:

```

SELECT COUNT(a.Collision_SK) as Collisions_Involving_Pedestrians_Count
FROM FCT_Collisions a, fct_person_vehicle b, Dim_Person_Type c
WHERE a.Collision_SK = b.Collision_SK
and b.Person_Type_SK = c.Person_Type_SK
and c.Person_Type = 'Pedestrian'

```

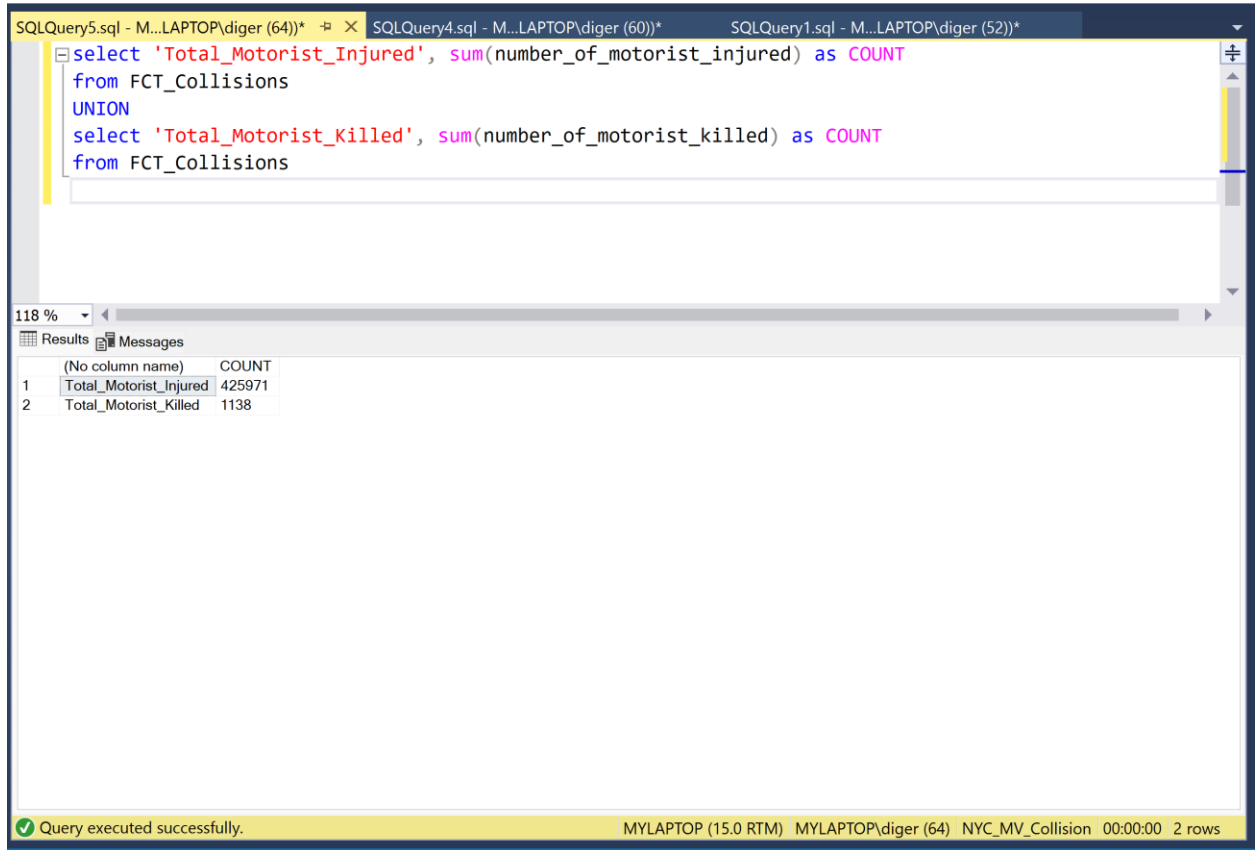
The Results pane shows a single row with the value 112013 for the column `Collisions_Involving_Pedestrians_Count`.

8. How Many Motorcyclists are Injured or Killed in NYC Accidents?

```

select 'Total_Motorist_Injured', sum(number_of_motorist_injured) as COUNT
from FCT_Collisions
UNION
select 'Total_Motorist_Killed', sum(number_of_motorist_killed) as COUNT
from FCT_Collisions

```



9. Are Trucks involved in many New York Accidents?

```

select count(a.Collision_SK) as Trucks_Involved_Count from dbo.FCT_Collisions a,
dbo.FCT_Person_Vehicle b, dbo.Dim_Vehicle c
where a.Collision_SK = b.Collision_SK
and c.Vehicle_SK = b.Vehicle_SK
and upper(c.Vehicle_Make) like '%TRUCK%'

```



```
SELECT COUNT(a.Collision_SK) as Collisions_Involving_Pedestrians_Count
FROM FCT_Collisions a, fct_person_vehicle b, Dim_Person_Type c
WHERE a.Collision_SK = b.Collision_SK
and b.Person_Type_SK = c.Person_Type_SK
and c.Person_Type = 'Pedestrian'
```

	Collisions_Involving_Pedestrians_Count
1	112013