

```
-- LOI ALESSIA - GROUPE 6 - 3971668
-- DEVOIR MAISON 2eme PARTIE - SGBD DE LA GESTION D'UN HOTEL (H2 Console)
-- Ce document peut être entièrement copié et collé dans H2
-----
```

```
-- Creation des tables
```

```
CREATE TABLE METIER(
metier VARCHAR(255) PRIMARY KEY,
heures_max_mois NUMBER(10,2), salaire_mois NUMBER(10,2) ) ;
```

```
CREATE TABLE PERSONNE(
id VARCHAR(255) PRIMARY KEY,
nom VARCHAR(255), prenom VARCHAR(255), date_naiss DATE, code_postal NUMBER(10),
mail VARCHAR(255), telephone NUMBER(10) ) ;
```

```
CREATE TABLE CLIENT(
id VARCHAR(255) PRIMARY KEY, carte_credit VARCHAR(255),
FOREIGN KEY (id) REFERENCES PERSONNE(id) );
```

```
CREATE TABLE EMPLOYE(
id VARCHAR(255) PRIMARY KEY, nPieceId VARCHAR(255), nSECU VARCHAR(255), IBAN
VARCHAR(255), metier VARCHAR(255), chef VARCHAR(255),
FOREIGN KEY (id) REFERENCES PERSONNE(id),
FOREIGN KEY (metier) REFERENCES METIER(metier),
FOREIGN KEY (chef) REFERENCES PERSONNE(id) ) ;
```

```
CREATE TABLE CATEGORIE(
codeC VARCHAR(255) PRIMARY KEY, description VARCHAR(255), nbmaxoccupants
NUMBER(10), tarif_TTC NUMBER(10,2) ) ;
```

```
CREATE TABLE ETAGE(
nEtage NUMBER(10) PRIMARY KEY,
qChambres VARCHAR(255), idFDC VARCHAR(255),
FOREIGN KEY (idFDC) REFERENCES PERSONNE(id) );
```

```
CREATE TABLE CHAMBRE(
nEtage NUMBER(10) NOT NULL, nChambre NUMBER(10) NOT NULL,
dimension NUMBER(10), description VARCHAR(255), codeC VARCHAR(255),
PRIMARY KEY (nEtage, nChambre),
FOREIGN KEY (nEtage) REFERENCES ETAGE(nEtage),
FOREIGN KEY (codeC) REFERENCES CATEGORIE(codeC)) ;
```

```
CREATE TABLE PRODUIT(
codeP VARCHAR(255) PRIMARY KEY,
description VARCHAR(255) ) ;
```

```
CREATE TABLE MARCHANDISE(
codeP VARCHAR(255) PRIMARY KEY,
tarif_TTC_piece NUMBER(10,2), qte_stock NUMBER(10),
FOREIGN KEY (codeP) REFERENCES PRODUIT(codeP) ) ;
```

```
CREATE TABLE SERVICE(
codeP VARCHAR(255) PRIMARY KEY,
tarif_TTC_heure NUMBER(10,2), duree NUMBER(10,2), nom_Professionnel
VARCHAR(255), contact_Professionnel VARCHAR(255),
FOREIGN KEY (codeP) REFERENCES PRODUIT(codeP) ) ;
```

```
CREATE TABLE RESERVATION(
nRes VARCHAR(255) PRIMARY KEY,
dateRes DATE, debutSejour DATE, finSejour DATE, nOccupants NUMBER(10), nPreauto
VARCHAR(255), datePreauto DATE, issuePreauto VARCHAR(255), idClient
VARCHAR(255), codeC VARCHAR(255), idReceptionniste VARCHAR(255),
FOREIGN KEY (idClient) REFERENCES PERSONNE(id),
```

```
FOREIGN KEY (codeC) REFERENCES CATEGORIE(codeC),
FOREIGN KEY (idReceptionniste) REFERENCES PERSONNE(id) );
```

```
CREATE TABLE ATTRIBUTION(
nRes VARCHAR(255) PRIMARY KEY,
nEtage NUMBER(10), nChambre NUMBER(10),
FOREIGN KEY (nRes) REFERENCES RESERVATION(nRes),
FOREIGN KEY (nEtage, nChambre) REFERENCES CHAMBRE(nEtage, nChambre) );
```

```
CREATE TABLE ENREGISTREMENT(
nEnr VARCHAR(255) PRIMARY KEY,
dateEnr DATE, nRes VARCHAR(255), codeP VARCHAR(255), quantite NUMBER(10),
FOREIGN KEY (nRes) REFERENCES RESERVATION(nRes),
FOREIGN KEY (codeP) REFERENCES PRODUIT(codeP) );
```

```
-- Insertions dans les tables
```

```
INSERT INTO METIER VALUES('receptionniste','140','1800');
INSERT INTO METIER VALUES('FDC','140','1300');
INSERT INTO METIER VALUES('informaticien','40','900');
INSERT INTO METIER VALUES('technicien','80','800');
```

```
INSERT INTO PERSONNE VALUES('P19900001','Boss','Patric', '1972-11-07', '75014',
NULL, NULL);
INSERT INTO PERSONNE VALUES('P19900007','Nandi','Lucia', '1970-05-30', '75015',
NULL, '0612345678');
INSERT INTO PERSONNE VALUES('P20050058','Clementine','Alyce', '1995-05-25',
'09045', 'alyce@mail.fr', NULL);
INSERT INTO PERSONNE VALUES('P20080890','Clio','Renoir', '2001-07-12', '22162',
'clio_2001@mail.fr', NULL);
INSERT INTO PERSONNE VALUES('P20200003','Liszt','Mark', '1998-12-06', '75003',
'liszt@mail.fr', '0655544422');
INSERT INTO PERSONNE VALUES('P20200018','Loddo','Vera', '1988-03-14', '75018',
'loddo@mail.fr', '0112121212');
INSERT INTO PERSONNE VALUES('P20200019','Sozzi','Lara', '2000-04-18', '75005',
NULL, '0600660066');
INSERT INTO PERSONNE VALUES('P20200078','Clont','Yannick', '1981-09-15',
'6548905', 'cy@mail.fr', NULL);
INSERT INTO PERSONNE VALUES('P20000012','Ricard','Adèle', '1978-12-17', '94025',
'ricardad@mail.fr', NULL);
INSERT INTO PERSONNE VALUES('P20190005','Legrand','Béatrice', '1985-08-23',
'75013', NULL, '0671994613');
INSERT INTO PERSONNE VALUES('P20180002','Bonisac','Esther', '1980-06-30',
'75006', 'bonisac@mail.fr', '0647923174');
INSERT INTO PERSONNE VALUES('P20190009','Lamont','Diletta', '1971-04-05',
'75018', NULL, '0695247203');
INSERT INTO PERSONNE VALUES('P20200021','Genod','Pierre', '1972-05-13', '92051',
NULL, '0647912412');
INSERT INTO PERSONNE VALUES('P20200131','Tuil','Ismael', '1967-07-30', '75001',
'tuillis@mail.fr', NULL);
INSERT INTO PERSONNE VALUES('P20200145','Vergnaud','Ferreol', '1953-11-08',
'69001', 'ferver@mail.fr', '0612668901');
INSERT INTO PERSONNE VALUES('P20200238','Cupertino','Diogo', '1943-10-23',
'75001', NULL, '0622387613');
INSERT INTO PERSONNE VALUES('P20200782','Corcos','Jules', '1962-03-17', '95021',
'corc@mail.fr', NULL);
INSERT INTO PERSONNE VALUES('P20170015','Lapensée','Béatrice', '1983-09-21',
'75020', 'lapensbea@mail.fr', '0621554391');
```

```
INSERT INTO CLIENT VALUES('P20050058','CCCCCCCCCCCC333');
INSERT INTO CLIENT VALUES('P20080890','CCCCCCCCCCCC885');
```

```

INSERT INTO CLIENT VALUES('P20200078','CCCCCCCCCCCC428');
INSERT INTO CLIENT VALUES('P20200021','CCCCCCCCCCCC776');
INSERT INTO CLIENT VALUES('P20200131','CCCCCCCCCCCC144');
INSERT INTO CLIENT VALUES('P20200145','CCCCCCCCCCCC925');
INSERT INTO CLIENT VALUES('P20200238','CCCCCCCCCCCC749');
INSERT INTO CLIENT VALUES('P20200782','CCCCCCCCCCCC329');

```

```

INSERT INTO EMPLOYE VALUES('P19900007','ABC123','XXX12','NNN14','FDC','P19900001');
INSERT INTO EMPLOYE VALUES('P20200018','XYZ099','XXX94','NNN45','FDC','P19900007');
INSERT INTO EMPLOYE VALUES('P20200003','RST777','XXX33','NNN09','receptionniste','P19900001');
INSERT INTO EMPLOYE VALUES('P20200019','JKY404','XXX75','NNN32','receptionniste','P20200003');
INSERT INTO EMPLOYE VALUES('P20000012','LRF144','XXX83','NNN17','FDC','P19900007');
INSERT INTO EMPLOYE VALUES('P20190005','JFK968','XXX42','NNN67','FDC','P19900007');
INSERT INTO EMPLOYE VALUES('P20180002','PCI946','XXX68','NNN23','FDC','P19900007');
INSERT INTO EMPLOYE VALUES('P20190009','RDS349','XXX58','NNN92','FDC','P19900007');
INSERT INTO EMPLOYE VALUES('P20170015','VFC979','XXX69','NNN38','receptionniste','P20200003');

```

```

INSERT INTO CATEGORIE VALUES('SGL','chambre single','1','70');
INSERT INTO CATEGORIE VALUES('DBL','chambre double','2','90');
INSERT INTO CATEGORIE VALUES('FML','chambre familiale','4','150');
INSERT INTO CATEGORIE VALUES('S30','suite 30mq','2','250');

```

```

INSERT INTO ETAGE VALUES('1','3','P20200018');
INSERT INTO ETAGE VALUES('2','5','P20200018');
INSERT INTO ETAGE VALUES('3','5','P19900007');
INSERT INTO ETAGE VALUES('4','5','P20000012');
INSERT INTO ETAGE VALUES('5','5','P20190005');

```

```

INSERT INTO CHAMBRE VALUES('1','01','15','Douche, Cour','SGL');
INSERT INTO CHAMBRE VALUES('1','02','22','Douche, Cour','DBL');
INSERT INTO CHAMBRE VALUES('1','03','30','Douche, Rue','FML');
INSERT INTO CHAMBRE VALUES('2','01','15','Douche, Cour','SGL');
INSERT INTO CHAMBRE VALUES('2','02','22','Douche, Cour','DBL');
INSERT INTO CHAMBRE VALUES('2','03','30','Douche, Rue, Balcon','FML');
INSERT INTO CHAMBRE VALUES('2','04','18','Baignoire, Rue','DBL');
INSERT INTO CHAMBRE VALUES('2','05','30','Baignoire, Rue, Balcon, Vue','S30');
INSERT INTO CHAMBRE VALUES('3','01','15','Douche, Cour','SGL');
INSERT INTO CHAMBRE VALUES('3','02','22','Douche, Cour','DBL');
INSERT INTO CHAMBRE VALUES('3','03','30','Douche, Rue, Balcon','FML');
INSERT INTO CHAMBRE VALUES('3','04','18','Baignoire, Rue','DBL');
INSERT INTO CHAMBRE VALUES('3','05','30','Baignoire, Rue, Balcon, Vue','S30');
INSERT INTO CHAMBRE VALUES('4','01','15','Douche, Cour','SGL');
INSERT INTO CHAMBRE VALUES('4','02','22','Douche, Cour','DBL');
INSERT INTO CHAMBRE VALUES('4','03','30','Douche, Rue, Balcon','FML');
INSERT INTO CHAMBRE VALUES('4','04','18','Baignoire, Rue','DBL');
INSERT INTO CHAMBRE VALUES('4','05','30','Baignoire, Rue, Balcon, Vue','S30');
INSERT INTO CHAMBRE VALUES('5','01','15','Douche, Cour','SGL');
INSERT INTO CHAMBRE VALUES('5','02','22','Douche, Cour','DBL');
INSERT INTO CHAMBRE VALUES('5','03','30','Douche, Rue, Balcon','FML');
INSERT INTO CHAMBRE VALUES('5','04','18','Baignoire, Rue','DBL');
INSERT INTO CHAMBRE VALUES('5','05','30','Baignoire, Rue, Balcon, Vue','S30');

```

```

INSERT INTO PRODUIT VALUES('PR001','Lit supplementaire');
INSERT INTO PRODUIT VALUES('PR002','Petit-déjeuner adulte');
INSERT INTO PRODUIT VALUES('PR003','Petit-déjeuner enfant -12');

```

```

INSERT INTO PRODUIT VALUES('PR004','Massage base');
INSERT INTO PRODUIT VALUES('PR005','Massage deluxe');
INSERT INTO PRODUIT VALUES('PR006','Repassage chemises');
INSERT INTO PRODUIT VALUES('PR007','Minibar alcool');
INSERT INTO PRODUIT VALUES('PR008','Minibar canette');

INSERT INTO SERVICE VALUES('PR004','60','1','M. Chen','0112312312');
INSERT INTO SERVICE VALUES('PR005','90','1.5','Mme Lola','lola@massage.fr');
INSERT INTO SERVICE VALUES('PR006','45','1','Mme Stiro','0698765432');

INSERT INTO MARCHANDISE VALUES('PR001','30','6');
INSERT INTO MARCHANDISE VALUES('PR002','18','50');
INSERT INTO MARCHANDISE VALUES('PR003','12','30');
INSERT INTO MARCHANDISE VALUES('PR007','8','43');
INSERT INTO MARCHANDISE VALUES('PR008','5','65');

INSERT INTO RESERVATION VALUES('R2019-110299','2019-10-10','2020-02-11','2020-02-13','2','WWW12','2019-10-10','VALIDEE','P20050058','DBL','P20200003');
INSERT INTO RESERVATION VALUES('R2020-000179','2020-01-05','2020-02-11','2020-02-15','1','WWW77','2020-01-05','VALIDEE','P20080890','SGL','P20200003');
INSERT INTO RESERVATION VALUES('R2020-000180','2020-01-05','2020-02-11','2020-02-15','2','WWW78','2020-01-05','VALIDEE','P20080890','DBL','P20200003');
INSERT INTO RESERVATION VALUES('R2020-000185','2020-02-10','2020-02-18','2020-02-19','2','WWW98','2020-02-10','REJETEE','P20200078','S30','P20200019');
INSERT INTO RESERVATION VALUES('R2020-000202','2020-02-10','2020-02-15','2020-02-16','2','WWW99','2020-02-10','VALIDEE','P20050058','DBL','P20200019');
INSERT INTO RESERVATION VALUES('R2020-000203','2020-02-10','2020-02-15','2020-02-16','2','WW101','2020-02-10','VALIDEE','P20050058','S30','P20200019');
INSERT INTO RESERVATION VALUES('R2020-000207','2020-02-11','2020-02-17','2020-02-19','2','WW103','2020-02-11','VALIDEE','P20200021','S30','P20200019');
INSERT INTO RESERVATION VALUES('R2020-000209','2020-02-11','2020-02-15','2020-02-16','2','WW106','2020-02-11','VALIDEE','P20200131','S30','P20200019');
INSERT INTO RESERVATION VALUES('R2020-000214','2020-02-12','2020-02-18','2020-02-19','2','WW111','2020-02-12','VALIDEE','P20200145','S30','P20200019');
INSERT INTO RESERVATION VALUES('R2020-000217','2020-02-12','2020-02-15','2020-02-18','1','WW115','2020-02-12','VALIDEE','P20200238','SGL','P20170015');
INSERT INTO RESERVATION VALUES('R2020-000221','2020-02-12','2020-02-17','2020-02-20','4','WW118','2020-02-12','REJETEE','P20200782','FML','P20200019');
INSERT INTO RESERVATION VALUES('R2020-000234','2020-02-14','2020-02-20','2020-02-27','2','WW124','2020-02-14','VALIDEE','P20200145','DBL','P20170015');
INSERT INTO RESERVATION VALUES('R2020-000235','2020-02-14','2020-02-21','2020-02-23','2','WW129','2020-02-14','VALIDEE','P20050058','DBL','P20200019');
INSERT INTO RESERVATION VALUES('R2020-000241','2020-02-14','2020-02-21','2020-02-25','4','WW135','2020-02-14','VALIDEE','P20200131','FML','P20200019');
INSERT INTO RESERVATION VALUES('R2020-000249','2020-02-14','2020-02-22','2020-02-24','2','WW141','2020-02-14','VALIDEE','P20200021','DBL','P20170015');

INSERT INTO ENREGISTREMENT VALUES('E2020-0018','2020-02-11','R2019-110299','PR008','1');
INSERT INTO ENREGISTREMENT VALUES('E2020-0021','2020-02-11','R2019-110299','PR008','1');
INSERT INTO ENREGISTREMENT VALUES('E2020-0023','2020-02-16','R2020-000217','PR004','1');
INSERT INTO ENREGISTREMENT VALUES('E2020-0028','2020-02-17','R2020-000217','PR007','3');
INSERT INTO ENREGISTREMENT VALUES('E2020-0031','2020-02-22','R2020-000241','PR002','2');
INSERT INTO ENREGISTREMENT VALUES('E2020-0035','2020-02-22','R2020-000241','PR003','2');
INSERT INTO ENREGISTREMENT VALUES('E2020-0039','2020-02-23','R2020-000249','PR006','1');
INSERT INTO ENREGISTREMENT VALUES('E2020-0040','2020-02-23','R2020-000249','PR004','1');

```

```

INSERT INTO ATTRIBUTION VALUES('R2019-110299','2','04');
INSERT INTO ATTRIBUTION VALUES('R2020-000179','1','01');
INSERT INTO ATTRIBUTION VALUES('R2020-000180','1','02');
INSERT INTO ATTRIBUTION VALUES('R2020-000202','1','02');
INSERT INTO ATTRIBUTION VALUES('R2020-000203','2','05');
INSERT INTO ATTRIBUTION VALUES('R2020-000207','3','05');
INSERT INTO ATTRIBUTION VALUES('R2020-000209','4','05');
INSERT INTO ATTRIBUTION VALUES('R2020-000214','5','05');
INSERT INTO ATTRIBUTION VALUES('R2020-000217','5','01');
INSERT INTO ATTRIBUTION VALUES('R2020-000234','3','04');
INSERT INTO ATTRIBUTION VALUES('R2020-000235','4','02');
INSERT INTO ATTRIBUTION VALUES('R2020-000241','5','03');
INSERT INTO ATTRIBUTION VALUES('R2020-000249','1','02');

```

-- Affichage du contenu des tables

```

SELECT * FROM METIER;
SELECT * FROM PERSONNE;
SELECT * FROM CLIENT;
SELECT * FROM EMPLOYE;
SELECT * FROM CATEGORIE;
SELECT * FROM ETAGE;
SELECT * FROM CHAMBRE;
SELECT * FROM PRODUIT;
SELECT * FROM MARCHANDISE;
SELECT * FROM SERVICE;
SELECT * FROM RESERVATION;
SELECT * FROM ATTRIBUTION;
SELECT * FROM ENREGISTREMENT;

```

-- Requetes SQL sur les tables

-- 1. Une requête de sélection avec un prédicat d'inégalité

-- Visualiser les informations des clients dont la préautorisation n'est pas valide

```

SELECT p.nom, p.prenom, cl.carte_credit, r.issuePreauto, r.nRes
FROM personne p, client cl, reservation r
WHERE p.id = cl.id AND cl.id = r.idClient
AND r.issuePreauto <> 'VALIDEE';

```

-- 2. Une différence (SQL et calcul relationnel)

-- Visualiser la liste des chambres pour "recouchage" (= nettoyage simple) le 15/02/2020

-- NB Les chambres dont le checkout est le 15/02/2020 doivent être nettoyées "à blanc" (= soigneusement)

```

SELECT c.nEtage, c.nChambre
FROM chambre c
MINUS
SELECT a.nEtage, a.nChambre
FROM reservation r, attribution a
WHERE r.nRes = a.nRes AND r.finSejour = '2020-02-15';

```

-- 3. Une auto-jointure (jointure d'une table avec elle-même) (SQL et relationnel)

-- Visualiser les noms des chefs et le nombre de subordonnés pour chacun d'entre eux

-- NB. La personne P19900001 est chef de P1900007 et de P20200003, mais il ne figure pas entre les employés car il est le propriétaire

```

SELECT p.nom, p.prenom, COUNT(*)
FROM personne p, employe e1, employe e2
WHERE p.id = e2.id AND e1.chef = e2.id
GROUP BY p.nom, p.prenom
ORDER BY COUNT(*) DESC ;

```

-- 4. Une division (SQL et calcul relationnel)

-- Visualiser la liste des receptionnistes ayant vendu toutes les differentes suites (S30) au moins une fois

```

SELECT p.nom, p.prenom
FROM personne p
WHERE NOT EXISTS (SELECT *
                  FROM chambre c
                  WHERE c.codeC = 'S30'
                  AND NOT EXISTS (SELECT *
                                FROM reservation r, attribution a
                                WHERE p.id = r.idReceptionniste
                                AND r.nRes = a.nRes
                                AND a.nEtag = c.nEtag
                                AND a.nChambre = c.nChambre));

```

-- 5. Une requête qui utilise une condition qui porte sur la différence entre deux dates (SQL)

-- Visualiser la liste de clients qui ont une réservation validée pour un séjour plus long que 5 nuits.

```

SELECT DISTINCT p.nom, p.prenom
FROM personne p, reservation r
WHERE p.id = r.idClient AND r.issuePreauto = 'VALIDEE'
AND DATEDIFF(DAY, r.debutSejour, r.finSejour) > 5
ORDER BY p.nom;

```

-- Calculer l'âge moyenne des clients de l'hôtel

```

SELECT round (AVG ( round (DATEDIFF(YEAR, p.date_naiss, sysdate),
2 ) ) , 2) as age_moyenne_clients
FROM personne p, client cl
WHERE p.id = cl.id AND p.date_naiss IS NOT NULL;

```

-- 6. Une requête avec une requête imbriquée utilisant ALL (SQL)

-- Visualiser le contact des collaborateurs professionnels n'ayant jamais vendu leur service

```

SELECT s.nom_professionnel, s.contact_professionnel, p.description
FROM service s, produit p
WHERE s.codeP = p.codeP
AND s.codeP <> ALL (SELECT enr.codeP
                  FROM enregistrement enr
                  GROUP BY enr.codeP) ;

```

-- 7. Une requête avec une requête imbriquée utilisant ANY (SQL)

-- Visualiser les noms des FDC qui sont responsables d'au moins un étage

```

SELECT DISTINCT p.nom, p.prenom
FROM personne p, etage e, employe emp
WHERE p.id = emp.id AND emp.metier = 'FDC'
AND emp.id = ANY (SELECT idFDC FROM Etage);

```

-- 8. Une requête avec UNION (SQL)

-- Visualiser le catalogue des prix des services, des marchandises et des chambres

```

SELECT cat.description, cat.tarif_TTC
FROM categorie cat
UNION
SELECT p.description, s.tarif_TTC_heure
FROM produit p, service s
WHERE p.codeP = s.codeP
UNION

```

```

SELECT p.description, m.tarif_TTC_piece
FROM produit p, marchandise m
WHERE p.codeP = m.codeP;

-- 9. Une requête avec deux INTERSECT (SQL)
-- Visualiser la liste des chambres avec douche plus grandes de 20 mq dont le
prix est inférieur à 180€
SELECT c.nEtage, c.nChambre
FROM chambre c
WHERE c.description LIKE '%Douche%'
INTERSECT
SELECT c.nEtage, c.nChambre
FROM chambre c, categorie cat
WHERE c.codeC = cat.codeC AND cat.tarif_TTC < 180
INTERSECT
SELECT c.nEtage, c.nChambre
FROM chambre c
WHERE c.dimension > 20;

-- 10. Donnez une requête d'agrégation avec GROUP BY
-- Visualiser le nombre des chambres par catégorie et leurs tarifs TTC
SELECT c.codeC, COUNT(*) AS Disponibilité, cat.tarif_TTC
FROM chambre c, categorie cat
WHERE c.codeC = cat.codeC
GROUP BY c.codeC, cat.tarif_TTC
ORDER BY cat.tarif_TTC;

-- 11. Donnez une requête d'agrégation avec GROUP BY et HAVING
-- Visualiser le nom des clients plus fidèles de l'hôtel (= ayant effectué le
maximum de réservations)
SELECT p.nom, p.prenom, COUNT(*) AS nbReservationsEffectués
FROM personne p, reservation r
WHERE p.id = r.idClient
GROUP BY p.nom, p.prenom
HAVING COUNT (*) = (SELECT MAX(tot)
FROM (SELECT COUNT (*) as tot
FROM reservation r
GROUP BY r.idClient) );

-----
-- Elimination des tables une à la fois
/* DROP TABLE IF EXISTS ENREGISTREMENT;
DROP TABLE IF EXISTS ATTRIBUTION;
DROP TABLE IF EXISTS RESERVATION;
DROP TABLE IF EXISTS SERVICE;
DROP TABLE IF EXISTS MARCHANDISE;
DROP TABLE IF EXISTS PRODUIT;
DROP TABLE IF EXISTS CHAMBRE;
DROP TABLE IF EXISTS ETAGE;
DROP TABLE IF EXISTS CATEGORIE;
DROP TABLE IF EXISTS EMPLOYE;
DROP TABLE IF EXISTS CLIENT;
DROP TABLE IF EXISTS PERSONNE;
DROP TABLE IF EXISTS METIER; */
-- ou sinon elimination globale : DROP ALL OBJECTS;

```