DESCRIPTION DES CHOIX EFFECTUES :

Le remplissage des tables a été effectué par un script codé en java.

Les valeurs d'idVoyageur et idUtilisateur que l'on peut trouver dans certaines requêtes ont été choisies arbitrairement pour que nous puissions tester les requêtes. En pratique, elles seraient remplacées par des variables fournies via PHP (par exemple) représentant l'utilisateur connecté.

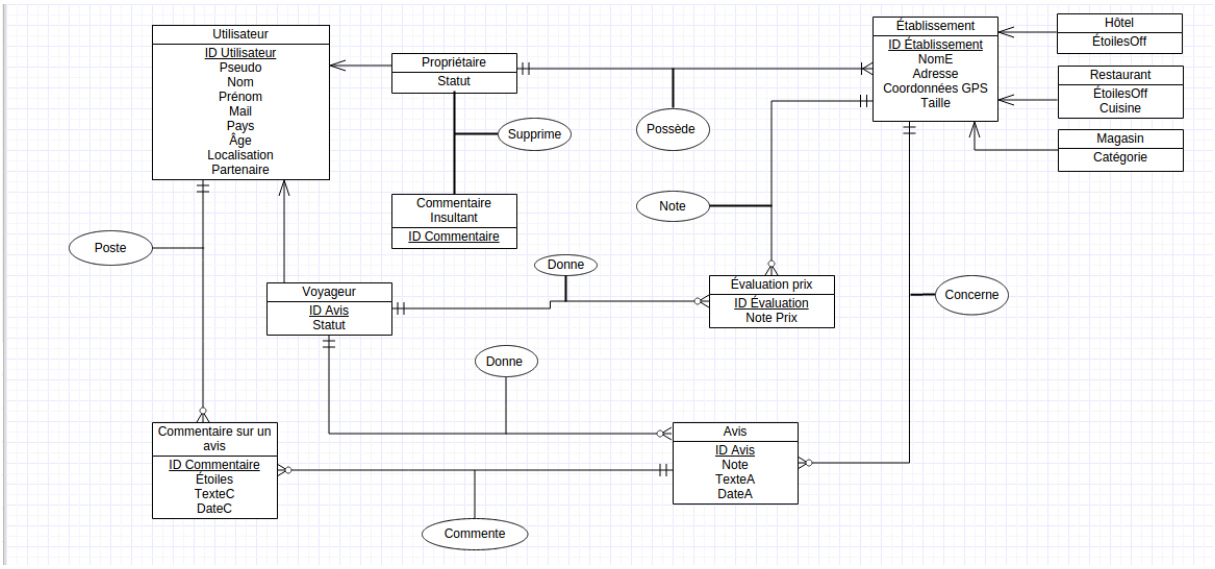
De la même manière, la localisation et le partenaire seraient récupérés et actualisés de manière dynamique.

Les clés primaires de certaines tables sont différentes de celles présentes sur le schéma de modélisation : il a fallu ajouter les clés étrangères qui sont également des clés primaires. Prenons comme exemple la table Voyageur : si idUtilisateur est l'unique clé primaire, alors chaque Voyageur ne pourra donner qu'un seul avis. Par conséquent, la clé primaire de Voyageur est composée des champs idUtilisateur et idAvis.

Vis-à-vis de la modélisation, nous avons modifié quelques noms de champ de la table commentaireAvis et de la table Avis. Le champ "texte de commentaire" devient "texteC", le champ "texte d'avis" devient "texteA". La même modification s'applique aux dates : "date de commentaire" devient "dateC" et "date d'avis" devient "dateA". Cela nous permet d'éviter les clauses ambigües lors des jointures de tables.

La même modification s'applique dans la table Etablissement, où "nom" devient "nomE", ainsi que dans les tables Hotel et Restaurant, où "etoiles" devient "etoilesoff".

MODELISATION FINAL :




LISTES DES TABLES :

Utilisateur
Voyageur
Proprietaire
Etablissement
Restaurant
Hotel
Magasin
Avis
CommentaireAvis
EvaluationPrix
CommentaireInsultant

CREATION DES TABLES :

SET foreign_key_checks =0; drop table if exists Utilisateur; drop table if exists Voyageur; drop table if exists Proprietaire; drop table if exists Etablissement; drop table if exists Restaurant; drop table if exists Hotel; drop

```
table if exists Magasin; drop table if exists Avis; drop table if exists EvaluationPrix; drop table if exists CommentaireInsultant; drop table if exists CommentaireAvis;


CREATE TABLE Utilisateur(idUtilisateur int (11) Auto_increment  NOT NULL , pseudo Varchar (25) NOT NULL , prenom Varchar (50) NOT NULL , nom Varchar (50) NOT NULL , email Varchar (60) NOT NULL , pays Varchar (40) , age TinyINT NOT NULL , Localisation Int NOT NULL, Partenaire Varchar (50) NOT NULL ,idCommentaire Int NOT NULL , PRIMARY KEY (idUtilisateur ) , UNIQUE (email) )ENGINE=InnoDB;


CREATE TABLE Voyageur(idUtilisateur Int NOT NULL, statut ENUM('standard','bronze','argent','or') NOT NULL, idAvis Int NOT NULL ,PRIMARY KEY (idUtilisateur,idAvis ))ENGINE=InnoDB;


CREATE TABLE Proprietaire(idUtilisateur Int NOT NULL, statut ENUM('standard','bronze','argent','or') NOT NULL, idEtablissement Int NOT NULL, idCommentaire Int NOT NULL, PRIMARY KEY (idUtilisateur,idEtablissement))ENGINE=InnoDB;


CREATE TABLE Etablissement(idEtablissement int (11) Auto_increment  NOT NULL, nomE Varchar (50) NOT NULL, adresse Varchar (100) NOT NULL, coordoneesGPS Int NOT NULL, taille Smallint NOT NULL, idAvis Int NOT NULL , PRIMARY KEY (idEtablissement,idAvis))ENGINE=InnoDB;


CREATE TABLE Restaurant( etoileOff ENUM('0','1','2','3','4','5','6','7') NOT NULL, cuisine Varchar (35) NOT NULL , idEtablissement Int NOT NULL, PRIMARY KEY (idEtablissement))ENGINE=InnoDB;


CREATE TABLE Hotel(etoileOff ENUM('0','1','2','3','4','5','6','7') NOT NULL, idEtablissement Int NOT NULL, PRIMARY KEY (idEtablissement))ENGINE=InnoDB;


CREATE TABLE Magasin(categorie Varchar (45) NOT NULL, idEtablissement Int NOT NULL, PRIMARY KEY (idEtablissement ))ENGINE=InnoDB;


CREATE TABLE Avis( idAvis int (11) Auto_increment  NOT NULL, Note ENUM('0','1','2','3','4','5') NOT NULL, texteA Varchar (255), dateA Date NOT NULL , idCommentaire Int NOT NULL, PRIMARY KEY (idAvis ))ENGINE=InnoDB;


CREATE TABLE CommentaireAvis( idCommentaire int (11) Auto_increment  NOT NULL, Etoiles ENUM('1','2','3','4','5') NOT NULL, TexteC Varchar (255) NOT NULL, Date Date NOT NULL, PRIMARY KEY (idCommentaire))ENGINE=InnoDB;


CREATE TABLE EvaluationPrix( notePrix ENUM('1','2','3','4','5') NOT NULL, idEtablissement Int NOT NULL, idUtilisateur Int NOT NULL, PRIMARY KEY (idEtablissement,idUtilisateur))ENGINE=InnoDB;


CREATE TABLE CommentaireInsultant( idCommentaire Int NOT NULL, PRIMARY KEY (idCommentaire ))ENGINE=InnoDB;


ALTER TABLE Utilisateur ADD CONSTRAINT FK_Utilisateur_idCommentaire FOREIGN KEY (idCommentaire) REFERENCES CommentaireAvis(idCommentaire);

ALTER TABLE Voyageur ADD CONSTRAINT FK_Voyageur_idUtilisateur FOREIGN KEY (idUtilisateur) REFERENCES Utilisateur(idUtilisateur);

ALTER TABLE Voyageur ADD CONSTRAINT FK_Voyageur_idAvis FOREIGN KEY (idAvis)
```

```
REFERENCES Avis(idAvis);

ALTER TABLE Proprietaire ADD CONSTRAINT FK_Proprietaire_idUtilisateur FOREIGN KEY
(idUtilisateur) REFERENCES Utilisateur(idUtilisateur);

ALTER TABLE Proprietaire ADD CONSTRAINT FK_Proprietaire_idEtablissement FOREIGN KEY
(idEtablissement) REFERENCES Etablissement(idEtablissement);

ALTER TABLE Proprietaire ADD CONSTRAINT FK_Proprietaire_idCommentaire FOREIGN KEY
(idCommentaire) REFERENCES CommentaireInsultant(idCommentaire);

ALTER TABLE Etablissement ADD CONSTRAINT FK_Etablissement_idAvis FOREIGN KEY (idAvis)
REFERENCES Avis(idAvis);

ALTER TABLE Restaurant ADD CONSTRAINT FK_Restaurant_idEtablissement FOREIGN KEY
(idEtablissement) REFERENCES Etablissement(idEtablissement);

ALTER TABLE Hotel ADD CONSTRAINT FK_Hotel_idEtablissement FOREIGN KEY (idEtablissement)
REFERENCES Etablissement(idEtablissement);

ALTER TABLE Magasin ADD CONSTRAINT FK_Magasin_idEtablissement FOREIGN KEY
(idEtablissement) REFERENCES Etablissement(idEtablissement);

ALTER TABLE Avis ADD CONSTRAINT FK_Avis_idCommentaire FOREIGN KEY (idCommentaire)
REFERENCES CommentaireAvis(idCommentaire);

ALTER TABLE EvaluationPrix ADD CONSTRAINT FK_EvaluationPrix_IdUtilisateur FOREIGN KEY
(IdUtilisateur) REFERENCES Voyageur(IdUtilisateur);

ALTER TABLE EvaluationPrix ADD CONSTRAINT FK_EvaluationPrix_IdEtablissement FOREIGN KEY
(IdEtablissement) REFERENCES Etablissement(IdEtablissement);
```

## LISTE DES REQUETES :

1

```
DELETE FROM Utilisateur
      WHERE idUtilisateur = 15;

   --INSERT INTO Utilisateur VALUES (0, 'Max', 'GARNIER', 'Thomas', 'Thomas@orange.com', 'Mandres', 22, 'Basebook', 56);
   --La ligne ci-dessus permet de réinsérer la quinzième entrée (qui ne sera du coup plus la quinzième mais la dernière).
```

2 trié par EvalPrix descendant et apres par etoileoff pour des restaurant parisien mais plusieurs evalPrix pour le meme restaurant

```
SELECT nomE, adresse
        FROM etablissement natural join evaluationPrix natural join restaurant
        where adresse like '%Paris'
        order by notePrix, etoileOff desc;
```

3

```
(SELECT nomE,idEtablissement FROM  Etablissement NATURAL JOIN proprietaire natural join restaurant where statut='or' and Adresse LIKE '%Paris%')
union
(SELECT nomE,idEtablissement FROM  Etablissement NATURAL JOIN proprietaire natural join restaurant where statut='argent' and Adresse LIKE '%Paris%')
union
(SELECT nomE,idEtablissement FROM  Etablissement NATURAL JOIN proprietaire natural join restaurant where statut='bronze' and Adresse LIKE '%Paris%')
union
(SELECT nomE,idEtablissement FROM  Etablissement NATURAL JOIN proprietaire natural join restaurant where statut='standard' and Adresse LIKE '%Paris%');
```

4 en supposant que utilisateur est a moins de cinq kilometre de bordeaux

```
select nomE  from magasin natural join etablissement natural join avis where categorie="bricolage" and Note>3 and adresse like '%Bordeaux';
```

5

```
SELECT nomE, adresse, note FROM Etablissement NATURAL JOIN Restaurant NATURAL JOIN Avis
      WHERE Adresse LIKE '%Paris%'
        AND cuisine = 'indien'
        AND Note > 4
   ORDER BY Note DESC;
```

6

```
select nomE,adresse
        from hotel
        natural join etablissement natural join avis
        where note>3 and etoileOff>2;
```

7

```
INSERT INTO Avis VALUES (270, 3, 'Super !', '13-08-2014', 2);
```

8 requete que pour utilisateur 20

```
select nomE
        from etablissement natural join voyageur natural join avis
        where dateA like '2012%'and idUtilisateur=20;
```

9

```
select idEtablissement,idUtilisateur,idAvis,datea,note from avis a1 natural join voyageur v2 natural join etablissement t1

where exists (


    (select idEtablissement
    from avis a2 natural join voyageur v1 natural join etablissement t2
    where t1.idEtablissement = t2.idEtablissement and
    v1.idUtilisateur = v2.idUtilisateur

     and datediff(a1.datea,a2.datea) between -365 and 365

    group by idEtablissement
    having count(*)>1)) order by idEtablissement, idUtilisateur asc;
```

10 jimpose annnee=2010

```
SELECT idUtilisateur
FROM avis natural join voyageur
WHERE(
    EXISTS (
      SELECT * FROM avis nautral join voyageur
      WHERE
        datea >= "2010-01-01" && datea<= "2010-01-31"
    ) &&
         EXISTS (
      SELECT * FROM avis nautral join voyageur
      WHERE
        datea >= "2010-02-01" && datea <= "2010-02-31"
    ) &&
         EXISTS (
      SELECT * FROM avis nautral join voyageur
      WHERE
        datea >= "2010-03-01" && datea <= "2010-03-31"
    ) &&
         EXISTS (
      SELECT * FROM avis nautral join voyageur
      WHERE
        datea >= "2010-04-01" && datea <= "2010-04-31"
    ) &&
         EXISTS (
      SELECT * FROM avis nautral join voyageur
      WHERE
        datea >= "2010-05-01" && datea <= "2010-05-31"
    ) &&
      EXISTS (
      SELECT * FROM avis nautral join voyageur
      WHERE
        datea >= "2010-06-01" && datea <= "2010-06-31"
    ) &&    EXISTS (
      SELECT * FROM avis nautral join voyageur
      WHERE
        datea >= "2010-07-01" && datea <= "2010-07-31"
    ) &&    EXISTS (
      SELECT * FROM avis nautral join voyageur
      WHERE
        datea >= "2010-08-01" && datea <= "2010-08-31"
    ) &&    EXISTS (
```

```
      SELECT * FROM avis nautral join voyageur
      WHERE
        datea >= "2010-09-01" && datea <= "2010-09-31"
    ) &&    EXISTS (
      SELECT * FROM avis nautral join voyageur
      WHERE
        datea >= "2010-10-01" && datea <= "2010-10-31"
    ) &&    EXISTS (
      SELECT * FROM avis nautral join voyageur
      WHERE
        datea >= "2010-11-01" && datea <= "2010-11-31"
    ) &&    EXISTS (
      SELECT * FROM avis nautral join voyageur
      WHERE
        datea >= "2010-12-01" && datea <= "2010-12-31"
    )group by idUtilisateur having count(datea)>10;
```

11

```
 --Affichage note moyenne
    SELECT AVG(Note) FROM Avis NATURAL JOIN Etablissement
      WHERE idEtablissement = 42;

    --Affichage évaluation moyenne du prix
    SELECT AVG(NotePrix) FROM EvaluationPrix NATURAL JOIN Avis NATURAL JOIN Etablissement
      WHERE idEtablissement = 42;
```

12 le proprietaire 71 a deux etablissement (75 a 5 etablissement)

```
select nomE,adresse,idAvis,Note,dateA,texteA
        from avis natural join etablissement
        where idEtablissement in (
                select idEtablissement
                from proprietaire natural join etablissement
                where proprietaire.idEtablissement = etablissement.idEtablissement and idutilisateur=71
        );
```

13

```
--soit on sélectionne par ville, par exemple Paris :
    SELECT nomE
    FROM Etablissement
    WHERE adresse LIKE '%Paris%' and idEtablissement in (
        select idEtablissement
                from proprietaire natural join etablissement
                where proprietaire.idEtablissement = etablissement.idEtablissement and idutilisateur!=75
    );

    --soit on sélectionne par type d'Etablissement et par ville :
    SELECT nomE
        FROM Etablissement NATURAL JOIN Restaurant
        WHERE adresse LIKE '%Paris%' and cuisine='chinois'and idEtablissement in (
                select idEtablissement
                        from proprietaire natural join etablissement
                        where proprietaire.idEtablissement = etablissement.idEtablissement and idutilisateur!=75
        );
```

14

```
select Note,nomE,adresse,texteA
        from avis natural join etablissement
        where idEtablissement in (
                select idEtablissement
                from proprietaire natural join etablissement
                where proprietaire.idEtablissement = etablissement.idEtablissement and idutilisateur=75
        )
        group by nomE;
```

15

```
select nomE,adresse
        from avis a1 natural join etablissement
        where
                exists (
                        select*
                                from avis a2 natural join etablissement
                                where a2.note>='4' and a1.note>='4'
                )
                and idEtablissement in (
                        select idEtablissement
                                from proprietaire natural join etablissement
                                where proprietaire.idEtablissement = etablissement.idEtablissement and idutilisateur=75
                )
        group by nomE having count(*)>1;
```

16 PAS FINI proprietaire faisant la demande est le 73

```
--LE MIEUX le manque le tri en fonction de la note
select idEtablissement,idAvis,idUtilisateur,datea,note
        from avis a2 natural join voyageur v2 natural join etablissement t1
        where exists (
                select *
                        from avis a1 natural join voyageur v1 natural join etablissement t2
                        where t1.idEtablissement = t2.idEtablissement and v1.idUtilisateur = v2.idUtilisateur and idEtablissement in (
                                select idEtablissement
                                        from etablissement natural join proprietaire
                                        where proprietaire.idUtilisateur =73
                                )
                                and datediff(a1.datea,a2.datea) between -730 and 730
                        group by idEtablissement  having count(*)>1);
```

17

```
UPDATE Voyageur SET statut = 'or'
      WHERE idUtilisateur = 42;
```

18

```
SET foreign_key_checks =0;
DELETE FROM avis WHERE idAvis='14';
DELETE FROM avis WHERE idAvis='89';
insert into avis values(14,3,"Je le conseil vivement pour une famille",'2013-01-04',58);
insert into avis values(89,2,"a part quelques petits defauts cest un sans faute",'2013-08-04',92);
```

19

```
--Soit on supprime un commentaire dont on a l'id :
    DELETE FROM CommentaireInsultant WHERE idCommentaire = 52;

--Soit on supprime tous les commentaires insultants d'un seul coup en effaçant toute la table (l'auto incrément se réinitialise automatiquement):
    TRUNCATE TABLE CommentaireInsultant;

--Soit on supprime tous les commentaires insultants d'un seul coup en effaçant toute la table :
SET foreign_key_checks =0;
drop table if exists CommentaireInsultant;
    CREATE TABLE CommentaireInsultant(
        idCommentaire Int NOT NULL ,
        PRIMARY KEY (idCommentaire )
    )ENGINE=InnoDB;
```

20

```
select nom,Prenom from commentaireAvis natural join voyageur natural join utilisateur where etoiles>=4 group by(email) order by (nom) asc;
```

21

```
    DELETE FROM Voyageur
      WHERE (((SELECT COUNT(Etoiles) FROM CommentaireAvis
        WHERE (Etoiles < '2' AND idUtilisateur = 42))/(SELECT COUNT(Etoiles) FROM CommentaireAvis
          WHERE idUtilisateur = 42))*100) > '30';
    INSERT INTO Utilisateur VALUES (0, 'Jean_Kévin', 'BERTHELOT', 'Bob', 'bob-bert@free.fr', 'Mandres', 22, 'Basebook', 56);
```

22

```
// repere si il y a une difference
select case when (
        select count(distinct(email))
                from voyageur natural join avis natural join utilisateur
        )
        !=
        (select count(distinct(localisation))
                from voyageur natural join avis natural join utilisateur
        )
        then 'tricheur'
        else 'ras'
        end;

// permet de connaitre les idUtilisateur utilisant la meme localisation
select idUtilisateur,localisation
        from utilisateur t1 natural join avis natural join voyageur
        where exists (
                select *
                        from utilisateur t2 natural join avis natural join voyageur
                        where t1.localisation = t2.localisation
                        group by localisation
                        having count(*)>1
        );
```

23

```
drop view if exists cinq;
create view cinq as
(select idEtablissement,avg(note) as diff
                    from etablissement natural join avis natural join restaurant
                    group by idEtablissement)
                    union
                    (select idEtablissement,avg(note) as diff
                    from etablissement natural join avis natural join hotel
                    group by idEtablissement)
                    union
                    (select idEtablissement,avg(note) as diff
                    from etablissement natural join avis natural join magasin
                    group by idEtablissement);
select idEtablissement, max(diff) from cinq where diff=6 group by idEtablissement;
```

24

```
select email,nom,prenom
        from avis natural join voyageur natural join utilisateur natural join etablissement natural join restaurant
        where etoileoff>7 and email in(
                select email
                        from avis natural join voyageur natural join utilisateur natural join etablissement natural join hotel
                        where etoileoff>7
        )

//si la requete etait un restaurant OU un hotel
select nom,prenom,email
        from avis natural join voyageur natural join utilisateur natural join etablissement natural join restaurant
        where etoileoff>7
union
select nom,prenom,email
        from avis natural join voyageur natural join utilisateur natural join etablissement natural join hotel
        where etoileoff>7;
```

25

```
drop view if exists quatre;
create view quatre as (
        select idEtablissement, abs(avg(note)-etoileoff) as diff
                from avis natural join etablissement natural join restaurant natural join voyageur
                group by idEtablissement
)
union
(select idEtablissement, abs(avg(note)-etoileoff) as diff
        from avis natural join etablissement natural join hotel natural join voyageur
        group by idEtablissement
);
select idEtablissement, max(diff) from quatre where diff=7 group by idEtablissement;
```