//Adding Constraints

CREATE CONSTRAINT ON (Oldcars:Oldcars) ASSERT Oldcars.oldcarid IS UNIQUE;

CREATE CONSTRAINT ON (city:city) ASSERT city.city IS UNIQUE;

CREATE CONSTRAINT ON (vin:vin) ASSERT vin.vin IS UNIQUE;

CREATE CONSTRAINT ON (daysonmarket: daysonmarket) ASSERT daysonmarket. daysonmarket IS UNIQUE;

CREATE CONSTRAINT ON maximum\_seating: maximum\_seating) ASSERT maximum\_seating. maximum\_seating IS UNIQUE;

CREATE CONSTRAINT ON (city\_mileage: city\_mileage) ASSERT city\_mileage. city\_mileage IS UNIQUE;

CREATE CONSTRAINT ON (fuel\_type: fuel\_type) ASSERT fuel\_type. fuel\_type IS UNIQUE;

CREATE CONSTRAINT ON (body\_type:body\_type) ASSERT body\_type.body\_type IS UNIQUE;

CREATE CONSTRAINT ON (dealer\_zip:dealer\_zip) ASSERT dealer\_zip.dealer\_zip IS UNIQUE;

CREATE CONSTRAINT ON (price: price) ASSERT price. price IS UNIQUE;

CREATE CONSTRAINT ON (model\_name:model\_name) ASSERT model\_name.model\_name IS UNIQUE;

CREATE CONSTRAINT ON (make\_name: make\_name) ASSERT make\_name. make\_name IS UNIQUE;

CREATE CONSTRAINT ON (body\_legroom: body\_legroom) ASSERT body\_legroom. body\_legroom IS UNIQUE;

CREATE CONSTRAINT ON (is\_new:is\_new) ASSERT is\_new.is\_new IS UNIQUE;

CREATE CONSTRAINT ON (wheelbase: wheelbasedisplay) ASSERT wheelbase. wheelbasedisplay IS UNIQUE;

CREATE CONSTRAINT ON (color:exterior\_color) ASSERT color.exterior\_color IS UNIQUE;

CREATE CONSTRAINT ON (fuel\_tank\_volume: fuel\_tank\_volume) ASSERT fuel\_tank\_volume.fuel\_tank\_volume IS UNIQUE;

CREATE CONSTRAINT ON (year: year) ASSERT year.year IS UNIQUE;

CREATE CONSTRAINT ON (trimId: trimId) ASSERT trimId. trimId IS UNIQUE;

CREATE CONSTRAINT ON (engine\_type: engine\_type) ASSERT engine\_type.engine\_type IS UNIQUE;

CREATE CONSTRAINT ON (height: height) ASSERT height.height IS UNIQUE;

CREATE CONSTRAINT ON (length: length) ASSERT length.length IS UNIQUE;

CREATE CONSTRAINT ON (listed\_date:listed\_date) ASSERT listed\_date. listed\_date IS UNIQUE;

CREATE CONSTRAINT ON (power: power) ASSERT power. power IS UNIQUE;

CREATE CONSTRAINT ON (transmission:transmission) ASSERT transmission.transmission IS UNIQUE;

CREATE CONSTRAINT ON (seller\_rating: seller\_rating) ASSERT seller\_rating. seller\_rating IS UNIQUE;

// Creating Nodes

// Creating oldcars

:auto USING PERIODIC COMMIT 500

LOAD CSV With HEADERS FROM 'file:///used\_car\_clean\_CSV.csv' AS row

MERGE

(Oldcars:Oldcars { oldcarid:row.oldcarid})

// Creating City

:auto USING PERIODIC COMMIT 500

LOAD CSV With HEADERS FROM 'file:///used\_car\_clean\_CSV.csv' AS row

MERGE

(city: city {city:row.city});

// Creating VIN //////

:auto USING PERIODIC COMMIT 500

LOAD CSV With HEADERS FROM 'file:///used\_car\_clean\_CSV.csv' AS row

MERGE

(vin:vin { vin:row.vin});

// Creating daysonmarket//////

:auto USING PERIODIC COMMIT 500

LOAD CSV With HEADERS FROM 'file:///used\_car\_clean\_CSV.csv' AS row

MERGE

(daysonmarket: daysonmarket{ daysonmarket:row. daysonmarket});

// Creating Maximum\_seating//////

:auto USING PERIODIC COMMIT 500

LOAD CSV With HEADERS FROM 'file:///used\_car\_clean\_CSV.csv' AS row

MERGE

(maximum\_seating: maximum\_seating{ maximum\_seating:row. maximum\_seating});

// Creating mileage

:auto USING PERIODIC COMMIT 500

LOAD CSV With HEADERS FROM 'file:///used\_car\_clean\_CSV.csv' AS row

MERGE

(city\_mileage: city\_mileage { city\_mileage:row. city\_mileage });

// Creating Fuel\_type

:auto USING PERIODIC COMMIT 500

LOAD CSV With HEADERS FROM 'file:///used\_car\_clean\_CSV.csv' AS row

MERGE

(fuel\_type: fuel\_type{ fuel\_type:row. fuel\_type });

// Creating Body\_type

:auto USING PERIODIC COMMIT 500

LOAD CSV With HEADERS FROM 'file:///used\_car\_clean\_CSV.csv' AS row

MERGE

(body\_type: body\_type { body\_type:row. body\_type })

// Creating Dealer\_zip//////

:auto USING PERIODIC COMMIT 500

LOAD CSV With HEADERS FROM 'file:///used\_car\_clean\_CSV.csv' AS row

MERGE

(dealer\_zip: dealer\_zip{ dealer\_zip:row. dealer\_zip})

// Creating price//////

:auto USING PERIODIC COMMIT 500

LOAD CSV With HEADERS FROM 'file:///used\_car\_clean\_CSV.csv' AS row

MERGE

(price: price{ price:row. price});

// Creating Model\_name//////

:auto USING PERIODIC COMMIT 500

LOAD CSV With HEADERS FROM 'file:///used\_car\_clean\_CSV.csv' AS row

MERGE

(model\_name: model\_name{ model\_name:row. model\_name});

// Creating Make\_name//////

:auto USING PERIODIC COMMIT 500

LOAD CSV With HEADERS FROM 'file:///used\_car\_clean\_CSV.csv' AS row

MERGE

(make\_name: make\_name{ make\_name:row. make\_name});

// Creating Body\_Legroom//////

:auto USING PERIODIC COMMIT 500

LOAD CSV With HEADERS FROM 'file:///used\_car\_clean\_CSV.csv' AS row

MERGE

(body\_legroom: body\_legroom{ body\_legroom:row. body\_legroom});

// Creating Is\_new

:auto USING PERIODIC COMMIT 500

LOAD CSV With HEADERS FROM 'file:///used\_car\_clean\_CSV.csv' AS row

MERGE

(Is\_new: Is\_new{ Is\_new: row. Is\_new});

// Creating wheelbase

:auto USING PERIODIC COMMIT 500

LOAD CSV With HEADERS FROM 'file:///used\_car\_clean\_CSV.csv' AS row

MERGE

(wheel\_base: wheel\_base { wheel\_base:row. wheel\_base });

// Creating color

:auto USING PERIODIC COMMIT 500

LOAD CSV With HEADERS FROM 'file:///used\_car\_clean\_CSV.csv' AS row

MERGE

(color: exterior\_color { color:row. exterior\_color });

// Creating Fuel\_tank\_volume

:auto USING PERIODIC COMMIT 500

LOAD CSV With HEADERS FROM 'file:///used\_car\_clean\_CSV.csv' AS row

MERGE

(fuel\_tank\_volume: fuel\_tank\_volume { fuel\_tank\_volume:row. fuel\_tank\_volume });

// Creating year

:auto USING PERIODIC COMMIT 500

LOAD CSV With HEADERS FROM 'file:///used\_car\_clean\_CSV.csv' AS row

MERGE

(year: year{ year:row. year});

// Creating trimid

:auto USING PERIODIC COMMIT 500

LOAD CSV With HEADERS FROM 'file:///used\_car\_clean\_CSV.csv' AS row

MERGE

(trimId: trimId { trimId:row. trimId });

// Creating Engine\_type

:auto USING PERIODIC COMMIT 500

LOAD CSV With HEADERS FROM 'file:///used\_car\_clean\_CSV.csv' AS row

MERGE

(engine\_type: engine\_type { engine\_type:row. engine\_type });

// Creating height

:auto USING PERIODIC COMMIT 500

LOAD CSV With HEADERS FROM 'file:///used\_car\_clean\_CSV.csv' AS row

MERGE

(height: height { height:row. height });

// Creating length

:auto USING PERIODIC COMMIT 500

LOAD CSV With HEADERS FROM 'file:///used\_car\_clean\_CSV.csv' AS row

MERGE

(length: length { length:row. length });

// Creating Listed\_date

:auto USING PERIODIC COMMIT 500

LOAD CSV With HEADERS FROM 'file:///used\_car\_clean\_CSV.csv' AS row

MERGE

(listed\_date: listed\_date { listed\_date:row. listed\_date });

// Creating Transmission

:auto USING PERIODIC COMMIT 500

LOAD CSV With HEADERS FROM 'file:///used\_car\_clean\_CSV.csv' AS row

MERGE

(transmission: transmission{ transmission:row. transmission});

// Creating power

:auto USING PERIODIC COMMIT 500

LOAD CSV With HEADERS FROM 'file:///used\_car\_clean\_CSV.csv' AS row

MERGE

(power: power{ power:row. power});

// Creating seller\_rating

:auto USING PERIODIC COMMIT 500

LOAD CSV With HEADERS FROM 'file:///used\_car\_clean\_CSV.csv' AS row

MERGE

(seller\_rating: seller\_rating{ seller\_rating:row. seller\_rating});

// Creating color

:auto USING PERIODIC COMMIT 500

LOAD CSV With HEADERS FROM 'file:///used\_car\_clean\_CSV.csv' AS row

MERGE

(exterior\_color: exterior\_color { exterior\_color:row. exterior\_color });

// Creating Transmission

:auto USING PERIODIC COMMIT 500

LOAD CSV With HEADERS FROM 'file:///used\_car\_clean\_CSV.csv' AS row

MERGE

(city: city{ city:row. city});

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_