

Cyriac THIBAUDEAU
Aymeric GOUDOUT
Ethan MACHE

Projet Base de Données Cahier des charges

Ce document a pour but de vous exposer notre mise en place d'une base de données portant sur la gestion des réservations et des commandes d'un restaurant.

Le projet a été réalisé dans le but d'être évalué, mais nous pensons qu'il est tout à fait utilisable dans un contexte de nécessité pour un restaurant de posséder une base de données gérant ces points-ci.

De part les différentes restrictions qui nous ont été imposées, certains éléments constituant un restaurant et sa gestion ont été mis à l'écart, pour concentrer nos efforts sur les points qui nous semblaient utiles. En effet, la forme de cette base de données est faite pour être simple d'utilisation et de compréhension, tout en étant complète dans ce qu'elle sert à faire.

Le restaurant que nous prendrons en exemple sera un restaurant totalement fictif. Celui-ci possède plusieurs zones dans notre cas un rez-de-chaussée que l'on appellera « intérieur », un « étage » et une « terrasse ». Dans chacune de ces zones, plusieurs tables sont disposées pour un total de 20 effectifs, et 60 places maximum pouvant être servies simultanément. Elles peuvent avoir un nombre de couverts différents. Ce restaurant propose par ailleurs une carte : d'entrées, plats, dessert et boissons.

Entrées	Desserts
Soupe à l'oignon ; 5€	Eclair au chocolat ; 3€
Salade niçoise ; 6€	Tarte aux pommes ; 4€
Salade de chèvre chaud ; 7€	Mousse au chocolat ; 4€
Quiche lorraine ; 8€	Crème brûlée ; 5€
Carpaccio de boeuf ; 8€	Tiramisu ; 5€
Terrine de foie gras ; 8€	Profiteroles ; 5€
	Fondant au chocolat ; 5€
Plats	Boissons
Ratatouille ; 9€	Bouteille d'eau plate ; 1€
Lasagne bolognaise ; 10€	Café ; 2€
Poulet rôti ; 11€	Thé vert ; 2€
Coq en vin ; 12€	Jus d'orange ; 3€
Paella ; 12€	Limonade ; 3€
Cassoulet ; 13€	Soda cola ; 3€
Bœuf Bourguignon ; 14€	

Pour finir, 3 employés travaillent en salle pour servir les clients.

Actuellement, le propriétaire aimerait pouvoir, par exemple, connaître, simplement, plusieurs statistiques et données sur son restaurant : les recettes par mois, le nombre de réservations en un temps donné ...

La base de données que l'on doit mettre en place doit donc pouvoir répondre à plusieurs questions que ce patron voudrait demander à cette base, pour ses différents besoins.

Le système doit permettre au clients de réserver une table pour une date et une heure spécifique. Certaines informations sont nécessaires à connaître lors des enregistrements de celles-ci, tel que le nom du client, le nombre de personnes, la date et l'heure et la zone dans laquelle ils souhaiteraient être enregistrés. De plus, pour pouvoir tracer chaque réservations, un identifiant doit leur être attribué. Les réservations se faisant par appels téléphoniques, les serveurs doivent pouvoir créer, supprimer ou modifier des réservations.

Le système doit permettre l'enregistrement des détails des tables du restaurant, tel que la capacité de celles-ci et encore une fois, un identifiant doit leur être affecté pour pouvoir tracer les tables affectées aux réservations. La disponibilité des tables doit aussi pouvoir être accessible, en connaissant les heures d'affectations des tables aux réservations.

Le système doit permettre l'enregistrement de la carte du restaurant et des données associées à chaque plat : le nom du plat, la catégorie et le prix. Par ailleurs, lors d'une commande, il devra être nécessaire de pouvoir spécifier la quantité des plats commandés.

Le système doit permettre l'enregistrement des informations relatives aux employés, notamment leur nom, prénom, numéro de téléphone, salaire, date d'embauche, adresse physique et adresse mail. La base de données doit aussi pouvoir indiquer quel employé a servi une commande, il est donc obligatoire que chaque employé ait un identifiant. On pourrait aussi pouvoir connaître le nombre de commandes servies et les pourboires reçu par les employés pour une commande. On précise que chaque employé sera considéré comme serveur. Cependant, les données des employés ne doivent pas être accessibles à tous. Des mesures de sécurités doivent être mises en place pour éviter cela.

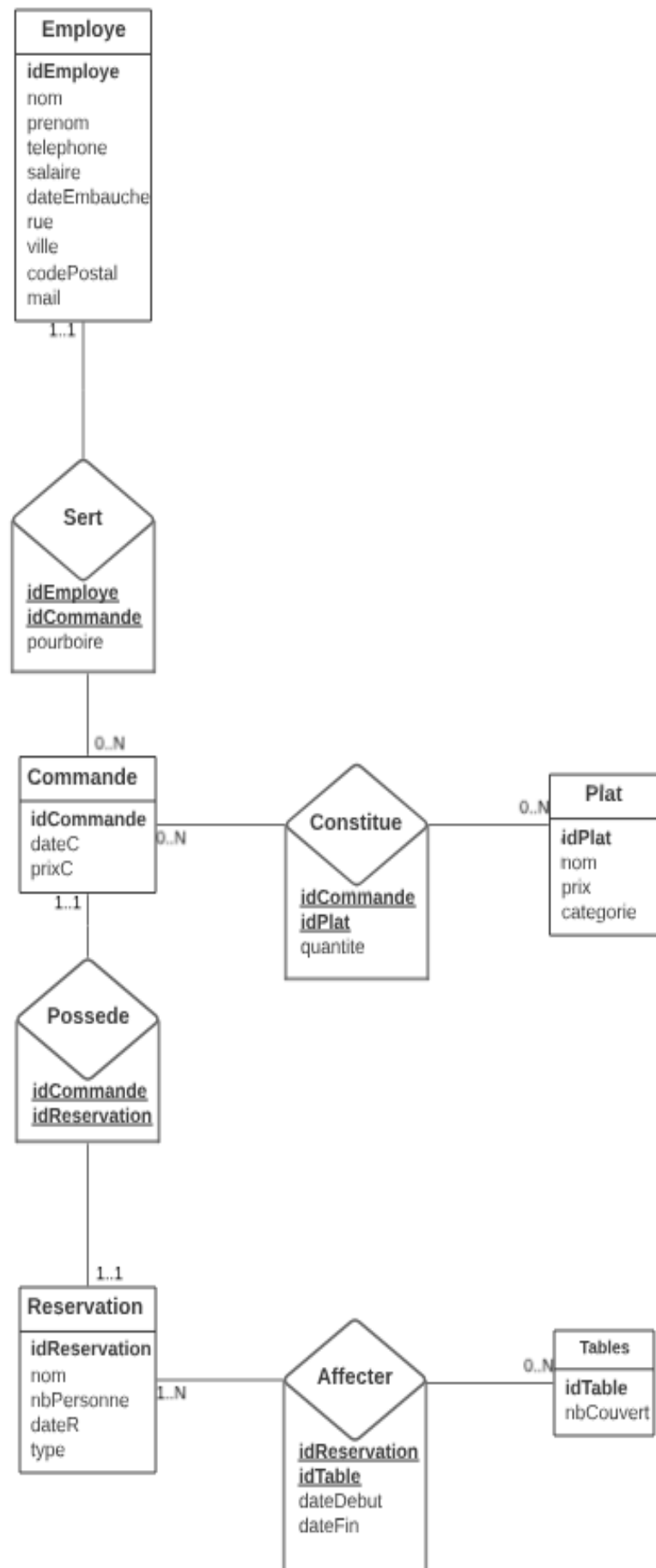
Le système doit finalement pouvoir gérer des commandes. On doit pouvoir sauvegarder la date de chaque commande et le prix total de celle-ci. A nouveau, un identifiant sera affecté à chaque commande. Cet identifiant servira à lier la réservation, les plats commandé et l'employé qui la sert.

Vis-à-vis de la sécurité de la base de données, il est nécessaire de mettre en place des utilisateurs différents avec leur propre mot de passe. L'administrateur de la base doit évidemment avoir accès à tout les privilèges possibles. Le patron, lui, pourra avoir accès à plusieurs données sous forme de vues. Chaque employés auront quant à eux accès à leur données personnelles et la valeur des pourboires perçus.

Pour les contraintes techniques du projet, le choix de l'hébergement local ou cloud revient au patron et de ses besoins. Si il n'est pas nécessaire de pouvoir y accéder autres part qu'au restaurant, un serveur local suffit. Cependant, si l'administrateur ne peut se déplacer et une maintenance immédiate est nécessaire, le fait de l'héberger sur un cloud donnerait la possibilité à celui-ci d'effectuer tout changement nécessaires instantanément.

Le projet sera réalisé sous Oracle 11.2.0.2.0 en utilisant SQLPlus pour exécuter des commandes PL/SQL dans le but de mettre en place des fonctions plus complexes. On l'utilisera notamment pour mettre en place des procédures et des triggers. Pour gérer l'organisation de l'insertion du code, on utilisera SQLLoader. Le code doit être documenté clairement pour une maintenance plus aisée par l'administrateur.

Voici le diagramme relationnel :



Voici les différentes tables :

Reservation : **idReservation** int, nom varchar(20), nbPersonne int, dateR date, type varchar(20)

Tables : **idTable** int, nbCouvert int

Commande : **idCommande** int, dateC date ,prixC float

Plat : **idPlat** int, nom varchar(35), prix float, categorie varchar(10)

Employe : **idEmploye** int, nom varchar(20), prenom varchar(20), telephone varchar(12), salaire float, dateEmbauche date, rue varchar(50), ville varchar(50), codePostal varchar(5), mail varchar(50)

Les différentes relations :

Sert : **idEmploye** int, **idCommande** int, pourboire float default 0.0

Affecter : **idReservation** int, **idTable** int, dateDebut date, dateFin date

Constitue : **idCommande** int, **idPlat** int, quantite int

Possede : **idCommande** int, **idReservation** int

Clés étrangères :

idEmploye fait référence à Employe sur la clé primaire idEmploye.

idCommande fait référence à Commande sur la clé primaire idCommande.

idReservation fait référence à Reservation sur la clé primaire idReservation.

idTable fait référence à Tables sur la clé primaire idTable.

idCommande fait référence à Commande sur la clé primaire idCommande.

idPlat fait référence à Plat sur la clé primaire idPlat.

idCommande fait référence à Commande sur la clé primaire idCommande.

idReservation fait référence à Reservation sur la clé primaire idReservation.

Cardinalités :

- Reservation(1..N) -- Affecter -- Tables(0..N) :
 - Chaque réservation peut être liée à au moins une ou plusieurs entrées de la table Table en fonction du nombre de tables nécessaires à la réservation.
- Commande(1..1) -- Possede -- Reservation(1..1) :
 - Une réservation peut être associée à une commande, et une commande est possédée par une seule réservation.
- Commande(0..N) -- Constitue -- Plat(0..N) :
 - Chaque plat peut être une partie d'une commande, et chaque plat peut constituer une commande.
- Commande(0..N) -- Sert -- Employe(1..1) :
 - Chaque employé peut servir au moins une commande ou aucune, et une commande est servie par un employé.

Questions pertinentes :

- Quel est la carte du restaurant ?
- Quel est le nombre de tables et de places maximum disponibles ?
- Quelles sont les réservations ayant eu lieu le 25 décembre ?
- Combien de tables sont occupées par une réservation ?
- Quelle est l'heure du pic de commandes pour un plat donné ?
- Quels plats sont consommés dans tous les secteurs ?
- Combien de fois une table a été occupée le 29 décembre 2023 ?
- Combien y a-t-il eu de réservations le soir ?
- Combien y a-t-il eu de réservations le midi ?
- Combien de commandes possède une entrée ?
- Combien d'argent a généré le poulet rôti ?
- Combien d'argent ont générés les boissons ?

Vues pertinentes :

- Afficher les plats les plus consommés
- Afficher les recettes du restaurant
- Afficher les bénéfices du mois du restaurant
- Afficher les secteurs les plus populaires
- Nombre de personnes moyen par réservation
- Le total des pourboires de l'employé X
- Afficher les détails de l'employé X

Voici le script permettant de créer les tables de la base de données :

```
create table Reservation (  
    idReservation int  
    nom varchar(20),  
    nbPersonne int  
    dateR date,  
    type varchar(20),  
    primary key (idReservation)  
);
```

```
create table Tables (  
    idTable int  
    nbCouvert int  
    primary key (idTable)  
);
```

```
create table Commande (  
    idCommande int  
    dateC date,  
    prixC float,  
    primary key (idCommande)  
);
```

```
create table Plat (  
    idPlat int  
    nom varchar(35),  
    prix float,  
    categorie varchar(10),  
    primary key (idPlat)  
);
```

```
create table Employe (  
    idEmploye int  
    nom varchar(20),  
    prenom varchar(20),  
    telephone varchar(12),  
    salaire float,  
    dateEmbauche date,  
    rue varchar(50),  
    ville varchar(50),  
    codePostal varchar(5),  
    mail varchar(50),  
    primary key (idEmploye)  
);
```

```
create table Sert (  
    idEmploye int  
    idCommande int  
    pourboire float default 0.0,  
    primary key (idEmploye, idCommande),  
    foreign key (idEmploye) references Employe (idEmploye),  
    foreign key (idCommande) references Commande (idCommande)  
);
```

```
create table Affecter (  
    idReservation int  
    idTable int  
    dateDebut date,  
    dateFin date,  
    primary key (idReservation, idTable),  
    foreign key (idReservation) references Reservation (idReservation),  
    foreign key (idTable) references Tables (idTable)  
);
```

```
create table Constitue (  
    idCommande int  
    idPlat int  
    quantite int  
    primary key (idCommande, idPlat),  
    foreign key (idCommande) references Commande (idCommande),  
    foreign key (idPlat) references Plat (idPlat)  
);
```

```

create table Possede (
    idCommande int
    idReservation int
    primary key (idCommande, idReservation),
    foreign key (idCommande) references Commande (idCommande),
    foreign key (idReservation) references Reservation (idReservation)
);

```

Voici les vues mises en place :

```

/* affiche les plats les plus consommé */
create or replace view plat_populaire as (
select p.nom, sum(c.quantite) as nb_total
from Constitue c, Plat p
where c.idPlat = p.idPlat
group by p.nom );

```

```

/* affiche les recettes du restaurant */
create or replace view recettes as (
select sum(prixC) as Recette
from Commande);

```

```

/* affiche les bénéfices du mois du restaurant */
create or replace view benefice_mois as (
select
(select * from recettes)
-
(select sum(salaire) from Employe) as Benefice
from dual);

```

```

/* affiche les secteurs les plus populaires */
create or replace view secteur_populaire as (
select type, count(type) as nombre
from Reservation
group by type);

```

```

/* nombre de personnes moyen par réservation */
create or replace view avg_people as (
select round(avg(nbPersonne)) as nb_personne_moyen
from Reservation);

```

```

/* le total des pourboires de l'employé 1 */
create or replace view val_pourboire_1 as (
select e.nom, e.prenom, sum(s.pourboire) as pourboire_total
from Employe e, Sert s
where e.idEmploye = s.idEmploye and e.idEmploye = 1
group by e.nom, e.prenom);

```

```

/* le total des pourboires de l'employé 2 */
create or replace view val_pourboire_2 as (
select e.nom, e.prenom, sum(s.pourboire) as pourboire_total
from Employe e, Sert s
where e.idEmploye = s.idEmploye and e.idEmploye = 2
group by e.nom, e.prenom);

```

```

/* le total des pourboires de l'employé 3 */
create or replace view val_pourboire_3 as (
select e.nom, e.prenom, sum(s.pourboire) as pourboire_total
from Employe e, Sert s
where e.idEmploye = s.idEmploye and e.idEmploye = 3
group by e.nom, e.prenom);

```

```

/* affiche les details de l'employé 1 */
create or replace view detail_employe_1 as (
select * from employe where idEmploye = 1);

```

```

/* affiche les details de l'employé 2 */
create or replace view detail_employe_2 as (
select * from employe where idEmploye = 2);

```

```

/* affiche les details de l'employé 3 */
create or replace view detail_employe_3 as (
select * from employe where idEmploye = 3);

```

Voici les différentes triggers :

```

/* Incrémente idTable en fonction du dernier élément le plus grand */
create or replace trigger tableId before insert on Tables for each row
declare
    nbmax Tables.idTable%type := 0;
begin
    select nvl(max(idTable), 0) into nbmax
    from Tables;
    :new.idTable := nbmax + 1;
end;
/

```

```

/* Incrémente idReservation en fonction du dernier élément le plus grand */
create or replace trigger reservationId before insert on Reservation for each row
declare
    nbmax Reservation.idReservation%type := 0;
begin
    select nvl(max(idReservation), 0) into nbmax
    from Reservation;
    :new.idReservation := nbmax + 1;
end;
/

```



```

/* Incrémente idCommande en fonction du dernier élément le plus grand */
create or replace trigger commandeId before insert on Commande for each row
    declare
        nbmax Commande.idCommande%type := 0;
    begin
        select nvl(max(idCommande), 0) into nbmax
        from Commande;
        :new.idCommande := nbmax + 1;
    end;
/

```

```

/* Incrémente idPlat en fonction du dernier élément le plus grand */
create or replace trigger platId before insert on Plat for each row
    declare
        nbmax Plat.idPlat%type := 0;
    begin
        select nvl(max(idPlat), 0) into nbmax
        from Plat;
        :new.idPlat := nbmax + 1;
    end;
/

```

```

/* Incrémente idEmploye en fonction du dernier élément le plus grand */
create or replace trigger employeId before insert on Employe for each row
    declare
        nbmax Employe.idEmploye%type := 0;
    begin
        select nvl(max(idEmploye), 0) into nbmax
        from Employe;
        :new.idEmploye := nbmax + 1;
    end;
/

```

```

/* Met à jour Affecter.dateDebut en fonction de la date de la réservation associée */
create or replace trigger dateAffecter before insert on Affecter for each row
    declare
        dte Affecter.dateDebut%type;
    begin
        select TO_DATE(dateR, 'DD-MM-YYYY HH24:MI') into dte from Reservation
        where idReservation = :new.idReservation;
        :new.dateDebut := dte;
    end;
/

```

```

/* Calcule le prix totale, prixC, de la commande lors d'une insertion ou d'une suppression sur Sert */
create or replace trigger prixTot before insert or delete on Constitue for each row
    declare
        prix Plat.prix%type;
    begin
        if (inserting) then
            select prix into prix from Plat where idPlat = :new.idPlat;
            update Commande set prixC = prixC + :new.quantite * prix where idCommande =

```

```

:new.idCommande;
    end if;
    if (deleting) then
        select prix into prix from Plat where idPlat = :old.idPlat;
        update Commande set prixC = prixC - :old.quantite * prix where idCommande =
:old.idCommande;
    end if;
    end;
/

/* Gère si il reste des places avant de valider une réservation */
create or replace trigger placeMax before insert on Reservation for each row
declare
    place_dispo int := 0;
begin
    select sum(t.nbCouvert) into place_dispo
    from Affecter a, Tables t
    where a.idTable = t.idTable and a.dateFin is not NULL;
    if (:new.nbPersonne > place_dispo) then
        raise_application_error(-20001, 'Pas assez de place disponible');
    end if;
end;
/

```

Liste des contraintes d'intégrités :

```

alter table Tables add constraint ch01 check (nbCouvert = 2 or nbCouvert = 4);
alter table Employe add constraint ch02 check (salaire >= 1500);
alter table Employe add constraint ch03 check (mail LIKE '%@%');
alter table Plat add constraint ch04 check (categorie in ('entree', 'resistance', 'dessert', 'boisson'));
alter table Constitue add constraint ch05 check (quantite > 0);
alter table Sert add constraint ch06 check (pourboire >= 0.0);
alter table Affecter add constraint ch07 check (dateDebut < dateFin);
alter table Reservation add constraint ch08 check ((TO_NUMBER(TO_CHAR(dateR, 'HH24'))
between 11 and 14) or (TO_NUMBER(TO_CHAR(dateR, 'HH24')) between 19 and 22));
alter table Reservation add constraint ch09 check (nbPersonne > 0);
alter table Reservation add constraint ch10 check (type in ('Interieur', 'Terasse', 'Etage'));
alter table Plat add constraint ch11 check (prix > 0);

```

Liste des insertions de données :

```

insert into Tables values (NULL, 2);
insert into Tables values (NULL, 2);
insert into Tables values (NULL, 4);
insert into Tables values (NULL, 4);
insert into Tables values (NULL, 4);
insert into Tables values (NULL, 4);
insert into Tables values (NULL, 4);
insert into Tables values (NULL, 4);
insert into Tables values (NULL, 4);
insert into Tables values (NULL, 2);
insert into Tables values (NULL, 2);
insert into Tables values (NULL, 4);
insert into Tables values (NULL, 2);

```

```
insert into Tables values (NULL, 2);
insert into Tables values (NULL, 4);
insert into Tables values (NULL, 2);
insert into Tables values (NULL, 2);
insert into Tables values (NULL, 2);
insert into Tables values (NULL, 2);
insert into Tables values (NULL, 4);
insert into Tables values (NULL, 4);
```

```
insert into Reservation values (NULL, 'Lemoine', 4, TO_DATE('19-12-2023 12:45', 'DD-MM-YYYY
HH24:MI'), 'Etage');
insert into Reservation values (NULL, 'Leroy', 7, TO_DATE('21-12-2023 12:30', 'DD-MM-YYYY
HH24:MI'), 'Etage');
insert into Reservation values (NULL, 'Durand', 2, TO_DATE('02-12-2023 20:10', 'DD-MM-YYYY
HH24:MI'), 'Etage');
insert into Reservation values (NULL, 'Thomas', 6, TO_DATE('29-12-2023 12:10', 'DD-MM-YYYY
HH24:MI'), 'Interieur');
insert into Reservation values (NULL, 'André', 7, TO_DATE('25-12-2023 11:35', 'DD-MM-YYYY
HH24:MI'), 'Etage');
insert into Reservation values (NULL, 'Duboi', 7, TO_DATE('03-12-2023 11:10', 'DD-MM-YYYY
HH24:MI'), 'Interieur');
insert into Reservation values (NULL, 'André', 5, TO_DATE('01-12-2023 11:55', 'DD-MM-YYYY
HH24:MI'), 'Etage');
insert into Reservation values (NULL, 'Dupon', 8, TO_DATE('27-12-2023 12:50', 'DD-MM-YYYY
HH24:MI'), 'Interieur');
insert into Reservation values (NULL, 'André', 4, TO_DATE('16-12-2023 20:40', 'DD-MM-YYYY
HH24:MI'), 'Terasse');
insert into Reservation values (NULL, 'Rocher', 2, TO_DATE('09-12-2023 11:00', 'DD-MM-YYYY
HH24:MI'), 'Etage');
insert into Reservation values (NULL, 'Laurent', 3, TO_DATE('04-12-2023 11:10', 'DD-MM-YYYY
HH24:MI'), 'Etage');
insert into Reservation values (NULL, 'Roux', 2, TO_DATE('18-12-2023 13:50', 'DD-MM-YYYY
HH24:MI'), 'Interieur');
insert into Reservation values (NULL, 'Lefebvre', 1, TO_DATE('13-12-2023 12:05', 'DD-MM-YYYY
HH24:MI'), 'Interieur');
insert into Reservation values (NULL, 'Roux', 6, TO_DATE('01-12-2023 11:20', 'DD-MM-YYYY
HH24:MI'), 'Terasse');
insert into Reservation values (NULL, 'Dupont', 4, TO_DATE('01-12-2023 22:40', 'DD-MM-YYYY
HH24:MI'), 'Terasse');
insert into Reservation values (NULL, 'fantome', 40, TO_DATE('31-12-2023 21:00', 'DD-MM-YYYY
HH24:MI'), 'Etage');
```

```
insert into Plat values (NULL, 'Tarte aux pommes', 4, 'dessert');
insert into Plat values (NULL, 'Coq au vin', 12, 'resistance');
insert into Plat values (NULL, 'Salade nicoise', 6, 'entree');
insert into Plat values (NULL, 'Creme brulee', 5, 'dessert');
insert into Plat values (NULL, 'Ratatouille', 9, 'resistance');
insert into Plat values (NULL, 'Soupe a l oignon', 5, 'entree');
insert into Plat values (NULL, 'Mousse au chocolat', 4, 'dessert');
insert into Plat values (NULL, 'Boeuf bourguignon', 14, 'resistance');
insert into Plat values (NULL, 'Terrine de foie gras', 8, 'entree');
insert into Plat values (NULL, 'Eclair au chocolat', 3, 'dessert');
insert into Plat values (NULL, 'Cassoulet', 13, 'resistance');
insert into Plat values (NULL, 'Quiche lorraine', 7, 'entree');
insert into Plat values (NULL, 'Tiramisu', 5, 'dessert');
insert into Plat values (NULL, 'Poulet roti', 11, 'resistance');
insert into Plat values (NULL, 'Salade de chevre chaud', 7, 'entree');
```

insert into Plat values (NULL, 'Fondant au chocolat', 5, 'dessert');
insert into Plat values (NULL, 'Lasagnes a la bolognaise', 10, 'resistance');
insert into Plat values (NULL, 'Carpaccio de boeuf', 8, 'entree');
insert into Plat values (NULL, 'Profiteroles', 5, 'dessert');
insert into Plat values (NULL, 'Paella', 12, 'resistance');
insert into Plat values (NULL, 'Eau plate', 1, 'boisson');
insert into Plat values (NULL, 'Cafe', 2, 'boisson');
insert into Plat values (NULL, 'The vert', 2, 'boisson');
insert into Plat values (NULL, 'Jus orange', 3, 'boisson');
insert into Plat values (NULL, 'Limonade', 3, 'boisson');
insert into Plat values (NULL, 'Coca cola', 3, 'boisson');

insert into Employe values (NULL, 'Dupont', 'Jean', '0623456789', 1500.0, TO_DATE('10-08-2023', 'DD-MM-YYYY'), '123 rue de la Paix', 'Paris', '75000', 'jean.dupont@gmail.com');
insert into Employe values (NULL, 'Martin', 'Marie', '0687654321', 1700.0, TO_DATE('15-09-2023', 'DD-MM-YYYY'), '456 boulevard Saint-Germain', 'Paris', '75000', 'marie.martin@gmail.com');
insert into Employe values (NULL, 'Constant', 'David', '0654216810', 2000.0, TO_DATE('04-07-2023', 'DD-MM-YYYY'), '112 rue de la Paix', 'Paris', '75000', 'Constant.David@gmail.com');

insert into Affecter values (1, 3, NULL, TO_DATE('19-12-2023 13:35', 'DD-MM-YYYY HH24:MI'));
insert into Affecter values (2, 3, NULL, TO_DATE('21-12-2023 14:05', 'DD-MM-YYYY HH24:MI'));
insert into Affecter values (2, 4, NULL, TO_DATE('21-12-2023 14:05', 'DD-MM-YYYY HH24:MI'));
insert into Affecter values (3, 1, NULL, TO_DATE('02-12-2023 20:50', 'DD-MM-YYYY HH24:MI'));
insert into Affecter values (4, 2, NULL, TO_DATE('29-12-2023 13:40', 'DD-MM-YYYY HH24:MI'));
insert into Affecter values (4, 5, NULL, TO_DATE('29-12-2023 13:40', 'DD-MM-YYYY HH24:MI'));
insert into Affecter values (5, 3, NULL, TO_DATE('25-12-2023 14:15', 'DD-MM-YYYY HH24:MI'));
insert into Affecter values (5, 4, NULL, TO_DATE('25-12-2023 14:15', 'DD-MM-YYYY HH24:MI'));
insert into Affecter values (14, 8, NULL, TO_DATE('01-12-2023 12:50', 'DD-MM-YYYY HH24:MI'));
insert into Affecter values (14, 7, NULL, TO_DATE('01-12-2023 12:50', 'DD-MM-YYYY HH24:MI'));
insert into Affecter values (16, 1, NULL, NULL);
insert into Affecter values (16, 2, NULL, NULL);
insert into Affecter values (16, 3, NULL, NULL);
insert into Affecter values (16, 4, NULL, NULL);
insert into Affecter values (16, 5, NULL, NULL);
insert into Affecter values (16, 6, NULL, NULL);
insert into Affecter values (16, 7, NULL, NULL);
insert into Affecter values (16, 8, NULL, NULL);
insert into Affecter values (16, 9, NULL, NULL);
insert into Affecter values (16, 10, NULL, NULL);
insert into Affecter values (16, 11, NULL, NULL);
insert into Affecter values (16, 12, NULL, NULL);
insert into Affecter values (16, 13, NULL, NULL);
insert into Affecter values (16, 14, NULL, NULL);
insert into Affecter values (16, 15, NULL, NULL);
insert into Affecter values (16, 16, NULL, NULL);
insert into Affecter values (16, 17, NULL, NULL);
insert into Affecter values (16, 18, NULL, NULL);
insert into Affecter values (16, 19, NULL, NULL);
insert into Affecter values (16, 20, NULL, NULL);

insert into Commande values (NULL, TO_DATE('19-12-2023 12:45', 'DD-MM-YYYY HH24:MI'), 0);
insert into Commande values (NULL, TO_DATE('21-12-2023 12:30', 'DD-MM-YYYY HH24:MI'), 0);
insert into Commande values (NULL, TO_DATE('02-12-2023 20:10', 'DD-MM-YYYY HH24:MI'), 0);
insert into Commande values (NULL, TO_DATE('29-12-2023 12:10', 'DD-MM-YYYY HH24:MI'), 0);
insert into Commande values (NULL, TO_DATE('25-12-2023 11:35', 'DD-MM-YYYY HH24:MI'), 0);
insert into Commande values (NULL, TO_DATE('01-12-2023 11:20', 'DD-MM-YYYY HH24:MI'), 0);
insert into Commande values (NULL, TO_DATE('31-12-2023 21:00', 'DD-MM-YYYY HH24:MI'), 0);

insert into Possede values (1, 1);
insert into Possede values (2, 2);
insert into Possede values (3, 3);
insert into Possede values (4, 4);
insert into Possede values (5, 5);
insert into Possede values (6, 14);
insert into Possede values (7, 16);

insert into Constitue values (1, 3, 3);
insert into Constitue values (1, 6, 1);
insert into Constitue values (1, 5, 2);
insert into Constitue values (1, 8, 2);
insert into Constitue values (2, 18, 7);
insert into Constitue values (2, 17, 4);
insert into Constitue values (2, 14, 3);
insert into Constitue values (2, 16, 7);
insert into Constitue values (3, 9, 1);
insert into Constitue values (3, 12, 1);
insert into Constitue values (3, 5, 2);
insert into Constitue values (3, 19, 1);
insert into Constitue values (3, 10, 1);
insert into Constitue values (4, 3, 4);
insert into Constitue values (4, 15, 2);
insert into Constitue values (4, 14, 5);
insert into Constitue values (4, 5, 1);
insert into Constitue values (5, 11, 3);
insert into Constitue values (5, 14, 4);
insert into Constitue values (5, 19, 6);
insert into Constitue values (5, 10, 1);
insert into Constitue values (6, 2, 3);
insert into Constitue values (6, 5, 3);
insert into Constitue values (6, 1, 1);
insert into Constitue values (6, 4, 2);
insert into Constitue values (6, 7, 3);
insert into Constitue values (7, 3, 7);
insert into Constitue values (7, 15, 8);
insert into Constitue values (7, 9, 15);
insert into Constitue values (7, 6, 2);
insert into Constitue values (7, 18, 6);
insert into Constitue values (7, 12, 2);
insert into Constitue values (7, 5, 6);
insert into Constitue values (7, 2, 4);
insert into Constitue values (7, 17, 20);
insert into Constitue values (7, 11, 3);
insert into Constitue values (7, 8, 7);
insert into Constitue values (7, 19, 24);
insert into Constitue values (7, 16, 6);
insert into Constitue values (7, 13, 10);
insert into Constitue values (7, 21, 24);
insert into Constitue values (7, 26, 16);

insert into Sert values (1, 1, 5.0);
insert into Sert values (2, 2, 5.0);
insert into Sert values (2, 3, 10.0);
insert into Sert values (3, 4, 2.0);
insert into Sert values (3, 5, 7.0);

```
insert into Sert values (3, 6, 20.0);
insert into Sert values (1, 7, 32.0);
```

Les commandes SQL répondants aux questions pertinentes :

```
/* Quel est la carte du restaurant ? */
select nom, categorie, prix from Plat
order by decode(categorie, 'entree', 1, 'resistance', 2, 'boisson', 3, 'dessert', 4);
```

```
/* Quel est le nombre de tables et de places maximum disponibles ? */
select count(idTable) as nb_table, sum(nbCouvert) as nb_places
from Tables;
```

```
/* Quelles sont les réservations ayant eu lieu le 25 décembre ? */
select * from Reservation where TO_CHAR(dateR, 'DD-MM') like '25-12';
```

```
/* Combien de tables sont occupées par une réservation ? */
select count(idTable) as nb_table from Affecter;
```

```
/* Quelle est l'heure du pic de commandes pour un plat donné ? */
select * from (
select p.nom, TO_CHAR(dateC, 'HH24') as heure, count(*) as nb_total from Commande com,
Constitue cons, Plat p
where com.idCommande = cons.idCommande and cons.idPlat = p.idPlat and p.nom = 'Poulet roti'
group by TO_CHAR(dateC, 'HH24'), p.nom);
```

```
/* Quels plats sont consommés dans tous les secteurs ? */
select pl.nom from
Reservation r, Possede p, Commande com, Constitue cons, Plat pl
where r.idReservation = p.idReservation and p.idCommande = com.idCommande and
com.idCommande = cons.idCommande and cons.idPlat = pl.idPlat
group by pl.nom
having count(r.type) = (select count(distinct type) from Reservation);
```

```
/* Combien de fois une table a été occupée le 29 décembre 2023 ? */
SELECT idTable, dateDebut, COUNT(*) as nombreReservations
FROM Affecter where dateDebut = TO_DATE('29-DEC-23', 'DD-MM-YYYY')
GROUP BY idTable, dateDebut;
```

```
/* Combien y a-t-il eu de réservations le soir ? */
select count(dateR) from Reservation
where TO_NUMBER(TO_CHAR(dateR, 'HH24')) between 11 and 14;
```

```
/* Combien y a-t-il eu de réservations le midi ? */
select count(dateR) from Reservation
where TO_NUMBER(TO_CHAR(dateR, 'HH24')) between 19 and 22;
```

```
/* Combien de commandes possède une entrée ? */
select count(com.idCommande) from Commande com, Constitue cons, Plat p
where com.idCommande = cons.idCommande and cons.idPlat = p.idPlat and categorie = 'entree';
```

```
/* Combien d'argent a rapporté le poulet rôti ? */  
select sum(p.prix * c.quantite) as argent_poulet_roti from plat p, Constitue c  
where c.idPlat = p.idPlat and p.nom = 'Poulet roti';
```

```
/* Combien d'argent ont rapporté les boissons ? */  
select sum(p.prix * c.quantite) as argent_boisson from plat p, Constitue c  
where c.idPlat = p.idPlat and p.categorie = 'boisson';
```

Création des utilisateurs :

```
create user admin identified by admin_mdp;  
create user patron identified by patron;  
create user employe1 identified by employe1;  
create user employe2 identified by employe2;  
create user employe3 identified by employe3;
```

Octroiement des privilèges :

```
grant all privilege to admin;  
grant select on recettes to patron;  
grant select on benefice_mois to patron;  
grant select on plat_populaire to patron;  
grant select on secteur_populaire to patron;  
grant select on avg_people to patron;
```

```
grant select on val_pourboire_1 to employe1;  
grant select on val_pourboire_2 to employe2;  
grant select on val_pourboire_3 to employe3;
```

```
grant select on plat_populaire to employe1;  
grant select on plat_populaire to employe2;  
grant select on plat_populaire to employe3;  
grant select on secteur_populaire to employe1;  
grant select on secteur_populaire to employe2;  
grant select on secteur_populaire to employe3;  
grant select on detail_employe_1 to employe1;  
grant select on detail_employe_2 to employe2;  
grant select on detail_employe_3 to employe3;  
grant insert, select, update, delete on Reservation to employe1;  
grant insert, select, update, delete on Reservation to employe2;  
grant insert, select, update, delete on Reservation to employe3;  
grant insert, select, update, delete on Reservation to patron;
```

Et pour finir, voici les procédures pour les métadonnées :

```
/* Procédure qui affiche la liste des contraintes */  
create or replace procedure liste_orc_constraints as  
cursor c1 is  
select ucc.table_name, uc.constraint_type, uc.search_condition  
from user_cons_columns ucc, user_constraints uc  
where ucc.constraint_name = uc.constraint_name  
order by ucc.table_name, uc.constraint_type;
```

```

begin
DBMS_OUTPUT.PUT_LINE('nom_table | type_contrainte | corps_contrainte');
for tuple in c1 loop
    if (tuple.table_name in ('RESERVATION', 'AFFECTER', 'TABLES', 'POSSEDE',
'COMMANDE', 'CONSTITUE', 'PLAT', 'SERT', 'EMPLOYE')) then
        DBMS_OUTPUT.PUT_LINE(tuple.table_name || ' | ' || tuple.constraint_type || ' | ' ||
tuple.search_condition);
    end if;
end loop;
end;
/

```

```

/* Procédure qui affiche la liste des triggers */
create or replace procedure liste_ora_triggers as
cursor c2 is
select at.table_name, at.trigger_name
from all_triggers at, all_tables nom
where at.table_name = nom.table_name
order by at.table_name;
begin
    DBMS_OUTPUT.PUT_LINE('nom_table | nom_triggers');
for tuple in c2 loop
    if (tuple.table_name in ('RESERVATION', 'AFFECTER', 'TABLES', 'POSSEDE',
'COMMANDE', 'CONSTITUE', 'PLAT', 'SERT', 'EMPLOYE')) then
        DBMS_OUTPUT.PUT_LINE(tuple.table_name || ' | ' || tuple.trigger_name);
    end if;
end loop;
end;
/

```

```

/* Procédure qui affiche la liste des vues */
create or replace procedure liste_ora_views as
cursor c3 is
select view_name from user_views
order by view_name;
begin
    DBMS_OUTPUT.PUT_LINE('nom_vue');
for tuple in c3 loop
    DBMS_OUTPUT.PUT_LINE(tuple.view_name);
end loop;
end;
/

```

```

/* Procédure qui affiche la liste des utilisateurs et leur privilèges */
create or replace procedure liste_ora_users as
cursor c4 is
select grantee, privilege, table_name, grantor
from USER_TAB_PRIVS;
begin
for tuple in c4 loop
    DBMS_OUTPUT.PUT_LINE('Utilisateur : ' || tuple.grantee || ' a ' || tuple.privilege || ' sur la
table ' || tuple.table_name);
end loop;
end;
/

```



```
    end loop;  
end;  
/
```

SQL Loader :

Nous avons réussi à mettre en place tout les dossiers nécessaires au bon fonctionnement du loader. Pour chaque table, un fichier .par, .ctl et .dat ont été créé. Malheureusement, lors de l'exécution de la commande sqldr permettant d'insérer les données, une erreur est retournée en boucle, indiquant qu'il est impossible d'ouvrir le fichier .ctl.

Exemple pour la table Employe :

-Pour le fichier .dat :

```
NULL$ 'Dupont'$ 'Jean'$ '0623456789'$ 1500.0$ TO_DATE('10-08-2023', 'DD-MM-YYYY')$ '123  
rue de la Paix'$ 'Paris'$ '75000'$ 'jean.dupont@gmail.com'  
NULL$ 'Martin'$ 'Marie'$ '0687654321'$ 1700.0$ TO_DATE('15-09-2023', 'DD-MM-YYYY')$  
'456 boulevard Saint-Germain'$ 'Paris'$ '75000'$ 'marie.martin@gmail.com'  
NULL$ 'Constant'$ 'David'$ '0654216810'$ 2000.0$ TO_DATE('04-07-2023', 'DD-MM-YYYY')$  
'112 rue de la Paix'$ 'Paris'$ '75000'$ Constant.David@gmail.com'
```

-Pour le fichier .ctl :

```
load data into table Employe  
insert  
fields terminated by "$"  
(  
idEmploye,  
nom,  
prenom,  
telephone,  
salaire,  
dateEmbauche,  
rue,  
ville,  
codePostal,  
mail  
)
```

-Pour le fichier .par :

```
userid=etudiant/etudiant  
control=Employe.ctl  
log=Employe.log  
bad=Employe.bad  
data=Employe.dat  
direct=true
```