

# Avant toute chose : comprendre les données brutes

- **Clés primaires des données**
- Nombre de lignes (objets)  
& de colonnes (attributs)
- Associations, multiplicité ?
- À quoi ces données  
peuvent-elles servir ?

Enregistrement automatique

Region

Rechercher

Fichier

Accueil

Insertion

Mise en page

Formules

Données

Révision

Affichage

Développeur

Coller

Police

Alignement

Nombre

Styles

Code\_dep\_code\_commune

	A	B	C	D	E	F	G	H		
1	Code_dep	cc	reg_code	reg_nom	aca_nom	dep_nom	com_nom	mid	dep_code	dep_nom_num
2		1001		84 Auvergne-Rhône-Alpes	Lyon	Ain	L'ABERGEMENT		1	Ain (01)
3		1002		84 Auvergne-Rhône-Alpes	Lyon	Ain	L'ABERGEMENT		1	Ain (01)
4		1003		84 Auvergne-Rhône-Alpes	Lyon	Ain	AMAREINS		1	Ain (01)
5		1004		84 Auvergne-Rhône-Alpes	Lyon	Ain	AMBERIEU EN		1	Ain (01)
6		1005		84 Auvergne-Rhône-Alpes	Lyon	Ain	AMBERIEUX E		1	Ain (01)
7		1006		84 Auvergne-Rhône-Alpes	Lyon	Ain	AMBLEON		1	Ain (01)
8		1007		84 Auvergne-Rhône-Alpes	Lyon	Ain	AMBRONAY		1	Ain (01)
9		1008		84 Auvergne-Rhône-Alpes	Lyon	Ain	AMBUTRIX		1	Ain (01)
10		1009		84 Auvergne-Rhône-Alpes	Lyon	Ain	ANDERT ET C		1	Ain (01)
11		1010		84 Auvergne-Rhône-Alpes	Lyon	Ain	ANGLEFORT		1	Ain (01)
12		1011		84 Auvergne-Rhône-Alpes	Lyon	Ain	APREMONT		1	Ain (01)
13		1012		84 Auvergne-Rhône-Alpes	Lyon	Ain	ARANC		1	Ain (01)
14		1013		84 Auvergne-Rhône-Alpes	Lyon	Ain	ARANDAS		1	Ain (01)
15		1014		84 Auvergne-Rhône-Alpes	Lyon	Ain	ARBENT		1	Ain (01)
16		1015		84 Auvergne-Rhône-Alpes	Lyon	Ain	ARBIGNIEU		1	Ain (01)
17		1016		84 Auvergne-Rhône-Alpes	Lyon	Ain	ARBIGNY		1	Ain (01)
18		1017		84 Auvergne-Rhône-Alpes	Lyon	Ain	ARGIS		1	Ain (01)
19		1018		84 Auvergne-Rhône-Alpes	Lyon	Ain	ARLOD		1	Ain (01)
20		1019		84 Auvergne-Rhône-Alpes	Lyon	Ain	ARMIX		1	Ain (01)

<

>

Region

+

Prêt

Accessibilité : non disponible

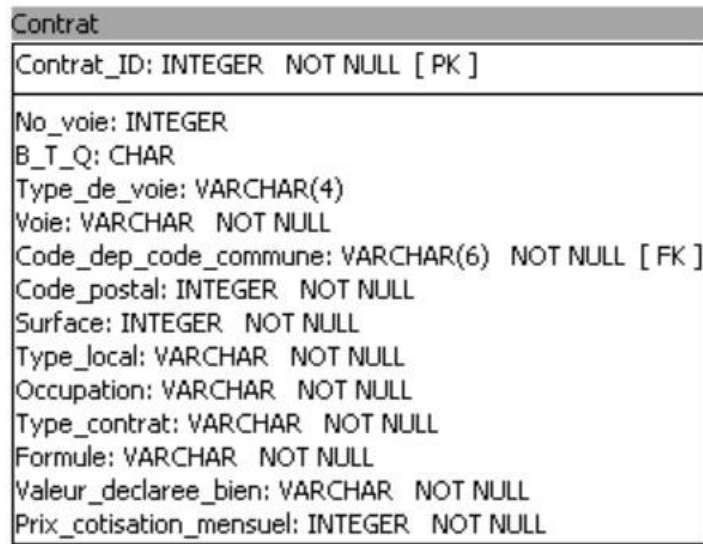
Moyenne : 16731

# Formaliser le format des données : dictionnaire des données

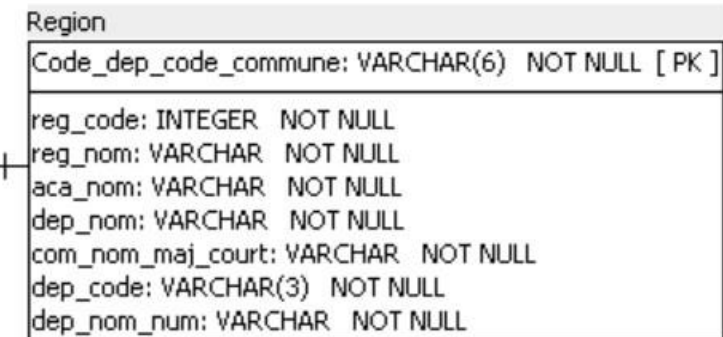
- **Clés primaires, clés étrangères ?**
- Type des données : texte, nombre ?
- Un champ peut-il être vide ?
- Données texte : taille maximale ?

	Nom des colonnes	Type de données	NULL ?	Taille	Clé	Description
CONTRAT.CSV	Contrat_ID	INTEGER	NOT		Clé primaire	Id unique pour les contrats
	No_voie	INTEGER				Numéro dans la voie pour l'adresse du logement assuré
	B_T_Q	CHAR				Indicateur éventuel de répétition pour l'adresse du logement assuré sur un caractère
	Type_de_voie	VARCHAR		4		Type de voie pour l'adresse du logement assuré: rue, av (Avenue), rte (Route), ...
	Voie	VARCHAR	NOT			Libellé de la voie pour l'adresse du logement assuré
	Code_dep_code_commune	VARCHAR	NOT	6	Clé étrangère	Concaténation du code département et code commune pour avoir une clé unique
	Code_postal	INTEGER	NOT			Code postal pour l'adresse du logement assuré
	Surface	INTEGER	NOT			Surface (en m²) du logement assuré
	Type_local	VARCHAR	NOT			Type de bien : appartement ou maison
	Occupation	VARCHAR	NOT			Statut immobilier de l'occupant du logement (contractant) : locataire ou propriétaire
	Type_contrat	VARCHAR	NOT			Type de contrat pour le logement assuré : résidence principale/secondaire, mise en location
	Formule	VARCHAR	NOT			Formule du contrat pour le logement assuré : classique ou integral
REGION.CSV	Valeur_declaree_bien	VARCHAR	NOT			Fourchette de la valeur déclarée du bien (en €)
	Prix_cotisation_mensuel	INTEGER	NOT			Montant (en €) de la cotisation mensuelle stipulée dans le contrat
	Code_dep_code_commune	VARCHAR	NOT	6	Clé primaire	Concaténation du code département et code commune pour avoir une clé unique
	reg_code	INTEGER	NOT			Code de la région administrative française (0 = Collectivités d'Outre-Mer, 1 chiffre = Outre-Mer, 2 chiffres = France métropolitaine ou Corse)
	reg_nom	VARCHAR	NOT			Libellé de la région administrative
	aca_nom	VARCHAR	NOT			Libellé de l'académie de rattachement
	dep_nom	VARCHAR	NOT			Libellé du département
	com_nom_maj_court	VARCHAR	NOT			Libellé du nom abrégé de la commune, uniquement avec des lettres capitales et des espaces
	dep_code	VARCHAR	NOT	3		Code du département : 2 chiffres pour les départements de France métropolitaine, 2A ou 2B pour la Corse, 3 chiffres pour les Outre-mers
	dep_nom_num	VARCHAR	NOT			Concaténation nom du département - espace - code département entre parenthèses

# Modèle relationnel / logique des données : diagramme UML



Réalisé avec **SQL Power Architect**



**Intérêt pratique** : **SQL Power Architect** génère du code SQL directement utilisable avec un SGBDR

# Créer la structure de la base de données dans le SGBDR

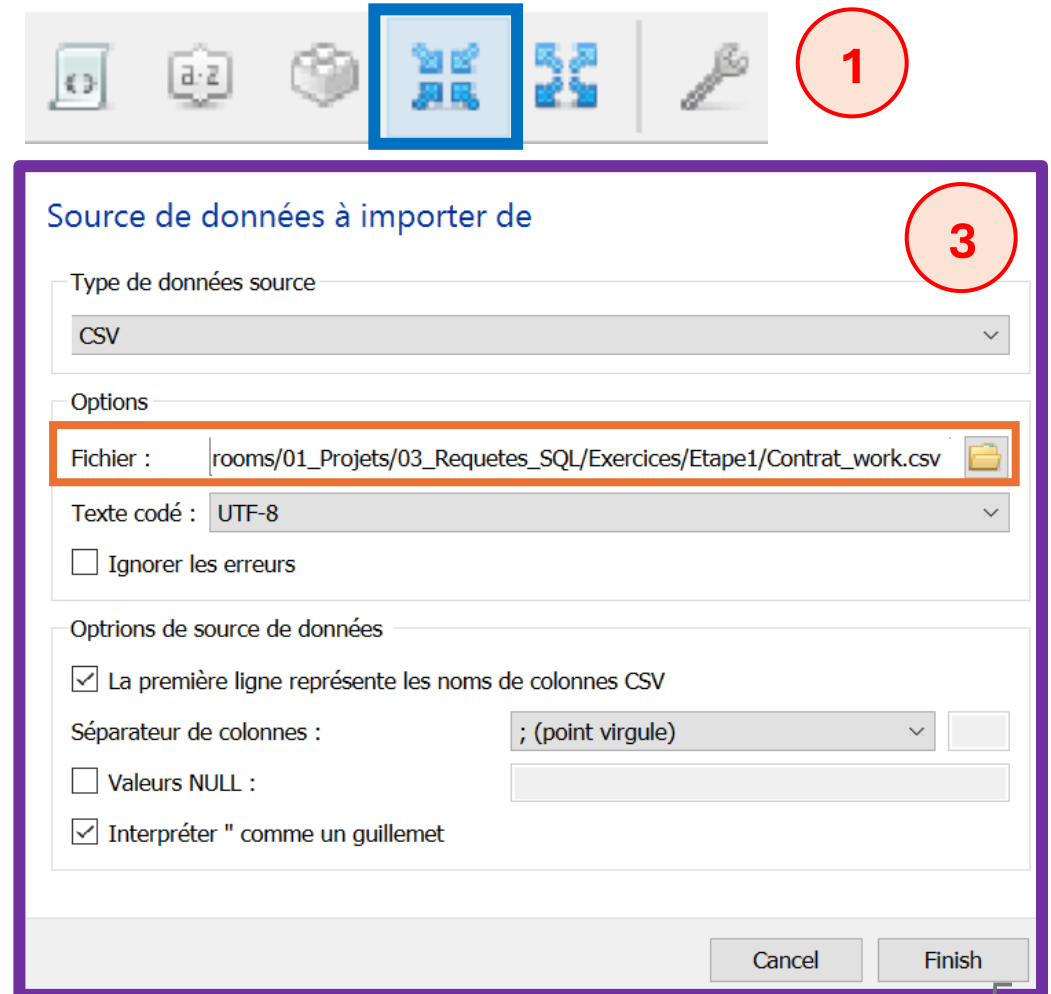
```
CREATE TABLE Region (  
    Code_dep_code_commune VARCHAR(6) NOT NULL,  
    reg_code INTEGER NOT NULL,  
    reg_nom VARCHAR NOT NULL,  
    aca_nom VARCHAR NOT NULL,  
    dep_nom VARCHAR NOT NULL,  
    com_nom_maj_court VARCHAR NOT NULL,  
    dep_code VARCHAR(3) NOT NULL,  
    dep_nom_num VARCHAR NOT NULL,  
    CONSTRAINT region_pk PRIMARY KEY (Code_dep_code_commune)  
);
```

```
CREATE TABLE Contrat (  
    Contrat_ID INTEGER NOT NULL,  
    No_voie INTEGER,  
    B_T_Q CHAR,  
    Type_de_voie VARCHAR(4),  
    Voie VARCHAR NOT NULL,  
    Code_dep_code_commune VARCHAR(6) NOT NULL,  
    Code_postal INTEGER NOT NULL,  
    Surface INTEGER NOT NULL,  
    Type_local VARCHAR NOT NULL,  
    Occupation VARCHAR NOT NULL,  
    Type_contrat VARCHAR NOT NULL,  
    Formule VARCHAR NOT NULL,  
    Valeur_declaree_bien VARCHAR NOT NULL,  
    Prix_cotisation_mensuel INTEGER NOT NULL,  
    CONSTRAINT contrat_pk PRIMARY KEY (Contrat_ID)  
);
```

# Importer les données brutes dans la base de données

## Sous SQLiteStudio :

1. Bouton **Importer**
2. Préciser dans quelle table de la BDD seront chargées les données brutes
3. Préciser la **source** et le **format** des données brutes  
ex : CSV encodage UTF-8  
séparateur ‘;’



# Vérifier l'import des données

Les nombres de lignes et de colonnes de chaque table concordent avec ceux des fichiers source

The screenshot shows the SQLiteStudio interface with the 'data\_assu' database selected. The left sidebar shows the 'Contrat' table with 14 columns. The main window displays a query: `1 SELECT * FROM Contrat`. A blue box highlights the number **30335**. Below the query, the 'Table' tab shows the first six rows of the 'Contrat' table.

	Contrat ID	No voie	B T Q	Type de v	Voie	C
1	100601	190	A	RUE	CENTRALE	1
2	100602	347		RUE	DU CHATEAU	1
3	100603	58		AV	DU MONT BLANC	1
4	100604	140		RUE	DE L'ABBE JOLIVET	1
5	100605	39		RUE	BUFFON	1
6	100606	8		RUE	DE GENEVE	1

The screenshot shows the SQLiteStudio interface with the 'data\_assu' database selected. The left sidebar shows the 'Region' table with 8 columns. The main window displays a query: `1 SELECT * FROM Region`. A blue box highlights the number **39816**. Below the query, the 'Table' tab shows the first seven rows of the 'Region' table.

	Code dep	req code	req nom	aca nom	dep nom	com nom
1	1001	84	Auvergne-Rhône-Alpes	Lyon	Ain	L ABERGE
2	1002	84	Auvergne-Rhône-Alpes	Lyon	Ain	L ABERGE
3	1003	84	Auvergne-Rhône-Alpes	Lyon	Ain	AMAREIN
4	1004	84	Auvergne-Rhône-Alpes	Lyon	Ain	AMBERIEL
5	1005	84	Auvergne-Rhône-Alpes	Lyon	Ain	AMBERIEL
6	1006	84	Auvergne-Rhône-Alpes	Lyon	Ain	AMBLEON
7	1007	84	Auvergne-Rhône-Alpes	Lyon	Ain	AMBRON

# Contrôle de la clé étrangère

Est-ce que toutes les valeurs de la clé étrangère de la table fille existent bien dans la clé primaire de la table mère ?






Exemple de requête :


```
SELECT DISTINCT FKey, Attribut_controle  
FROM Table_fille  
WHERE FKey NOT IN (SELECT PKey FROM Table_mere);
```

Si nécessaire et si possible, corriger les discordances dans les fichiers source et réimporter les données brutes


# Assignment de la clé étrangère sous *SQLiteStudio*


1. Dans l'arborescence de la BDD à gauche, double-cliquer sur la table fille
2. Dans le **gestionnaire des attributs**, double-cliquer sur l'attribut concerné
3. Parmi les **contraintes**, cocher **Clef étrangère** et configurer l'assignation

	Nom	Type de données	Clé primaire	Clé étrangère	Unique	Contrôle	Non NULL	Collecter	Généré
1	Contrat_ID	INTEGER							
2	No_voie	INTEGER							
3	B_T_Q	CHAR							
4	Type_de_voie	VARCHAR (4)							
5	Voie	VARCHAR							
6	Code_dep_code_commune	VARCHAR (6)							
7	Code_postal	INTEGER							
8	Surface	INTEGER							
9	Type_local	VARCHAR							

 **FOREIGN KEY** (Références table Region, colonne Code\_dep\_code\_commune)

**Contraintes**

☐  Clef primaire Conf

☒  Clef étrangère Configurer



# Exemple de requête (#9) :

Quelle est la surface moyenne des logements sous contrat à Paris ?

Code SQL :

```
SELECT avg(surface)
FROM Contrat
WHERE code_dep_code_commune IN (
    SELECT code_dep_code_commune
    FROM Region
    WHERE lower(com_nom_maj_court)
        LIKE 'paris%'
    AND dep_code = '75'
);
```

Résultat :

Table		Formulaire	
			
			
1		 Nombre de lignes chargées : 1	
avg(surface)			
1	51.76954988591579		










# Exemple de requête (#10) :

les 10 départements où le prix moyen de la cotisation est le plus élevé

Code SQL :

```
SELECT dep_nom_num, avg(prix_cotisation_mensuel)
FROM (
    SELECT code_dep_code_commune,
           prix_cotisation_mensuel, dep_nom_num
    FROM Contrat c
    LEFT JOIN (
        SELECT code_dep_code_commune rcdcc,
               dep_nom_num
        FROM Region)
    ON c.code_dep_code_commune = rcdcc)
GROUP BY dep_nom_num
ORDER BY avg(prix_cotisation_mensuel) DESC LIMIT 10 ;
```

Résultat :

Table		Formulaire
		     1    Nombre de lignes chargées : 10
	dep nom num	avq(prix cotisation mensuel)
1	Paris (75)	36.39908732628086
2	Hauts-de-Seine (92)	26.26935840707965
3	Val-de-Marne (94)	19.82416396979504
4	Yvelines (78)	18.88806970509383
5	Rhône (69)	18.48551829268293
6	Ain (01)	18.23888888888889
7	Alpes-Maritimes (06)	18.13762287756926
8	Charente-Maritime (17)	17.31730769230769
9	Haute-Savoie (74)	17.14873417721519
10	Corse-du-Sud (2A)	17.07438016528926

# Exemple de requête (#12) :

## Nombre de contrats pour chaque région






Code SQL :

```
SELECT reg_nom, count(contrat_id)
FROM (
    SELECT code_dep_code_commune, reg_nom,
           reg_code, contrat_id
    FROM Region r
    LEFT JOIN (
        SELECT code_dep_code_commune ccdcc,
               contrat_id
        FROM Contrat)
    ON r.code_dep_code_commune = ccdcc
    WHERE reg_code > 0)
GROUP BY reg_nom
ORDER BY count(contrat_id) DESC ;
```




Résultat :

Table

Formulaire



1








Nombre de lignes chargées : 18




	req nom	count(contrat id)
1	Ile-de-France	14177
2	Provence-Alpes-Côte d'Azur	3279
3	Auvergne-Rhône-Alpes	3042
4	Nouvelle-Aquitaine	2038
5	Occitanie	1609
6	Pays de la Loire	1196
7	Hauts-de-France	1189
8	Bretagne	947
9	Normandie	824

Table

Formulaire



1



Nombre de lignes chargées : 18

	req nom	count(contrat id)
10	Grand Est	769
11	Centre-Val de Loire	598
12	Bourgogne-Franche-Comté	293
13	Corse	247
14	Martinique	73
15	Guyane	37
16	La Réunion	17
17	Mayotte	0
18	Guadeloupe	0