CLAIM INVESTIGATION

--- Let's create a copy of the dataset

INSERT INTO claims\_study

SELECT \* FROM [dbo].[claims\_reports]

-- checking for duplicated. We will use the Claim # to check for duplactes because it is likely to be unique indentifier for each claim

-- Some claims have beeng updated over time and we will be working only with the most recent disposition date

WITH duplicates as

(SELECT

COUNT(\*) OVER(partition by CLAIM ORDER BY DISPOSITION\_DATE DESC) as uniq\_check,

\*

FROM [dbo].[claims\_study]

)

DELETE FROM duplicates WHERE uniq\_check >1

--- For CONSISTENCY we need to upadate the table to standardize the different claim types

UPDATE [dbo].[claims\_study]

SET [CLAIM\_TYPE] = UPPER(TRIM(REPLACE([CLAIM\_TYPE], '(PI)' , ' ')))

FROM [dbo].[claims\_study]

--- UPDATE to get the unique description for the claim type

UPDATE [dbo].[claims\_study]

SET [CLAIM\_TYPE] =

CASE

WHEN [CLAIM\_TYPE] = 'NON COVERED AGENCY/CITY ' THEN 'NON COVERED AGENCY '

WHEN [CLAIM\_TYPE] = 'WATER MAIN ' THEN 'WATERMAIN BREAK '

WHEN [CLAIM\_TYPE] = 'DEFECT TRAF/LIGHT/STOP SIGN ' THEN 'DEFECTIVE TRAFF/LIGHT/SIGN '

WHEN [CLAIM\_TYPE] = 'PARKS & RECREATION ' THEN 'RECREATION '

WHEN [CLAIM\_TYPE] = 'CIVIL RIGHTS CLAIMS ' THEN 'CIVIL RIGHTS '

WHEN [CLAIM\_TYPE] = 'DEFECTIVE SIDEWALK' THEN 'SIDEWALK'

WHEN [CLAIM\_TYPE] = 'DEFECTIVE ROADWAY' THEN 'ROADWAY'

WHEN [CLAIM\_TYPE] = 'CITY ONLY(NON COV AGY/BODY)' THEN 'CITY PROPERTY'

WHEN [CLAIM\_TYPE] = 'NON COVERED AGENCY AND CITY' THEN 'NON COVERED AGENCY'

WHEN [CLAIM\_TYPE] = 'EMPLOYEE UNIFORMED SERVICE' THEN 'UNIFORMED SERVICES EMPLOYEE'

WHEN [CLAIM\_TYPE] IN ('PEACE OFFICER (POLICE ACT) ','PEACE OFFICER/POLICE ACTION ') THEN 'POLICE ACTION'

WHEN [CLAIM\_TYPE] IN ('BUILDING AND PROPERTY ','BUILDINGS AND PROPERTY ') THEN 'CITY PROPERTY'

ELSE [CLAIM\_TYPE] END

FROM [dbo].[claims\_study]

-- Total number of claims

SELECT

COUNT(\*)

FROM [dbo].[claims\_study]

A white background with black border

AI-generated content may be incorrect.

--- total claim in regard to claim type

SELECT top 10

CLAIM\_TYPE,

COUNT(\*) AS number\_of\_claim

FROM [dbo].[claims\_study]

GROUP BY [CLAIM\_TYPE]

order by number\_of\_claim desc

A white background with black text

AI-generated content may be incorrect.

--Claim status count

select

[CLAIM\_ACTION],

count([CLAIM]) status\_count

from [dbo].[claims\_study]

group by [CLAIM\_ACTION]

order by status\_count desc

A white rectangular object with a black border

AI-generated content may be incorrect.

-- Claim by borough

SELECT TOP 10

[BOROUGH],

count([CLAIM]) as claim\_per\_borough

FROM [dbo].[claims\_study]

where BOROUGH is not null

GROUP BY [BOROUGH]

ORDER BY claim\_per\_borough desc

A screenshot of a computer

AI-generated content may be incorrect.

---yearly distribution of claims

SELECT

datepart(year,[OCCURRENCE\_DATE]),

count(\*) as total

FROM[dbo].[claims\_study]

WHERE [OCCURRENCE\_DATE] is not null

GROUP BY datepart(year,[OCCURRENCE\_DATE])

ORDER BY total desc;

A screenshot of a computer

AI-generated content may be incorrect.

--what the avereage claim duration by claim type

SELECT

[CLAIM\_TYPE],

AVG(datediff(YEAR,[FILED\_DATE], [DISPOSITION\_DATE])) as year\_duration

from [dbo].[claims\_study]

where [DISPOSITION\_DATE] is not null and [FILED\_DATE] is not null

GROUP BY [CLAIM\_TYPE]

ORDER BY year\_duration DESC

A screenshot of a computer

AI-generated content may be incorrect.

---Year-over-Year Analysis of Monthly Claim Durations

with x as

(SELECT

FISCAL\_YEAR\_FY,

AVG(datediff(month,FILED\_DATE, DISPOSITION\_DATE)) as PreviousMontlyDuration

FROM[dbo].[claims\_study]

WHERE DISPOSITION\_DATE is not null and FILED\_DATE is not null

GROUP BY FISCAL\_YEAR\_FY

), y as

(SELECT

\*,

LAG( PreviousMontlyDuration,1,0) OVER(ORDER BY FISCAL\_YEAR\_FY ) as CurentMonthlyDuration

from x

)

SELECT \*,

PreviousMontlyDuration-CurentMonthlyDuration as DifferenceChange

FROM Y

A screenshot of a computer

AI-generated content may be incorrect.

-- Claim disposition summary

SELECT

COUNT(\*) AS Total\_Claims,

COUNT(Disposition\_Amount) AS Non\_Null\_Disposition\_Count,

AVG(Disposition\_Amount) AS Average\_Disposition\_Amount,

MIN(Disposition\_Amount) AS Min\_Disposition\_Amount,

MAX(Disposition\_Amount) AS Max\_Disposition\_Amount

FROM [dbo].[claims\_study];

A screenshot of a computer

AI-generated content may be incorrect.

---let's investigate those claims with the min and max disposition amount.

SELECT \*

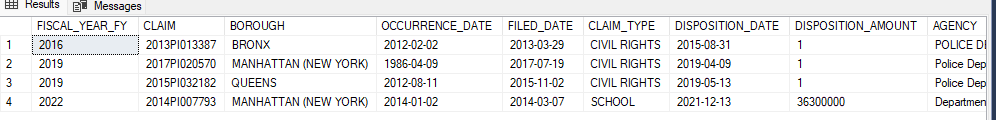
FROM [dbo].[claims\_study]

WHERE

DISPOSITION\_AMOUNT = (select min([DISPOSITION\_AMOUNT]) from [dbo].[claims\_study])

or

DISPOSITION\_AMOUNT = (select max([DISPOSITION\_AMOUNT]) from [dbo].[claims\_study]);



---Top and bottom Claims by disposition amount

WITH claimCost as

(SELECT

[CLAIM],

[DISPOSITION\_AMOUNT],

ROW\_NUMBER() OVER (ORDER BY[DISPOSITION\_AMOUNT] DESC) AS TopAnalysis,

ROW\_NUMBER() OVER(ORDER BY [DISPOSITION\_AMOUNT] ASC) BottomAnalysis

FROM [dbo].[claims\_study]

WHERE [DISPOSITION\_AMOUNT]

IS NOT NULL

)

SELECT

CLAIM,

[DISPOSITION\_AMOUNT],

'TOP 10' AS top10

FROM claimCost

WHERE TopAnalysis <=10

UNION all

SELECT

[CLAIM],

[DISPOSITION\_AMOUNT],

'BOTOM 10'

FROM claimCost

WHERE BottomAnalysis <=10;

A screenshot of a computer

AI-generated content may be incorrect.

-- let rank the BOROUGH based on their disposition amount

SELECT

[BOROUGH],

sum(DISPOSITION\_AMOUNT) as total\_sum,

ROW\_NUMBER() over( ORDER BY sum(DISPOSITION\_AMOUNT) DESC) Ranking

FROM[dbo].[claims\_study]

WHERE BOROUGH is not null AND DISPOSITION\_AMOUNT IS NOT NULL

GROUP BY BOROUGH;

A screenshot of a computer

AI-generated content may be incorrect.