Databases Project

Student Name: Rami Soussi

Student Number : 0583600

* I sent with this report 4 other documents :

- diagram.drawio (question 1)

- creation.sql (question 3)

- insertion.sql (question 3)

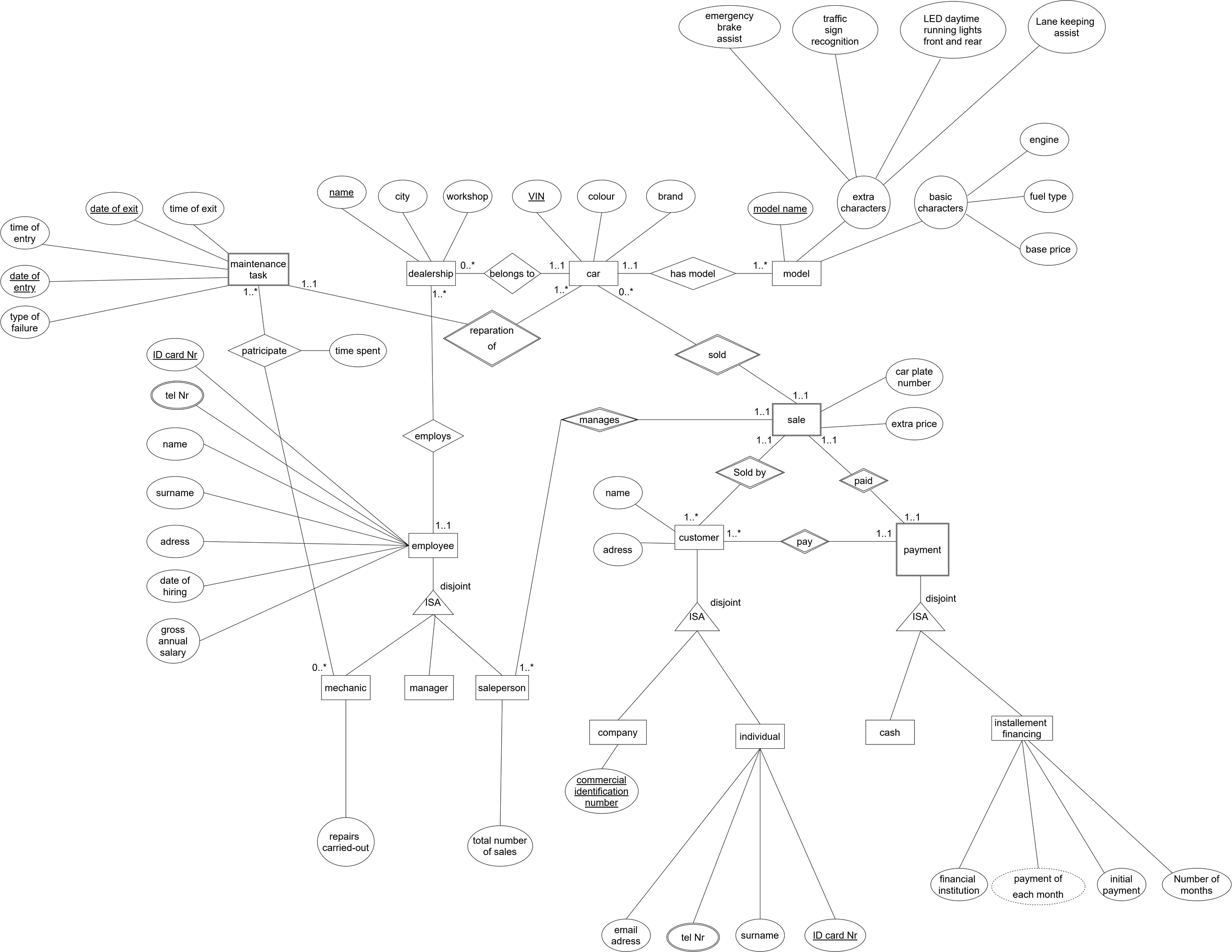
- queries.sql (question 4)

Question 1: (E)ER Modelling

The entities in my diagram are:

- Dealership : the entity of dealership has dealership_name as a primary key and two attributes that are city and the its the name of it's workshop. It has a relation with employee, maintenance task and car
- Car : this entity has the attributes VIN (primary key) , colour and brand. Each car is related to the dealership to which it belongs, it's model and its sale operation.
- Model : this entity has the model_name as a primary key, extra characteristics (LED daytime running lights front and rear, Traffic Sign Recognition, Emergency Brake Assist, Lane Keeping Assist) and basic characteristics (engine, fuel type and base price)
- Sale : this entity is a weak entity because it doesn't have it's primary key. It has the attributes car plate number and extra_price, and it's has relations with : customer, payment, car and maintenance task.
- Customer : this is the customer who buys the car. He has the entities name (primary key) and adress. It has a disjointness constraint because the customer can be either a company or individual.
- Company : this entity is the company that buys the car, it has the primary key : commercial identification number
- Individual : this is the individual customer. He has the attributes ID card Nr (primary key), telephone number, email adress and surname.
- Payment : this is the payment of the car sale. It's a weak entity because it doesn't have a primary key. It has a disjointness constraints because it can be either cash or installement financing.
- Cash : the first method of payment

- Installement financing : this the second method of payment. It has the attributes financial institution, payment of each month, initial payment and the number of months.
- Employee : this entity is the employee of the dealership. It has a disjointness constraint because each employee could be either a mechanic, manager or a saleperson. The employee has the following attributes : Id card Nr, tel Nr, name, surname, adress, date of hiring, gross annual salary
- Mechanic : this is the professional catergory of mechanics. They have the attribute: number of repairs he carried out. This entity has a relation with maintenance task because the machanic participate in those tasks.
- Manager : this the second professional category in the dealership
- Saleperson : This is the employee responsible of the sales. He has the attribute: total number of sales. He has a relation with the attribute sale since he manages the sales of the dealership cars.
- Maintenance task : this entity has the attributes date of entry and exit (primary keys), time of entry and exit, and the type of failure (Alternator, Electrical, Brakes, Clutch...). It has a relation with sale since the car repaired will be sold, and xith mechanics who participate in this task.


emergency brake assist
traffic sign recognition
LED daytime running lights front and rear
Lane keeping assist
engine
fuel type
base price
extra characters
basic characters
model name
name
city
workshop
VIN
colour
brand
date of exit
time of exit
time of entry
date of entry
type of failure
maintenance task
dealership
0..*
belongs to
1..1
car
1..1
has model
1..*
model
1..*
1..1
1..*
1..*
0..*
reparation of
sold
patricipate
time spent
ID card Nr
tel Nr
name
surname
adress
date of hiring
gross annual salary
employs
1..1
employee
disjoint
ISA
0..*
1..*
mechanic
manager
saleperson
repairs carried-out
total number of sales
manages
1..1
sale
car plate number
extra price
1..1
1..1
Sold by
paid
name
adress
1..*
customer
1..*
pay
1..1
payment
1..1
1..1
disjoint
ISA
disjoint
ISA
company
individual
cash
installement financing
commercial identification number
email adress
tel Nr
surname
ID card Nr
financial institution
payment of each month
initial payment
Number of months

Question 2: (E)ER Reducing, Functional Dependencies and Normalisation

(E)ER Reducing :

The schemas are :

Dealership (dealership_name, city, workshop)

Mechanic (ID card Nr, name, surname, tel_Nr, adress, date_of_hiring, gross_annual_salary, repairs_carried_out, dealership_name)

Saleperson (ID card Nr, name, surname, tel_Nr, adress, date_of_hiring, gross_annual_salary, total_number_of_sales, dealership_name)

Manager (ID card Nr, name, surname, tel_Nr, adress, date_of_hiring, gross_annual_salary,

dealership_name)

Car (VIN, volour, brand, model_name, dealership_name)

Model (model_name, LED_daytime_running_lights_front_and_rear, Traffic_Sign_Recognition, Emergency_Brake_Assist, Lane Keeping Assist, engine, fuel_type, base_price)

Sale (VIN, customer_ID, salesperson_ID, car_plate_number, extra_price)

Cash (VIN, customer_ID, salesperson_ID)

Installement_financing (VIN, customer_ID, salesperson_ID, financial_institution, payment_of_each_month, initial_payment, number_of_months)

Customer_company (commercial_identification_number, name, adress)

Individual_customer (ID_card_Nr, name, surname, email adress, tel Nr, adress)

Maintenance_task (VIN, date_of_entry, date_of_exit, time_of_entry, time_of_exit, type_of_failure)

Repair_participation (VIN, date_of_entry, date_of_exit, mechanic_ID_card_Nr, time-spent)

Functional dependencies :

- Dealership :

dealership_name ⟵ city, workshop

- Mechanic :

ID_card_Nr ⟵ name, surname, tel_Nr, adress, date_of_hiring, gross_annual_salary, repairs_carried_out, dealership_name

- Salesperson :

Saleperson_ID_card_Nr ⟵ name, surname, tel_Nr, adress, date_of_hiring, gross_annual_salary, total_number_of_sales, dealership_name

- Manager :

Manager_ID_card_Nr ⟵ name, surname, tel_Nr, adress, date_of_hiring, gross_annual_salary,

dealership_name

- Car :

VIN ⟵ colour, brand, model_name, dealership_name

Model_name ⟵ brand

- Model :

model_name ⟵ LED_daytime_running_lights_front_and_rear, Traffic_Sign_Recognition, Emergency_Brake_Assist, Lane Keeping Assist, engine, fuel_type, base_price

Engine, fuel_type ⟵ base_price

- Sale :

VIN, customer_ID, salesperson_ID ⟵ car_plate_number, extra_price

VIN ⟵ car_plate_number

- Installement_financing :

VIN, customer_ID, salesperson_ID ⟵ financial_institution, payment_of_each_month, initial_payment, number_of_months

VIN, customer_ID, salesperson_ID, initial_payment, number of months ⟵payment_of_each_month

- Customer_company :

commercial_identification_number ⟵ name, adress

- Individual_customer :

ID_card_Nr ⟵ name, surname, email adress, tel Nr, adress

- Maintenance_task :

VIN, date_of_entry, date_of_exit ⟵ time_of_entry, time_of_exit, type_of_failure

- Repair_participation :

VIN, date_of_entry, date_of_exit, mechanic_ID_card_Nr ⟵ time-spent

Normalisation :

- Dealership :

Dealership (dealership_name, city, workshop)

- Mechanic :

Mechanic (ID card Nr, name, surname, adress, date_of_hiring, gross_annual_salary, repairs_carried_out, dealership_name)

Mechanic_phone (ID card Nr, phone_Nr)

- Salesperson :

Saleperson (ID card Nr, name, surname, adress, date_of_hiring, gross_annual_salary, total_number_of_sales, dealership_name)

Salesperson_phone (ID card Nr, phone_Nr)

- Manager :

Manager (ID card Nr, name, surname, adress, date_of_hiring, gross_annual_salary,

dealership_name)

Manager_phone (ID card Nr, phone_Nr)

- Car:

Car (VIN, volour, model_name, dealership_name)

Car_brand(brand, model_name)

- Model :

Model (model_name, LED_daytime_running_lights_front_and_rear, Traffic_Sign_Recognition, Emergency_Brake_Assist, Lane Keeping Assist, engine, fuel_type)

Basic_characteristics (model_name, engine, fuel_type, base_price)

- Sale :

Sale (VIN, customer_ID, salesperson_ID, extra_price)

Car_sold (VIN, car_plate_number)

- Cash :

Cash (VIN, customer_ID, salesperson_ID)

- Installement financing :

Installement_financing (VIN, customer_ID, salesperson_ID, financial_institution, initial_payment, number_of_months)

Monthly_payment(VIN, customer_ID, salesperson_ID, initial_payment, number_of_months, payment_of_each_month)

- Customer_company :

Customer_company (commercial_identification_number, name, adress)

- Individual_customer :

Individual_customer (ID_card_Nr, name, surname, email adress, adress)

Individual_customer _phone (ID card Nr, phone_Nr)

- Maintenance_task :

Maintenance_task (VIN, date_of_entry, date_of_exit, time_of_entry, time_of_exit, type_of_failure)

- Repair_participation :

Repair_participation (VIN, date_of_entry, date_of_exit, mechanic_ID_card_Nr, time-spent)

Question 3: SQL (Creating):

Sql creation instruction :

CREATE TABLE dealership (dealership_name varchar(50), city varchar(20), workshop varchar(20), PRIMARY KEY (dealership_name));	CREATE table manager (IDcardNr bigint, name varchar(20), surname varchar(20), adress varchar(50), gross_annual_salary integer, date_of_hiring Date, dealership_name varchar(50), PRIMARY KEY (IDcardNr), FOREIGN KEY (dealership_name) REFERENCES dealership(dealership_name));
CREATE table salesperson (IDcardNr bigint, name varchar(20), surname varchar(20), adress varchar(50), gross_annual_salary integer, date_of_hiring Date, total_number_of_sales integer, dealership_name varchar(50), PRIMARY KEY (IDcardNr), FOREIGN KEY (dealership_name) REFERENCES dealership(dealership_name));	CREATE table mechanic (IDcardNr bigint, name varchar(20), surname varchar(20), adress varchar(50), gross_annual_salary integer, date_of_hiring Date, repairs_carried_out integer, dealership_name varchar(50), PRIMARY KEY (IDcardNr), FOREIGN KEY (dealership_name) REFERENCES dealership(dealership_name));
CREATE table mechanic_phone (IDcardNr bigint, phone_Nr bigint , PRIMARY key (phone_Nr), FOREIGN KEY (IDcardNr) REFERENCES mechanic(IDcardNr));	CREATE table salesperson_phone (IDcardNr bigint, phone_Nr bigint , PRIMARY key (phone_Nr), FOREIGN KEY (IDcardNr) REFERENCES salesperson(IDcardNr));
CREATE table manager_phone (IDcardNr bigint, phone_Nr bigint , PRIMARY key (phone_Nr),	CREATE TABLE car (VIN BIgint, colour varchar(20), model_name varchar(20), dealership_name varchar(50),

```
  FOREIGN KEY (IDcardNr) REFERENCES
manager(IDcardNr));
```

```
  PRIMARY KEY (VIN),
  FOREIGN KEY (dealership_name) REFERENCES
dealership(dealership_name),
  FOREIGN KEY (model_name) REFERENCES
model(model_name));
```

```
CREATE TABLE car_brand (
  model_name varchar(20),
  brand varchar(20),
  PRIMARY KEY (model_name),
  FOREIGN KEY (model_name) REFERENCES
car(model_name) );
```

```
CREATE TABLE model (
  model_name varchar(20),
  engine varchar(20),
  fuel_type varchar(20),
  lane_keeping_assist varchar(20),
  LED_daytime_running_lights_front_and_rear
varchar(20),
  Traffic_sign_recognition varchar(20),
  Emergency_brake_assist varchar(20),
  PRIMARY KEY (model_name));
```

```
CREATE TABLE basic_characteristics (
  model_name varchar(20),
  engine varchar(20),
  fuel_type varchar(20),
  base_price integer,
  PRIMARY KEY (model_name, engine, fuel_type));
```

```
CREATE TABLE sale (
  customer_ID bigint,
  VIN bigint,
  sales_person_idcardNr bigint,
  extra_price integer,
  date_of_sale date,
  PRIMARY KEY (customer_ID, VIN,
sales_person_idcardNr));
```

```
CREATE TABLE car_sold (
  VIN bigint,
  car_plate_number bigint,
  PRIMARY key (VIN),
  FOREIGN KEY (VIN) REFERENCES sale(VIN));
```

```
create TABLE payment (
  customer_ID bigint,
  VIN bigint,
  sales_person_idcardNr bigint,
  PRIMARY KEY (customer_ID, VIN,
sales_person_idcardNr));
```

```
CREATE TABLE cash (
  customer_ID bigint,
  VIN bigint,
  sales_person_idcardNr bigint,
  PRIMARY KEY (customer_ID, VIN,
sales_person_idcardNr),
  FOREIGN key (customer_ID, VIN,
sales_person_idcardNr) REFERENCES
payment(customer_ID, VIN, sales_person_idcardNr)
);
```

```
CREATE table installement_financing(
  customer_ID bigint,
  VIN bigint,
  sales_person_idcardNr bigint,
  financial_institution varchar(50),
  initial_payment integer,
  number_of_months integer,
  PRIMARY KEY (customer_ID, VIN,
sales_person_idcardNr)
  FOREIGN key (customer_ID, VIN,
sales_person_idcardNr) REFERENCES
payment(customer_ID, VIN,
sales_person_idcardNr));
```

```
CREATE TABLE monthly_payment (
  customer_ID bigint,
  VIN bigint,
  sales_person_idcardNr bigint,
  initial_payment integer,
  number_of_months integer,
  payment_of_each_month integer,
```

```
create TABLE customer_company (
  commercial_identification_number bigint,
  company_name varchar(50),
  adress varchar(50),
  PRIMARY KEY (commercial_identification_number)
  );
```

```
  PRIMARY key (customer_ID, VIN,
sales_person_idcardNr,initial_payment,
number_of_months)
  FOREIGN KEY (customer_ID, VIN,
sales_person_idcardNr,initial_payment,
number_of_months) REFERENCES
installement_financing(customer_ID, VIN,
sales_person_idcardNr,initial_payment,
number_of_months));
```

```
CREATE TABLE individual_customer (
  IDcardNr bigint,
  indiv_name varchar(20),
  surname varchar(20),
  email_adress varchar(20),
  adress varchar(50),
  PRIMARY KEY (IDcardNr)
  );
```

```
CREATE table individual_customer_phone (
  IDcardNr bigint,
  phone_Nr bigint ,
  PRIMARY key (phone_Nr),
  FOREIGN KEY (IDcardNr) REFERENCES
individual_customer(IDcardNr));
```

```
CREATE TABLE maintenance_task (
  VIN varchar(50),
  date_of_entry date,
  time_of_entry time,
  date_of_exit date,
  time_of_exit time,
  type_of_failure varchar(50),
  PRIMARY KEY (VIN, date_of_entry, date_of_exit),
  FOREIGN KEY (VIN) REFERENCES car(VIN);
```

```
create Table repair_participation (
  VIN varchar(50),
  date_of_entry date,
  date_of_exit date,
  mechanic_ID_card varchar(50),
  time_spent integer,
  PRIMARY KEY
(VIN,date_of_entry,date_of_exit,mechanic_ID_card)
,
  FOREIGN KEY (VIN,date_of_entry,date_of_exit)
REFERENCES
maintenance_task(VIN,date_of_entry,date_of_exit),
  FOREIGN KEY (mechanic_ID_card) REFERENCES
mechanic(IDcardNr));
```

Insertion of data :

```
insert into dealership
VALUES ('dealership 1', 'Brussels' , 'workshop 1'),
      ('dealership 2', 'Anderlecht' , 'workshop 2'),
      ('dealership 3', 'Genk' , 'workshop 3'),
      ('dealership 4', 'Liége' , 'workshop 4');
```

```
insert into car
VALUES (2568871256784,'red','clio','dealership 1'),
      (5968411266587,'white','c4','dealership 1'),
      (3687452632569,'red','kia rio','dealership 2'),
      (3236985698412,'black','megane','dealership 3'),
      (2568876580235,'blue','peugeot 206','dealership 3'),
      (6352558569840,'white','seat ibiza','dealership 4');
```

```
insert into model
VALUES ('clio','V Engine','diesel','A','B','C','D'),
      ('c4','Twin Cylinders','gasoline','A','B','C','D'),
      ('kia rio','Twin Cylinders','electric','A','B','C','D'),
      ('megane','Four Cylinders','hybrid','A','B','C','D'),
      ('peugeot 206','V Engine','hybrid','A','B','C','D'),
      ('seat ibiza','Four Cylinders','electric','A','B','C','D');
```

```
insert into basic_characteristics
VALUES ('clio','V Engine','diesel',40),
      ('c4','Twin Cylinders','gasoline',74),
      ('kia rio','Twin Cylinders','electric',55),
      ('megane','Four Cylinders','hybrid',39),
      ('peugeot 206','V Engine','hybrid',26),
      ('seat ibiza','Four Cylinders','electric',68);
```

```
insert into car_brand
```

```
insert into customer_company
```

```
VALUES ('clio','renault'),
    ('c4','citroen'),
    ('kia rio','kia'),
    ('megane','renault'),
    ('peugeot 206','peugeot'),
    ('seat ibiza','seat');
```

```
VALUES (5669655412236,'Pfizer','Brussels'),
    (5963287412562,'Adidas','Paris'),
    (1256987126325,'Carrefour','Liége');
```

```
insert into sale
VALUES (5669655412236, 2568871256784, 896554474223, 42, '2020-04-23'),
        (5695332158412, 5968411266587, 558744566320, 89, '2021-03-22'),
        (5963287412562, 3687452632569, 786210236021, 40, '2020-02-21'),
        (1256984223665, 3236985698412, 896523654122, 56, '2021-01-25'),
        (1256987126325, 2568876580235, 589365423658, 67, '2020-03-28'),
        (1569874236587, 6352558569840, 236589745213, 57, '2021-02-19');
```

```
insert into maintenance_task
VALUES (2568871256784,'2021-03-28','09:30:00','2021-04-15','11:30:00','Brakes'),
        (2568871256784,'2021-04-28','09:30:00','2021-05-16','11:30:00',' Alternator '),
        (5968411266587,'2021-04-22','09:10:00','2021-05-02','09:00:00','Alternator'),
        (3687452632569,'2021-01-22','10:30:00','2021-03-27','08:20:00','Clutch'),
        (3687452632569,'2021-04-23','10:30:00','2021-05-02','08:20:00',' Alternator ');
```

```
insert into salesperson
VALUES (896554474223,'reda','saloui','Paris',420,'2010-04-23',20,'dealership 1'),
    (558744566320,'tylor','vardey','New york',700,'2009-03-22',22,'dealership 1'),
    (786210236021,'sami','chalbi','Marseille',300,'2005-02-21',30,'dealership 1'),
    (896523654122,'alex','durand','Milano',560,'2011-01-25',18,'dealership 3'),
    (589365423658,'mario','gilardini','Liege',640,'2012-03-28',18,'dealership 3'),
    (236589745213,'augustine','fresnel','Bruxelles',560,'2013-02-19',16,'dealership 3');
```

```
insert into installement_financing
VALUES (5669655412236,2568871256784,896554474223,'BNP Paribas',5,10),
    (5963287412562,3687452632569,786210236021,'ING',10,15),
    (1256987126325,2568876580235,589365423658,'Societé générale',11,16),
    (1569874236587,6352558569840,236589745213,'BNP Paribas',8,12);
```

```
insert into mechanic
VALUES (8965544742255,'rami','sodi','Paris',420,'2010-04-23',35,'dealership 1'),
    (5587445663458,'hichem','vardey','New york',700,'2009-03-22',30,'dealership 1'),
    (4586210236021,'alexandre','parout','Marseille',300,'2005-02-21',50,'dealership 2'),
    (5698523654122,'alex','durand','Milano',560,'2011-01-25',45,'dealership 2'),
    (8963265423658,'mario','gustavo','Liege',640,'2012-03-28',25,'dealership 2'),
    (2365699745213,'augustin','frame','Bruxelles',560,'2013-02-19',21,'dealership 4');
```

```
insert into repair_participation
VALUES (2568871256784,'2021-03-28','2021-04-15',8965544742255,120),
    (3687452632569,'2021-04-28','2021-05-16',5587445663458,160),
    (5968411266587,'2021-04-22','2021-05-02',4586210236021,200),
    (3236985698412,'2021-01-22','2021-03-27',5698523654122,320),
    (6352558569840,'2021-04-23','2021-05-02',2365699745213,230);
```

Question 4: Relational Algebra, SQL and Optimising

1)

a) SQL Query :

```
SELECT dealership_name , count(*) as total_amount_of_cars
FROM car
GROUP by dealership_name ;
```

b) Relational Algebra Formula :

$$R \leftarrow \text{dealership\_name } \mathcal{G} \text{ count(VIN) } \textbf{AS} \text{ total\_amount\_of\_cars } (\text{car})$$

c) We cannot optimize this query

2)

a) SQL Query :

```
SELECT VIN , dealership_name ,date_of_sale
FROM car
JOIN model
USING (model_name)
JOIN basic_characteristics
USING (model_name,engine,fuel_type)
join sale
USING (VIN)
```

where base_price < 100

AND date_of_sale > '2021-03-21';

b) Relational Algebra Formula :

$$R1 \leftarrow (((\text{car} \bowtie \text{model}) \bowtie \text{basic\_characteristics}) \bowtie \text{sale})$$

$$R \leftarrow \pi_{\text{VIN , dealership\_name ,date\_of\_sale}} (\sigma_{\text{base\_price} < 100 \wedge \text{date\_of\_sale} > \text{'2021-03-21'}} (R1))$$

c) we can optimize this query by :

$$R1 \leftarrow \pi_{\text{model\_name,engine,fuel\_type}} (\sigma_{\text{base\_price} < 100} (\text{basic\_characteristics}))$$

$$R2 \leftarrow \pi_{\text{VIN}} (\sigma_{\text{date\_of\_sale} > \text{'2021-03-21'}} (\text{sale}))$$

$$R3 \leftarrow \pi_{\text{VIN , dealership\_name}} (\text{car})$$

$$R4 \leftarrow \pi_{\text{model\_name,engine,fuel\_type}} (\text{model})$$

$$R5 \leftarrow \text{model\_name} (R1 \bowtie R4)$$

$$R6 \leftarrow \pi_{\text{VIN , dealership\_name ,date\_of\_sale}} (R5 \bowtie (R2 \bowtie R3))$$

We extracted from each table just the necessary attributes

3)

a) SQL Query :

```
with number_of_sales_of_model (model_name , number_of_sales, fuel_type) as
( SELECT model_name , count(*) as number_of_sales, fuel_type
FROM car
join model
USING (model_name)
```

```
join sale
USING (VIN)
group by model_name)
SELECT brand
FROM car_brand
join number_of_sales_of_model
using (model_name)
where fuel_type='electric' or fuel_type='hybrid'
and number_of_sales = (select max(number_of_sales) from number_of_sales_of_model);
```

b) Relational Algebra Formula :

$$R1 \leftarrow \pi_{\text{model\_name, number\_of\_sales, fuel\_type}} \left({}_{\text{model\_name}}\mathcal{G}_{\text{count(VIN) AS number\_of\_sales}} ((\text{car} \bowtie \text{model}) \bowtie \text{sale}) \right)$$

$$R2 \leftarrow \mathcal{G}_{\text{max(number\_of\_sales) AS max}} (R1)$$

$$R \leftarrow \pi_{\text{brand}} (\sigma_{\text{fuel\_type='electric'} \vee \text{fuel\_type='hybrid'}}) \wedge \text{number\_of\_sales} = \text{max} ((\text{car\_brand} \bowtie R1) \times R2))$$

c) we can optimize the query by :

$$R1 \leftarrow \pi_{\text{model\_name}} (\sigma_{\text{fuel\_type='electric'} \vee \text{fuel\_type='hybrid'}} (\text{model}))$$

$$R2 \leftarrow \mathcal{G}_{\text{max(number\_of\_sales) AS max}} (R1)$$

$$R3 \leftarrow \pi_{\text{VIN}} (\text{sale})$$

$$R4 \leftarrow \pi_{\text{VIN, model\_name}} (\text{car})$$

$$R5 \leftarrow \pi_{\text{model\_name, number\_of\_sales}} \left({}_{\text{model\_name}}\mathcal{G}_{\text{count(VIN) AS number\_of\_sales}} (R3 \bowtie R4) \right)$$

$R6 \leftarrow \pi_{brand}(\sigma_{number\_of\_sales = max}((R1 \times R2) \bowtie (R5 \bowtie car\_brand)))$

4)

a) SQL Query :

```
with number_of_repairs_of_model(model_name , number_of_repairs) as
( SELECT model_name , count(*) as number_of_repairs
FROM car
join maintenance_task
USING (VIN)
GROUP BY model_name )
SELECT model_name,brand,number_of_repairs
FROM number_of_repairs_of_model
JOIN car_brand
USING (model_name)
ORDER by number_of_repairs DESC
limit 10 ;
```

b) Relational Algebra Formula :

$R1 \leftarrow \pi_{model\_name , number\_of\_repairs}(_{model\_name}G_{count(VIN)\ \textbf{AS}\ number\_of\_repairs}(car \bowtie maintenance\_task))$

$R2 \leftarrow \pi_{model\_name , brand, number\_of\_repairs}(car\_brand \bowtie R1)$

$R3 \leftarrow \pi_{model\_name , brand, number\_of\_repairs}(R2) - \pi_{t1.model\_name , t1.brand, t1.number\_of\_repairs}(\sigma_{t1.\ number\_of\_repairs\ <\ t2.\ number\_of\_repairs}(\rho_{t1}(R2) \times \rho_{t2}(R2)))$

$$R4 \leftarrow \pi_{model\_name,\ brand,\ number\_of\_repairs}(R2) - \pi_{t1.model\_name,\ t1.brand,\ t1.number\_of\_repairs}(\sigma_{t1.\ number\_of\_repairs < t2.\ number\_of\_repairs}(\rho_{t1}(R2\text{-}R3) \times \rho_{t2}(R2\text{-}R3)))$$

.

.

We repeat this operation 10 times to extract the brands having the biggest number of reparation.

Finally:

$$Rfinal \leftarrow R3 \cup R4 \ldots\ldots \cup R12$$

c) optimisation :

$$R1 \leftarrow \pi_{VIN}(maintenace\_task)$$

$$R2 \leftarrow \pi_{VIN,\ model\_name}(car)$$

$$R3 \leftarrow \pi_{model\_name,\ number\_of\_repairs}(model\_name\ G\ count(VIN)\ \textbf{AS}\ number\_of\_repairs\ (R1 \bowtie R2))$$

$$R4 \leftarrow \pi_{model\_name,\ brand,\ number\_of\_repairs}(car\_brand \bowtie R3)$$

And then we subtitute R4 by R3 of the last question and we complete the classification procedure.

5)

a) SQL Query :

```
SELECT sum(extra_price) as money_earned_by_company_in_2020
FROM sale
where date_of_sale BETWEEN '2020-01-01' AND '2020-12-31' ;
```

b) Relational Algebra Formula :

$R \leftarrow G_{sum(extra\_price)} (\sigma_{date\_of\_sale > '2020\text{-}01\text{-}01' \wedge date\_of\_sale < '2020\text{-}12\text{-}31'} (sale))$

c) optimisation :

$R1 \leftarrow \pi_{date\_of\_sale, extra\_price} (sale)$

$R2 \leftarrow G_{sum(extra\_price)} (\sigma_{date\_of\_sale > '2020\text{-}01\text{-}01' \wedge date\_of\_sale < '2020\text{-}12\text{-}31'} (R1))$

6)

a) SQL Query :

```
with money_of_cars_sold(sales_person_idcardnr , price_of_all_cars) as
( SELECT sales_person_idcardnr, sum(base_price + extra_price) as price_of_all_cars
FROM sale
join car
USING (VIN)
join basic_characteristics
USING (model_name)
where date_of_sale BETWEEN '2020-01-01' AND '2020-12-31'
GROUP by sales_person_idcardnr )
SELECT name, sales_person_idcardnr , gross_annual_salary + 0.05*price_of_all_cars as
money_made_by_saleperson
FROM money_of_cars_sold, salesperson
WHERE money_of_cars_sold.sales_person_idcardnr = salesperson.IDcardNr ;
```

b) Relational Algebra Formula :

$R1 \leftarrow \pi$ sales_person_idcardnr, base_price, extra_price $((($sale $\bowtie$ car $) \bowtie$ basic_characteristics $)$

$R2 \leftarrow \pi$ sales_person_idcardnr, price_of_all_cars $($ salesperson_ID_card_Nr G sum(base_price + extra_price) AS price_of_all_cars $($ σ date_of_sale > '2020-01-01' $\wedge$ date_of_sale < '2020-12-31' $(R1)))$

$R \leftarrow \pi$ name, sales_person_idcardnr , gross_annual_salary + 0.05*price_of_all_cars **AS** money_made_by_saleperson $($ σ R2.sales_person_idcardnr = salesperson.IDcardNr $(R2 \times$ saleperson$))$

c) optimisation :

$R1 \leftarrow \pi$ sales_person_idcardnr, VIN,extra_price, $(\sigma$ date_of_sale > '2020-01-01' $\wedge$ date_of_sale < '2020-12-31' $($sale$))$

$R2 \leftarrow \pi$ VIN , model_name $($car$)$

$R3 \leftarrow \pi$ base_price , model_name $($basic_characteristics$)$

$R4 \leftarrow \pi$ VIN , base_price $(R2 \times R3)$

$R5 \leftarrow \pi$ sales_person_idcardnr, VIN,extra_price $(R4 \times R1)$

$R6 \leftarrow \pi$ sales_person_idcardnr, price_of_all_cars $($ salesperson_ID_card_Nr G sum(base_price + extra_price) **AS** price_of_all_cars $(R5)$

$R7 \leftarrow \pi$ name, IDcardNr, gross_annual_salary $($salesperson$)$

$R8 \leftarrow \pi$ name, sales_person_idcardnr , gross_annual_salary + 0.05*price_of_all_cars **AS** money_made_by_saleperson (σ R6.sales_person_idcardnr = R7.IDcardNr (R6 x R7))

7)

a) SQL Query :

```
SELECT name, surname
FROM salesperson , installement_financing
where salesperson.idcardnr = installement_financing.sales_person_idcardnr

EXCEPT

SELECT name , surname
FROM salesperson , cash
where salesperson.idcardnr = cash.sales_person_idcardnr;
```

b) Relational Algebra Formula :

$R1 \leftarrow \pi$ name, surname (σ salesperson.idcardnr = installement_financing.sales_person_idcardnr (salesperson x installement_financing))

$R2 \leftarrow \pi$ name, surname (σ salesperson.idcardnr = cash.sales.sales_person_idcardnr (salesperson x cash))

$R \leftarrow R1 - R2$

c) optimisation :

$R1 \leftarrow \pi$ name, surname, idcardnr (salesperson)

$R2 \leftarrow \pi_{\text{sales\_person\_idcardnr}} (\text{installement\_financing})$

$R3 \leftarrow \pi_{\text{name, surname}} (\sigma_{\text{R1.idcardnr = R2.sales\_person\_idcardnr}} (R1 \times R2))$

$R4 \leftarrow \pi_{\text{sales\_person\_idcardnr}} (\text{cash})$

$R5 \leftarrow \pi_{\text{name, surname}} (\sigma_{\text{salesperson.idcardnr = installement\_financing.sales\_person\_idcardnr}} (R1 \times R4))$

$R6 \leftarrow R3 - R5$

8)

a) SQL Query :

```
with cars_purchased_by_companies(VIN) as (
SELECT VIN
FROM sale, customer_company
where sale.customer_id = customer_company.commercial_identification_number)
SELECT name, surname,time_spent
FROM mechanic, repair_participation, cars_purchased_by_companies
where mechanic.idcardnr = repair_participation.mechanic_id_card
and cars_purchased_by_companies.VIN = repair_participation.VIN
ORDER BY time_spent ;
```

b) Relational Algebra Formula :

$R1 \leftarrow \pi_{\text{VIN}} (\sigma_{\text{sale.customer\_id = customer\_company.commercial\_identification\_number}} (\text{sale} \times \text{customer\_company}))$

$R \leftarrow \pi$ name, surname, time_spent (σ mechanic.idcardnr = repair_participation.mechanic_id_card $\wedge$ R1.VIN = repair_participation.VIN ((mechanic x repair_participation) x R1))

c) optimisation :

$R1 \leftarrow \pi$ VIN, customer_id (sale)

$R2 \leftarrow \pi$ commercial_identification_number (customer_company)

$R3 \leftarrow \pi$ VIN (σ R1.customer_id = R2.commercial_identification_number (R1 x R2))

$R4 \leftarrow \pi$ mechanic_id_card, time_spent, VIN (repair_participation)

$R5 \leftarrow \pi$ idcardnr, name, surname (mechanic)

$R6 \leftarrow \pi$ name, surname, time_spent, VIN (σ R5.idcardnr = R4.mechanic_id_card (R4 x R5))

$R7 \leftarrow \pi$ name, surname, time_spent (σ R6.VIN = R3.VIN (R4 x R5))

9)

a) SQL Query :

```
SELECT dealership_name, type_of_failure, count(*) as number_of_vehicles
from car
JOIN maintenance_task
USING (VIN)
where type_of_failure = 'Alternator'
```

group by (dealership_name)

ORDER by number_of_vehicles DESC

limit 1;

b) Relational Algebra Formula :

$$R1 \leftarrow \pi_{\text{dealership\_name, type\_of\_failure, number\_of\_vehicles}} ({}_{\text{dealership\_name}}\mathcal{G}_{\text{count(dealership\_name) AS number\_of\_vehicles}} (\sigma_{\text{type\_of\_failure='Alternator'}} (\text{car} \bowtie \text{maintenance\_task})))$$

$$R \leftarrow \pi_{\text{dealership\_name, type\_of\_failure, number\_of\_vehicles}} (R1) - \pi_{\text{t1.dealership\_name , t1.type\_of\_failure, t1. number\_of\_vehicles}} (\sigma_{\text{t1.number\_of\_vehicles < t2.number\_of\_vehicles}} (\rho_{t1}(R1) \times \rho_{t2}(R1)))$$

c) optimisation :

$$R1 \leftarrow \pi_{\text{type\_of\_failure,VIN}} (\sigma_{\text{type\_of\_failure = 'Alternator'}} (\text{maintenance\_task}))$$

$$R2 \leftarrow \pi_{\text{VIN, dealership\_name}} (\text{car})$$

$$R3 \leftarrow \pi_{\text{dealership\_name, type\_of\_failure, number\_of\_vehicles}} ({}_{\text{dealership\_name}}\mathcal{G}_{\text{count(dealership\_name) AS number\_of\_vehicle}} (R1 \times R2))$$

$$R4 \leftarrow \pi_{\text{dealership\_name, type\_of\_failure, number\_of\_vehicles}} (R3) - \pi_{\text{t1.dealership\_name , t1.type\_of\_failure, t1. number\_of\_vehicles}} (\sigma_{\text{t1.number\_of\_vehicles < t2.number\_of\_vehicles}} (\rho_{t1}(R3) \times \rho_{t2}(R3)))$$

10)

a) SQL Query :

```
with money_earned_by_each_saleperson (sales_person_idcardnr,
dealership_name,money_earned_by_saleperson) AS

( SELECT sales_person_idcardnr,salesperson.dealership_name, sum(base_price +
extra_price) as money_earned_by_saleperson

FROM sale ,salesperson

join car

USING (VIN)

join basic_characteristics

USING (model_name)

where sale.sales_person_idcardnr = salesperson.idcardnr

GROUP by sales_person_idcardnr )


SELECT name , surname, s.money_earned_by_saleperson,  adress,
salesperson.dealership_name

FROM salesperson, money_earned_by_each_saleperson s

where s.sales_person_idcardnr = salesperson.idcardnr

and money_earned_by_saleperson >


(SELECT avg(money_earned_by_saleperson)

FROM money_earned_by_each_saleperson f

WHERE f.dealership_name = s.dealership_name);
```

b) Relational Algebra Formula :

$R1 \leftarrow \pi$ sales_person_idcardnr, salesperson.dealership_name,

money_earned_by_saleperson $($ sales_person_idcardnr $\mathbf{G}$ sum(base_price + extra_price **AS** money_earned_by_saleperson

$((\sigma_{\text{sale.sales\_person\_idcardnr = salesperson.idcardnr}}(\text{sale} \times \text{saleperson})) \bowtie \text{car} \bowtie \text{basic\_characteristics})$

$R2 \leftarrow \pi_{\text{dealership\_name, average}}(\text{dealership\_name } G_{\text{AVG(money\_earned\_by\_saleperson)}}(R1))$

$R \leftarrow \pi_{\text{name , surname, money\_earned\_by\_saleperson,adress, salesperson.dealership\_name}}(\sigma_{\text{R2.dealership\_name = salesperson.dealership\_name} \wedge \text{R1.sales\_person\_idcardnr = salesperson.idcardnr} \wedge \text{money\_earned\_by\_saleperson} > \text{average}}((\text{saleperson} \times R1) \times R2))$

c) optimisation :

$R1 \leftarrow \pi_{\text{VIN, model\_name}}(\text{car})$

$R2 \leftarrow \pi_{\text{base\_price, model\_name}}(\text{basic\_characteristics})$

$R3 \leftarrow \pi_{\text{base\_price, VIN}}(\text{basic\_characteristics} \times \text{car})$

$R4 \leftarrow \pi_{\text{idcardnr, dealership\_name}}(\text{salesperson})$

$R5 \leftarrow \pi_{\text{extra\_price, VIN, sales\_person\_idcardnr}}(\text{sale})$

$R6 \leftarrow \pi_{\text{sales\_person\_idcardnr, dealership\_name, money\_earned\_by\_saleperson}}(\text{sales\_person\_idcardnr } G_{\text{sum(base\_price + extra\_price AS money\_earned\_by\_saleperson}}(\sigma_{\text{R5.sales\_person\_idcardnr = R4.idcardnr}}((R4 \times R5) \bowtie R3)))$

$$R7 \leftarrow \pi \text{ dealership\_name, average } (\text{dealership\_name } G \text{ AVG(money\_earned\_by\_saleperson)} (R5))$$

$$R8 \leftarrow \pi \text{ idcardnr, dealership\_name , name, surname, adress } (\text{salesperson})$$

$$R9 \leftarrow \text{name , surname, money\_earned\_by\_saleperson, adress, salesperson.dealership\_name}$$

$$(\sigma \text{ R7.dealership\_name = R8.dealership\_name } \wedge \text{ R6.sales\_person\_idcardnr = R8.idcardnr) } \wedge \text{ money\_earned\_by\_saleperso > average } ((R8x\ R6)\ x\ R7))$$