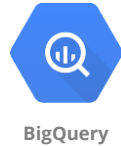


# NYC MV Collisions

NYC OpenData



## Assignment 2

- Load Data from Staging to Integration Schema (dimensional model) with Talend
- Use SQL Server, Azure SQL, MySQL or Azure SQL



# Assignment 2: Tasks

- Load data from STG tables to dimensional model & perform data cleansing using Talend
  - Same database as STG tables reside
- Document any data cleansing tasks and results and explain
- Query dimensional data model with the listed business questions as a minimum

# Assignment 2

- Upload
  - Screen shots of completed loads
  - Time each load took (should be on screen shots above)
    - Overall Orchestrator or Master job
  - Completed Talend jobs
  - List of tables with rows counts
  - Results of each query

# NYC MV Collisions

## Data Model

# Staging Tables: Crashes, Vehicles, Persons

stg\_nyc\_mv\_collision\_persons

UNIQUE_ID	BIGINT
COLLISION_ID (FK)	BIGINT
CRASH_DATE	DATETIME
CRASH_TIME	TIME/DATETIME
PERSON_ID	VARCHAR(80)
PERSON_TYPE	VARCHAR(80)
PERSON_INJURY	VARCHAR(80)
VEHICLE_ID	VARCHAR(80)
PERSON_AGE	INTEGER
EJECTION	VARCHAR(80)
EMOTIONAL_STATUS	VARCHAR(80)
BODILY_INJURY	VARCHAR(80)
POSITION_IN_VEHICLE	VARCHAR(255)
SAFETY_EQUIPMENT	VARCHAR(255)
PED_LOCATION	VARCHAR(255)
PED_ACTION	VARCHAR(255)
COMPLAINT	VARCHAR(255)
PED_ROLE	VARCHAR(255)
CONTRIBUTING_FACTOR_1	VARCHAR(255)
CONTRIBUTING_FACTOR_2	VARCHAR(255)
PERSON_SEX	VARCHAR(10)
DI_PID	VARCHAR(20)
DI_Create_Date	DATETIME

stg\_nyc\_mv\_collisions\_BigQuery

COLLISION_ID	BIGINT
collision_dt	DATETIME
collision_day	DATE
collision_time	TIME/DATETIME
collision_hour	INTEGER
collision_dayoftheweek	INTEGER
borough	VARCHAR(40)
zip_code	VARCHAR(40)
off_street_name	VARCHAR(40)
on_street_name	VARCHAR(40)
cross_street_name	VARCHAR(40)
latitude	NUMERIC(24,6)
longitude	NUMERIC(24,6)
location	VARCHAR(256)
contributing_factor_vehicle_1	VARCHAR(256)
contributing_factor_vehicle_2	VARCHAR(256)
contributing_factor_vehicle_3	VARCHAR(256)
contributing_factor_vehicle_4	VARCHAR(256)
contributing_factor_vehicle_5	VARCHAR(256)
number_of_cyclist_injured	INTEGER
number_of_cyclist_killed	INTEGER
number_of_motorist_injured	INTEGER
number_of_motorist_killed	INTEGER
number_of_pedestrians_injured	INTEGER
number_of_pedestrians_killed	INTEGER
number_of_persons_injured	INTEGER
number_of_persons_killed	INTEGER
vehicle_type_code1	VARCHAR(80)
vehicle_type_code2	VARCHAR(80)
vehicle_type_code_3	VARCHAR(80)
vehicle_type_code_4	VARCHAR(80)
vehicle_type_code_5	VARCHAR(80)
DI_JobID	VARCHAR(20)
DI_CreateDate	DATETIME

stg\_nyc\_mv\_collision\_vehicles

UNIQUE_ID	BIGINT
COLLISION_ID (FK)	BIGINT
CRASH_DATE	DATETIME
CRASH_TIME	TIME/DATETIME
VEHICLE_ID	VARCHAR(80)
STATE_REGISTRATION	VARCHAR(80)
VEHICLE_TYPE	VARCHAR(80)
VEHICLE_MAKE	VARCHAR(80)
VEHICLE_MODEL	VARCHAR(80)
VEHICLE_YEAR	VARCHAR(80)
TRAVEL_DIRECTION	VARCHAR(255)
VEHICLE_OCCUPANTS	INTEGER
DRIVER_SEX	VARCHAR(80)
DRIVER_LICENSE_STATUS	VARCHAR(255)
DRIVER_LICENSE_JURISDICTION	VARCHAR(255)
PRE_CRASH	VARCHAR(255)
POINT_OF_IMPACT	VARCHAR(255)
VEHICLE_DAMAGE	VARCHAR(255)
VEHICLE_DAMAGE_1	VARCHAR(255)
VEHICLE_DAMAGE_2	VARCHAR(255)
VEHICLE_DAMAGE_3	VARCHAR(255)
PUBLIC_PROPERTY_DAMAGE	VARCHAR(1024)
PUBLIC_PROPERTY_DAMAGE_TYPE	VARCHAR(1024)
CONTRIBUTING_FACTOR_1	VARCHAR(255)
CONTRIBUTING_FACTOR_2	VARCHAR(255)
DI_PID	VARCHAR(20)
DI_Create_Date	DATETIME

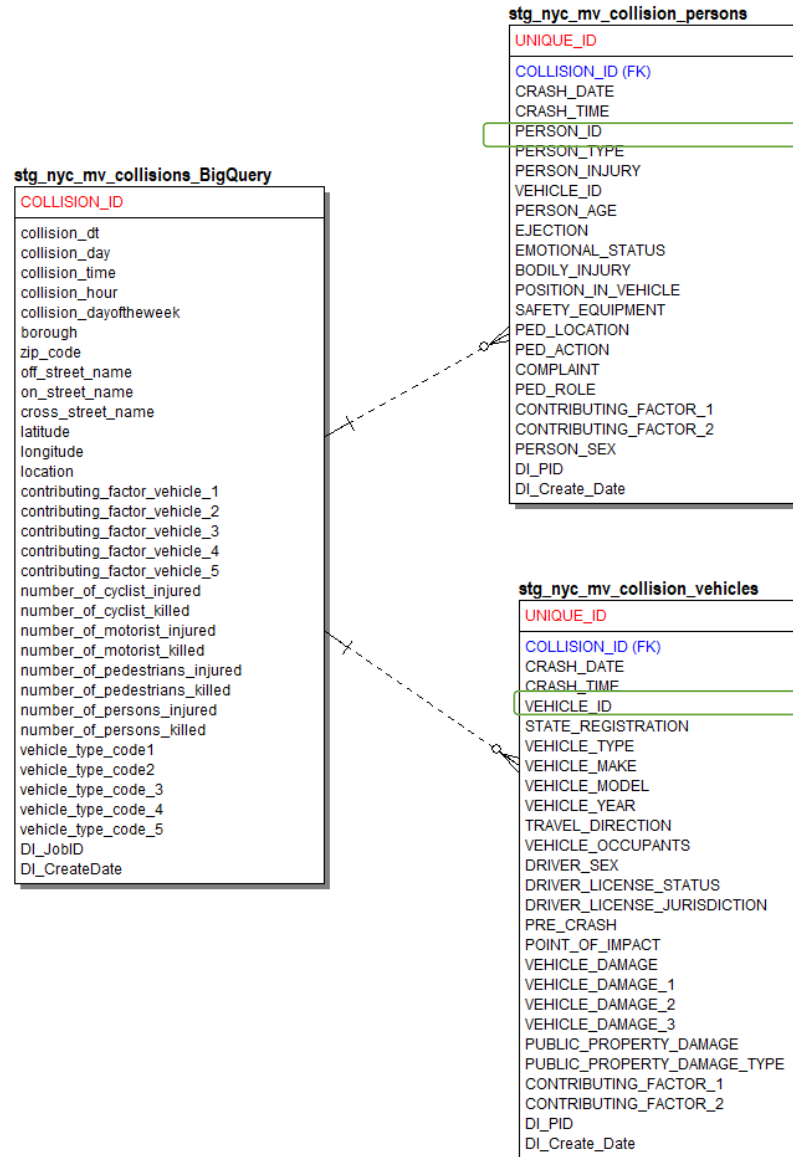
# Dimensional Model: Examining Stage Tables

Primark Keys (PKs):

- Collision\_ID & two Unique\_IDs
- The Unique\_IDs are surrogate keys & are NOT related to each other

Crash\_date & Crash\_Time (or collision\_day & collision\_time) repeated in each table

- Not necessary in tables when tables are used together

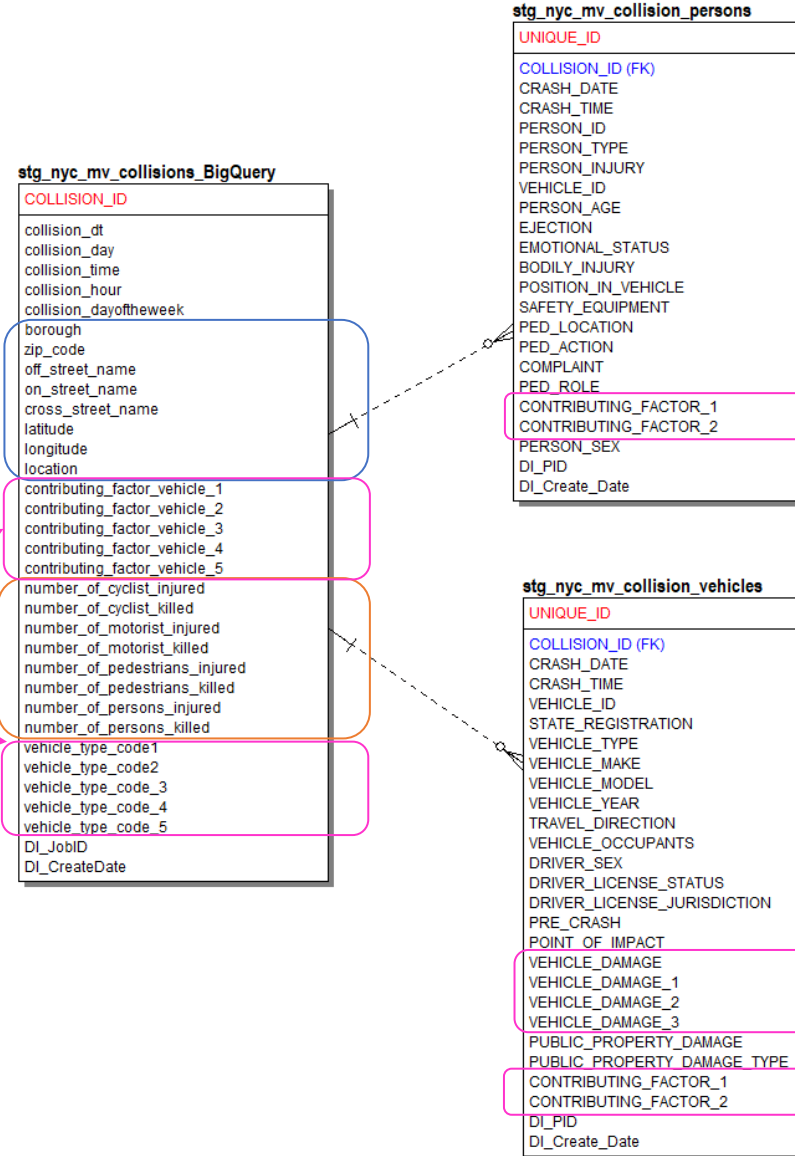


- 1-n number of rows per collision
- Person\_ID
  - how it is populated changes over time (based on data profiling NOT documentation)
  - NOT a unique identifier
- 1-n number of rows per collision
- Vehicle\_ID
  - how it is populated changes over time (based on data profiling NOT documentation)
  - NOT a unique identifier



# Dimensional Model: Examining Stage Tables

- Collision Location
- Summary statistics
- Repeating Groups
  - Assumes 5 or less vehicles but there are sometimes more than 5
  - \_n is associated with the n number vehicle
  - This data is redundant with what is in Vehicles tables but should we ignore?



## Repeating Groups

- Used to provide multiple values in a single row
- Should we normalize????

# Dimensional Data Model (INT Schema)

- nyc mv collisions dimensional model.sql



# Dimensional Model

## Surrogate Keys (SKs) & Foreign Keys (FK) Data Quality

# Dimensions: NULL

Each dimension should have a “No Value Provided” or appropriate phrase with -99 (or other negative number) as the value for the surrogate key (SK)

```
INSERT INTO Dim_COMPLAINT  
(COMPLAINT_SK, COMPLAINT, DI_PID, DI_Create_Date)  
VALUES(-99, 'No Value Provided', 'Manually input', getdate());
```

Note: In this dataset there are some values in the dimensions such as “Unknown”, “Not Applicable”, etc. Those should be used and NOT be replaced by the -99 SK above

# Dimensions

- Cleanup
  - Dim\_TRAVEL\_DIRECTION – there should not be redundant directions descriptions so do some cleaning
  - VEHICLE\_YEAR and PERSON\_AGE have many invalid values. Next slide discusses cleaning
- Bonus
  - What other dimensions or dimensional attributes would you suggest to cleanup. You would have to have a reasonable ability to cleanup the data.

# Two Error Tables

Determine what criteria you would use to invalidate model year or person age?  
Based on that criteria populate the tables below and replace NULL with invalid value in FCT table

```
CREATE TABLE ERR_Model_Year (  
  UNIQUE_ID int NOT NULL,  
  COLLISION_ID int NULL,  
  VEHICLE_ID varchar(80),  
  CRASH_DATE date NULL,  
  VEHICLE_YEAR int NULL,  
  DI_PID varchar(20) NULL,  
  DI_Create_Date datetime DEFAULT getdate()  
  NULL,  
  PRIMARY KEY (UNIQUE_ID)  
);
```

```
CREATE TABLE ERR_Person_Age (  
  UNIQUE_ID int NOT NULL,  
  COLLISION_ID int NULL,  
  PERSON_ID varchar(80),  
  CRASH_DATE date NULL,  
  PERSON_AGE int NULL,  
  DI_PID varchar(20) NULL,  
  DI_Create_Date datetime DEFAULT getdate()  
  NULL,  
  PRIMARY KEY (UNIQUE_ID)  
);
```