| **Cahier des charges** |
| --- |

L’entreprise **La Grenouille Verte** propose à ses clients des randonnées à VTT en France (Lozère, Aude …) mais aussi à l’étranger (Espagne, Inde …). Elle offre à ses clients appelés **randonneurs** des circuits soigneusement balisés et classés en fonction de leurs difficultés (Débutant, Intermédiaire, …) mais également des prestations comme la réservation d’hôtels, l’encadrement de randonnées par un accompagnateur …

L’entreprise connaissant un succès grandissant, le personnel ne peut plus, comme il l’a fait jusqu’à présent, noter sur des fiches les renseignements relatifs aux randonneurs, aux randonnées et aux réservations.

Pour gérer les données de l’entreprise, Corinne Lebon a fait acheter un système de gestion de bases de données relationnelles (SGBDR). Il est envisagé d’utiliser le SGBDR Mysql qui hébergera les tables de l’application ainsi que Phpmyadmin pour la gestion des interfaces clients.

**Tout le système repose sur le modèle client-serveur.**

|  |
| --- |

| **Travail À faire** |
| --- |

**Q1.**  Définir la notion de Middleware et donner des exemples.

**Un middleware sert a la communication entre les applications comme :**

* les serveurs web ;
* les systèmes d’authentification unique ;
* les systèmes d’accès aux bases de données (requêtes) ;

<https://www.atelog2i.com/actualites-atelog/a-quoi-sert-un-middleware/>

<https://www.redhat.com/fr/topics/middleware/what-is-middleware>

**Q2.** Créer la base de données VTT.les instructions sont ci-dessous.

|  |  |
| --- | --- |
|  |  |

Aller dans file ensuite New Model

1. Cliquer pour changer le nom de la base
2. Nommer votre base de données VTT
3. Cliquer sur add diagram
4. Cliquer sur le tableau ensuite sur la page pour ajouter un tableau
5. Cliquer sur le tableau ajouté afin de le remplir.
6. Créer la structure de la table « **randonnees** » à partir des renseignements suivants : le tableau décrit chaque champ de la table. Préférer le type varchar au type Text.

**documentation:** <https://www.w3schools.com/sql/sql_datatypes.asp>

1. Si votre table est fini allez dans **database** ensuite sur **synchronize Model, toujours next en sélectionnant votre base**

|  |
| --- |

**Votre table sera ajouté dans les SCHÉMAS comme suit:**

|  |
| --- |

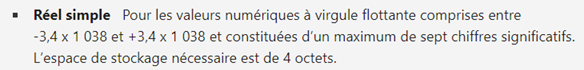
**ATTENTION !! pour les noms de la base ou des champs : ne pas mettre d’espaces ni de caractères particuliers (°, accents, etc ….). Les tirets sont acceptés.**

|  |
| --- |

Remarque : les codes Numéro de randonnée et Numéro de dépôt sont organisés de la manière suivante : les 2 premiers caractères désignent le pays ( ES pour Espagne par exemple), les 2 suivants la région (Ca pour Catalogne par exemple) et un numéro ordonnant les randonnées dans une région ; ESCA1 est la première randonnée en Catalogne. Les dépôts suivent le même principe de codification.

Pour la taille du champs





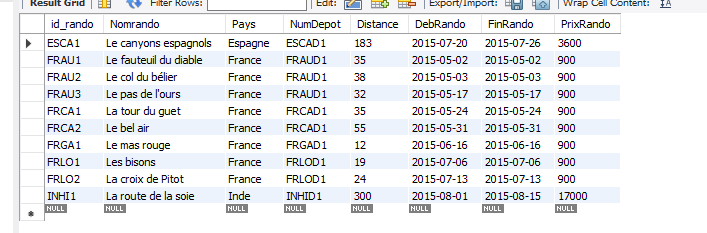
**Q3.** Insérer les valeurs dans la table randonnée en utilisant le langage SQL.

voir : <https://sql.sh/cours/insert-into>

**ATTENTION : mysql prend le format américain pour les dates : AA/MM/JJ**

**Table RANDONNÉE**

|  |
| --- |



**Q4.** Créer la table randonneur, clique droit sur Tables

|  |
| --- |

1. donner le nom de la table “randonneur”.
2. remonter l’espace de saisie pour déclarer les colonnes.
3. Remplir les colonnes de la table.

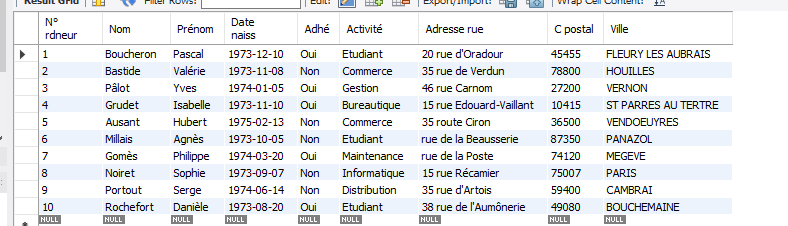
|  |
| --- |

Pour le id\_randonneur choisir primary key et aurto increment.

1. insérer les valeurs en utilisant le fichier randonneur.sql

<https://drive.google.com/file/d/1bzsDdBI33S1n1Q6Gd_ENJFOM1XgflIX5/view?usp=sharing>

|  |
| --- |



**Q5.** Vos tables ne respectent pas la notion d’intégrité référentielle.

1. Suivre la vidéo ci-dessous pour comprendre la notion de clé primaire(primary key) et clé et étrangers,

<https://www.youtube.com/watch?v=4JhXRll-jkQ>

Pourquoi faut-il mettre CASCADE au moment de la suppression ou la modification de la clé étrangers(foreign key) ?.

1. Modifier la structure de votre table **randonnée** pour y insérer une clé primaire

|  |
| --- |

1. Créer une relation entre les deux tables.

|  |
| --- |

1. remplir votre clé étrangers sur la table

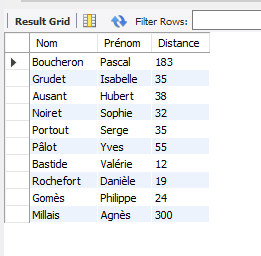
|  |
| --- |

**Q6.** Lire cours:

<https://drive.google.com/file/d/1MTEI3BeU-likwTieaT75MBoKAVM8DJ07/view?usp=sharing>

Après lecture du cours, faire une requête pour afficher le nom et le prénom des randonneurs ainsi que leurs distances.

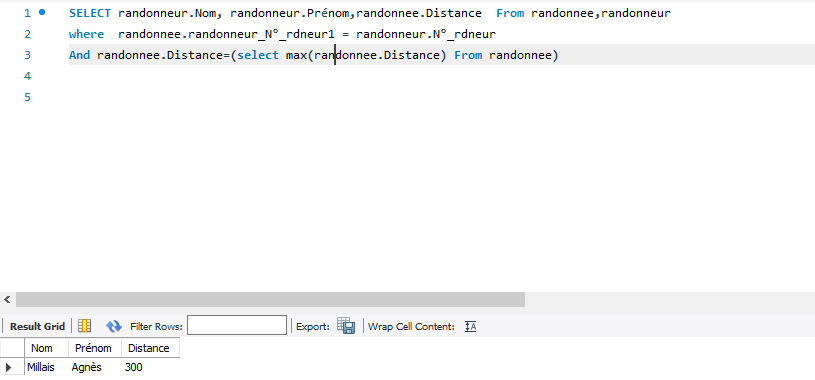
|  |
| --- |



**Q7.** Faire une synthèse en précisant les points les plus importants qui favorisent l’utilisation d’une jointure entre tables.

L’utilisation de la jointure sert à optimiser la lisaison entre tables

**Q8)** Sélectionner le nom et prénom du randonneur qui fait la plus grande distance.



**Q9.** Sélectionner le nom, le prénom ,la date de naissance des étudiants français.

**Q10.** Visualiser et tester la vidéo suivant:

<https://www.youtube.com/watch?v=tDa5vF1-Qys>

1. test 1. cliquer sur le lien

[SQL Joins](https://www.w3schools.com/sql/sql_join.asp)

1. test2

<https://www.w3schools.com/sql/sql_join_inner.asp>

1. test3

<https://www.w3schools.com/sql/sql_join_left.asp>

1. test4

<https://www.w3schools.com/sql/sql_join_right.asp>

1. test5

<https://www.w3schools.com/sql/sql_join_full.asp>

Quelle jointure correspond le plus à notre cas, Réaliser la requête pour avoir le même résultat précédent.

| **Le schéma relationnel de la base** |
| --- |

**Introduction:**

Nous avons vu au cours précédent que l'informatisation des données courantes de l'entreprise nécessite la création de plusieurs tables, reliées entre elles via des clés. Un schéma relationnel va donc décrire toutes les tables et les relations entre elles. On distingue 2 grands types de relation

* les relations 1­n : de un à plusieurs
* les relations n­n : de plusieurs à plusieurs

| **Les relations 1-n** |
| --- |

Examinons la relation qui existe entre le réalisateur et le film. Excluons le cas où un film est réalisé par plusieurs réalisateurs. Nous pouvons alors construire les deux phrases suivantes :

* Un film est réalisé par un et un seul réalisateur: 1,1
* Un réalisateur a réalisé un ou plusieurs films: 1,n

Si nous plaçons le nom du réalisateur dans la même table que les données du film,

nous créons de la redondance à chaque fois qu'un film est réalisé par un même réalisateur.

**Nous placerons donc les réalisateurs et les films dans des tables distinctes, et nous créerons une relation entre ces tables.**

|  |
| --- |

| **Les relations n-n** |
| --- |

On souhaiterait connaître les acteurs qui jouent dans les différents films et inversement pour un acteur on souhaiterait connaître les films dans lesquels il a joué.

Si l'on choisit le film 2012---->

|  |
| --- |

Si l'on choisit le lien : Roland Emmerich, on liste tous les films du réalisateur→

|  |
| --- |

Et pour chaque film, nous avons des liens avec les principaux acteurs. Ces liens vont permettre d'afficher les films dans lesquels ils ont joué.

Exemple avec John Cusak --->

|  |
| --- |

Comme pour les réalisateurs, on va devoir créer une table acteur, en effet on ne peut pas faire figurer les acteurs dans la table film. Certains films n'ayant pas d'acteur ( exemple les films d'animation) alors que d'autres ont un nombre d'acteurs très important ( exemple le seigneur des anneaux).

|  |
| --- |

Quelle est la relation entre la table Film et la table Acteur ?

* les relations 1­n : de un à plusieurs.
* les relations n­n : de plusieurs à plusieurs.

Non !!!!!!

Nous nous trouvons alors face à une relation de "plusieurs à plusieurs" ou "n-­n". Pour gérer une telle relation, on ne peut pas faire figurer l'identifiant d'une des tables dans l'autre table. En effet on ne peut pas mettre dans la table Film l'identifiant de l'acteur, on choisit lequel si il y a une centaine d'acteurs. De la même manière, on ne peut pas faire figurer l'identifiant du film dans la table Acteur.

Pour gérer une telle relation, il faut créer une table supplémentaire que l'on va appeler FilmActeur , dans laquelle on introduit les identifiants des tables Film et Acteur.

|  |
| --- |

**A la lecture des 2 enregistrements de cette table, on sait que dans le film dont l'identifiant est 39 (2012) ont joué les acteurs dont l'identifiant est 1 ( john Cusak) et 4 (amanda Peet).**

Cet exemple nous montre que toute relation n-­n peut être scindée en deux relations en introduisant une table supplémentaire appelée table de jonction. Ainsi dans la table **FilmActeur** on aura 2 clés étrangères que l'on va utiliser pour construire l'identifiant et ainsi éviter d'avoir 2 enregistrements identiques.

**SQL :**

On veut connaître les acteurs qui ont joué dans Avatar :

**SELECT Acteur.nom, Film.titre**

**FROM Acteur, Film, FilmActeur**

**WHERE Acteur.idacteur = FilmActeur.idacteur**

**AND Film.idfilm = FilmActeur.idfilm**

**AND Film.titre = 'Avatar'**

| **TP SQL: Les jointures.** |
| --- |

Créer une nouvelle base de données et importer le script video.sql dans votre base de données phpmyadmin dans.

lien du script:

: <https://drive.google.com/file/d/1cYqdswP_PNrsXSpqWM7dZouSQ9hCVcWI/view?usp=sharing>

|  |
| --- |

|  |
| --- |

**Travail sur les requêtes:**

1. Afficher le nom des acteurs qui ont joué dans Avatar.
2. Afficher le nom des acteurs qui ont joué sous la direction de Georges Lucas.
3. Afficher le nom des acteurs qui ont joué dans des films fantastiques.
4. Indiquer dans quel film a joué Dicaprio.
5. Afficher le nom des acteurs qui ont joué dans des films dont le réalisateur est américain.
6. Afficher le nom et la date de naissance de l'acteur le plus jeune dans la base.
7. Afficher le nom de l'acteur et le nombre de films dans lequel ils ont joué.

<https://www.w3resource.com/sql/aggregate-functions/max-count.php>