

# TP\_Modelisation

June 10, 2020

```
In [1]: %reload_ext sql
        %config SqlMagic.displaycon = False
        %config SqlMagic.autolimit = 100
```

```
In [2]: %sql sqlite:///bon-vivant8.db
```

```
In [3]: %%sql

        PRAGMA foreign_keys=1;
```

Done.

```
Out[3]: []
```

## 0.1 L'énoncé :

## 1 Premier Diagramme Entités/Associations réalisé avec Mocodo

### 1.1 Schéma SQL proposé par Mocodo

```
CREATE TABLE "APPELLATION" (
    "idappellation" VARCHAR(42),
    "type" VARCHAR(42),
    "nom" VARCHAR(42),
    PRIMARY KEY ("idappellation")
);
```

```
CREATE TABLE "DOMAINE" (
    "iddomaine" VARCHAR(42),
    "nom" VARCHAR(42),
    PRIMARY KEY ("iddomaine")
);
```

```
CREATE TABLE "STOCKAGE" (
    "idlotbouteille" VARCHAR(42),
    "idsite" VARCHAR(42),
    "nombre" VARCHAR(42),
    "idlotbouteille_1" VARCHAR(42),
```

M. BONVIVANT gère actuellement sa cave à vin avec une unique feuille de calcul dans un tableur. Un extrait de la feuille de calcul est donné en figure 1 : chaque ligne représente un vin dont M. BONVIVANT peut détenir plusieurs bouteilles. Les colonnes utilisées sont les suivantes :

**Où, H, Tot** La colonne **Où** indique sur quel site le vin est stocké. La colonne **Tot** indique le nombre *total* de bouteilles détenues. Le site de « Lyon » est particulier, car il y a deux lieux de stockage : la colonne **H** précise le nombre de bouteilles qui sont stockées sur le site « Haut », les autres bouteilles étant implicitement stockées sur le site « Bas ». M. BONVIVANT dispose d'autres lieux de stockage, dans la Drôme notamment.

**Couleur, Vol, Année, Degré** Ce sont des indications générales sur le type de vin, les valeurs autorisées pour **Couleur** sont « Blanc », « Rouge » et « Rosé ». **Vol** indique quel est la contenance en litres des bouteilles et **Année** précise le millésime du vin. **Degré** indique quel est le titre alcoolique du vin. Notons que le vin de Champagne n'est généralement pas millésimé.

**Région, Appellation** Ces colonnes précisent l'appellation viticole du vin. La colonne **Région** est une des grandes régions viticole de France. La colonne **Appellation** précise le vin dans sa région. Sans précision, les appellations sont des *Appellations d'Origine Contrôlée* (AOC). En revanche, les appellations *Vin De Pays* (VDP) et d'*Indication Géographique Protégée* (IGP) sont explicites.

**Domaine, Cuvée** Ces colonnes donnent des informations sur le producteur du vin : le nom du **Domaine** de production et le nom de la **Cuvée** au sein du domaine. Notons que des domaines différents peuvent tout à fait utiliser le même nom pour leur cuvée.

**Prix, Commentaire** Ces deux dernières colonnes indiquent le **Prix** d'achat en euros, quand il est connu, et donnent éventuellement des informations comme la date ou le motif d'achat, le cépage utilisé ou les recommandations du producteur quant à la consommation du vin.

### Exercice 1 : Normalisation de données tabulaires

1. En utilisant les attributs de la feuille de calcul, proposer un diagramme E/A modélisant ce cas d'étude.
2. Traduire ce diagramme en schéma de base de données relationnelle SQL. Préciser en particulier les clefs primaires, les clefs étrangères ainsi que les autres contraintes **UNIQUE** (et éventuellement celles **NOT NULL** ou **CHECK**).
3. Anticiper des problèmes de qualité des données que vous allez rencontrer lors du peuplement du schéma normalisé à partir du contenu de la feuille de calcul.
4. L'état actuel de la feuille de calcul est celui des bouteilles disponibles dans sa cave. M. BONVIVANT aimerait historiser les bouteilles qu'il a déjà dégustées. Proposer une modification pour cela.

énoncé

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Qv	H	Tol	Couleur	Vol	Région	Appellation	Année	Degré	Domaine	Cuvée	Prix	
2	Lyon	1	8	Blanc	0.75	Champagne	Champagne brut			Pierre Moncuit	Huques de Goulmet	16.75	salon vigneron indépendants 10/2012
3	Lyon	1	1	Blanc	0.75	Champagne	Champagne brut	12.00		Robert Allait			cadeau Annie et Rémi
4	Lyon	1	4	Blanc	0.75	Val-de-Loire	Méthode traditionnelle	12.00		Domaine Dardan	La Marionnerie	6.05	Cheverny (vacances été 2014)
5	Lyon	1	1	Rosé	0.75	Champagne	Champagne brut	2009	12	Henriot	Millesime 2009		
6	Lyon	2	2	Blanc	0.75	Monbazillac	Monbazillac	2009	12.50	de Grange neuve		8.26	salon des vigneron indépendants 03/2012
7	Lyon	4	4	Blanc	0.75	Anjou	Anjou	2009	14.50	Château Pierre-Bise	Le haut de la garde	9.50	salon des vigneron indépendants 03/2012, boire avant 2
8	Lyon	1	1	Blanc	0.375	Jurançon	Jurançon	2009	13.00	Château Joly	Terrae Escencia		cadeau panier baronne du Barry
9	Lyon	1	4	Blanc	0.75	Bourgogne	Pouilly Fuisé	2010	13.50	Marin		9.15	
10	Lyon	1	1	Blanc	0.75	Côtes-du-Rhône	Condrieu	2010	13.50	Jocelyne et Yves Lafay	Aux ruses	22.00	Marché aux vins d'Ampuis 2012
11	Lyon	1	1	Blanc	0.75	CDR	Condrieu	2010	15.00	Jean-Pierre Lézin		23.00	Marché aux vins d'Ampuis 2012
12	Lyon	2	2	Blanc	0.75	Val-de-Loire	Cote Cheverny	2010	13.50	Domaine Dardan	chênes	15.05	vacances été 2014
13	Lyon	2	2	Blanc	0.75	Côtes-du-Rhône	Rhodanien	2011	12.50	Pierre Finon	Mélessa	6.50	Roussanne
14	Lyon	3	3	Blanc	0.75	Côtes-du-Rhône	Rhodanien	2011	14.50	Richard	Vosnier	10.00	Marché aux vins de Chavanay 12/2012
15	Lyon	3	3	Blanc	0.75	Côtes-du-Rhône	Vosnier	2011	13.50	Christine David	Vosnier en fut de chêne	8.30	Marché aux vins de Chavanay 12/2012
16	Lyon	2	2	Blanc	0.75	Côtes-du-Rhône	Saint Joseph	2011	13.50	Coursodon	Silice	16.0	Marché aux vins d'Ampuis 2013
17	Lyon	4	4	Blanc	0.75	Beaujolais	Beaujolais levres	2011		Bruno Jeanteau		6.5	Marché de la gastronomie 2014
18	Lyon	1	6	Blanc	0.75	Bourgogne	Saint Vran	2011	13.50	Marin		7.00	
19	Lyon	3	3	Blanc	0.75	Alsace	Edelszwicker	2012	12.00	Gérard Metz		6.06	salon des vigneron indépendants 28/10/2013
20	Lyon	2	2	Blanc	0.75	CDR	Vin de pays des collines Rhodanien	2012	12.50	Chirat	NouvelR (chardonnay)	7.00	Marché aux vins de Chavanay 2013, Boire avant 2018
21	Lyon	3	3	Blanc	0.75	Bourgogne	Pouilly Fuisé	2012	13.50	Marin			
22	Lyon	1	1	Blanc	0.75	Val-de-Loire	Jasniers	2013	12.00	Les gauleteries, Lelais			
23	Lyon	1	1	Rosé	0.75	Bordeaux	Bordeaux Claret	2010	13.00	Château Gennesquin			
24	Lyon	1	1	Rosé	0.75	Provence	Bandol	2011	13.50	de Souviou			
25	Lyon	1	1	Rouge	0.75	Bordeaux	Médoc	1998	12.50	Grand Saint-Brice			
26	Lyon	1	1	Rouge	0.75	Bordeaux	Côte de Bourg	2008	12.00	Haut Mévrat	Tradition		
27	Lyon	1	1	Rouge	0.75	Beaujolais	Village	2006	14.50	Lapierre			
28	Lyon	3	3	Rouge	0.75	Beaujolais	Pécharmant	2006	14.00	Château La Renaudie	Tradition	7.50	Salon vigneron indépendants 28/10/2013
29	Lyon	2	2	Rouge	0.75	Beaujolais	Côte de Brouilly	2010	12.50	Patricia et Bernard Jomain	des héritiers	7.35	
30	Lyon	2	2	Rouge	0.75	CDR	Côtes-du-rhône village	2010	14.50	Sablé	Les lauriers du terroir		

énoncé

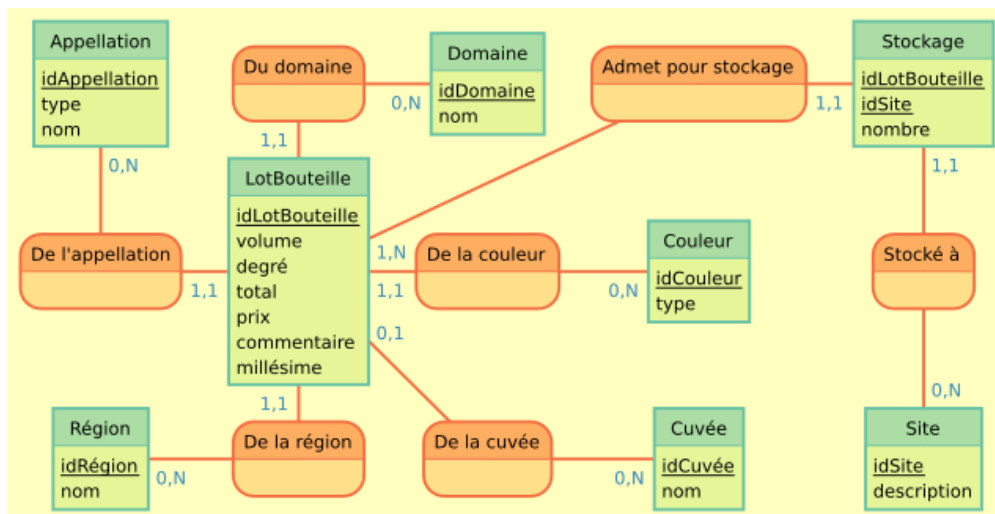


diagramme E/A

```

    "idsite_1" VARCHAR(42),
    PRIMARY KEY ("idlotbouteille", "idsite"),
    FOREIGN KEY ("idlotbouteille_1") REFERENCES "LOTBOUTEILLE" ("idlotbouteille"),
    FOREIGN KEY ("idsite_1") REFERENCES "SITE" ("idsite")
);

CREATE TABLE "LOTBOUTEILLE" (
    "idlotbouteille" VARCHAR(42),
    "volume" VARCHAR(42),
    "degré" VARCHAR(42),
    "total" VARCHAR(42),
    "prix" VARCHAR(42),
    "commentaire" VARCHAR(42),
    "millésime" VARCHAR(42),
    "idrégion" VARCHAR(42),
    "idappellation" VARCHAR(42),
    "idcuvée" VARCHAR(42),
    "iddomaine" VARCHAR(42),
    "idcouleur" VARCHAR(42),
    PRIMARY KEY ("idlotbouteille"),
    FOREIGN KEY ("idrégion") REFERENCES "RÉGION" ("idrégion"),
    FOREIGN KEY ("idappellation") REFERENCES "APPELLATION" ("idappellation"),
    FOREIGN KEY ("idcuvée") REFERENCES "CUVÉE" ("idcuvée"),
    FOREIGN KEY ("iddomaine") REFERENCES "DOMAINE" ("iddomaine"),
    FOREIGN KEY ("idcouleur") REFERENCES "COULEUR" ("idcouleur")
);

CREATE TABLE "COULEUR" (
    "idcouleur" VARCHAR(42),
    "type" VARCHAR(42),
    PRIMARY KEY ("idcouleur")
);

CREATE TABLE "RÉGION" (
    "idrégion" VARCHAR(42),
    "nom" VARCHAR(42),
    PRIMARY KEY ("idrégion")
);

CREATE TABLE "CUVÉE" (
    "idcuvée" VARCHAR(42),
    "nom" VARCHAR(42),
    PRIMARY KEY ("idcuvée")
);

CREATE TABLE "SITE" (
    "idsite" VARCHAR(42),
    "description" VARCHAR(42),

```

```
PRIMARY KEY ("idsite")
);
```

## 1.2 Description du schéma SQL

Appellation ( idAppellation, type, nom )

Le champ idAppellation constitue la clef primaire de la table. C'était déjà un identifiant de l'entité Appellation.

Les champs type et nom étaient déjà de simples attributs de l'entité Appellation.

Domaine ( idDomaine, nom )

Le champ idDomaine constitue la clef primaire de la table. C'était déjà un identifiant de l'entité Domaine.

Le champ nom était déjà un simple attribut de l'entité Domaine.

Stockage ( idLotBouteille, idSite, nombre, idLotBouteille.1, idSite.1 )

Les champs idLotBouteille et idSite constituent la clef primaire de la table. C'était déjà des identifiants de l'entité Stockage.

Le champ nombre était déjà un simple attribut de l'entité Stockage.

Le champ idLotBouteille.1 est une clef étrangère. Il a migré à partir de l'entité LotBouteille par l'association de dépendance fonctionnelle Admet pour stockage en perdant son caractère identifiant.

Le champ idSite.1 est une clef étrangère. Il a migré à partir de l'entité Site par l'association de dépendance fonctionnelle Stocké à en perdant son caractère identifiant.

LotBouteille ( idLotBouteille, volume, degré, total, prix, commentaire, millésime, idRégion, idAppellation, idCuvée, idDomaine, idCouleur )

Le champ idLotBouteille constitue la clef primaire de la table. C'était déjà un identifiant de l'entité LotBouteille.

Les champs volume, degré, total, prix, commentaire et millésime étaient déjà de simples attributs de l'entité LotBouteille.

Le champ idRégion est une clef étrangère. Il a migré à partir de l'entité Région par l'association de dépendance fonctionnelle De la région en perdant son caractère identifiant.

Le champ idAppellation est une clef étrangère. Il a migré à partir de l'entité Appellation par l'association de dépendance fonctionnelle De l'appellation en perdant son caractère identifiant.

Le champ idCuvée est une clef étrangère. Il a migré à partir de l'entité Cuvée par l'association de dépendance fonctionnelle De la cuvée en perdant son caractère identifiant.

Le champ idDomaine est une clef étrangère. Il a migré à partir de l'entité Domaine par l'association de dépendance fonctionnelle Du domaine en perdant son caractère identifiant.

Le champ idCouleur est une clef étrangère. Il a migré à partir de l'entité Couleur par l'association de dépendance fonctionnelle De la couleur en perdant son caractère identifiant.

Couleur ( idCouleur, type )

Le champ idCouleur constitue la clef primaire de la table. C'était déjà un identifiant de l'entité Couleur.

Le champ type était déjà un simple attribut de l'entité Couleur.

Région ( idRégion, nom )

Le champ idRégion constitue la clef primaire de la table. C'était déjà un identifiant de l'entité Région.

Le champ nom était déjà un simple attribut de l'entité Région.

Cuvée ( idCuvée, nom )

Le champ idCuvée constitue la clef primaire de la table. C'était déjà un identifiant de l'entité Cuvée.

Le champ nom était déjà un simple attribut de l'entité Cuvée.

Site ( idSite, description )

Le champ idSite constitue la clef primaire de la table. C'était déjà un identifiant de l'entité Site.

Le champ description était déjà un simple attribut de l'entité Site.

### 1.3 Affinement du schéma SQL (types et contraintes)

```
CREATE TABLE "APPELLATION" (  
  "idappellation" INTEGER NOT NULL PRIMARY KEY,  
  "type" VARCHAR(42) CHECK( "type" IN ("AOC", "VDP", "IGP")), --NULL représenterait AOC  
  "nom" VARCHAR(42) NOT NULL  
);
```

```
CREATE TABLE "DOMAINE" (  
  "iddomaine" INTEGER NOT NULL PRIMARY KEY,  
  "nom" VARCHAR(42) NOT NULL UNIQUE -- un nom de domaine est unique  
);
```

```
CREATE TABLE "STOCKAGE" (  
  "idlotbouteille" INTEGER NOT NULL  
  "idsite" INTEGER NOT NULL,  
  "nombre" INTEGER NOT NULL,  
  PRIMARY KEY ("idlotbouteille", "idsite"),  
  FOREIGN KEY ("idlotbouteille") REFERENCES "LOTBOUTEILLE" ("idlotbouteille"),  
  FOREIGN KEY ("idsite") REFERENCES "SITE" ("idsite")  
);
```

```
CREATE TABLE "LOTBOUTEILLE" (  
  "idlotbouteille" INTEGER NOT NULL PRIMARY KEY,  
  "volume" REAL NOT NULL CHECK(volume >= 0),  
  "degré" REAL NOT NULL CHECK(degré >= 0),  
  "total" INTEGER NOT NULL CHECK(total >= 0),  
  "prix" REAL NOT NULL CHECK(prix >= 0),  
  "commentaire" VARCHAR(42),  
  "millésime" INTEGER CHECK(millésime >= 0),  
  "idrégion" INTEGER NOT NULL,  
  "idappellation" INTEGER NOT NULL,  
  "idcuvée" INTEGER NOT NULL,  
  "iddomaine" INTEGER NOT NULL,  
  "idcouleur" INTEGER NOT NULL,  
  PRIMARY KEY ("idlotbouteille"),  
  FOREIGN KEY ("idrégion") REFERENCES "RÉGION" ("idrégion"),  
  FOREIGN KEY ("idappellation") REFERENCES "APPELLATION" ("idappellation"),  
  FOREIGN KEY ("idcuvée") REFERENCES "CUVÉE" ("idcuvée"),  
  FOREIGN KEY ("iddomaine") REFERENCES "DOMAINE" ("iddomaine"),  
  FOREIGN KEY ("idcouleur") REFERENCES "COULEUR" ("idcouleur")
```

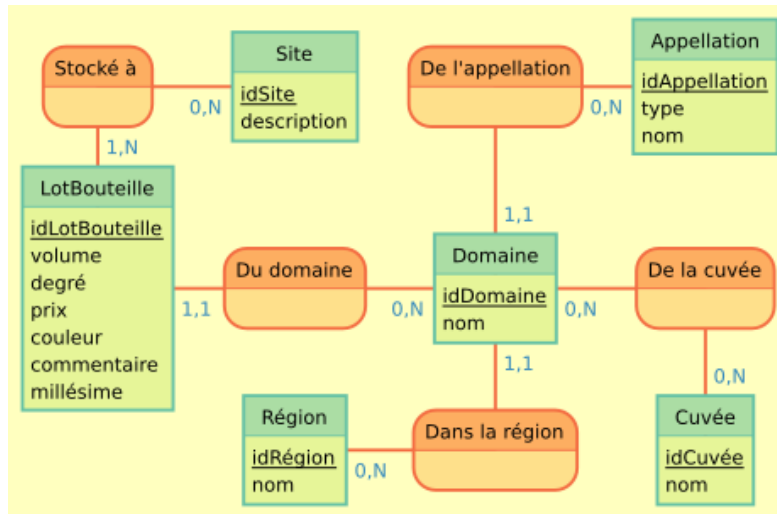


diagramme E/A

);

```

CREATE TABLE "COULEUR" (
  "idcouleur" INTEGER NOT NULL PRIMARY KEY,
  "type" VARCHAR(42) NOT NULL UNIQUE
);
  
```

```

CREATE TABLE "RÉGION" (
  "idrégion" INTEGER NOT NULL PRIMARY KEY,
  "nom" NOT NULL UNIQUE
);
  
```

```

CREATE TABLE "CUVÉE" (
  "idcuvée" INTEGER NOT NULL PRIMARY KEY,
  "nom" VARCHAR(42) NOT NULL UNIQUE
);
  
```

*--un nom de cuvée est unique*

```

CREATE TABLE "SITE" (
  "idsite" INTEGER NOT NULL PRIMARY KEY,
  "description" VARCHAR(42) NOT NULL UNIQUE
);
  
```

## 1.4 Second Diagramme Entités/Associations réalisé avec Mocodo

Stocké à ( idLotBouteille, idSite )

Le champ idLotBouteille fait partie de la clef primaire de la table. C'est une clef étrangère qui a migré directement à partir de l'entité LotBouteille.

Le champ idSite fait partie de la clef primaire de la table. C'est une clef étrangère qui a migré directement à partir de l'entité Site.

Site ( idSite, description )

Le champ idSite constitue la clef primaire de la table. C'était déjà un identifiant de l'entité Site.  
Le champ description était déjà un simple attribut de l'entité Site.

Appellation ( idAppellation, type, nom )

Le champ idAppellation constitue la clef primaire de la table. C'était déjà un identifiant de l'entité Appellation.

Les champs type et nom étaient déjà de simples attributs de l'entité Appellation.

LotBouteille ( idLotBouteille, volume, degré, prix, couleur, commentaire, millésime, idDomaine )

Le champ idLotBouteille constitue la clef primaire de la table. C'était déjà un identifiant de l'entité LotBouteille.

Les champs volume, degré, prix, couleur, commentaire et millésime étaient déjà de simples attributs de l'entité LotBouteille.

Le champ idDomaine est une clef étrangère. Il a migré à partir de l'entité Domaine par l'association de dépendance fonctionnelle Du domaine en perdant son caractère identifiant.

Domaine ( idDomaine, nom, idAppellation, idRégion )

Le champ idDomaine constitue la clef primaire de la table. C'était déjà un identifiant de l'entité Domaine.

Le champ nom était déjà un simple attribut de l'entité Domaine.

Le champ idAppellation est une clef étrangère. Il a migré à partir de l'entité Appellation par l'association de dépendance fonctionnelle De l'appellation en perdant son caractère identifiant.

Le champ idRégion est une clef étrangère. Il a migré à partir de l'entité Région par l'association de dépendance fonctionnelle Dans la région en perdant son caractère identifiant.

De la cuvée ( idDomaine, idCuvée )

Le champ idDomaine fait partie de la clef primaire de la table. C'est une clef étrangère qui a migré directement à partir de l'entité Domaine.

Le champ idCuvée fait partie de la clef primaire de la table. C'est une clef étrangère qui a migré directement à partir de l'entité Cuvée.

Région ( idRégion, nom )

Le champ idRégion constitue la clef primaire de la table. C'était déjà un identifiant de l'entité Région.

Le champ nom était déjà un simple attribut de l'entité Région.

Cuvée ( idCuvée, nom )

Le champ idCuvée constitue la clef primaire de la table. C'était déjà un identifiant de l'entité Cuvée.

Le champ nom était déjà un simple attribut de l'entité Cuvée.

## 2 Schéma SQL avec contraintes

```
CREATE TABLE "APPELLATION" (  
  "idappellation" INTEGER NOT NULL PRIMARY KEY,  
  "type" VARCHAR(42) CHECK( "type" IN ("AOC", "VDP", "IGP")), --NULL représenterait AOC  
  "nom" VARCHAR(42) NOT NULL  
);
```



```

CREATE TABLE "STOCKAGE" (
  "idlotbouteille" INTEGER NOT NULL,
  "idsite" INTEGER NOT NULL,
  "nombre" INTEGER NOT NULL CHECK(nombre >= 0),
  PRIMARY KEY ("idlotbouteille", "idsite"),
  FOREIGN KEY ("idlotbouteille") REFERENCES "LOTBOUTEILLE" ("idlotbouteille"),
  FOREIGN KEY ("idsite") REFERENCES "SITE" ("idsite")
);

```

```

CREATE TABLE "DOMAINE" (
  "iddomaine" INTEGER NOT NULL PRIMARY KEY,
  "nom" VARCHAR(42) NOT NULL,          -- un nom de domaine n'est pas forcément unique (des do
  "idrégion" INTEGER NOT NULL,
  "idcuvée" INTEGER NOT NULL,
  "idappellation" INTEGER NOT NULL,
  FOREIGN KEY ("idappellation") REFERENCES "APPELLATION" ("idappellation"),
  FOREIGN KEY ("idcuvée") REFERENCES "CUVÉE" ("idcuvée"),
  FOREIGN KEY ("idrégion") REFERENCES "RÉGION" ("idrégion")
);

```

```

CREATE TABLE "LOTBOUTEILLE" (
  "idlotbouteille" INTEGER NOT NULL PRIMARY KEY,
  "volume" REAL NOT NULL CHECK(volume >= 0),
  "degré" REAL NOT NULL CHECK(degré >= 0),
  "prix" REAL NOT NULL CHECK(prix >= 0),
  "commentaire" VARCHAR(42), --peut être NULL
  "millésime" INTEGER CHECK(millésime >= 0),
  "couleur" VARCHAR(42) NOT NULL CHECK(lower(couleur) in ("blanc", "rosé", "rouge")),
  "iddomaine" INTEGER NOT NULL,
  FOREIGN KEY ("iddomaine") REFERENCES "DOMAINE" ("iddomaine")
);

```

```

CREATE TABLE "RÉGION" (
  "idrégion" INTEGER NOT NULL PRIMARY KEY,
  "nom" NOT NULL UNIQUE
);

```

```

CREATE TABLE "CUVÉE" (
  "idcuvée" INTEGER NOT NULL PRIMARY KEY,
  "nom" VARCHAR(42) UNIQUE          --un nom de cuvée est unique mais peut être vide
);

```

```

CREATE TABLE "SITE" (

```

```

    "idsite" INTEGER NOT NULL PRIMARY KEY,
    "description" VARCHAR(42) NOT NULL UNIQUE
);

```

```

CREATE TABLE "DE_LA_CUVÉE" (
    "iddomaine" VARCHAR(42),
    "idcuvée" VARCHAR(42),
    PRIMARY KEY ("iddomaine", "idcuvée"),
    FOREIGN KEY ("iddomaine") REFERENCES "DOMAINE" ("iddomaine"),
    FOREIGN KEY ("idcuvée") REFERENCES "CUVÉE" ("idcuvée")
);

```

In [29]: %%sql

```

CREATE TABLE "APPELLATION" (
    "idappellation" INTEGER NOT NULL PRIMARY KEY,
    "type" VARCHAR(42) CHECK( "type" IN ("AOC", "VDP", "IGP")), --NULL représenterait A
    "nom" VARCHAR(42) NOT NULL
);

```

```

CREATE TABLE "STOCKAGE" (
    "idlotbouteille" INTEGER NOT NULL,
    "idsite" INTEGER NOT NULL,
    "nombre" INTEGER NOT NULL CHECK(nombre >= 0),
    PRIMARY KEY ("idlotbouteille", "idsite"),
    FOREIGN KEY ("idlotbouteille") REFERENCES "LOTBOUTEILLE" ("idlotbouteille"),
    FOREIGN KEY ("idsite") REFERENCES "SITE" ("idsite")
);

```

```

CREATE TABLE "DOMAINE" (
    "iddomaine" INTEGER NOT NULL PRIMARY KEY,
    "nom" VARCHAR(42) NOT NULL,          -- un nom de domaine n'est pas forcément unique
    "idrégion" INTEGER NOT NULL,
    "idcuvée" INTEGER NOT NULL,
    "idappellation" INTEGER NOT NULL,
    FOREIGN KEY ("idappellation") REFERENCES "APPELLATION" ("idappellation"),
    FOREIGN KEY ("idcuvée") REFERENCES "CUVÉE" ("idcuvée"),
    FOREIGN KEY ("idrégion") REFERENCES "RÉGION" ("idrégion")
);

```

```

CREATE TABLE "LOTBOUTEILLE" (

```

```

        "idlotbouteille" INTEGER NOT NULL PRIMARY KEY,
        "volume" REAL NOT NULL CHECK(volume >= 0),
        "degré" REAL CHECK(degré >= 0),
        "prix" REAL CHECK(prix >= 0),
        "commentaire" TEXT, --peut être NULL
        "millésime" INTEGER CHECK(millésime >= 0),
        "couleur" VARCHAR(5) NOT NULL CHECK(lower(couleur) in ("blanc", "rosé", "rouge")),
        "iddomaine" INTEGER NOT NULL,
        FOREIGN KEY ("iddomaine") REFERENCES "DOMAINE" ("iddomaine")
    );

CREATE TABLE "RÉGION" (
    "idrégion" INTEGER NOT NULL PRIMARY KEY,
    "nom" NOT NULL UNIQUE
);

CREATE TABLE "CUVÉE" (
    "idcuvée" INTEGER NOT NULL PRIMARY KEY,
    "nom" VARCHAR(42) UNIQUE --un nom de cuvée est unique mais peut être NULL
);

CREATE TABLE "SITE" (
    "idsite" INTEGER NOT NULL PRIMARY KEY,
    "description" VARCHAR(42) NOT NULL UNIQUE
);

```

Done.  
 Done.  
 Done.  
 Done.  
 Done.  
 Done.  
 Done.

Out[29]: []

## 2.1 Problèmes de qualité des données que nous risquons de rencontrer lors du peuplement de la table

- Le deuxième schéma propose un meilleur découpage des données entre stockage, lot de bouteille et terroir (domaine, région, appellation)
- Un même lot de bouteilles peut être stocké dans plusieurs sites grâce à la table stockage : on distinguera par exemple dans la table site les deux sites de stockage sur Lyon. On a supprimé l'attribut total pour un lot de bouteilles, on peut le retrouver en faisant la somme des nombres de bouteilles de tous les stockages associés à un lot.
- avec ce modèle on ne peut pas distinguer des achats du même millés

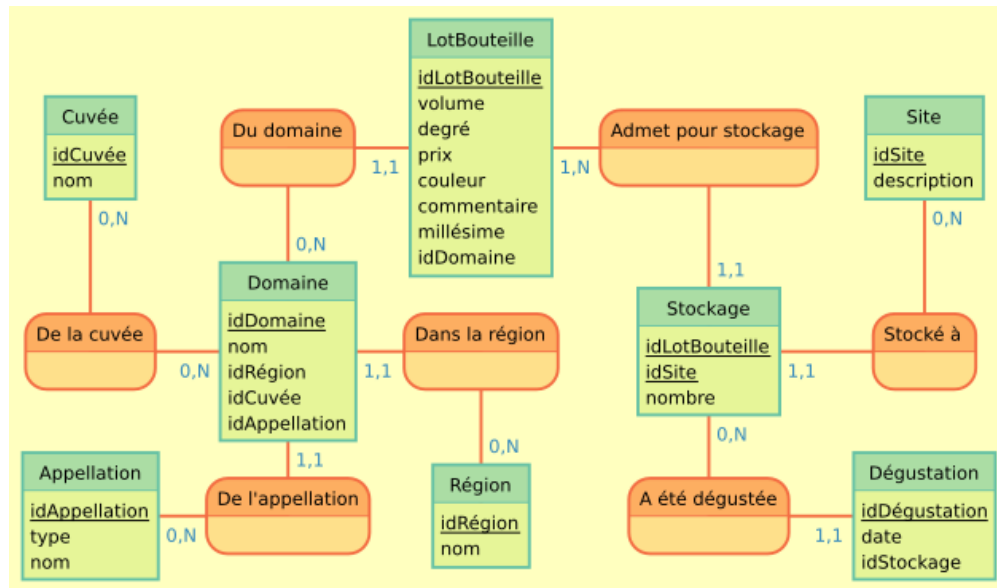


diagramme E/A

- Les orthographes de certains attributs (régions, appellations) sont variables
- Les formats des titrages en degrés sont variables
- 

## 2.2 Rajout d'un table Dégustation pour historiser les dégustations

Site ( idSite, description )

Le champ idSite constitue la clef primaire de la table. C'était déjà un identifiant de l'entité Site.

Le champ description était déjà un simple attribut de l'entité Site.

Région ( idRégion, nom )

Le champ idRégion constitue la clef primaire de la table. C'était déjà un identifiant de l'entité Région.

Le champ nom était déjà un simple attribut de l'entité Région.

Cuvée ( idCuvée, nom )

Le champ idCuvée constitue la clef primaire de la table. C'était déjà un identifiant de l'entité Cuvée.

Le champ nom était déjà un simple attribut de l'entité Cuvée.

Stockage ( idStockage, idLotBouteille, idSite, nombre, idLotBouteille.1, idSite.1 )

Les champs idStockage, idLotBouteille et idSite constituent la clef primaire de la table. C'était déjà des identifiants de l'entité Stockage.

Le champ nombre était déjà un simple attribut de l'entité Stockage.

Le champ idLotBouteille.1 est une clef étrangère. Il a migré à partir de l'entité LotBouteille par l'association de dépendance fonctionnelle Admet pour stockage en perdant son caractère identifiant.

Le champ idSite.1 est une clef étrangère. Il a migré à partir de l'entité Site par l'association de dépendance fonctionnelle Stocké à en perdant son caractère identifiant.

Domaine ( idDomaine, nom, idRégion, idCuvée, idAppellation, idAppellation.1, idCuvée.1, idRégion.1 )

Le champ idDomaine constitue la clef primaire de la table. C'était déjà un identifiant de l'entité Domaine.

Les champs nom, idRégion, idCuvée et idAppellation étaient déjà de simples attributs de l'entité Domaine.

Le champ idAppellation.1 est une clef étrangère. Il a migré à partir de l'entité Appellation par l'association de dépendance fonctionnelle De l'appellation en perdant son caractère identifiant.

Le champ idCuvée.1 est une clef étrangère. Il a migré à partir de l'entité Cuvée par l'association de dépendance fonctionnelle De la cuvée en perdant son caractère identifiant.

Le champ idRégion.1 est une clef étrangère. Il a migré à partir de l'entité Région par l'association de dépendance fonctionnelle Dans la région en perdant son caractère identifiant.

Dégustation ( idDégustation, date, idStockage, idStockage.1, idLotBouteille, idSite )

Le champ idDégustation constitue la clef primaire de la table. C'était déjà un identifiant de l'entité Dégustation.

Les champs date et idStockage étaient déjà de simples attributs de l'entité Dégustation.

Le champ idStockage.1 est une clef étrangère. Il a migré à partir de l'entité Stockage par l'association de dépendance fonctionnelle A été dégustée en perdant son caractère identifiant.

Le champ idLotBouteille est une clef étrangère. Il a migré à partir de l'entité Stockage par l'association de dépendance fonctionnelle A été dégustée en perdant son caractère identifiant.

Le champ idSite est une clef étrangère. Il a migré à partir de l'entité Stockage par l'association de dépendance fonctionnelle A été dégustée en perdant son caractère identifiant.

LotBouteille ( idLotBouteille, volume, degré, prix, couleur, commentaire, millésime, idDomaine, idDomaine.1 )

Le champ idLotBouteille constitue la clef primaire de la table. C'était déjà un identifiant de l'entité LotBouteille.

Les champs volume, degré, prix, couleur, commentaire, millésime et idDomaine étaient déjà de simples attributs de l'entité LotBouteille.

Le champ idDomaine.1 est une clef étrangère. Il a migré à partir de l'entité Domaine par l'association de dépendance fonctionnelle Du domaine en perdant son caractère identifiant.

Appellation ( idAppellation, type, nom )

Le champ idAppellation constitue la clef primaire de la table. C'était déjà un identifiant de l'entité Appellation.

Les champs type et nom étaient déjà de simples attributs de l'entité Appellation.

### Schéma SQL

```
CREATE TABLE "RÉGION" (  
  "idrégion" INTEGER NOT NULL PRIMARY KEY,  
  "nom" NOT NULL UNIQUE  
);
```

```
CREATE TABLE "CUVÉE" (  
  "idcuvée" INTEGER NOT NULL PRIMARY KEY,  
  "nom" VARCHAR(42) UNIQUE  
);
```

*--un nom de cuvée est unique mais peut être vide*

```
CREATE TABLE "SITE" (  
  "idsite" INTEGER NOT NULL PRIMARY KEY,  
  "description" VARCHAR(42) NOT NULL UNIQUE  
);
```

```

CREATE TABLE "STOCKAGE" (
    "idstockage" INTEGER NOT NULL PRIMARY KEY,
    "idlotbouteille" INTEGER NOT NULL,
    "idsite" INTEGER NOT NULL,
    "nombre" INTEGER NOT NULL,
    -- PRIMARY KEY ("idstockage", "idlotbouteille"),
    FOREIGN KEY ("idlotbouteille") REFERENCES "LOTBOUTEILLE" ("idlotbouteille"),
    FOREIGN KEY ("idsite") REFERENCES "SITE" ("idsite")
);

```

```

CREATE TABLE "DOMAINE" (
    "iddomaine" INTEGER NOT NULL PRIMARY KEY,
    "nom" VARCHAR(42) NOT NULL,          -- un nom de domaine n'est pas forcément unique (des do
    "idrégion" INTEGER NOT NULL,
    "idcuvée" INTEGER NOT NULL,
    "idappellation" INTEGER NOT NULL,
    FOREIGN KEY ("idappellation") REFERENCES "APPELLATION" ("idappellation"),
    FOREIGN KEY ("idcuvée") REFERENCES "CUVÉE" ("idcuvée"),
    FOREIGN KEY ("idrégion") REFERENCES "RÉGION" ("idrégion")
);

```

```

CREATE TABLE "DÉGUSTATION" (
    "iddégustation" INTEGER NOT NULL PRIMARY KEY,
    "date" VARCHAR(42),
    "idstockage" INTEGER NOT NULL,
    FOREIGN KEY ("idstockage") REFERENCES "STOCKAGE" ("idstockage")
);

```

```

CREATE TABLE "LOTBOUTEILLE" (
    "idlotbouteille" INTEGER NOT NULL PRIMARY KEY,
    "volume" REAL NOT NULL CHECK(volume >= 0),
    "degré" REAL CHECK(degré >= 0),
    "prix" REAL CHECK(prix >= 0),
    "commentaire" VARCHAR(42), --peut être NULL
    "millésime" INTEGER CHECK(millésime >= 0),
    "couleur" VARCHAR(42) NOT NULL CHECK(lower(couleur) in ("blanc", "rosé", "rouge")),
    "iddomaine" INTEGER NOT NULL,
    FOREIGN KEY ("iddomaine") REFERENCES "DOMAINE" ("iddomaine")
);

```

```

CREATE TABLE "APPELLATION" (
    "idappellation" INTEGER NOT NULL PRIMARY KEY,

```

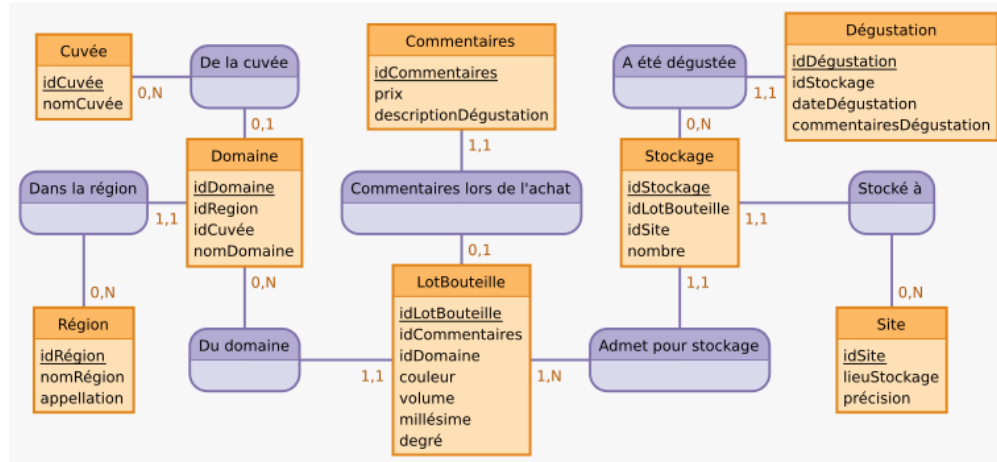


diagramme Brigitte

```
"type" VARCHAR(42) CHECK( "type" IN ("AOC", "VDP", "IGP")), --NULL représenterait AOC
"nom" VARCHAR(42) NOT NULL
);
```

## 2.3 Version de Brigitte

Description du schéma SQL :

```
PRAGMA foreign_keys=1;
```

```
DROP TABLE if EXISTS Site;
DROP TABLE if EXISTS Stockage;
DROP TABLE if EXISTS Dégustation;
DROP TABLE if EXISTS LotBouteille;
DROP TABLE if EXISTS Région;
DROP TABLE if EXISTS Domaine;
DROP TABLE if EXISTS Cuvée;
DROP TABLE if EXISTS Commentaires;
```

```
CREATE TABLE Site(
    idSite INTEGER NOT NULL PRIMARY KEY,
    lieuStockage TXT,
    précision Stockage VARCHAR(1) CHECK (précision IN ('H','B'))
);
```

```
CREATE TABLE Stockage (
    idStockage INTEGER NOT NULL PRIMARY KEY,
    idLotBouteille INTEGER NOT NULL REFERENCES LotBouteille (idLotBouteille),
    idSite INTEGER NOT NULL REFERENCES SITE (idSite),
    nombre INTEGER NOT NULL -- faire une somme sur différents sites contenant même bouteille.
);
```

```

CREATE TABLE LotBouteille(
    idLotBouteille INTEGER NOT NULL PRIMARY KEY,
    idCommentaires INTEGER NOT NULL REFERENCES Commentaires(idCommentaires),
    idDomaine INTEGER NOT NULL REFERENCES Domaine(idDomaine),
    couleur VARCHAR(5) CHECK(LOWER(couleur) IN ('rouge','rose','blanc')),
    volume REAL NOT NULL CHECK(volume >= 0),
    millésime INTEGER CHECK(millésime >= 0),
    degré REAL CHECK(degré >= 0) -- Attention il faudra nettoyer % ¯
);

CREATE TABLE Dégustation (
    idDégustation INTEGER NOT NULL PRIMARY KEY,
    idStockage INTEGER NOT NULL REFERENCES Stockage(idStockage),
    dateDégustation TXT,
    commentairesDégustation TXT
);

CREATE TABLE Région (
    idRégion INTEGER NOT NULL PRIMARY KEY,
    nomRégion TXT NOT NULL UNIQUE,
    appellation TXT
);

CREATE TABLE Domaine (
    idDomaine INTEGER NOT NULL PRIMARY KEY,
    idRegion INTEGER NOT NULL REFERENCES Région(idRégion),
    idCuvée INTEGER NOT NULL REFERENCES Cuvée(idCuvée),
    nomDomaine TXT -- un nom de domaine n'est pas forcément unique
);

CREATE TABLE Cuvée (
    idCuvée INTEGER NOT NULL PRIMARY KEY,
    nomCuvée TXT UNIQUE --un nom de cuvée est unique mais peut être vide
);

CREATE TABLE Commentaires (
    idCommentaires INTEGER NOT NULL PRIMARY KEY,
    prix REAL CHECK(prix >= 0),
    descriptionDégustation TXT
);

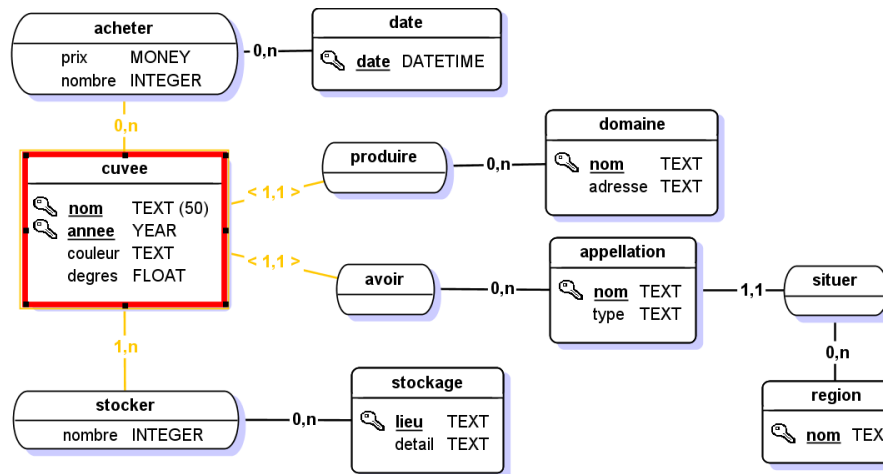
```

## 2.4 Correction

**ATTENTION : la proposition suivante est une base de discussion, pas une correction absolue (s'il en existe une).**

Ce qui est peut-être le plus important sur cet exemple ou sur votre proposition c'est d'être capable de comprendre et critiquer le modèle, de déterminer les impacts de tel ou tel choix de modélisation. Concrètement, cela signifie être capable de répondre à des questions comme : un domaine peut-il produire plusieurs





schéma

vins ? un vin peut t'il être acheté plusieurs fois à des prix différents ? peut-on stocker un même vin à différent endroits ? etc.

Des choix faits dans cette proposition peuvent être discutés ad nauseam et pour mettre fin aux discussions et avancer il faudra trancher en accord avec le client. Une question d'apparence simple comme "qu'est ce qui permet d'identifier un vin" n'est en fait pas évidente, donc que le client nous le précise !

Voir le [diagramme E/A JMerise](#) de la proposition, [sa screenshot png](#), et [sa contrepartie SQL](#). Le SQL a été généré par JMerise et légèrement repris à la main.

Quelques remarques :

- la région bénéficie de sa propre *entité* mais *couleur* n'est qu'un attribut (avec une contrainte CHECK dans le schéma SQL). En effet, on peut imaginer qu'on ait plus de choses à dire sur une région viticole (e.g., sa définition géographique) que sur la couleur du vin
- le type de l'appellation (IGP, VdP, AOP) n'est souvent pas explicitée (e.g., "Collines rhodaniennes" est une IGP, "Condrieu" une AOP), il faudra compléter la base.

Le plus discutable est le choix **d'une cuvée millésimée** pour représenter "une bouteille", "un vin". Ici, la *cuvée* est une *entité faible* (dite aussi relative), c'est-à-dire que le *domaine* et l'*appellation* font partie de la clef de la *cuvée* qui devient ainsi un quadruple :

- cela n'apparaît pas dans l'exemple mais un producteur peut bien avoir plusieurs cuvées dont il a plusieurs millésimes pour une même appellation,
- certaines bouteilles n'ont pas de cuvée spécifiée car le producteur ne produit qu'un seul vin par appellation, il faudra donc un nom par défaut (e.g., le nom de l'appellation, la chaîne vide),
- pour le champagne non millésimé, comme l'année fait partie de la clef, elle ne peut pas être nulle et il faudra faire un choix (e.g., l'année d'achat ou 1900), on aura le même problème si on souhaitait étendre aux alcools distillés

Cette proposition de schéma a donc un *impact* sur les données car il faut, d'une façon ou d'une autre, remplir les colonnes où il y a des trous dans la feuille de calcul avant d'importer dans la base. Une autre proposition qui aurait comme postulat *on ne doit pas toucher aux données d'origine* aboutirait à un modèle substantiellement différent, moins contraint.

Une alternative à l'entité faible serait de tirer un identifiant qui servirait de clef. C'est très prisé par les développeurs, si ça simplifie un peu la gestion de la clef (ici un quadruple), ça ne résoud rien. Une autre possibilité serait de distinguer *cuvée* de *cuvée millésimée*, dans l'idée que la cuinée reste et que le vigneron la produit, ou pas, d'une année sur l'autre.

#### 2.4.1 Autres questions

Pour la qualité des données, on va s'attendre à des problèmes typiques de données mal typées et saisies par des utilisateurs : unités variables ou absentes dans les cases, cellules vides, erreur de saisies. Le plus difficile étant le champs de *l'appellation* où la variété des saisies est importante (abréviation, casse, orthographe etc.)

Pour l'historisation, il faut préciser ce que l'on veut, mais on pourrait par exemple ajouter une nouvelle association *degustation* entre *date* et *cuvée* avec comme attribut un commentaire. L'idée serait de remplir la table associée après destockage.