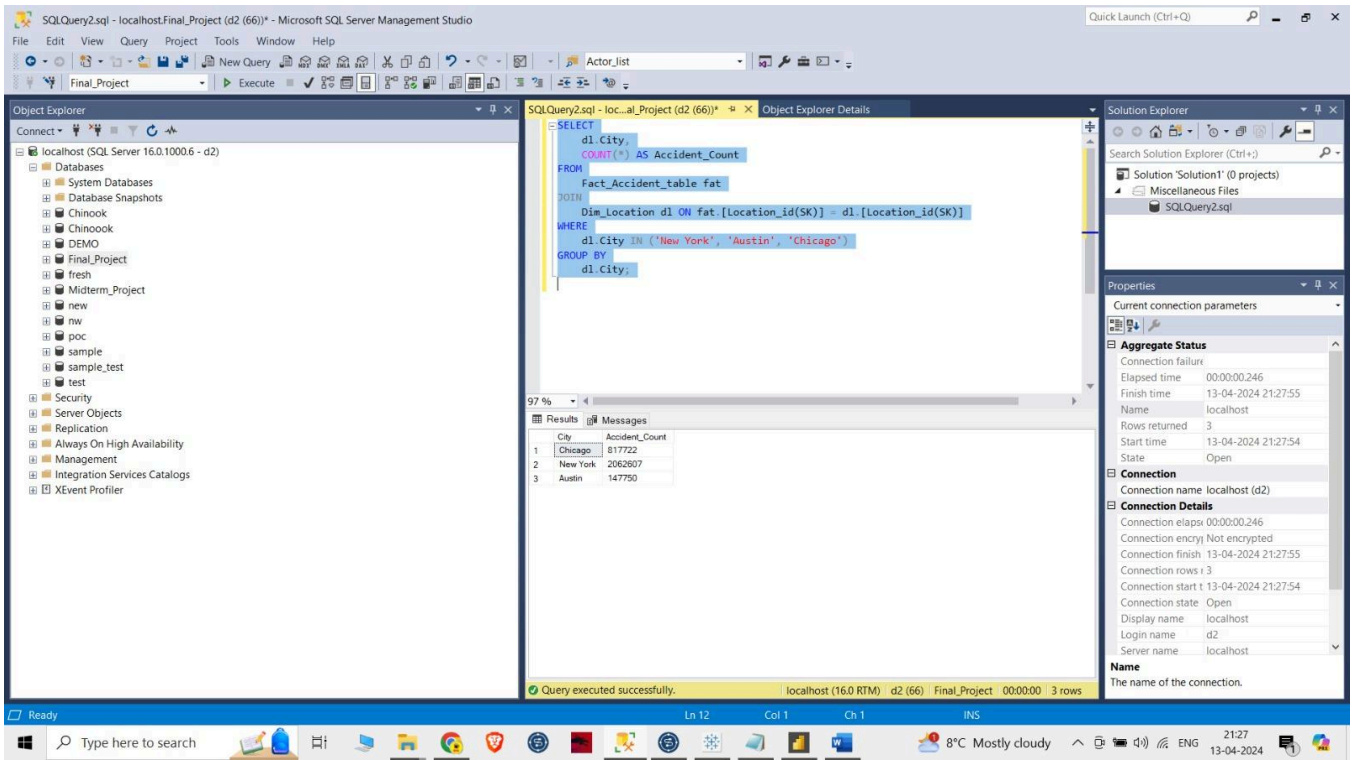
Validation Queries for Given Questions

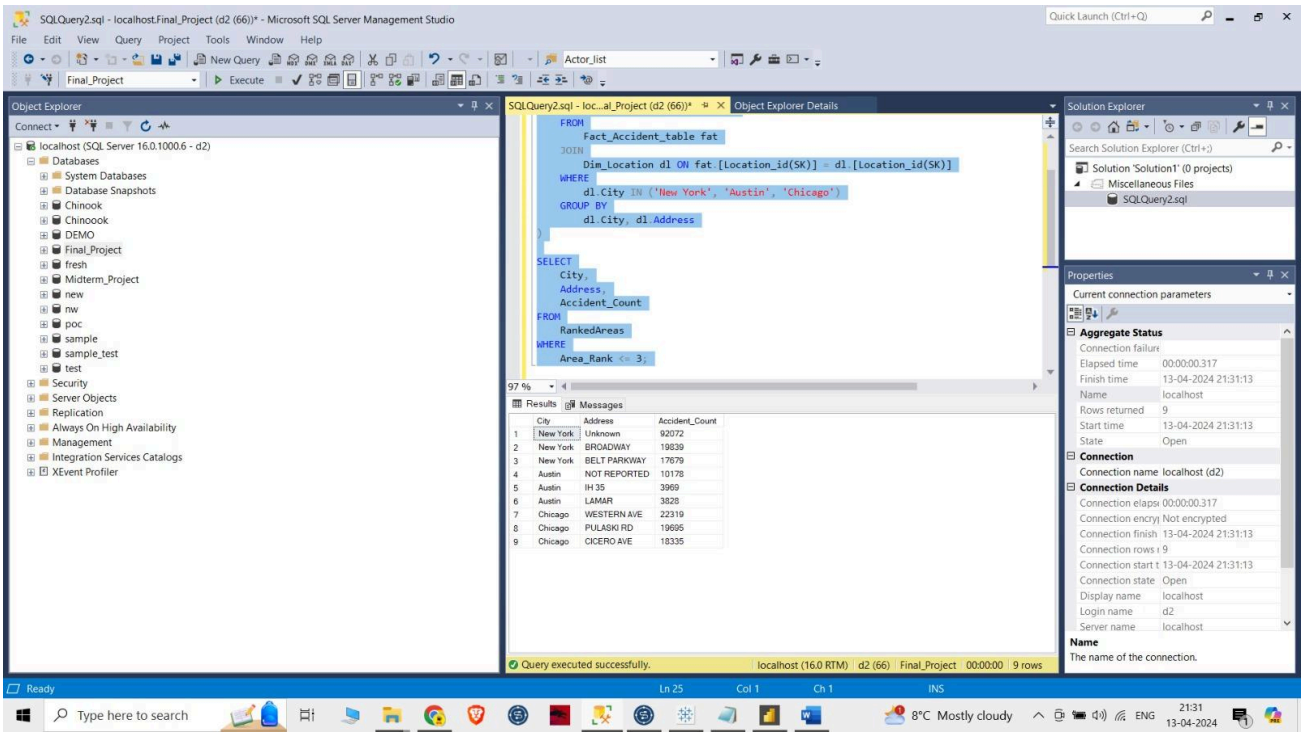
1. **How many accidents occurred in NYC, Austin, and Chicago?**

```
SELECT
    dl.City,
    COUNT(*) AS Accident_Count
FROM
    Fact_Accident_table fat
JOIN
    Dim_Location dl ON fat.[Location_id(SK)] = dl.[Location_id(SK)]
WHERE
    dl.City IN ('New York', 'Austin', 'Chicago')
GROUP BY
    dl.City;
```



2. Which areas in the 3 cities had the greatest number of accidents?
top 3 areas in each city

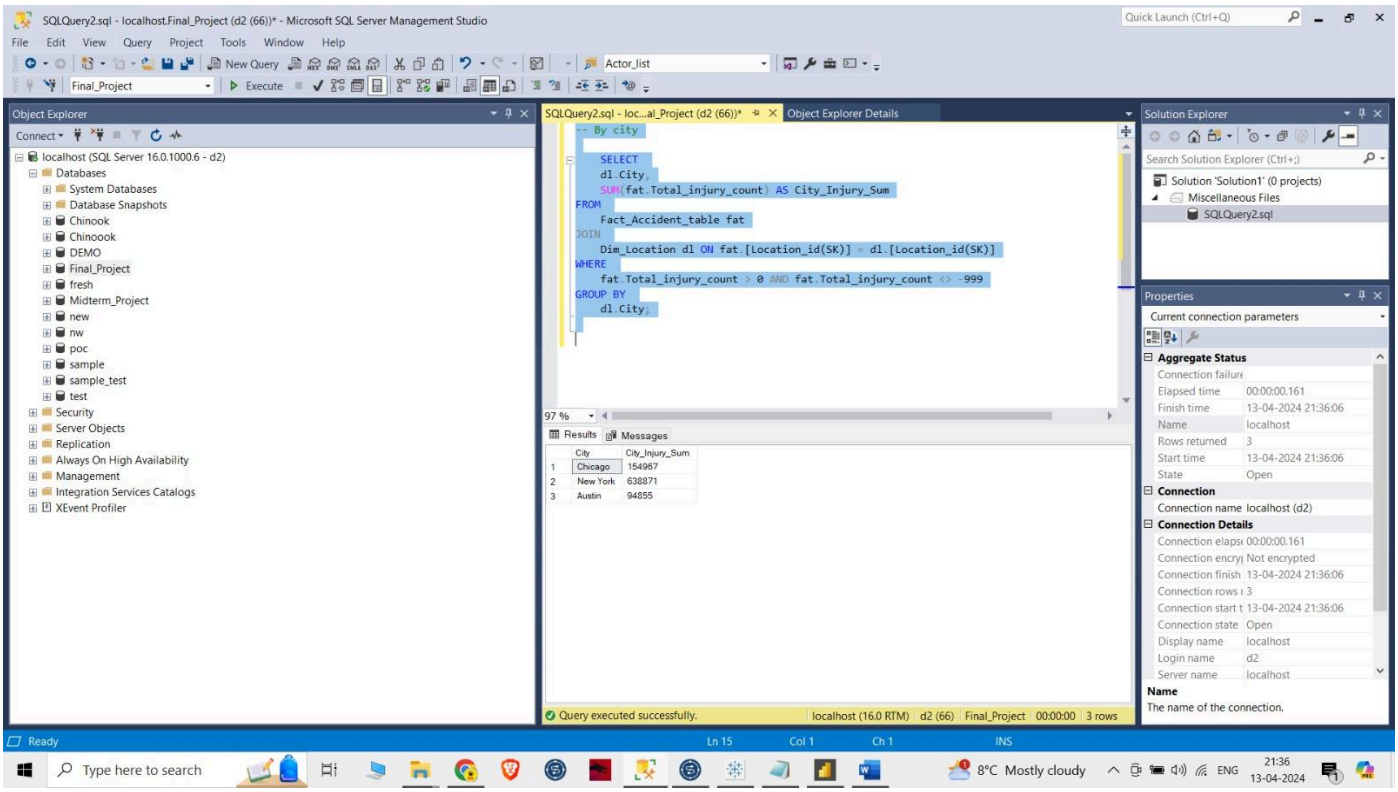
```
WITH RankedAreas AS (
    SELECT
        dl.City,
        dl.Address,
        ROW_NUMBER() OVER (PARTITION BY dl.City ORDER BY COUNT(*) DESC) AS Area_Rank,
        COUNT(*) AS Accident_Count
    FROM
        Fact_Accident_table fat
    JOIN
        Dim_Location dl ON fat.[Location_id(SK)] = dl.[Location_id(SK)]
    WHERE
        dl.City IN ('New York', 'Austin', 'Chicago')
    GROUP BY
        dl.City, dl.Address
)
SELECT
    City,
    Address,
    Accident_Count
FROM
    RankedAreas
WHERE
    Area_Rank <= 3;
```



3. **How many accidents resulted in just injuries?**
   this report needs to be generated at 2 levels, 1 -> overall, 3 -> by city

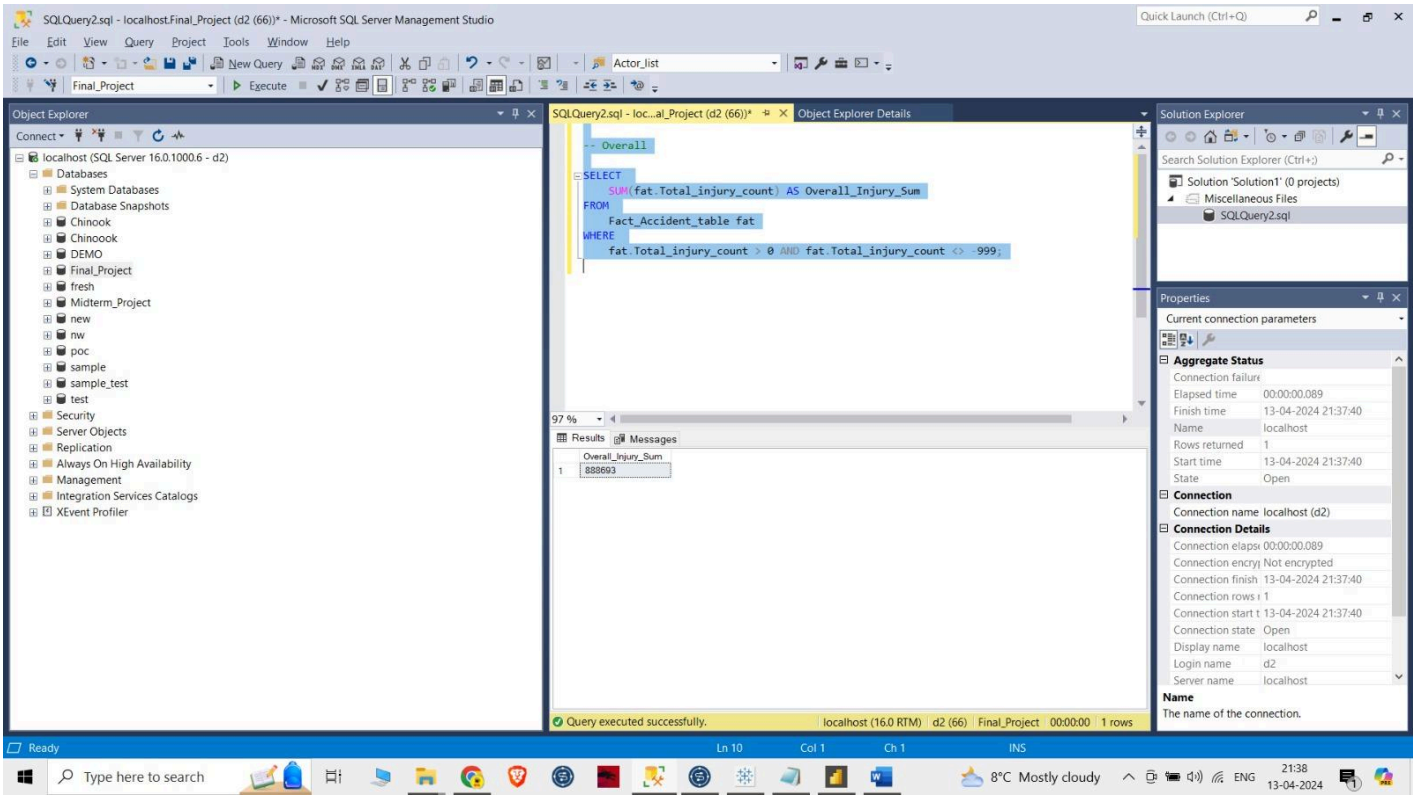
```
-- By city

    SELECT
    dl.City,
    SUM(fat.Total_injury_count) AS City_Injury_Sum
FROM
    Fact_Accident_table fat
JOIN
    Dim_Location dl ON fat.[Location_id(SK)] = dl.[Location_id(SK)]
WHERE
    fat.Total_injury_count > 0 AND fat.Total_injury_count <> -999
GROUP BY
    dl.City;
```



```sql
-- Overall

SELECT
    SUM(fat.Total_injury_count) AS Overall_Injury_Sum
FROM
    Fact_Accident_table fat
WHERE
    fat.Total_injury_count > 0 AND fat.Total_injury_count <> -999;
```

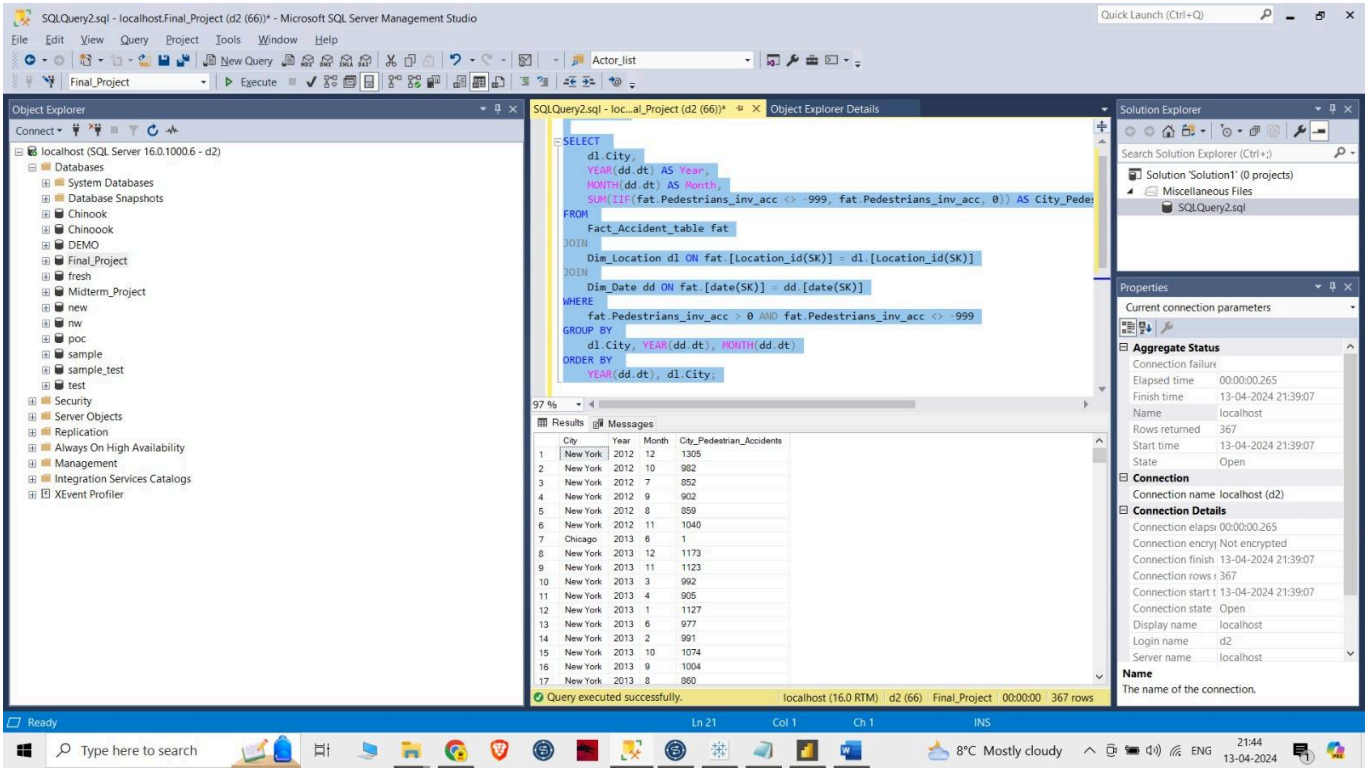


4. **How often are pedestrians involved in accidents?**

   this report needs to be generated at 2 levels, 1 -> overall, 3 -> by city

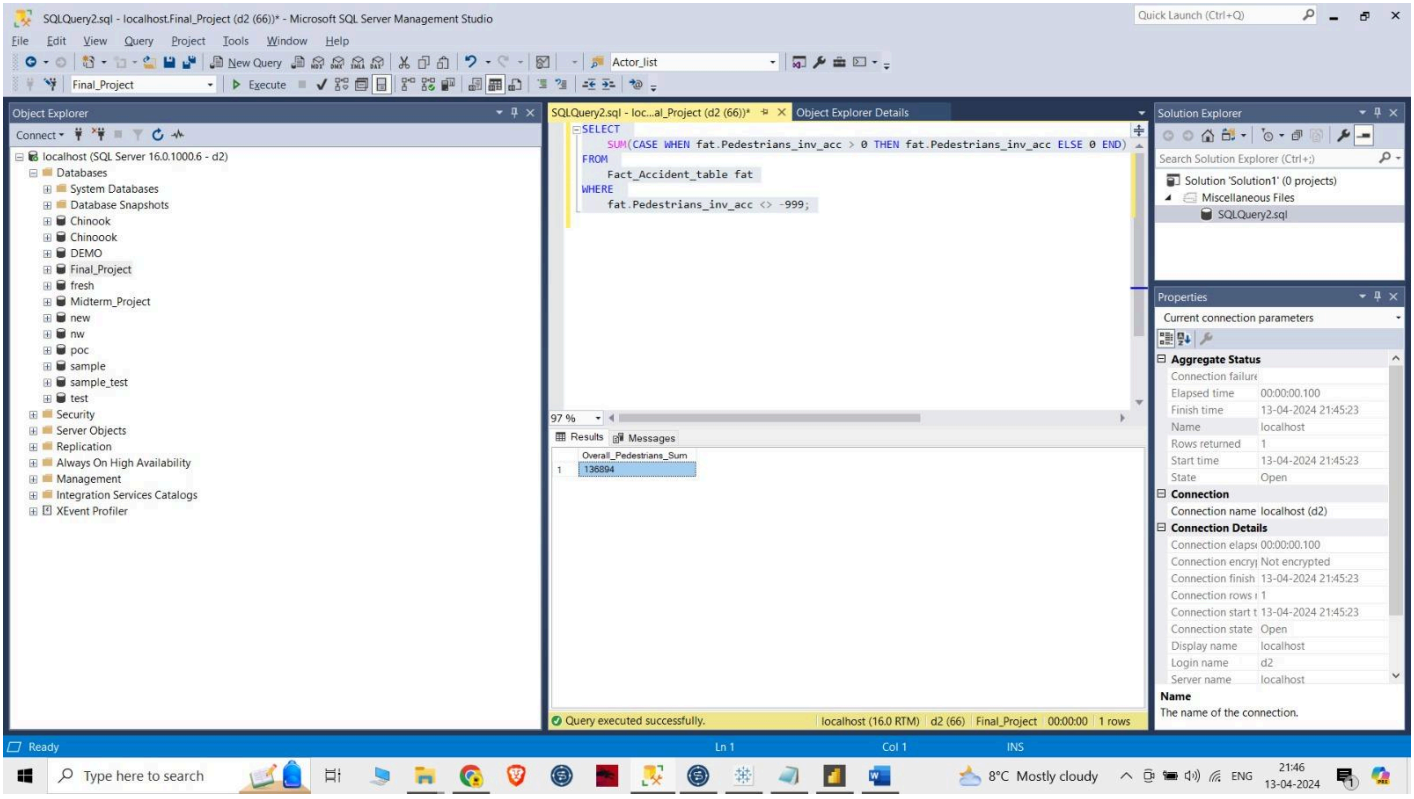
```
-- By city

SELECT
   dl.City,
   YEAR(dd.dt) AS Year,
   MONTH(dd.dt) AS Month,
            SUM(IIF(fat.Pedestrians_inv_acc <> -999, fat.Pedestrians_inv_acc, 0)) AS
City_Pedestrian_Accidents
FROM
   Fact_Accident_table fat
JOIN
   Dim_Location dl ON fat.[Location_id(SK)] = dl.[Location_id(SK)]
JOIN
   Dim_Date dd ON fat.[date(SK)] = dd.[date(SK)]
WHERE
   fat.Pedestrians_inv_acc > 0 AND fat.Pedestrians_inv_acc <> -999
GROUP BY
   dl.City, YEAR(dd.dt), MONTH(dd.dt)
ORDER BY
   YEAR(dd.dt), dl.City;
```



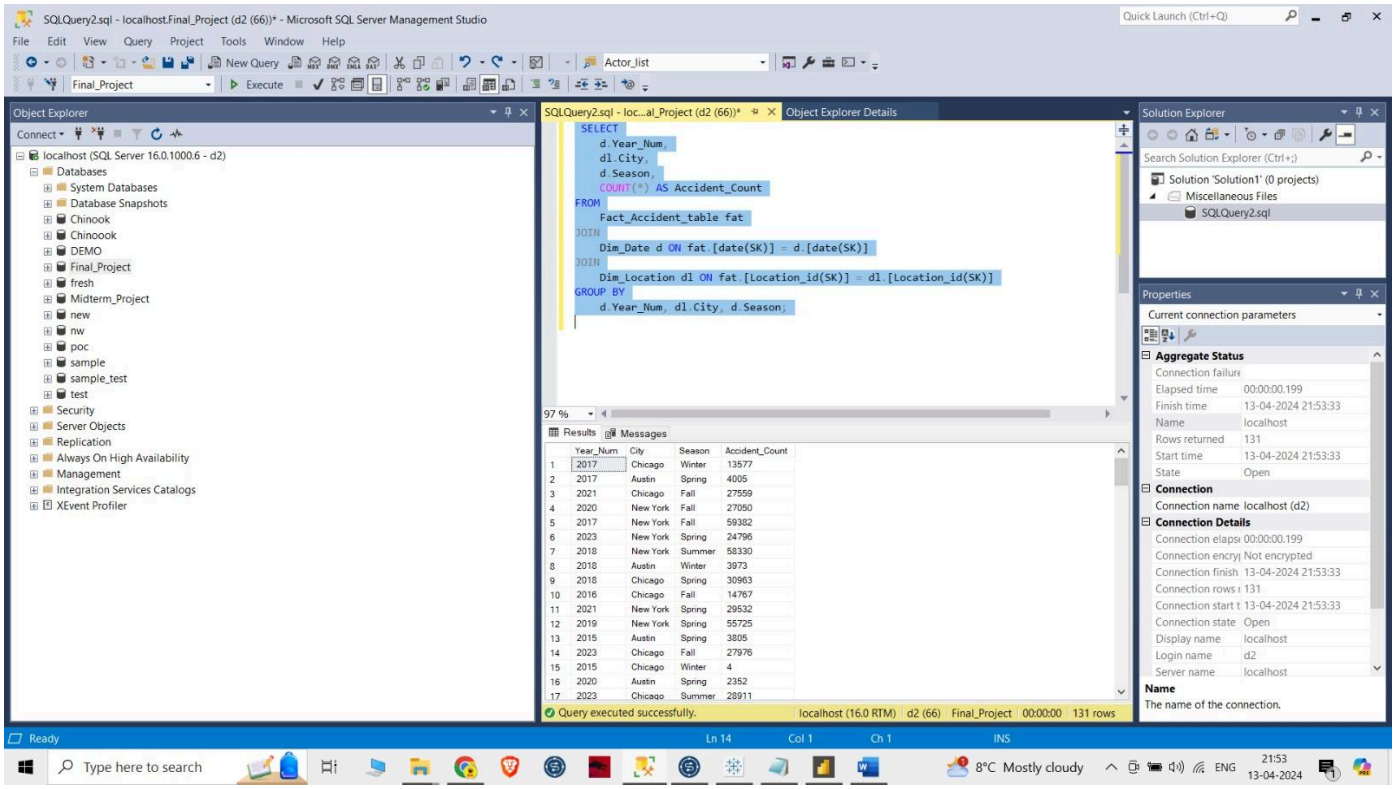
```
-- Overall

SELECT
        SUM(CASE WHEN fat.Pedestrians_inv_acc > 0 THEN fat.Pedestrians_inv_acc ELSE 0
END) AS Overall_Pedestrians_Sum
FROM
   Fact_Accident_table fat
WHERE
   fat.Pedestrians_inv_acc <> -999;
```



5. **When do most accidents happen?**
   seasonality report

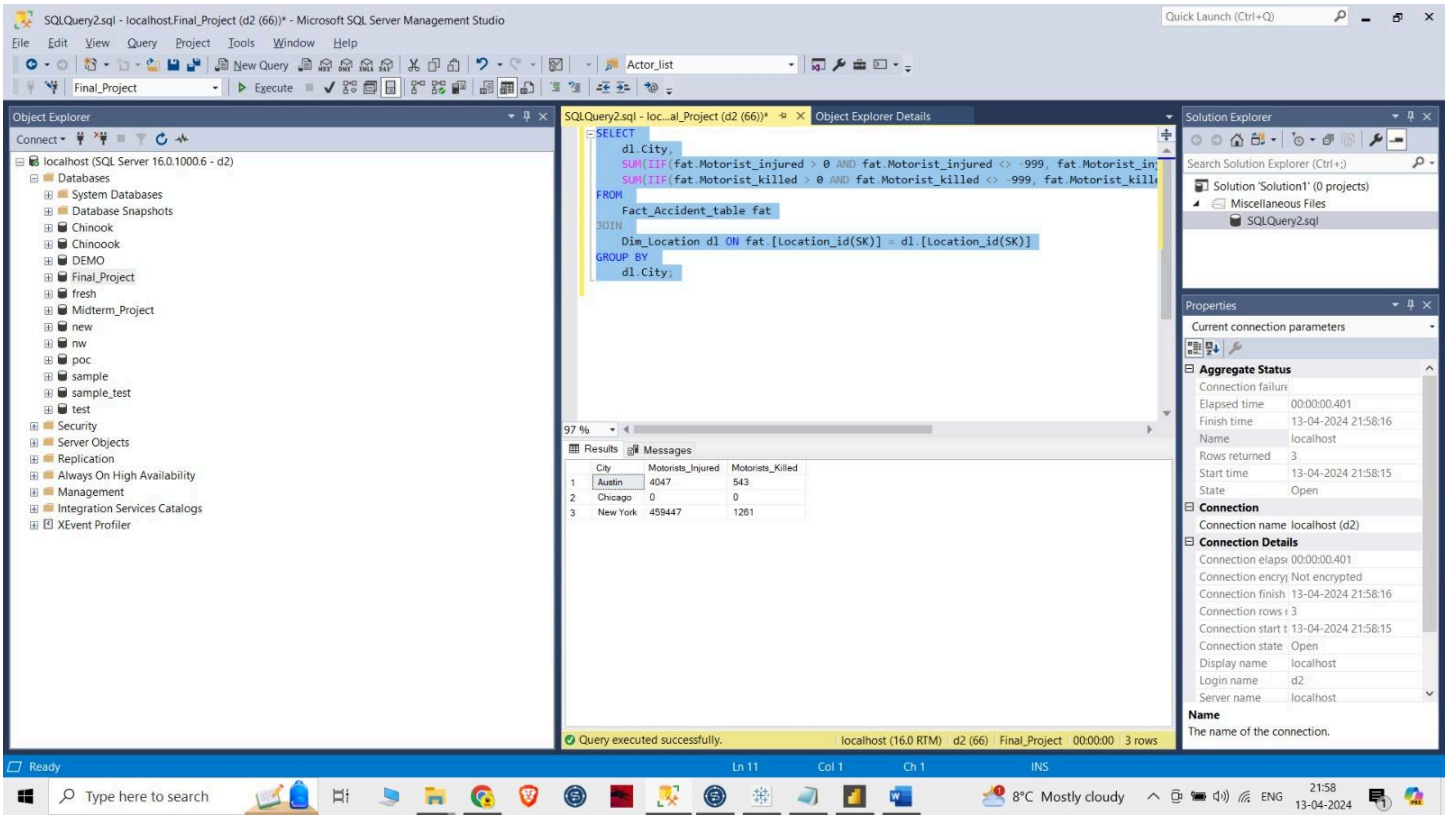
```
SELECT
    d.Year_Num,
    dl.City,
    d.Season,
    COUNT(*) AS Accident_Count
FROM
    Fact_Accident_table fat
JOIN
    Dim_Date d ON fat.[date(SK)] = d.[date(SK)]
JOIN
    Dim_Location dl ON fat.[Location_id(SK)] = dl.[Location_id(SK)]
GROUP BY
    d.Year_Num, dl.City, d.Season;
```



6. **How many motorists are injured or killed in accidents?**
   this report needs to be generated at 2 levels, 1 -> overall, 3 -> by city

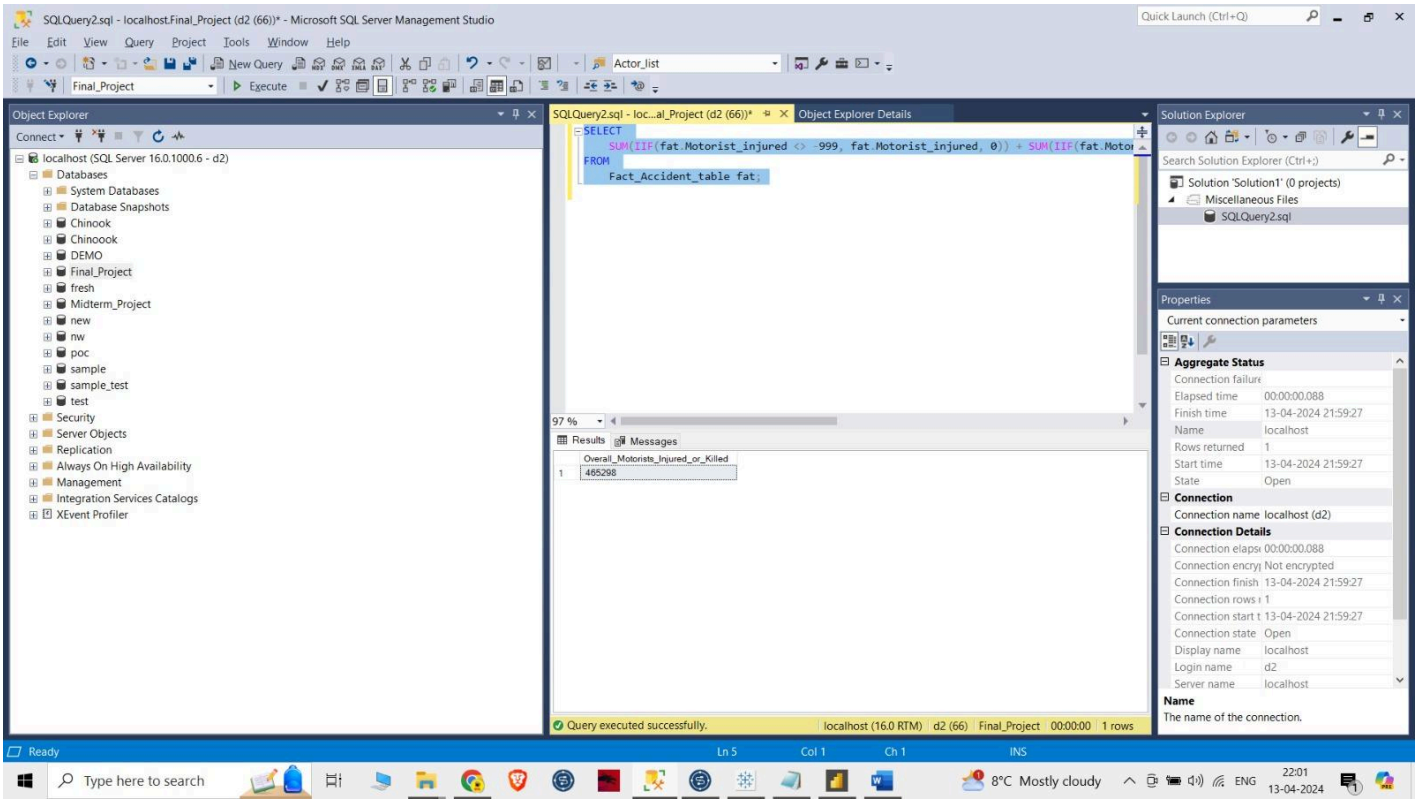
– By city

```
SELECT
    dl.City,
    SUM(IIF(fat.Motorist_injured <> -999, fat.Motorist_injured, 0)) + SUM(IIF(fat.Motorist_killed <> -999,
fat.Motorist_killed, 0)) AS City_Motorists_Injured_or_Killed
FROM
    Fact_Accident_table fat
JOIN
    Dim_Location dl ON fat.[Location_id(SK)] = dl.[Location_id(SK)]
 WHERE
    (fat.Motorist_injured > 0 OR fat.Motorist_killed > 0) AND (fat.Motorist_injured <> -999 OR
fat.Motorist_killed <> -999)
 GROUP BY
dl.City;
```



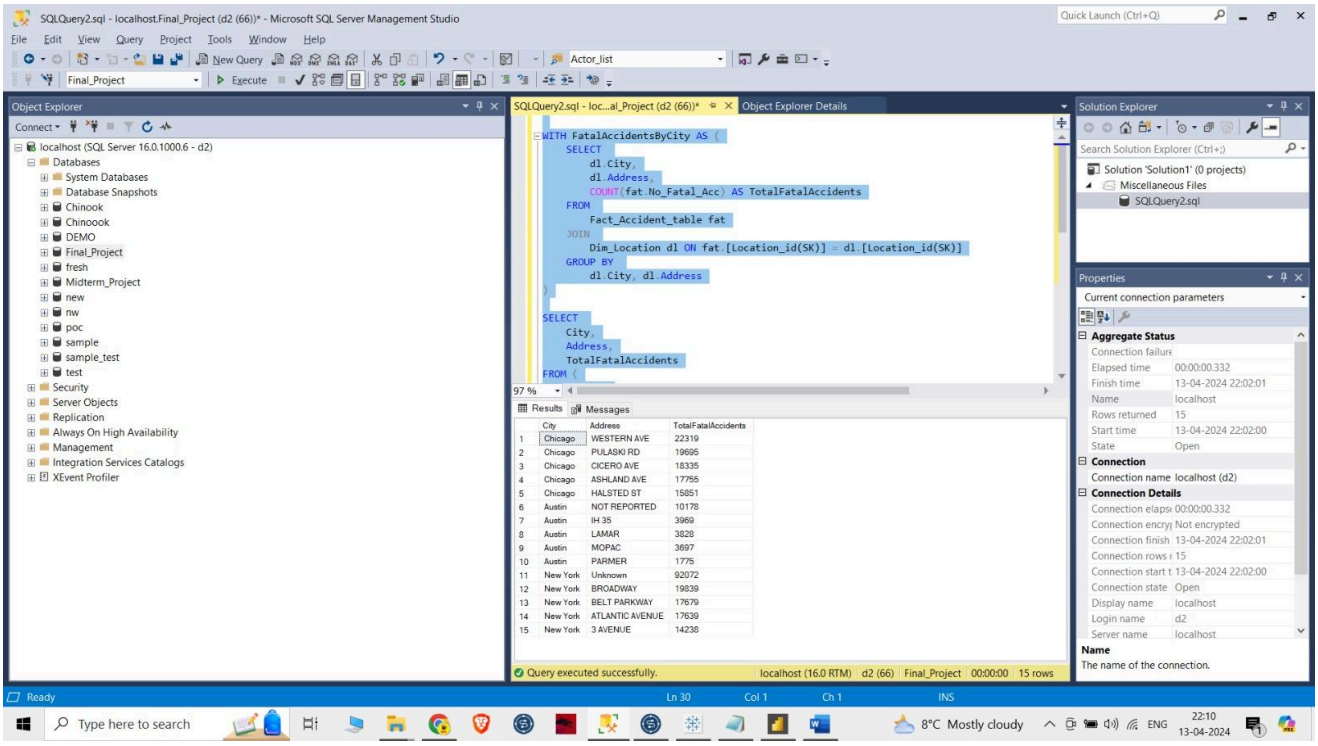
– Overall

```
SELECT
    SUM(IIF(fat.Motorist_injured <> -999, fat.Motorist_injured, 0)) + SUM(IIF(fat.Motorist_killed <> -999, fat.Motorist_killed, 0)) AS Overall_Motorists_Injured_or_Killed
FROM
    Fact_Accident_table fat;
```



7. **Which top 5 areas in 3 cities have the most fatal number of accidents?**

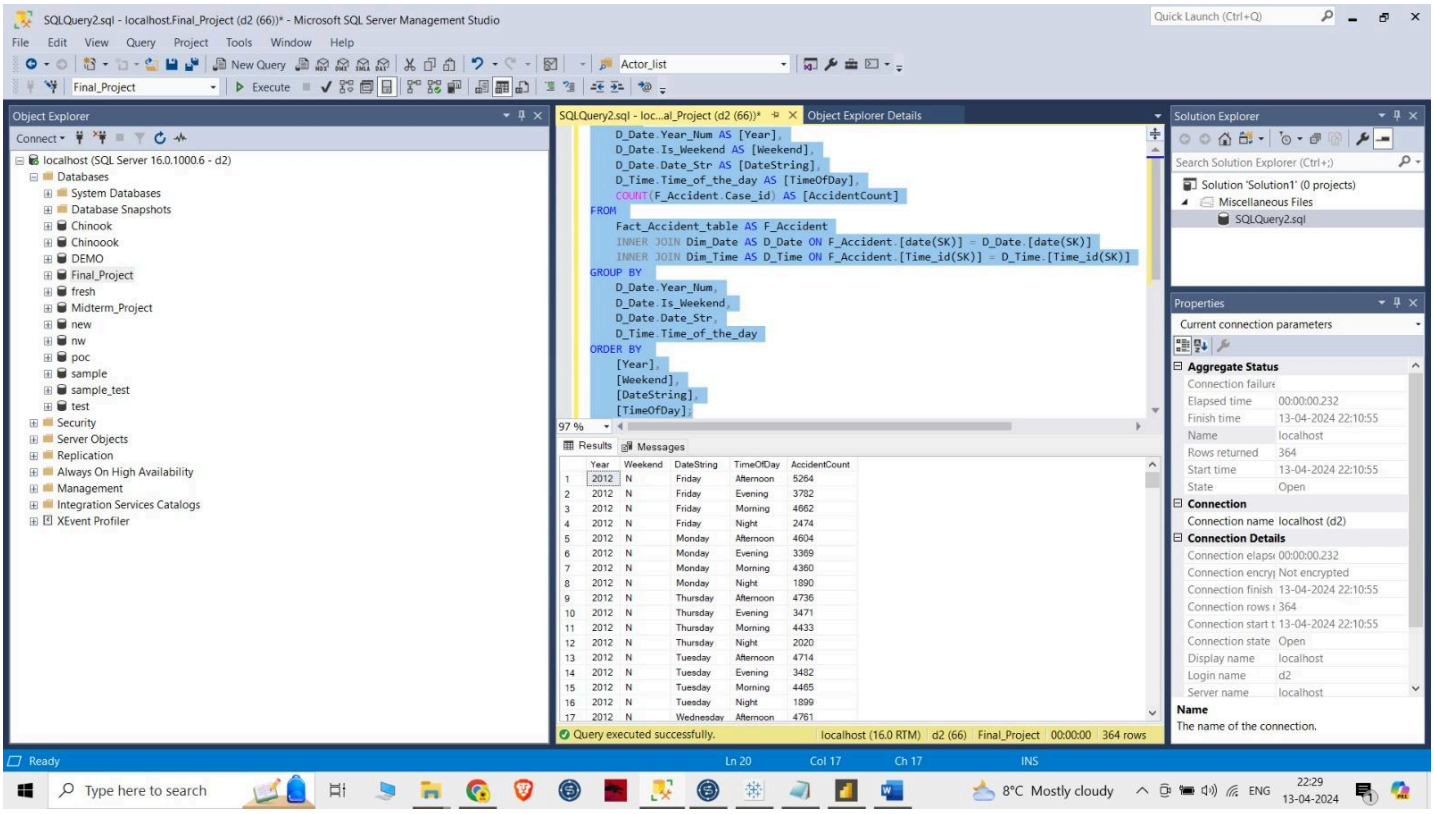
```
WITH FatalAccidentsByCity AS (
    SELECT
        dl.City,
        dl.Address,
        COUNT(fat.No_Fatal_Acc) AS TotalFatalAccidents
    FROM
        Fact_Accident_table fat
    JOIN
        Dim_Location dl ON fat.[Location_id(SK)] = dl.[Location_id(SK)]
    GROUP BY
        dl.City, dl.Address
)
SELECT
    City,
    Address,
    TotalFatalAccidents
FROM (
    SELECT
        City,
        Address,
        TotalFatalAccidents,
        ROW_NUMBER() OVER(PARTITION BY City ORDER BY TotalFatalAccidents DESC) AS
RowNumber
    FROM
        FatalAccidentsByCity
) AS RankedFatalAccidents
WHERE
    RowNumber <= 5;
```



8. Time-based analysis of accidents

Time of the day, day of the week, weekdays or weekends

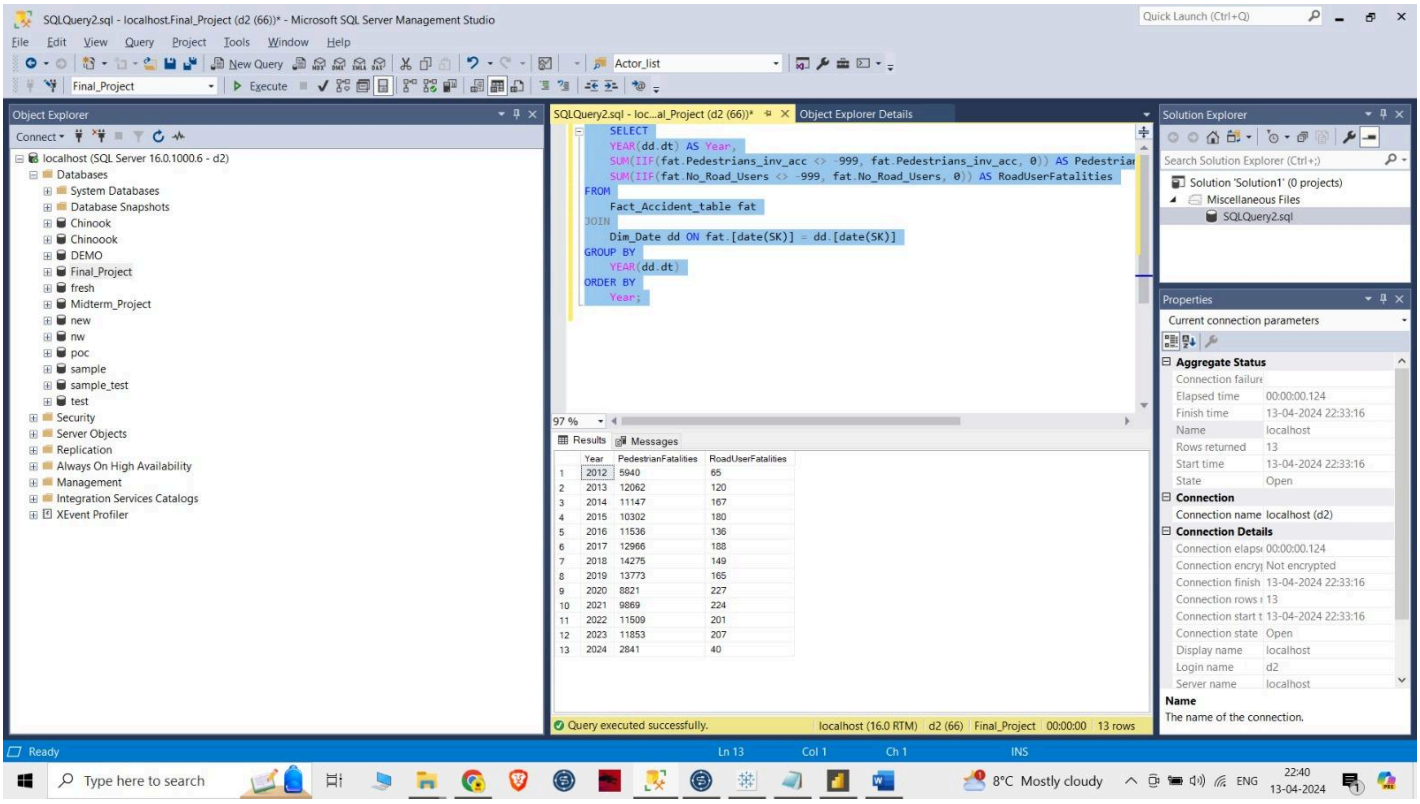
```
SELECT
    D_Date.Year_Num AS [Year],
    D_Date.Is_Weekend AS [Weekend],
    D_Date.Date_Str AS [DateString],
    D_Time.Time_of_the_day AS [TimeOfDay],
    COUNT(F_Accident.Case_id) AS [AccidentCount]
FROM
    Fact_Accident_table AS F_Accident
    INNER JOIN Dim_Date AS D_Date ON F_Accident.[date(SK)] = D_Date.[date(SK)]
    INNER JOIN Dim_Time AS D_Time ON F_Accident.[Time_id(SK)] = D_Time.[Time_id(SK)]
GROUP BY
    D_Date.Year_Num,
    D_Date.Is_Weekend,
    D_Date.Date_Str,
    D_Time.Time_of_the_day
ORDER BY
    [Year],
    [Weekend],
    [DateString],
    [TimeOfDay];
```



9. Fatality analysis

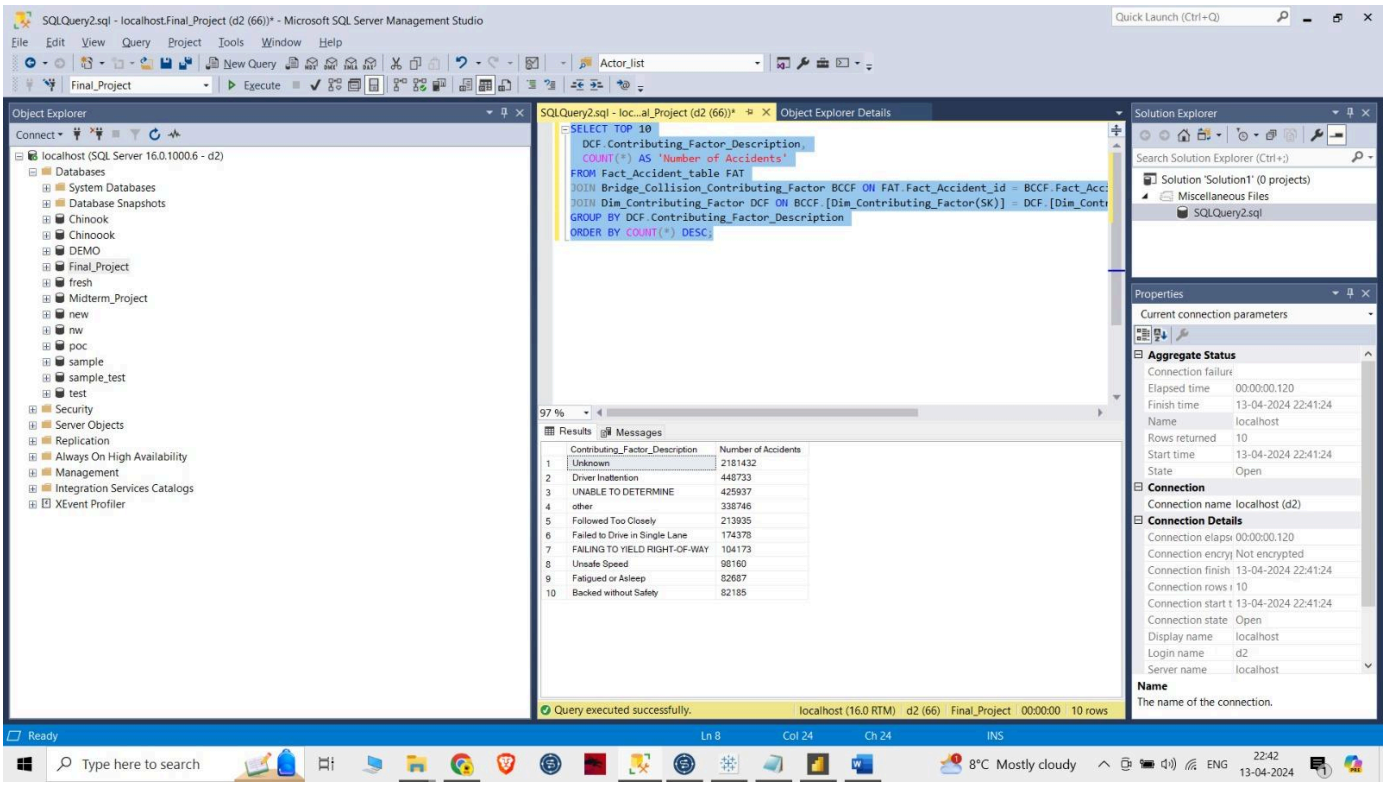
Are pedestrians killed more often than road users?

```
SELECT
    YEAR(dd.dt) AS Year,
    SUM(IIF(fat.Pedestrians_inv_acc <> -999, fat.Pedestrians_inv_acc, 0)) AS PedestrianFatalities,
    SUM(IIF(fat.No_Road_Users <> -999, fat.No_Road_Users, 0)) AS RoadUserFatalities
FROM
    Fact_Accident_table fat
JOIN
    Dim_Date dd ON fat.[date(SK)] = dd.[date(SK)]
GROUP BY
    YEAR(dd.dt)
ORDER BY
    Year;
```



10. What are the most common factors involved in accidents?

```
SELECT TOP 10
  DCF.Contributing_Factor_Description,
  COUNT(*) AS 'Number of Accidents'
FROM Fact_Accident_table FAT
JOIN Bridge_Collision_Contributing_Factor BCCF ON FAT.Fact_Accident_id = BCCF.Fact_Accident_id
JOIN Dim_Contributing_Factor DCF ON BCCF.[Dim_Contributing_Factor(SK)] =
DCF.[Dim_Contributing_Factor(SK)]
GROUP BY DCF.Contributing_Factor_Description
ORDER BY COUNT(*) DESC;
```

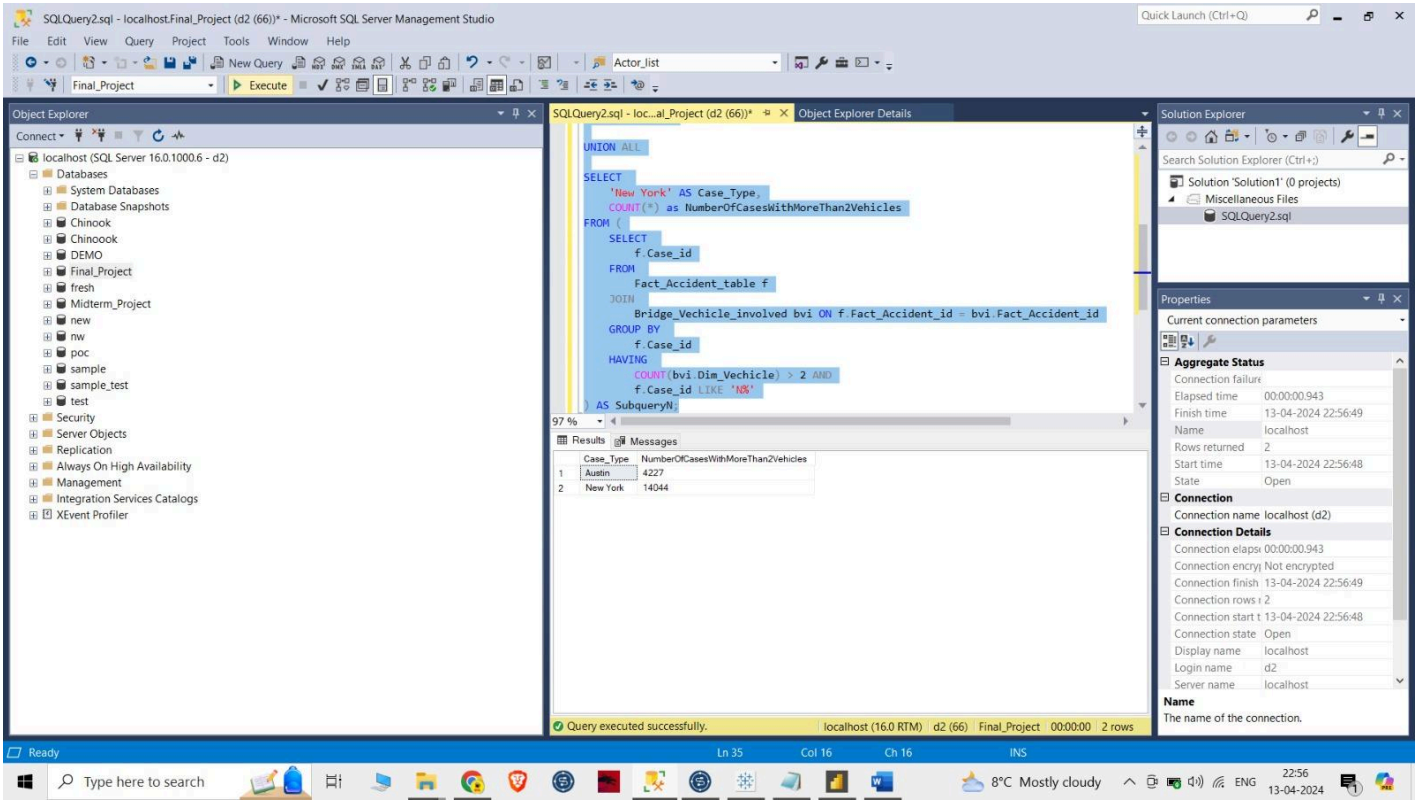


11. **Using Austin and NYC datasets, Create a visualization to show a number of incidents that involved more than 2 vehicles. Show this data as a comparison between these 2 cities.**

```
SELECT
    'Austin' AS Case_Type,
    COUNT(*) as NumberOfCasesWithMoreThan2Vehicles
FROM (
    SELECT
        f.Case_id
    FROM
        Fact_Accident_table f
    JOIN
        Bridge_Vechicle_involved bvi ON f.Fact_Accident_id = bvi.Fact_Accident_id
    GROUP BY
        f.Case_id
    HAVING
        COUNT(bvi.Dim_Vechicle) > 2 AND
        f.Case_id LIKE 'A%'
) AS SubqueryA

UNION ALL

SELECT
    'New York' AS Case_Type,
    COUNT(*) as NumberOfCasesWithMoreThan2Vehicles
FROM (
    SELECT
        f.Case_id
    FROM
        Fact_Accident_table f
    JOIN
        Bridge_Vechicle_involved bvi ON f.Fact_Accident_id = bvi.Fact_Accident_id
    GROUP BY
        f.Case_id
    HAVING
        COUNT(bvi.Dim_Vechicle) > 2 AND
        f.Case_id LIKE 'N%'
) AS SubqueryN;
```


SQLQuery2.sql - localhost.Final_Project (d2 (66))* - Microsoft SQL Server Management Studio
Quick Launch (Ctrl+Q)
File Edit View Query Project Tools Window Help
New Query
Actor_list
Final_Project
Execute
Object Explorer
Connect
localhost (SQL Server 16.0.1000.6 - d2)
Databases
System Databases
Database Snapshots
Chinook
Chinoook
DEMO
Final_Project
fresh
Midterm_Project
new
nw
poc
sample
sample_test
test
Security
Server Objects
Replication
Always On High Availability
Management
Integration Services Catalogs
XEvent Profiler
SQLQuery2.sql - loc...al_Project (d2 (66))*
Object Explorer Details
UNION ALL
SELECT
'New York' AS Case_Type,
COUNT(*) as NumberOfCasesWithMoreThan2Vehicles
FROM (
SELECT
f.Case_id
FROM
Fact_Accident_table f
JOIN
Bridge_Vechicle_involved bvi ON f.Fact_Accident_id = bvi.Fact_Accident_id
GROUP BY
f.Case_id
HAVING
COUNT(bvi.Dim_Vechicle) > 2 AND
f.Case_id LIKE 'N%'
) AS SubqueryN
97 %
Results
Messages
Case_Type
NumberOfCasesWithMoreThan2Vehicles
1 Austin 4227
2 New York 14044
Query executed successfully.
localhost (16.0 RTM)
d2 (66)
Final_Project
00:00:00
2 rows
Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'Solution1' (0 projects)
Miscellaneous Files
SQLQuery2.sql
Properties
Current connection parameters
Aggregate Status
Connection failure
Elapsed time 00:00:00.943
Finish time 13-04-2024 22:56:49
Name localhost
Rows returned 2
Start time 13-04-2024 22:56:48
State Open
Connection
Connection name localhost (d2)
Connection Details
Connection elapse 00:00:00.943
Connection encryp Not encrypted
Connection finish 13-04-2024 22:56:49
Connection rows r 2
Connection start t 13-04-2024 22:56:48
Connection state Open
Display name localhost
Login name d2
Server name localhost
Name
The name of the connection.
Ready
Ln 35
Col 16
Ch 16
INS
Type here to search
8°C Mostly cloudy
ENG
22:56
13-04-2024