



Projet 3 :

**Requêtez une base de données
avec SQL**

Plan de réalisation

- ▶ 1: Identification des différents types de données
- ▶ 2 : Création du schéma relationnel normalisé en 3NF ([SQL Power Architects](#))
- ▶ 3 : Code SQL de génération des tables ([SQL Power Architects](#))
- ▶ 4 : Création des tables et import des données ([SGBD SQLite](#))
- ▶ 5 : Interrogation de la base de données ([SGBD SQLite](#))

Identification des différents types de données

1

CONTRAT

	Nom des colonnes	Type de données	Taille	Clé	Description
CONTRAT.CSV	Contrat_ID	INT	6	Clé primaire	Id unique pour les contrats
	No_voie	INT	4		Numéro dans la voie pour l'adresse du logement assuré
	B_T_Q	CHAR	1		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	26		Libellé de la voie pour l'adresse du logement assuré
	Code_dep_code_commune	VARCHAR	6	Clé secondaire	Concaténation du code département et code commune pour avoir une clé unique
	Code_postal	INT	5		Code postal pour l'adresse du logement assuré
	Commune	VARCHAR	29		Libellé de la commune de l'adresse du logement
	Code_département	VARCHAR	3		Code département
	Surface	float	3		Surface en m²
	Type_local	VARCHAR	11		Type du local
	Occupation	VARCHAR	12		Statut de l'occupant
	Type_contrat	VARCHAR	20		Type du contrat
	Formule	VARCHAR	9		Formule du contrat souscrite
	Valeur_declaree_biens	VARCHAR	12		Valeur des biens déclarée
REGION.CSV	Prix_cotisation_mensuel	float	3		Prix de la cotisation mensuel du contart
	Code_dep_code_commune	VARCHAR	6	Clé primaire	Concaténation du code département et code commune pour avoir une clé unique
	reg_code	INT	2		Code de la région
	reg_nom	VARCHAR	26		Libellé région
	aca_nom	VARCHAR	24		Libellé de l'académie
	dep_nom	VARCHAR	43		Libellé du département
	com_nom_maj_court	VARCHAR	32		Libellé court de la commune en majuscule
	dep_code	VARCHAR	3		Code département
	dep_nom_num	VARCHAR	49		Libellé du département et code

REGION

Schéma relationnel normalisé en 3NF

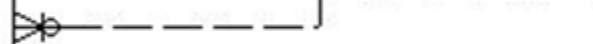
2

Contrat

Contrat_ID: INTEGER NOT NULL [PK]
No_voie: INTEGER
B_T_Q: CHAR(1)
Type_de_voie: VARCHAR(4)
Voie: VARCHAR(26)
Code_dep_code_commune: VARCHAR(6) NOT NULL [FK]
Code_postal: INTEGER
Commune: VARCHAR(29)
Code_département: VARCHAR(3)
Surface: FLOAT
Type_local: VARCHAR(11)
Occupation: VARCHAR(12)
Type_contrat: VARCHAR(20)
Formule: VARCHAR(9)
Valeur_declaree_biens: VARCHAR(12)
Prix_cotisation_mensuel: FLOAT

Region

Code_dep_code_commune: VARCHAR(6) NOT NULL [PK]
Reg_code: INTEGER
Reg_nom: VARCHAR(26)
aca_nom: VARCHAR(24)
dep_nom: VARCHAR(43)
com_nom_maj_court: VARCHAR(32)
dep_code: VARCHAR(3)
dep_nom_num: VARCHAR(49)



- Clé primaire [PK] : **Contrat_ID**
- Clés étrangère [FK] : **Code_dep_code_commune**

- Clé primaire [PK] : **Code_dep_code_commune**

Relation de dépendance de la table " Region " à l'égard de la table " Contrat "

Code SQL de génération des tables

Respect des contraintes
et dépendances
paramétrées

```
CREATE TABLE IF NOT EXISTS Contrat (
  Contrat_ID          INTEGER NOT NULL,
  No_voie             INTEGER,
  B_T_Q              CHAR (1),
  Type_de_voie        VARCHAR (4),
  Voie                VARCHAR (26),
  Code_dep_code_commune VARCHAR (6) NOT NULL,
  Code_postal         INTEGER,
  Commune             VARCHAR (29),
  Code_departement    VARCHAR (3),
  Surface             FLOAT,
  Type_local          VARCHAR (11),
  Occupation          VARCHAR (12),
  Type_contrat        VARCHAR (20),
  Formule             VARCHAR (9),
  Valeur_declaree_biens VARCHAR (12),
  Prix_cotisation_mensuel FLOAT,
  CONSTRAINT Contrat_pk PRIMARY KEY ( Contrat_ID ),
  FOREIGN KEY ( Code_dep_code_commune )
  REFERENCES Region (Code_dep_code_commune) );
```

```
CREATE TABLE IF NOT EXISTS Region (
  Code_dep_code_commune VARCHAR (6) NOT NULL,
  Reg_code              INTEGER,
  Reg_nom               VARCHAR (26),
  aca_nom               VARCHAR (24),
  dep_nom               VARCHAR (43),
  com_nom_maj_court     VARCHAR (32),
  dep_code              VARCHAR (3),
  dep_nom_num           VARCHAR (49),
  CONSTRAINT Region_pk PRIMARY KEY ( Code_dep_code_commune ) );
```


Création des tables et import des données

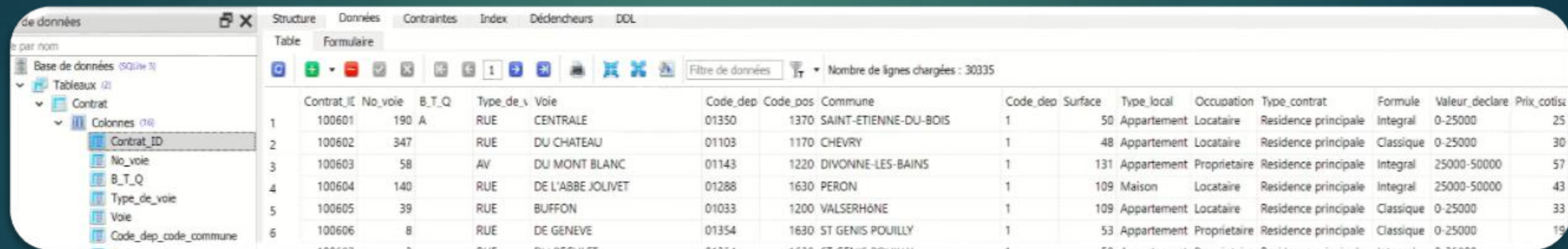
4

- **Création des tables** "Contrat" et "Region" :

1) Copie du code dans l'éditeur SQL du SGBD SQLite

2) Exécution du Code SQL

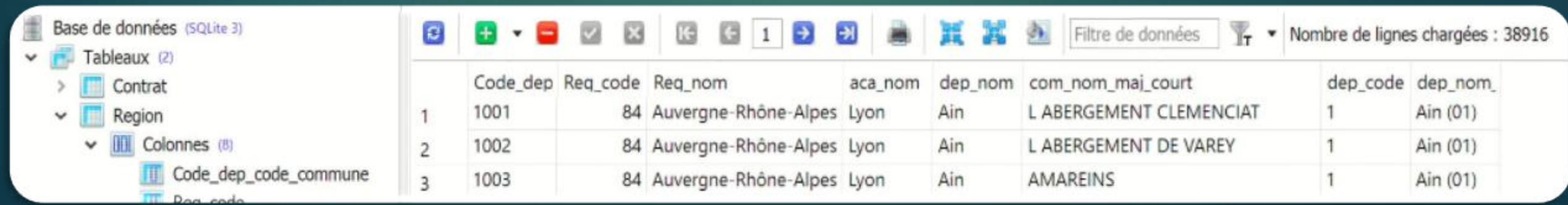
- **Importer** les données de **Contrat.CSV** dans la **table Contrat** : 30 335 Lignes chargées



The screenshot shows the SQLite Data Browser interface. The left sidebar displays the database structure with 'Base de données (SQLite 3)' and 'Tableaux (2)' (Contrat, Region). The 'Contrat' table is selected, showing its columns: Contrat_ID, No_voie, B_T_Q, Type_de_voie, Voie, Code_dep, Code_pos, Commune, Code_dep, Surface, Type_local, Occupation, Type_contrat, Formule, Valeur_declare, and Prix_cotise. The main window shows the first 6 rows of data. The top right corner indicates 'Nombre de lignes chargées : 30335'.

	Contrat_ID	No_voie	B_T_Q	Type_de_voie	Voie	Code_dep	Code_pos	Commune	Code_dep	Surface	Type_local	Occupation	Type_contrat	Formule	Valeur_declare	Prix_cotise
1	100601	190	A	RUE	CENTRALE	01350	1370	SAINT-ETIENNE-DU-BOIS	1	50	Appartement	Locataire	Residence principale	Integral	0-25000	25
2	100602	347		RUE	DU CHATEAU	01103	1170	CHEVRY	1	48	Appartement	Locataire	Residence principale	Classique	0-25000	30
3	100603	58		AV	DU MONT BLANC	01143	1220	DIVONNE-LES-BAINS	1	131	Appartement	Proprietaire	Residence principale	Integral	25000-50000	57
4	100604	140		RUE	DE L'ABBE JOLIVET	01288	1630	PERON	1	109	Maison	Locataire	Residence principale	Integral	25000-50000	43
5	100605	39		RUE	BUFFON	01033	1200	VALSERHÖNE	1	109	Appartement	Locataire	Residence principale	Classique	0-25000	33
6	100606	8		RUE	DE GENEVE	01354	1630	ST GENIS POUILLY	1	53	Appartement	Proprietaire	Residence principale	Classique	0-25000	19

- **Importer** les données de **Region.CSV** dans la **table Region** : 38 916 Lignes chargées



The screenshot shows the SQLite Data Browser interface. The left sidebar displays the database structure with 'Base de données (SQLite 3)' and 'Tableaux (2)' (Contrat, Region). The 'Region' table is selected, showing its columns: Code_dep, Req_code, Req_nom, aca_nom, dep_nom, com_nom_maj_court, dep_code, and dep_nom. The main window shows the first 3 rows of data. The top right corner indicates 'Nombre de lignes chargées : 38916'.

	Code_dep	Req_code	Req_nom	aca_nom	dep_nom	com_nom_maj_court	dep_code	dep_nom
1	1001	84	Auvergne-Rhône-Alpes	Lyon	Ain	L ABERGEMENT CLEMENCIAT	1	Ain (01)
2	1002	84	Auvergne-Rhône-Alpes	Lyon	Ain	L ABERGEMENT DE VAREY	1	Ain (01)
3	1003	84	Auvergne-Rhône-Alpes	Lyon	Ain	AMAREINS	1	Ain (01)

Interrogation de la base de données

5

- Requête 1 :

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

```
SELECT contrat_id, surface
FROM contrat
WHERE Commune = 'CAEN' ;
```

	contrat_id	surface
1	103791	35
2	103792	99
3	103793	40
4	103794	20

- Requête 2 :

Lister les numéros de contrats avec le type de contrat et leur formule pour les maisons du département 71.

```
SELECT Contrat_ID, Type_contrat, Formule
FROM contrat
WHERE Code_departement = '71'
AND Type_local = 'Maison' ;
```

	Contrat_ID	Type_contrat	Formule
1	114768	Residence principale	Integral
2	114779	Residence principale	Classique
3	114782	Residence principale	Classique
4	114812	Residence principale	Integral

Instruction SELECT + Argument : Spécifie les colonnes à récupérer dans le résultat.

Instruction FROM + Argument : Indique la table source.

Instruction WHERE + Conditions : Filtre les lignes en fonction des conditions spécifiées.

Opérateur logique AND : Utilisé pour combiner plusieurs conditions.

- Requête 3 :

Lister le nom des régions de France

```
SELECT DISTINCT Reg_nom
FROM Region
```

Opérateur DISTINCT : Sélectionner uniquement les valeurs distinctes d'une colonne.

	Reg_nom
1	Auvergne-Rhône-Alpes
2	Hauts-de-France
3	Provence-Alpes-Côte d'Azur
4	Grand Est
5	Occitanie
6	Normandie
7	Nouvelle-Aquitaine
8	Centre-Val de Loire
9	Bourgogne-Franche-Comté
10	Bretagne
11	Pays de la Loire
12	Ile-de-France
13	Guadeloupe
14	Martinique
15	Guyane
16	La Réunion
17	Collectivités d'outre-mer
18	Mayotte
19	Corse

- Requête 4 :

Combien existe-t-il de contrats sur les résidences principales ?

```
SELECT COUNT(*) AS Contrat_ID
FROM Contrat
WHERE Type_contrat = 'Residence principale' ;
```

	Nombre_de_Contrats
1	25620

Fonction COUNT() : Compter le nombre d'enregistrements compte toutes les lignes.

***** : Utilisé comme raccourci pour représenter toutes les lignes.

AS : Mot-clé utilisé pour renommer des colonnes ou des résultats.

- Requête 5 :

Quelle est la surface moyenne des logements avec un contrat à Paris ?

```
SELECT AVG(Surface) AS SurfaceMoyenne
FROM Contrat
WHERE Code_departement IN (75) ;
```

	SurfaceMoyenne
1	51.76954988591579

Fonction AVG : Calcule la moyenne de la colonne sélectionné.

- Requête 6 :

Quels sont les 5 contrats qui ont les surfaces les plus élevées ?

```
SELECT *
FROM Contrat
ORDER BY Surface DESC
LIMIT 5 ;
```

	Contrat_ID	Surface
1	104211	815
2	105463	742
3	130878	595
4	100822	570
5	109872	559

Clause ORDER BY : Trier les résultats de la requête basée sur les colonnes spécifiées

DESC : Mot-clé utilisé pour spécifier l'ordre de tri (descendant)

Clause LIMIT : Limite le nombre de lignes renvoyées par la requête.

- Requête 7 :

Quel est le prix moyen de la cotisation mensuelle ?

```
SELECT AVG(Prix_cotisation_mensuel) AS PrixMoyen
FROM Contrat
```

	PrixMoyen
1	19.32869622548212

- Requête 8 :

Quel est le nombre de contrats pour chaque catégorie de prix de la valeur déclarée des biens ?

```
SELECT Valeur_declaree_biens, COUNT(*)
      AS Nombre_de_contrat
FROM Contrat
GROUP BY Valeur_declaree_biens
```

	Valeur declaree biens	Nombre de contrat
1	0-25000	22720
2	100000+	104
3	25000-50000	6815
4	50000-100000	696

Clause GROUP BY : Groupe les résultats en fonction des valeurs d'une ou plusieurs colonnes

- Requête 9 :

Quel est le classement des 10 départements où le prix moyen de la cotisation est le plus élevé ?

```
SELECT Code_departement,
      AVG(Prix_cotisation_mensuel) AS PrixMoyenCotisation
FROM Contrat
GROUP BY Code_departement
ORDER BY PrixMoyenCotisation DESC
LIMIT 10
```

	Code departement	PrixMoyenCotisation
1	75	36.39908732628086
2	92	26.26935840707965
3	94	19.82416396979504
4	78	18.880187416332
5	69	18.45681818181818
6	1	18.23888888888889
7	6	18.13762287756926
8	17	17.31730769230769
9	74	17.1583850931677
10	20	17.02834008097166

Requête 10 :

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

```
SELECT COUNT(*) AS NombreContrats
FROM Contrat
LEFT JOIN Region
ON Contrat.Code_dep_code_commune =
   Region.Code_dep_code_commune
WHERE Region.reg_nom = "Pays de la Loire"
AND Contrat.Formule ="Integral";
```

	NombreContrats
1	561

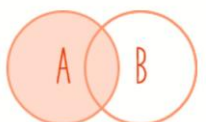
Jointure LEFT JOIN : Renvoie tous les enregistrements de la table de gauche (A) ainsi que les enregistrements correspondants de la table de droite (B)

Le nombre de contrats et la région sont inscrits dans des tables séparées, il faut donc effectuer une **jointure Gauche** afin de compléter la table la plus grande (**Contrat**) avec les informations de la table la plus petite (**Region**)

Mot-clé **ON** + Conditions : Spécifie la condition qui doit être satisfaite pour établir une correspondance entre les tables.

Ici les valeurs de la colonne "Code_dep_code_commune" dans la table "Contrat" doivent être égales à celles dans la table "Region"

La jointure Left



- Requête 11 :

Quelle est la liste des communes ayant au moins 150 contrats ?

```
SELECT Commune, COUNT(*) AS NombreContrats
FROM Contrat
GROUP BY Commune
HAVING NombreContrats >= 150
```

	Commune	NombreContrats
1	0	4021
2	BORDEAUX	302
3	COURBEVOIE	163
4	GRENOBLE	220
5	LILLE	161
6	NANTES	291
7	NICE	387
8	PARIS 10	263
9	PARIS 11	381
10	PARIS 12	252
11	PARIS 14	222
12	PARIS 15	407
13	PARIS 16	394
14	PARIS 17	468
15	PARIS 18	515
16	PARIS 19	266
17	PARIS 20	302
18	TOULON	170
19	TOULOUSE	187

Clause HAVING + condition : Utilisée pour filtrer les résultats définis par la clause GROUP BY basée sur des conditions spécifiées.

Ici les valeur de la table Contrat supérieur ou égale à 150 groupé par commune.

- Requête 12 :

Quel est le nombre de contrats pour chaque région ?

```
SELECT Reg_nom, COUNT(Contrat.contrat_ID)
AS NombreContrats
FROM Region
LEFT JOIN Contrat
ON Contrat.Code_dep_code_commune =
Region.Code_dep_code_commune
GROUP BY Region.reg_nom
```

	Reg_nom	NombreContrats
1	Auvergne-Rhône-Alpes	2588
2	Bourgogne-Franche-Comté	402
3	Bretagne	945
4	Centre-Val de Loire	804
5	Collectivités d'outre-mer	0
6	Corse	247
7	Grand Est	776
8	Guadeloupe	0
9	Guyane	37
10	Hauts-de-France	1200
11	Ile-de-France	13474
12	La Réunion	4
13	Martinique	60
14	Mayotte	0
15	Normandie	898
16	Nouvelle-Aquitaine	2097
17	Occitanie	1809
18	Pays de la Loire	1138
19	Provence-Alpes-Côte d'Azur	1948

Jointure LEFT JOIN : Renvoie tous les enregistrements de la table de gauche ainsi que les enregistrements correspondants de la table de droite

Ici mettre la table Region à gauche permet d'afficher les noms des régions pour lesquelles le nombre de contrats est égal à zéro.

Mettre la table Contrat à gauche me permettrait d'obtenir le nombre de contrats où aucune région n'est indiquée :

	Req nom	NombreContrats
1	NULL	1908
2	Auvergne-Rhône-Alpes	2588