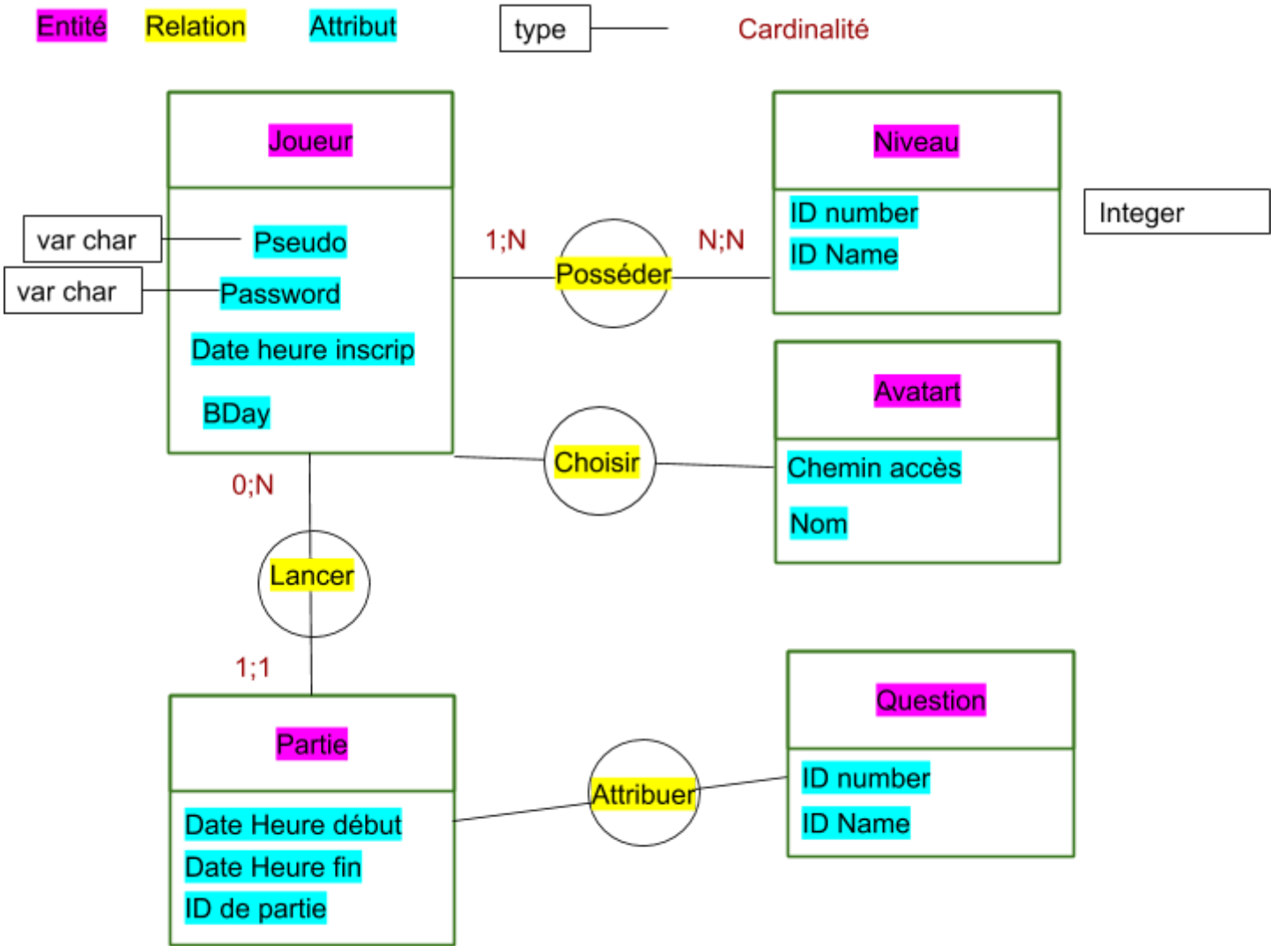
Lancer JMerise en ligne de commande :

- cd dossier dans lequel se trouve JMerise
- java -jar JMerise.jar &

Code pile : https://www.codepile.net/pile/BE1G0pL6



Attention, ici ⇑les cardinalités ne sont pas toutes écrites, il en faudrait une par extrémité de relation pour être juste. Idem pour les types qui doivent décrire chacun des attributs

Entités commencent par une majuscule et sont au singulier
Attributs commencent par une minuscule
Relations (aussi appelées associations) verbes à l'infinitif. Unique dans le modèle.
Cardinalités :

- 0,1
- 1,1
- 0,n
- 1,n

Clé primaire composite = clé primaire composée de plusieurs éléments

Étude de cas #2: Autoroute

FX COTE 13:58 (Modification : 16:33)

Vous devez concevoir un MCD/MLD/MPD permettant de gérer les tarifs et le trafic autoroutier.

Dans ce document, l'autoroute Lyon → Grenoble sera utilisée comme exemple

A chaque véhicule correspond une classe de véhicule. Il existe 5 classes de véhicule : de la classe 1 (véhicule léger) à la classe 5 (2 roues). On supposera qu'un véhicule ne change pas de classe.

La classe du véhicule détermine le prix du trajet entre deux gares de péages.

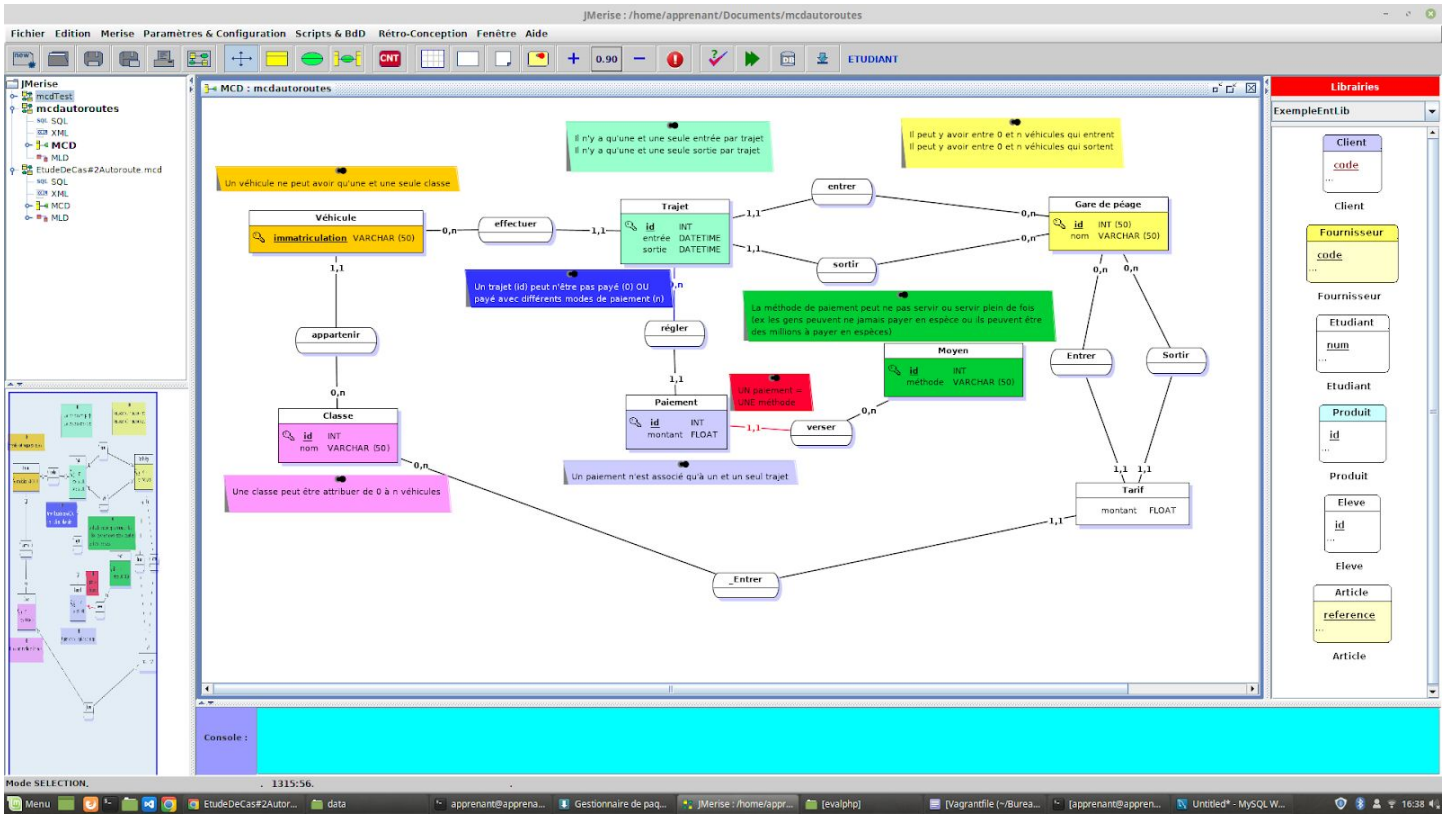
Exemple de trajet : Lyon → Grenoble
Véhicule : AN-985-MZ
Classe : 1
Gare d'entrée : Lyon à 08h02
Gare de sortie : Grenoble à 09h04
Prix : 7,00 euros
A noter : le prix du trajet inverse est systématiquement le même

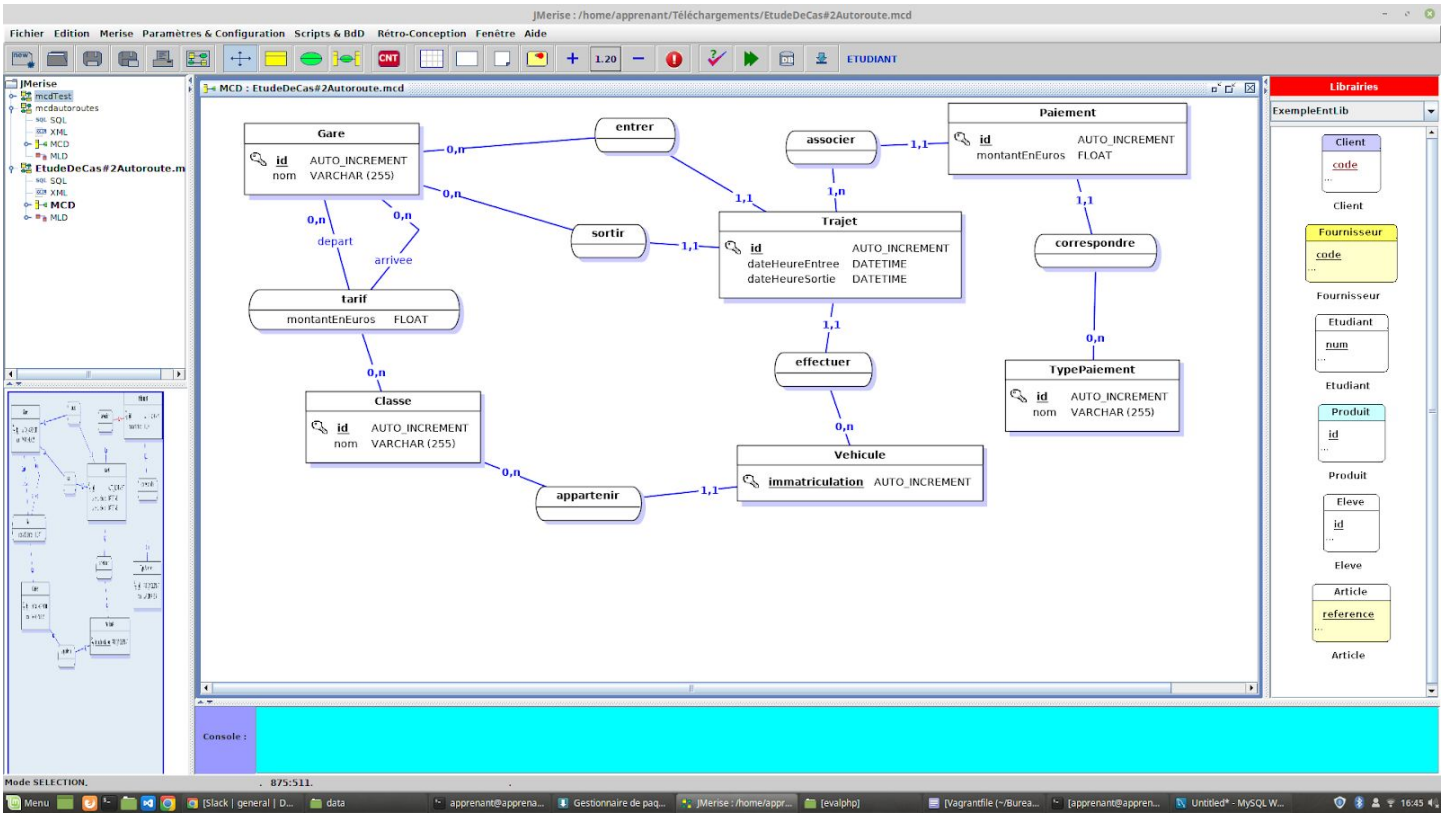
Il est possible de payer le trajet par CB, espèces et/ou par chèque voyage.

EtudeDeCas#2Autoroute.mcd
Fichier binaire

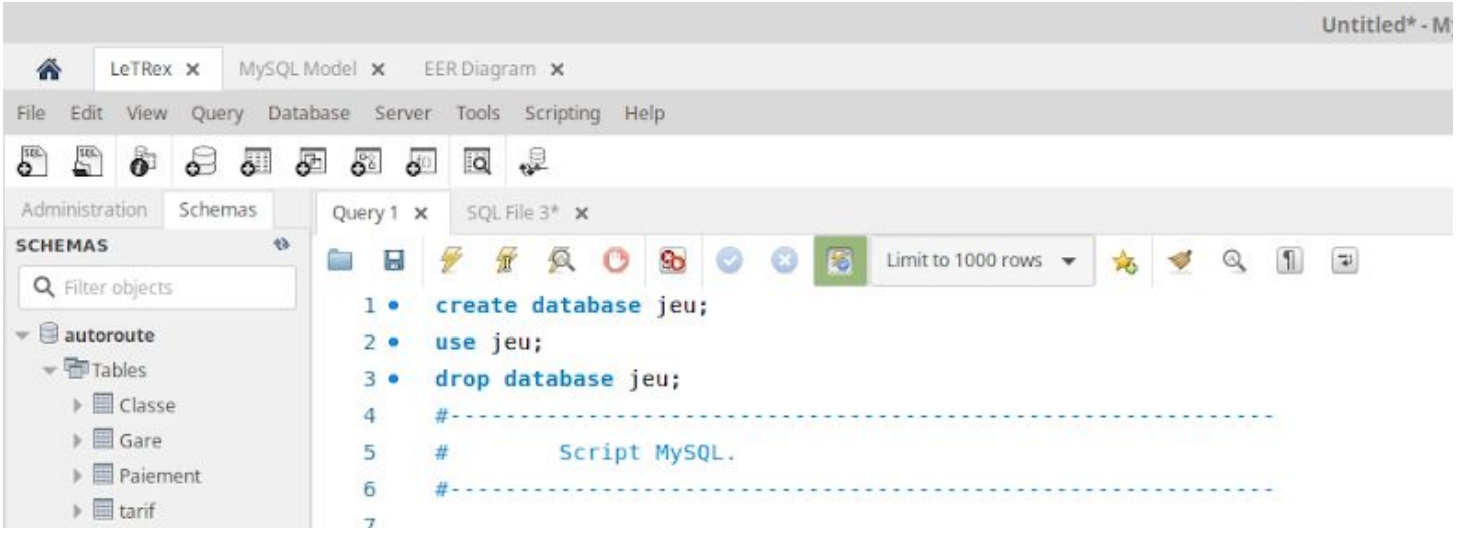
Commentaires ajoutés au cours

Ajouter un commentaire au cours...





 Sur JMerise pour obtenir le code à coller dans MySQL Workbench



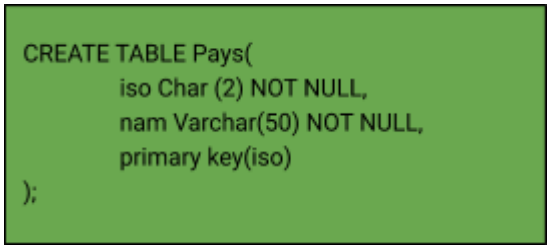
Sur MySQL Workbench :

- Coller code obtenu dans JMerise
- Ajouter create database nomBDD;
- use nomBDD;
- clic sur l'éclair

Pour abandonner une BDD :

- drop database nomBDD;

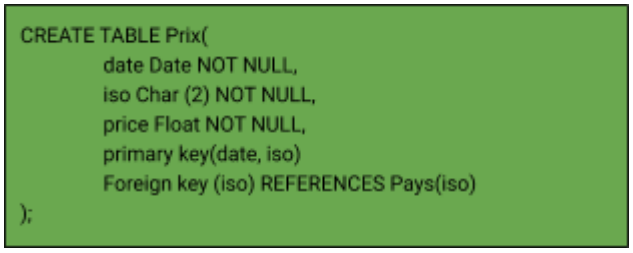
Requêtes SQL demandées:
01- Donner la commande pour créer la base de données sqlburger
create database sqlburger;
02- Donner la commande pour utiliser la base de données sqlburger
use sqlburger;
03- Écrire le script de création de la table pays




```
CREATE TABLE Pays(
      iso  Char (2) NOT NULL ,
      name Varchar (50) NOT NULL
            ,CONSTRAINT Pays_PK PRIMARY KEY (iso)
)ENGINE=InnoDB;
```

Nom de la table
Attribut type (longueur)
Clé primaire de la table
Moteur de stockage dans lequel se trouvera la table

04- Écrire le script de création de la table prix




```
CREATE TABLE Prix(
      iso_Pays Char (2) NOT NULL ,
      iso      Char (2) NOT NULL ,
      date     Date NOT NULL ,
      price    Float NOT NULL
            ,CONSTRAINT Prix_PK PRIMARY KEY (iso_Pays,iso,date)

            ,CONSTRAINT Prix_Pays_FK FOREIGN KEY (iso_Pays) REFERENCES Pays(iso)
)ENGINE=InnoDB;
```

Foreign key (iso_Pays) fait référence à Pays(iso)

05- Écrire le code SQL permettant d'afficher le nom de toutes les tables

```
USE `sqlburger`;
 SHOW TABLES;

 SHOW TABLES FROM `sqlburger`;
```

06- Écrire la requête qui permet d'ajouter l'Espagne (code ISO: ES)

```
INSERT INTO  Pays (iso, name)
            VALUES ('ES', 'Espagne');
```

07- Écrire la requête qui permet d'ajouter le Portugal (code ISO: PT) et le Luxembourg (code ISO: LU)

```
INSERT into Pays (iso, name)
VALUES ('PT', 'Portugal')
       ('LU', 'Luxembourg');
```

08- Écrire la requête qui permet d'insérer le prix du SQL Burger en France à la date du 04/11/2019: 3.80 €

```
INSERT into Pays (iso, name) VALUES ('FR', 'France');
INSERT INTO Prix (iso, date, price) VALUES ('FR', '2019-11-04', 3.8);
```

09- Écrire la requête permettant d'obtenir le prix moyen du SQL Burger dans tous les pays dont le nom commence par la lettre F

```
SELECT AVG(Prix.price)
FROM Pays, Prix
WHERE  Pays.iso=Prix.iso and Pays.name like 'F%';
```

SELECT AVG(Prix.price)
FROM Pays, Prix
WHERE
-- Version old school d'une jointure
-- Jointure = collage entre deux tables sur une colonne commune
Pays.iso=Prix.iso
-- ET deuxième critère: on ne veut que les pays dont le nom commence par F
AND Pays.name like 'F%';

10- Écrire la requête permettant d'obtenir le prix moyen du SQL Burger au mois d'Octobre 2019

```
SELECT AVG(price)
FROM Prix
WHERE date like '20019-10-%';
```

11- Écrire la requête permettant d'obtenir le prix minimum du SQL Burger au mois d'Octobre 2019

```
SELECT MIN(price)
FROM Prix
WHERE date like '20019-10-%';
```

12- Écrire la requête permettant d'obtenir le prix du SQL Burger le plus récent en France

```
SELECT (Prix.price)
FROM Pays, Prix
WHERE Pays.iso=Prix.iso AND
Pays.name = 'France'
ORDER BY Prix.date DESC
LIMIT 1;
```

13- Écrire la requête permettant d'obtenir le prix du SQL Burger le plus récent en France, en Espagne et au Portugal

```
SELECT Prix.price, Pays.name
FROM Pays, Prix
WHERE Pays.iso=Prix.iso
-- AND (Pays.name = 'France' OR Pays.name = 'Portugal' OR Pays.name = 'Espagne')
AND Pays.name in ('France', 'Portugal', 'Espagne')
ORDER BY Prix.date DESC
LIMIT 3;
```

14- Écrire la requête permettant d'obtenir les pays dont le nom commence par F et dont le prix du SQL Burger n'a jamais atteint 3 euros

```
SELECT Prix.price, Pays.name
FROM Pays, Prix
WHERE Pays.iso=Prix.iso
AND Pays.name like 'F%' AND price < 3;
```

15- Écrire la requête permettant d'obtenir la liste des pays pour lesquels aucun prix de SQL Burger n'a été défini

```
SELECT Pays.name
FROM Pays
WHERE iso NOT IN (SELECT iso FROM Prix);
```

16- Écrire la requête permettant d'obtenir le prix moyen par pays (les pays seront triés par ordre alphabétique) et le résultat de la requête présenté suivant le modèle ci-dessous:
nom
code_iso
prix_moyen

```
SELECT Pays.name, Pays.iso, AVG (Prix.price)
FROM Pays, Prix
WHERE Pays.iso=Prix.iso
GROUP BY iso
ORDER BY Pays.name;
```

17- Donner les droits select, insert et update à l'utilisateur gourmetDeMadrid sur la machine locale

```
-- QUESTION 17 --
-- Je cree User @'localhost' que sur ca machine et indentifier par mdp
CREATE USER gourmetDeMadrid@'localhost' IDENTIFIED BY 'patate';
-- je lui accorde select , insert update on nom de la bdd ou du table to le nom user
GRANT SELECT, INSERT, UPDATE ON sqlburger.* TO 'gourmetDeMadrid'@'localhost';
-- met en place les privilège
FLUSH PRIVILEGES;
```

18- Écrire le code SQL permettant d'ajouter un index sur la colonne nom de la table pays

ALTER TABLE Pays add index `index` (`nom`) ;

CREATE UNIQUE INDEX idx_Pays_nom ON Pays(nom);

19- Écrire le code SQL qui supprime tous les prix concernant l'Italie

DELETE FROM Prix where codeIso='IT'

20- Écrire le code SQL qui supprime la table prix

DROP table Prix

L'application Web YouTravel.fr accueille deux types d'utilisateur:

• consommateur
• réceptif

Après s'être authentifié, le consommateur choisit deux villes d'un même pays, exemple: Tokyo et Kyoto. Il choisit ensuite une date de départ et une date de retour puis il précise les voyageurs, c'est-à-dire la liste complète des personnes participant au voyage.

Un ensemble d'activités (disponibles aux dates choisies) lui sera ensuite proposé.
Ces activités (quelque soit le pays de destination) sont classées par thème: culture, sport, bien-être, gourmet.
Une activité appartient à un et un seul thème.

Le consommateur peut choisir une activité pour une demi-journée (matin ou après-midi).
Il peut laisser libre (i.e sans activité) une ou plusieurs demi-journées du voyage.

A noter: une nuit du voyage permet au(x) voyageur(s) de rejoindre l'autre ville sélectionnée.

Exemple de voyage choisi par le consommateur:

Voyageur 1: Mlle Fabienne Martin, date de naissance: 05/12/1986
Voyageur 2: M. Yann Bonnet, date de naissance: 18/08/1984

Départ de Lyon le 21/02/2020, arrivée à Tokyo le 21/02/2020

21/02/2020: Tokyo
Matinée gourmet: Visite du marché de poissons de Tokyo et dégustation de sushis et sashimis
Après-midi culturelle: visite du musée Ghibli à Tokyo

22/02/2020: Kyoto
Matinée culturelle: Visite des Temples de Kyoto

Après-midi bien-être: Onsen (spa) de Kyoto

Retour à Lyon le 23/02/2020.

Montant global du voyage: 1500 euros

Le consommateur doit régler en ligne l'ensemble du voyage dès qu'il l'a validé.

A noter:

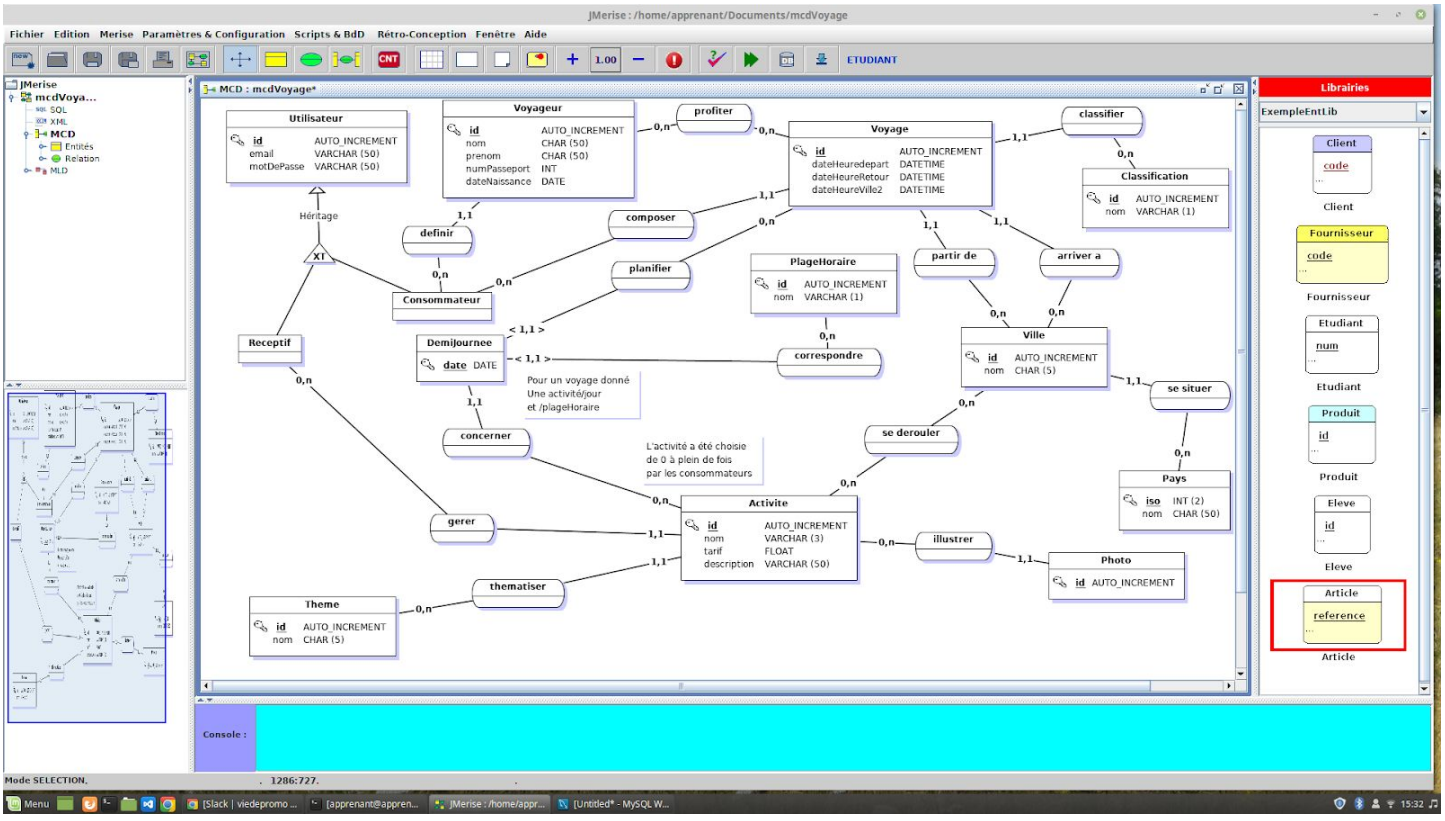
- le calcul du prix des billets d'avion ainsi que la réservation des billets d'avion sont assurés par Amadeus, un système secondaire.

- les réservations des hôtels sont assurées par un système secondaire de réservation hôtelière. L'application Web demandera au consommateur la classification souhaitée (2 étoiles, 3 étoiles, 4 étoiles ou 5 étoiles). Ce choix s'appliquera à toutes les nuitées.

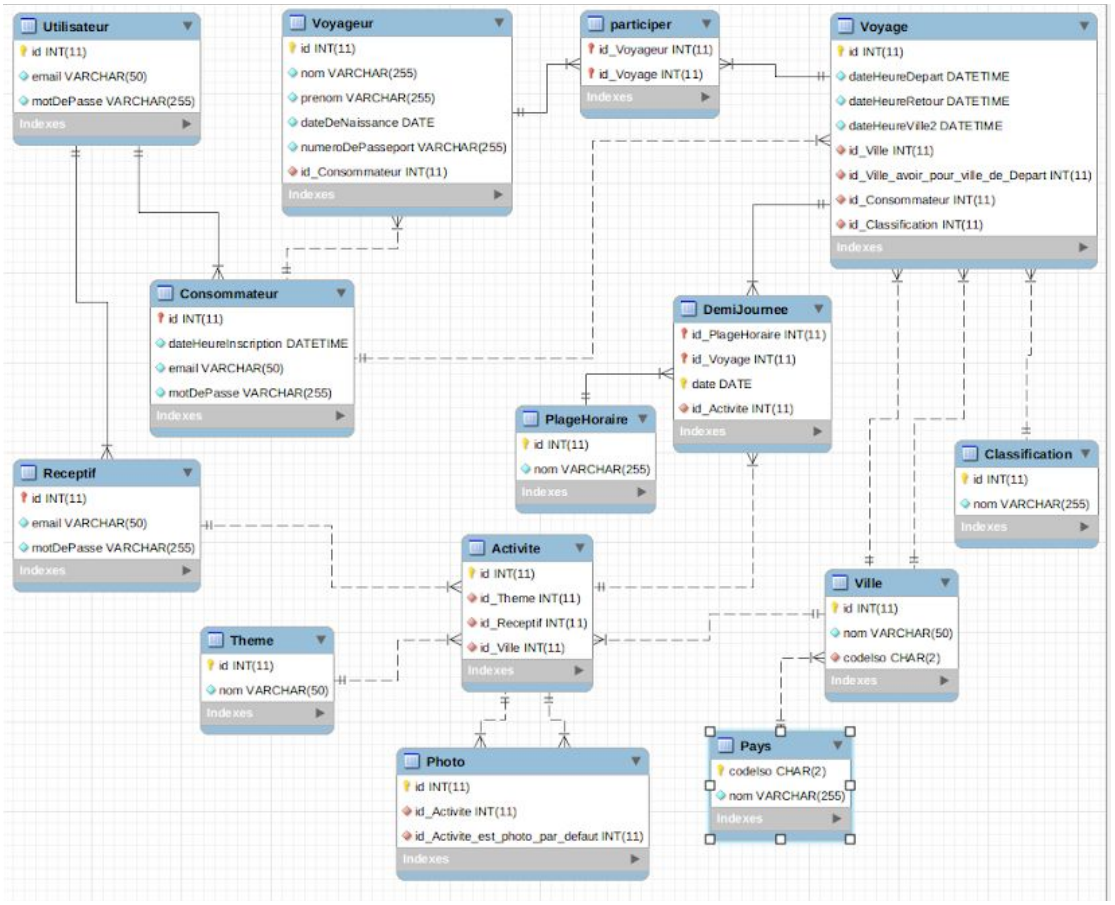
- le paiement du voyage est assuré par un système secondaire de paiement en ligne.

Le réceptif assure un ensemble d'activités.
Après s'être authentifié, il peut ajouter des activités en précisant une description en plusieurs langues et en téléversant un ou plusieurs photos de l'activité.
Une activité doit avoir une photo par défaut.
Le prix de l'activité est systématiquement exprimé en euros

Travail demandé:
1- Rédiger les règles de gestion
2- Rédiger le dictionnaire de données
3- Établir le MCD avec JMerise



4- Créer la base de données MySQL youtravel dans MySQL Workbench



5- Ajouter les thèmes en une seule requête SQL

```
INSERT INTO Theme (nom)
VALUES ('culture'),('sport'),('bienEtre'),('gourmet');
```

6- Ajouter deux pays de votre choix

```
INSERT INTO Pays (codeIso, nom)
VALUES ('JP', 'japon'),('UY','uruguay');
```

7- Ajouter deux villes dans les pays créés en 6

```
INSERT INTO Ville (codeIso, nom)
VALUES ('JP', 'tokyo'),('UY','montevideo');
```

8- Écrire la requête ajoutant 4 réceptifs

```
INSERT INTO Utilisateur (email, motDePasse)
VALUES ('tanaka.naoko@gmail.com','uma'),
('luis.suarez@gmail.com','celeste'),
('jose.mujica@gmail.com','pepe'),
('totoro@gmail.com','ghibli');
```

9- Écrire la requête ajoutant une colonne libelle de type varchar(255) dans la table Activite

```
ALTER TABLE Activite
ADD libelle Varchar(255);
```

10- Écrire la requête ajoutant une colonne prix de type float dans la table Activite

```
ALTER TABLE Activite
ADD prix Float;
```

11- Écrire les requêtes qui ajoutent des activités dans chaque ville

```
INSERT INTO Activite (id_Theme, id_Receptif, id_Ville, libelle, prix)
VALUES ('3','1','1','onsen',400),('2', '2', '2', 'foot', 400),
('4','4','1','petit dej au flux cafe',1207),('1','3','2','andes museum',301.34);
```

12- Écrire la requête donnant le nombre d'activité dans chaque pays

```
SELECT Pays.nom, COUNT(Activite.id) AS NB_ACTIVITES
FROM Activite, Ville, Pays
WHERE Activite.id_Ville=Ville.id AND Ville.codeIso=Pays.codeIso
GROUP BY Pays.nom
ORDER BY NB_ACTIVITES DESC;
```

13- Écrire la requête donnant le prix moyen des activités pour chaque ville

```
SELECT Ville.nom, AVG(prix) AS PX_ACTIVITE
FROM Activite, Ville
WHERE Activite.id_ville=Ville.id
GROUP BY Ville.nom
ORDER BY PX_ACTIVITE DESC;
```

14- Écrire la requête donnant pour chaque pays les activités triées par popularité

```
SELECT COUNT(Pays.nom) AS NB_ACTIVITES, Pays.nom
FROM DemiJournee, Activite, Ville,Pays
WHERE Ville.id=Activite.id_Ville AND DemiJournee.id_Activite=Activite.id AND Pays.codeIso=Ville.codeIso
GROUP BY Pays.codeIso
-- moins de caract requete plus efficace--
ORDER BY NB_ACTIVITES DESC;
```

15- Écrire la requête donnant les activités dont le libellé débute par A ou par G

```
SELECT Activite.libelle
FROM Activite
WHERE Activite.libelle like 'a%' OR Activite.libelle like 'g%';
```

16- Ajouter un index pour optimiser le temps de réponse de la requête précédente
Index : Allège les ressources nécessaires à l'exécution d'une requête.
Lorsqu'on fera une recherche sur le libelle des activités, au lieu de "scanner" toute la BDD, MySQL scannera l'index

```
CREATE INDEX index_abc ON Activite(libelle);
```

17-Supprimer les attributs email et motDePasse générés par erreur dans les tables Consommateur et Receptif :

```
ALTER TABLE Receptif DROP email;
ALTER TABLE Receptif DROP motDePasse;
ALTER TABLE Consommateur DROP email;
ALTER TABLE Consommateur DROP motDePasse;
```

On imagine une application Web destinée à trois types d'utilisateurs:
- jardinier
- fournisseur de graines potagères
- administrateur du site

Un utilisateur est soit jardinier soit fournisseur soit administrateur du site.

Grâce à l'application, le jardinier peut:

1. s'inscrire (en précisant son nom, son prénom, une adresse email, sa date de naissance ainsi qu'un mot de passe). Un email est envoyé automatiquement. Pour confirmer son inscription, le jardinier doit cliquer sur un lien hypertext contenu dans le mail.

2. se connecter (en donnant une adresse email et son mot de passe)

3. (une fois connecté) connaître ce qu'il peut planter dans son potager en précisant le numéro de la semaine dans l'année.

4. (une fois connecté) coller et sauvegarder une recette dans un champ de texte puis obtenir la liste des graines potagères dont il a besoin. Exemple: le jardinier colle la recette de la moussaka. L'application en extrait les ingrédients puis liste les graines d'aubergines, de tomates et d'oignons jaunes.

A noter: lorsque le jardinier s'inscrit sur l'application, elle enregistre la date et l'heure de l'inscription.

Pour chaque graine potagère, le jardinier a besoin des informations suivantes:
- description
- famille à laquelle la graine appartient, exemples: aromatique, légume, haricot
- une collection de photo
- numéro de semaine de plantation min
- numéro de semaine de plantation max
- espacement entre pieds (exprimé en centimètres)
- espacement entre lignes (exprimé en centimètres)
- conseils
- le prix et le poids du ou des sachet(s) disponible(s) ainsi que le nom du fournisseur associé

Un tableau d'exemple est disponible en pièce jointe.

Grâce à l'application, le fournisseur peut:

1. s'inscrire (en précisant son nom, son prénom, une adresse email, un mot de passe ainsi qu'un numero de fax)

2. se connecter (en donnant une adresse email et son mot de passe)

3. gérer les graines potagères qu'il est en mesure de fournir en précisant pour chaque graine le prix du sachet ainsi que son poids.

Grâce à l'application, l'administrateur du site peut:

1. s'inscrire (en précisant son nom, son prénom, une adresse email, son numero de téléphone portable ainsi qu'un mot de passe)

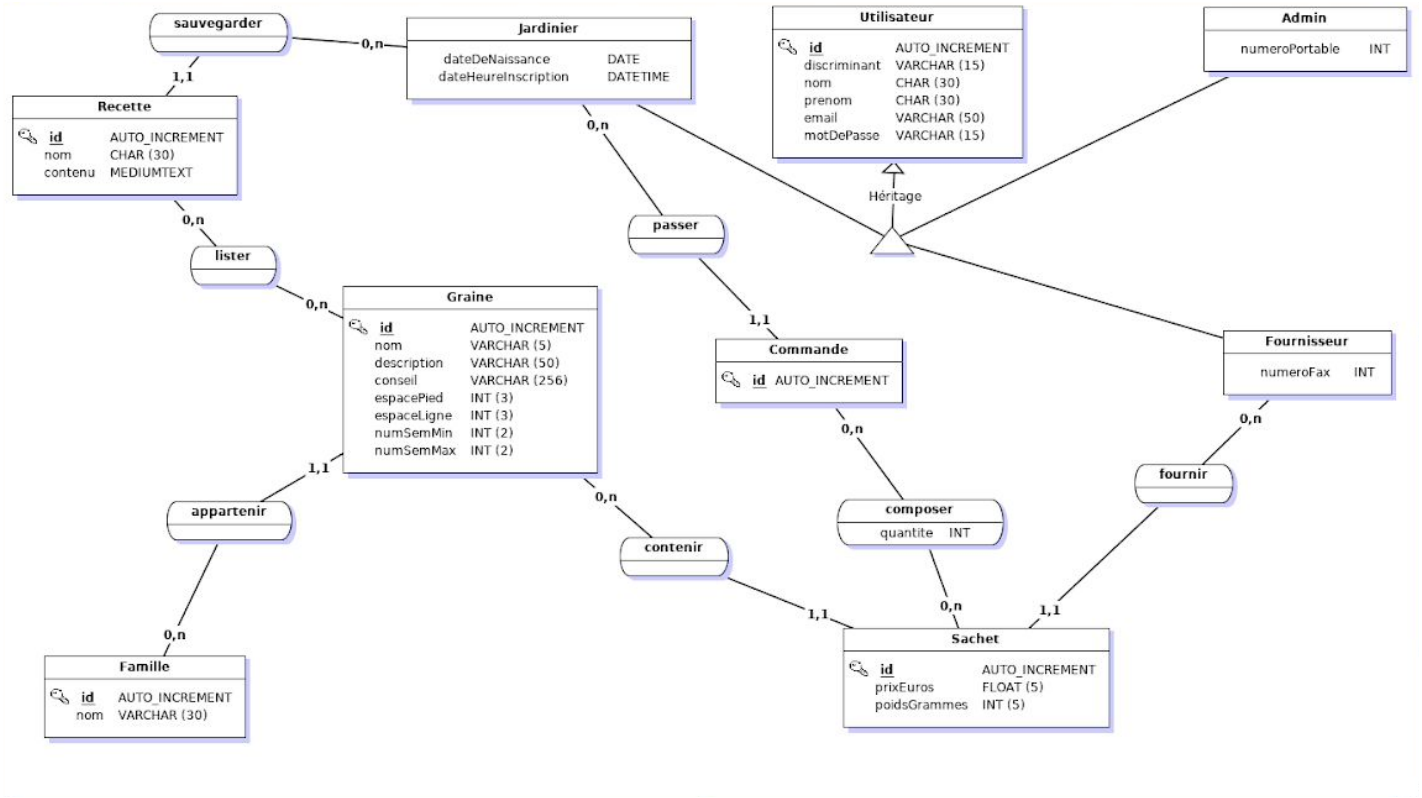
2. se connecter (en donnant une adresse email et son mot de passe)

3. gérer les graines potagères

3. gérer les jardiniers

4. gérer les fournisseurs

Travail demandé:
1- Établir le MCD
2- Créer la base de données MySQL graines dans MySQL Workbench



3- Donner le script de création de la table graine

```
CREATE TABLE Graine(
        id          Int  Auto_increment  NOT NULL ,
        nom         Varchar (5) NOT NULL ,
        description Varchar (50) NOT NULL ,
        conseil     Varchar (256) NOT NULL ,
        espacePied  Int NOT NULL ,
        espaceLigne Int NOT NULL ,
        numSemMin   Int NOT NULL ,
        numSemMax   Int NOT NULL ,
        id_Famille  Int NOT NULL,
        PRIMARY KEY (id),
        FOREIGN KEY (id_Famille) REFERENCES Famille(id)
);
```

4- Écrire la requête ajoutant les familles

```
INSERT INTO Famille (nom)
VALUES          ('ombéllifères'),
                ('cucurbitacées'),
                ('chénopodiacées'),
                (‘labiées’),
                (‘fabiacées’)
                (‘composées’),
                (‘liliacées’),
                (‘crucifères’),
                (‘solanacées’);
```

CORRIGER NOM D’UNE VALEUR :

```
UPDATE Famille
SET nom = 'fabacées'
WHERE id = 5;
```

5- Écrire les requêtes ajoutant les graines présentées sur le tableau en pj

```
INSERT INTO Graine (nom, description, conseil, espacePied, espaceLigne, numSemMin, numSemMax, id_Famille)
VALUES('Bette', 'Feuilles vertes, côtes blanches', 'Faîtes-en des gratins', 5, 40, 14, 26, 3),
        ('Betterave', 'Truc dégueu mais jolie couleur', 'Faut vraiment être con pour planter ça', 11, 40, 14, 22, 3),
        ('carotte', 'Truc orange et bon', 'Mangez-en plein', 0, 20, 14, 31, 1),
        ('Concombre', 'Truc vert et rafraichissant', 'Faîtes-en des salades', 50, 90, 14, 22, 2),
        ('Courgette', 'Truc vert cool', 'Mangez-en crues et cuites', 80, 80, 14, 28, 2),
        ('HaricotVert', 'Miam', 'Cuisez les pas trop', 25, 40, 14, 31, 5),
        ('LaituePommée', 'Truc vert feuillu', 'Salades à gogo', 30, 30, 14, 31, 6),
        ('Oignon', 'Truc rond et jaune', 'faites le fondre', 10, 25, 5, 18, 7),
        ('Radis', 'truc rose et blanc', 'Arrosez les beaucoup sinon ça pique', 0, 20, 9, 39, 8),
        ('Tomate', 'Truc rond et rouge', 'Faites en des salades et des sauces', 70, 70, 14, 26, 9);
```

6- Écrire une requête SQL permettant d'obtenir les graines qu'il sera possible de planter entre les semaines 20 et 23 inclus

```
SELECT Graine.nom
FROM Graine
WHERE Graine.numSemMin <= 20 AND Graine.numSemMax >= 23;
```

7- Écrire une requête permettant d'augmenter de 5% le poids de tous les sachets

```
UPDATE Sachet
SET poidsGrammes=poidsGrammes*1.05;
```

8- Écrire une requête permettant d'augmenter de 50% le prix tous les sachets de la famille labiées (=aromatiques)

```
UPDATE Sachet
SET prixEuros=prixEuros*1.5
WHERE Sachet.id IN ( Select Sachet.id
        FROM Graine, Famille
        WHERE Sachet.id_Graine=Graine.id
        AND Graine.id_Famille=Famille.id
        AND Famille.id=4);
```

9- Écrire une requête SQL permettant d'effacer toutes les graines de la famille fabacées (=haricots)

```
DELETE FROM Graine
WHERE id_Famille=5;
```

Les plages italiennes sont découpées en concessions privées. Chaque concession est divisée en plusieurs files. Sur chaque file, un ensemble de parasols sera aligné. L'emplacement des parasols est fixe.

L'exemple en contient 8 files, chaque file contient 36 parasols. La concession possède une allée centrale pour faciliter l'accès aux parasols.

La première file est la plus proche de la mer.
Un parasol est identifié par son numéro de file et par un numéro dans la file, exemple: 15F4, représenté sur la matrice par un X.

Chaque année la saison estivale débute le 01 Juin et se termine le 06 Octobre.

Pendant cette saison, pour une durée minimale d'un jour, un client peut louer jusqu'à 10 parasols. Sous chaque parasol, il faut louer au choix:
- un lit
- deux lits
- un fauteuil de réalisateur
- un fauteuil de réalisateur et un lit
- deux fauteuils de réalisateur

Sur chaque équipement placé sous un parasol, on écrit le numéro du parasol.

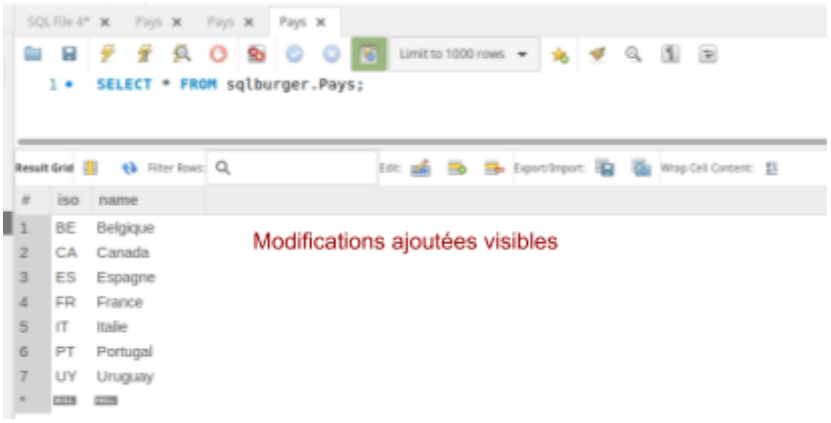
Le montant de la location est fonction:
- de son type (un lit coûte plus cher qu'un fauteuil de réalisateur)
- du nombre de jours de location
- du nombre d'années de location du client concerné
- de la proximité avec la mer
- du lien de parenté avec le gérant de la concession (exemples : 50 % de réduction pour les frères et sœurs du concessionnaire, 25 % de réduction pour les cousins et cousines du concessionnaire)

Travail demandé:
1- Créer la base de données MySQL meriseplage à l'aide du fichier mcd en pj
2- Écrire la requête qui donne tous les clients qui ont loué au moins deux fois
3- Écrire la requête qui donne les vingt locataires italiens ayant passé le plus grand nombre d'années à la concession
4- Écrire la requête qui permet d'obtenir toutes les locations de 2014 de clients italiens
ayant duré plus de 15 jours
5- Écrire la requête SQL qui permet d'obtenir le nombre de locations par année par mois

---

Le coup de foudre sert à appliquer toutes les lignes de commande
Le coup de foudre curseur sert à appliquer la ligne sélectionnée

Clique droit sur la table à laquelle on a appliqué une modification
Puis clic sur Select rows



#	iso	name
1	BE	Belgique
2	CA	Canada
3	ES	Espagne
4	FR	France
5	IT	Italie
6	PT	Portugal
7	UY	Uruguay

Pour décompresser Tomcat :
tar -zxvf apache-tomcat-9.0.27.tar.gz

cd apache-tomcat-9.0.27/bin

./startup.sh

sur navigateur :
localhost:8080

Toutes les applications développées “sur” tomcat se trouvent dans webapps

---

RESSOURCES

https://fr.wikibooks.org/wiki/MySQL/Parcourir_les_bases_de_donn%C3%A9es#Show_all_tables

https://openclassrooms.com/fr/courses/1959476-administrez-vos-bases-de-donnees-avec-mysql/1960995-inserez-des-donnees

https://sql.sh/
https://sqlpro.developpez.com/cours/definitions/

Code IATA Villes
http://www.logistiqueconseil.org/Fiches/Transport-aerien/Code-IATA-villes.pdf
Code ISO Pays
https://fr.wikipedia.org/wiki/ISO_3166-1

```
--TP SQLBURGER---
Insert into Pays Values('FR','FRANCE');
Insert into Prix Values('FR','2019-11-04','3.80');
Select ROUND(100*AVG(Prix.prixEnEuros))/100 FROM Pays, Prix WHERE Prix.codeIso= Pays.codeIso AND
Pays.nom LIKE 'F%';

SELECT AVG(prixEnEuros) From Prix WHERE  date LIKE '2019-10-%';
Insert into Prix Values('FR','2019-10-02','3.95');
SELECT ROUND(100*MIN(prixEnEuros))/100 FROM Prix WHERE date BETWEEN '2019-10-01' AND
'2019-10-31';
Insert into Prix value('FR','2018-05-24','4.5');
-- JE SELECTIONNE LE PRIX LE PLUS RECENT EN FRANCE--
SELECT Prix.prixEnEuros
From Pays, Prix
-- JE FAIS UNE JOINTURE POUR ACCEDER AU NOM FRANCE--
WHERE Prix.codeIso= Pays.code
AND Pays.nom='FRANCE'
-- JE TRIE PAR DATE  (ORDER BY) D'UNE FACON DESCENDANTE DU PLUS PETIT AU PLUS
GRAND(DESC) OU ASC A L INVERSE--
ORDER BY Prix.date desc
-- J'EN AFFICHE QU UN POUR VOIR SEULEMENT LE PLUS RECENT AVEC LIMIT --
LIMIT 1;
Insert into Prix Values('ES','2019-08-24','3.2');
Insert into Prix Values('PT','2019-06-24','3.1');
Insert into Pays Values('ES','ESPAGNE');
Insert into Pays Values('PT','PORTUGAL');
-- QUESTION 13--
SELECT  Prix.prixEnEuros, Pays.nom
FROM Pays, Prix
WHERE Prix.codeIso=Pays.codeIso
AND Pays.nom in('FRANCE','ESPAGNE','PORTUGAL')
ORDER BY Prix.date desc
LIMIT 3;
-- Ou --
SELECT Prix.prixEnEuros, Pays.nom, max(Prix.date)
FROM Pays, Prix
```

```
WHERE Pays.codeIso=Prix.codeIso AND Pays.nom='FRANCE'
GROUP by Pays.codeIso
UNION
SELECT Prix.prixEnEuros, Pays.nom, max(Prix.date)
FROM Pays, Prix
WHERE Pays.codeIso=Prix.codeIso AND Pays.nom='ESPAGNE'
GROUP by Pays.codeIso
UNION
SELECT Prix.prixEnEuros, Pays.nom , max(Prix.date)
FROM Pays, Prix
WHERE Pays.codeIso=Prix.codeIso AND Pays.nom='PORTUGAL'
GROUP by Pays.codeIso;

-- QUESTION 14--
insert into Pays Values('FI','FINLANDE');
insert into Prix Values('FI','2019-09-04','2.92');
SELECT Pays.nom, Prix.prixEnEuros
FROM Pays, Prix
WHERE Prix.codeIso=Pays.codeIso
AND Pays.nom like 'F%'
AND Prix.prixEnEuros < 3;
insert into Pays Values ('IT','ITALIE');
-- QUESTION 15 --
SELECT Pays.nom
FROM Pays
Where codeIso not in (select distinct codeIso From Prix);
-- QUESTION 16--
SELECT  Pays.nom, Pays.codeIso, AVG(Prix.prixEnEuros)
From Prix, Pays
WHERE Prix.codeIso=Pays.codeIso
GROUP BY Pays.codeIso
ORDER BY Pays.nom;
-- QUESTION 17 --
-- Je cree User @'localhost' que sur ca machine et indentifier par mdp
CREATE USER gourmetDeMadrid@'localhost' IDENTIFIED BY 'patate';
-- je lui accorde select , insert update on nom de la bdd ou du table to le nom user
GRANT SELECT, INSERT, UPDATE ON sqlburger.* TO 'gourmetDeMadrid'@'localhost';
-- met en place les privilège
FLUSH PRIVILEGES;
-- Question 18 ---
ALTER TABLE Pays add index `index`  (`nom`) ;
-- Question 19 --
ALTER TABLE Prix Drop column  `prixEnEuros`;
-- Question 20 --
DROP table Prix  ;


--TP Voyage Voyage --

INSERT INTO Receptif VALUES("1","toto@gmail.com","patates");
-- J'ajoute des activités--
INSERT INTO Activite(id_Receptif,id_Theme, id_Ville, libelle)
```

```
values
("1","2","4","randonnee dans les fjords"),
("1","3","4","decouverte sauna typique"),
("1","1","3","visite des musees de la ville "),
("1","4","3","degustation des produits dans le port"),
("1","2","3","randonnee ski nordique"),
("1","2","2","tour de madrid en velo"),
("1","1","1","visite de la sagrada familia"),
("1","4","1","degustation de tapas"),
("1","3","1","massage iberique au chorizo");
-- INSERER DES PRIX QUI SERONt egale a 100* id pour chaque id de activite--
UPDATE Activite
SET prix = id*100
WHERE id>0;

-- COMPTE LE NOMBRE D'ACTIVITE PAR VILLE --

SELECT Ville.nom , count(Activite.id) AS NB_ACTIVITES
FROM Activite, Ville
WHERE Ville.id=Activite.id_Ville
GROUP BY Ville.id
ORDER BY NB_ACTIVITES DESC;

--mOYENNNE DES PRIX DES ACTIVITE PAR VILLE--
SELECT Ville.nom, AVG(Activite.prix) AS PX_ACTIVITES
FROM Activite, Ville
WHERE Ville.id=Activite.id_Ville
GROUP BY Ville.id
ORDER BY PX_ACTIVITES DESC;


-- ACTIVITE LES PLUS POPULAIRE PAR PAYS --
SELECT COUNT(Pays.nom) AS NB_ACTIVITES, Pays.nom
FROM DemiJournee, Activite, Ville,Pays
WHERE Ville.id=Activite.id_Ville AND DemiJournee.id_Activite=Activite.id AND Pays.codeIso=Ville.codeIso
GROUP BY Pays.codeIso -- moins de caract requete plus efficace--
ORDER BY NB_ACTIVITES DESC;

-- requete pour select libelle commencant par a ou g --
SELECT Activite.libelle
FROM Activite
WHERE Activite.libelle LIKE "A%"
OR Activite.libelle LIKE "G%";

-- Je cree un index sur la colonne libelle pour parcourir que la colone libele dans ma requette au dessus et non toute la table Activite--
CREATE INDEX index_abc ON Activite(libelle);


--potager td---
use potager;
Alter table Graine drop photo;
```

```
INSERT INTO Famille Value ("1","aromatique");


-- selectionne le nom des graines qui se plante entre la semaine 20 et 22 --
INSERT INTO Graine(nomLegume, semainePlantationMin, semainePlantationMax, espacementPiedsCm,
espacementLignesCm, id_Famille)
VALUES
("carotte","13","14","60","5","1"),
("courgette","22","23","120","75","1"),
("haricot","22","23","100","55","1"),
("laitue","13","14","55","25","1"),
("oignon","14","14","45","8","1"),
("radis","14","16","40","2.5","1"),
("tomate","21","24","90","80","1");
SELECT Graine.nomLegume
FROM Graine
WHERE Graine.semainePlantationMin>"20"
AND Graine.semainePlantationMax<="23";
-- augmenter le poid de tout les sachet --
UPDATE Sachet SET poids = (poids*0.05 + poids) ;
-- augmenter les prix de 10% de toute la famille aromatique --

UPDATE Sachet SET prix = (prix*0.1 + prix)
WHERE Sachet.id IN( Select Sachet.id
FROM  Graine, Famille
WHERE Sachet.id_Graine=Graine.id
AND Graine.id_Famille=Famille.id
AND Famille.nomFamille="aromatique");
-- supprimer --
DELETE FROM  Graine  where Graine.nomLegume in
(Select Graine.nomLegume
From Famille
where Graine.id_Famille=Famille.id
AND Famille.nomFamille="haricots");
```

Partager DB sur GitHub :

mysqldump -uroot -p nomDeLaBasdeDeDonnées > nomSouhaitéDuFichier.sql

envoyer le fichier .sql sur github

Quand on veut récupérer la DB et l'utiliser :

Télécharger le fichier .sql sur github puis

mysql -uroot -p < nomDuFichier.sql