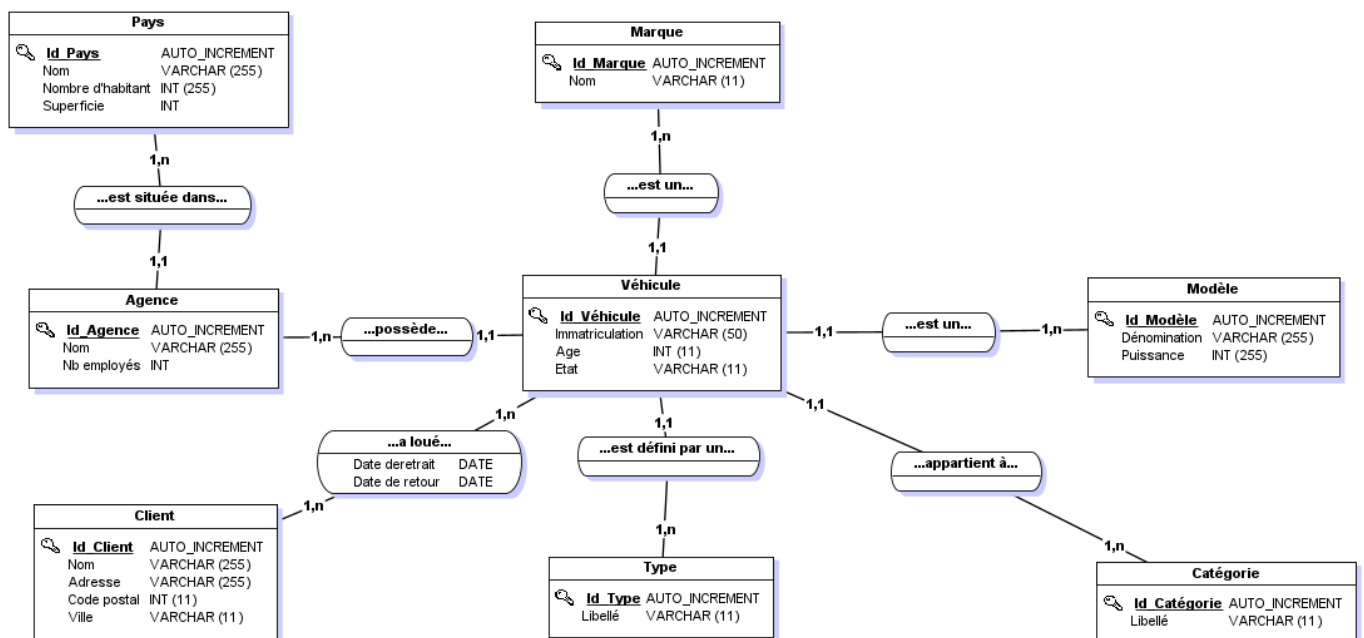


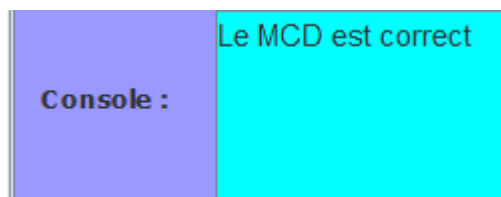
PPE3

1. Création JMerise

Pour commencer, j'ai installé [JMerise 5](#) pour pouvoir concevoir mon MCD (modèle conceptuel de données). Ce logiciel permet de créer des diagrammes de base de données pour ensuite les exporter en fichier SQL pour faciliter la tâche au développeur. Après avoir installé JMerise, je me suis chargé de concevoir le MCD.



Une fois le MCD fait, il faut le vérifier pour voir si tout fonctionne bien et s'il n'y a pas d'erreur dans notre conception. Cela permet de convertir notre MCD en script SQL sans erreur.



Juste après, j'ai converti le MCD en fichier [SQL](#). Cela permettra de mettre dans mon phpmyadmin mon script SQL pour pouvoir faire des requêtes par la suite.

2. Requêtes SQL

Pour pouvoir effectuer des requêtes SQL, il faut d'abord créer les tables, les attributs, les relations, les clés primaires... sur notre MySQL. Cela peut sembler fastidieux et redondant, c'est pour cela que juste auparavant nous avons fait un MCD et que nous l'avons converti en fichier [SQL](#). Il nous reste juste à le transférer dans notre base de données en drag and drop sur notre serveur MySQL. Et nous aurons notre base de données déjà toutes faite.

	Table	Action
<input type="checkbox"/>	agence	Parcourir Structure Rechercher Insérer Vider Supprimer
<input type="checkbox"/>	categorie	Parcourir Structure Rechercher Insérer Vider Supprimer
<input type="checkbox"/>	client	Parcourir Structure Rechercher Insérer Vider Supprimer
<input type="checkbox"/>	marque	Parcourir Structure Rechercher Insérer Vider Supprimer
<input type="checkbox"/>	modele	Parcourir Structure Rechercher Insérer Vider Supprimer
<input type="checkbox"/>	pays	Parcourir Structure Rechercher Insérer Vider Supprimer
<input type="checkbox"/>	type	Parcourir Structure Rechercher Insérer Vider Supprimer
<input type="checkbox"/>	vehicule	Parcourir Structure Rechercher Insérer Vider Supprimer

Maintenant place aux requêtes SQL 😊.

- a) Afficher toutes les informations sur les véhicules loués par le Client n° 122.

✓ MySQL a retourné un résultat vide (c'est à dire aucune ligne). (traitement en 0,0006 seconde(s).)

```
SELECT * FROM vehicule INNER JOIN client ON vehicule.Id_Vehicule = client.Id_Client WHERE client.Id_Client = 122;
```

- b) Afficher toutes les locations réalisées par le client n° 122

✓ MySQL a retourné un résultat vide (c'est à dire aucune ligne). (traitement en 0,0002 seconde(s).)

```
SELECT vehicule.Id_Vehicule FROM vehicule INNER JOIN client ON vehicule.Id_Vehicule = client.Id_Client WHERE client.Id_Client = 122;
```

- c) Afficher l'immatriculation, l'âge et l'état de tous les véhicules.

✓ MySQL a retourné un résultat vide (c'est à dire aucune ligne). (traitement en 0,0002 seconde(s).)

```
SELECT vehicule.Immatriculation, vehicule.Age, vehicule.Etat FROM vehicule;
```

- d) Afficher les noms des clients et les adresses des clients qui habitent à << Nice >>.

✓ MySQL a retourné un résultat vide (c'est à dire aucune ligne). (traitement en 0,0002 seconde(s).)

```
SELECT client.Nom, client.Adresse FROM client WHERE client.Ville = "Nice";
```

- e) Affiche la liste des clients par ordre alphabétique croissant des noms.

✓ MySQL a retourné un résultat vide (c'est à dire aucune ligne). (traitement en 0,0003 seconde(s).)

```
SELECT client.Nom FROM client ORDER BY client.Nom ASC;
```

- f) Ajouter l'attribut kilométrage et Afficher la liste des voitures par ordre décroissant des compteurs (kilométrage).

✓ MySQL a retourné un résultat vide (c'est à dire aucune ligne). (traitement en 0,0008 seconde(s).)

```
ALTER TABLE vehicule ADD kilométrage VARCHAR(255);
```

✓ MySQL a retourné un résultat vide (c'est à dire aucune ligne). (traitement en 0,0002 seconde(s).)

```
SELECT vehicule.kilométrage FROM vehicule ORDER BY vehicule.kilométrage DESC;
```

- g) Afficher les informations sur les clients qui ont loué la voiture EW 25 EW.

✓ MySQL a retourné un résultat vide (c'est à dire aucune ligne). (traitement en 0,0004 seconde(s).)

```
SELECT * FROM client INNER JOIN vehicule on client.Id_Client = vehicule.Id_Vehicule WHERE vehicule.Immatriculation = "EW 25 EW";
```

- h) Afficher toutes les voitures noires.

✓ MySQL a retourné un résultat vide (c'est à dire aucune ligne). (traitement en 0,0004 seconde(s).)

```
ALTER TABLE vehicule ADD couleur VARCHAR(255);
```

✓ MySQL a retourné un résultat vide (c'est à dire aucune ligne). (traitement en 0,0009 seconde(s).)

```
SELECT vehicule.couleur FROM vehicule;
```

- i) Afficher toutes les voitures ayant un kilométrage <10000 km.

✓ MySQL a retourné un résultat vide (c'est à dire aucune ligne). (traitement en 0,0006 seconde(s).)

```
SELECT vehicule.kilométrage FROM vehicule WHERE vehicule.kilométrage < 10000;
```

- j) Afficher toutes les informations sur les locations réalisées avant 2018.

✓ MySQL a retourné un résultat vide (c'est à dire aucune ligne). (traitement en 0,0002 seconde(s).)

```
ALTER TABLE vehicule ADD date_retrait DATE;
```

✓ MySQL a retourné un résultat vide (c'est à dire aucune ligne). (traitement en 0,0002 seconde(s).)

```
SELECT * from vehicule WHERE vehicule.date_retrait < "2018-01-01";
```

- k) Afficher la moyenne des kilométrages de tous les véhicules du parc.

✓ Affichage des lignes 0 - 0 (total de 1, traitement en 0,0002 seconde(s).)

```
SELECT AVG(vehicule.kilométrage) FROM vehicule;
```

- l) Afficher toutes les locations réalisées en 2018.

✓ MySQL a retourné un résultat vide (c'est à dire aucune ligne). (traitement en 0,0002 seconde(s).)

```
SELECT vehicule.Id_Vehicule FROM vehicule WHERE vehicule.date_retrait > "2018-01-01" AND vehicule.date_retrait < "2019-01-01";
```

- m) Afficher le nombre de voitures ayant un kilométrage <10 000 kilomètres.

✓ MySQL a retourné un résultat vide (c'est à dire aucune ligne). (traitement en 0,0002 seconde(s).)

```
SELECT vehicule.Id_Vehicule FROM vehicule WHERE vehicule.kilométrage < 10000 ORDER BY vehicule.Id_Vehicule DESC;
```

Partie 2

- a) Obtenir la liste des véhicules empruntés et rendus le même jour ainsi que l'agence de rattachement.

✓ MySQL a retourné un résultat vide (c'est à dire aucune ligne). (traitement en 0,0002 seconde(s).)

```
SELECT vehicule.Id_Vehicule, agence.Nom FROM vehicule INNER JOIN agence ON vehicule.Id_Vehicule = agence.Id_Agence WHERE vehicule.date_retrait = vehicule.date_retour;
```

- b) Obtenir le nombre de véhicules pour chaque marque.

La requête SQL a été exécutée avec succès.

```
SELECT COUNT(vehicule.Id_Vehicule) FROM marque INNER JOIN vehicule on marque.Id_Marque = vehicule.Id_Vehicule;
```

- c) Obtenir les noms des clients qui ont loué plus de 10 véhicules de marque Renault.

✓ MySQL a retourné un résultat vide (c'est à dire aucune ligne). (traitement en 0,0003 seconde(s).)

```
SELECT client.Nom FROM client INNER JOIN vehicule on client.Id_Client = vehicule.Id_Vehicule INNER JOIN marque on vehicule.Id_Vehicule = marque.Id_Marque WHERE marque.Nom = "Renault" AND client.nombre_achat >= 10;
```

d) Obtenir le nombre d'agences et d'employés par pays.

La requête SQL a été exécutée avec succès.

```
SELECT agence.Nb_employes, COUNT(agence.Nom) FROM pays INNER JOIN agence on pays.Id_Pays = agence.Id_Agence;
```

Exercice 2

J'ai refait un [schéma](#) JMerise pour faire l'exercice 2 et je l'ai converti en fichier [SQL](#).

a) Les informations relatives aux étudiants (Code, Nom et Date de naissance) selon l'ordre alphabétique croissant du nom.

✓ MySQL a retourné un résultat vide (c'est à dire aucune ligne). (traitement en 0,0002 seconde(s).)

```
SELECT * FROM etudiant ORDER BY etudiant.NomEt ASC;
```

b) Les noms et les grades des enseignants de la matière dont le nom est 'BD'.

✓ MySQL a retourné un résultat vide (c'est à dire aucune ligne). (traitement en 0,0003 seconde(s).)

```
SELECT enseignant.NomEns, enseignant.GradeEns FROM enseignant INNER JOIN matiere on enseignant.CodeEns = matiere.CodeMat WHERE matiere.NomMat = "BD";
```

c) La liste distincte formée des noms et les coefficients des différentes matières qui sont enseignées par des enseignants de grade 'Grd3'.

✓ MySQL a retourné un résultat vide (c'est à dire aucune ligne). (traitement en 0,0003 seconde(s).)

```
SELECT matiere.CoeffMat, matiere.NomMat FROM matiere INNER JOIN enseignant ON matiere.CodeMat = enseignant.CodeEns WHERE enseignant.GradeEns = "Grd3";
```

d) La liste des matières (Nom et Coefficient) qui sont suivies par l'étudiant de code 'Et321'.

✓ MySQL a retourné un résultat vide (c'est à dire aucune ligne). (traitement en 0,0003 seconde(s).)

```
SELECT matiere.NomMat, matiere.CoeffMat FROM matiere INNER JOIN enseignant ON matiere.CodeMat = enseignant.CodeEns INNER JOIN etudiant ON enseignant.CodeEns = etudiant.CodeEt WHERE etudiant.CodeEt = 321;
```

e) Le nombre d'enseignants de la matière dont le nom est 'Informatique'.

La requête SQL a été exécutée avec succès.

```
SELECT COUNT(enseignant.NomEns) FROM enseignant INNER JOIN matiere ON enseignant.CodeEns = matiere.CodeMat WHERE matiere.NomMat = "Informatique";
```

exercice 3

J'ai refait un [schéma](#) JMerise pour faire l'exercice 3 et je l'ai converti en fichier [SQL](#).

a) Quelle est la composition de l'équipe Festina (Numéro, nom et pays des coureurs) ?

✓ MySQL a retourné un résultat vide (c'est à dire aucune ligne). (traitement en 0,0002 seconde(s).)

```
SELECT coureur.NuméroCoureur, coureur.NomCoureur, coureur.CodePays FROM coureur INNER JOIN equipe on  
coureur.NuméroCoureur = equipe.CodeEquipe WHERE equipe.NomEquipe = "Festina";
```

b) Quel est le nombre de kilomètres total du Tour de France 97 ?

La requête SQL a été exécutée avec succès.

```
SELECT COUNT(km) FROM attribuer_bonification;
```

c) Quel est le nombre de kilomètres total des étapes de type "Haute Montagne" ?

✓ Affichage des lignes 0 - 0 (total de 1, traitement en 0,0012 seconde(s).)

```
SELECT SUM(NbKm) AS TotalKmHauteMontagne FROM ETAPE WHERE CodeType = ( SELECT CodeType FROM TYPE_ETAPE WHERE  
LibelleType = 'Haute Montagne' );
```

d) Quels sont les noms des coureurs qui n'ont pas obtenu de bonifications ?

✓ MySQL a retourné un résultat vide (c'est à dire aucune ligne). (traitement en 0,0012 seconde(s).)

```
SELECT DISTINCT NomCoureur FROM COUREUR WHERE NuméroCoureur NOT IN ( SELECT DISTINCT NuméroCoureur FROM  
ATTRIBUER_BONIFICATION );
```

e) Quels sont les noms des coureurs qui ont participé à toutes les étapes ?

✓ MySQL a retourné un résultat vide (c'est à dire aucune ligne). (traitement en 0,0008 seconde(s).)

```
SELECT NomCoureur FROM COUREUR WHERE NOT EXISTS ( SELECT etape.NuméroEtap FROM ETAPE WHERE NOT EXISTS ( SELECT * FROM  
PARTICIPER WHERE PARTICIPER.NuméroEtap = ETAPE.NuméroEtap AND PARTICIPER.NuméroCoureur = COUREUR.NuméroCoureur ) );
```

f) Quel est le classement général des coureurs (nom, code équipe, code pays et temps des coureurs) à l'issue des 13 premières étapes sachant que les bonifications ont été intégrées dans les temps réalisés à chaque étape ?

✓ MySQL a retourné un résultat vide (c'est à dire aucune ligne). (traitement en 0,0011 seconde(s).)

```
SELECT COUREUR.NomCoureur, EQUIPE.CodeEquipe, COUREUR.CodePays, SUM(TIME_TO_SEC(PARTICIPER.TempsRealisé) +  
ATTRIBUER_BONIFICATION.NbSecondes) AS TempsTotal FROM COUREUR JOIN PARTICIPER ON COUREUR.NuméroCoureur =  
PARTICIPER.NuméroCoureur JOIN ETAPE ON PARTICIPER.NuméroEtap = ETAPE.NuméroEtap JOIN EQUIPE ON COUREUR.CodeEquipe =  
EQUIPE.CodeEquipe LEFT JOIN ATTRIBUER_BONIFICATION ON PARTICIPER.NuméroEtap = ATTRIBUER_BONIFICATION.NuméroEtap AND  
PARTICIPER.NuméroCoureur = ATTRIBUER_BONIFICATION.NuméroCoureur WHERE ETAPE.NuméroEtap <= 13 GROUP BY  
COUREUR.NuméroCoureur ORDER BY TempsTotal;
```

g) Quel est le classement par équipe à l'issue des 13 premières étapes (nom et temps des équipes) ?

✓ MySQL a retourné un résultat vide (c'est à dire aucune ligne). (traitement en 0,0004 seconde(s).)

```
SELECT EQUIPE.NomEquipe, SUM(TIME_TO_SEC(PARTICIPER.TempsRealisé) + ATTRIBUER_BONIFICATION.NbSecondes) AS TempsTotal
FROM EQUIPE JOIN COUREUR ON EQUIPE.CodeEquipe = COUREUR.CodeEquipe JOIN PARTICIPER ON COUREUR.NuméroCoureur =
PARTICIPER.NuméroCoureur JOIN ETAPE ON PARTICIPER.NuméroEtap = ETAPE.NuméroEtap LEFT JOIN ATTRIBUER_BONIFICATION ON
PARTICIPER.NuméroEtap = ATTRIBUER_BONIFICATION.NuméroEtap AND PARTICIPER.NuméroCoureur =
ATTRIBUER_BONIFICATION.NuméroCoureur WHERE ETAPE.NuméroEtap <= 13 GROUP BY EQUIPE.CodeEquipe ORDER BY TempsTotal;
```

Exercice 4

a) 1. la liste des clients de Marrakech.

✓ MySQL a retourné un résultat vide (c'est à dire aucune ligne). (traitement en 0,0002 seconde(s).)

```
SELECT * FROM Client WHERE adressescli LIKE '%Marrakech%';
```

☐ Profilage [[Éditer en ligne](#)] [[Éditer](#)] [[Expliquer SQL](#)] [[Créer le code source PHP](#)] [[Actualiser](#)]

b) la liste des produits (Numprod, désignation, prix) classés de plus cher au moins cher.

✓ MySQL a retourné un résultat vide (c'est à dire aucune ligne). (traitement en 0,0002 seconde(s).)

```
SELECT Numprod, designation, prix FROM Produit ORDER BY prix DESC;
```

c) noms et adresses des vendeurs dont le nom commence par la lettre 'M'.

✓ MySQL a retourné un résultat vide (c'est à dire aucune ligne). (traitement en 0,0002 seconde(s).)

```
SELECT Nomvendeur, adresse_vend FROM Vendeur WHERE Nomvendeur LIKE 'M%';
```

d) la liste des commandes effectuées par le vendeur "Mohammed" entre le 1er et 30 janvier 2020.

✓ MySQL a retourné un résultat vide (c'est à dire aucune ligne). (traitement en 0,0003 seconde(s).)

```
SELECT * FROM Commande INNER JOIN Vendeur ON Commande.Idvendeur = Vendeur.Idvendeur WHERE Vendeur.Nomvendeur =
'Mohammed' AND date_com BETWEEN '2020-01-01' AND '2020-01-30';
```

e) le nombre des commandes contenant le produit n° 365.

La requête SQL a été exécutée avec succès.

```
SELECT COUNT(*) FROM Commande WHERE Numprod = 365;
```

Exercice 5

a) La liste de tous les étudiants.

✓ Affichage des lignes 0 - 2 (total de 3, traitement en 0,0002 seconde(s).)

```
SELECT * FROM Etudiant;
```

☐ Profilage [Éditer en ligne] [Éditer] [Expliquer SQL] [Créer le code source PHP] [Actualiser]

☐ Tout afficher | Nombre de lignes : 25 | Filtrer les lignes: Chercher dans cette table | Trier par clé : Aucun(e)

Options supplémentaires

	numéro_carte_étudiant	Nom	Prénom	Date_Naissance
<input type="checkbox"/> Éditer Copier Supprimer	1234567	Ben Salah	Ahmed	1988-08-12
<input type="checkbox"/> Éditer Copier Supprimer	1234568	Ben Mahmoud	Sami	1990-09-02
<input type="checkbox"/> Éditer Copier Supprimer	1234569	Marzougui	Rami	1988-01-23

b) Nom et coefficient des matières.

✓ Affichage des lignes 0 - 1 (total de 2, traitement en 0,0002 seconde(s).)

```
SELECT nom_matière, coefficient FROM Matière;
```

☐ Profilage [Éditer en ligne] [Éditer] [Expliquer SQL] [Créer le code source PHP] [Actualiser]

☐ Tout afficher | Nombre de lignes : 25 | Filtrer les lignes: Chercher dans cette table | Trier par clé : Aucun(e)

Options supplémentaires

	nom_matière	coefficient
<input type="checkbox"/> Éditer Copier Supprimer	Base de données	1.5
<input type="checkbox"/> Éditer Copier Supprimer	Algorithme	3

c) Les numéros des cartes d'identité des étudiants dont la moyenne entre 7 et 12.

✓ Affichage des lignes 0 - 2 (total de 3, traitement en 0,0008 seconde(s).)

```
SELECT numéro_carte_étudiant FROM Note GROUP BY numéro_carte_étudiant HAVING AVG(note_examen) >= 7 AND AVG(note_examen) <= 12;
```

☐ Profilage [Éditer en ligne] [Éditer] [Expliquer SQL] [Créer le code source PHP] [Actualiser]

☐ Tout afficher | Nombre de lignes : 25 | Filtrer les lignes: Chercher dans cette table | Trier par clé : Aucun(e)

Options supplémentaires

	numéro_carte_étudiant
<input type="checkbox"/> Éditer Copier Supprimer	1234567
<input type="checkbox"/> Éditer Copier Supprimer	1234568
<input type="checkbox"/> Éditer Copier Supprimer	1234569

d) La liste des étudiants dont le nom commence par 'ben'.

✓ Affichage des lignes 0 - 1 (total de 2, traitement en 0,0001 seconde(s).)

```
SELECT * FROM Etudiant WHERE Nom LIKE 'Ben%';
```

☐ Profilage [Éditer en ligne] [Éditer] [Expliquer SQL] [Créer le code source PHP] [Actualiser]

☐ Tout afficher | Nombre de lignes : 25 | Filtrer les lignes: Chercher dans cette table | Trier par clé : Aucun(e)

Options supplémentaires

	numéro_carte_étudiant	Nom	Prénom	Date_Naissance
<input type="checkbox"/> Éditer Copier Supprimer	1234567	Ben Salah	Ahmed	1988-08-12
<input type="checkbox"/> Éditer Copier Supprimer	1234568	Ben Mahmoud	Sami	1990-09-02

e) Le nombre des étudiants qui ont comme matière '12518'.

La requête SQL a été exécutée avec succès.

```
SELECT COUNT(DISTINCT numéro_carte_étudiant) FROM Note WHERE code_matière = 12518;
```

☐ Profilage [Éditer en ligne] [Éditer] [Expliquer SQL] [Créer le code source PHP] [Actualiser]

Options supplémentaires

COUNT(DISTINCT numéro_carte_étudiant)
3

f) La somme des coefficients des matières.

✓ Affichage des lignes 0 - 0 (total de 1, traitement en 0,0002 seconde(s).)

```
SELECT SUM(coefficient) FROM Matière;
```

☐ Profilage [Éditer en ligne] [Éditer] [Expliquer SQL] [Créer le code source PHP] [Actualiser]

☐ Tout afficher | Nombre de lignes : 25 | Filtrer les lignes: Chercher dans cette table

Options supplémentaires

SUM(coefficient)
4.5

g) Les noms des étudiants qui ont une note_examen >10.

✓ Affichage des lignes 0 - 1 (total de 2, traitement en 0,0003 seconde(s).)

```
SELECT Nom, Prénom FROM Etudiant WHERE numéro_carte_étudiant IN ( SELECT numéro_carte_étudiant FROM Note WHERE note_examen > 10 );
```

☐ Profilage [Éditer en ligne] [Éditer] [Expliquer SQL] [Créer le code source PHP] [Actualiser]

☐ Tout afficher | Nombre de lignes : 25 | Filtrer les lignes: Chercher dans cette table | Trier par clé : Aucun(e)

Options supplémentaires

	Nom	Prénom
<input type="checkbox"/> Éditer Copier Supprimer	Ben Salah	Ahmed
<input type="checkbox"/> Éditer Copier Supprimer	Ben Mahmoud	Sami

h) Afficher les noms et les coefficients des matières étudiées par l'étudiant "01234568".

✓ Affichage des lignes 0 - 0 (total de 1, traitement en 0,0002 seconde(s).)

```
SELECT Matière.nom_matière, Matière.coefficient FROM Matière INNER JOIN Note ON Matière.code_matière = Note.code_matière WHERE Note.numéro_carte_étudiant = 01234568;
```

☐ Profilage [Éditer en ligne] [Éditer] [Expliquer SQL] [Créer le code source PHP] [Actualiser]

☐ Tout afficher | Nombre de lignes : 25 | Filtrer les lignes: Chercher dans cette table

Options supplémentaires

nom_matière	coefficient
Algorithmme	3