

# Car Sharing Relational Database

Design the data model given a target knowledge domain

**A company wishes to set up a car-sharing platform between private individuals with the following expectations:**

Individuals participating in the program may or may not own a car; an individual may have several cars. → **Cardinality, the user can own none or multiple cars.**

A car can be shared with an individual by providing at least the following information: a start and end time, a pick-up location. A car can be shared with different individuals, but not for a period overlapping existing sharing slots.

→ **Relation**

Each person wishing to use the car-sharing service must be registered and provide at least the following information: name, date of birth, address, valid ID number, valid driver's license, profile photo. → **To use the service, each user must be registered, giving name, birthdate, address...**

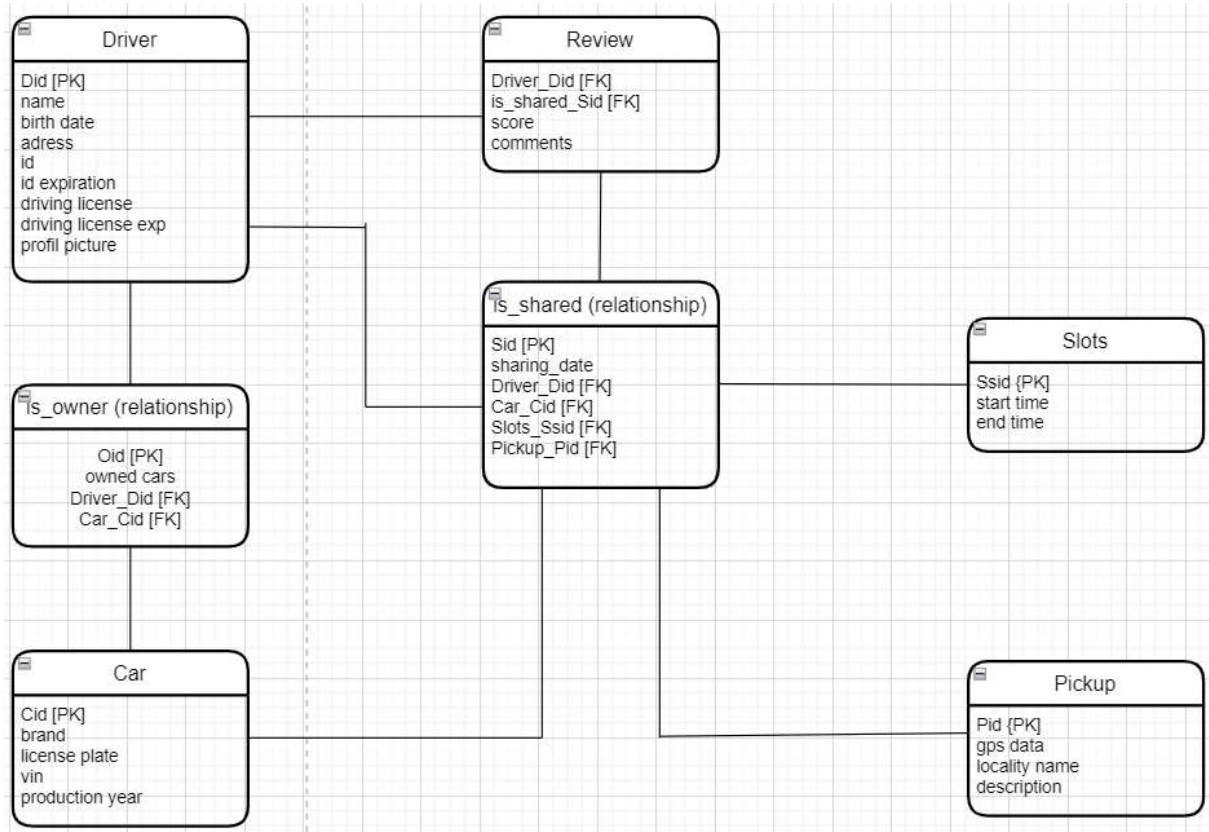
Each shared car must be registered with at least the following information: make, license plate, vehicle identification number (VIN) and year of production. → **To use the service, each car must be registered, giving brand, license plate...**

To facilitate the management of pick-up locations, a list of predefined locations will be validated and registered on the platform. Each pick-up location is defined by a name, a GPS position, and any other useful information. → **The platform must contain a list of predefined pick-up locations, with a name, or GPS data...**

To improve user confidence in the platform, each user should be able to give feedback on a carpooling experience. → **From the user, possibility to leave feedback.**

Produce a DB Model

**ER Model with integrity constraints (Normalized)**



## Produce the DB schema

1. Retrieve all individuals with the following attributes: name, id, id expiration date, driver license, driver license expiration date

```
MariaDB [carsharing2]> SELECT D.id, name, id, id_expiration, driving_license, d_license_expiration FROM Driver;
+-----+-----+-----+-----+-----+-----+
| D.id | name   | id      | id_expiration | driving_license | d_license_expiration |
+-----+-----+-----+-----+-----+-----+
| 1    | Jean Valjean | 12233445 | 1809-02-17    | 1              | 1819-03-18          |
| 2    | Cosette     | 23344556 | 1819-03-18    | 2              | 1829-04-20          |
| 3    | Javert      | 34455667 | 1829-04-20    | 3              | 1839-05-21          |
| 4    | Fantine     | 45566778 | 1839-05-21    | 4              | 1849-06-22          |
| 5    | Gavroche    | 56677889 | 1849-06-22    | 5              | 1859-07-23          |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.000 sec)
```

2. Retrieve all individuals that are car owners with the following attributes: name and car license plate

```
MariaDB [carsharing2]> SELECT D.name,
-> C.license_plate FROM Driver D INNER JOIN Car C
-> ON D.Did = C.Cid;
+-----+-----+
| name   | license_plate |
+-----+-----+
| Jean Valjean | AA001AA      |
| Cosette     | BB002BB      |
| Javert      | CC003CC      |
+-----+-----+
```

3. Retrieve all individuals associated with an active car sharing as a user.

```
MariaDB [carsharing2]> SELECT DISTINCT D.name FROM Driver D INNER JOIN Is_shared I_S ON D.Did = I_S.Sid ;
Empty set (0.000 sec)
```

➔ Nobody is sharing a car

4. Retrieve all individuals with expired ID card or expired driver license.

```
MarlaDB [carsharing2]> SELECT name,
-> Did FROM Driver WHERE id_expiration < NOW() or d_license_expiration < NOW()
-> ;
```

name	Did
Jean Valjean	1
Cosette	2
Javert	3
Fantine	4
Gavroche	5

```
5 rows in set (0.000 sec)
```

➔ Every person registered has id and license expired (due to chronological association)

5. Assuming that the access to certain pickup locations can be prohibited to old cars, provide a query that allows to retrieve all cars that are older than year 2007

```
MarlaDB [carsharing2]> SELECT * FROM Car WHERE prod_year < '2007-01-01'
-> ;
```

Cid	brand	license_plate	v_in	prod_year
1	NISSAN	AA001AA	123432	2005-02-17
2	MITSUBISHI	BB002BB	234565	2006-03-18

```
2 rows in set (0.001 sec)
```

➔ There's 2 cars manufactured before year 2007

Create the DB (Neo4j as NoSQL relational DB)

**Create samples: 10 individuals and 6 cars**

CREATE

```
(i1:Individuals {name : 'Jean Valjean', birth_date: '1769-02-17', adress: 'Rue de L Ouest Paris', id: '12233445', id_expiration: '2029-02-17', driving_l: '122334', driving_l_expiration: '2030-03-18'}), (i2:Individuals {name : 'Cosette', birth_date: '1779-03-18', adress: 'Rue de L Est Paris', id: '23344556', id_expiration: '2030-03-18', driving_l: '233445', driving_l_expiration: '2031-04-19'}), (i3:Individuals {name: 'Fantine', birth_date: '1780-04-19', adress: 'Rue du Nord Paris', id: '34455667', id_expiration: '2031-04-19', driving_l: '344556', driving_l_expiration: '2032-05-20'}), (i4:Individuals {name: 'Gavroche', birth_date: '1781-05-20', adress: 'Rue du Sud Paris', id: '45566778', id_expiration: '2032-05-20', driving_l: '455667', driving_l_expiration: '2033-06-21'}), (i5:Individuals {name: 'Javert', birth_date: '1782-06-21', adress: 'Rue de la gauche Paris', id: '56677889', id_expiration: '2033-06-21', driving_l: '566778', driving_l_expiration: '2034-07-22'}), (i6:Individuals {name: 'Marius', birth_date: '1783-07-22', adress: 'Rue de la droite Paris', id: '67788991', id_expiration: '2034-07-22', driving_l: '677889', driving_l_expiration: '2035-08-23'}), (i7:Individuals {name: 'Eponine', birth_date: '1784-08-23', adress: 'Rue du haut Paris', id: '78899112', id_expiration: '2035-08-23', driving_l: '788991', driving_l_expiration: '2036-09-24'}), (i8:Individuals {name: 'Monseigneur Myriel', birth_date: '1785-09-24', adress: 'Rue du bas Paris', id: '89911223', id_expiration: '2036-09-24', driving_l: '899112', driving_l_expiration: '2037-10-25'}), (i9:Individuals {name: 'Enjolras', birth_date: '1786-10-25', adress: 'Rue du devant Paris', id: '91122334', id_expiration: '2037-10-25', driving_l: '911223', driving_l_expiration: '2038-11-26'}), (i10:Individuals {name: 'Thénardier', birth_date: '1787-11-26', adress: 'Rue du derriere Paris', id: '12233446', id_expiration: '2038-11-26', driving_l: '122335', driving_l_expiration: '2039-12-27'});
```

CREATE

```
(c1:Cars{brand: 'NISSAN', license_plate: 'AA001AA', v_in: '123456', prod_year: '2005'}), (c2:Cars{brand: 'MITSUBISHI', license_plate: 'BB002BB', v_in: '234567', prod_year: '2005'}), (c3:Cars{brand: 'TOYOTA', license_plate: 'CC003CC', v_in: '345678', prod_year: '2006'}), (c4:Cars{brand: 'SUZUKI', license_plate: 'DD004DD', v_in: '456789', prod_year: '2006'}), (c5:Cars{brand: 'HONDA', license_plate: 'EE005EE', v_in: '567891', prod_year: '2007'}), (c6:Cars{brand: 'SUBARU', license_plate: 'FF006FF', v_in: '678912', prod_year: '2007'});
```

**Create relationships that allow to assign the 6 cars to 6 different individuals.**

MATCH

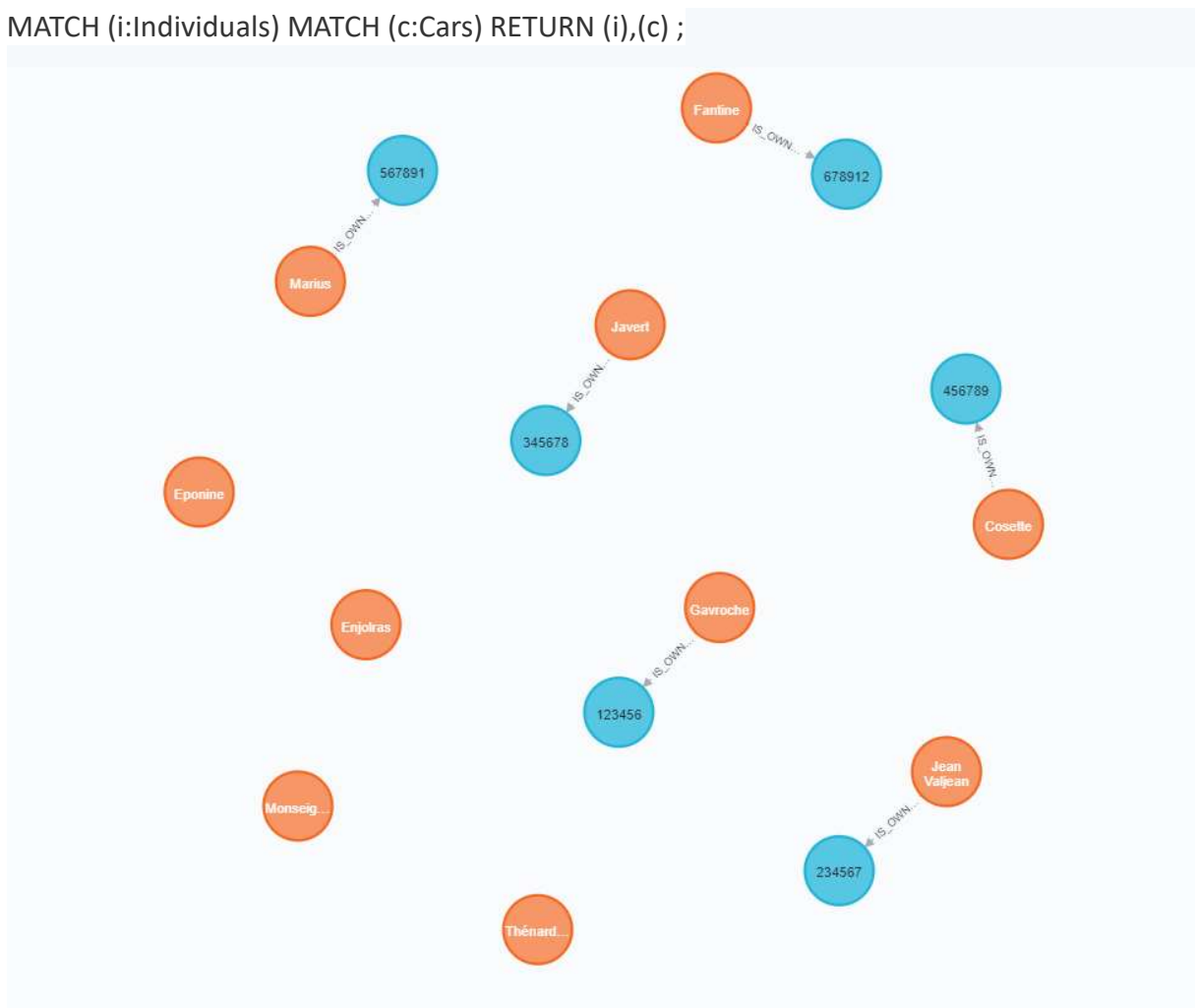
```
(i1:Individuals {name : 'Jean Valjean'}), (c2:Cars {v_in: '234567'}), (i2:Individuals {name : 'Cosette'}), (c4:Cars {v_in: '456789'}), (i3:Individuals {name : 'Fantine'}), (c6:Cars {v_in: '678912'}), (i4:Individuals {name : 'Gavroche'}), (c1:Cars {v_in: '123456'}), (i5:Individuals {name : 'Javert'}), (c3:Cars {v_in: '345678'}), (i6:Individuals {name : 'Marius'}), (c5:Cars {v_in: '567891'})
```

CREATE

```
(i1)-[OWN1:IS_OWNED_BY{sharing_date:'2022-01-25'}]->(c2),  
(i2)-[OWN2:IS_OWNED_BY{sharing_date:'2022-01-26'}]->(c4),  
(i3)-[OWN3:IS_OWNED_BY{sharing_date:'2022-01-27'}]->(c6),  
(i4)-[OWN4:IS_OWNED_BY{sharing_date:'2022-01-28'}]->(c1),  
(i5)-[OWN5:IS_OWNED_BY{sharing_date:'2022-01-29'}]->(c3),  
(i6)-[OWN6:IS_OWNED_BY{sharing_date:'2022-01-30'}]->(c5);
```

**Export the state of the DB as a PNG file**

```
MATCH (i:Individuals) MATCH (c:Cars) RETURN (i),(c) ;
```



**Create 10 car sharing relationships (one relation for each registered individual). Each car sharing slot shall include the following information: a pickup coordinates (longitude, latitude), a start and an end date in format YYYY-MM-DD HH:MM:SS. Make sure that for a given car, the different car sharing slots do not overlap with each other**

MATCH

(i1:Individuals {name:'Jean Valjean'}),

(i2:Individuals {name : 'Cosette'}),

(i3:Individuals {name : 'Fantine'}),

(i4:Individuals {name:'Gavroche'}), (i5:Individuals {name:'Javert'}), (i6:Individuals {name:'Marius'}), (i7:Individuals {name:'Eponine'}), (i8:Individuals {name:'Monseigneur Myriel'}), (i9:Individuals {name:'Enjolras'}), (i10:Individuals {name:'Thénardier'}),

(c1:Cars{v\_in:'123456'}), (c2:Cars{v\_in:'234567'}), (c3:Cars{v\_in:'345678'}),

(c4:Cars{v\_in:'456789'}), (c5:Cars{v\_in:'567891'}), (c6:Cars{v\_in:'678912'})

CREATE

(i1)-[BWRD1:HAS\_BORROWED{pickup\_long:'2.3339772',  
pickup\_lat:'48.8782846',start\_time:'2022-01-25 09:00:00',end\_time:'2022-01-25 09:30:00',  
bor\_review:'Thanks for the ride',r\_date:'2022-01-25 09:31:00'}}->(c3),

(i2)-[BWRD2:HAS\_BORROWED{pickup\_long:'2.3030724',  
pickup\_lat:'48.8707328',start\_time:'2022-01-25 10:00:00',end\_time:'2022-01-25 10:30:00',  
bor\_review:'Awesome',r\_date:'2022-01-25 10:31:00'}}->(c5),

(i3)-[BWRD3:HAS\_BORROWED{pickup\_long:'2.3247895',  
pickup\_lat:'48.8547572',start\_time:'2022-01-25 11:00:00',end\_time:'2022-01-25 11:30:00',  
bor\_review:'A bit too fast but thanks',r\_date:'2022-01-25 11:31:00'}}->(c1),

(i4)-[BWRD4:HAS\_BORROWED{pickup\_long:'2.3427994',  
pickup\_lat:'48.8528126',start\_time:'2022-01-25 12:00:00',end\_time:'2022-01-25 12:30:00',  
bor\_review:'long way, straight ride',r\_date:'2022-01-25 12:31:00'}}->(c2),

(i5)-[BWRD5:HAS\_BORROWED{pickup\_long:'2.3486659',  
pickup\_lat:'48.8458675',start\_time:'2022-01-26 09:00:00',end\_time:'2022-01-26 09:30:00',  
bor\_review:'Delay solved thx',r\_date:'2022-01-26 09:31:00'}}->(c4),

(i6)-[BWRD6:HAS\_BORROWED{pickup\_long:'2.3499905',  
pickup\_lat:'48.8601987',start\_time:'2022-01-26 10:00:00',end\_time:'2022-01-26 10:30:00',  
bor\_review:'I waited too long please be more on time',r\_date:'2022-01-26 10:31:00'}}->(c6),

(i7)-[BWRD7:HAS\_BORROWED{pickup\_long:'2.3594438',  
pickup\_lat:'48.8644249',start\_time:'2022-01-26 11:00:00',end\_time:'2022-01-26 11:30:00',  
bor\_review:'Kind driver',r\_date:'2022-01-26 11:31:00'}}->(c1),

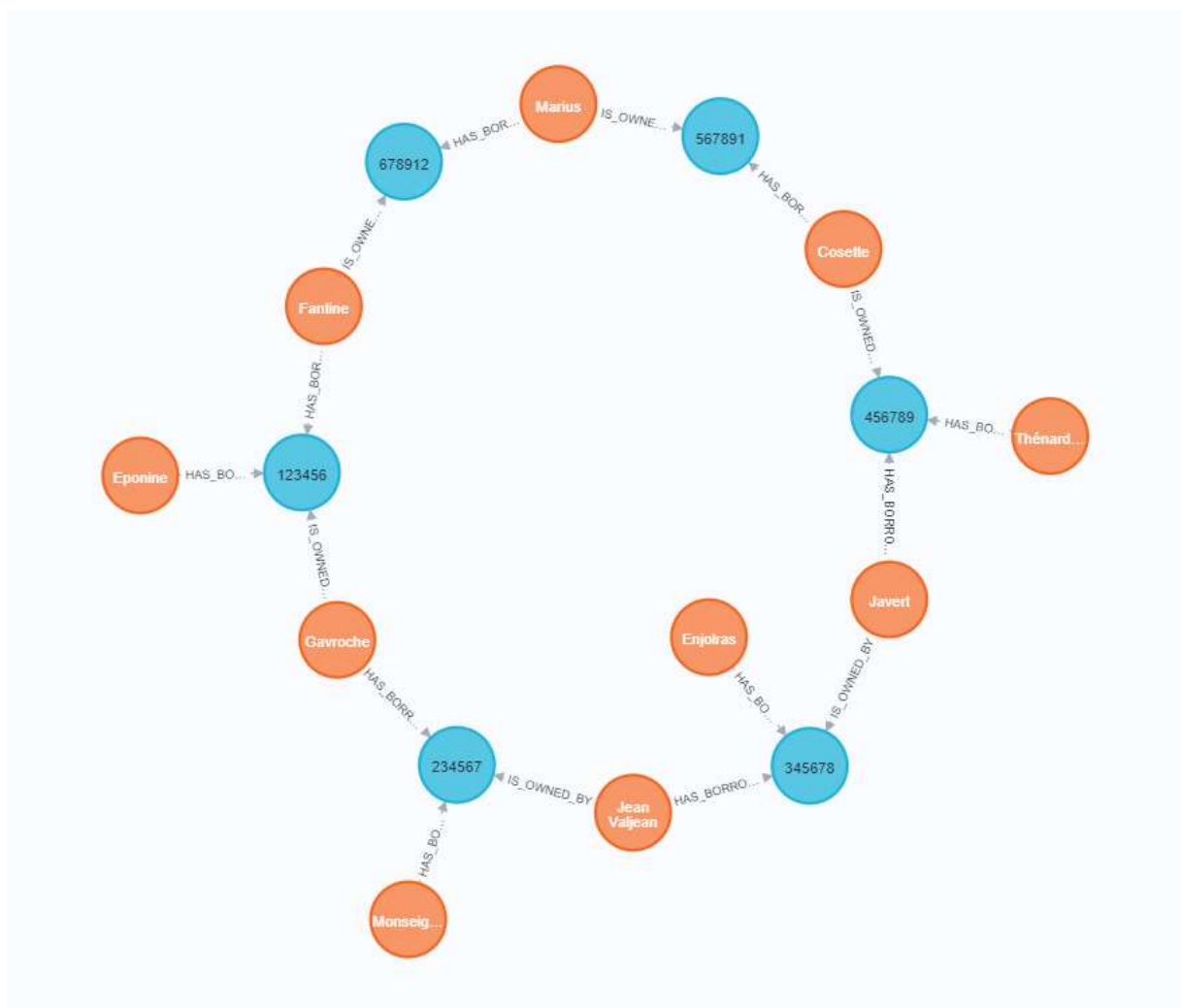
(i8)-[BWRD8:HAS\_BORROWED{pickup\_long:'2.3487999',  
pickup\_lat:'48.8643041',start\_time:'2022-01-26 12:00:00',end\_time:'2022-01-26 12:30:00',  
bor\_review:'OMG',r\_date:'2022-01-26 12:31:00'}]->(c2),

(i9)-[BWRD9:HAS\_BORROWED{pickup\_long:'2.3424388',  
pickup\_lat:'48.8620604',start\_time:'2022-01-27 09:00:00',end\_time:'2022-01-27 09:30:00',  
bor\_review:'Yoo that was a slow ride',r\_date:'2022-01-27 09:31:00'}]->(c3),

(i10)-[BWRD10:HAS\_BORROWED{pickup\_long:'2.3496389',  
pickup\_lat:'48.8741659',start\_time:'2022-01-27 10:00:00',end\_time:'2022-01-27 10:30:00',  
bor\_review:'Stinking driver',r\_date:'2022-01-27 10:31:00'}]->(c4) ;

### Export the state of the DB as a PNG file

MATCH (i:Individuals) MATCH (c:Cars) RETURN (i),(c) ;





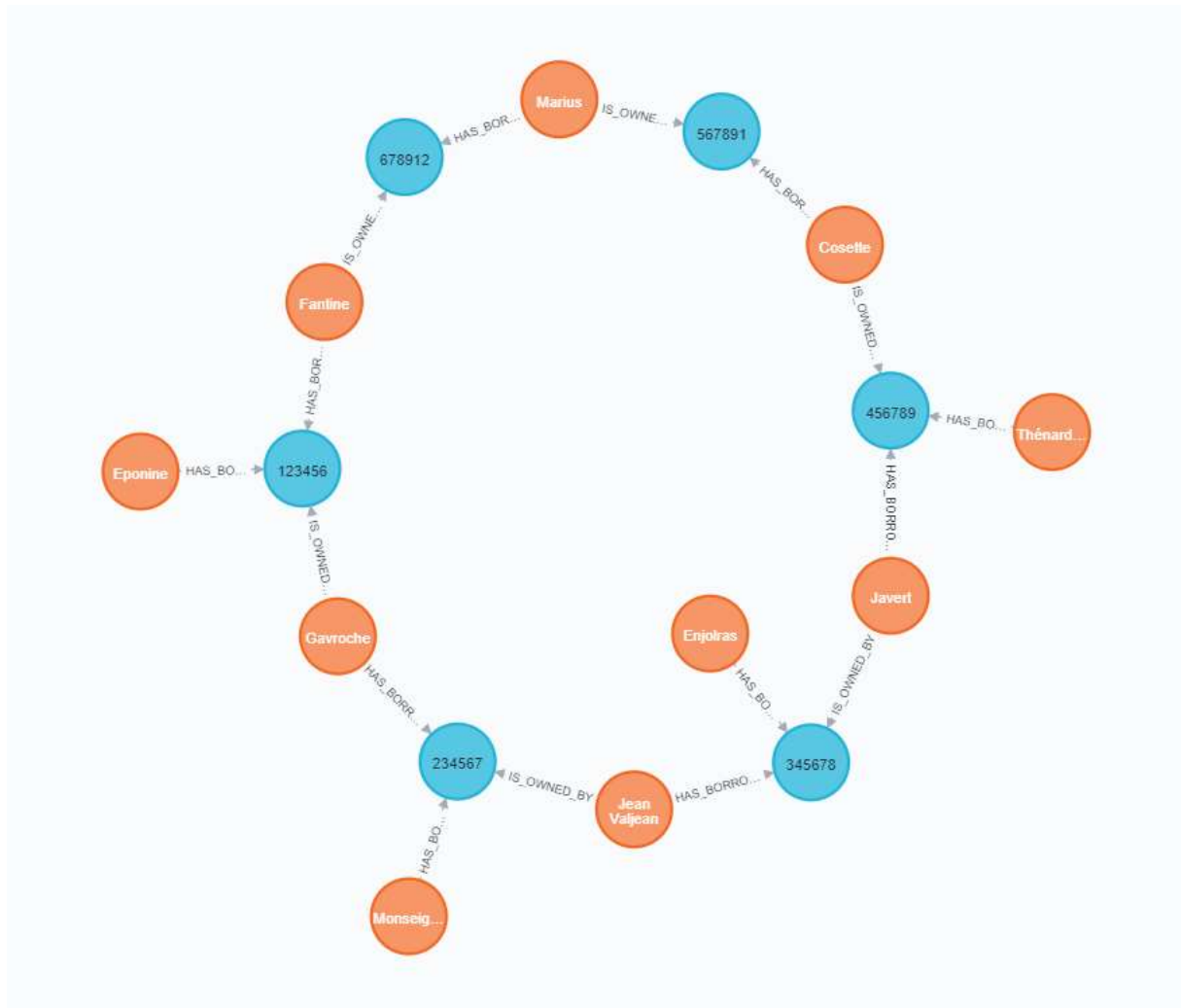
**Given an individual (e.g. i1) and a car (c1) with a car sharing relationship between them, provide a query to update the car sharing to set an return date, as well as a review comment**

```
MATCH (i1{name:'Jean Valjean'})-[return:HAS_BORROWED]-(c3{v_in:'345678'})  
SET return.r_date = '2022-01-25 09:32:00', return.bor_review = 'Thanks for the Chop' ;
```

**Perform the previous query on three relationships of your choice**

```
MATCH (i2{name:'Cosette'})-[return:HAS_BORROWED]-(c4{v_in:'456789'})  
SET return.r_date = '2022-01-25 10:32:00', return.bor_review = 'What a car' ;  
MATCH (i3{name:'Fantine'})-[return:HAS_BORROWED]-(c5{v_in:'567891'})  
SET return.r_date = '2022-01-25 11:32:00', return.bor_review = 'Brakes not found' ;  
MATCH (i4{name:'Gavroche'})-[return:HAS_BORROWED]-(c2{v_in:'234567'})  
SET return.r_date = '2022-01-25 12:32:00', return.bor_review = 'Crazy Ride as hell' ;
```

## Export DB



**Provide a query to match all individuals' owners of a car. The query shall display only the following information: individual (name, id number, driver license), car (license plate, brand, year of production). Export the output as a plain text file**

```
MATCH (i:Individuals)
```

```
MATCH (c:Cars)
```

```
WHERE (i)-[:IS_OWNED_BY]-(c)
```

```
RETURN (i.name), (i.id), (i.driving_l),(c.license_plate), (c.brand), (c.prod_year) ;
```

"(i.name)"	"(i.id)"	"(i.driving_l)"	"(c.license_plate)"	"(c.brand)"	"(c.prod_year)"
"Jean Valjean"	"12233445"	"122334"	"BB002BB"	"MITSUBISHI"	"2005"
"Cosette"	"23344556"	"233445"	"DD004DD"	"SUZUKI"	"2006"
"Fantine"	"34455667"	"344556"	"FF006FF"	"SUBARU"	"2007"
"Gavroche"	"45566778"	"455667"	"AA001AA"	"NISSAN"	"2005"
"Javert"	"56677889"	"566778"	"CC003CC"	"TOYOTA"	"2006"
"Marius"	"67788991"	"677889"	"EE005EE"	"HONDA"	"2007"

**Provide a query to match all individuals without a car. Export the output as a CSV file (options in Neo4j)**

```
MATCH (i:Individuals)
MATCH (c:Cars)
WHERE NOT (i)-[:IS_OWNED_BY]-(c)
RETURN (i.name) ;
```

**Provide a query that allows to match all completed car sharing slots (i.e. with return time and optionally a review comment).**

```
MATCH (i)-[h:HAS_BORROWED]-(c)
WHERE h.end_date < localtime()
RETURN i
```

All the sharing rides are returned since they are all completed.

**Provide a query that allows to match all non-completed car sharing slots**

```
MATCH (i)-[h:HAS_BORROWED]-(c)
WHERE h.end_date > localtime()
RETURN i
```

No sharing ride is returned since all the rides are completed.