

SAÉ 3.01 A&D : Développement d'application et Gestion de projet

Pôle gestion de données (C4)

Projet n°21 - BonPlan&Co

Sommaire

- **Présentation des membres de l'équipe**
- **Notre projet**
- **Les objectifs du projet**
- **Utilisateurs impliqués dans le projet**
- **Nos requêtes significatives**

Les membres du groupe

- ❖ Timéo JULLIARD (Product owner + développeur)
- ❖ Xan SALLENAVE (Product owner + développeur)
- ❖ Elsa LAVERGNE (Développeuse)
- ❖ Virgile ESPINASSE (Scrum master + développeur)
- ❖ Nicolas CONGUISTI (Scrum master + développeur)



Notre projet : *BonPlan & Co*

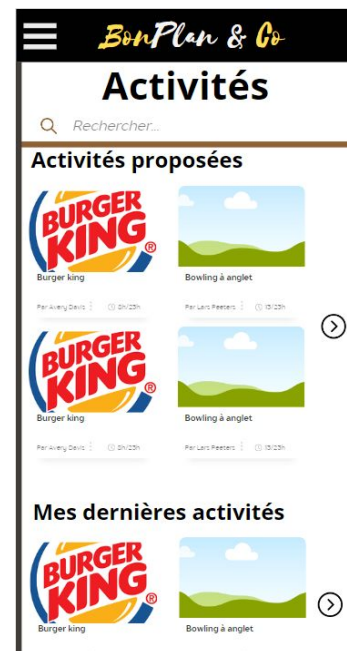
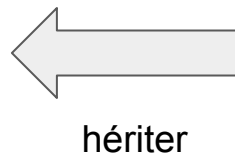
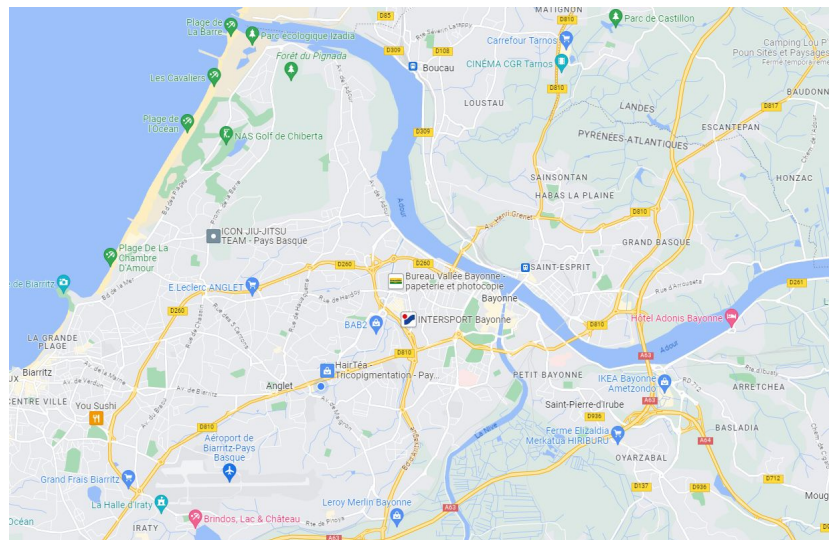
- ❑ Application web développée par des étudiants pour des étudiants sur le BAB
- ❑ Offre d'une multitude de fonctionnalités
 - ❑ Recherche d'activités (tourisme, sport, culture, restaurants, etc...)
 - ❑ Recherche d'événements (activités importantes à durée limitée)
 - ❑ Recherche d'offres d'emplois, de covoiturage, de colocation
 - ❑ Système de commentaires écrits par les utilisateurs pour chaque bon plan
- ❑ Carte interactive répertoriant tous les bons plans



Objectif du projet

Proposer des bons plans

Présenter le BAB



Utilisateurs impliqués dans le projet

Étudiants



Pouvant être groupés par bahut

Administrateurs



n'étant pas grouper car trop disperser

Pourquoi ces requêtes :



Requête 1 (multi-jointure):

Description :

>On sélectionne les adresses des utilisateurs ayant participé à une activité

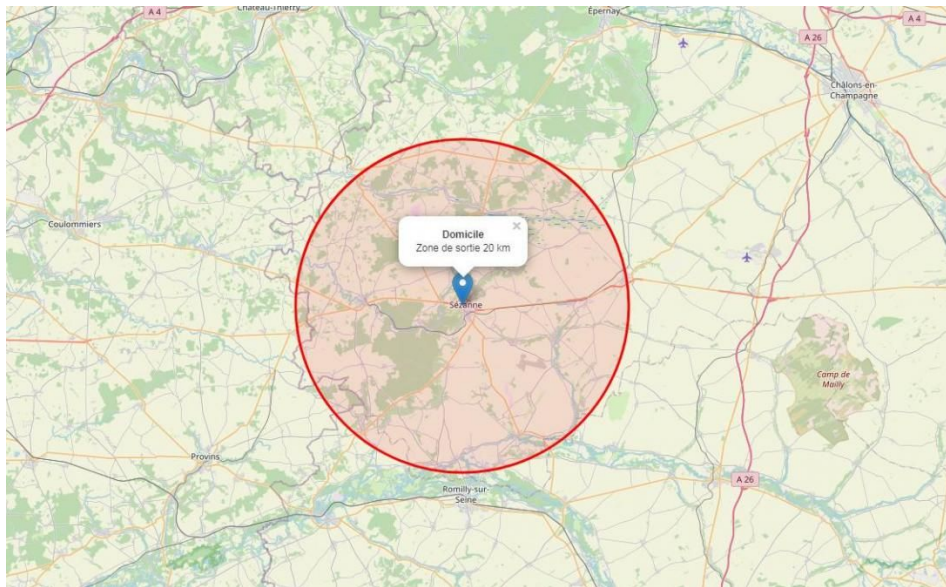
Utilité :

>Permet de déterminer si les gens choisissent leur activité en fonction de leur localisation ou non

Utilisateur concerné :

>Etudiant

Cela pourra devenir un facteur dans l'algorithme de recommandation plus tard. Si un utilisateur semble plus enclin à choisir des activités proches de chez lui, alors il faudrait lui proposer idéalement celles qui le sont.



Requête 2 (agrégation):

Description :

>Nombre de participant pour chaque événement

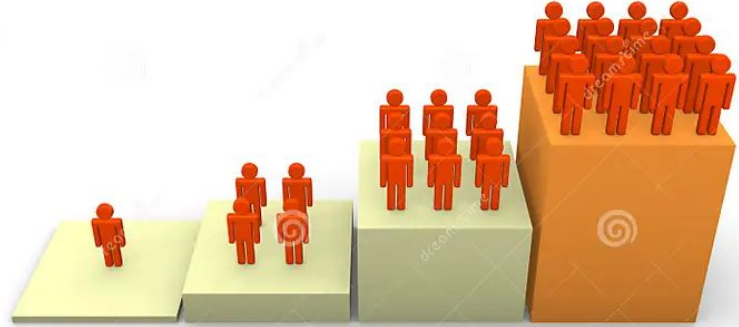
Utilité :

>Voir la popularité d'un événement, voir s'il touche beaucoup de monde ou non.

Cible :

>Tous les étudiants.

Cette requête permettrait de proposer les événements les plus populaires en règle générale.



Requête 3 (order by):

Description :

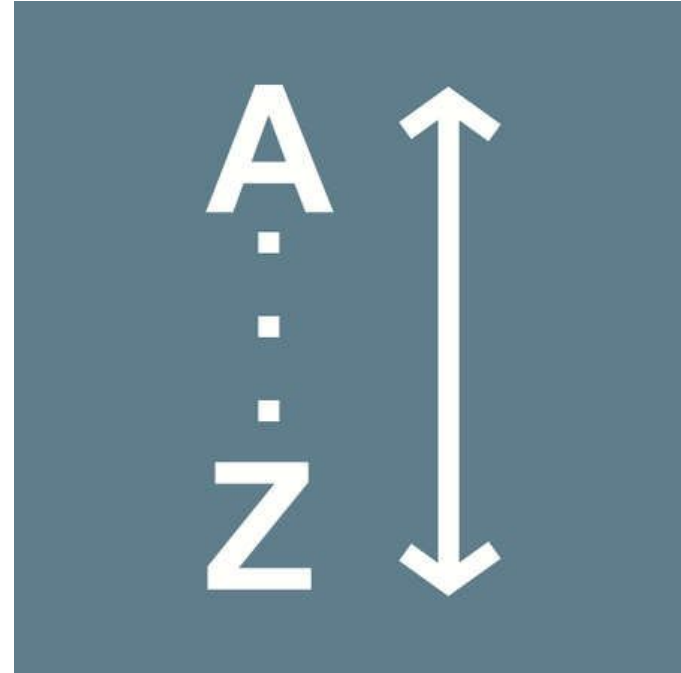
>Sélection des offres de covoiturage ordonnées par prix croissant

Utilité :

>L'objectif est de pouvoir voir les offres de covoiturage en commençant par les prix les plus abordables

Cible :

>Etudiant



L'utilisateur pourra choisir une offre plus adaptée à son budget

Requête 4 (Group by):

Description :

>On sélectionne les colocations qui ne sont pas remplies complètement groupées par leur prix

Utilité :

>L'objectif est de proposer un panel de colocations à l'utilisateur qui sont encore disponibles et par échelles de prix

Cible :

>Etudiant



L'étudiant pourra choisir des logements et le système n'en proposera pas des indisponibles. Ils seront aussi groupés par prix ce qui peut être intéressant pour les recherches par prix.

Requête 5 (Sélection):

Description :

>Sélectionner les étudiants qui font partie de ceux ayant eu une colocation a plus de la moyenne de prix groupés par leurs moyen de transport

Utilité :

>Déterminer si ceux ayant un moyen de transport on dans la globalité une meilleur aisance financière

Cible:

>Etudiant



Ceci permettra d'enrichir notre algorithme de recommandation dans le futur

Requête 6 (Order by):

Description :

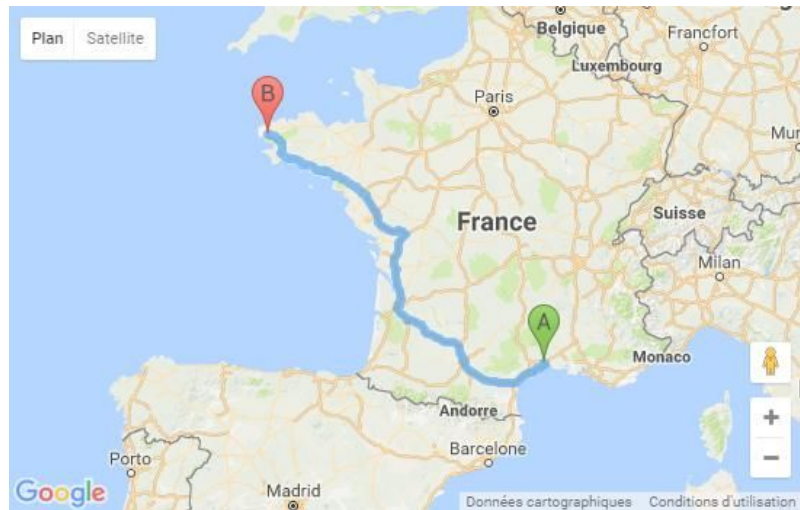
>On sélectionne les points d'arrivée des covoiturages en fonction du point de départ saisi par l'utilisateur. Les résultats sont triés dans l'ordre alphabétique du point d'arrivée.

Utilité :

>L'objectif est qu'un étudiant puisse avoir la liste de toutes les destinations auxquelles il peut accéder à partir d'un point A.

Cible :

>Etudiant



Requête 7 (Group by + agrégation):

Description :

>On sélectionne le nombre d'évènement accepter par les administrateur grouper par étudiant

Utilité :

>L'objectif est de pouvoir si certain utilisateur on des meilleur suggestion en matière d'évènement

Cible :

>Administrateur



Cela a comme aspect pratique de savoir la pertinence des suggestion de certain utilisateur.

Requête 8 (Group by + sous requête + agrégation):

Description :

>On sélectionne les événement grouper par l'administrateur qui les a créés et avec le nb de participant de chacun des évènement

Utilité :

>L'objectif est de pouvoir voir si les administrateur ont des bon choix en matière de filtrage d'évènement.
en fonction de la participation des gens au événement qu'ils créent

Cible :

>administrateur



Requête 9 (requête paramétrée):

Description :

>On sélectionne les événements selon une date saisie par l'utilisateur

Utilité :

>L'objectif est de pouvoir rechercher des événements à une date précise

Cible :

>Etudiant



Requête 10 (Insertion):

Description :

>On insère des noms d'activités que l'utilisateur a recherché quand il était sur le site

Utilité :

>L'objectif est de pouvoir remplir la base de données des préférences de l'utilisateur pour favoriser ses recommandations

Cible :

>Etudiant



Requête 11 (Insertion):

Description :

>L'administrateur insère un nouvel événement

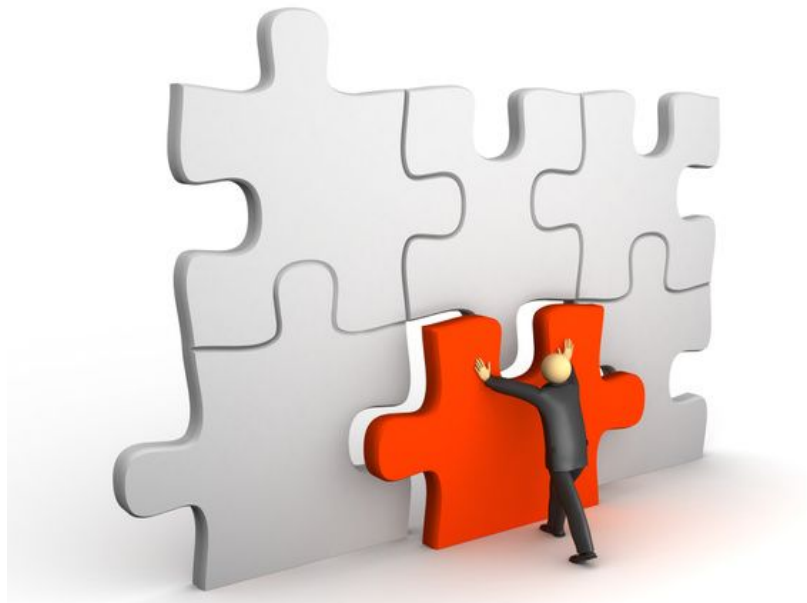
Utilité :

>Permet d'ajouter de nouveaux événements à venir sur le site

Cible :

>Administrateur

La requête permet d'ajouter des nouveaux événements



Requête 12 (Agrégation):

Description :

>Sélectionner le nombre de fois où un étudiant a participé à des événements

Utilité :

>L'objectif est de voir si un étudiant aime les événements ou non.

Cible :

>Etudiant



Requête 13 (Cohérence):

Description :

>Vérifier le nombre d'utilisateurs participant à une activité avec des places limitées

Utilité :

>Vérifier que le nombre de participants n'excède pas la limite de places disponibles

Cible :

>Etudiant

Cela permet de vérifier la cohérence des données dans la base.



Requête 14 (Cohérence):

Description :

>Vérifier que l'utilisateur n'est pas dans plusieurs événements issus d'une même plage horaire.

Utilité :

>Pouvoir libérer des places aux dit événement car l'utilisateur ne pourra pas y participer a toute

Cible :

>Etudiant



Requête 15 (sous-requête):

Description :

>Sélectionner les offres d'emploi d'une entreprise en fonction de si la plage horaire saisie par l'utilisateur correspond à celle donnée dans l'offre

Utilité :

>Pour savoir si une entreprise propose des emploi avec une plage horaire convenant à l'utilisateur.

Utilisateur concerné :

>Etudiant



Les étudiants sont souvent contraints par leurs horaires quand ils recherchent un travail en parallèle de leurs études. De ce fait, il serait bien pour eux de filtrer les offres en fonction des horaires.

Requête 16 (UPDATE):

Description :

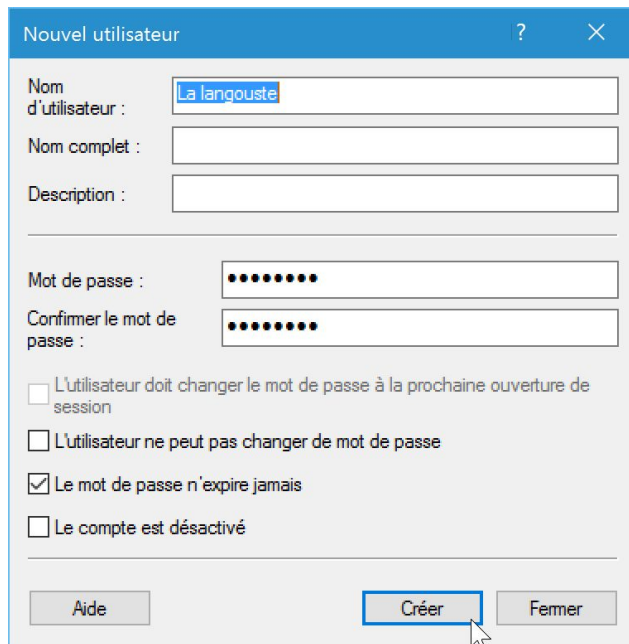
>Modifier un paramètre du compte de l'utilisateur

Utilité :

>Permet de mettre à jour ses données si elles sont erronées ou ne correspondent plus à la réalité

Utilisateur concerné :

>Etudiant



Peut permettre de modifier, par exemple, son numéro de téléphone.

Requête 17 (requête paramétrée):

Description :

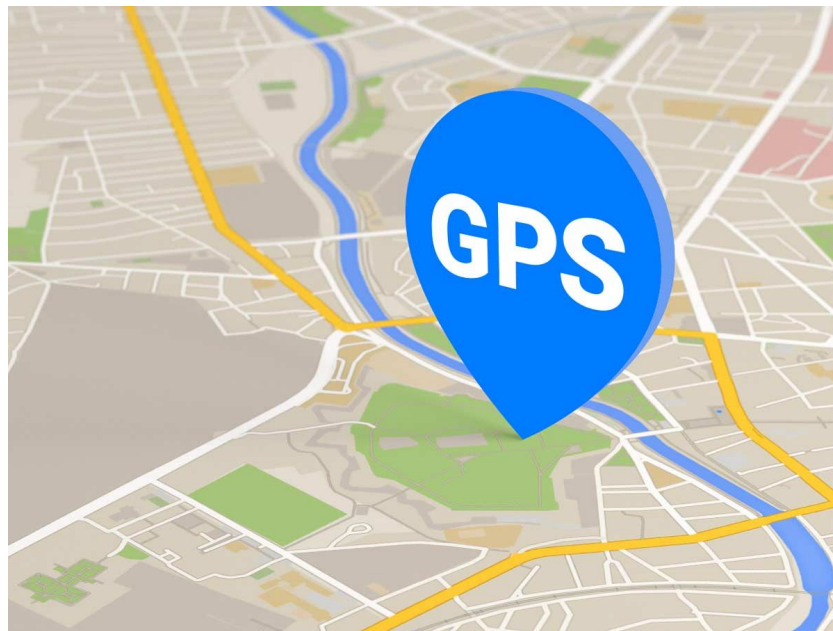
> Sélectionner une activité en fonction d'une adresse rentrée par l'utilisateur

Utilité :

> Afficher à l'utilisateur les activités sur une certaine zone

Utilisateur concerné :

> Etudiant



Si l'utilisateur veut découvrir des activités sur une zone particulière, il peut directement effectuer une recherche comme ça.

Requête 18 (Group by):

Description :

>On veut avoir le nombre d'événements générés à partir d'une même activité

Utilité :

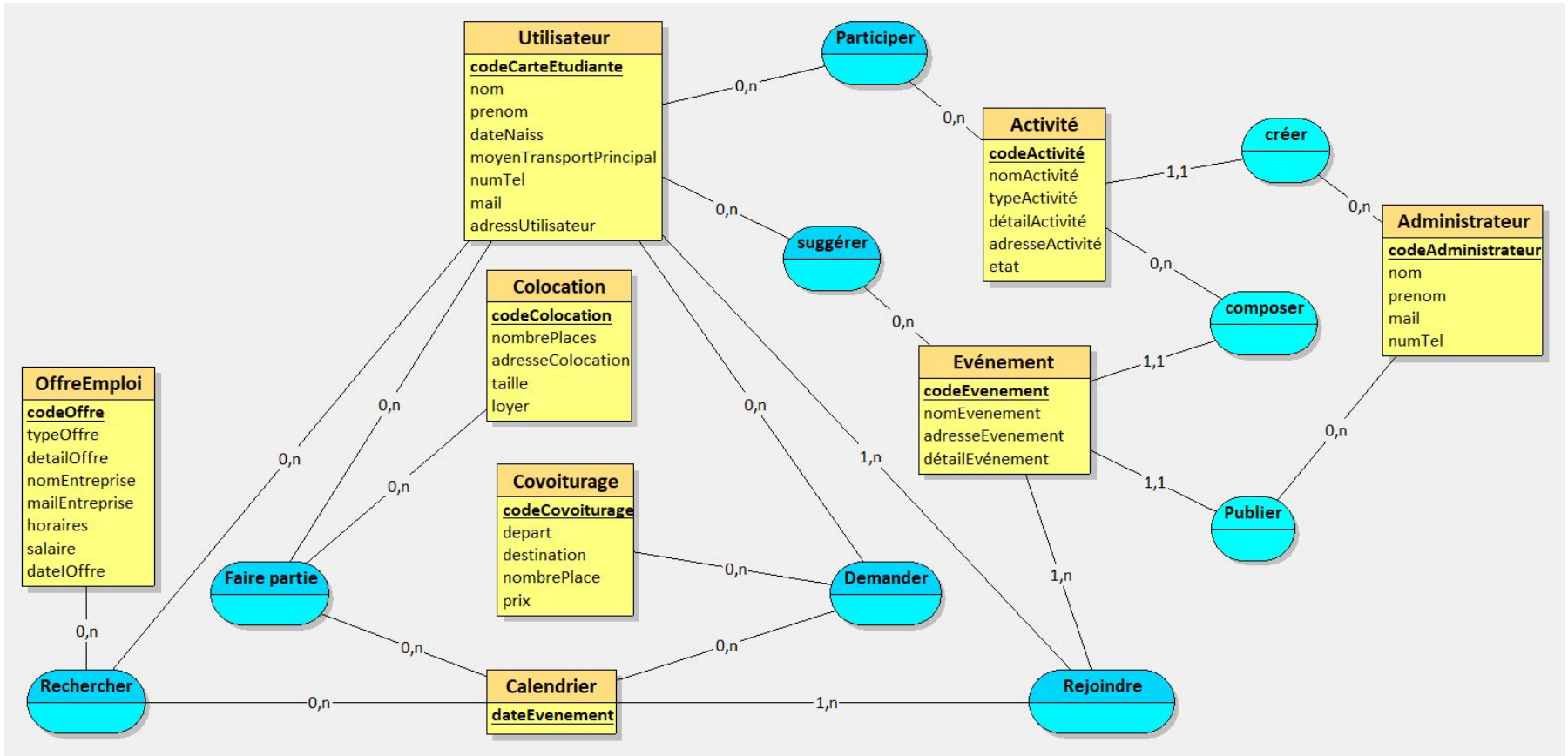
>Cela permettrait de savoir quelles activités sont les plus susceptibles de générer des événements et éventuellement les plus demandées (comme un utilisateur peut demander à créer un événement qui sera validé plus tard par l'administrateur)

Utilisateur concerné :

>Etudiant



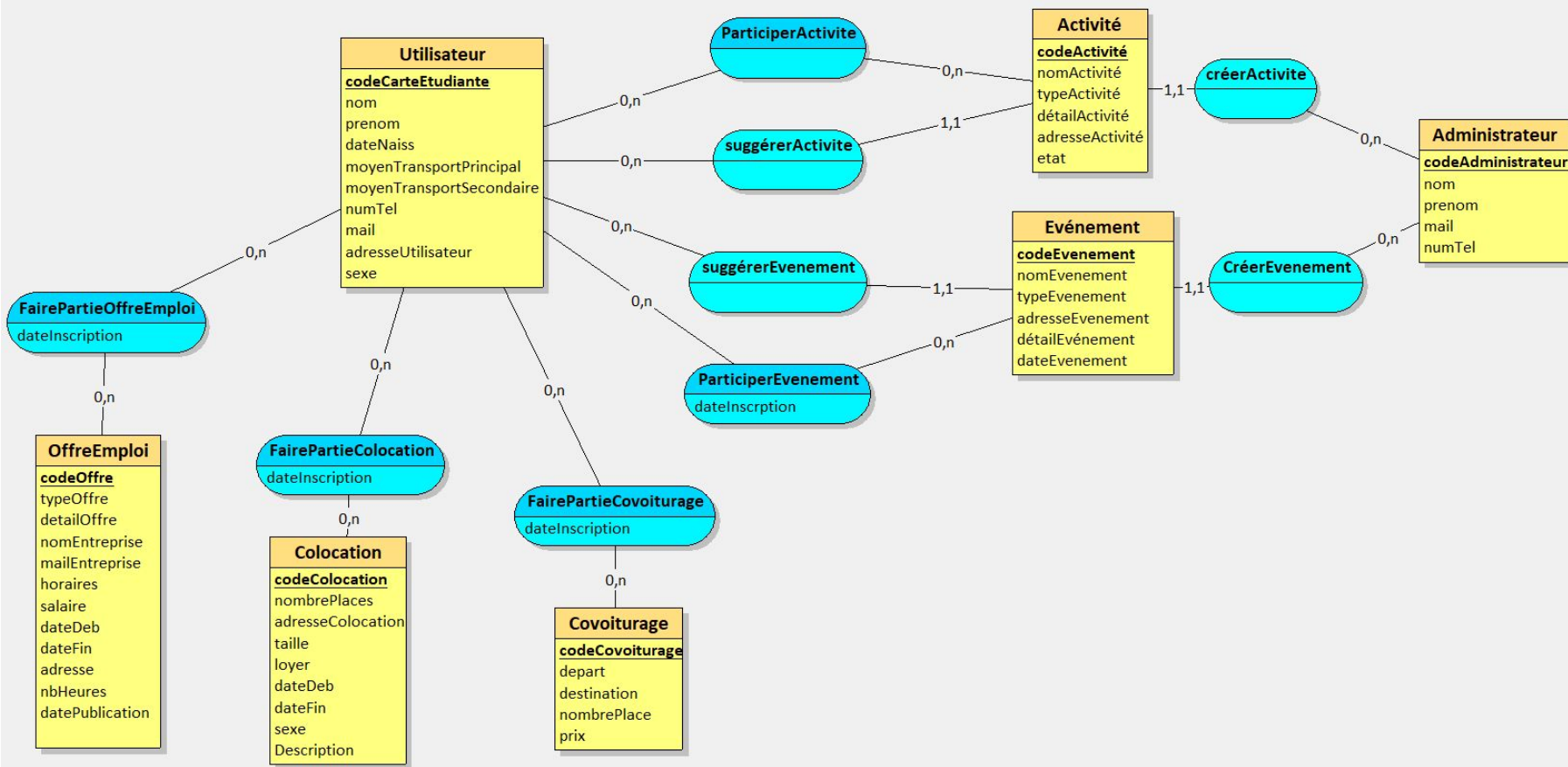
Ancien schéma conceptuel :



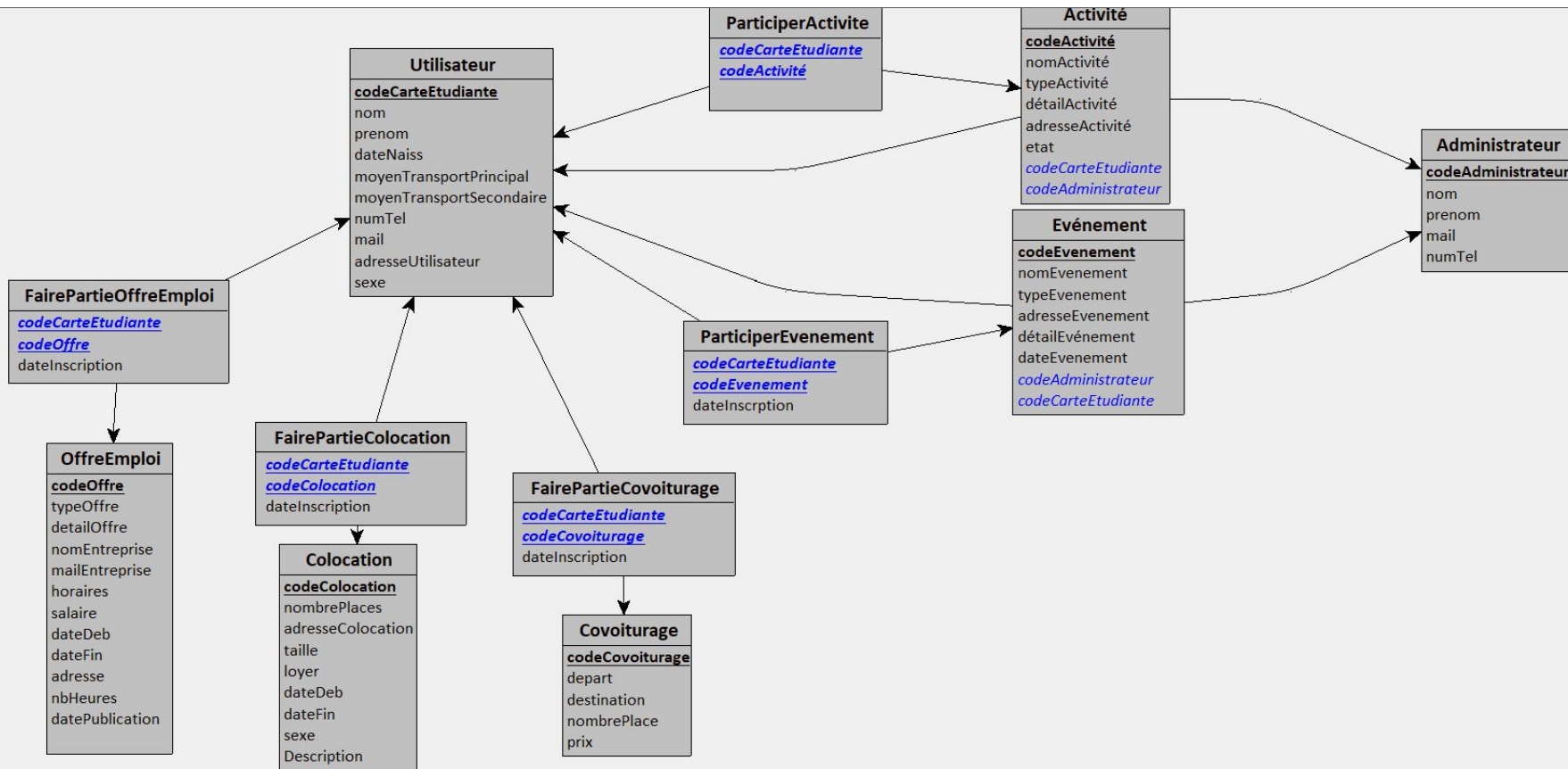
Ancien schéma relationnel

```
Utilisateur = (codeCarteEtudiante VARCHAR(6), nom VARCHAR(30), prenom VARCHAR(30), dateNaiss DATE, moyenTransportPrincipal VARCHAR(25), numTel VARCHAR(10), mail VARCHAR(30), adressUtilisateur VARCHAR(50));  
Administrateur = (codeAdministrateur VARCHAR(4), nom VARCHAR(30), prenom VARCHAR(30), mail VARCHAR(30), numTel VARCHAR(10));  
OffreEmploi = (codeOffre VARCHAR(4), typeOffre VARCHAR(10), detailOffre VARCHAR(50), nomEntreprise VARCHAR(50), mailEntreprise VARCHAR(50), horaires VARCHAR(50), salaire VARCHAR(50), date!Offre DATE);  
Colocation = (codeColocation VARCHAR(4), nombrePlaces INT, adresseColocation VARCHAR(50), taille DECIMAL(15,2), loyer DECIMAL(15,2));  
Covoiturage = (codeCovoiturage VARCHAR(50), depart VARCHAR(50), destination VARCHAR(50), nombrePlace VARCHAR(50), prix DECIMAL(15,2));  
Calendrier = (dateEvenement VARCHAR(50));  
Activité = (codeActivité VARCHAR(4), nomActivité VARCHAR(50), typeActivité VARCHAR(50), detailActivité VARCHAR(50), adresseActivité VARCHAR(50), etat VARCHAR(50), #codeAdministrateur);  
Événement = (codeEvenement VARCHAR(4), nomEvenement VARCHAR(50), adresseEvenement VARCHAR(50), detailEvenement VARCHAR(50), #codeActivité, #codeAdministrateur);  
Participer = (#codeCarteEtudiante, #codeActivité);  
Rejoindre = (#codeCarteEtudiante, #codeEvenement, #dateEvenement);  
Rechercher = (#codeCarteEtudiante, #codeOffre, #dateEvenement);  
Faire_partie = (#codeCarteEtudiante, #codeColocation, #dateEvenement);  
Demander = (#codeCarteEtudiante, #codeCovoiturage, #dateEvenement);  
suggérer = (#codeCarteEtudiante, #codeEvenement);
```

Nouveau schéma conceptuel :



Nouveau schéma conceptuel (forme relationnelle):



Nouveau schéma relationnel

Utilisateur = (codeCarteEtudiante VARCHAR(6), nom VARCHAR(30), prenom VARCHAR(30), dateNaiss DATE, moyenTransportPrincipal VARCHAR(25), moyenTransportSecondaire VARCHAR(50), numTel VARCHAR(10), mail VARCHAR(30), adresseUtilisateur VARCHAR(50), sexe VARCHAR(50));

Administrateur = (codeAdministrateur VARCHAR(4), nom VARCHAR(30), prenom VARCHAR(30), mail VARCHAR(30), numTel VARCHAR(10));

OffreEmploi = (codeOffre VARCHAR(4), typeOffre VARCHAR(10), detailOffre VARCHAR(50), nomEntreprise VARCHAR(50), mailEntreprise VARCHAR(50), horaires VARCHAR(50), salaire VARCHAR(50), dateDeb DATE, dateFin DATE, adresse VARCHAR(50), nbHeures INT, datePublication VARCHAR(50));

Colocation = (codeColocation VARCHAR(4), nombrePlaces INT, adresseColocation VARCHAR(50), taille DECIMAL(15,2), loyer DECIMAL(15,2), dateDeb DATE, dateFin DATE, sexe VARCHAR(10), Description VARCHAR(50));

Covoiturage = (codeCovoiturage VARCHAR(50), depart VARCHAR(50), destination VARCHAR(50), nombrePlace VARCHAR(50), prix DECIMAL(15,2));

Activité = (codeActivité VARCHAR(4), nomActivité VARCHAR(50), typeActivité VARCHAR(50), détailActivité VARCHAR(50), adresseActivité VARCHAR(50), etat VARCHAR(50), #codeCarteEtudiante, #codeAdministrateur);

Événement = (codeEvenement VARCHAR(4), nomEvenement VARCHAR(50), typeEvenement VARCHAR(50), adresseEvenement VARCHAR(50), détailEvenement VARCHAR(50), dateEvenement DATE, #codeAdministrateur, #codeCarteEtudiante);

ParticiperActivite = (#codeCarteEtudiante, #codeActivité);

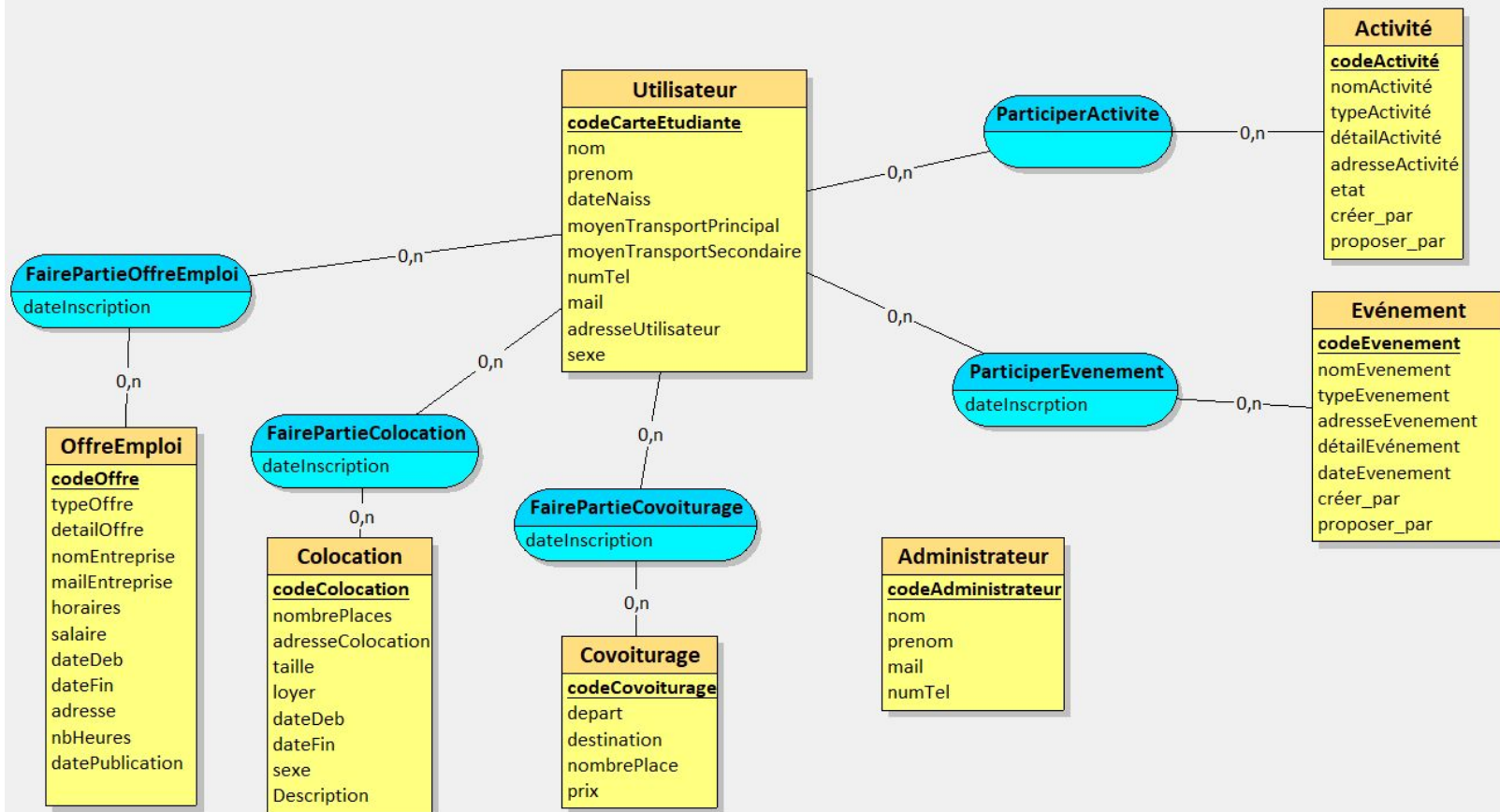
FairePartieOffreEmploi = (#codeCarteEtudiante, #codeOffre, dateInscription VARCHAR(50));

FairePartieColocation = (#codeCarteEtudiante, #codeColocation, dateInscription VARCHAR(50));

FairePartieCovoiturage = (#codeCarteEtudiante, #codeCovoiturage, dateInscription VARCHAR(50));

ParticiperEvenement = (#codeCarteEtudiante, #codeEvenement, dateInscription VARCHAR(50));

Proposition de M.Roose



FIN

Merci pour votre écoute