

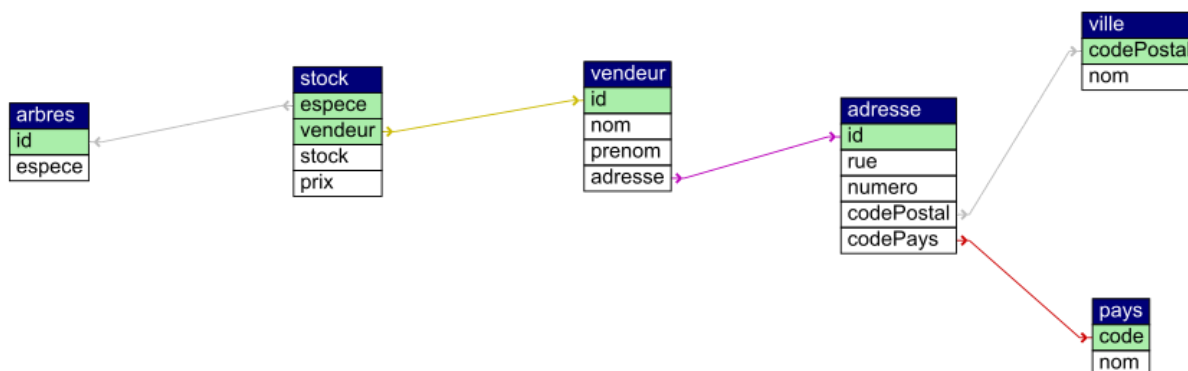
# Compléments de bases de données

Jean Vanneste

Version colorisé : <https://github.com/JeanVanneste/database/blob/master/request.md>

Utilisation de MySQL

## Structure de la base de données



## Requêtes simples

1. Prix maxima, minima, prix moyens, nombres de vendeurs par espèces

```

SELECT arbres.espece , max(prix), min(prix), avg(prix), count(vendeur)
FROM stock INNER JOIN arbres ON arbres.id = stock.espece
GROUP BY arbres.espece;

```

2. Vendeurs de pins par code postal

```

SELECT nom, prenom, codePostal
FROM arbres INNER JOIN stock ON arbres.id = stock.espece
      INNER JOIN vendeur ON stock.vendeur = vendeur.id
      INNER JOIN adresse ON vendeur.adresse = adresse.id
WHERE arbres.espece = 'Pin'
ORDER BY codePostal;

```

3. Totaux de stocks par vendeurs

```

SELECT nom, prenom, SUM(stock)
FROM vendeur INNER JOIN stock ON vendeur.id = stock.vendeur
GROUP BY vendeur.id
ORDER BY vendeur.nom;

```

#### 4. Vue d'ensemble de toute la DB sous forme de VUE

```
CREATE VIEW db AS
SELECT arbres.espece, vendeur.nom, vendeur.prenom, adresse.rue,
adresse.numero, ville.codePostal, ville.nom as ville, pays.code, pays.nom
as pays, stock.prix, stock.stock
FROM arbres JOIN stock ON arbres.id = stock.espece
JOIN vendeur ON vendeur.id = stock.vendeur
JOIN adresse ON vendeur.adresse = adresse.id
JOIN ville ON ville.codePostal = adresse.codePostal
JOIN pays ON adresse.codePays = pays.code
ORDER BY arbres.espece, stock.prix;
```

#### 5. Requête pour augmenter de 10% le prix de vente de tous les pins

```
UPDATE stock INNER JOIN arbres ON stock.espece = arbres.id
SET stock.prix = (stock.prix * 1.1)
WHERE arbres.espece = 'Pin';
```

#### 6. Requête pour afficher tous les prix (HTVA) et les les prix (TVAC) pour les chênes de tout les vendeurs

```
SELECT vendeur.nom, vendeur.prenom, stock.prix AS `prix (HTVA)`,
(stock.prix * 1.21) AS `prix (TVAC)`
FROM stock INNER JOIN arbres ON arbres.id = stock.espece
INNER JOIN vendeur ON vendeur.id = stock.vendeur
WHERE arbres.espece = 'Chêne';
```

#### 7. Requête permettant d'extraire les informations nécessaires pour créer des étiquettes d'adresses pour tous les vendeurs de 7531 Tournai

```
SELECT CONCAT(vendeur.nom, ' ', vendeur.prenom, '\n', adresse.rue, ' ',
adresse.numero, '\n7531 TOURNAI\n', pays.code, ' ', pays.nom) AS Étiquette
FROM vendeur INNER JOIN adresse ON vendeur.adresse = adresse.id
INNER JOIN ville ON ville.codePostal = adresse.codePostal
INNER JOIN pays ON pays.code = adresse.codePays
WHERE ville.codePostal = 7531 AND ville.nom = 'TOURNAI';
```

## Base de données

```
-- phpMyAdmin SQL Dump
-- version 4.6.6deb5
-- https://www.phpmyadmin.net/
```

```
--
-- Client : localhost:3306
-- Généré le : Lun 08 Juillet 2019 à 19:59
-- Version du serveur : 5.7.26-0ubuntu0.18.04.1
-- Version de PHP : 7.2.19-0ubuntu0.18.04.1

SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
SET time_zone = "+00:00";

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8mb4 */;

--
-- Base de données : `arbres`
--

--
-- Structure de la table `adresse`
--

CREATE TABLE `adresse` (
  `id` int(11) NOT NULL,
  `rue` varchar(60) NOT NULL,
  `numero` int(4) NOT NULL,
  `codePostal` int(5) NOT NULL,
  `codePays` varchar(2) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Contenu de la table `adresse`
--

INSERT INTO `adresse` (`id`, `rue`, `numero`, `codePostal`, `codePays`)
VALUES
(1, 'Rue des pinsons ', 2, 7730, 'BE'),
(2, 'Rue Longuesault ', 3, 7500, 'BE'),
(3, 'Résidence du Bourgeon ', 14, 7531, 'BE'),
(4, 'Rue Longuesault', 14, 7500, 'BE'),
(5, 'Chemin de la Parmerie', 4, 7522, 'BE'),
(6, 'Rue du Pont d\'Eau', 174, 7502, 'BE'),
(7, 'Résidence du Jonc', 14, 7531, 'BE'),
(8, 'Rue de la Place', 11, 7502, 'BE'),
(14, 'Rue Desaix', 12, 59200, 'FR'),
(15, 'Rue du Vert Pré', 14, 59830, 'FR'),
(16, 'Résidence du Bourgeon', 5, 7531, 'BE'),
(17, 'Rue de la Place', 10, 7502, 'BE'),
(18, 'rue de Liège', 174, 7502, 'BE'),
(20, 'Rue Charles Fourrier', 28, 59000, 'FR');
```

```
-- -----  
  
--  
-- Structure de la table `arbres`  
--  
  
CREATE TABLE `arbres` (  
  `id` int(11) NOT NULL,  
  `espece` varchar(16) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;  
  
--  
-- Contenu de la table `arbres`  
--  
  
INSERT INTO `arbres` (`id`, `espece`) VALUES  
(1, 'Chêne'),  
(2, 'Hêtre'),  
(3, 'Pin'),  
(4, 'Houx'),  
(5, 'Taxus'),  
(6, 'Bruyère');  
  
-- -----  
  
--  
-- Doublure de structure pour la vue `db`  
-- (Voir ci-dessous la vue réelle)  
--  
CREATE TABLE `db` (  
  `espece` varchar(16)  
, `nom` varchar(30)  
, `prenom` varchar(30)  
, `rue` varchar(60)  
, `numero` int(4)  
, `codePostal` int(5)  
, `ville` varchar(20)  
, `code` varchar(2)  
, `pays` varchar(20)  
, `prix` int(3)  
, `stock` int(6)  
) ;  
  
-- -----  
  
--  
-- Structure de la table `pays`  
--  
  
CREATE TABLE `pays` (  
  `code` varchar(2) NOT NULL,  
  `nom` varchar(20) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
--
-- Contenu de la table `pays`
--

INSERT INTO `pays` (`code`, `nom`) VALUES
('BE', 'Belgique'),
('FR', 'France');

-----

--
-- Structure de la table `stock`
--

CREATE TABLE `stock` (
  `espece` int(11) NOT NULL,
  `vendeur` int(11) NOT NULL,
  `stock` int(6) NOT NULL,
  `prix` int(3) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Contenu de la table `stock`
--

INSERT INTO `stock` (`espece`, `vendeur`, `stock`, `prix`) VALUES
(1, 1, 40, 15),
(1, 9, 14, 41),
(1, 15, 10, 14),
(2, 2, 32, 10),
(2, 9, 12, 21),
(2, 16, 11, 47),
(3, 3, 32, 10),
(3, 6, 17, 32),
(3, 8, 20, 47),
(3, 10, 20, 51),
(3, 12, 14, 41),
(3, 14, 41, 51),
(3, 17, 14, 24),
(4, 3, 14, 32),
(4, 4, 11, 19),
(4, 11, 12, 13),
(5, 3, 24, 12),
(5, 5, 15, 21),
(5, 18, 14, 88),
(6, 7, 18, 41),
(6, 13, 32, 44);

-----

--
-- Structure de la table `vendeur`
--
```

```

CREATE TABLE `vendeur` (
  `id` int(11) NOT NULL,
  `nom` varchar(30) NOT NULL,
  `prenom` varchar(30) NOT NULL,
  `adresse` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Contenu de la table `vendeur`
--

INSERT INTO `vendeur` (`id`, `nom`, `prenom`, `adresse`) VALUES
(1, 'VERMEULEN', 'Axel', 1),
(2, 'DENIS', 'Geoffrey', 4),
(3, 'DEFERNEZ', 'Romain', 3),
(4, 'DENIS', 'Tressy', 2),
(5, 'VERNIER', 'Amandine', 5),
(6, 'ES.SAYEN', 'Sélim', 16),
(7, 'DUMONT', 'Hélène', 6),
(8, 'JOLIOT', 'Romain', 7),
(9, 'DUPONT', 'Emilie', 8),
(10, 'FERRET', 'Jonathan', 14),
(11, 'LOCHT', 'Nathalie', 15),
(12, 'HENRIEST', 'Rémy', 20),
(13, 'MESPLOMB', 'Marie', 17),
(14, 'VACCA', 'Jonathan', 14),
(15, 'ACRONIS', 'Jules', 18),
(16, 'BERE', 'Antoine', 8),
(17, 'GAROT', 'Jonathan', 14),
(18, 'GHIOT', 'Romain', 3);

-----

--
-- Structure de la table `ville`
--

CREATE TABLE `ville` (
  `codePostal` int(5) NOT NULL,
  `nom` varchar(20) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Contenu de la table `ville`
--

INSERT INTO `ville` (`codePostal`, `nom`) VALUES
(7500, 'TOURNAI'),
(7502, 'TOURNAI'),
(7522, 'TOURNAI'),
(7531, 'TOURNAI'),
(7730, 'ESTAIMPUIS'),
(59000, 'ROUBAIX'),
(59200, 'TOURCOING'),

```

```

(59830, 'WANNEHAIN');

-- -----

--
-- Structure de la vue `db`
--
DROP TABLE IF EXISTS `db`;

CREATE ALGORITHM=UNDEFINED DEFINER=`jean`@`localhost` SQL SECURITY DEFINER
VIEW `db` AS select `arbres`.`espece` AS `espece`,`vendeur`.`nom` AS
`nom`,`vendeur`.`prenom` AS `prenom`,`adresse`.`rue` AS
`rue`,`adresse`.`numero` AS `numero`,`ville`.`codePostal` AS
`codePostal`,`ville`.`nom` AS `ville`,`pays`.`code` AS `code`,`pays`.`nom`
AS `pays`,`stock`.`prix` AS `prix`,`stock`.`stock` AS `stock` from
(((((`arbres` join `stock` on((`id` = `stock`.`espece`))) join `vendeur`
on((`vendeur`.`id` = `stock`.`vendeur`))) join `adresse`
on((`vendeur`.`adresse` = `adresse`.`id`))) join `ville`
on((`ville`.`codePostal` = `adresse`.`codePostal`))) join `pays`
on((`adresse`.`codePays` = `pays`.`code`))) order by
`espece`,`stock`.`prix` ;

--
-- Index pour les tables exportées
--

--
-- Index pour la table `adresse`
--
ALTER TABLE `adresse`
  ADD PRIMARY KEY (`id`),
  ADD KEY `pays` (`codePays`),
  ADD KEY `ville` (`codePostal`);

--
-- Index pour la table `arbres`
--
ALTER TABLE `arbres`
  ADD PRIMARY KEY (`id`);

--
-- Index pour la table `pays`
--
ALTER TABLE `pays`
  ADD PRIMARY KEY (`code`);

--
-- Index pour la table `stock`
--
ALTER TABLE `stock`
  ADD PRIMARY KEY (`espece`,`vendeur`),
  ADD KEY `vendeur` (`vendeur`);

--

```

```
-- Index pour la table `vendeur`
--
ALTER TABLE `vendeur`
  ADD PRIMARY KEY (`id`),
  ADD KEY `adresse` (`adresse`);

--
-- Index pour la table `ville`
--
ALTER TABLE `ville`
  ADD PRIMARY KEY (`codePostal`);

--
-- AUTO_INCREMENT pour les tables exportées
--

--
-- AUTO_INCREMENT pour la table `adresse`
--
ALTER TABLE `adresse`
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=21;
--
-- AUTO_INCREMENT pour la table `arbres`
--
ALTER TABLE `arbres`
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=7;
--
-- AUTO_INCREMENT pour la table `vendeur`
--
ALTER TABLE `vendeur`
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=19;
--
-- Contraintes pour les tables exportées
--

--
-- Contraintes pour la table `adresse`
--
ALTER TABLE `adresse`
  ADD CONSTRAINT `pays` FOREIGN KEY (`codePays`) REFERENCES `pays`
(`code`),
  ADD CONSTRAINT `ville` FOREIGN KEY (`codePostal`) REFERENCES `ville`
(`codePostal`);

--
-- Contraintes pour la table `stock`
--
ALTER TABLE `stock`
  ADD CONSTRAINT `espece` FOREIGN KEY (`espece`) REFERENCES `arbres`
(`id`),
  ADD CONSTRAINT `vendeur` FOREIGN KEY (`vendeur`) REFERENCES `vendeur`
(`id`);

--
```



```
-- Contraintes pour la table `vendeur`  
--  
ALTER TABLE `vendeur`  
  ADD CONSTRAINT `adresse` FOREIGN KEY (`adresse`) REFERENCES `adresse`  
  (`id`);  
  
/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;  
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;  
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
```