



République Algérienne Démocratique et Populaire Ministère de l'Enseignement

Supérieur et de la Recherche Scientifique

Université des Sciences et de la Technologie Houari Boumediene

Faculté d'Informatique

Département IA & SD

Filière: Informatique

Spécialité: Master1 SII

# **RAPPORT DE PROJET TP BDA PARTIE 1-2**

**Chargé de COURS : BOUKHEDOUMA SAIDA** 

<u>Chargé du TP :</u> KESSI KAHINA

Noms et Prénoms: BENKOUITEN - AYMEN - 191931046409

**KENAI - IMAD EDDINE** - 191932017671

**Section et Groupe:** Master1 Systèmes Informatiques Intelligents 'SII' G-3

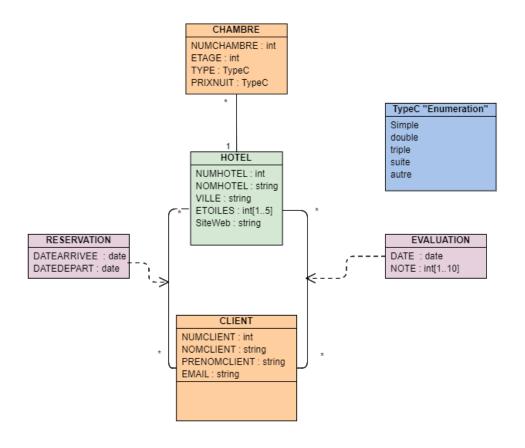
Année de scolarité : 2022/2023

# **PARTIE 1**

# 1. Partie I: Modélisation orientée objet

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

# **Réponse:**



# 2. Partie II : Création des TablesSpaces et utilisateur

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

# Réponse:

```
CREATE TABLESPACE SQL3_TBS DATAFILE 'C:\SQL3_TBS.dat'SIZE 100M AUTOEXTEND ON;

CREATE TEMPORARY TABLESPACE SQL3_TempTBS TEMPFILE 'C:\SQL3_TempTBS.dat' SIZE 100M AUTOEXTEND on;
```

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

# Réponse:

```
CREATE USER SQL3 IDENTIFIED BY psw
DEFAULT TABLESPACE SQL3_TBS
TEMPORARY TABLESPACE SQL3_TempTBS
QUOTA UNLIMITED ON SQL3_TBS;
```

4. Donner tous les privilèges à cet utilisateur

# **Réponse:**

```
GRANT ALL PRIVILEGES TO SQL3;
```

```
SQL> CREATE USER SOL3 IDENTIFIED BY psw

2 DEFAULT TABLESPACE SQL3_TBS

3 TEMPORARY TABLESPACE SQL3_TempTBS

4 QUOTA UNLIMITED ON SQL3_TBS;

User created.

SQL> GRANT ALL PRIVILEGES TO SQL3;

Grant succeeded.
```

### Etat après la création d'user « SQL3 » :

# 3. Partie III : 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.

# **Réponse:**

```
/*connexion avec SQL3*/
connect SQL3/psw.
-- création des types nécessaires :
```

J'ai fait une capture d'écran juste pour la création des deux types « Reservation\_Type, Evaluation\_Type » mais les autres types sont bien crées.

```
SQL> connect SQL3A/psw;
Connected.
SQL>
SQL>
SQL> create type Hotel_Type;
2 /
Type created.

SQL> create type Chambre_Type;
2 /
Type created.

SQL> create type Client_Type;
2 /
Type created.

SQL> create type Reservation_Type;
2 /
Type created.

SQL> create type Reservation_Type;
2 /
Type created.

SQL> create type Evaluation_Type;
2 /
Type created.
```

```
SQL> create type t_set_ref_Hotel as table of ref Hotel_Type;

2 /

Type created.

SQL> SQL> create type t_set_ref_Chambre as table of ref Chambre_Type;

2 /

Type created.

SQL> SQL> create type t_set_ref_Client as table of ref Client_Type;

2 /

Type created.

SQL> SQL> create type t_set_ref_Reservation as table of ref Reservation_Type;

2 /

Type created.

SQL> SQL> create type t_set_ref_Reservation as table of ref Evaluation_Type;

2 /

Type created.

SQL> SQL> create type t_set_ref_Evaluation as table of ref Evaluation_Type;

2 /

Type created.
```

Création de type incomplet « Hotel\_Type »

```
SQL> CREATE OR REPLACE Type Hotel_Type AS OBJECT (
         NumHotel INTEGER,
 2
         NomHotel VARCHAR(50),
 3
         Ville VARCHAR(50),
 5
         Etoiles INTEGER,
         SiteWeb VARCHAR(100),
 6
    HotelChambre t_set_ref_Chambre,
 8
    HotelClient t_set_ref_Client,
 9
    HotelEvaluation t set ref Evaluation
10
11
     );
12
ype created.
```

### Création de type incomplet « Chambre\_Type » :

```
SQL>
SQL> -------Creation du type Chambre
SQL>
SQL> CREATE OR REPLACE Type Chambre_Type AS OBJECT (

NumChambre INTEGER,

NumHotel INTEGER,

Etage INTEGER,

TypeChambre VARCHAR(50),

PrixNuit INTEGER,

ChambreHotel ref Hotel_Type , -- pour referencier la clé etrangere numHotel dans chambre

ChambreReservation t_set_ref_Reservation

);

10 /

Type created.
```

## Création de type incomplet « Client\_Type » :

```
SQL> -----Creation du type client
SQL>
SQL> CREATE OR REPLACE Type Client Type AS OBJECT (
 2
         NumClient INTEGER,
 3
         NomClient varchar2(50),
    PrenomClient varchar2(50),
 5
         Email VARCHAR(100),
 6
    ClientReservation t_set_ref_Reservation ,
    ClientEvaluation t_set_ref_Evaluation ,
 7
    ClientHotel t_set_ref_Hotel
 8
 9
 10
 11
    );
 12
Type created.
```

### Création de type incomplet « Reservation\_Type » :

```
SQL> CREATE OR REPLACE Type Reservation_Type AS OBJECT (
  2
         NumClient INTEGER,
  3
    NUMCHAMBRE INTEGER,
    NumHotel INTEGER,
  4
         DateDepart DATE,
    DateArrivee DATE,
  6
    ReservationClient ref Client_Type,
  8
    ReservationChambre ref Chambre_Type,
  9
    ReservationHotel ref Hotel_Type
10
     );
11
Type created.
```

### Création de type incomplet « Evaluation\_Type » :

```
SQL> create or replace type Evaluation_Type as object (
2 NumHotel INTEGER,
3 NumClient INTEGER,
4 DateEvaluation date,
5 note INTEGER,
6 EvaluationClient ref Client_Type,
7 EvaluationHotel ref Hotel_Type
8 );
9 /
Type created.
```

# Etat après création de toutes les types :

### **6.** <u>Définir les méthodes permettant de :</u>

• Calculer pour chaque client, le nombre de réservations effectuées.

```
alter type Client_Type add member function resClient return numeric cascade;

create or replace type body Client_Type
as member function resClient return numeric
is
    nbrRes number;
Begin
    Select cardinality(self.ClientReservation) into nbrRes
    From CLIENT;
    Return nbrRes;
End;
//
```

```
SQL> alter type Client_Type
 2 add member function resClient return numeric
 3 cascade;
Type altered.
SQL> --le corps
SQL> create or replace type body Client_Type
 2 as member function resClient return numeric 3 is
 4 nbrRes number;
 5 Begin
 6 Select cardinality(self.ClientReservation) into nbrRes
 7 From CLIENT;
 8 Return nbrRes ;
 9 End;
10 End;
11
Type body created.
```

• Calculer pour chaque hôtel, le nombre de chambres.

```
alter type Hotel_Type add member function nbCham return numeric cascade;

create or replace type body Hotel_Type
as member function nbCham return numeric
is
    nbrchambre number;
begin
    select cardinality(self.HotelChambre) into nbrchambre
    from HOTEL;
    return nbrchambre;
end;
end;
//
```

```
SQL> alter type Hotel_Type add member function nbCham return numeric cascade;
Type altered.
SQL>
SQL> create or replace type body Hotel_Type
 2 as member function nbCham return numeric
  3 is
 4
        nbrchambre number;
  5 begin
         select cardinality(self.HotelChambre) into nbrchambre
  6
        from HOTEL;
 8
         return nbrchambre;
 9 end;
 10
    end;
 11
Type body created.
```

<u>Calculer pour chaque chambre, son chiffre d'affaire.</u>

```
alter type Chambre_Type add MEMBER FUNCTION ChiffreAffaire RETURN numeric cascade;

CREATE OR REPLACE TYPE BODY Chambre_Type AS

MEMBER FUNCTION ChiffreAffaire RETURN numeric

AS

....v_chiffre_affaire number;

BEGIN

SELECT SUM(PrixNuit * (r.DateDepart - r.DateArrivee)) INTO v_chiffre_affaire
    FROM Reservation r, Chambre c

WHERE r.NumChambre = c.NumChambre AND c.NumChambre = self.NumChambre;
    RETURN v_chiffre_affaire;

END;

END;
//
```

```
SQL> CREATE OR REPLACE TYPE BODY Chambre Type AS
 2 MEMBER FUNCTION ChiffreAffaire RETURN numeric
 3 AS
        v_chiffre_affaire number;
    BEGIN
        SELECT SUM(PrixNuit * (r.DateDepart - r.DateArrivee)) INTO v_chiffre_affaire
        FROM Reservation r, Chambre c
 8
        WHERE r.NumChambre = c.NumChambre AND c.NumChambre = self.NumChambre;
 9
        RETURN v chiffre affaire;
10
   END;
11
    END;
12
Type body created.
```

• <u>Calcule r pour chaque hôtel, le nombre d'évaluations</u> reçues à une date donnée (01-01-2022).

```
alter type Hotel_Type add member function nbEval(p_dateEvaluation DATE) return numeric cascade;
CREATE OR REPLACE TYPE BODY Hotel Type AS
 MEMBER FUNCTION nbCham RETURN NUMERIC
   nbrchambre NUMBER;
 BEGIN
   SELECT COUNT(*) INTO nbrchambre
   FROM Chambre
   WHERE NumHotel = self.NumHotel;
   RETURN nbrchambre;
  END;
 MEMBER FUNCTION nbEval(p_dateEvaluation DATE) RETURN NUMERIC
   NbEvaluations NUMBER;
 BEGIN
   SELECT COUNT(*) INTO NbEvaluations
   FROM Evaluation e
   WHERE e.NumHotel = self.NumHotel AND e.DateEvaluation = p_dateEvaluation;
   RETURN NbEvaluations;
 END:
END;
```

```
SQL> CREATE OR REPLACE TYPE BODY Hotel Type AS
       MEMBER FUNCTION nbCham RETURN NUMERIC
  2
  3
       IS
  4
         nbrchambre NUMBER;
  5
       BEGIN
         SELECT COUNT(*) INTO nbrchambre
  6
  7
         FROM Chambre
         WHERE NumHotel = self.NumHotel;
  8
  9
         RETURN nbrchambre;
 10
      END;
 11
       MEMBER FUNCTION nbEval(p dateEvaluation DATE) RETURN NUMERIC
 12
 13
        NbEvaluations NUMBER;
 14
 15
       BEGIN
         SELECT COUNT(*) INTO NbEvaluations
 16
 17
         FROM Evaluation e
         WHERE e.NumHotel = self.NumHotel AND e.DateEvaluation = p_dateEvaluation;
 18
 19
         RETURN NbEvaluations;
 20
      END;
 21
    END;
 22
Type body created.
```

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

### Création de la table « HOTEL »:

### Création de la table « CHAMBRE » :

```
SQL>
SQL> CREATE TABLE Chambre OF Chambre_Type (
2 constraint pk_Chambre primary key(NumChambre, NumHotel),
3 constraint fk_Chambre1 foreign key(ChambreHotel) references Hotel,
4 CONSTRAINT CHK_TypeCHAMBRE CHECK (TypeChambre IN ('simple', 'double', 'triple', 'suite', 'autre'))
5 )
6 nested table ChambreReservation store as table_ChambreReservation;
Table created.
```

### Création de la table « CLIENT » :

```
SQL> CREATE TABLE CLIENT OF Client_Type (
2 constraint pk_Client primary key(NumClient)
3 )
4 nested table ClientReservation store as table_ClientReservation,
5 nested table ClientEvaluation store as table_ClientEvaluation,
6 nested table ClientHotel store as table_ClientHotel;
Table created.
```

### Création de la table « RESERVATION » :

## Création de la table « EVALUATION » :

# Etat après création de toutes les types :

```
SQL> SELECT table_name FROM user_tables;

TABLE_NAME
TABLE_HOTELCHAMBRE
TABLE_HOTELCLIENT
TABLE_HOTELEVALUATION
TABLE_CHAMBRERESERVATION
TABLE_CLIENTRESERVATION
TABLE_CLIENTRESERVATION
TABLE_CLIENTEVALUATION
TABLE_CLIENTEVALUATION
TABLE_CLIENTHOTEL

7 rows selected.
```

# 5. Partie VI : Langage de manipulation de données

8. Remplir toutes les tables par les instances fournies en annexe (compléter les données d'évaluation).

#### Insertion dans la table « Hotel »:

#### Exécution:

```
SQL> INSERT INTO Hotel values (Hotel_Type (1, 'Renaissance ', 'Tlemcen', 5,null,t_set_ref_Chambre(),t_set_ref_Client(),t_set_ref_Evaluation()));

1 row created.
```

### Insertion dans la table « Chambre »:

#### Exécution:

```
SQL>
SQL> INSERT INTO chambre VALUES (Chambre_Type(1, 4, 0, 'autre', 13000,(select ref(a) from HOTEL a where NumHotel=4),t_s
et_ref_Reservation()));
1 row created.
```

#### Insertion dans la table « Client »:

#### Exécution :

```
SQL> INSERT INTO CLIENT VALUES (Client_Type (1, 'BOUROUBI', 'Taous', null,t_set_ref_Reservation(),t_set_ref_Evaluation(),t_set_ref_Hotel()));
1 row created.
```

#### Insertion dans la table « Reservation »:

#### Exécution:

```
SQL>
SQL> INSERT INTO reservation VALUES (Reservation_Type(1, 1, 5, '2022-05-15', '2022-05-11', (select ref(a) from Client a where NumClient=1 and rownum = 1),(select ref(a) from Hotel a where NumChambre=1 and rownum = 1),(select ref(a) from Hotel a where NumHotel=5 and rownum = 1)));

1 row created.
```

### Insertion dans la table « Evaluation » :

### Exécution :

```
SQL> INSERT INTO Evaluation VALUES (Evaluation_Type(5, 1, '2022-05-15', 4,(select ref(a) from Client a where NumClient=1),(select ref(b) from Hotel b where NumHotel=5)));

1 row created.

SQL> INSERT INTO Evaluation values (Evaluation_Type(5, 2, '2022-04-18', 3,(select ref(a) from Client a where NumClient=2),(select ref(b) from Hotel b where NumHotel=5)));

1 row created.
```

```
Insertion dans la table imbriquée « HotelClient » :
```

#### Exécution :

```
5QL> insert into table (select l.HotelClient from Hotel l where NUMHOTEL=5)
2 (select ref(c) from CLIENT c where NUMCLIENT=1);
1 row created.
5QL> insert into table (select l.HotelClient from Hotel l where NUMHOTEL=5)
```

#### Insertion dans la table imbriquée « ClientReservation » :

### Exécution :

```
SQL>
SQL>
insert into table (select l.ClientReservation from CLIENT l where NUMCLIENT=1)
2 (select ref(c) from RESERVATION c where NUMCLIENT=1);

1 row created.

SQL> insert into table (select l.ClientReservation from CLIENT l where NUMCLIENT=2)
2 (select ref(c) from RESERVATION c where NUMCLIENT=2);

1 row created.
```

#### Insertion dans la table imbriquée « HotelChambre » :

#### Exécution

```
SQL>
SQL> insert into table (select 1.HotelChambre from hotel 1 where NUMHOTEL=2)
2 (select ref(c) from chambre c where NUMCHAMBRE=1);
5 rows created.

SQL>
SQL> insert into table (select 1.HotelChambre from hotel 1 where NUMHOTEL=4)
2 (select ref(c) from chambre c where NUMCHAMBRE=1);
5 rows created.
```

#### Insertion dans la table imbriquée « HotelEvaluation » :

#### Exécution:

```
SQL>
SQL> insert into table (select 1.HotelEvaluation from hotel 1 where NUMHOTEL=1)
2 (select ref(c) from Evaluation c where NUMHOTEL=1);

1 row created.

SQL> insert into table (select 1.HotelEvaluation from hotel 1 where NUMHOTEL=2)
2 (select ref(c) from Evaluation c where NUMHOTEL=2);

1 row created.
```

# 6. Partie V : Langage d'interrogation de données

**9.** Lister les noms d'hôtels et leurs villes respectives.

```
SELECT CONCAT (NomHotel, Ville) FROM Hotel;
```

```
SQL> SELECT CONCAT (NomHotel, Ville) FROM Hotel;

CONCAT(NOMHOTEL,VILLE)

Renaissance Tlemcen
Seybouse Annaba
Hôtel Novotel Constantine
Saint George d'AlgerAlger
Ibis Alger AéroportAlger
El Mountazah AnnabaAnnaba
Hôtel Albert 1erAlger
Chems Oran
Colombe Oran
Mercure Alger
Le Méridien Oran

CONCAT(NOMHOTEL,VILLE)

Hôtel Sofitel Alger

12 rows selected.
```

**10.** Lister les hôtels sur lesquels porte au moins une réservation.

```
select H.NOMHOTEL
from hotel H, table(H.HotelClient) c
group by H.NOMHOTEL;
```

```
SQL> Select H.NOMHOTEL

2 from hotel H, table(H.HotelClient) c

3 group by H.NOMHOTEL;

NOMHOTEL

Seybouse
Le Méridien
Colombe
Chems
Saint George d'Alger
Hôtel Albert 1er
Renaissance
Mercure
Ibis Alger Aéroport
Hôtel Sofitel
El Mountazah Annaba

11 rows selected.
```

11. Quels sont les clients qui ont toujours séjourné au premier étage ?

```
SELECT C.NumClient, C.NomClient, C.PrenomClient
                        FROM Client C
                        WHERE NOT EXISTS (
                              SELECT *
                              FROM Reservation R
                              INNER JOIN Chambre CH ON R.ReservationChambre = REF(CH)
                              WHERE R.ReservationClient = REF(C) AND CH.Etage != 1
                        );
                                                                                                                                        NUMCLIENT NOMCLIENT
                                                     NUMCLIENT NOMCLIENT NUMCLIENT NOMCLIENT
                                                                                                NUMCLIENT NOMCLIENT NUMCLIENT NOMCLIENT
                                                                                                                                       -PRENOMCLIENT
                                                                                               - PRENOMCLIENT
                                                                                                                PRENOMCLIENT
                                                     PRENOMCLIENT
                               PRENOMCLIENT
    SELECT C.NUMCLIERC, C.NOMC
FROM Client C
WHERE NOT EXISTS (
SELECT F
FROM Reservation R
INNER JOIN Chambre CH ON
WHERE R.ReservationClien
                                                                         __PRENOMCLIENT
                                                                                               41 BELKACEMI
Hocine
                                                    20 BENOUADAH
Mohammed
                                10 ABAYAHIA
Abdelkader
                                                                                                                          52 RAHALI
                                                                                 30 BAHBOUH
                                                                          Naima
                               13 ABBOU
Mohamed
                                                                                                                                              63 CHALABI
                                                                                                                           54 TERKI
                                                                                 32 BAKIR
                                                                                                                                        Mourad
                                                                                                                    Amina
                                                                                               NUMCLIENT NOMCLIENT NUMCLIENT NOMCLIENT NUMCLIENT NOMCLIENT
                               NUMCLIENT NOMCLIENT
                                                     NUMCLIENT NOMCLIENT
        2 BOUZIDI
                                                                           NUMCLIENT NOMCLIENT
                                PRENOMCLIENT
                                                     PRENOMCLIENT
                                                                                                                   PRENOMCLIENT
                                                                                                                                        PRENOMCLIENT
                                                                                               PRENOMCLIENT
                                                                          PRENOMCLIENT
                                14 ABDELAZIZ
                                                                          33 BALI
Malika
                                                                                                                   55 CHAOUI
Farid
        4 BOUCHEMLA
                                                                                                                                       64 MOHAMMEDI
Mustapha
Flias
                                                    24 AISSAT
Salima
ouhir
                                                                                                                          57 CHAKER
                                                                                                                                               66 SAIDOUNI
                                                                                                                   Nadia
PRENOMCLIENT
                                                                                                NUMCLIENT NOMCLIENT NUMCLIENT NOMCLIENT
                                NUMCLIENT NOMCLIENT
                                                     NUMCLIENT NOMCLIENT
                                                                           NUMCLIENT NOMCLIENT
                                                                                                                                        NUMCLIENT NOMCLIENT
Hakim
                                                                                               PRENOMCLIENT PRENOMCLIENT
                                PRENOMCLIENT
                                                     PRENOMCLIENT
                                                                          PRENOMCLIENT
                               17 ABDEMEZIANE
Madjid
8 ABAD
bdelhamid
                                                                                                      49 BELHAMIDI Fatima
a
                                                                                                                                               67 YALAOUI
                                                                          38 BELAKERMI
Mohammed
                                                                                                      59 IGOUDJIL
50 BELKACEMI Redouane
                                                                                                                                               68 AYATA
                                                    28 BABACI
Mourad
                                                                                                      51 BELKOUT Lakhdar
                                                                                 39 BELGHALI
                                                                                                                                              69 TEBIBEL
                                                                                                                                        Nabila
```

```
Nabila
60 rows selected.
SQL>
```

12. Quels sont les hôtels (nom, ville) qui offrent des suites ? et donner le prix pour chaque suite

```
select h.NOMHOTEL, h.VILLE
from hotel h, table(h.HotelChambre) ch
where deref(value(ch)).TYPECHAMBRE='suite';
```

```
SQL> select h.NOMHOTEL, h.VILLE

2 from hotel h, table(h.HotelChambre) ch

3 where deref(value(ch)).TYPECHAMBRE='suite';

NOMHOTEL

VILLE

Hôtel Sofitel

Alger
```

**13.** Quel est le type de chambre le plus réservé habituellement, pour chaque hôtel d'Alger ?

```
SELECT h.NomHotel, c.TypeChambre, COUNT(*) AS Nombre_de_reservations
FROM Reservation r

JOIN Chambre c ON r.ReservationChambre = REF(c)

JOIN Hotel h ON r.ReservationHotel = REF(h)

WHERE h.Ville = 'Alger'

GROUP BY h.NomHotel, c.TypeChambre

ORDER BY h.NomHotel, Nombre_de_reservations DESC;
```

```
SQL> SELECT h.NomHotel, c.TypeChambre, COUNT(*) AS Nombre_de_reservations
 2 FROM Reservation r
 3 JOIN Chambre c ON r.ReservationChambre = REF(c)
 4 JOIN Hotel h ON r.ReservationHotel = REF(h)
 5 WHERE h.Ville = 'Alger'
 6 GROUP BY h.NomHotel, c.TypeChambre
 7 ORDER BY h.NomHotel, Nombre_de_reservations DESC;
NOMHOTEL
TYPECHAMBRE
                                                  NOMBRE DE RESERVATIONS
Hôtel Sofitel
simple
                                                                       2
Ibis Alger Aéroport
simple
Ibis Alger Aéroport
autre
                                                                       2
```

**14.** Quels sont les hôtels (nom, ville) ayant obtenu une moyenne de notes >=6, durant l'année 2022

```
SQL> SELECT h.NOMHOTEL, h.VILLE

2 FROM hotel h, TABLE(h.HotelEvaluation) he

3 where EXTRACT(YEAR FROM deref(VALUE(he)).DateEvaluation) = 2022

4 GROUP BY h.NOMHOTEL, h.VILLE

5 HAVING AVG(deref(VALUE(he)).note) >= 6;

no rows selected
```

Ca c'est pour les hotels qui ont obtenu une moyenne de notes >=6, on va essayer pour >=5 :

```
SELECT h.NOMHOTEL, h.VILLE

FROM hotel h, TABLE(h.HotelEvaluation) he

where EXTRACT(YEAR FROM deref(VALUE(he)).DateEvaluation) = 2022

GROUP BY h.NOMHOTEL, h.VILLE

HAVING AVG(deref(VALUE(he)).note) >= 6;
```

### **15.** Quel est l'hôtel ayant réalisé le meilleur chiffre d'affaire durant l'été 2022

```
SELECT h.NomHotel, SUM(c.PrixNuit * (r.DateDepart -r. DateArrivee)) AS Chiffre_daffaire
FROM Reservation r
JOIN Chambre c ON r.ReservationChambre = REF(c)
JOIN Hotel h ON r.ReservationHotel = REF(h)
WHERE EXTRACT(MONTH FROM r.DateArrivee) IN (6, 7, 8)
GROUP BY h.NomHotel
ORDER BY Chiffre_daffaire DESC;
```

```
SQL> SELECT h.NomHotel, SUM(c.PrixNuit * (r.DateDepart -r. DateArrivee)) AS Chiffre_daffaire

2 FROM Reservation r

3 JOIN Chambre c ON r.ReservationChambre = REF(c)

4 JOIN Hotel h ON r.ReservationHotel = REF(h)

5 WHERE EXTRACT(MONTH FROM r.DateArrivee) IN (6, 7, 8)

6 GROUP BY h.NomHotel

7 ORDER BY Chiffre_daffaire DESC;

NOMHOTEL

CHIFFRE_DAFFAIRE

Ibis Alger Aéroport

39000
```

# **PARTIE 2**

# A- Modélisation:

```
<u>1-</u>
 "hotels": [
    "numHotel": 1,
    "nomHotel": "Hotel A",
    "ville": "Paris",
    "etoiles": 4,
    "siteWeb": "www.hotel.com",
    "chambres": [
       "numChambre": 1,
       "etage": 1,
       "typeChambre": "simple",
       "prixNuit": 100,
       "reservations": [
         "numClient": 1,
"dateArrivee": "2023-05-01",
         "dateDepart": "2023-05-05"
       "evaluations": [
         "numClient": 1,
         "date": "2023-06-01",
         "note": 8
       "numChambre": 2,
       "etage": 2,
       "typeChambre": "double",
       "prixNuit": 150,
       "reservations": [
         "numClient": 2,
         "dateArrivee": "2023-05-02", "dateDepart": "2023-05-06"
       evaluations": [
         "numClient": 2,
         "date": "2023-06-02",
         "note": 9
```

```
"numHotel": 2,
  "nomHotel": "Hotel B",
  "ville": "Lyon",
  "etoiles": 3,
  "siteWeb": "www.hotelb.com",
  "chambres": [
     "numChambre": 3,
     "etage": 1,
     "typeChambre": "simple",
     "prixNuit": 80,
"reservations": [],
     "evaluations": []
   },
     "numChambre": 4,
     "etage": 2,
     "typeChambre": "double",
     "prixNuit": 120,
     "reservations": [],
     "evaluations": []
"clients": [
{
"numClient": 1,
"Client": "D
  "nomClient": "Dupont",
  "prenomClient": "Jean",
  "email": "jean.dupont@gmail.com"
  "numClient": 2,
  "nomClient": "Martin",
  "prenomClient": "Marie",
  "email": "marie.martin@gmail.com"
```

<u>2- Voici un exemple d'un document de la collection "Hotels" qui illustre la modélisation orientée</u> document proposée ci-dessus :

```
"hotels": [
  "numHotel": 1,
  "nomHotel": "Sheraton",
  "ville": "Alger",
  "etoiles": 4,
  "siteWeb": "www.Sheraton.dz",
  "chambres": [
     "numChambre": 1,
     "etage": 1,
     "typeChambre": "simple",
     "prixNuit": 10000,
     "reservations": [
        "numClient": 1,
       "dateArrivee": "2023-05-01",
       "dateDepart": "2023-05-05"
     "evaluations": [
        "numClient": 1,
       "date": "2023-06-01",
       "note": 8
    ]
```

### <u>3-</u>

Dans cette modélisation, chaque document correspond à un hôtel, une chambre ou un client. Les chambres et les clients sont imbriqués dans les documents des hôtels correspondants, ce qui permet d'avoir toutes les informations nécessaires pour chaque objet en une seule requête.

Dans chaque document d'hôtel, on trouve les attributs du schéma relationnel HOTEL, ainsi qu'un tableau de chambres et un tableau de clients. Chaque document de chambre contient les attributs de la relation chambre, ainsi que des tableaux de réservations et d'évaluations. Chaque document de client contient les attributs de la relation CLIENT.

# Remarque:

Pour le reste du TP on va travailler avec une autre modélisation plus simple qui représente chacun des Objets, Hôtel, Client, Chambre, réservation ou évaluation dans une collection à part.

Ça va nous permettre d'exploiter l'opérateur **\$lookup** qui a été introduit dans les versions >> mongodb 3.0

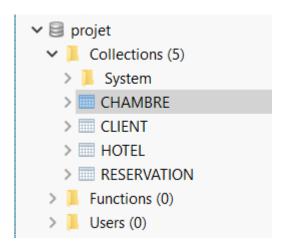
```
- Pour la collection "Hotels"
    "_id": 0,
     "numhotel": 000,
     "nomhotel": "Nom de l'hôtel",
     "ville": "Ville de l'hôtel",
     "etoiles": 3,
     "siteweb": "URL du site web de l'hôtel",
     "evaluations": [
       {
          "numclient": 1,
          "date": 2023-05-04,
          "note": 8
          "numclient": 2,
          "date":2023-05-01,
          "note": 9
    ]
  }
- Pour la collection "Chambres",
    "_id": 00,
     "numchambre": 101,
     "numhotel": 123,
     "etage": 1,
     "typechambre": "double",
     "prixnuit": 1000
  }
- Pour la collection "Clients", on peut utiliser les champs suivants :
     "_id": 0
     "numclient": 1,
     "nomclient": "Nom du client",
     "prenomclient": "Prénom du client",
     "email": "Adresse email du client"
  }
- Pour la collection "Reservations", on peut utiliser les champs suivants :
  {
     "_id": 1,
     "numclient": 1,
     "numhotel": 123,
     "numchambre": 101,
     "datearrivee":2023-05-01,
     "datedepart": 2023-05-03
  }
```

### **B-Insertion:**

Pour l'insertion, on va utiliser la commande LOAD() du mongodb qui permet d'exécuter un fichier.js qui contient plusieurs commandes

```
| db.HOTEL.insertOne({ "NUMHOTEL": 1,"NOMHOTEL": "Renaissance", "VILLE": "Tlemcen","ETOILES": A db.HOTEL.insertOne({ "NUMHOTEL": 2,"NOMHOTEL": "Seybouse", "VILLE": "Annaba","ETOILES": 3 })  
| db.HOTEL.insertOne({ "NUMHOTEL": 3,"NOMHOTEL": "HotelNovotel", "VILLE": "Constantine","ETOILI  
| db.HOTEL.insertOne({ "NUMHOTEL": 4,"NOMHOTEL": "Saint George d'Alger", "VILLE": "Alger","ETOIL  
| db.HOTEL.insertOne({ "NUMHOTEL": 5,"NOMHOTEL": "Ibis Alger Aeroport", "VILLE": "Alger","ETOIL  
| db.HOTEL.insertOne({ "NUMHOTEL": 6,"NOMHOTEL": "ElMountazah Annaba", "VILLE": "Annaba","ETOIL  
| db.HOTEL.insertOne({ "NUMHOTEL": 7,"NOMHOTEL": "Hote lalbert ler", "VILLE": "Alger","ETOIL  
| db.HOTEL.insertOne({ "NUMHOTEL": 7,"NOMHOTEL": "Colombe", "VILLE": "Oran","ETOILES": 2 })  
| db.HOTEL.insertOne({ "NUMHOTEL": 9,"NOMHOTEL": "Colombe", "VILLE": "Oran","ETOILES": 2 })  
| db.HOTEL.insertOne({ "NUMHOTEL": 10,"NOMHOTEL": "Mercure", "VILLE": "Oran","ETOILES": 3 })  
| db.HOTEL.insertOne({ "NUMHOTEL": 12,"NOMHOTEL": "Le Meridien", "VILLE": "Oran","ETOILES": 5 }  
| db.HOTEL.insertOne({ "NUMHOTEL": 12,"NOMHOTEL": "Le Meridien", "VILLE": "Alger","ETOILES": 5 }  
| db.CIIENT.insertOne({ "NUMHOTEL": 1,"NOMHOTEL": "BOUROUBI","PRENOMCLIENT": "Rause"})  
| db.CIIENT.insertOne({ "NUMCLIENT": 2,"NOMCLIENT": "BOUROUBI","PRENOMCLIENT": "BOURIENT": "BOURIENT": "BOURIENT": "BOURIENT": "Bourien"}  
| db.CIIENT.insertOne({ "NUMCLIENT": 3,"NOMCLIENT": "LACHEMI","PRENOMCLIENT": "Bourien"}  
| db.CIIENT.insertOne({ "NUMCLIENT": 3,"NOMCLIENT": "BOURCHENT": "Bourche
```

```
> use projet
switched to db projet
> load("D:/M1 SII/BDA/miniprojet2/insertion.js")
true
```



# C- requêtes:

-Afficher tous les hôtels classés « 3 étoiles » :

```
db.HOTEL.find({ETOILES: 3})
" id" : ObjectId("645441312b4861248b7be263"),
   "NUMHOTEL" : 2.0,
   "NOMHOTEL" : "Seybouse",
   "VILLE" : "Annaba",
   "ETOILES": 3.0
   " id" : ObjectId("645441312b4861248b7be267"),
   "NUMHOTEL" : 6.0,
   "NOMHOTEL" : "ElMountazah Annaba",
   "VILLE" : "Annaba",
   "ETOILES" : 3.0
   " id" : ObjectId("645441312b4861248b7be268"),
   "NUMHOTEL" : 7.0,
   "NOMHOTEL" : "Hote lAlbert ler",
   "VILLE" : "Alger",
   "ETOILES" : 3.0
   " id" : ObjectId("645441312b4861248b7be26a"),
   "NUMHOTEL" : 9.0,
   "NOMHOTEL" : "Colombe",
   "VILLE" : "Oran",
   "ETOILES" : 3.0
```

-Récupérer dans une nouvelle collection Hotels-NbResv, les noms des hôtels et le nombre total de réservations par hôtel ; la collection devra être ordonnée par ordre décroissant du nombre de réservations. Afficher le contenu de la collection.

\$lookup: faire une jointure entre RESERVATION et HOTEL avec l'attribut NUMHOTEL

\$unwind : éclater le résultat de la jointure en plusieurs documents.

\$group: sommer pour chaque hotel

\$sort : -1 pour ordre descendant

\$out : résultat sera dans la collection HOTELSNBRESV.



- Dans une collection HotelsPas-cher, récupérer les hôtels dont le prix des chambres ne dépasse pas 6000 DA. Afficher le contenu de la collection.

\$project : faire passer que le nom de l'hotel

- Afficher tous les noms d'hôtels ayant obtenu une note moyenne >= 5

- Afficher toutes les réservations d'un client donné (on donnera l'e-mail du client). On affichera le nom de l'hôtel, le numéro de chambre et la date d'arrivée.

- Afficher toutes les évaluations postées par un client donné (on donnera l'e-mail du client). On affichera le nom de l'hôtel, la date d'évaluation, la note.

- Augmenter de 2000DA, le prix unitaire de toutes les chambres des hôtels classés « 5 étoiles »

```
() 0.002 sec.
/* 1 */
{
    "acknowledged" : true,
    "matchedCount" : 66.0,
    "modifiedCount" : 66.0
}
```

On récupère dans hôtels les NUMHOTEL distincts pour les hotels qui ont 5 étoiles.

On utilise après la méthode UPDATEMANY qui permet de faire des mise a jours sur plusieurs documents au même temps.

- Reprendre la 2ème requête à l'aide du paradigme Map-Reduce :

```
var map = function() {emit(this.NUMHOTEL, 1);};

var reduce = function(key, values) {
  return Array.sum(values);};

var finalizeFunction = function(key, reducedValue) {
  return { hotel: key, nb_reservations: reducedValue };
};

db.RESERVATION.mapReduce(map,reduce,{
   out: "Hotels_NbResv",
   finalize: finalizeFunction
}
);
```

# D- Analyse:

# Explorons les deux approches :

## 1- Collections séparées :

## - Avantages :

- Offre la flexibilité de consulter et de mettre à jour des objets individuellement.
- Prise en charge de la scalabilité et des performances lors du traitement de grandes collections.
- Permet des indexations et des requêtes plus faciles sur des champs spécifiques.

### - Inconvénients :

- Nécessite plusieurs requêtes ou jointures pour récupérer des données liées.
- Complexité accrue lors de la gestion des relations entre les objets.
- Peut entraîner des données redondantes si plusieurs objets partagent des attributs communs.

### 2- Objets imbriqués :

### - Avantages :

- Regroupe des données liées dans un seul document, réduisant ainsi le besoin de jointures.
- Simplifie la récupération de hiérarchies d'objets complets.
- Peut améliorer les performances en lecture, notamment pour les charges de travail orientées lecture.

### - Inconvénients :

- Flexibilité limitée lors de la consultation et de la mise à jour d'objets spécifiques imbriqués.
- Complexité accrue lors de la modification ou de la mise à jour de structures imbriquées.
- Les grands tableaux imbriqués peuvent avoir un impact sur les performances en écriture et les limites de taille des documents.

Conclusion
Si on se base seulement sur le fait que la plupart des requêtes vont porter sur les hôtels, alors la modélisation en utilisant les objets imbriqués est meilleure.
Par contre si on prend cette base de données dans une situation réel ça ne sera pas vraiment efficace d'utiliser cette modélisation.
Une solution serait d'exploiter les avantages des 02 solutions en adoptant une modélisation hybride.