République Algérienne Démocratique et Populaire Ministère de l'Enseignement Supérieur et de la Recherche Scientifique Université des Sciences et Technologies Houari Boumediene

Faculté d'électronique et d'informatique Département d'informatique

Master 1 Informatique Ingénieurie de Logiciels

Bases de Données Avancées

Rapport Projet : SQL3-Oracle et NoSQL (MongoDB)

Réalisé par:

FETHALLAH Mohamed Racim

LARIBI Abdelalim

Enseignante:

Pr. Azzouz

Section: Master 1 IL Group 2

Année académique

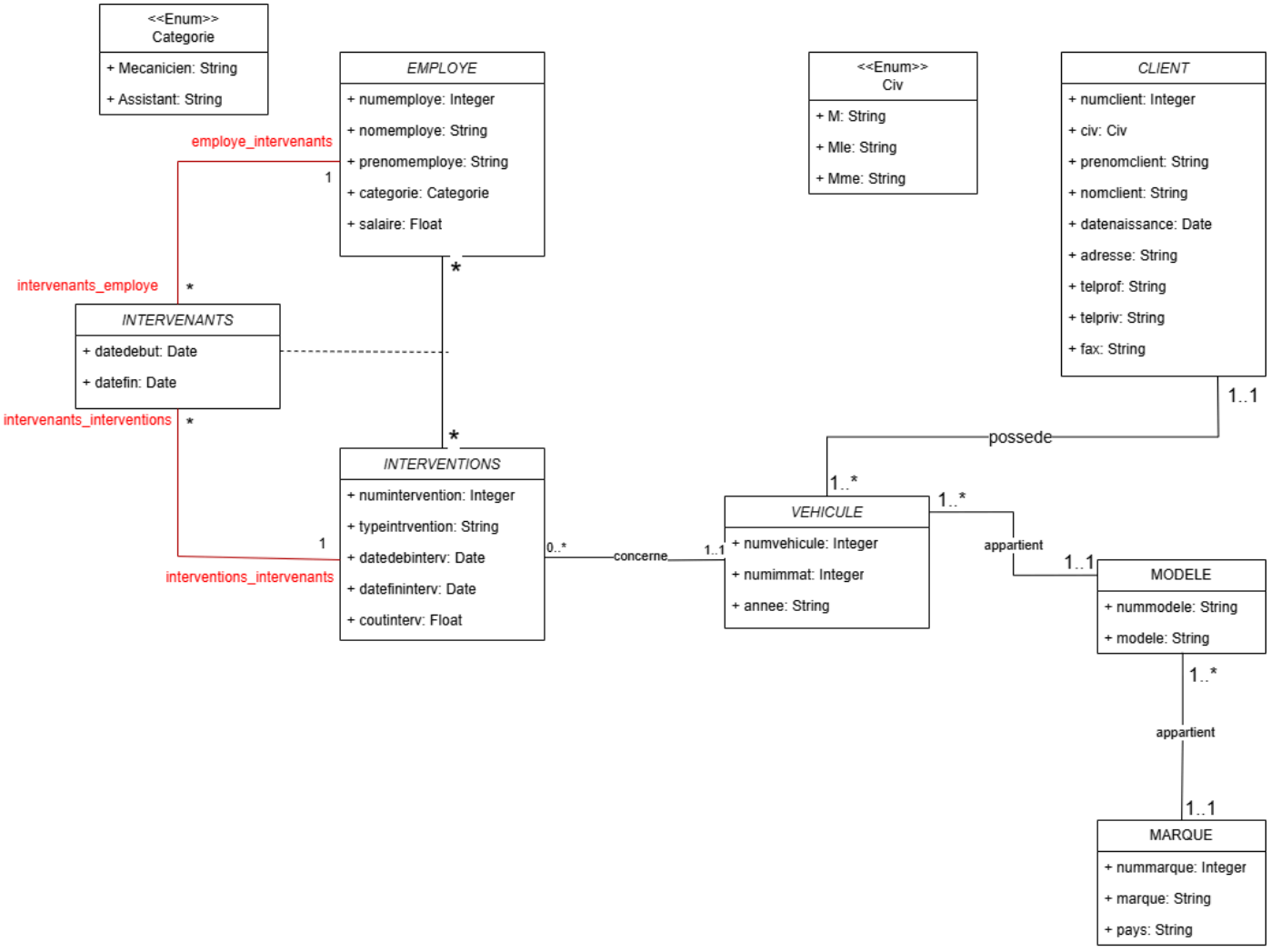
2023/2024

Partie 1: Relationnel-Objet **3**
A - Modelisation orienté objet: 3
1- Diagramme de classe: 3
B - Création des TableSpaces et utilisateur: 4
2. Création des tablespaces SQL3_TBS et SQL3_TempTBS 4
3. Créer un utilisateur SQL3 en lui attribuant les deux tablespaces. 4
4. Donner tous les privilèges à cet utilisateur. 4
C - Langage de définition de données: 5
5. Définition des types nécessaires par rapport au diagramme de classe. 5
6. Définir les méthodes permettant de : 7
1. Calculer pour chaque employé, le nombre d'interventions effectuées: 7
2. Calculer pour chaque marque, le nombre de modèles: 8
3. Calculer pour chaque modèle, le nombre de véhicules: 8
4. Lister pour chaque client, ses véhicules: 9
5. Calculer pour chaque marque, son chiffre d'affaire: 9
7. Définir les tables nécessaires à la base de données. 10
D - Langage de manipulation de données: 11
8. Remplir toutes les tables par les instances fournies en annexe. (script SQL3.sql) 11
E - Langage d'interrogation de données: 11
9. Lister les modèles et leur marque: 11
10. Lister les véhicules sur lesquels il y a au moins une intervention: 12
11. Quelle est la durée moyenne d'une intervention? (en jours) 12
12. Donner le montant global des interventions dont le coût d'intervention est supérieur à 30000 DA? 13
13. Donner la liste des employés ayant fait le plus grand nombre d'interventions. 14
Partie 2: NoSQL – Modèle orienté «documents» **16**
A - Modélisation orientée document 16
B- Remplir la collection (Script mongodb.js) 17
C- Répondre aux requetes: 17
C. 1 Afficher tous les véhicules de la marque «PORSCHE» 17
C. 2 Récupérer dans une nouvelle collection Véhicules_Interventions, les matricules des véhicules et le nombre total des interventions par véhicule ; la collection devra être ordonnée par ordre décroissant du nombre des interventions. 18
C. 3 Dans une collection véhicule_bcp_pannes, récupérer les véhicules dont le nombre des interventions dépasse 6 pannes. 19
C. 4 Récupérer dans une collection employe-interv, toutes les interventions d'un employé. 19
C. 5 Augmenter de 8000DA, le salaire des employés de catégorie « Mécanicien» 20
C. 6 Reprendre la 4ième requête à l'aide du paradigme Map-Reduce. 20
D. Analyse 21
Donnez votre analyse par rapport à ces requêtes: 21

Partie 1: Relationnel-Objet

A - Modelisation orienté objet:

1- Diagramme de classe:



B - Création des TableSpaces et utilisateur:

2. Création des tablespaces SQL3_TBS et SQL3_TempTBS

```
--TableSpace
CREATE TABLESPACE SQL3_TBS
DATAFILE 'c:\sql3_tbs.dat' SIZE 50M
AUTOEXTEND ON;

--Temporary Tablespace
CREATE TEMPORARY TABLESPACE SQL3_TempTBS
TEMPFILE 'c:\sql3_Temptbs.dat' SIZE 25M;
```

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

```
CREATE USER SQL3_BDA IDENTIFIED BY admin
DEFAULT TABLESPACE SQL3_TBS
TEMPORARY TABLESPACE SQL3_TempTBS
QUOTA UNLIMITED ON SQL3_TBS;
```

4. Donner tous les privilèges à cet utilisateur.

```
GRANT ALL PRIVILEGES TO SQL3_BDA;
```

```
SQL> CREATE TABLESPACE SQL3_TBS
  2  DATAFILE 'c:\sql3_tbs.dat' SIZE 50M
  3  AUTOEXTEND ON;

Tablespace created.

SQL> --Temporary Tablespace
SQL> CREATE TEMPORARY TABLESPACE SQL3_TempTBS
  2  TEMPFILE 'c:\sql3_Temptbs.dat' SIZE 25M;

Tablespace created.

SQL> CREATE USER SQL3_BDA IDENTIFIED BY admin
  2  DEFAULT TABLESPACE SQL3_TBS
  3  TEMPORARY TABLESPACE SQL3_TempTBS
  4  QUOTA UNLIMITED ON SQL3_TBS;

User created.

SQL> GRANT ALL PRIVILEGES TO SQL3_BDA;

Grant succeeded.
```

C - Langage de définition de données:

5. Définition des types nécessaires par rapport au diagramme de classe.

```
CREATE type tclient;
/
CREATE type temploye;
/
CREATE type tmarque;
/
CREATE type tmodele;
/
CREATE type tvehicule;
/
CREATE type tinterventions;
/
CREATE type tintervenants;
/

CREATE type nt_ref_modele AS TABLE OF ref tmodele;
/
CREATE type nt_ref_vehicule AS TABLE OF ref tvehicule;
/
CREATE type nt_ref_interventions AS TABLE OF ref tinterventions;
/
CREATE type nt_ref_intervenants AS TABLE OF ref tintervenants;
/


CREATE OR REPLACE TYPE tclient AS OBJECT(
    numclient INTEGER,
    civ VARCHAR2(3),
    prenomclient VARCHAR2(50),
    NOMCLIENT VARCHAR2(50),
    DATENAISSANCE DATE,
    ADRESSE VARCHAR2(100),
    TELPROF VARCHAR2(10),
    TELPRIV VARCHAR2(10),
    FAX VARCHAR2(10),
    client_vehicule nt_ref_vehicule
);
/
```

```
CREATE OR REPLACE TYPE temploye AS OBJECT(
    numemploye INTEGER,
    nomemploye VARCHAR2(50),
    prenomemploye VARCHAR2(50),
    categorie VARCHAR2(50),
    salaire float,
    employe_intervenants nt_ref_intervenants
);
/



CREATE OR REPLACE TYPE tmarque AS OBJECT(
    NUMMARQUE INTEGER,
    MARQUE VARCHAR2(50),
    PAYS VARCHAR2(50),
    marque_modele nt_ref_modele
);
/



CREATE OR REPLACE TYPE tmodele AS OBJECT(
    nummodele INTEGER,
    modele VARCHAR2(50),
    marque REF tmarque,
    modele_vehicule nt_ref_vehicule
);
/




CREATE OR REPLACE TYPE tvehicule AS OBJECT(
    NUMVEHICULE INTEGER,
    NUMIMMAT VARCHAR2(50),
    ANNEE VARCHAR2(50),
    client REF tclient,
    modele REF tmodele,
    vehicule_interventions nt_ref_interventions
```

```
);
/

CREATE OR REPLACE TYPE tinterventions AS OBJECT(
    NUMINTERVENTION INTEGER,
    TYPEINTERVENTION VARCHAR2(50),
    DATEDEBINTERV DATE,
    DATEFININTERV DATE,
    COUTINTERV float,
    vehicule REF tvehicule,
    interventions_intervenants nt_ref_intervenants
);
/

CREATE OR REPLACE TYPE tintervenants AS OBJECT(
    DATEDEBUT DATE,
    DATEFIN DATE,
    interventions REF tinterventions,
    employe REF temploye
);
/
```

6. Définir les méthodes permettant de :

1. Calculer pour chaque employé, le nombre d'interventions effectuées:

```
ALTER TYPE temploye
ADD MEMBER FUNCTION calcul_interventions RETURN INTEGER
CASCADE;


CREATE OR REPLACE TYPE BODY temploye AS
    MEMBER FUNCTION calcul_interventions RETURN INTEGER IS
        nombre_interventions INTEGER := 0;
    BEGIN
        nombre_interventions := self.employe_intervenants.COUNT;
        RETURN nombre_interventions;
    END;
END;
/
```

2. Calculer pour chaque marque, le nombre de modèles:

```
ALTER TYPE tmarque
ADD MEMBER FUNCTION calcul_modeles RETURN INTEGER
CASCADE;

--good
CREATE OR REPLACE TYPE BODY tmarque AS
    MEMBER FUNCTION calcul_modeles RETURN INTEGER IS
        nombre_modeles INTEGER := 0;
    BEGIN
        nombre_modeles := self.marque_modele.COUNT;
        RETURN nombre_modeles;
    END;
END;
/
```

3. Calculer pour chaque modèle, le nombre de véhicules:

```
ALTER TYPE tmodele
ADD MEMBER FUNCTION calcul_vehicules RETURN INTEGER
CASCADE;


CREATE OR REPLACE TYPE BODY tmodele AS
    MEMBER FUNCTION calcul_vehicules RETURN INTEGER IS
        nombre_vehicules INTEGER := 0;
    BEGIN
        nombre_vehicules := self.modele_vehicule.COUNT;
        RETURN nombre_vehicules;
    END;
END;
/
```

4. Lister pour chaque client, ses véhicules:

```
ALTER TYPE tclient
ADD MEMBER FUNCTION lister_vehicules RETURN nt_ref_vehicule
CASCADE;


CREATE OR REPLACE TYPE BODY tclient AS
    MEMBER FUNCTION lister_vehicules RETURN nt_ref_vehicule IS
    BEGIN
        RETURN self.client_vehicule;
    END;
END;
/
```

5. Calculer pour chaque marque, son chiffre d'affaire:

```
ALTER TYPE tmarque
ADD MEMBER FUNCTION chiffre_affaire RETURN INTEGER
CASCADE;


CREATE OR REPLACE TYPE BODY tmarque AS
   MEMBER FUNCTION chiffre_affaire RETURN INTEGER IS
        total_chiffre_affaire INTEGER := 0;
    BEGIN
        -- Parcours de la collection des références de modèles
        FOR i IN 1..self.marque_modele.COUNT LOOP
            -- Parcours de la collection des références de véhicules pour
chaque modèle
            FOR j IN 1..self.marque_modele(i).modele_vehicule.COUNT LOOP
                -- Parcours de la collection des références d'interventions
pour chaque véhicule
                FOR k IN
1..self.marque_modele(i).modele_vehicule(j).vehicule_interventions.COUNT LOOP
                    -- Ajout du coût de l'intervention au chiffre d'affaires
total
                    total_chiffre_affaire := total_chiffre_affaire +
self.marque_modele(i).modele_vehicule(j).vehicule_interventions(k).COUTINTERV;
                END LOOP;
            END LOOP;
        END LOOP;
```

```
        -- Retour du chiffre d'affaires total pour la marque
        RETURN total_chiffre_affaire;
    END chiffre_affaire;
END;
/
```

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

```
CREATE TABLE client OF tclient (
    CONSTRAINT numclient_pk PRIMARY KEY (numclient),
    CONSTRAINT civ_check CHECK (civ IN ('M.', 'Mle', 'Mme')))
NESTED TABLE vehicules STORE AS tab_client_vehicules;



CREATE TABLE employe OF temploye (
    CONSTRAINT numemploye_pk PRIMARY KEY (NUMEMPLOYE),
    CONSTRAINT categorie_check CHECK (categorie IN ('Mecanicien',
'Assistant')))
NESTED TABLE intervenants STORE AS tab_employe_intervenant;


CREATE TABLE marque OF tmarque (
    CONSTRAINT nummarque_pk PRIMARY KEY (NUMMARQUE))
NESTED TABLE modeles STORE AS tab_marque_modele;

CREATE TABLE modele OF tmodele (
    CONSTRAINT nummodele_pk PRIMARY KEY (NUMMODELE),
    CONSTRAINT nummarque_fk FOREIGN KEY (marque) REFERENCES marque)
NESTED TABLE vehicules STORE AS tab_modele_vehicule;

CREATE TABLE vehicule OF tvehicule (
    CONSTRAINT numvehicule_pk PRIMARY KEY (NUMVEHICULE),
    CONSTRAINT numclient_fk FOREIGN KEY (client) REFERENCES client,
    CONSTRAINT nummodele_fk FOREIGN KEY (modele) REFERENCES modele)
NESTED TABLE interventions STORE AS tab_vehicule_intervention;

CREATE TABLE interventions OF tinterventions (
    CONSTRAINT numintervention_pk PRIMARY KEY (NUMINTERVENTION),
    CONSTRAINT numvehicule_fk FOREIGN KEY (vehicule) REFERENCES
vehicule)
```

```
    NESTED TABLE intervenants STORE AS tab_intervenant_intervention;

CREATE TABLE intervenants OF tintervenants (
    CONSTRAINT numintervention_fk FOREIGN KEY (intervention) REFERENCES
interventions,
    CONSTRAINT numemploye_fk FOREIGN KEY (employe) REFERENCES employe);
```

D - Langage de manipulation de données:

8. Remplir toutes les tables par les instances fournies en annexe. (script SQL3.sql)

E - Langage d'interrogation de données:

9. Lister les modèles et leur marque:

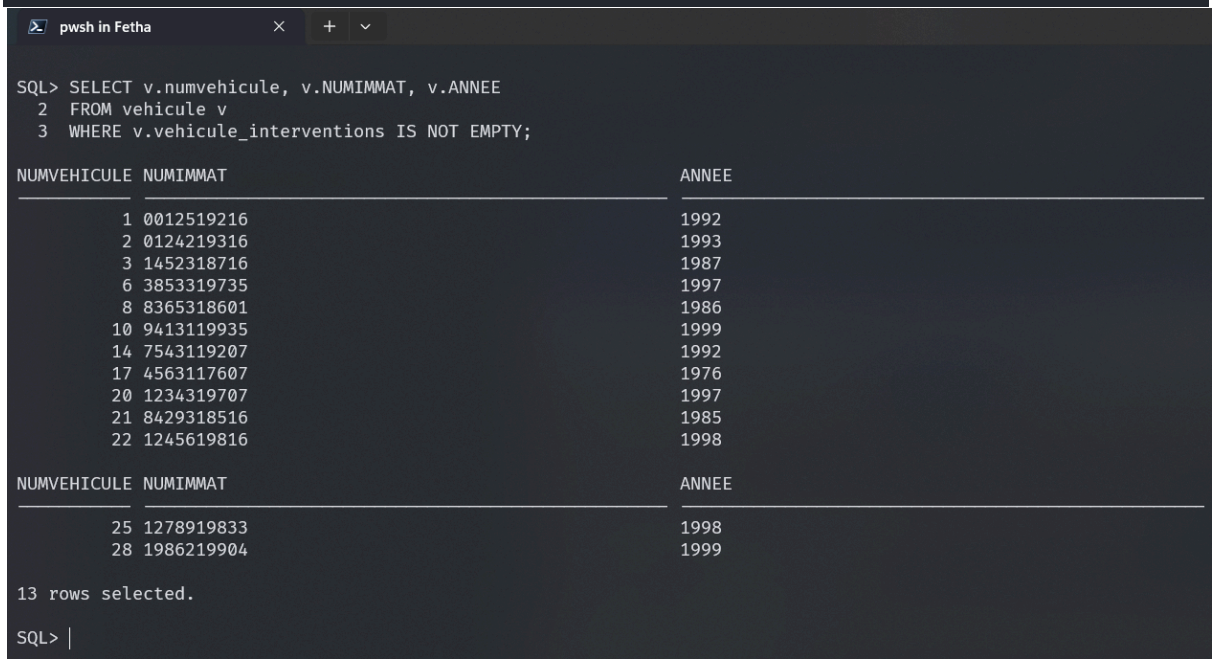
```
SELECT m.nummodele, m.modele, deref(m.marque).marque
FROM modele m;
```

pwsh in Fetha

```
SQL> SELECT m.nummodele, m.modele, deref(m.marque).marque
  2  FROM modele m;

 NUMMODELE MODELE                                             DEREF(M.MARQUE).MARQUE
---------- -------------------------------------------------- --------------------------------------------------
         2 Diablo                                             LAMBORGHINI
         3 Série 5                                            AUDI
        12 Bentley-Continental                                ROLLS-ROYCE
        19 M 3                                                BMW
        25 Séville                                            CADILLAC
        18 300 M                                              CHRYSLER
        11 550 Maranello                                      FERRARI
        20 XJ 8                                               JAGUAR
        13 Spider                                             ALFA-ROMEO
         4 NSX                                                ALFA-ROMEO
        24 GS 300                                             LEXUS

 NUMMODELE MODELE                                             DEREF(M.MARQUE).MARQUE
---------- -------------------------------------------------- --------------------------------------------------
         8 Esprit                                             LOTUS
        14 Evoluzione                                         MASERATI
        23 Classe E                                           MERCEDES
         5 Classe C                                           MERCEDES
        21 406 Coupé                                          PEUGEOT
         9 605                                                PEUGEOT
        16 Boxter                                             PORSCHE
        15 Carrera                                            PORSCHE
         6 Safrane                                            RENAULT
        26 95 Cabriolet                                       SAAB
        10 Prévia                                             TOYOTA

 NUMMODELE MODELE                                             DEREF(M.MARQUE).MARQUE
---------- -------------------------------------------------- --------------------------------------------------
        22 300 Atlantic                                       VENTURI
         7 400 GT                                             VENTURI
        17 S 80                                               VOLVO

25 rows selected.

SQL>
```

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

```sql
SELECT v.numvehicule, v.NUMIMMAT, v.ANNEE
FROM vehicule v
WHERE v.vehicule_interventions IS NOT EMPTY;

-- ou --

SELECT v.NUMIMMAT, v.ANNEE
FROM vehicule v
WHERE EXISTS (
    SELECT 1
    FROM TABLE(v.vehicule_interventions) i
);
```



```
pwsh in Fetha

SQL> SELECT v.numvehicule, v.NUMIMMAT, v.ANNEE
  2  FROM vehicule v
  3  WHERE v.vehicule_interventions IS NOT EMPTY;

NUMVEHICULE NUMIMMAT                                          ANNEE
----------- ------------------------------------------------- --------------------------------------------------
          1 0012519216                                        1992
          2 0124219316                                        1993
          3 1452318716                                        1987
          6 3853319735                                        1997
          8 8365318601                                        1986
         10 9413119935                                        1999
         14 7543119207                                        1992
         17 4563117607                                        1976
         20 1234319707                                        1997
         21 8429318516                                        1985
         22 1245619816                                        1998

NUMVEHICULE NUMIMMAT                                          ANNEE
----------- ------------------------------------------------- --------------------------------------------------
         25 1278919833                                        1998
         28 1986219904                                        1999

13 rows selected.

SQL>
```

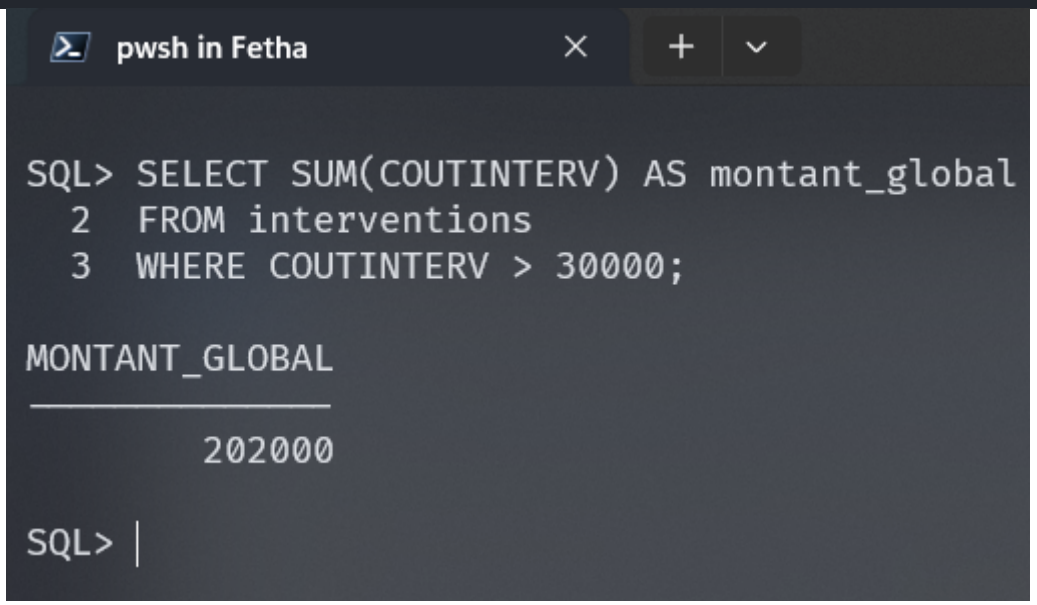
11. Quelle est la durée moyenne d’une intervention? (en jours)

```sql
SELECT i.NUMINTERVENTION, AVG(DATEFININTERV - DATEDEBINTERV) AS
duree_moyenne_intervention
FROM interventions i
group by i.NUMINTERVENTION;
```

```
pwsh in Fetha

SQL> SELECT i.NUMINTERVENTION, AVG(DATEFININTERV - DATEDEBINTERV) AS duree_moyenne_intervention
  2  FROM interventions i
  3  group by i.NUMINTERVENTION;

NUMINTERVENTION DUREE_MOYENNE_INTERVENTION
--------------- --------------------------
              1                      1,125
              6                 1,16666667
             11                 8,91666667
             13                 ,166666667
              2                      1,375
             14                 1,91666667
              4                       ,375
              5                      3,375
              8                 1,16666667
              3                 2,91666667
              7                       ,375

NUMINTERVENTION DUREE_MOYENNE_INTERVENTION
--------------- --------------------------
              9                       ,375
             10                      1,375
             12                      2,375
             15                       ,125
             16                      3,125

16 rows selected.

SQL>
```

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

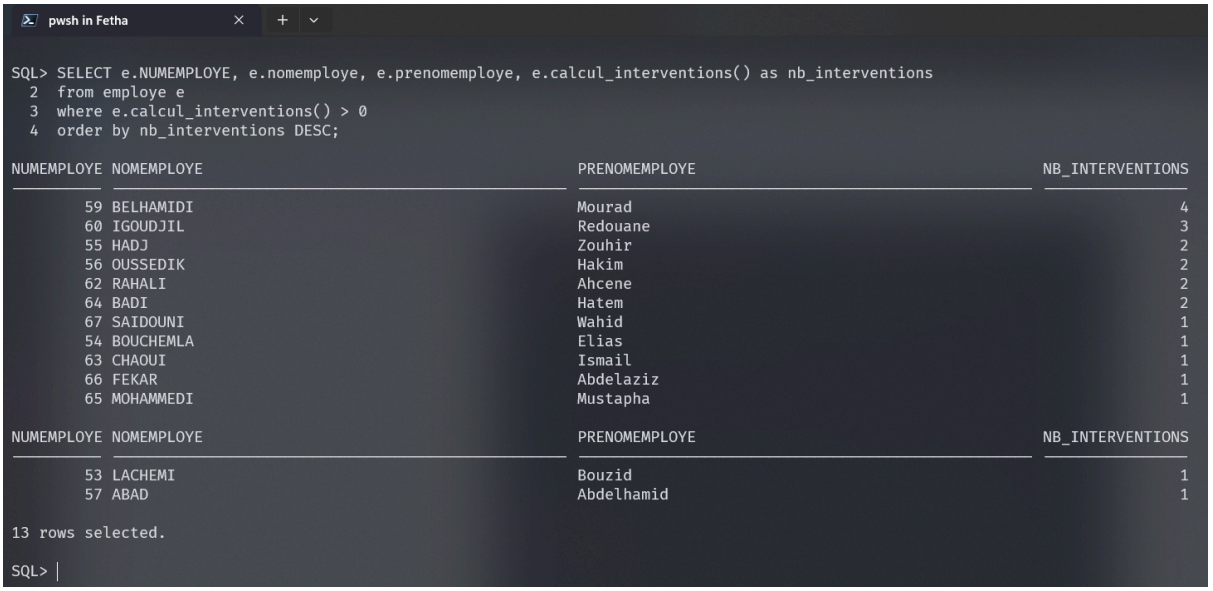
```sql
SELECT SUM(COUTINTERV) AS montant_global
FROM interventions
WHERE COUTINTERV > 30000;
```



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

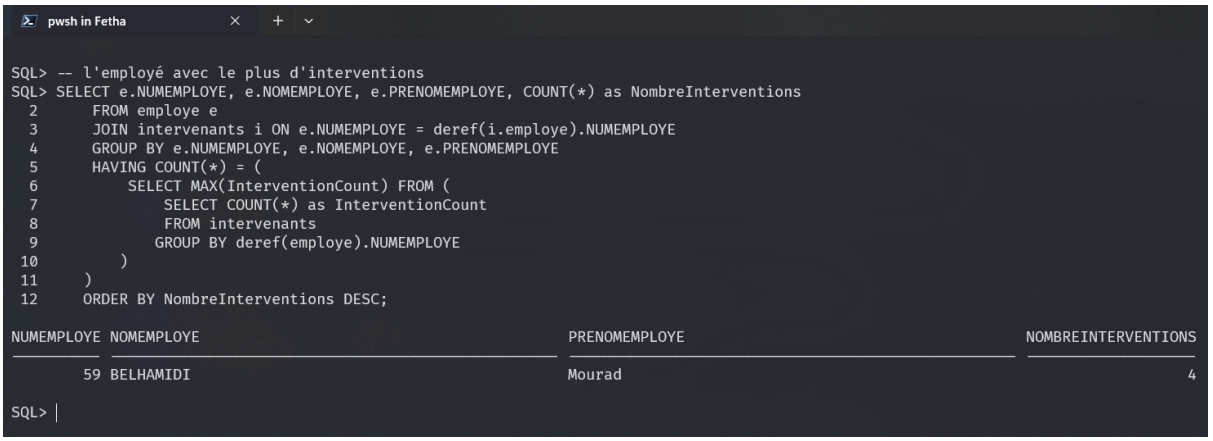
Soit en utilisant la méthode pré defini auparavant calcul_intervntions ou une autre méthode

```
SELECT e.NUMEMPLOYE, e.nomemploye, e.prenomemploye, e.calcul_interventions()
as nb_interventions
from employe e
where e.calcul_interventions() > 0
order by nb_interventions DESC;


SELECT e.numemploye, e.nomemploye, e.prenomemploye, count(*) as
NombreInterventions
FROM employe e
JOIN intervenants i ON e.numemploye = deref(i.employe).numemploye
group by  e.numemploye, e.nomemploye, e.prenomemploye
order by NombreInterventions DESC;









-- l'employé avec le plus d'interventions
SELECT e.NUMEMPLOYE, e.NOMEMPLOYE, e.PRENOMEMPLOYE, COUNT(*) as
NombreInterventions
    FROM employe e
    JOIN intervenants i ON e.NUMEMPLOYE = deref(i.employe).NUMEMPLOYE
    GROUP BY e.NUMEMPLOYE, e.NOMEMPLOYE, e.PRENOMEMPLOYE
    HAVING COUNT(*) = (
        SELECT MAX(InterventionCount) FROM (
            SELECT COUNT(*) as InterventionCount
            FROM intervenants
           GROUP BY deref(employe).NUMEMPLOYE
       )
   )
   ORDER BY NombreInterventions DESC;
```



```
pwsh in Fetha

SQL> SELECT e.NUMEMPLOYE, e.nomemploye, e.prenomemploye, e.calcul_interventions() as nb_interventions
  2  from employe e
  3  where e.calcul_interventions() > 0
  4  order by nb_interventions DESC;

NUMEMPLOYE NOMEMPLOYE                                         PRENOMEMPLOYE                                      NB_INTERVENTIONS
---------- -------------------------------------------------- -------------------------------------------------- ----------------
        59 BELHAMIDI                                          Mourad                                                            4
        60 IGOUDJIL                                           Redouane                                                          3
        55 HADJ                                               Zouhir                                                            2
        56 OUSSEDIK                                           Hakim                                                             2
        62 RAHALI                                             Ahcene                                                            2
        64 BADI                                               Hatem                                                             2
        67 SAIDOUNI                                           Wahid                                                             1
        54 BOUCHEMLA                                          Elias                                                             1
        63 CHAOUI                                             Ismail                                                            1
        66 FEKAR                                              Abdelaziz                                                         1
        65 MOHAMMEDI                                          Mustapha                                                          1

NUMEMPLOYE NOMEMPLOYE                                         PRENOMEMPLOYE                                      NB_INTERVENTIONS
---------- -------------------------------------------------- -------------------------------------------------- ----------------
        53 LACHEMI                                            Bouzid                                                            1
        57 ABAD                                               Abdelhamid                                                        1

13 rows selected.

SQL>
```

L’employé avec le plus d’interventions:



```
pwsh in Fetha

SQL> -- l'employé avec le plus d'interventions
SQL> SELECT e.NUMEMPLOYE, e.NOMEMPLOYE, e.PRENOMEMPLOYE, COUNT(*) as NombreInterventions
  2      FROM employe e
  3      JOIN intervenants i ON e.NUMEMPLOYE = deref(i.employe).NUMEMPLOYE
  4      GROUP BY e.NUMEMPLOYE, e.NOMEMPLOYE, e.PRENOMEMPLOYE
  5      HAVING COUNT(*) = (
  6          SELECT MAX(InterventionCount) FROM (
  7              SELECT COUNT(*) as InterventionCount
  8              FROM intervenants
  9             GROUP BY deref(employe).NUMEMPLOYE
 10         )
 11     )
 12     ORDER BY NombreInterventions DESC;

NUMEMPLOYE NOMEMPLOYE                                         PRENOMEMPLOYE                                      NOMBREINTERVENTIONS
---------- -------------------------------------------------- -------------------------------------------------- -------------------
        59 BELHAMIDI                                          Mourad                                                              4

SQL>
```

Partie 2: NoSQL – Modèle orienté «documents»

A - Modélisation orientée document

A.1 Proposer une modélisation orientée document de la base de données décrite dans la partie I, dans ce cas:

Comme la plupart des requêtes porteront sur les véhicules, leurs marques et leurs interventions par les employés, une modélisation orientée document basée sur l'imbrication centrée sur "véhicules" contenant les informations sur le véhicule, sa marque, son modèle et ses interventions, ainsi que les employés serait suffisante pour modeliser notre base de données.

```
//Exemple Collection véhicules
  {
    "numVehicule": 2,
    "numImmat": "0124219316",
    "annee": 1993,
    "modele": {
      "numModele": 20,
      "marque": {
        "numMarque": 9,
        "nomMarque": "JAGUAR",
        "pays": "GRANDE-BRETAGNE"
      },
      "nomModele": "XJ 8"
    },
    "interventions": [
      {
        "numIntervention": 10,
        "typeIntervention": "Entretien et Réparation",
        "dateDeb": ISODate("2006-04-08T09:00:00.000"),
        "dateFin": ISODate("2006-04-09T18:00:00.000"),
        "cout": 45000,
        "employes": [
          {
            "numEmploye": 63,
            "nom": "CHAOUI",
            "prenom": "Ismail",
            "categorie": "Assistant",
            "salaire": 13000,
            "dateDeb": ISODate("2006-04-09T14:00:00.000"),
            "dateFin": ISODate("2006-04-09T18:00:00.000")
          },
          {
            "numEmploye": 67,
            "nom": "SAIDOUNI",
```

Justification du choix de conception:

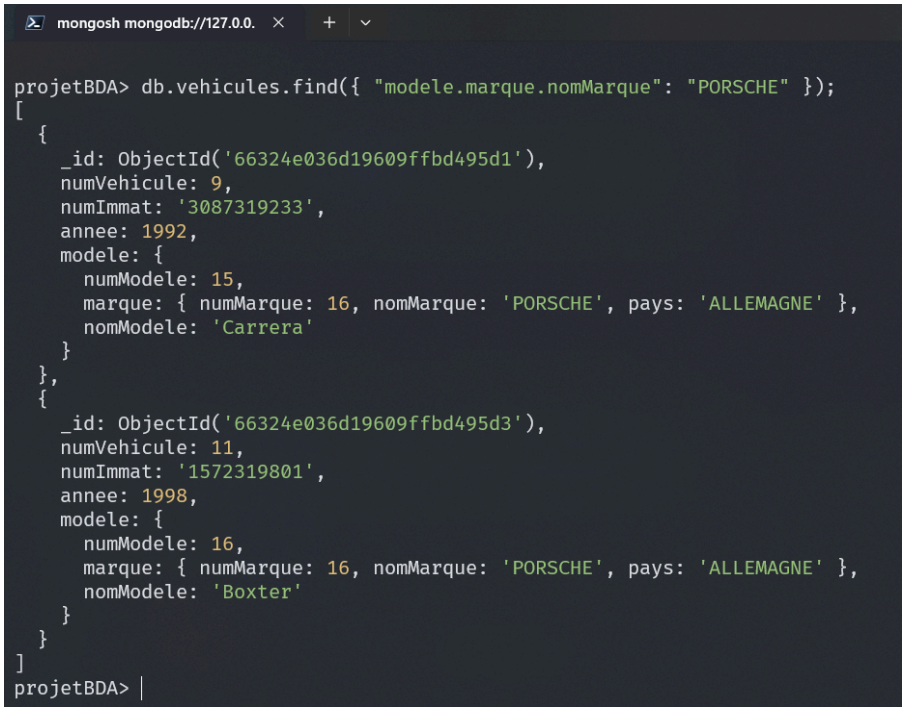
- Chaque document représente un véhicule avec ses informations (modèle, marque, immatriculation, année) et la liste de ses interventions.
- Chaque intervention possede ses informations ainsi que les employé ayant participé dans cette intervention.
- Pour chaque intervention, on stocke les détails (numéro, type, dates, coût) ainsi que les employés intervenant avec leurs informations.
- Cette modélisation permet de récupérer efficacement toutes les informations relatives à un véhicule et ses interventions, et leur employés en une seule requête.
- Les requêtes courantes comme lister les véhicules d'une marque donnée, récupérer les interventions d'un véhicule, etc. seront très performantes, car l'on evite les jointure.

B- Remplir la collection (Script mongodb.js)

C- Répondre aux requetes:

C. 1 Afficher tous les véhicules de la marque «PORSCHE»

```
db.vehicules.find({ "modele.marque.nomMarque": "PORSCHE" });
```



C. 2 Récupérer dans une nouvelle collection Véhicules_Interventions, les matricules des véhicules et le nombre total des interventions par véhicule ; la collection devra être ordonnée par ordre décroissant du nombre des interventions.

```
db.vehicules.aggregate([
  {
    $match: {
      interventions: { $exists: true } //pour recuperer les véhicules qui ont des interventions
    }
  },
  {
    $project: {
      numImmat: 1, // pour recuperer le matricule des véhicules
      nbInterventions: { $size: "$interventions" } // pour recuperer le nombre total des interventions
    }
  },
  {
    $sort: { nbInterventions: -1 } //ordonner de maniere decroissante avec -1 du nombre des interventions
  },
  {
```

```
    $out: "vehicules_interventions" //nom de la nouvelle collection
  },
]);
```

C. 3 Dans une collection véhicule_bcp_pannes, récupérer les véhicules dont le nombre des interventions dépasse 6 pannes.

```
db.vehicules.aggregate([
  {
    $match: {
      "interventions": { $exists: true },
      $expr: { $gt: [{ $size: "$interventions" }, 6] }
    }
  },
```

```
  {
    $out: "vehicule_bcp_pannes"
  }
]);  //il n'as pas de véhicules avec plus de 6 pannes, collection vide
```

```
projetBDA> db.vehicule_bcp_pannes.countDocuments()
0
projetBDA>
```

C. 4 Récupérer dans une collection employe-interv, toutes les interventions d'un employé.

```
db.vehicules.aggregate([
  {
    $unwind: "$interventions" // Décompose le tableau "interventions" pour
avoir un document par intervention
  },
  {
    $unwind: "$interventions.employes" // Décompose le tableau "employes" à
l'intérieur de chaque intervention pour avoir un document par employé
  },
  {
    $group: { // Regroupe les documents par numéro d'employé
      _id: "$interventions.employes.numEmploye", // Clé de regroupement basée
sur le numéro d'employé
      nbInterventions: { $sum: 1 },// Compte le nombre d'interventions pour
chaque employé
      interventions: { $push: "$interventions" }, // Ajoute chaque
intervention à un tableau pour chaque employé
    }
  },
  {
    $project: { // Restructure les documents de sortie
      _id: 0, // Supprime le champ "_id" par défaut
      numEmploye: "$_id", // Crée un champ "numEmploye" à partir de la clé de
regroupement "_id"
      nbInterventions: 1, // Conserve le champ "nbInterventions"
      interventions: 1, // Conserve le champ "interventions"
    }
  },
  {
```

```
    $sort: { numEmploye: 1 } //ordonner les numéro d'employé par ordre
croissant
  },
  {
    $out: "employe-interv" // Écrit les résultats dans la collection
"employe-interv"
  }
]);
```

C. 5 Augmenter de 8000DA, le salaire des employés de catégorie « Mécanicien»

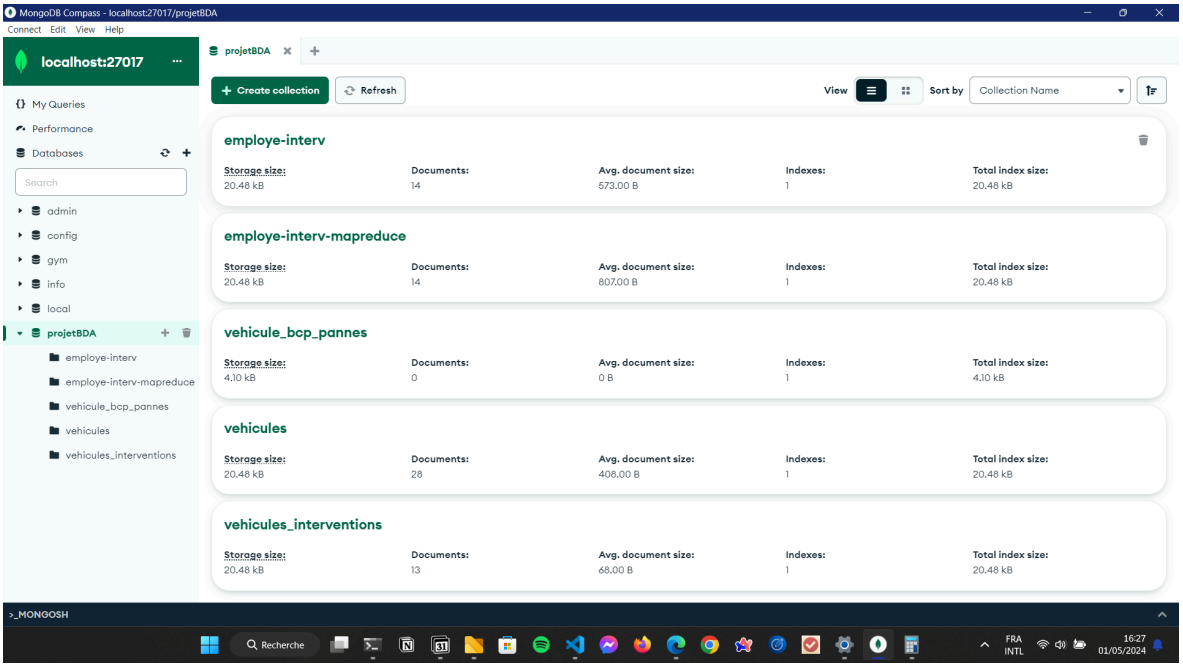
```
db.vehicules.updateMany(
  { "interventions.employes.categorie": "Mécanicien" },
  { $inc: { "interventions.$[].employes.$[emp].salaire": 8000 } },
  { arrayFilters: [{ "emp.categorie": "Mécanicien" }] }
);
```

C. 6 Reprendre la 4ième requête à l'aide du paradigme Map-Reduce.

```
//1. fontion map pour générer des documents clés-valeurs pour les transmettre
en entrée à Reduce.
var map = function() {
  if (this.interventions) { //parce que ce n'est pas tout les vehicules qui
ont des interventions
      this.interventions.forEach(function(intervention) {
          intervention.employes.forEach(function(employe) {
              emit(employe.numEmploye, intervention);
          });
      });
  }
};

//2. pour retourne le résultat agrégé à partir de ces documents en entrée.
numEmploye: key, interventions: value
var reduce = function(numEmploye, interventions) {
  return interventions;
};

// la collection dans laquelle le résultat sera placé
db.vehicules.mapReduce(
```

```
  map,
  reduce,
  {
      out: "employe-interv-mapreduce"
  }
);
```

Au final on se retrouve avec ces 5 collections:



D. Analyse

Donnez votre analyse par rapport à ces requêtes:

Dans les requêtes utilisé précèdement, nous pouvons constater les points clés suivants:

- **Performance** : Des requetes performantes et efficaces grâce à l'adoption d'une bonne structure de document, ainsi que Les mises à jour ciblées avec `arrayFilters` devraient être plus performantes que les mises à jour globales sur tous les documents.
- **Scalabilité** : La capacité du système à gérer une grande quantité de données doit être évaluée. Une modélisation et une implémentation qui permettent une évolutivité facile sont préférables.

- **Maintenabilité** : Une conception qui facilite la maintenance et l'ajout de nouvelles fonctionnalités est importante pour assurer la durabilité du système à long terme.
- **Flexibilité** : La capacité du système à s'adapter aux besoins changeants grâce à la structure flexible de la collection véhicules ainsi que des autres collections.

En résumé, l’utilisation du NoSQL avec une base de données non relationnel tel que mongodb offre une meilleur fléxibilité en ce qui concernce les documents, ainsi qu’une meilleur performance des requetes et une mailleurs scalabilité comparé au relationnel en SQL3, néanmoins, la complexité des requetes est reduite par rapport au SQL3.