Projet BDA

**SQL3-Oracle et NoSQL (MongoDB)**

M1 IL 2023/2024

HIOUANI Lydia 202031060760

OULD SLIMANE Neila 20209DZA6946

**Partie I : Relationnel-Objet**

**A- Modélisation orientée objet**

**1.** Transformez ce schéma relationnel en un schéma Objet (diagramme de classes)

**B- Création des TableSpaces et utilisateur**

**2.** Créer deux TableSpaces *SQL3\_TBS et SQL3\_TempTBS*

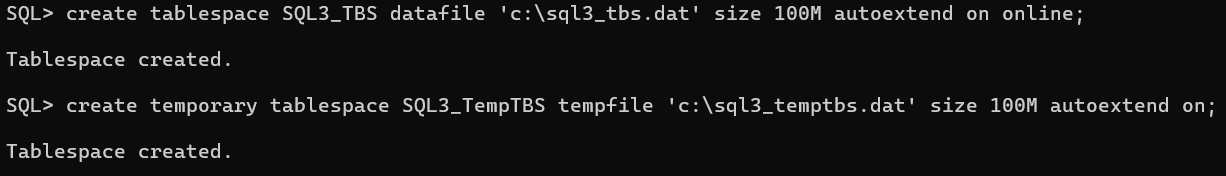
**- Requêtes SQL**

create tablespace SQL3\_TBS datafile 'c:\sql3\_tbs.dat' size 100M autoextend on online;

create temporary tablespace SQL3\_TempTBS tempfile 'c:\sql3\_temptbs.dat' size 100M autoextend on;

**- Explication**

**- Captures**



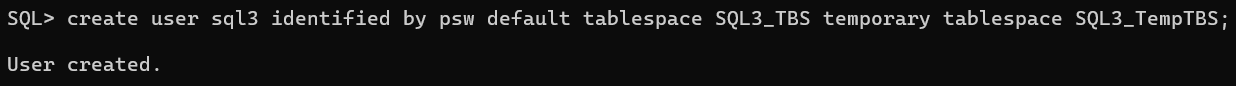
**3.** Créer un utilisateur SQL3 en lui attribuant les deux tablespaces créés précédemment

**- Requêtes SQL**

create user sql3 identified by psw default tablespace SQL3\_TBS temporary tablespace SQL3\_TempTBS;

**- Explication**

**- Captures**



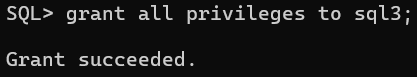
**4.** Donner tous les privilèges à cet utilisateur.

**- Requêtes SQL**

grant all privileges to sql3;

**- Explication**

**- Captures**



**C- Langage de définition de données**

**5.** En se basant sur le diagramme de classes fait, définir tous les types nécessaires. Prendre en compte toutes les associations qui existent.

**- Requêtes SQL**

**Pour les types :**  
create or replace type <nomType> as object(

<att1> <type1>,

…

<attn> <typen>,

<member/static> function …,

…

<member/static> function …

) ;/

**Pour les tables imbriquées :**

create type nomTable as table of ref nomType;

/

**- Explication**

**Cas association à classe :**

Transformer en 2 associations (comme l’association INTERVENANTS)

**Cas association 1..\* :**

Soit classe 1 qui possède plusieurs classe 2 et classe 2 appartient à 1 seule classe 1

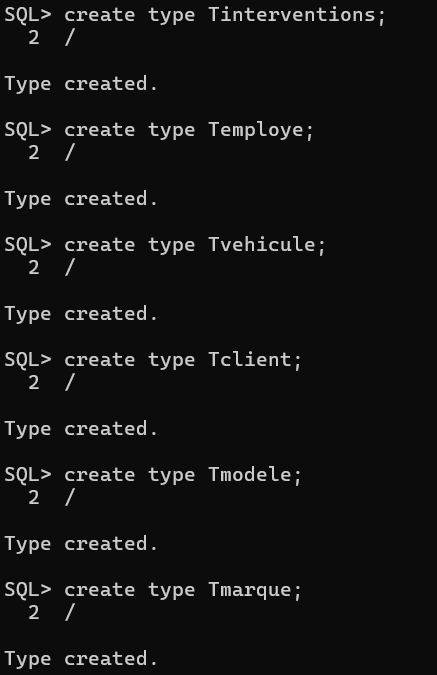
On ajoute un attribut dans classe 1 qui est une table de références ver classe2

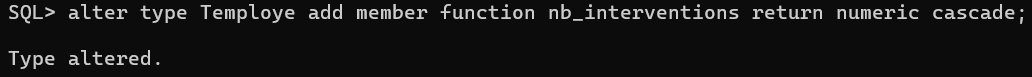
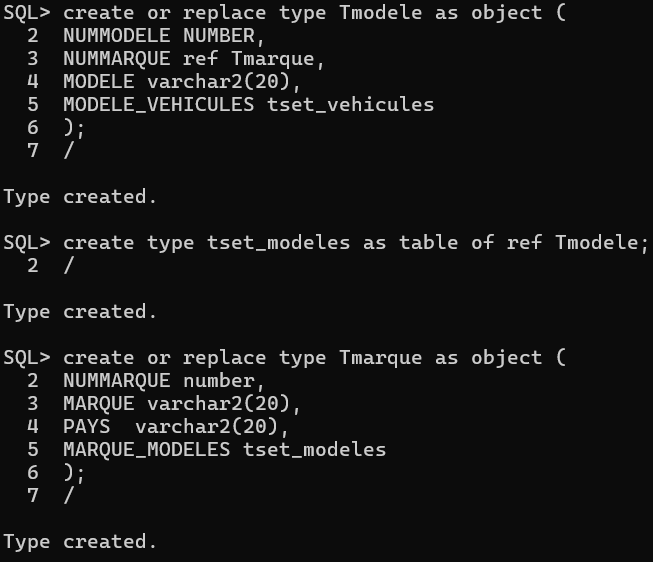
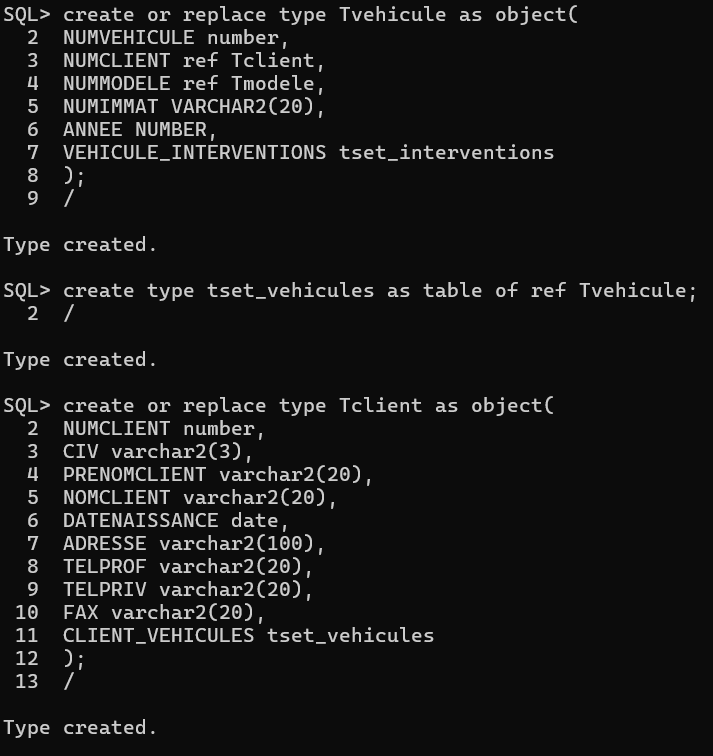
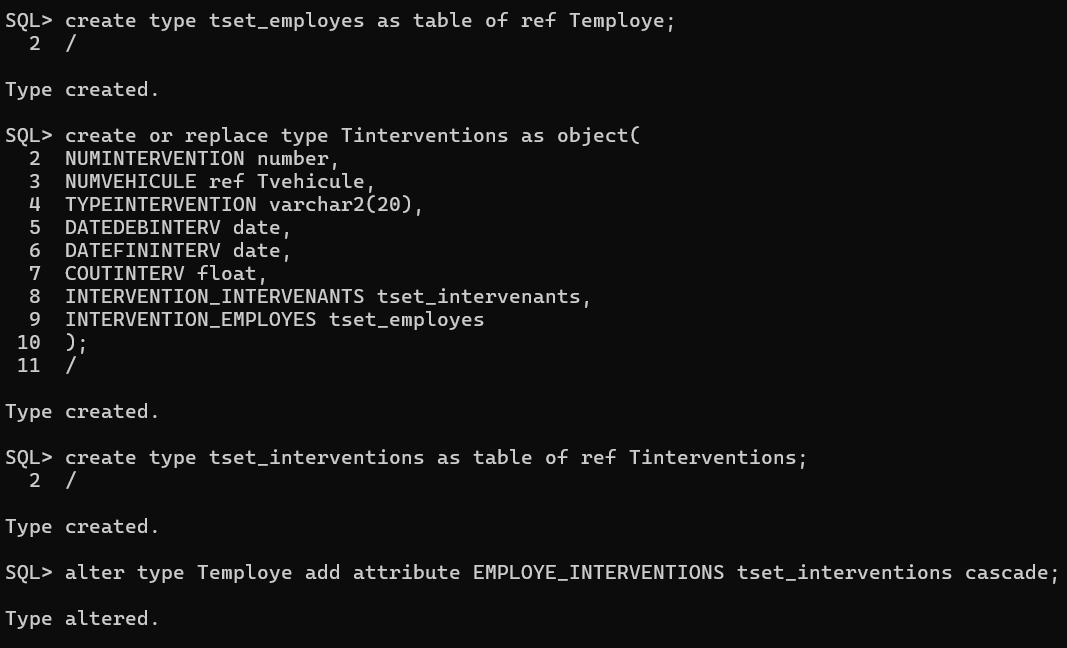
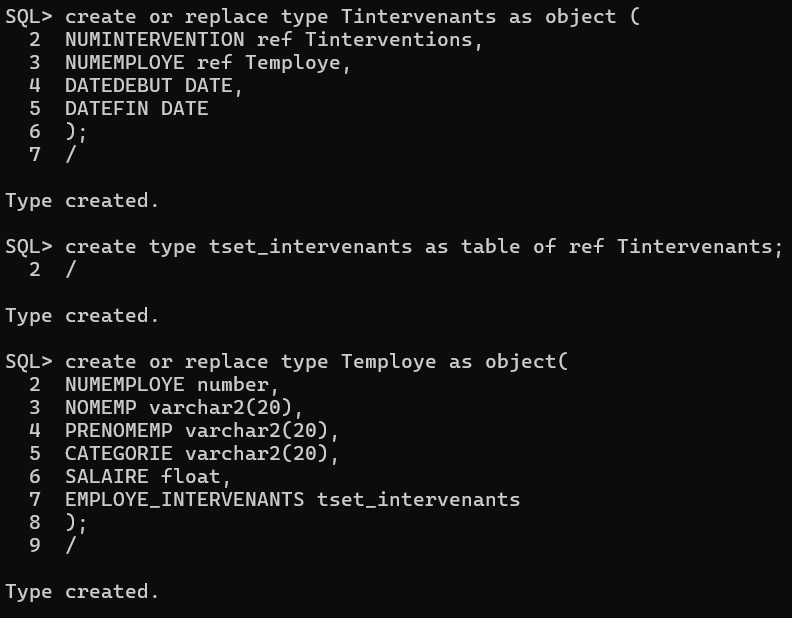
On ajoute un attribut dans classe 2 sui sauvegarde la référence de classe 1

**Cas association \*..\* :**

Chaque classe aura une table de références de l’autre classe

**- Captures**





**6.** Définir les méthodes permettant de :

**-** Calculer pour chaque employé, le nombre des interventions effectuées.

**- Requêtes SQL**

CREATE OR REPLACE TYPE BODY Temploye AS

MEMBER FUNCTION nb\_interventions RETURN NUMBER IS

BEGIN

RETURN SELF.EMPLOYE\_INTERVENTIONS.COUNT;

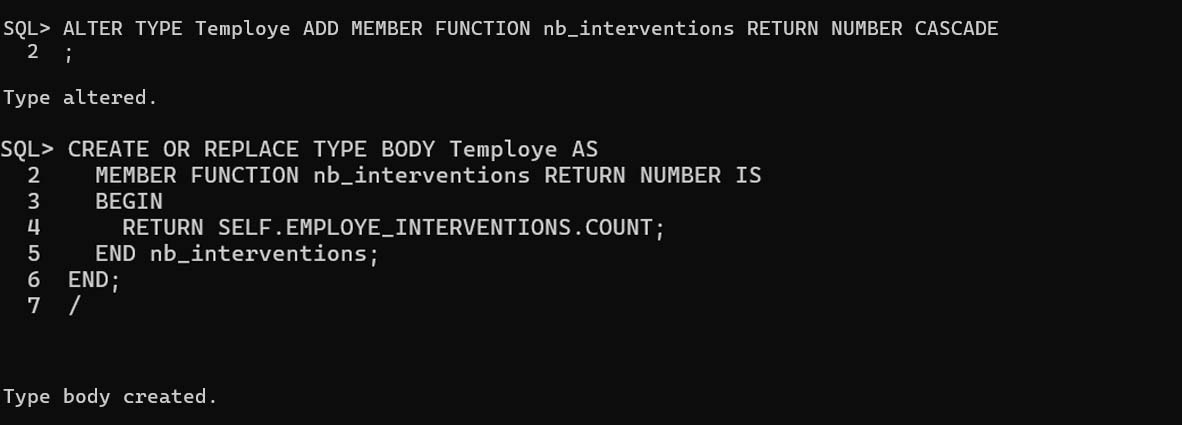
END nb\_interventions;

END;

/

**- Explication**

**La requete utilisée dans nb\_interventions accède à la table imbriquée EMPLOYE\_INTERVENTIONS de l’objet Employe courant « self » de l’employé et calcule le nombre de lignes de cette table qui est équivalent à le nombre d’interventions de l’employé**



**-** Calculer pour chaque marque, le nombre de modèles.

alter type Tmarque add member function nb\_modeles return number cascade;

CREATE OR REPLACE TYPE BODY Tmarque AS

MEMBER FUNCTION nb\_modeles RETURN NUMBER IS

BEGIN

RETURN self.MARQUE\_MODELES.COUNT;

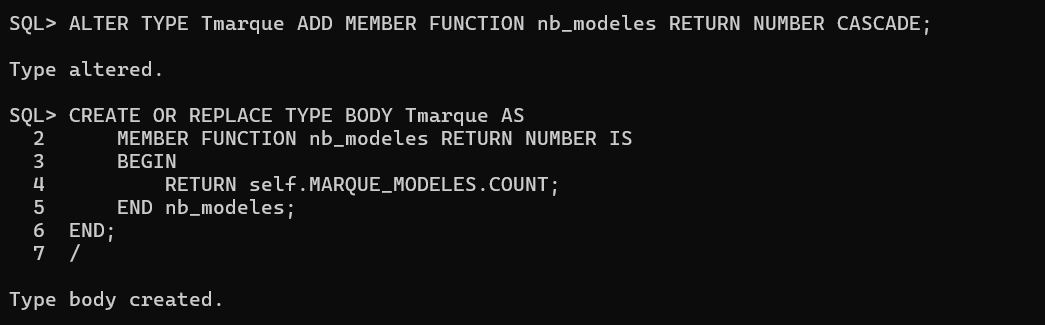
END nb\_modeles;

END;

/

**- Explication**

**La requête utilisée dans nb\_modeles accède à la table imbriquée MARQUE\_MODELES de l’objet Marque courant « self » du Marque et calcule le nombre de lignes de cette table qui est équivalent à le nombre des modèles de la marque**



**-** Calculer pour chaque modèle, le nombre de véhicules.

alter type Tmodele add member function nb\_vehicules return number cascade;

CREATE OR REPLACE TYPE BODY Tmodele AS

MEMBER FUNCTION nb\_vehicules RETURN NUMBER IS

BEGIN

RETURN self.MODELE\_VEHICULES.COUNT;

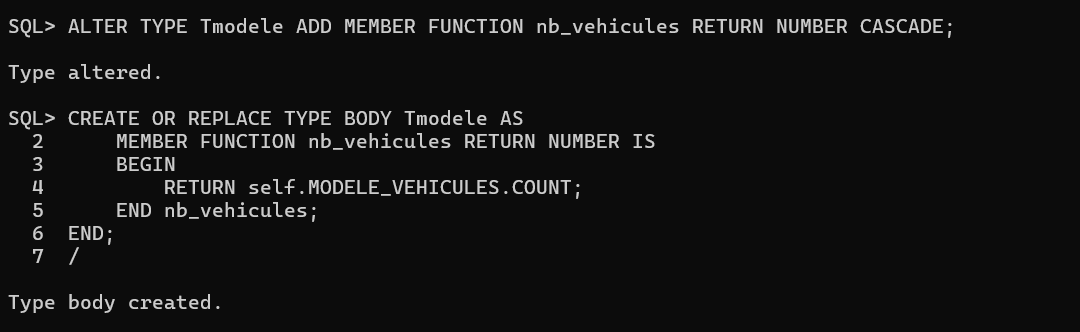
END nb\_vehicules;

END;

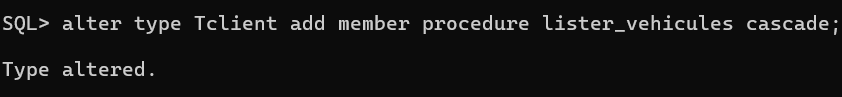
/

**- Explication**

**La requête utilisée dans nb\_ vehicules accède à la table imbriquée MODELE\_VEHICULES de l’objet Modèle courant « self » du Modèle et calcule le nombre de lignes de cette table qui est équivalent à le nombre des véhicules du modèle**



**-** Lister pour chaque client, ses véhicules.



**-** Calculer pour chaque marque, son chiffre d’affaire.

**7.** Définir les tables nécessaires à la base de données.

**- Requêtes SQL**

Create table nomTable of nomType(…);

**Si il y a une table imbriquée alors :**  
Create table nomTable of nomType(…) nested table nomNTable store as nomStore;

**- Explication**

**Les tables sont créées à partir des types définis précédemment.**

**Ajouter les contraintes.**

**Les tables imbriquées sont stockées dans un store**

**(meme chose pour toutes les créations de tables)**

**Création de la table pour le type Tmarque**

CREATE TABLE Marques OF Tmarque (

constraint pk\_marques primary key(NUMMARQUE)

) NESTED TABLE MARQUE\_MODELES STORE AS marque\_modeles\_table;

**Création de la table pour le type Tmodele**

CREATE TABLE Modeles OF Tmodele (

constraint pk\_modeles primary key(NUMMODELE)

) NESTED TABLE MODELE\_VEHICULES STORE AS modele\_vehicules\_table;

**Création de la table pour le type Tclient**

CREATE TABLE Clients OF Tclient (

constraint pk\_clients primary key(NUMCLIENT)

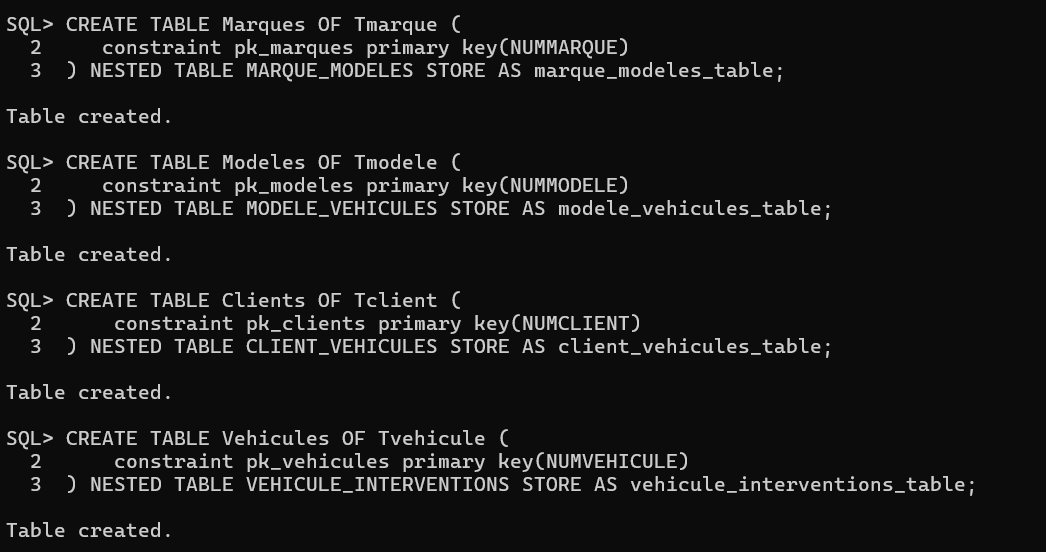
) NESTED TABLE CLIENT\_VEHICULES STORE AS client\_vehicules\_table;

**Création de la table pour le type Tvehicule**

CREATE TABLE Vehicules OF Tvehicule (

constraint pk\_vehicules primary key(NUMVEHICULE)

) NESTED TABLE VEHICULE\_INTERVENTIONS STORE AS vehicule\_interventions\_table;



**Création de la table pour le type Temploye**

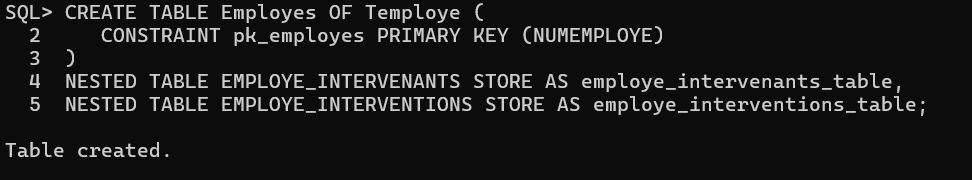
CREATE TABLE Employes OF Temploye (

CONSTRAINT pk\_employes PRIMARY KEY (NUMEMPLOYE)

)

NESTED TABLE EMPLOYE\_INTERVENANTS STORE AS employe\_intervenants\_table,

NESTED TABLE EMPLOYE\_INTERVENTIONS STORE AS employe\_interventions\_table;



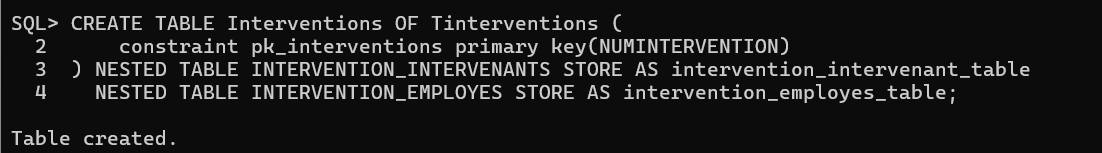
**Création de la table pour le type Tinterventions**

CREATE TABLE Interventions OF Tinterventions (

constraint pk\_interventions primary key(NUMINTERVENTION)

) NESTED TABLE INTERVENTION\_INTERVENANTS STORE AS intervention\_intervenant\_table

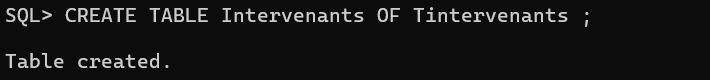
NESTED TABLE INTERVENTION\_EMPLOYES STORE AS intervention\_employes\_table;



**Création de la table pour le type Tintervenants**

CREATE TABLE Intervenants OF Tintervenants (

)



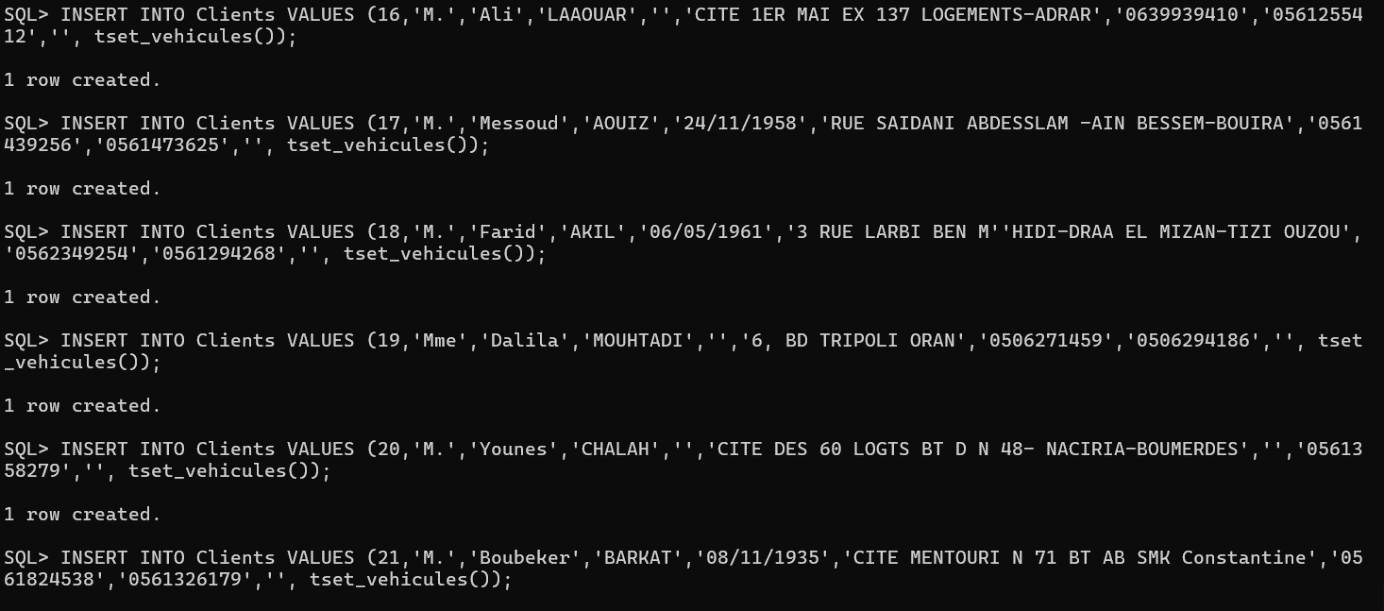
**D- Langage de manipulation de données**

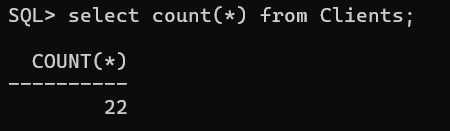
**8.** Remplir toutes les tables par les instances fournies en annexe.

**- Requêtes SQL + explication**

**Sur table Clients :**

INSERT INTO Clients VALUES (1,'Mme','Cherifa','MAHBOUBA','08/08/1957','CITE 1013 LOGTS BT 61 Alger','0561381813','0562458714','', tset\_vehicules());

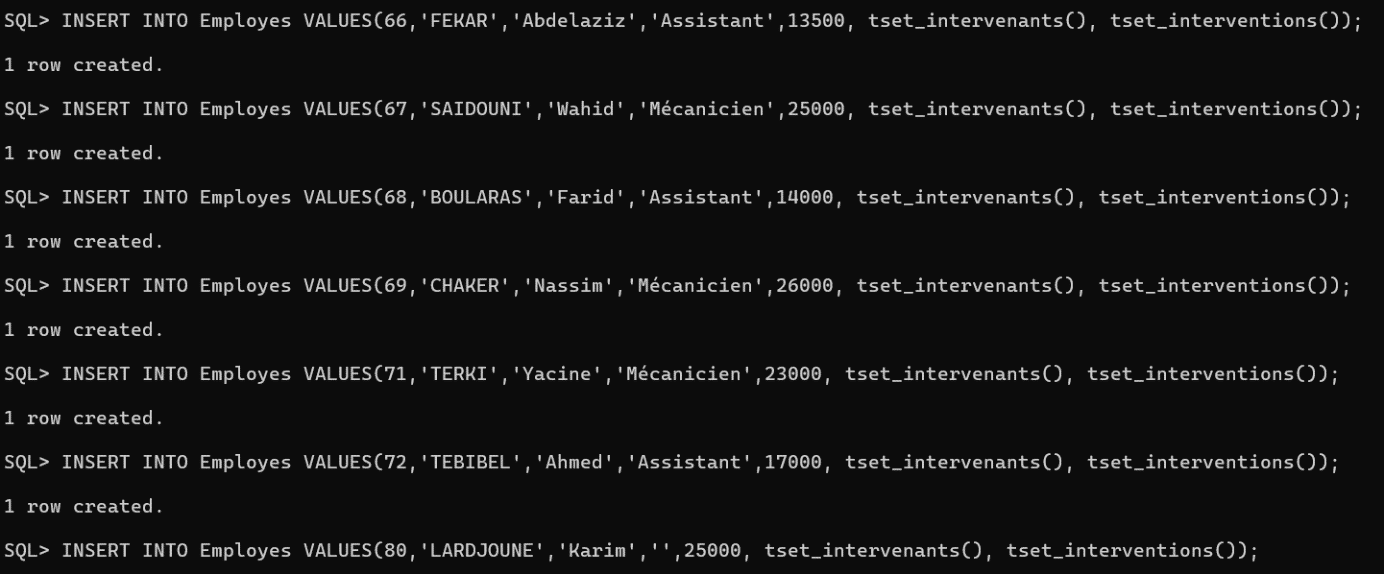


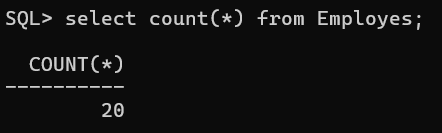


**Remplir les valeurs des attributs de la table + utiliser un constructeur pour la table + utiliser select ref() pour les références des objets imbriquée (même chose pour toutes les tables)**

**Sur table Employes:**

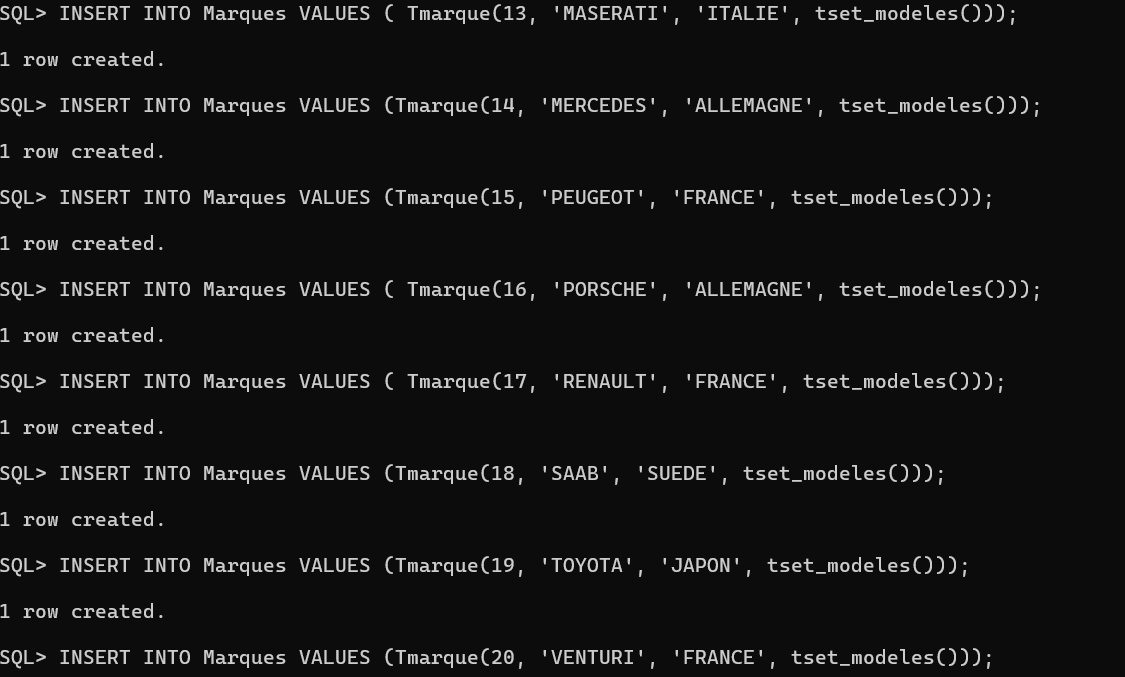
INSERT INTO Employes VALUES(53,'LACHEMI','Bouzid','Mécanicien',25000, tset\_intervenants(), tset\_interventions());

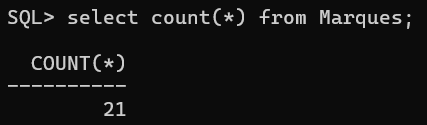




**Sur table Marques :**

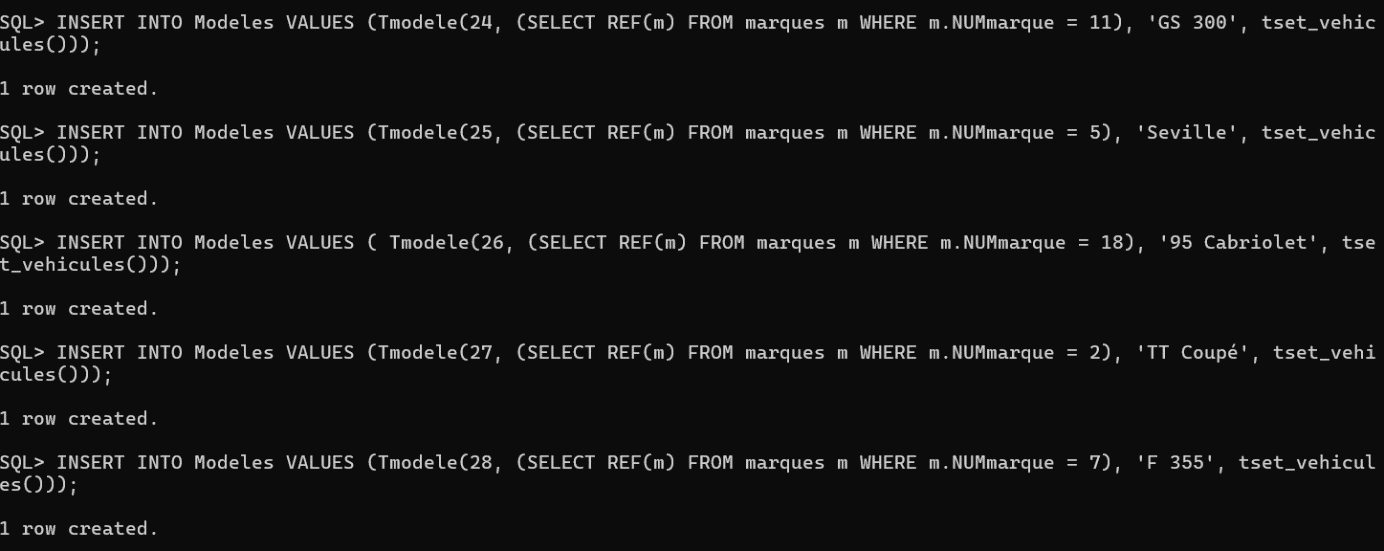
INSERT INTO Marques VALUES (Tmarque(1, 'LAMBORGHINI', 'ITALIE', tset\_modeles()));

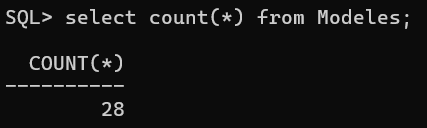




**Sur table Modeles :**

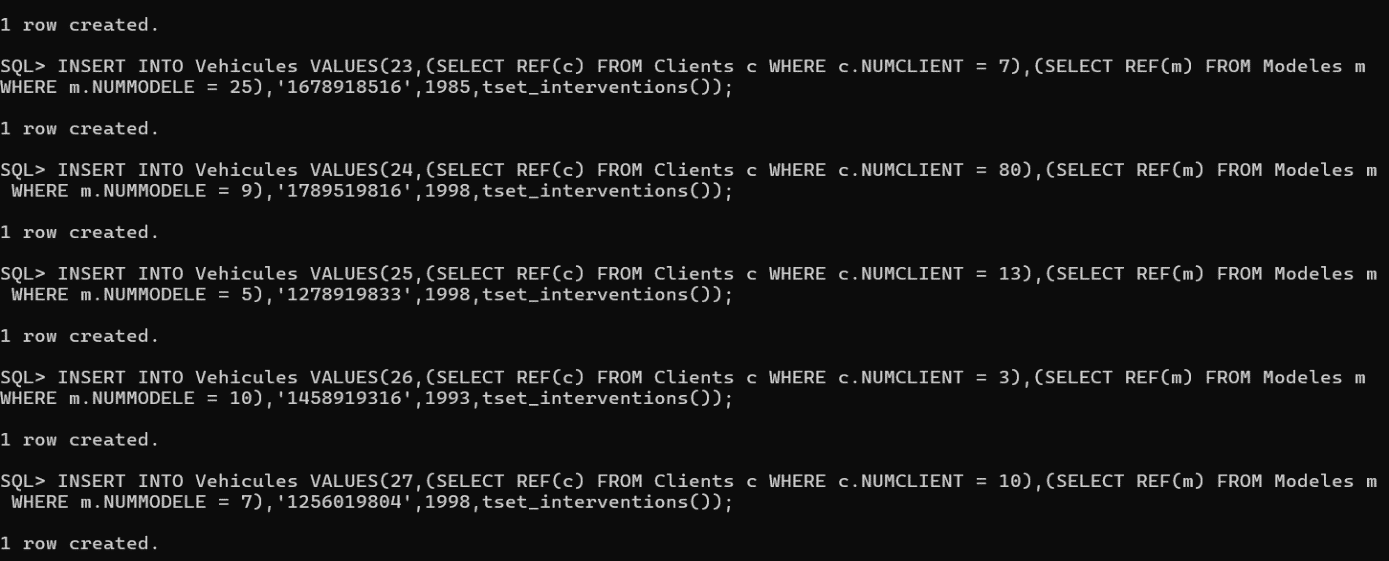
INSERT INTO Modeles VALUES ( Tmodele(2, (SELECT REF(m) FROM Marques m WHERE m.NUMMARQUE = 1), 'Diablo', tset\_vehicule()));

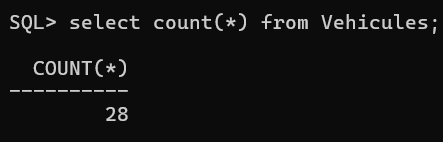




**Sur table Vehicules:**

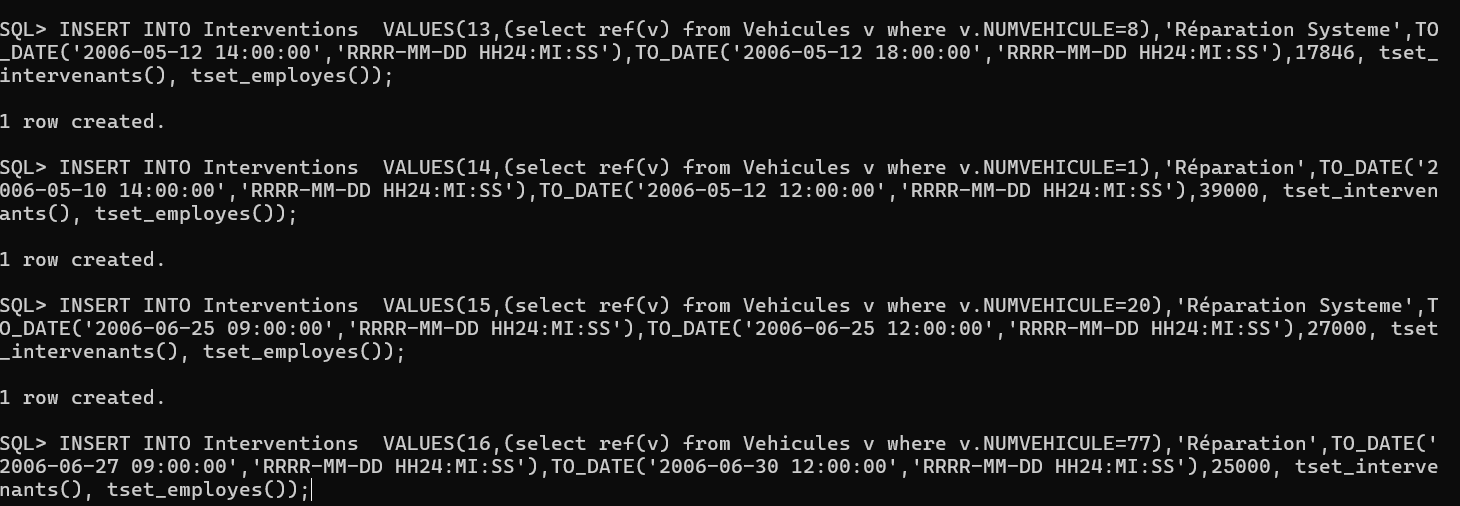
INSERT INTO Vehicules VALUES(1,(SELECT REF(c) FROM Clients c WHERE c.NUMCLIENT = 2),(SELECT REF(m) FROM Modeles m WHERE m.NUMMODELE = 6),'0012519216',1992,tset\_interventions());

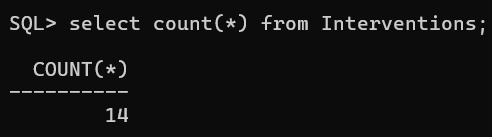




**Sur table Interventions:**

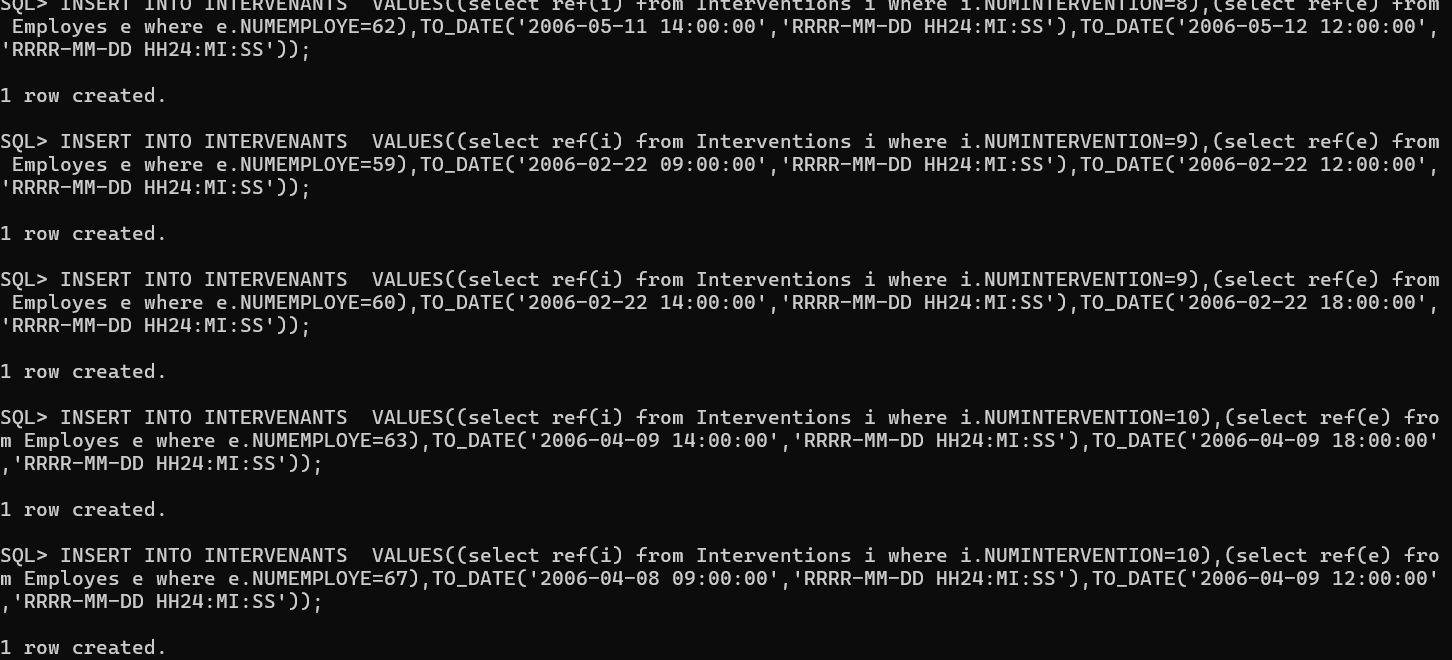
INSERT INTO Interventions VALUES(1,(select ref(v) from Vehicules v where v.NUMVEHICULE=3),'Réparation',TO\_DATE('2006-02-25 09:00:00','RRRR-MM-DD HH24:MI:SS'),TO\_DATE('2006-02-26 12:00:00','RRRR-MM-DD HH24:MI:SS'),30000, tset\_intervenants(), tset\_employes());

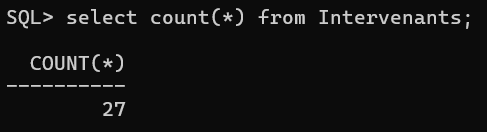




**Sur la table Intervenants :**

INSERT INTO INTERVENANTS VALUES((select ref(i) from Interventions i where i.NUMINTERVENTION=1),(select ref(e) from Employes e where e.NUMEMPLOYE=54),To\_DATE('2006-02-26 09:00:00','RRRR-MM-DD HH24:MI:SS'),TO\_DATE('2006-02-26 12:00:00','RRRR-MM-DD HH24:MI:SS'));



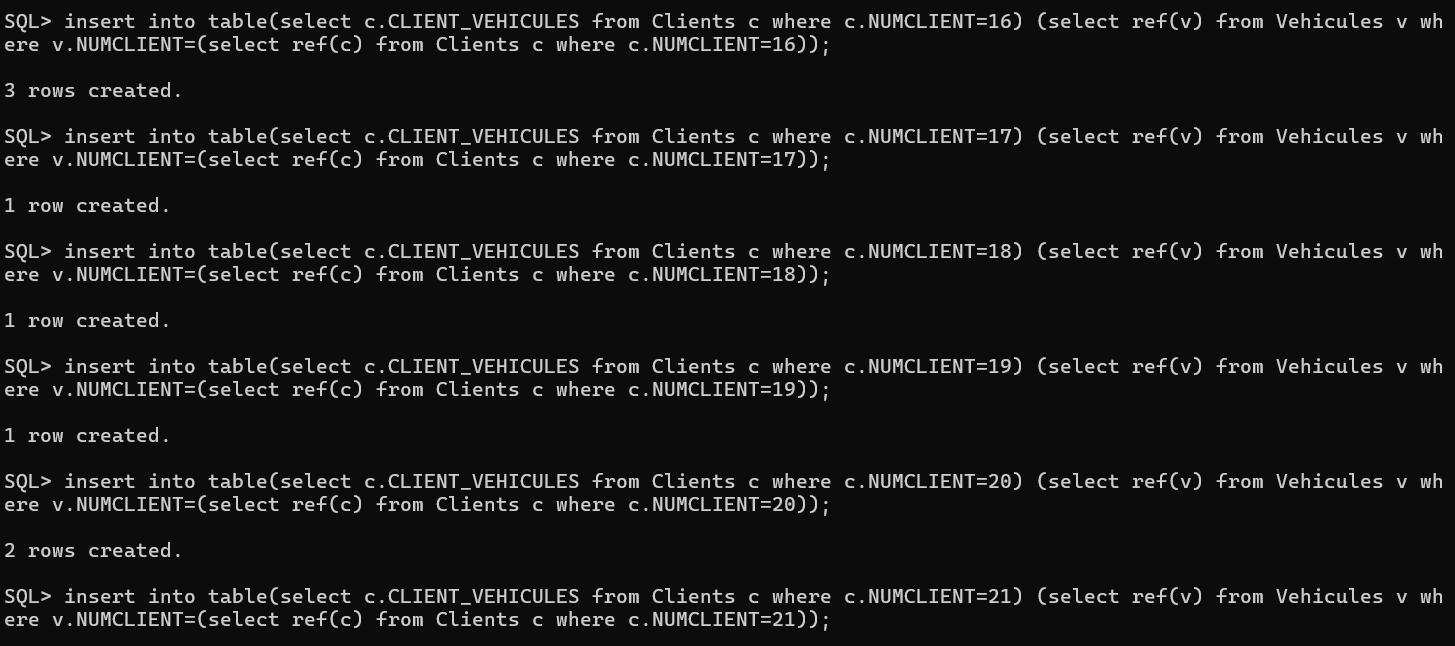


**Sur la table imbriquée CLIENT\_VEHICULES :**

insert into table(select c.CLIENT\_VEHICULES from Clients c where c.NUMCLIENT=1) (select ref(v) from Vehicules v where v.NUMCLIENT=(select ref(c) from Clients c where c.NUMCLIENT=1));

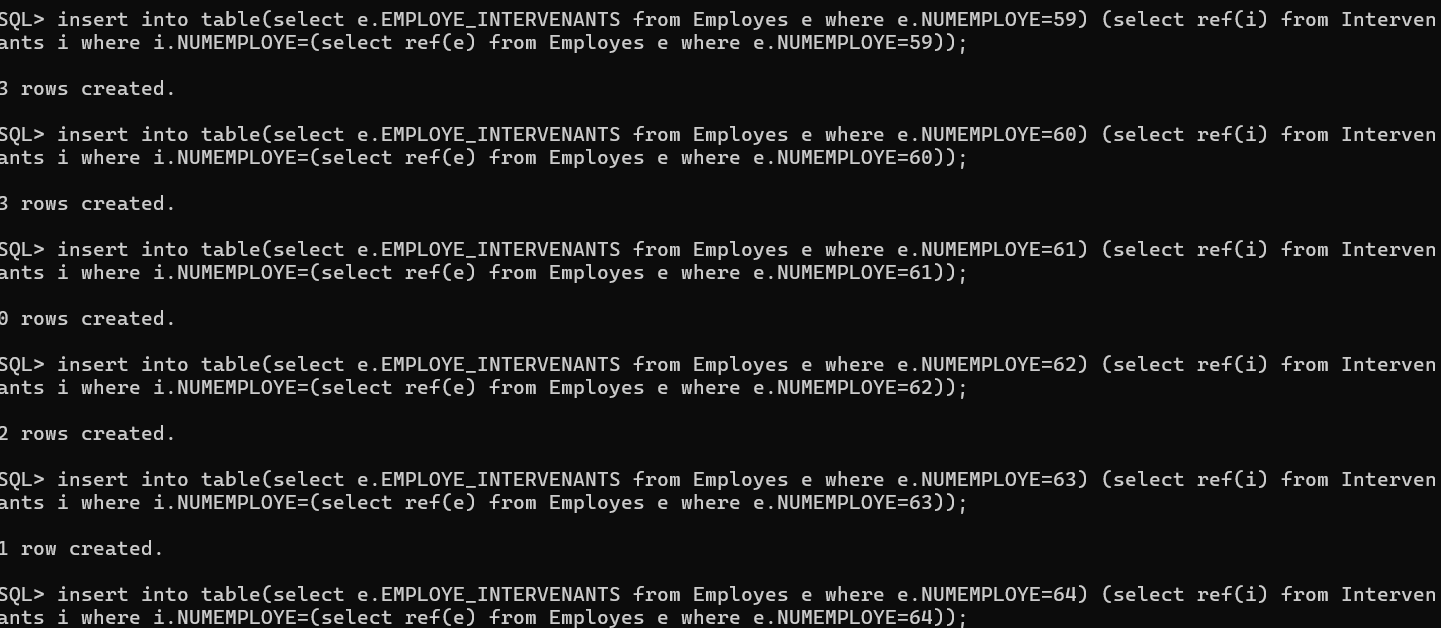
**on séléctionne la table imbriquée associée au client trouvée dans la table Clients et insérer la liste des références des véhicules apparetennants au client en utilisant la référence de l’objet client sauvegardée comme attribut dans la table Vehicules**

**(même idée s’applique pour le autres tables imbriquées sauf EMPLOYE\_INTERVENTIONS et INTERVENTION\_EMPLOYES)**

****

**sur la table imbriquée EMPLOYE\_INTERVENANTS :**

insert into table(select e.EMPLOYE\_INTERVENANTS from Employes e where e.NUMEMPLOYE=53) (select ref(i) from Intervenants i where i.NUMEMPLOYE=(select ref(e) from Employes where e.NUMEMPLOYE=53));

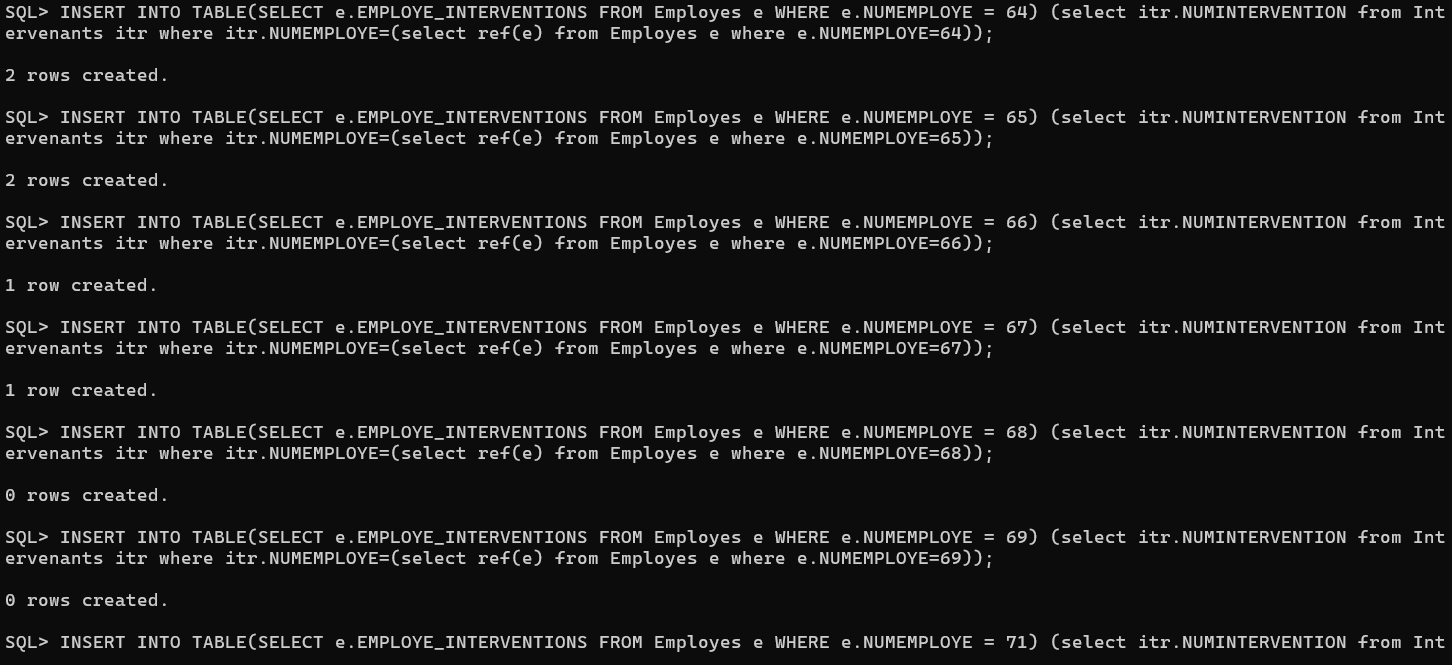


**sur la table imbriquée EMPLOYE\_INTERVENTIONS :**

INSERT INTO TABLE(SELECT e.EMPLOYE\_INTERVENTIONS FROM Employes e WHERE e.NUMEMPLOYE = 53) (select itr.NUMINTERVENTION from Intervenants itr where itr.NUMEMPLOYE=(select ref(e) from Employes e where e.NUMEMPLOYE=53));

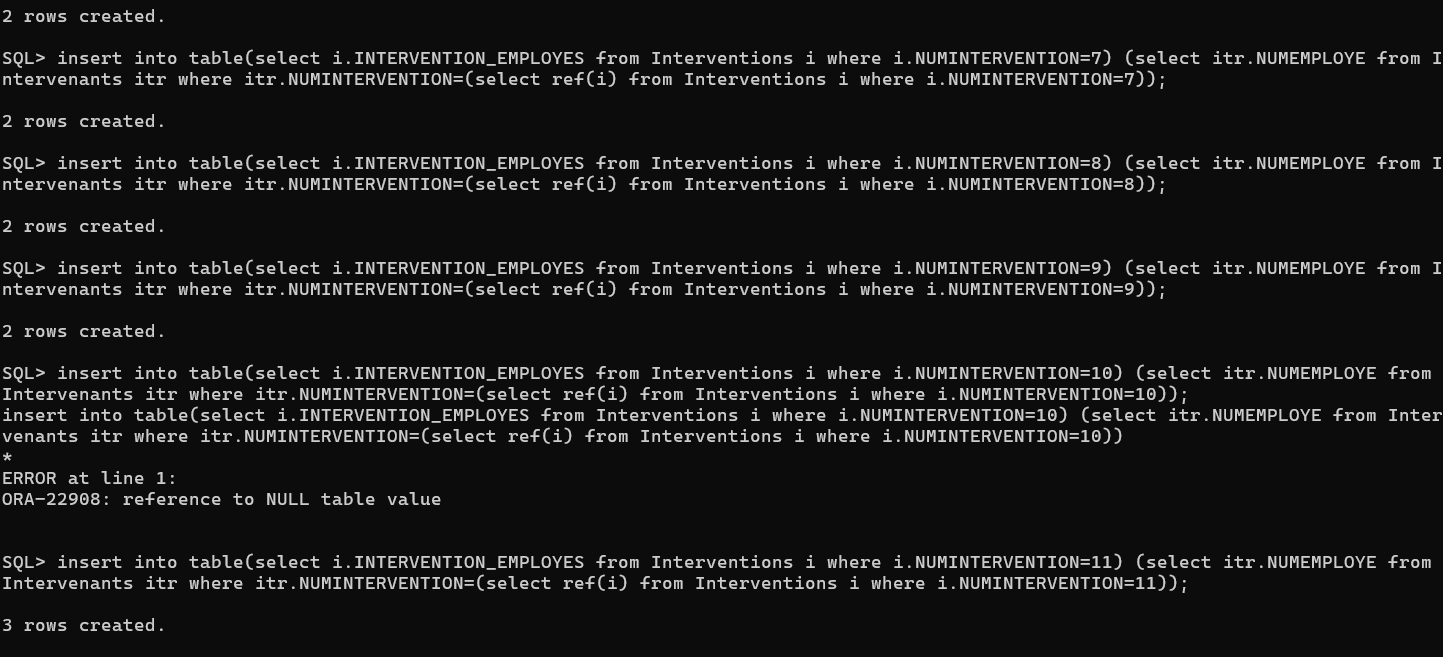
**Ici, les références des interventions associés à l’employé sont insérés à partir de la table Intervenants qui contient toutes les interventions et leur employés associés, donc il suffit juste de retrouvée la référence de l’employé dans la table Intervenants pour trouver toutes les interventions dont il a participé**

**(on utilise le même principe pour la table imbriquée INTERVENTION\_EMPLOYES)**

****

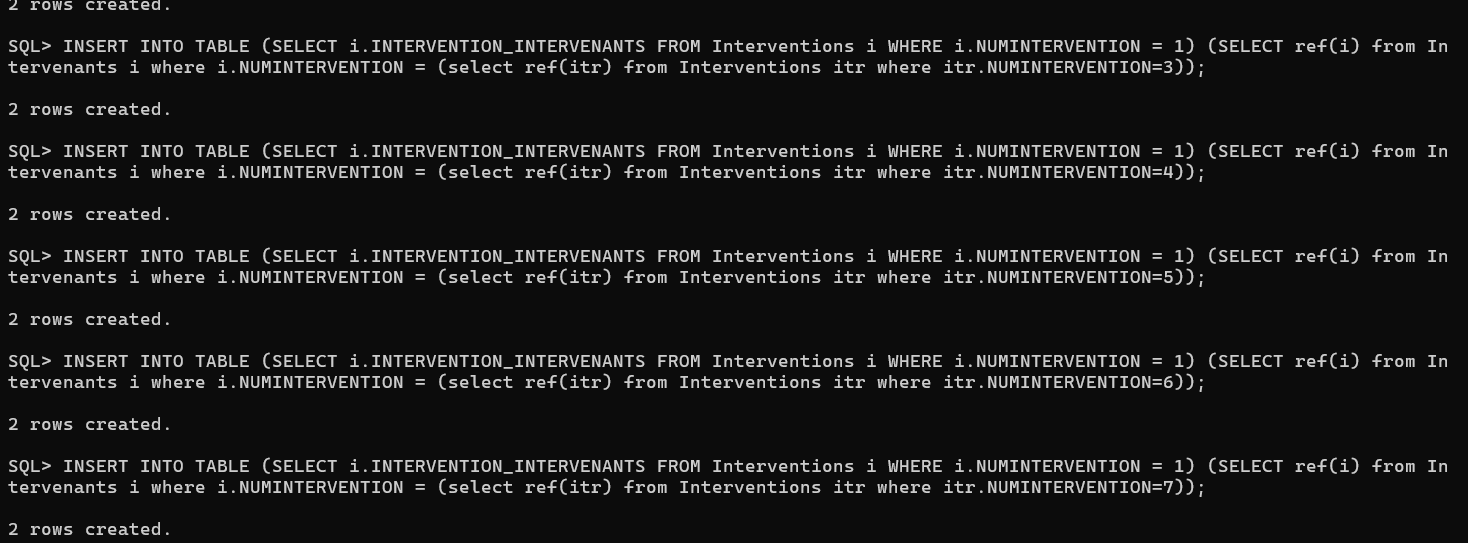
**sur la table imbriquée INTERVENTION\_EMPLOYES :**

insert into table(select i.INTERVENTION\_EMPLOYES from Interventions i where i.NUMINTERVENTION=1) (select itr.NUMEMPLOYE from Intervenants itr where itr.NUMINTERVENTION=(select ref(i) from Interventions i where i.NUMINTERVENTION=1));



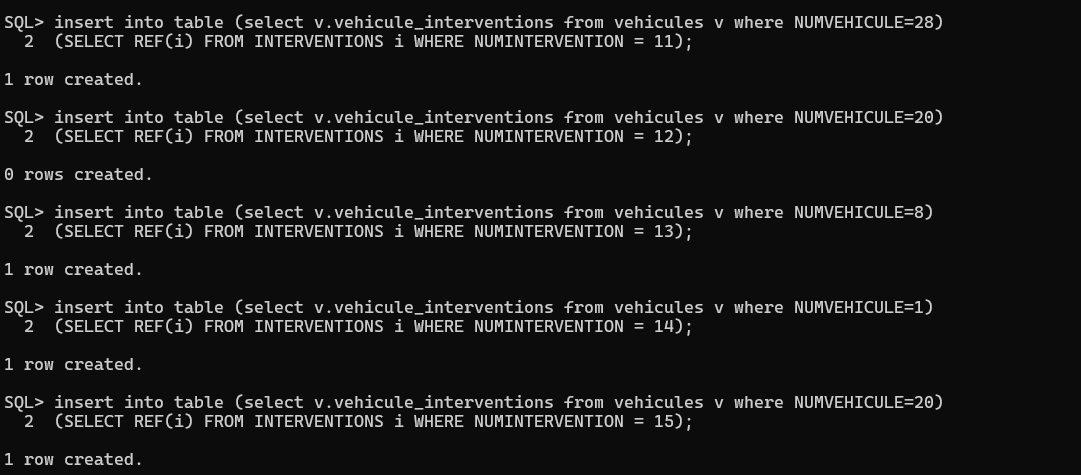
**sur la table imbriquée INTERVENTION\_INTERVENANTS :**

INSERT INTO TABLE (SELECT i.INTERVENTION\_INTERVENANTS FROM Interventions i WHERE i.NUMINTERVENTION = 1) (SELECT ref(i) from Intervenants i where i.NUMINTERVENTION = (select ref(itr) from Interventions itr where itr.NUMINTERVENTION=1));

****

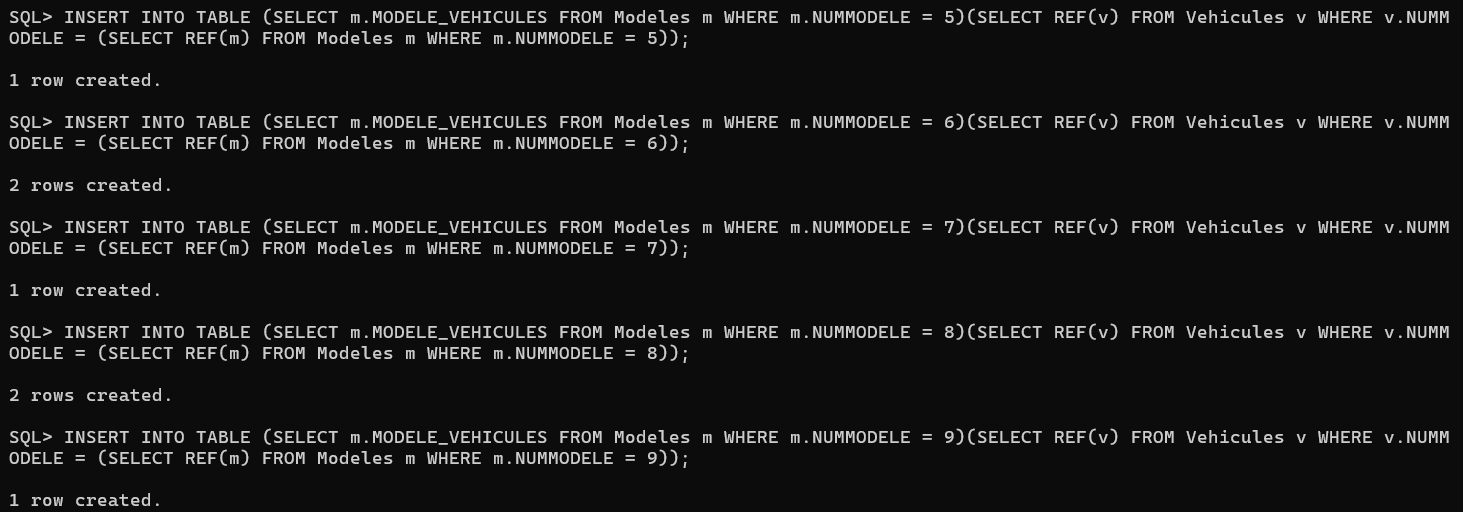
**sur la table VEHICULE\_INTERVENTIONS :**

INSERT INTO TABLE (SELECT v.VEHICULE\_INTERVENTIONS FROM Vehicules v WHERE v.NUMVEHICULE = 1) (SELECT REF(i) FROM Interventions i WHERE i.NUMVEHICULE = (SELECT REF(v) FROM Vehicules v WHERE v.NUMVEHICULE = 1));

****

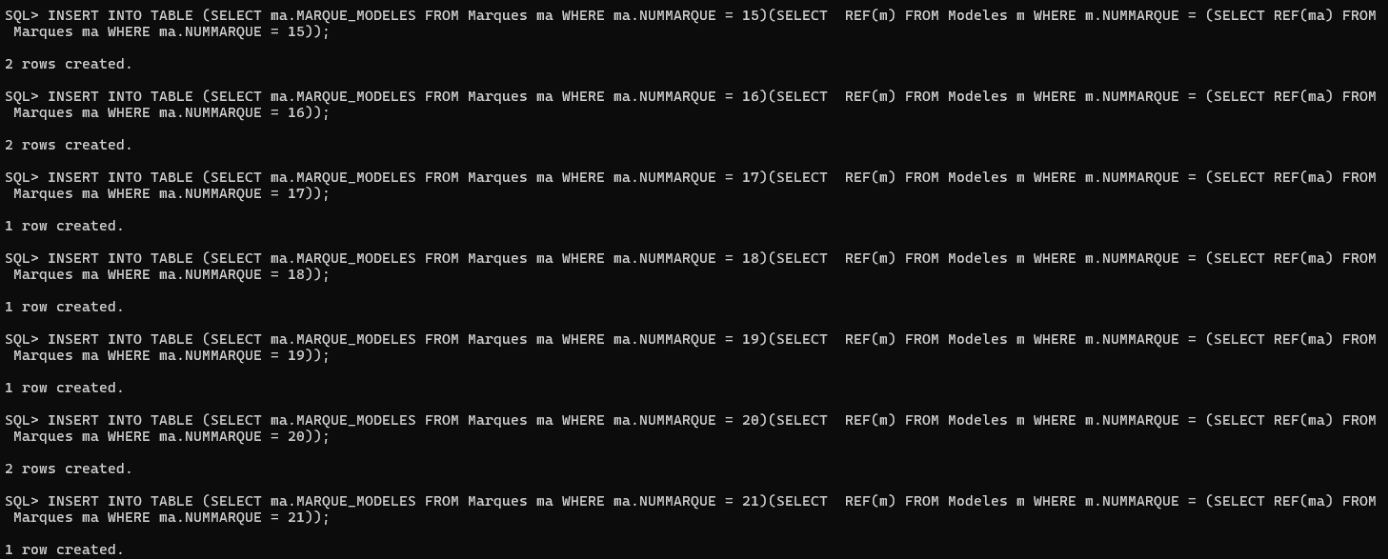
**sur la table MODELE\_VEHICULES :**

INSERT INTO TABLE (SELECT m.MODELE\_VEHICULES FROM Modeles m WHERE m.NUMMODELE = 2)(SELECT REF(v) FROM Vehicules v WHERE v.NUMMODELE = (SELECT REF(m) FROM Modeles m WHERE m.NUMMODELE = 2));

****

**sur la table MARQUE\_MODELES :**

INSERT INTO TABLE (SELECT ma.MARQUE\_MODELES FROM Marques ma WHERE ma.NUMMARQUE = 1)(SELECT REF(m) FROM Modeles m WHERE m.NUMMARQUE = (SELECT REF(ma) FROM Marques ma WHERE ma.NUMMARQUE = 1));

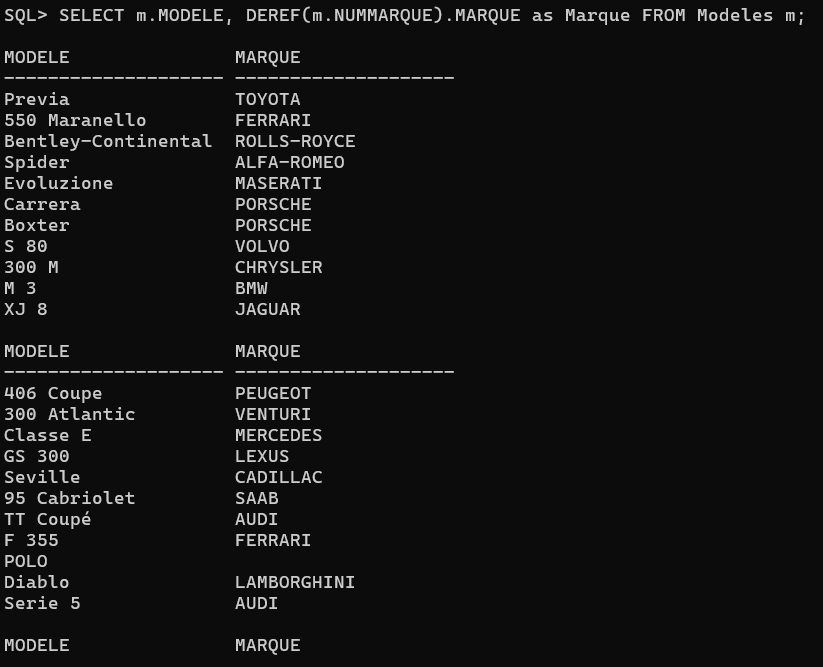


**E- Langage d’interrogation de données**

**9. Lister les modèles et leur marque.**

SELECT m.MODELE, DEREF(m.NUMMARQUE).MARQUE as Marque FROM Modeles m;

**On trouve l’objet Marque associé au modèle à partir de sa référence stockée dans l’attribut NUMMARQUE dans la table Modèle en utilisant DEREF**



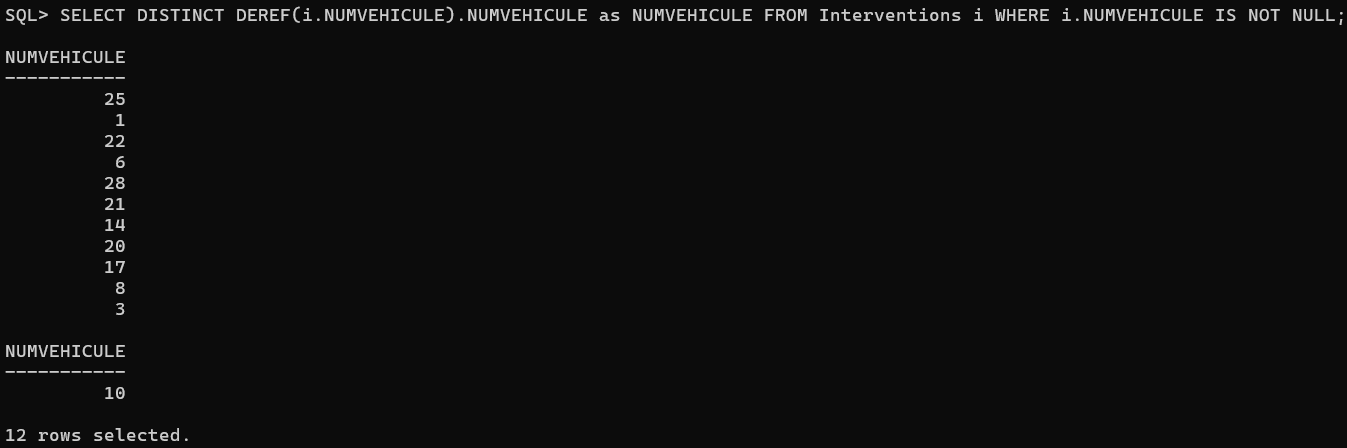
**10. Lister les véhicules sur lesquels, il y a au moins une intervention.**

SELECT DISTINCT DEREF(i.NUMVEHICULE).NUMVEHICULE as NUMVEHICULE FROM Interventions i WHERE i.NUMVEHICULE IS NOT NULL;

**On trouve l’objet Vehicule associé à l’intervention à partir de sa référence stockée dans l’attribut NUMVEHICULE dans la table Modèle en utilisant DEREF**

**On récupère le véhicule de chaque intervention**

**On utilise distinct pour éliminer la répétition dans le cas où un véhicule est passé par plusieurs interventions**

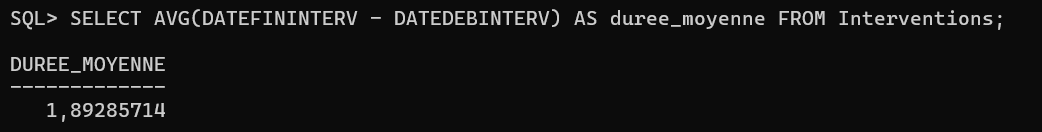


**11. Quelle est la durée moyenne d’une intervention?**

SELECT AVG(DATEFININTERV - DATEDEBINTERV) AS duree\_moyenne FROM Interventions;

**On calcule la durée de chaque intervention (date fin – date début)**

**Et on calcule la moyennes de ces durée en utilisant AVG()**

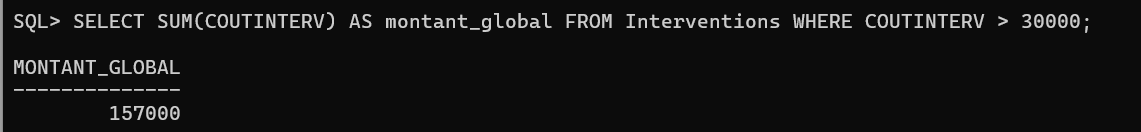


**12. Donner le montant global des interventions dont le coût d’intervention est supérieur à 30000 DA?**

SELECT SUM(COUTINTERV) AS montant\_global FROM Interventions WHERE COUTINTERV > 30000;

**On séléctionne le montant de chaque intervention qui a un montant supérieur à 3000**

**On fait une somme de ces montants**

****

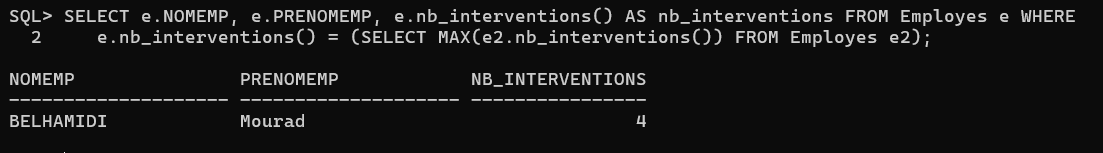
**13.** Donner la liste des employés ayant fait le plus grand nombre d’interventions.

SELECT e.NOMEMP, e.PRENOMEMP, e.nb\_interventions() AS nb\_interventions FROM Employes e WHERE e.nb\_interventions() = (SELECT MAX(e2.nb\_interventions()) FROM Employes e2);

**On calcule le nombre d’interventions de chaque employé en utilisant la méthode nb\_interventions définie précédemment**

**On séléctionne le nombre d’interventions maximale en utilisant MAX()**

**On affiche les noms, prénoms, nombre d’interventions des employés à qui le nombre d’interventions correspond au nombre maximale des interventions trouvé**



**Partie II : NoSQL – Modèle orienté « documents »**

**A-Modélisation orientée document**

**- Proposer une modélisation orientée document de la base de données décrite dans la partie I, dans ce cas.**

**{**

**"numVehicule": "<numero\_vehicule>",**

**"immatriculation": "<numero\_immatriculation>",**

**"annee": "<annee\_vehicule>",**

**"marque": {**

**"numMarque": "<numero\_marque>",**

**"marque": "<nom\_marque>",**

**"pays": "<pays\_marque>"**

**},**

**"interventions": [**

**{**

**"numIntervention": "<numero\_intervention>",**

**"typeIntervention": "<type\_intervention>",**

**"dateDebut": "<date\_debut\_intervention>",**

**"dateFin": "<date\_fin\_intervention>",**

**"cout": "<cout\_intervention>",**

**"employes": [**

**{**

**"numEmploye": "<numero\_employe>",**

**"nomEmploye": "<nom\_employe>",**

**"prenomEmploye": "<prenom\_employe>",**

**"categorieEmploye": "<categorie\_employe>",**

**"salaiereEmploye": "<salaire\_employe>"**

**}**

**]**

**}**

**]**

**}**

**- Illustrez votre modélisation par un exemple (ou plus) de la BD fournie**

**- exemple 1:**

{

"numVehicule": 1,

"immatriculation": "0012519216",

"annee": 1992,

"marque": {

"numMarque": 17,

"marque": "RENAULT",

"pays": "FRANCE"

},

"interventions": [

{

"numIntervention": 7,

"typeIntervention": "Entretien",

"dateDebut": "2006-04-09 09:00:00",

"dateFin": "2006-04-09 18:00:00",

"cout": 8000,

"employes": [

{

"numEmploye": 55,

"nomEmploye": "HADJ",

"prenomEmploye": "Zouhir",

"categorieEmploye": "Assistant",

"salaireEmploye": 12000

},

{

"numEmploye": 65,

"nomEmploye": "MOHAMMEDI",

"prenomEmploye": "Mustapha",

"categorieEmploye": "Mécanicien",

"salaireEmploye": 24000

}

]

}

]

}

- exemple 2:

{

"numVehicule": 28,

"immatriculation": "1986219904",

"annee": 1999,

"marque": {

"numMarque": 2,

"marque": "AUDI",

"pays": "ALLEMAGNE"

},

"interventions": [

{

"numIntervention": 11,

"typeIntervention": "Réparation",

"dateDebut": "2006-03-08 14:00:00",

"dateFin": "2006-03-17 12:00:00",

"cout": 36000,

"employes": [

{

"numEmploye": 59,

"nomEmploye": "BELHAMIDI",

"prenomEmploye": "Mourad",

"categorieEmploye": "Mécanicien",

"salaireEmploye": 19500

},

{

"numEmploye": 64,

"nomEmploye": "BADI",

"prenomEmploye": "Hatem",

"categorieEmploye": "Assistant",

"salaireEmploye": 14000

},

{

"numEmploye": 53,

"nomEmploye": "LACHEMI",

"prenomEmploye": "Bouzid",

"categorieEmploye": "Mécanicien",

"salaireEmploye": 25000

}

]

}

]

}

-exemple3:

{

"numVehicule" : 15,

"immatriculation" : "6254319916",

"annee" : 1999,

"marque" : {

"numMarque" : 4,

"marque" : "BMW",

"pays" : "ALLEMAGNE"

},

"interventions" : [

{

"numIntervention" : 1234,

"typeIntervention" : "Réparation",

"dateDebut" : ISODate("2022-03-15T08:00:00Z"),

"dateFin" : ISODate("2022-03-18T16:00:00Z"),

"cout" : 45000,

"employes" : [

{

"numEmploye" : 53,

"nomEmploye" : "LACHEMI",

"prenomEmploye" : "Bouzid",

"categorieEmploye" : "Mécanicien",

"salaiereEmploye" : 25000

},

{

"numEmploye" : 59,

"nomEmploye" : "BELHAMIDI",

"prenomEmploye" : "Mourad",

"categorieEmploye" : "Mécanicien",

"salaiereEmploye" : 19500

}

]

}

]

}

**- Justifiez votre choix de conception**

Nous avons choisi cette conception car elle permet de représenter les relations entre les différentes entités (véhicule, marque, interventions, employés) en imbriquant les données liées dans un seul document, ainsi, on évite les jointures coûteuses des bases de données relationnelles.

Cette conception est bien adaptée aux requêtes qui vont porter sur les véhicules, leur marque et leurs interventions par les employés car dans ce modèle, chaque document contient des informations sur un véhicule spécifique, y compris sa marque en tant qu'objet imbriqué, les détails des interventions effectuées sur ce véhicule sous forme d'objets imbriqués, ainsi que les employés impliqués dans chaque intervention sous forme de tableaux d'objets à l’intérieur de l'intervention correspondante.

**B-Remplir la collection**

db.vehicules.insertMany([

{

"numVehicule": 1,

"immatriculation": "0012519216",

"annee": 1992,

"marque": {

"numMarque": 17,

"marque": "RENAULT",

"pays": "FRANCE"

},

"interventions": [

{

"numIntervention": 7,

"typeIntervention": "Entretien",

"dateDebut": "2006-04-09 09:00:00",

"dateFin": "2006-04-09 18:00:00",

"cout": 8000,

"employes": [

{

"numEmploye": 55,

"nomEmploye": "HADJ",

"prenomEmploye": "Zouhir",

"categorieEmploye": "Assistant",

"salaireEmploye": 12000

},

{

"numEmploye": 65,

"nomEmploye": "MOHAMMEDI",

"prenomEmploye": "Mustapha",

"categorieEmploye": "Mécanicien",

"salaireEmploye": 24000

}

]

}

]

},

{

"numVehicule" : 15,

"immatriculation" : "6254319916",

"annee" : 1999,

"marque" : {

"numMarque" : 4,

"marque" : "BMW",

"pays" : "ALLEMAGNE"

},

"interventions" : [

{

"numIntervention" : 1234,

"typeIntervention" : "Réparation",

"dateDebut" : ISODate("2022-03-15T08:00:00Z"),

"dateFin" : ISODate("2022-03-18T16:00:00Z"),

"cout" : 45000,

"employes" : [

{

"numEmploye" : 53,

"nomEmploye" : "LACHEMI",

"prenomEmploye" : "Bouzid",

"categorieEmploye" : "Mécanicien",

"salaiereEmploye" : 25000

},

{

"numEmploye" : 59,

"nomEmploye" : "BELHAMIDI",

"prenomEmploye" : "Mourad",

"categorieEmploye" : "Mécanicien",

"salaiereEmploye" : 19500

}

]

}

]

},

{

"numVehicule": 28,

"immatriculation": "1986219904",

"annee": 1999,

"marque": {

"numMarque": 2,

"marque": "AUDI",

"pays": "ALLEMAGNE"

},

"interventions": [

{

"numIntervention": 11,

"typeIntervention": "Réparation",

"dateDebut": "2006-03-08 14:00:00",

"dateFin": "2006-03-17 12:00:00",

"cout": 36000,

"employes": [

{

"numEmploye": 59,

"nomEmploye": "BELHAMIDI",

"prenomEmploye": "Mourad",

"categorieEmploye": "Mécanicien",

"salaireEmploye": 19500

},

{

"numEmploye": 64,

"nomEmploye": "BADI",

"prenomEmploye": "Hatem",

"categorieEmploye": "Assistant",

"salaireEmploye": 14000

},

{

"numEmploye": 53,

"nomEmploye": "LACHEMI",

"prenomEmploye": "Bouzid",

"categorieEmploye": "Mécanicien",

"salaireEmploye": 25000

}

]

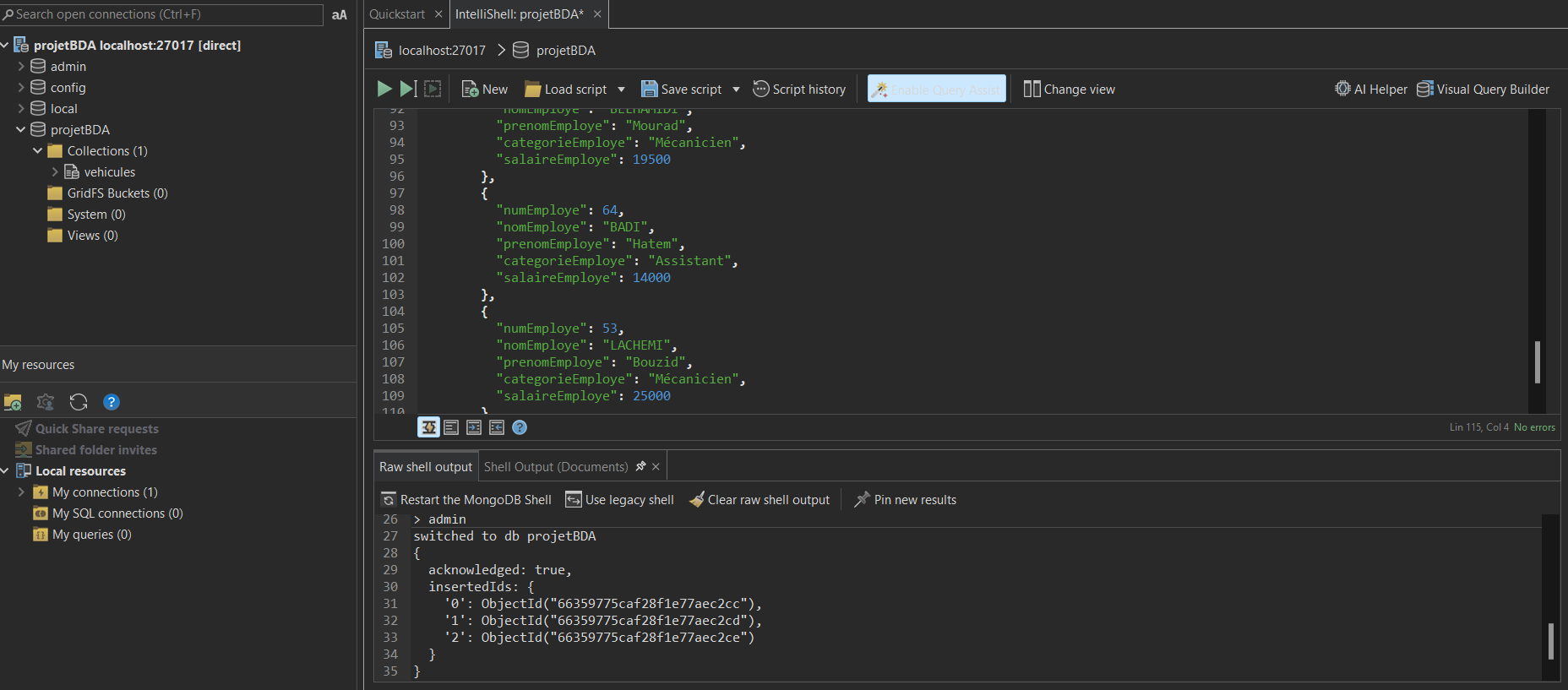
}

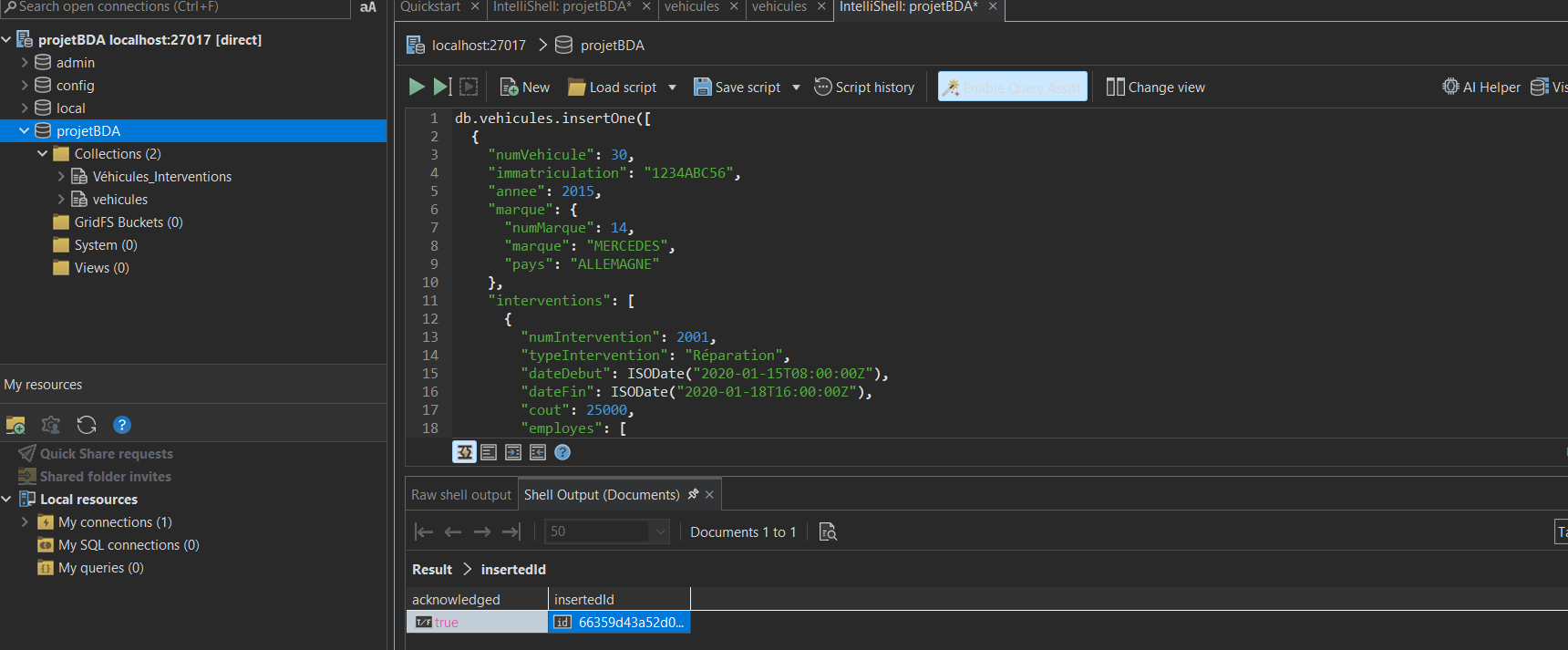
]

},

])

**Capture :**

****

****

**C- Répondre aux requêtes suivantes :**

**- Afficher tous les véhicules de la marque «PORSCHE»**

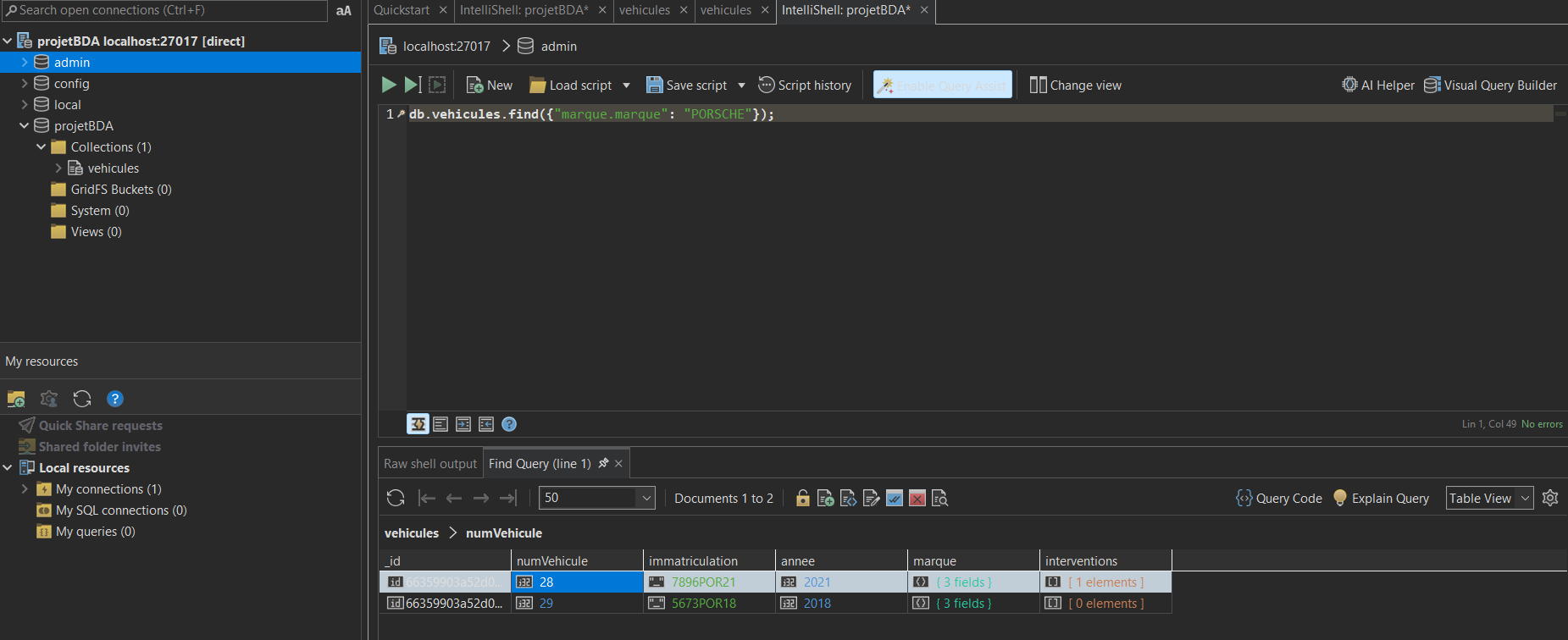
**Explication :**

**Pour visualiser tous les documents de la collection on utilise la fonction find() et on filtre la recherche avec *{"marque.marque": "PORSCHE"}***

**Requete :**

db.vehicules.find({"marque.marque": "PORSCHE"})

**Capture :**

****

**- Récupérer dans une nouvelle collection Véhicules\_Interventions, les matricules des véhicules et le nombre total de s interventions par véhicule ; la collection devra être ordonnée par ordre décroissant du nombre des interventions.**

**Explication :**

**On utilise le *framework d'agrégation* de mongodb et certains de ses opérateurs :**

***$project* : redéfinition des documents (si nécessaire).**

***$sort* : tri sur les agrégats.**

***$out* : spécifie que les résultats de l'opération d'agrégation doivent être écrits dans une nouvelle collection appelée "Véhicules\_Interventions".**

**Requête :**

db.vehicules.aggregate([

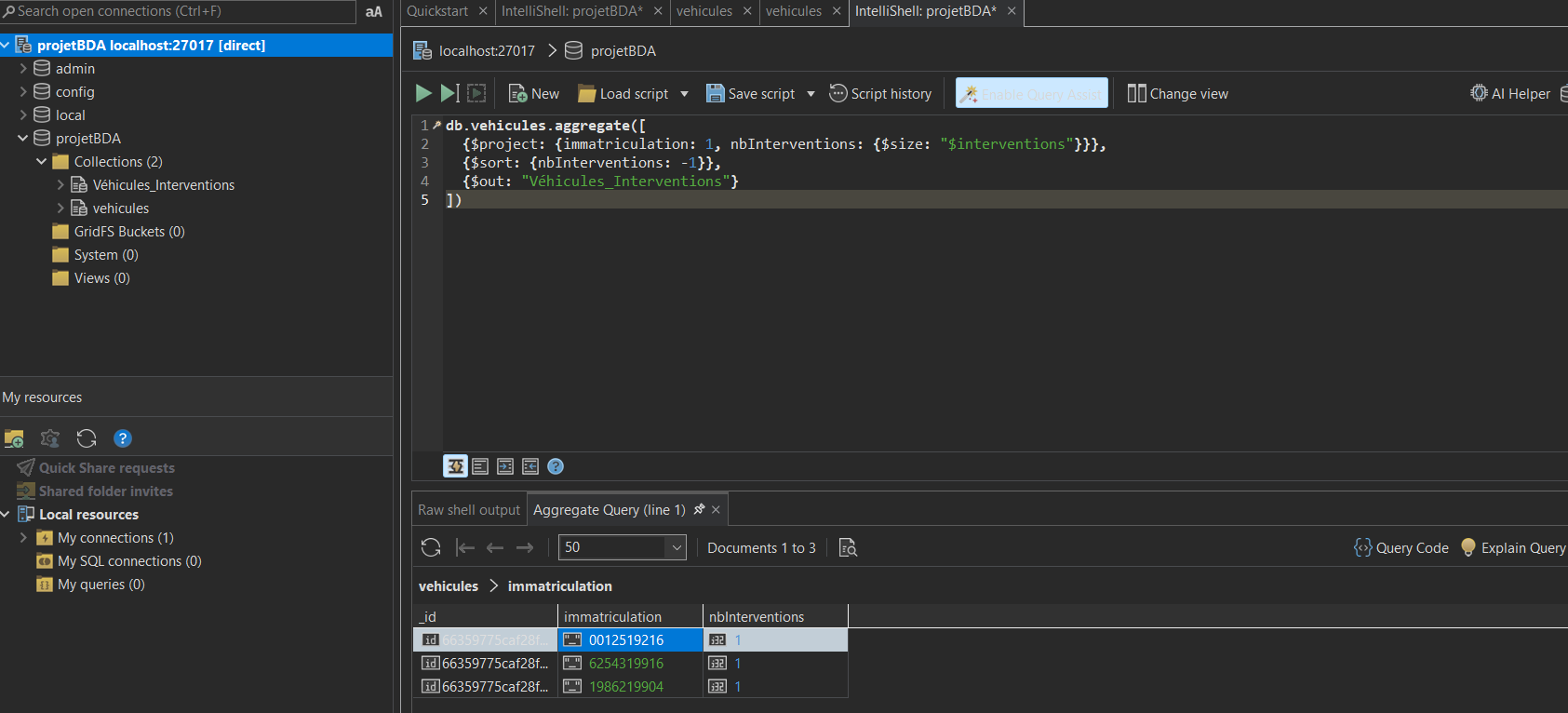
{$project: {immatriculation: 1, nbInterventions: {$size: "$interventions"}}},

{$sort: {nbInterventions: -1}},

{$out: "Véhicules\_Interventions"}

])

**Capture :**



**- Dans une collection véhicule\_bcp\_pannes, récupérer les véhicules dont le nombre des interventions dépasse 6 pannes.**

**Explication :**

**On utilise le *framework d'agrégation* de mongodb et certains de ses opérateurs :**

***$match* : restriction sur les documents à utiliser.**

***$expr*: effectuer des comparaisons ou des opérations mathématiques ici , avec $gt Elle évalue si la taille du tableau interventions est supérieure à 6 pour filtrer les vehicule avec plus de 6 pannes**

***$out* : spécifie que les résultats de l'opération d'agrégation doivent être écrits dans une nouvelle collection appelée "** **véhicule\_bcp\_pannes ".**

**Requête :**

db.vehicules.aggregate([

{

$match: {

interventions: { $exists: true, $type: "array" },

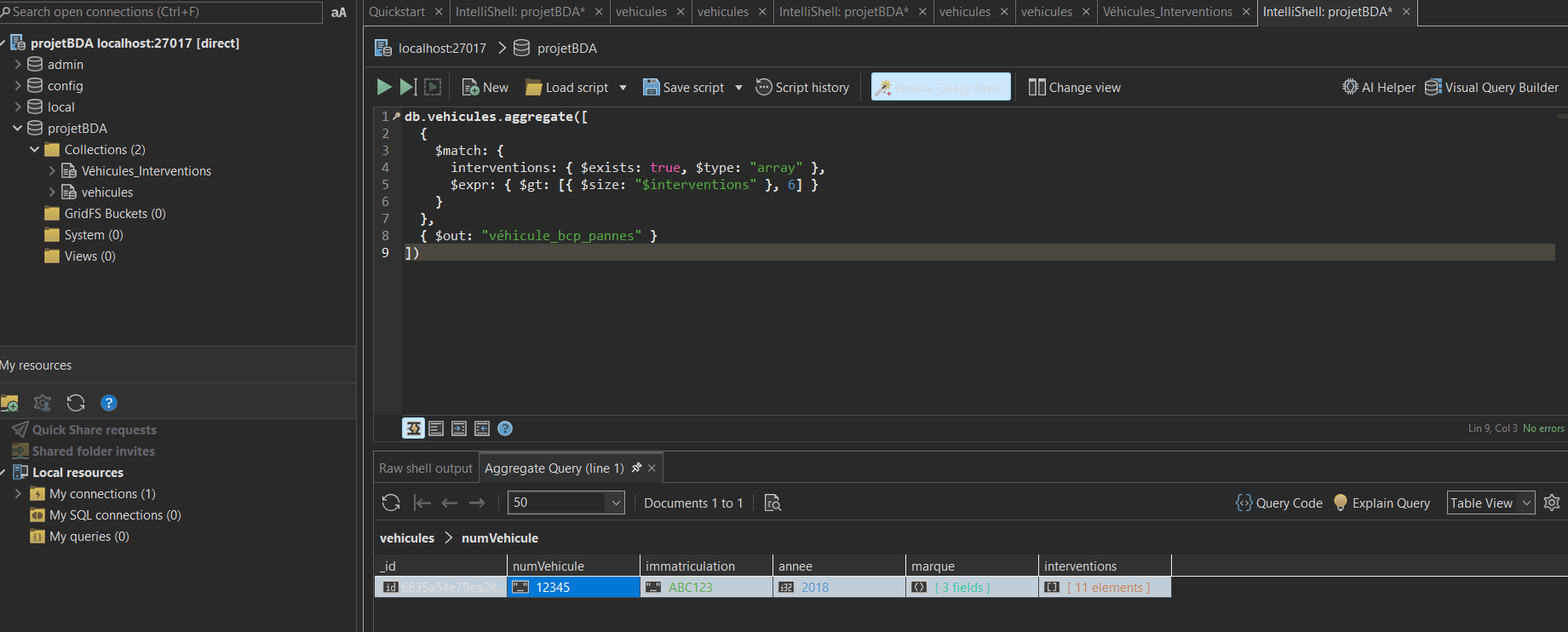
$expr: { $gt: [{ $size: "$interventions" }, 6] }

}

},

{ $out: "véhicule\_bcp\_pannes" }

])

**Capture :**

**- Récupérer dans une collection employe-interv, toutes les interventions d’un employé.**

**Explication :**

**On utilise le *framework d'agrégation* de mongodb et certains de ses opérateurs :**

***$match* : restriction sur les documents à utiliser. filtre les documents en fonction du nulero de lemployé**

***$out* : spécifie que les résultats de l'opération d'agrégation doivent être écrits dans une nouvelle collection appelée "** **employe-interv ".**

***$group* : regroupements et calculs à effectuer. elle regroupe les documents en fonction du numéro d'intervention, puis crée un nouveau tableau employes**

***$unwind* : découpage de tableaux**. **décompose le tableau interventions en autant de documents distincts que d'éléments dans ce tableau.**

**Requête :**

db.vehicules.aggregate([

{$unwind: "$interventions"},

{$unwind: "$interventions.employes"},

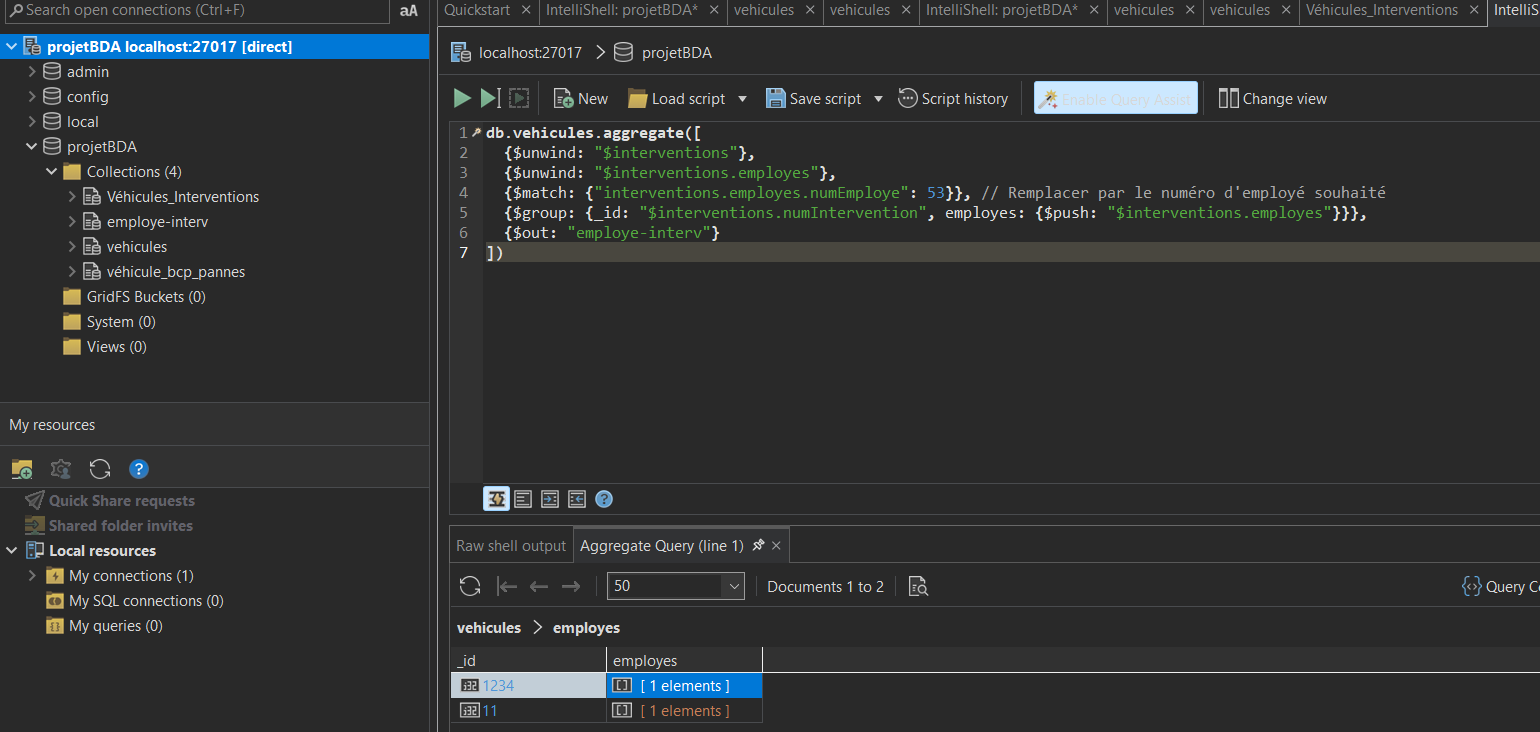
{$match: {"interventions.employes.numEmploye": 53}},

{$group: {\_id: "$interventions.numIntervention", employes: {$push: "$interventions.employes"}}},

{$out: "employe-interv"}

])

**Capture :**



**- Augmenter de 8000DA, le salaire des employés de catégorie « Mécanicien»**

**$inc: {"interventions.employes.salaiereEmploye": 8000} :** **on incrémente le champ salaiereEmploye de chaque employé de interventions.employes des documents sélectionnés par 8000.**

**"interventions.employes.categorieEmploye": "Mécanicien"} : on precise que la mise a jour se fait uniquement ou la categorie de lemploye est « mecanicien »**

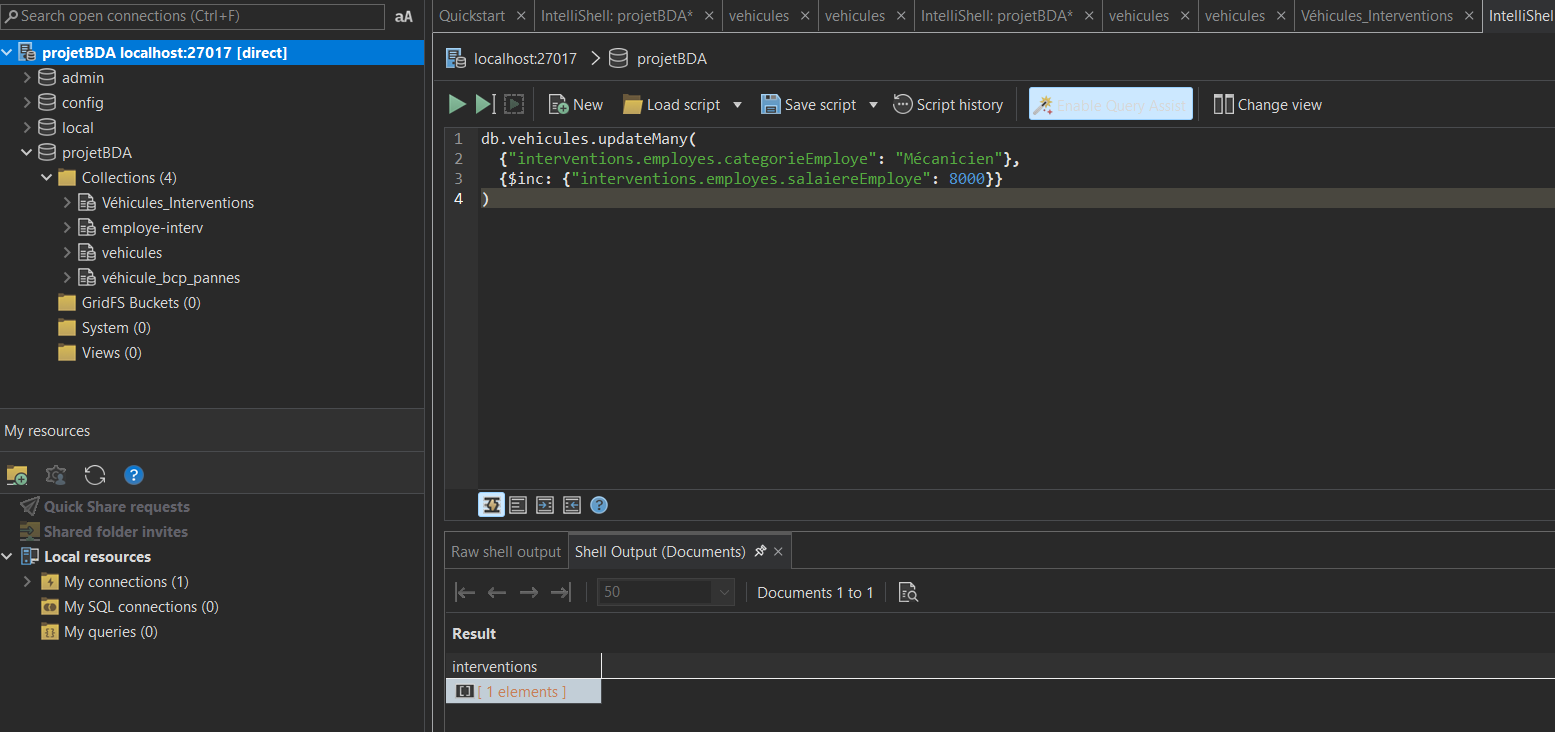
**Requête :**

db.vehicules.updateMany(

{"interventions.employes.categorieEmploye": "Mécanicien"},

{$inc: {"interventions.employes.salaiereEmploye": 8000}}

)

**Capture :**

**- Reprendre la 4ième requête à l’aide du paradigme Map-Reduce.**

**Requête :**

**var Map = function() {**

**if (this.interventions) {**

**this.interventions.forEach(function(intervention) {**

**intervention.employes.forEach(function(employe) {**

**emit(employe.numEmploye, intervention.numIntervention);**

**});**

**});**

**}**

**};**

**var Reduce = function(numEmploye, interventions) {**

**return Array.from(new Set(interventions));**

**};**

**db.vehicules.mapReduce(Map, Reduce, {**

**query: {"interventions.employes.numEmploye": 53},**

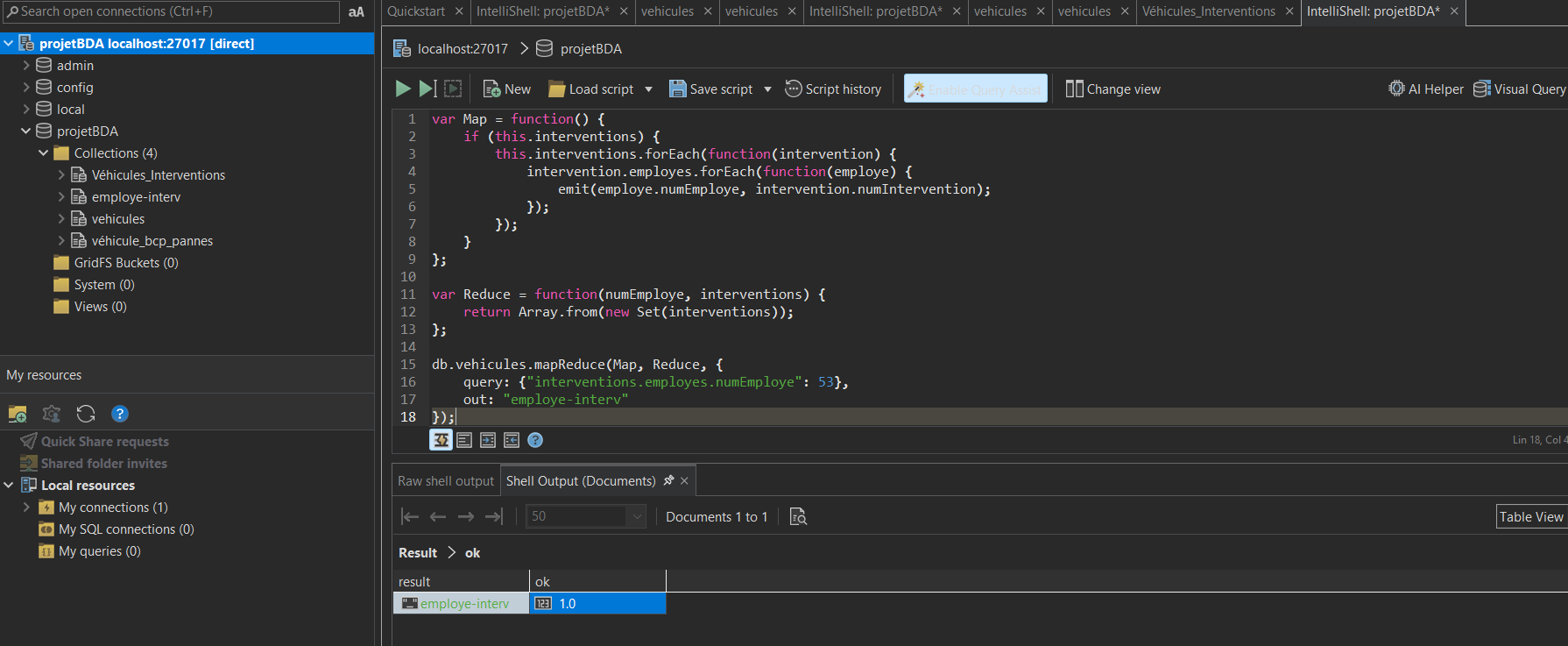
**out: "employe-interv"**

**});**

**Explication :**

**Cette requête mapReduce va : Mapper chaque employé avec ses interventions associées. Réduire les interventions en un ensemble pour éliminer les doublons. Finalement, stocker le résultat dans la collection employe-interv pour les employés ayant un numéro d'employé égal à 53.**

**Capture :**

****

**D- Analyse**

**Donnez votre analyse par rapport à ces requêtes**

Cette modélisation orientée document convient bien aux requêtes qui portent principalement sur les véhicules et leurs interventions, en évitant les jointures coûteuses. L'imbrication des données liées facilite l'écriture des requêtes et les rend plus performantes.

Cependant, il faut faire attention à la redondance des données, car les mêmes informations (employés, marque) sont répétées dans chaque document véhicule. Cela peut poser des problèmes lors des mises à jour.

De plus, pour des requêtes qui se portent sur d’autres entités cette modélisation serait moins optimale. Il faudrait alors envisager d'autres modèles de données plus adaptés.