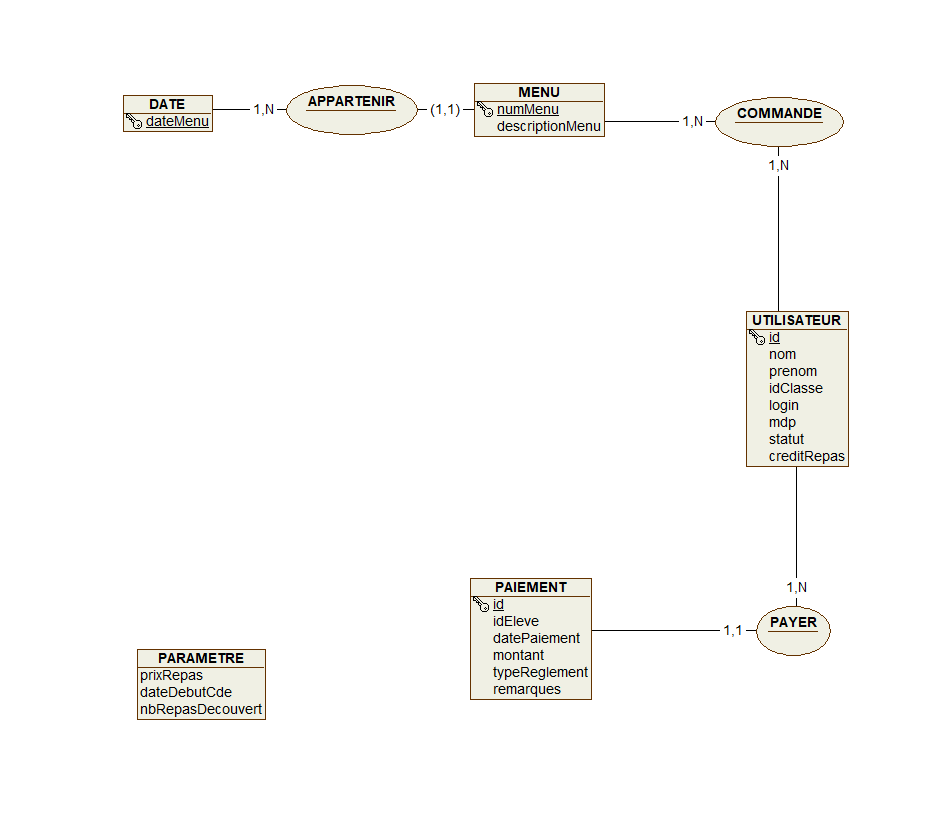
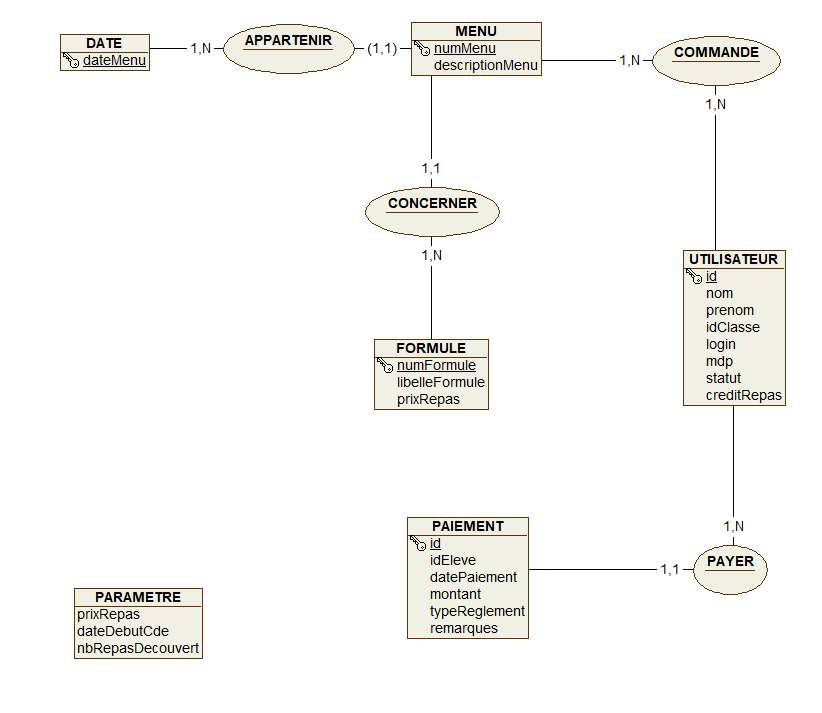
MCD (Thomas et Esteban) :



MCD Modif :



Jointures (Oliwer) :

1)

"Select dateMenu, nom, prenom "  
." from elior\_utilisateur "  
." left join elior\_commande on elior\_utilisateur.id = idUtil "  
." where ".$where." "  
. "Order by dateMenu ASC ";

LEFT JOIN permet de retourner les valeurs d'un tableau A et B, même si les éléments du tableau A ne se trouve pas dans le tableau B. Les colonnes pour lesquelles les éléments A ne sont pas joint au tableau B seront remplacé par null.   
Par exemple, imaginons ce tableau A avec des utilisateurs :

|  |  |  |
| --- | --- | --- |
| **idUser** | **nom** | **prenom** |
| 1 | Pierre | Jacques |
| 2 | Dujardin | Jean |

Et un tableau B avec des commandes :

|  |  |  |
| --- | --- | --- |
| **id** | **libelle** | **numCmd** |
| 1 | Souris | 2022 |
| 1 | Clavier | 2023 |

**idUser** et **id** sont liés.

Si nous faisons une requête SQL avec un left join, notre résultat sera :   
**SELECT \* from table1 LEFT JOIN table2 on table2.id = table1.idUser**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **id** | **libelle** | **numCmd** | **nom** | **prenom** |
| 1 | Souris | 2022 | Pierre | Jacques |
| 1 | Clavier | 2023 | Pierre | Jacques |
| 2 | *null* | *null* | Dujardin | Jean |

LEFT JOIN permet en quelque sorte de joindre 2 tables. Si au lieu d’utiliser LEFT JOIN on aurait utilisé INNER JOIN, on aurait eu comme résultat :

**SELECT \* from table1 INNER JOIN table2 on table2.id = table1.idUser**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **id** | **nom** | **prenom** | **libelle** | **numCmd** |
| 1 | Pierre | Jacques | Souris | 2022 |
| 1 | Pierre | Jacques | Clavier | 2023 |

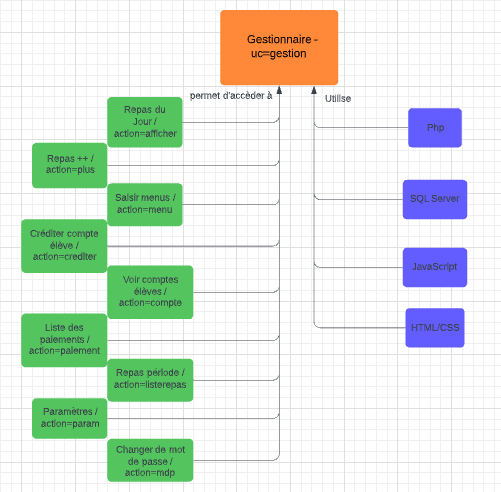
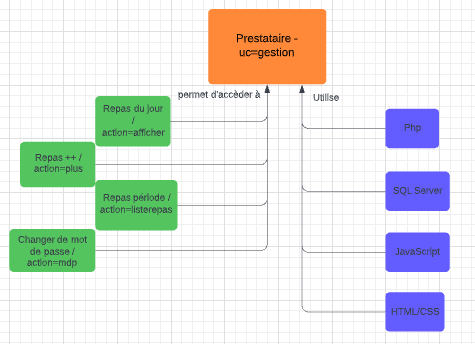
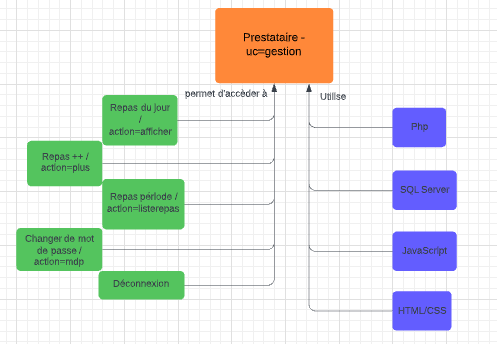
2)

Les jointures qui sont fait dans toutes les requêtes sont des LEFT JOIN.   
Cette jointure se fait entre elior\_commande et elior\_utilisateur. On peut consulter qui sont les élèves qui ont passé une commande et les élèves qui n’ont pas passé les commandes en modifiant la commande en supprimant la condition des dates. Sinon, la fonctionnalité reste la même qu’avec un INNER JOIN, tant que nous gardons la condition.

3)

Les relations sont entre les id et idUtil de elior\_commande et dateMenu de elior\_menu et dateMenu de elior\_commande. Les 2 requêtes permettent de voir si un utilisateur a commandé une journée précise ou non.

Cartographies (Romain) :



|  |  |  |
| --- | --- | --- |
| Numero | Description | Élève |
| 1 | Dans la saisie des menus, introduire la sélection de la formule associée pour le gestionnaire | Esteban |
| 2 | Dans la liste des repas d'une période il y'a actuellement deux choix : Nombre total de repas par date et détail des repas pris par date. Rajouter Nombre total de repas par formule pour le gestionnaire. | Oliwer |
| 3 | Enregistrement des résultats au-dessus au format PDF et EXCEL pour le gestionnaire. | Oliwer |
| 4 | Dans la saisie des paramètres, supprimer la saisie du prix d'un repas. | Romain |
| 5 | Ajouter une option de menu Formules : Ajouter, modifier ou supprimer le libellé et le prix d'une formule. Contraintes : Libelle unique et suppression impossible pour les formules saisies pour un menu pour le gestionnaire. | Romain |
| 6 | Dans la saisie des menus, introduire la sélection de la formule associée pour le prestataire. | Esteban |
| 7 | Dans la liste des repas d'une période il y'a actuellement deux choix : Nombre total de repas par date et détail des repas pris par date. Rajouter Nombre total de repas par formule pour le prestataire. | Oliwer |
| 8 | Modification base de donnée | Thomas |
| 9 | Lors de la validation de la commande : Cout du repas déduit des crédit de l'élève. | Thomas |
| 10 | Option annuler commande : Rends les crédits à l'élève | Thomas |
| 11 | Interface ajout/suppression utilisateurs. Ajouter pour le gestionnaire | Romain |
| 12 | Ajout d'un bouton retour après modification de mot de passe. | Esteban |
| 13 | Modification du MCD | Esteban / Thomas |