RAPPORT DE PROJET

**GESTION D’UNE RESIDENCE UNIVERSITAIRE**

## PAR

* **Tom Naudin (Gestion, FR)**
* **Christopher Yepmo (Informatique, CMR)**
* **Eva-Johanna Herrlich (Gestion, GER)**

**Professeur**

* **Sorin Stratulat**

**TABLE DES MATIERES**

[Le sujet 3](#_bookmark0)

[Dictionnaire des données 4](#_bookmark1)

[MCD 5](#_bookmark2)

[Modèle relationnel](#_bookmark3) 5

[Entités 5](#_bookmark4)

[Association 6](#_bookmark5)

[Liste des tables sur ACCESS 7](#_bookmark6)

[Exemples de formulaire 11](#_bookmark7)

[Requêtes 14](#_bookmark8)

[Conclusion 14](#_bookmark9)

[Bibliographie 17](#_TOC_250000)

# Le sujet

Nous avons choisi ce thème car nous sommes tous des étudiants et la plupart d‘entre nous habitons dans des résidences estudiantines et nous avons beaucoup de connaissances par rapport à ce type de logements.

Les résidences universitaires que notre application permettra de gérer sont les résidences CROUS situées à Metz (comme celle du Saulcy, de Bridoux ou du Technopôle). Chaque résidence est située à une station qui a un nom, une surface et un nombre de bâtiments spécifique. Dans chacune de ces stations travaillent des employés, qui sont identifiés par leurs noms et prénoms, leur date de naissance, leurs horaires de travail, leur sexe, leur contact et leur poste. Les employés peuvent travailler dans plusieurs bâtiments et/ou établissements. Les établissements ont un nom, des horaires d’ouvertures et des informations de contact, tandis que les bâtiments n’ont qu’un nom et/ou un numéro.

Dans chaque bâtiment, on retrouve plusieurs chambres occupées par des étudiants et l’on cherchera à connaître le numéro de cette chambre, sa superficie, le type de chambre que c’est (si c’est une colocation ou autre chose) et le loyer payé par l’étudiant. Le système informatique aimerait également savoir si la chambre est occupée ou si celle-ci n’est que réservée. Dans le cas d’une occupation, les étudiants doivent, pour des raisons d’assurance donner leur nom, prénom, les dates d’arrivée et de départ, leur contact, leur niveau scolaire et leur IBAN. Ils doivent également fournir leur assurance habitation comprenant leur nom, la date de validité de cette assurance, leur contact et leur adresse. Enfin, pour attester qu’ils sont étudiants, la résidence leur demandera l’université dans laquelle ils étudient avec le nom de l’école, l’adresse et le nom du directeur de l’université.

Si la chambre n’est que réservée, l’administration souhaiterait savoir le numéro et la date de la réservation, le nombre de chambre réservées et le temps que cela durera.

Un étudiant obtient également des aides aux logement et il faudrait que la résidence connaisse le type d’aide, l’adresse et le contact de l’organisation qui apporte l’aide et le montant de cette dernière.

Enfin, les différentes stations CROUS possèdent un système de communication qui permet aux employés et aux étudiants d’envoyer des messages et la date de leur envoi pour stipuler des problèmes rencontrés (par exemple une panne). Les employés pourront donc appartenir à une équipe de réparation qui se chargera de la panne et d’expliquer le type de panne et son coût de réparation.

# Dictionnaire des données

* Horaire\_travail: Heures de travail
* Date\_nais: Date de naissance
* Nombre\_Bâtiment: Nombre de bâtiment dans une station
* Contact: Informations contact
* Numéro: Numéro de chambre
* Superficie: Superficie de chambre
* TypeCham: Type de chambre
* DateArr: Date d’arrivée
* DateDep: Date de départ
* Type\_aide: Type de l’aide (CAF)
* Niv\_etude: Niveau scolaire
* Assurance\_habitation: Assurance d’habitation d’un etudiant
* NumRes: Numéro de réservation
* DateRes: Date de réservation
* TempsRestant: Temps qui reste pour valider la réservation
* Sujet: Sujet de message
* Message: Message
* Date\_env: Date d’envoie
* Type\_panne: Type de panne
* Coût\_repar: Coût de réparation
* Equipe\_reparation: Equipe de réparation
* Date\_payement: Date de paiement
* Nom: Nom d’étudiant

# Modèle Conceptuel de Donnée

Il se trouve en annexe de ce document (mcd\_residence\_universitaire.pdf)

# Modèle relationnel des Données

## Entités

Horaire\_travail (Id\_horraire, Lundi, Mardi, Mercredi, Jeudi, Vendredi, Samedi, Dimanche)

Bâtiment (Id\_bat, Nom, Nombre\_etage, Nombre\_chambre, #Id\_station) Panne (Id\_panne, Type\_panne, Coût\_repar)

Equipe\_reparation (Id\_eqp, Entreprise, Nbre\_personnel)

Station (Id\_station, Nom, Superficie, Nombre\_bâtiment, Adresse)

Employé (Matricule, Nom, Prénom, Date\_nais, Sexe, Contact, Fonction)

Chambre (Numéro, HO, Superficie, TypeCham, Loyer, #Id\_bat, #N°\_INE, NumRes, DateRes, TempsRestant, Caution)

Etablissement (Id\_etabl, Rôle, HD, Info\_contact, #Id\_station, #id\_horaire)

Etudiant (N°\_INE, Nom, Prenom, DateArr, DateDep, Contact, Email, Niv\_etude, IBAN, N°\_chambre)

Assurance\_habitation (Id\_assur, Nom\_compagnie, Contact, Adresse) Aide\_logement (Id\_aide, Type\_aide, Contact, Montant, Adresse)

Adresse (Id\_add, Numéro, Nom\_voie, Code\_postale, Ville, Pays) Communication (Id\_com, Sujet, Message, Date\_env)

Université (Nom, Directeur, #Id\_add)

## Association

Dans (\_Id\_bat, Id\_panne)

Reparer (Id\_eqp, Matricule, Id\_panne) Travailler\_etabl (Matricule, Id\_etabl, Est\_directeur) Vivre\_à (Matricule, Id\_add)

Recevoir\_Empl (Matricule, Id\_com) Envoyer\_Empl (Matricule, Id\_com) Recevoir\_Etud (\_N°\_INE, Id\_com) A (\_Id\_aide, Id\_assur, N°\_INE) Etudier (\_N°\_INE, Nom)

Payer (\_N°\_INE, \_Id\_station, montant, Motif, Date\_payement)

# Exemples de table sur ACCESS

Pour mener à bien le projet, nous avons dû créer une trentaine de table que nous avons remplis avec des données fictives pour effectuer nos tests. Ci- dessous se trouve des exemples.

Table Chambre :

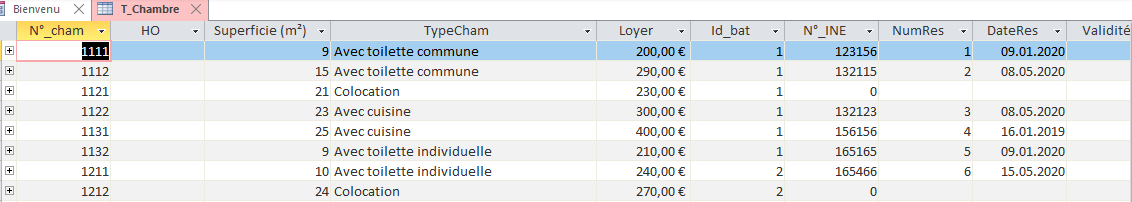


Table Etablissement :

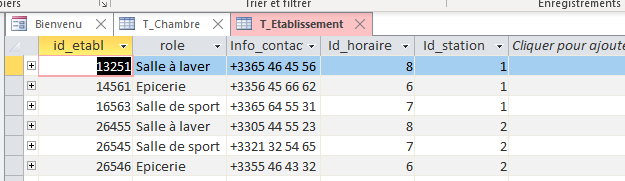


Table Étudiant :

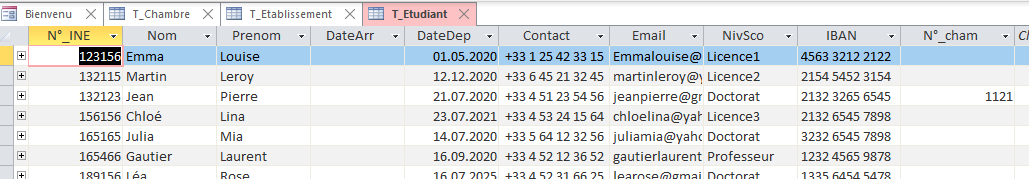


Table Horaire\_travail :

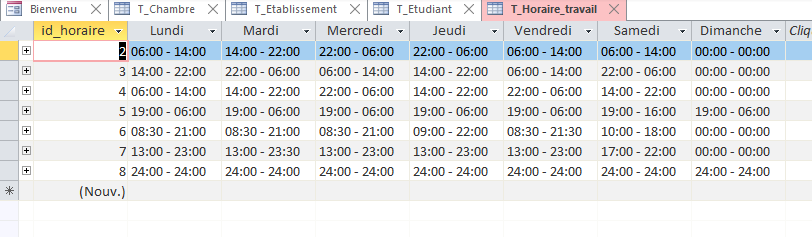


Table Panne :

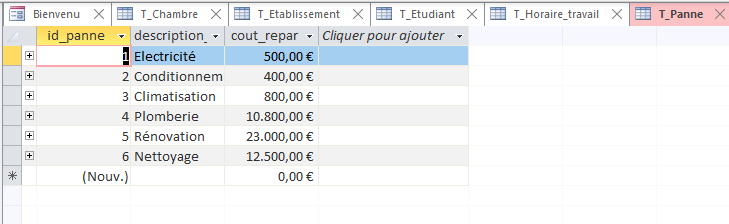


Table Station :

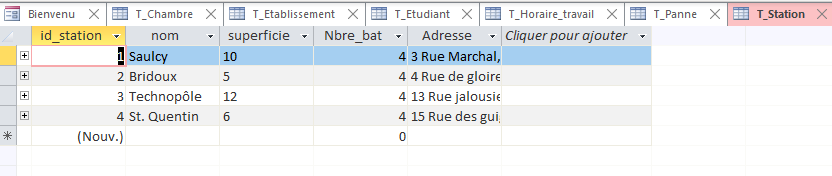


Table Université :

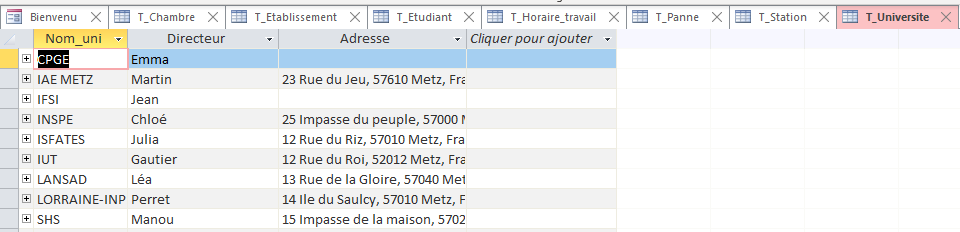


Table Batiment :

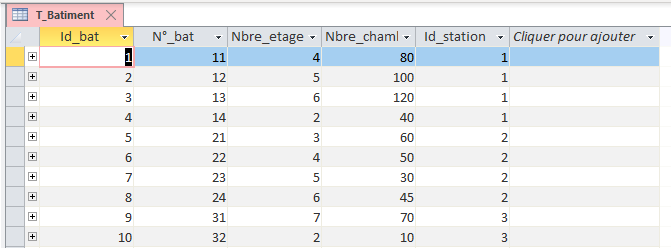


Table Équipe de réparation :



Table Employé :

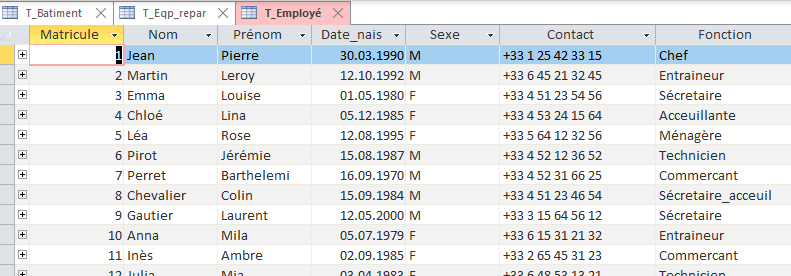


Table assurance habitation :

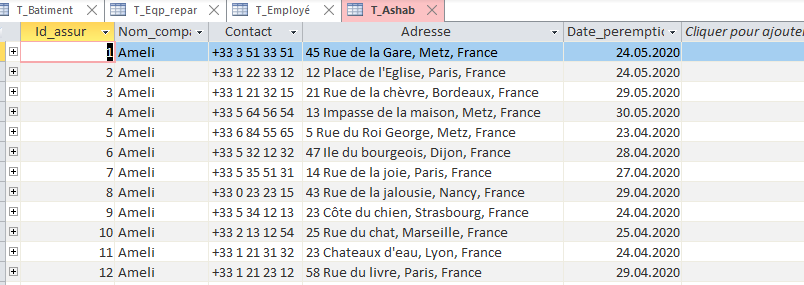


Table Aide Logement :

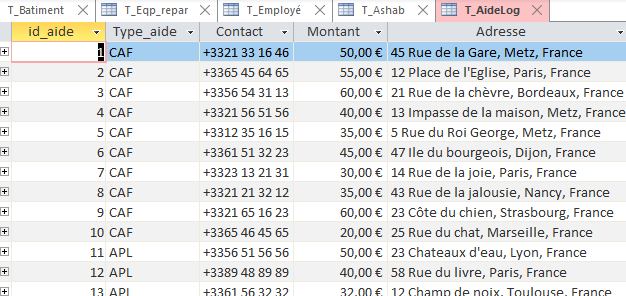


Table Adresse :

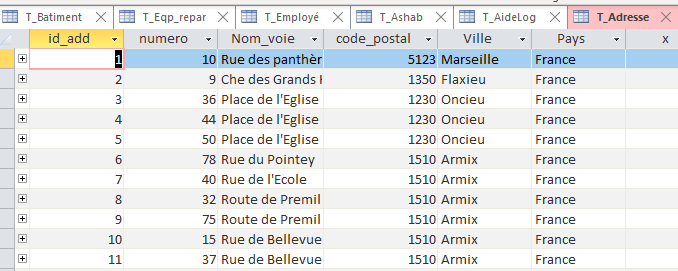
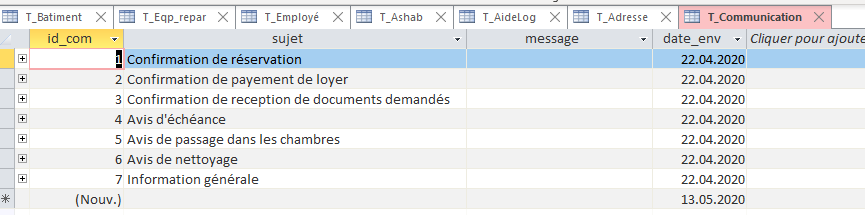


Table Communication :



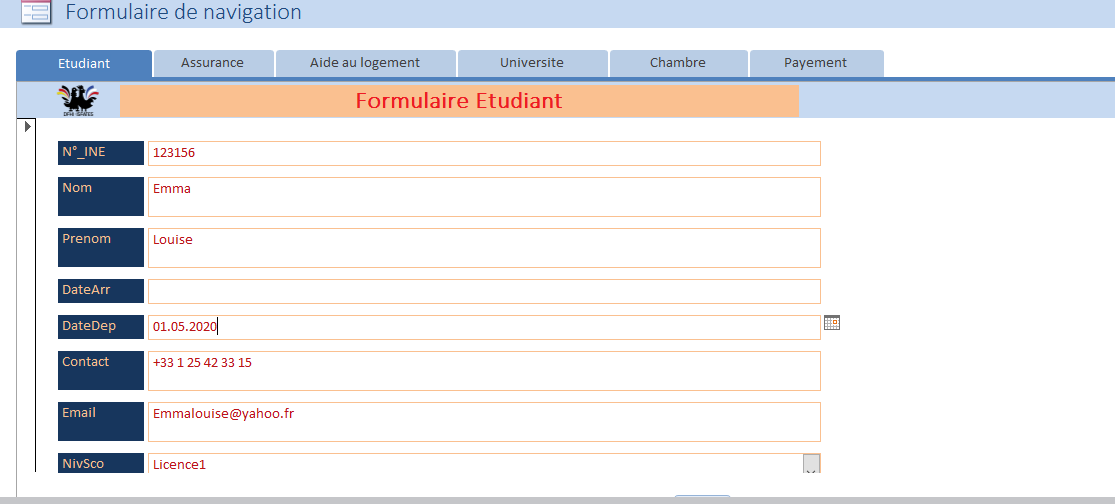
# Formulaires

Pour les formulaires nous avons créé trois types de formulaires. Un formulaire de démarrage qui se présente à l’ouverture de l’application et contenant des boutons vers les différents formulaires de navigation. Nous avons 4 formulaires de navigation (Employés, étudiants, pannes, stations). Ces formulaires de navigation contiennent des formulaires pour nous permettre la modification des données de toutes les tables. Ci-dessous se trouvent quelques exemples des différents formulaires.

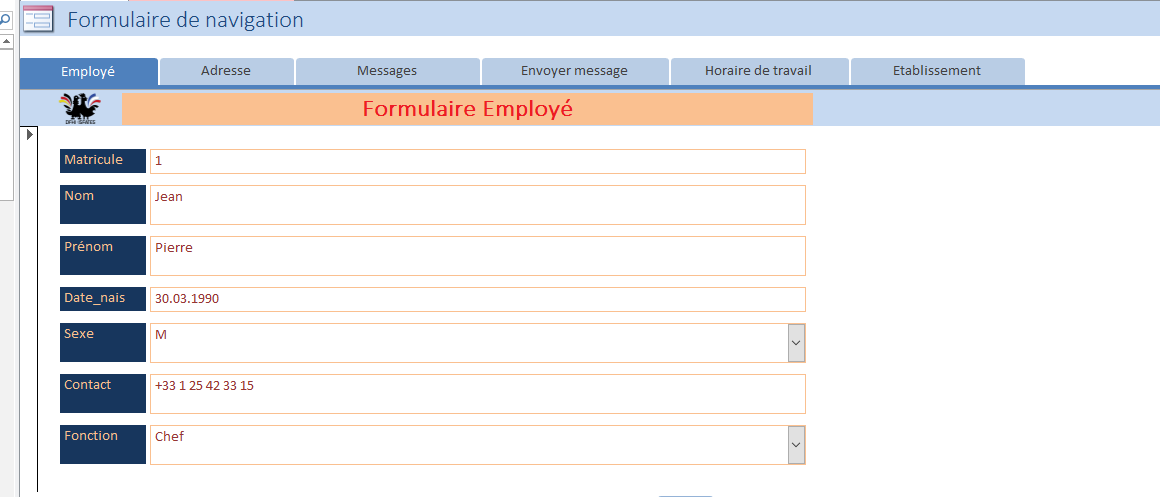
**Formulaire de Démarrage contenant des boutons vers les formulaires de navigation :**



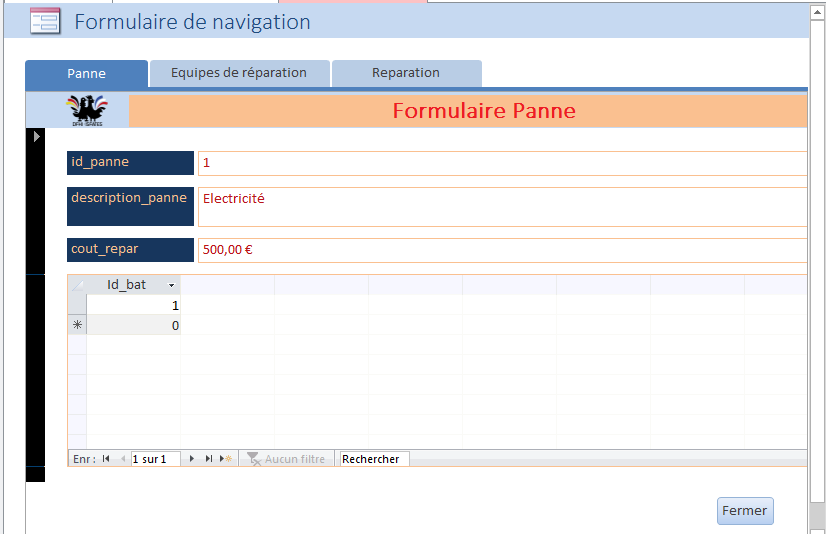
**Formulaire de navigation « étudiant » contenant les autres formulaires pour la gestion des informations sur les étudiants :**



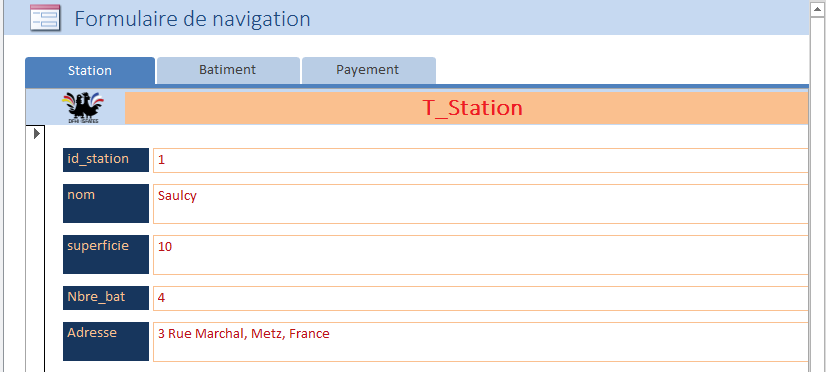
**Formulaire de navigation « employé » contenant les formulaires pour la gestion des informations des employés :**



**Formulaire de navigation « Panne » contenant les formulaires pour la gestion des informations sur les pannes :**



**Formulaire de navigation « Station » contenant les formulaires pour la gestion des informations sur les stations :**



# Requêtes

Comme listé dans notre cahier de charge lors de la première présentation en début de projet, ci-dessous se trouve les requêtes que nous devrions effectuer.

1. **Quels sont les employés qui travaillent à une certaine station ?**

SELECT T\_Station.id\_station, T\_Employé.Matricule, T\_Employé.Nom, T\_Employé.Prénom

FROM (T\_Station INNER JOIN T\_Etablissement ON T\_Station.[id\_station] = T\_Etablissement.[Id\_station]) INNER JOIN (T\_Employé INNER JOIN R\_Travailler\_etabl ON T\_Employé.[Matricule] = R\_Travailler\_etabl.[Matricule]) ON T\_Etablissement.[id\_etabl] = R\_Travailler\_etabl.[Id\_etabl];

### Quelles sont les assurances habitation qui ne sont plus valide et à qui est-ce qu’elles appartiennent ?

SELECT R\_A.Id\_assur, R\_A.N°\_INE, T\_Ashab.Date\_peremption, T\_Etudiant.Nom, T\_Etudiant.Prenom

FROM T\_Etudiant INNER JOIN (T\_Ashab INNER JOIN R\_A ON T\_Ashab.Id\_assur = R\_A.Id\_assur) ON T\_Etudiant.N°\_INE = R\_A.N°\_INE

WHERE (((T\_Ashab.Date\_peremption) < Date()));

### Quels bâtiments, sur quels sites ont quelles pannes ?

SELECT R\_Dans.Id\_bat, R\_Dans.Id\_panne, T\_Batiment.Id\_bat, T\_Station.id\_station, T\_Station.nom

FROM T\_Panne INNER JOIN ((T\_Batiment INNER JOIN T\_Station ON T\_Batiment.Id\_station = T\_Station.id\_station) INNER JOIN R\_Dans ON T\_Batiment.Id\_bat = R\_Dans.Id\_bat) ON T\_Panne.id\_panne = R\_Dans.Id\_panne;

### Quels employés font partir d’une certaine équipe de réparation ?

SELECT R\_Reparer.Id\_eqp, R\_Reparer.Matricule, T\_Employé.Nom, T\_Employé.Prénom

FROM T\_Eqp\_repar INNER JOIN (T\_Employé INNER JOIN R\_Reparer ON T\_Employé.Matricule = R\_Reparer.Matricule) ON T\_Eqp\_repar.id\_eqp = R\_Reparer.Id\_eqp

WHERE (((R\_Reparer.Id\_eqp)=1));

### Quelles chambres sont encore disponibles sur le site XY?

SELECT T\_Etudiant.N°\_cham, T\_Chambre.N°\_cham, T\_Chambre.NumRes, T\_Station.id\_station

FROM ((T\_Batiment INNER JOIN T\_Station ON T\_Batiment.Id\_station = T\_Station.id\_station) INNER JOIN T\_Chambre ON T\_Batiment.Id\_bat = T\_Chambre.Id\_bat) LEFT JOIN T\_Etudiant ON T\_Chambre.N°\_cham = T\_Etudiant.N°\_cham

WHERE (((T\_Etudiant.N°\_cham) Is Null) AND ((T\_Chambre.NumRes) Is Null));

### Combien d’étudiants déménageront en Juillet?

SELECT T\_Etudiant.DateDep, T\_Etudiant.N°\_INE

FROM T\_Chambre LEFT JOIN T\_Etudiant ON T\_Chambre.N°\_cham = T\_Etudiant.N°\_cham WHERE (((T\_Etudiant.DateDep) Like "\*/07/\*"));

### Quel étudiant n’a pas encore payé le loyer pour le mois XY?

SELECT R\_Payer.Motif, T\_Etudiant.N°\_INE, T\_Etudiant.Nom, T\_Etudiant.Prenom, T\_Etudiant.N°\_cham

FROM T\_Etudiant LEFT JOIN R\_Payer ON T\_Etudiant.N°\_INE = R\_Payer.N°\_INE WHERE (((R\_Payer.Motif)<>"Loyer")) OR (((R\_Payer.Motif) Is Null));

### Quel est le numéro de réservation d’un certain étudiant?

SELECT T\_Chambre.NumRes, T\_Chambre.N°\_INE

FROM T\_Chambre LEFT JOIN T\_Etudiant ON T\_Chambre.N°\_cham = T\_Etudiant.N°\_cham WHERE (((T\_Chambre.NumRes) Is Not Null));

### Combien d’étudiants de l’université A vivent dans la résidence X?

SELECT T\_Etudiant.N°\_INE, T\_Etudiant.Nom, T\_Etudiant.Prenom, R\_Etudier.Nom\_uni, T\_Chambre.Id\_bat, T\_Chambre.N°\_cham, T\_Station.nom

FROM T\_Universite INNER JOIN ((((T\_Station RIGHT JOIN T\_Batiment ON T\_Station.id\_station = T\_Batiment.Id\_station) RIGHT JOIN T\_Chambre ON T\_Batiment.Id\_bat = T\_Chambre.Id\_bat) RIGHT JOIN T\_Etudiant ON T\_Chambre.N°\_cham = T\_Etudiant.N°\_cham) INNER JOIN R\_Etudier ON

T\_Etudiant.N°\_INE = R\_Etudier.N°\_INE) ON T\_Universite.Nom\_uni = R\_Etudier.Nom\_uni;

### Combien de chambres ont été réservées pour le mois de janvier?

SELECT T\_Chambre.N°\_cham, T\_Etudiant.N°\_INE, T\_Chambre.NumRes, T\_Chambre.DateRes

FROM T\_Etudiant INNER JOIN T\_Chambre ON T\_Etudiant.N°\_INE = T\_Chambre.N°\_INE WHERE (((T\_Chambre.DateRes) Like '\*/01/\*'));

### Qui est le directeur de l’université dans laquelle étudie l’étudiant Y?

SELECT T\_Universite.Directeur, R\_Etudier.Nom\_uni, R\_Etudier.N°\_INE, T\_Etudiant.Nom, T\_Etudiant.Prenom

FROM T\_Universite INNER JOIN (T\_Etudiant INNER JOIN R\_Etudier ON T\_Etudiant.N°\_INE

= R\_Etudier.N°\_INE) ON T\_Universite.Nom\_uni = R\_Etudier.Nom\_uni WHERE (((T\_Etudiant.N°\_INE)=165466));

### Combien a coûté la réparation de la panne dans l’établissement X?

SELECT T\_Batiment.Id\_bat, R\_Reparer.Id\_panne, T\_Panne.cout\_repar

FROM (T\_Panne INNER JOIN R\_Reparer ON T\_Panne.id\_panne = R\_Reparer.Id\_panne) INNER JOIN (T\_Batiment INNER JOIN R\_Dans ON T\_Batiment.Id\_bat = R\_Dans.Id\_bat) ON T\_Panne.id\_panne = R\_Dans.Id\_panne

WHERE (((T\_Batiment.Id\_bat)=1) AND ((R\_Reparer.Id\_panne)=1));

**Conclusion**

Nous sommes trois étudiants de nationalités différentes. À cause de la situation que nous vivons actuellement (COVID-19) il était plus difficile de rester en contact et chacun avait également d’autres travaux à réaliser.

Toutefois la communication a bien fonctionné. Nous avons apprécié travailler sur ce projet car le thème était libre. Cela nous a permis d’associer le travail au sujet qui nous est très familier, une résidence universitaire. Comme beaucoup d’étudiants vivent, au moins pour le début de leurs études dans une résidence universitaire, il n’était pas difficile de décider.

Nous avons rencontré des difficultés sur Access, comme c’était la première fois de l’utiliser. Grace à l’étudiant en Informatique nous pourrions obtenir une explication sur le fonctionnement de l'accès. Quand nous étions encore en présentiel à Metz nous avons déjà élaboré le MCD et cela aurait été difficile à distance puisqu’il faut que le MCD et les relations sur Access soient cohérent et que chacun comprenne le processus.

**Bibliographie**

<https://support.office.com/en-us/access> <https://www.youtube.com/playlist?list=PLSdtjcTY3VgRyfCNwJBvcLMGCtpUXvt9I>

# Qui a fait quoi ?

Eva-Johanna : Mise en page de la présentation, contribué au MCD, certaines requêtes, contribué au dictionnaire des données