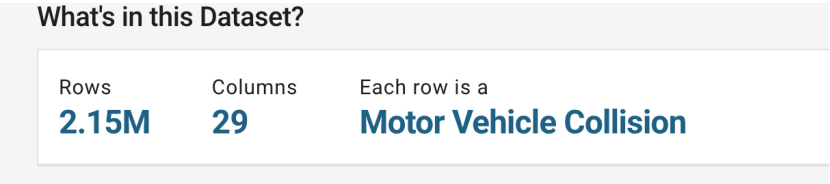
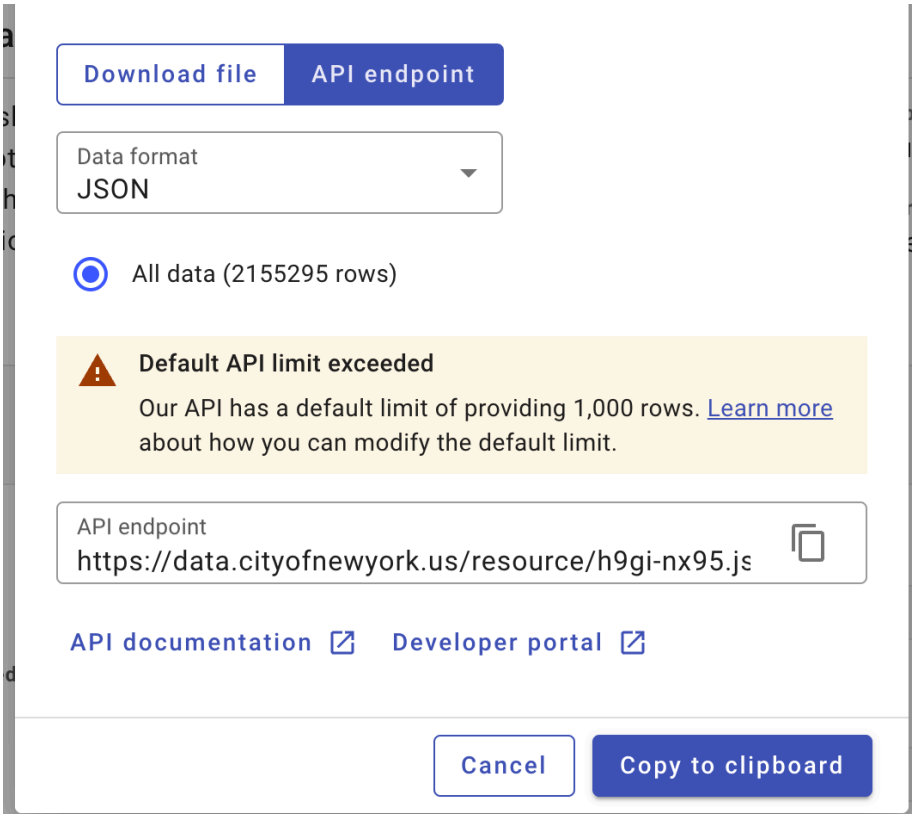
Analyze crash data stored in ADLS Gen2 using a Synapse Analytics Dedicated SQL Pool.

Data: https://data.cityofnewyork.us/Public-Safety/Motor-Vehicle-Collisions-Crashes/h9gi-nx95/about_data



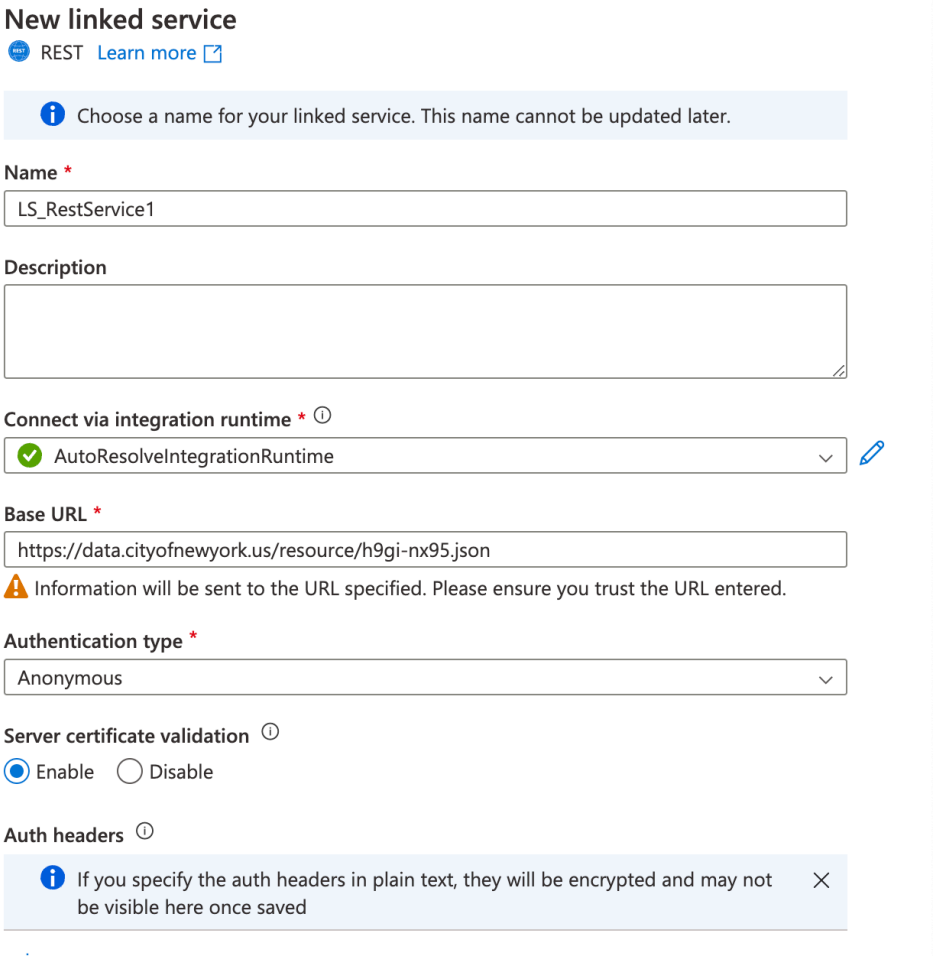
API ENDPOINT



ETL using azure synapse:

1. **Create a pipeline with copy data activity**

2. **For source create a dataset and linked service to the REST api end point of the data.**
   Source REST Linked Service



Set properties

Name

RestResource

Linked service *

LS_RestService

Connect via integration runtime *

AutoResolveIntegrationRuntime

Advanced

SOURCE DATA in JSON format

Preview data

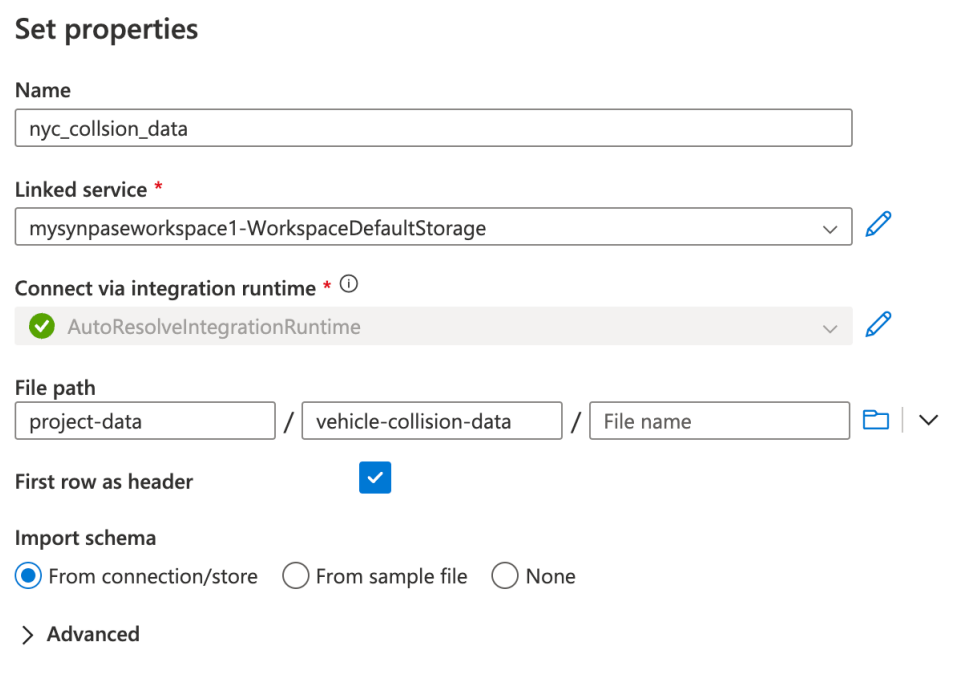
Linked service: LS_RestService

Object:

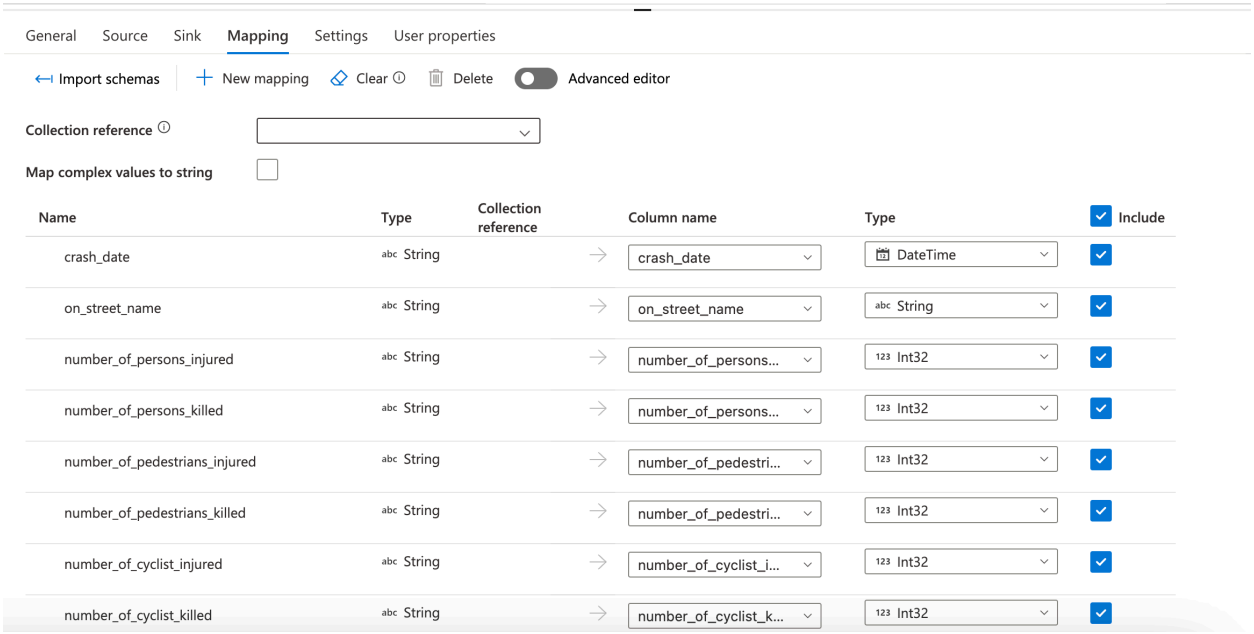

```
[
  {
    "crash_date": "2021-09-11T00:00:00",
    "crash_time": "2:39",
    "on_street_name": "WHITESTONE EXPRESSWAY",
    "off_street_name": "20 AVENUE",
    "number_of_persons_injured": "2",
    "number_of_persons_killed": "0",
    "number_of_pedestrians_injured": "0",
    "number_of_pedestrians_killed": "0",
    "number_of_cyclist_injured": "0",
    "number_of_cyclist_killed": "0",
    "number_of_motorist_injured": "2",
    "number_of_motorist_killed": "0",
    "contributing_factor_vehicle_1": "Aggressive Driving/Road Rage",
    "contributing_factor_vehicle_2": "Unspecified",
    "collision_id": "4455765",
    "vehicle_type_code1": "Sedan",
    "vehicle_type_code2": "Sedan"
  },
```

3. **Created a sink dataset and linked service to Azure data lake gen2 Load data into the ADLS container**

Dataset for sink container



Mapping the source data to sink.



Run pipeline debug to lookup the data.

Pipeline run successfully.

Details Refresh

Learn more on copy performance details from here.

Activity run id: 72868b91-61f2-4a69-8346-6b6a1e4908e8

REST → Succeeded → Azure Data Lake Storage Gen2
Region: Canada Central

Data read:	1.236 MB	**Data written:**	121.395 KB
Objects read:	1,000	**Files written:**	1
Peak connections:	1	**Rows written:**	1,000
		Peak connections:	1

Copy duration	00:00:10
Throughput:	618.208 KB/s

REST → Azure Data Lake Storage Gen2

Start time	2/13/2025, 3:54:50 PM
Used DIUs	4
Used parallel copies	1
Duration	00:00:10

Details	Working duration	Total duration
Queue		00:00:05
Time to first byte	00:00:01	

How satisfied or dissatisfied are you with the performance of this copy activity?

Converted JSON Data to CSV data and Load into ADLS container.

Upload | Add Directory | Refresh | Rename | Delete | Change tier | Acquire leas

Authentication method: Access key (Switch to Microsoft Entra user account)
Location: project-data / vehicle-collision-data

Search blobs by prefix (case-sensitive)

Name	Modified	Access tier	Archive status
data_72868b91-61f2-4a69-8346-6b6a1...	2/13/2025, 3:55:01 PM	Hot (Inferred)	

Upload Add Directory ···

Authentication method: Access key (Switch to Microsoft Entra user account)
Location: project-data / vehicle-collision-data

Search blobs by prefix (case-...

Show deleted objects

Name

- data_3bad1642-4c66-434e-8a...
- data_72868b91-61f2-4a69-83...

vehicle-collision-data/data_72868b91-61f2-4a69-8346-6b6a1e4908e...

Blob

Save Discard Download Refresh Delete

Overview Versions Edit Generate SAS

```
crash_date,on_street_name,number_of_persons_injured,number_of_persons_killed,number_of_pedestrians_injur
2021-09-11 00:00:00.0000000,"WHITESTONE EXPRESSWAY",2,0,0,0,0,2,0,"Aggressive Driving/Road Rage","Unsp
2022-03-26 00:00:00.0000000,"QUEENSBORO BRIDGE UPPER",1,0,0,0,0,1,0,"Pavement Slippery",,4513547,"Seda
2023-11-01 00:00:00.0000000,"OCEAN PARKWAY",1,0,0,0,0,1,0,"Unspecified","Unspecified",4675373,"Moped"
2022-06-29 00:00:00.0000000,"THROGS NECK BRIDGE",0,0,0,0,0,0,0,"Following Too Closely","Unspecified",4
2022-09-21 00:00:00.0000000,"BROOKLYN BRIDGE",0,0,0,0,0,0,0,"Passing Too Closely","Unspecified",45661
2023-04-26 00:00:00.0000000,"WEST 54 STREET",0,0,0,0,0,0,0,"Unspecified","Unspecified",4623759,"Sedan"
2023-11-01 00:00:00.0000000,"HUTCHINSON RIVER PARKWAY",0,0,0,0,0,0,0,"Following Too Closely","Driver I
2023-11-01 00:00:00.0000000,"WEST 35 STREET",0,0,0,0,0,0,0,"Failure to Yield Right-of-Way",,4675769,"S
2023-04-26 00:00:00.0000000,,0,0,0,0,0,0,0,"Unspecified",,4623865,"Sedan"
2021-09-11 00:00:00.0000000,,0,0,0,0,0,0,0,"Unspecified",,4456314,"Sedan"
2021-12-14 00:00:00.0000000,"SARATOGA AVENUE",0,0,0,0,0,0,0,,,4486609,
2021-04-14 00:00:00.0000000,"MAJOR DEEGAN EXPRESSWAY RAMP",0,0,0,0,0,0,0,"Unspecified","Unspecified",4
2021-12-14 00:00:00.0000000,"BROOKLYN QUEENS EXPRESSWAY",0,0,0,0,0,0,0,"Passing Too Closely","Unspecif
2021-12-14 00:00:00.0000000,,2,0,0,0,0,0,2,0,"Unspecified","Unspecified",4486660,"Sedan"
2021-12-14 00:00:00.0000000,,0,0,0,0,0,0,0,0,"Driver Inexperience","Unspecified",4487074,"Sedan"
2021-12-14 00:00:00.0000000,"3 AVENUE",0,0,0,0,0,0,0,"Passing Too Closely","Unspecified",4486519,"Seda
2021-12-13 00:00:00.0000000,"MYRTLE AVENUE",0,0,0,0,0,0,0,"Passing or Lane Usage Improper","Unspecifie
```

Dedicated pool

1. Create a Dedicated SQL Pool

New dedicated SQL pool

Validation succeeded.

Basics * | Additional settings * | Tags | **Review + create**

Product details

Azure Synapse Analytics dedicated SQL pool by Microsoft
Terms of use | Privacy policy

Est. cost per hour
1.33 USD
View pricing details

Terms

By clicking "Create", I (a) agree to the legal terms and privacy statement(s) associated with the Marketplace offering(s) listed above; (b) authorize Microsoft to bill my current payment method for the fees associated with the offering(s), with the same billing frequency as my Azure subscription; and (c) agree that Microsoft may share my contact, usage and transactional information with the provider(s) of the offering(s) for support, billing and other transactional activities. Microsoft does not provide rights for third-party offerings. For additional details see Azure Marketplace Terms

Data source

Dedicated SQL pool name	dedicatedSqlPool1
Performance level	DW100c

Additional settings

Use existing data	Blank
Collation	SQL_Latin1_General_CP1_CI_AS

Pool name	Type	Version	Status	Size	CPU utilizat...	Memory uti...	Created on
Built-in	Serverless	v2	Online	Auto	N/A	N/A	N/A
dedicatedSqlPool1	Dedicated	v2	Online	DW100c	N/A	N/A	2/13/2025, 4:04:54 PM

Bulk Load the data into the collision data table.

Bulk load

Select target SQL pool
Specify the target location for your load including the SQL pool, table, and column mapping. Learn more

Select SQL pool*
dedicatedSqlPool1

Select a database*
dedicatedSqlPool1

Target table
Existing table Create new

New target table
dbo.trafficCollisonData Clustered columnstore index

Configure column mapping

Load data
Automatically Using SQL script
This will generate a SQL script and you will be required to run the SQL script.

1. Create a table to bulk load the data

```sql
IF NOT EXISTS (SELECT * FROM sys.objects O JOIN sys.schemas S ON O.schema_id = S.schema_id WHERE O.NAME = 'collisionData' AND O.TYPE = 'U'
        AND S.NAME = 'dbo')
CREATE TABLE dbo.collisionData
    (
        crash_date DATE,
        on_street_name NVARCHAR(255),
        number_of_persons_injured INT,
        number_of_persons_killed INT,
        number_of_pedestrians_injured INT,
        number_of_pedestrians_killed INT,
        number_of_cyclist_injured INT,
        number_of_cyclist_killed INT,
        number_of_motorist_injured INT,
        number_of_motorist_killed INT,
        contributing_factor_vehicle_1 NVARCHAR(255),
        contributing_factor_vehicle_2 NVARCHAR(255),
        collision_id BIGINT,
        vehicle_type_code1 NVARCHAR(255)
    )
WITH
    (
    DISTRIBUTION = ROUND_ROBIN,
     CLUSTERED COLUMNSTORE INDEX
     -- HEAP
    )
GO
```

2. Load data using COOPY INTO statement

```sql
COPY INTO dbo.collisionData
(crash_date 1, on_street_name 2, number_of_persons_injured 3, number_of_persons_killed 4, number_of_pedestrians_injured 5, number_of_pedestrians
FROM 'https://synapsedatasetadls.dfs.core.windows.net/project-data/vehicle-collision-data/nyc_collision_data.csv'
WITH
(
    FILE_TYPE = 'CSV'
    ,MAXERRORS = 0
    ,FIRSTROW = 02
);
```

Results Messages

View Table Chart Export results

Search

crash_date	on_street_name	number_of_pe...	number_of_pe...	number_of_pe...	number_of_pe...	number_of_cyc...	number_of_cyc...	number_of_m...
2022-03-26	QUEENSBORO ...	1	0	0	0	0	0	1
2023-11-01	HUTCHINSON ...	0	0	0	0	0	0	0
2021-09-11	WHITESTONE E...	2	0	0	0	0	0	2
2023-04-26	WEST 54 STREET	0	0	0	0	0	0	0
2023-11-01	OCEAN PARKW...	1	0	0	0	0	0	1
2022-06-29	THROGS NECK ...	0	0	0	0	0	0	0
2022-09-21	BROOKLYN BRI...	0	0	0	0	0	0	0

3. Create a view for yearly collision summary

```sql
-- Create a view for yearly collision summary
CREATE VIEW YearlyCollisionSummary AS
SELECT
    YEAR(crash_date) AS crash_year,
    COUNT(*) AS total_collisions,
    SUM(number_of_persons_injured) AS total_injuries,
    SUM(number_of_persons_killed) AS total_deaths
FROM dbo.collisionData
GROUP BY YEAR(crash_date);
```

```
68
69  select * from YearlyCollisionSummary;
70
```

Results Messages

View Table Chart Export results

Search

crash_year	total_collisions	total_injuries	total_deaths
2022	141	70	0
2021	846	376	4
2023	9	2	0
2016	1	0	0
2020	2	0	0
2019	1	0	0

Analysis on the data :

Top 5 streets with the most collisions

```sql
-- Query 1: Top 5 streets with the most collisions
SELECT TOP 5
    on_street_name,
    COUNT(*) AS collision_count
FROM dbo.collisionData
GROUP BY on_street_name
ORDER BY collision_count DESC;
```

Results Messages

View Table Chart Export results

Search

on_street_name	collision_count
(NULL)	258
BELT PARKWAY	16
FDR DRIVE	14
MAJOR DEEGA...	10
BRONX RIVER ...	9

Total injuries and deaths per vehicle type

```sql
-- Query 2: Total injuries and deaths per vehicle type
SELECT
    vehicle_type_code1,
    SUM(number_of_persons_injured) AS total_injuries,
    SUM(number_of_persons_killed) AS total_deaths
FROM dbo.collisionData
GROUP BY vehicle_type_code1
ORDER BY total_injuries DESC;
```

Results Messages

View Table Chart Export results

Search

vehicle_type_code1	total_injuries	total_deaths
Sedan	206	1
Station Wagon/Sport Utility Vehi...	158	1
Taxi	29	1
(NULL)	10	0
Bike	9	0
Motorcycle	7	0
E-Bike	6	0

Monthly collision trend

Run Undo Publish Query plan Connect to dedicatedSqlPool1 Use database dedica

```sql
-- Query 3: Monthly collision trend
SELECT
    YEAR(crash_date) AS crash_year,
    MONTH(crash_date) AS crash_month,
    COUNT(*) AS total_collisions
FROM dbo.collisionData
GROUP BY YEAR(crash_date), MONTH(crash_date)
ORDER BY crash_year, crash_month;
```

Results Messages

View Table Chart Export results

Search

crash_year	crash_month	total_collisions
2016	4	1
2019	5	1
2020	1	1
2020	4	1
2021	2	1
2021	3	5
2021	4	517

Contribution factors causing the most injuries

```sql
-- Query 4: Contribution factors causing the most injuries
SELECT TOP 5
    contributing_factor_vehicle_1,
    SUM(number_of_persons_injured) AS total_injuries
FROM dbo.collisionData
GROUP BY contributing_factor_vehicle_1
ORDER BY total_injuries DESC;
```

Results Messages

View Table Chart Export results

Search

contributing_factor_vehicle_1	total_injuries
Driver Inattention/Distraction	95
Unspecified	85
Failure to Yield Right-of-Way	57
Following Too Closely	41
Traffic Control Disregarded	25

Collisions involving pedestrians

Run Undo Publish Query plan Connect to dedicatedSqlPool1 Use database dedicatedSqlPool1

```sql
-- Query 5: Collisions involving pedestrians
SELECT
    crash_date,
    on_street_name,
    number_of_pedestrians_injured,
    number_of_pedestrians_killed
FROM dbo.collisionData
WHERE number_of_pedestrians_injured > 0 OR number_of_pedestrians_killed > 0
ORDER BY crash_date DESC;
```

Results Messages

View Table Chart Export results

Search

crash_date	on_street_name	number_of_pedestrians_injured	number_of_pedestrians_killed
2022-04-22	EAST 107 STREET	1	0
2022-04-12	UTICA AVENUE	1	0
2022-03-26	EAST 168 STREET	1	0
2022-03-26	(NULL)	1	0
2022-03-26	WEST 30 STREET	1	0
2022-03-26	EAST 94 STREET	2	0

Yearly collisions by severity (injuries and deaths)

```sql
-- Query 6: Yearly collisions by severity (injuries and deaths)
SELECT
    YEAR(crash_date) AS crash_year,
    SUM(CASE WHEN number_of_persons_injured > 0 THEN 1 ELSE 0 END) AS collisions_with_injuries,
    SUM(CASE WHEN number_of_persons_killed > 0 THEN 1 ELSE 0 END) AS collisions_with_deaths
FROM dbo.collisionData
GROUP BY YEAR(crash_date)
ORDER BY crash_year;
```

Results Messages

View Table Chart Export results

Search

crash_year	collisions_with_injuries	collisions_with_deaths
2016	0	0
2019	0	0
2020	0	0
2021	279	4
2022	58	0
2023	2	0