



Présentation projet Dat'Habitation

Méthodologie et étapes suivies





DB Browser 3.12.2 pour la création de la base de données DATASSUR.DB par importation directe des fichiers CSV [DB Browser for SQLite](#)

Contrats
123 Contrat_ID
123 No_voie
ABC B_T_Q
ABC Type_de_voie
ABC Voie
ABC Code_postal
ABC Commune
ABC Code_departement
123 Surface
ABC Type_local
ABC Occupation
ABC Type_contrat
ABC Formule
ABC Valeur_declaree_biens
123 Prix_cotisation_mensuel

Region
ABC Code_postal
ABC reg_code
ABC reg_nom
ABC aca_nom
ABC dep_nom
ABC com_nom_maj
ABC dep_code
ABC dep_nom_num



DB Browser for SQLite - \\mac\Home\Library\CloudStorage\GoogleDrive-fa.leroy57@gmail.com\Mon Drive\PROJETS OPENCLASSROOMS\PROJET 3\DATASSUR.db

Fichier Édition Vue Outils Aide

Ouvrir un Projet Enregistrer le projet

Structure de la Base de Données Parcourir les données Éditer les Pragma Exécuter le SQL

Créer une table Créer un Index Imprimer

Nom	Type	Schéma
Tables (2)		
Contrats		CREATE TABLE Contrats (Contrat_ID INTEGER PRIMARY KEY, No_voie INTEGER (5), B_T_Q TEXT (10), Type_de...
Contrat_ID	INTEGER	"Contrat_ID" INTEGER
No_voie	INTEGER(5)	"No_voie" INTEGER(5)
B_T_Q	TEXT(10)	"B_T_Q" TEXT(10)
Type_de_voie	TEXT(3)	"Type_de_voie" TEXT(3)
Voie	TEXT(20)	"Voie" TEXT(20)
Code_postal	TEXT(5)	"Code_postal" TEXT(5)
Commune	TEXT(25)	"Commune" TEXT(25)
Code_departement	TEXT(3)	"Code_departement" TEXT(3)
Surface	REAL(4)	"Surface" REAL(4)
Type_local	TEXT(10)	"Type_local" TEXT(10)
Occupation	TEXT(12)	"Occupation" TEXT(12)
Type_contrat	TEXT(20)	"Type_contrat" TEXT(20)
Formule	TEXT(10)	"Formule" TEXT(10)
Valeur_declaree_biens	TEXT(15)	"Valeur_declaree_biens" TEXT(15)
Prix_cotisation_mensuel	INTEGER(3)	"Prix_cotisation_mensuel" INTEGER(3)
Region		CREATE TABLE Region (Code_postal TEXT (5), reg_code TEXT (2), reg_nom TEXT (35), aca_nom TEXT (20), d...
Code_postal	TEXT(5)	"Code_postal" TEXT(5)
reg_code	TEXT(2)	"reg_code" TEXT(2)
reg_nom	TEXT(35)	"reg_nom" TEXT(35)
aca_nom	TEXT(20)	"aca_nom" TEXT(20)
dep_nom	TEXT(20)	"dep_nom" TEXT(20)
com_nom_maj	TEXT(35)	"com_nom_maj" TEXT(35)
dep_code	TEXT(3)	"dep_code" TEXT(3)
dep_nom_num	TEXT(20)	"dep_nom_num" TEXT(20)
Index (0)		
Vues (0)		
Déclencheurs (0)		

SGBD utilisé: SQLite Studio 3.4.1

- Importation de DATASSUR.DB créé avec DB Browser.
- Vérification des 2 tables et des colonnes.
- Vérification et modification si nécessaire du type des données (VARCHAR, INT, FLOAT) et nombre de caractères max.
- Vérification et déclaration si nécessaire des clés primaires et étrangères.

The screenshot displays the SQLite Studio 3.4.1 interface. The left sidebar shows the database structure for 'DATASSUR (SQLite 3)'. It contains two tables: 'Contrats' and 'Region'. The 'Contrats' table has 15 columns, and the 'Region' table has 8 columns. The right pane shows the 'Structure' tab for both tables, listing columns, data types, and constraints.

Contrats (DATASSUR)

	Nom	Type de données	Primary Key	Foreign Key	Unique	Contrôle	Not NULL	Collate	Generated
1	Contrat_ID	INTEGER	✓						NULL
2	No_voie	INTEGER (5)							NULL
3	B_T_Q	TEXT (10)							NULL
4	Type_de_voie	TEXT (3)							NULL
5	Voie	TEXT (20)							NULL
6	Code_postal	TEXT (5)		✓					NULL
7	Commune	TEXT (25)							NULL
8	Code_departement	TEXT (3)							NULL
9	Surface	REAL (4)							NULL
10	Type_local	TEXT (10)							NULL
11	Occupation	TEXT (12)							NULL
12	Type_contrat	TEXT (20)							NULL
13	Formule	TEXT (10)							NULL
14	Valeur_declaree_biens	TEXT (15)							NULL
15	Prix_cotisation_mensuel	INTEGER (3)							NULL

Region (DATASSUR)

	Nom	Type de données	Primary Key	Foreign Key	Unique	Contrôle	Not NULL	Collate	Generated
1	Code_postal	TEXT (5)	✓						NULL
2	reg_code	TEXT (2)							NULL
3	reg_nom	TEXT (35)							NULL
4	aca_nom	TEXT (20)							NULL
5	dep_nom	TEXT (20)							NULL
6	com_nom_maj	TEXT (35)							NULL
7	dep_code	TEXT (3)							NULL
8	dep_nom_num	TEXT (20)							NULL

Types de données

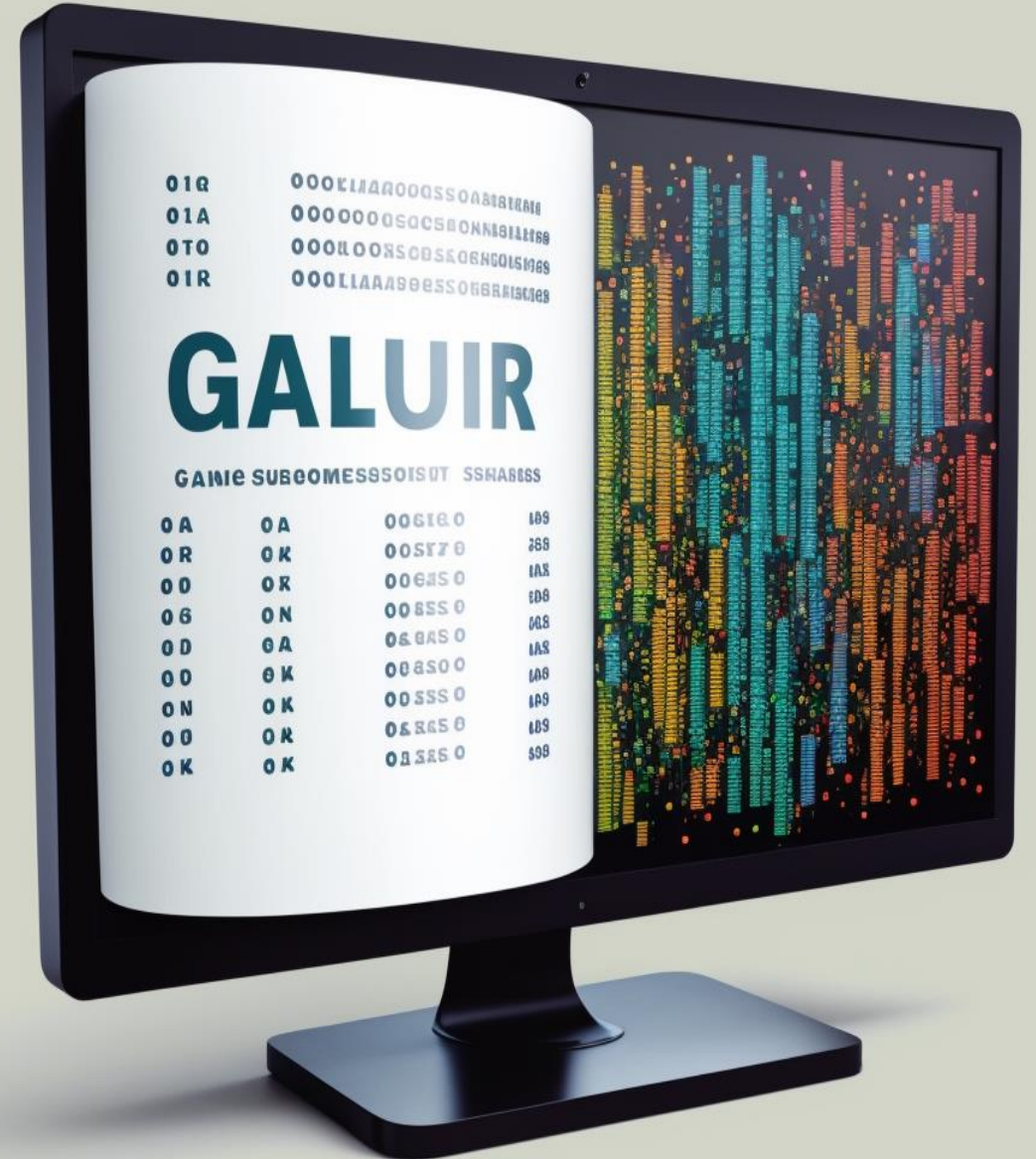
Les types de données SQL définissent le format de la donnée qui peut être stockée dans une colonne ou une variable

```
CREATE <table> (  
    <nom_colonne> <type_données> <contraintes>,  
)
```

Colonne 'voie' dans la table 'contrat'
Type CHAR car stocke du texte.

Colonne 'Surface' dans la table 'contrat'
Type FLOAT (ou REAL) car stocke des nombres à virgule flottante.

Colonne 'Contrat_id' dans la table 'contrat'
Type INT (ou REAL) car stocke des nombres entiers.



WHERE

Le WHERE est utilisé pour filtrer les résultats. Il peut être utilisé avec une combinaison de conditions et d'opérateurs logiques.

*Lister les numéros de contrats (contrat_ID)
avec leur surface pour la commune de Caen*

```
select  
contrat_id, surface  
from Contrats  
where Commune = 'CAEN'
```

contrat_id	surface
103791	35
103792	99
103793	40
103794	20





Group by

permet de regrouper les résultats d'une requête selon une ou plusieurs colonne.

Nombre de contrats pour chaque catégorie de prix de la valeur déclarées des biens ?

```
select valeur_declaree_biens, count(contrat_id) as  
nb_contrats  
from Contrats  
group by valeur_declaree_biens  
order by nb_contrats desc
```

valeur_declaree_biens	nb_contrats
0-25000	22720
25000-50000	6815
50000-100000	696
100000+	104



ORDER BY

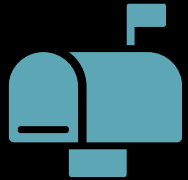
ORDER BY vous permet de trier vos résultats en fonction d'une colonne spécifique. Vous pouvez également définir si vous souhaitez trier les résultats par ordre croissant ou décroissant.

Quels sont les 5 contrats qui ont la surface la plus élevées ?

```
Select
Contrat_id as 'Contrats', Surface
from Contrats
order by Surface desc
limit 5
```

Contrats	Surface
104211	815
105463	742
130878	595
100822	570
109872	559

FONCTIONS



HAVING

Précise un critère de restriction, lié à la clause GROUP BY

Liste des communes ayant eu au moins 150 contrats

```
SELECT commune, COUNT(*) as nb_contrats
FROM contrats
GROUP BY commune
HAVING COUNT(*) >= 150
ORDER BY nb_contrats DESC
```



DISTINCT

permet de sélectionner les valeurs uniques d'une colonne dans les résultats d'une requête.

Lister le nom des régions de France

```
SELECT DISTINCT reg_nom as 'Régions'
From Region
```



AVG ou COUNT

Permet de retourner des valeurs moyennes ou un nombre d'enregistrement.

Quel est le prix moyen de la cotisation mensuelle ?

```
Select
round(avg(Prix_cotisation_mensuel), 2) as 'Prix moyen de la cotisation mensuelle'
from Contrats
```


Relations entre tables

Une base de données peut contenir plusieurs tables qui sont liées entre elles. Pour extraire des informations à partir de plusieurs tables, nous pouvons utiliser des jointures qui combinent des données d'une table à celles d'une autre.

Quel est le nombre de contrats avec des formules "intégral" pour la région Pays de la Loire ?

```
SELECT  
count(contrat_id) as 'Nombre de contrats', reg_nom as 'nom de  
Région', formule  
from Contrats  
join Region on contrats.Code_postal = region.Code_postal  
where region.reg_nom = 'Pays de la Loire' and  
Contrats.Formule = 'Integral'
```

Nombre de contrats	nom de Région	formule
590	Pays de la Loire	Integral

