

Conception du projet d'annonces immobilières JEE

Master Data science et Big Data

Réalisé par : BERRADA Anas

Encadre par : BELANGOUR Abdessamad

Chapitre 3 : Conception

1 – introduction

Ce chapitre sera consacré à la conception de notre site web, en présentant le modèle de la base de données ainsi que les architecture du travail du projet

2 – Conception

2.1. Génération de la base de données

La base de données est le noyau de chaque application, donc pour la générer nous nous basons sur le diagramme de classe en utilisant Visual Paradigme.

Donc ce code SQL représente la génération complète de la base de données.

```
/*=====*/  
/* Nom de SGBD : MySQL 5.0 */  
/* Date de création : 17/01/2020 13:52:32 */  
/*=====*/  
  
drop table if exists Admin;  
  
drop table if exists Adresse;  
  
drop table if exists Annonce;  
  
drop table if exists Appart;  
  
drop table if exists BurLocCom;  
  
drop table if exists Favoris;
```

drop table if exists Immobilier;

drop table if exists Location;

drop table if exists LocationVacances;

drop table if exists Message;

drop table if exists TerFer;

drop table if exists User;

drop table if exists Vente;

drop table if exists VilMasLuxRiadi;

drop table if exists envoyer;

drop table if exists recevoir;

```
/*=====*/
/* Table : Admin                                */
/*=====*/
```

```
create table Admin
(
    idUser      int not null,
    idAdmin     int not null,
    primary key (idUser, idAdmin)
);
```

```
/*=====*/
/* Table : Adresse                             */
/*=====*/
```

```
create table Adresse
(
    idAdr       int not null,
    rue         varchar(254),
    numero      int,
    ville       varchar(254),
    arrondissement varchar(254),
    quartier    varchar(254),
    primary key (idAdr)
);
```

```
/*=====*/
/* Table : Annonce                             */
/*=====*/
```

```
create table Annonce
(
    idAnn       int not null,
    idImmo      int not null,
    titre       varchar(254),
```

```

description      varchar(254),
date             datetime,
duree            int,
nbrVue           int,
primary key (idAnn)
);

/*=====*/
/* Table : Appart */
/*=====*/
create table Appart
(
    idImmo         int not null,
    idApp          int not null,
    salleBain      int,
    chambre        int,
    piece          int,
    etage          int,
    balcon         bool,
    parking        bool,
    cuisine        bool,
    primary key (idImmo, idApp)
);

/*=====*/
/* Table : BurLocCom */
/*=====*/
create table BurLocCom
(
    idImmo         int not null,
    idBl           int not null,
    salleBain      int,
    numEtage       int,
    primary key (idImmo, idBl)
);

/*=====*/
/* Table : Favoris */
/*=====*/
create table Favoris
(
    idUser         int not null,
    idAnn          int not null,
    idFavoris      int not null,
    etatfav       int,
    primary key (idUser, idAnn)
);

/*=====*/
/* Table : Immobilier */
/*=====*/
create table Immobilier
(

```

```

idImmo          int not null,
idAdr           int not null,
surface         float,
prix            int,
etat            char(8),
image           varchar(254),
primary key (idImmo)
);

/*=====*/
/* Table : Location */
/*=====*/
create table Location
(
  idAnn          int not null,
  idLoc          int not null,
  primary key (idAnn, idLoc)
);

/*=====*/
/* Table : LocationVacances */
/*=====*/
create table LocationVacances
(
  idAnn          int not null,
  idLocVac       int not null,
  primary key (idAnn, idLocVac)
);

/*=====*/
/* Table : Message */
/*=====*/
create table Message
(
  idMsg          int not null,
  content        varchar(254),
  date           datetime,
  etat           bool,
  primary key (idMsg)
);

/*=====*/
/* Table : TerFer */
/*=====*/
create table TerFer
(
  idImmo         int not null,
  idTf           int not null,
  type           varchar(254),
  primary key (idImmo, idTf)
);

/*=====*/

```

```

/* Table : User                                     */
/*=====*/
create table User
(
    idUser          int not null,
    nom             varchar(254),
    prenom          varchar(254),
    age             int,
    email           varchar(254),
    mdp             varchar(254),
    primary key (idUser)
);

/*=====*/
/* Table : Vente                                     */
/*=====*/
create table Vente
(
    idAnn           int not null,
    idVente         int not null,
    primary key (idAnn, idVente)
);

/*=====*/
/* Table : VilMasLuxRiadi                           */
/*=====*/
create table VilMasLuxRiadi
(
    idImmo          int not null,
    idVmr           int not null,
    salleBain       int,
    chambre         int,
    piece           int,
    primary key (idImmo, idVmr)
);

/*=====*/
/* Table : envoyer                                   */
/*=====*/
create table envoyer
(
    idUser          int not null,
    idMsg           int not null,
    primary key (idUser, idMsg)
);

/*=====*/
/* Table : recevoir                                 */
/*=====*/
create table recevoir
(
    idMsg           int not null,
    idUser          int not null,

```

```

primary key (idMsg, idUser)
);

alter table Admin add constraint FK_Generalisation_8 foreign key (idUser)
references User (idUser) on delete restrict on update restrict;

alter table Annonce add constraint FK_conserver foreign key (idImmo)
references Immobilier (idImmo) on delete restrict on update restrict;

alter table Appart add constraint FK_Generalisation_3 foreign key (idImmo)
references Immobilier (idImmo) on delete restrict on update restrict;

alter table BurLocCom add constraint FK_Generalisation_1 foreign key (idImmo)
references Immobilier (idImmo) on delete restrict on update restrict;

alter table Favoris add constraint FK_favoris foreign key (idAnn)
references Annonce (idAnn) on delete restrict on update restrict;

alter table Favoris add constraint FK_favoris foreign key (idUser)
references User (idUser) on delete restrict on update restrict;

alter table Immobilier add constraint FK_avoir foreign key (idAdr)
references Adresse (idAdr) on delete restrict on update restrict;

alter table Location add constraint FK_Generalisation_6 foreign key (idAnn)
references Annonce (idAnn) on delete restrict on update restrict;

alter table LocationVacances add constraint FK_Generalisation_7 foreign key (idAnn)
references Annonce (idAnn) on delete restrict on update restrict;

alter table TerFer add constraint FK_Generalisation_4 foreign key (idImmo)
references Immobilier (idImmo) on delete restrict on update restrict;

alter table Vente add constraint FK_Generalisation_5 foreign key (idAnn)
references Annonce (idAnn) on delete restrict on update restrict;

alter table VilMasLuxRiadi add constraint FK_Generalisation_2 foreign key (idImmo)
references Immobilier (idImmo) on delete restrict on update restrict;

alter table envoyer add constraint FK_envoyer foreign key (idMsg)
references Message (idMsg) on delete restrict on update restrict;

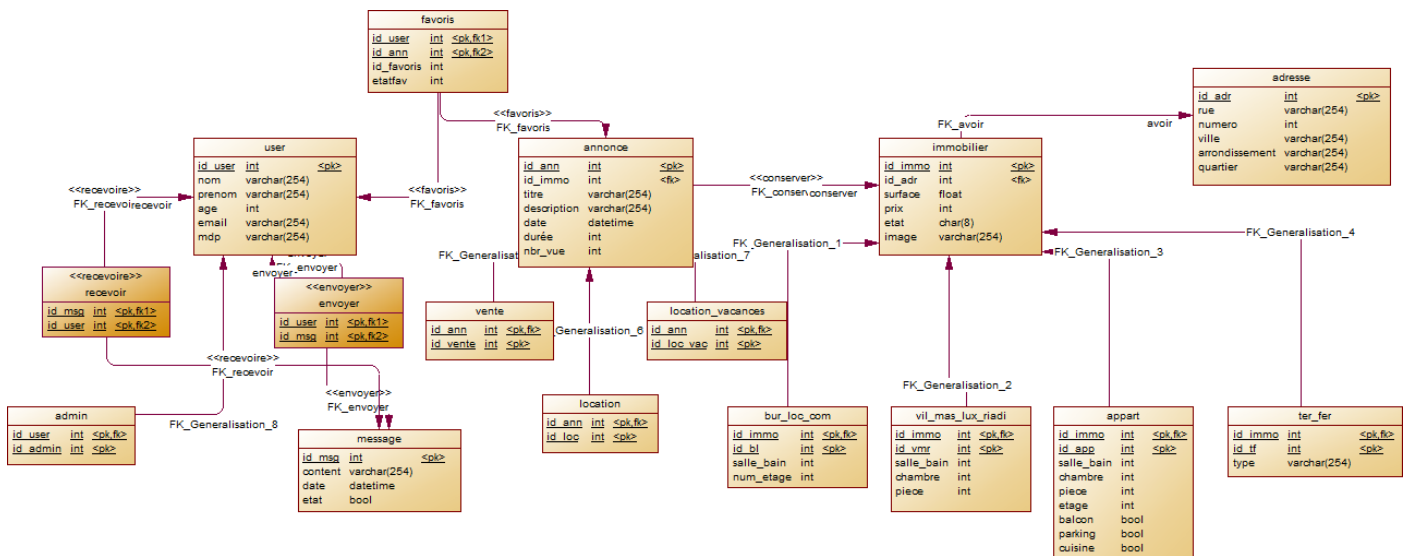
alter table envoyer add constraint FK_envoyer foreign key (idUser)
references User (idUser) on delete restrict on update restrict;

alter table recevoir add constraint FK_recevoir foreign key (idMsg)
references Message (idMsg) on delete restrict on update restrict;

alter table recevoir add constraint FK_recevoir foreign key (idUser)
references User (idUser) on delete restrict on update restrict;

```

2.2. diagramme physique de données



2.3. Résumé

Cette partie nous montre les différentes classes se génère dans un code du format SQL qui nous donne la main de l'exploiter en tout fiabilité. Ce code est généré automatiquement à partir d'un logiciel de modélisation et de conception.

2.4. Architecture physique adoptée

Dans notre application, l'architecture physique qu'on a adoptée se base sur deux serveurs :

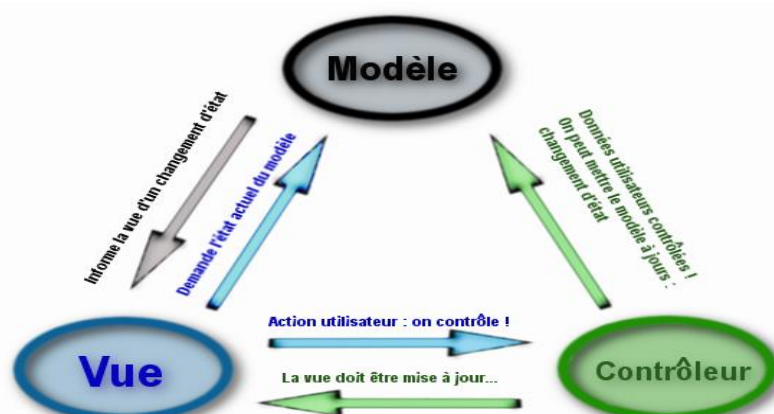
- 🚀 Serveur de base de données -MySQL : MySQL est un serveur de bases de données relationnelles
- 🚀 Serveur HTTP - Apache : Apache HTTP Server est un serveur HTTP créé et maintenu au sein de la fondation Apache. C'est le serveur HTTP le plus populaire du WorldWideWeb.

2.5. Architecture logique adoptée

2.5.1. Design pattern MVC

Modèle-vue-contrôleur ou MVC est un motif d'architecture logicielle destiné aux interfaces graphiques lancé en 1978 et très populaire pour les applications web. Le motif est composé de trois types de modules ayant trois responsabilités différentes : les modèles, les vues et les contrôleurs.

- ✓ Un modèle (Model) : contient les données à afficher.
- ✓ Une vue (View) : contient la présentation de l'interface graphique.
- ✓ Un contrôleur (Controller) : contient la logique concernant les actions effectuées par l'utilisateur.



2.6. Conclusion

Ce chapitre a été consacré la conception qui a été mise en œuvre tout au long de la réalisation de ce projet. Cette étape de creation du projet est vraiment critique pour le bon fonctionnement de l'application après et avant la réalisation.