



# Requêtez une base de données avec SQL

(Structured Query Language)

## Projet N° 3

---

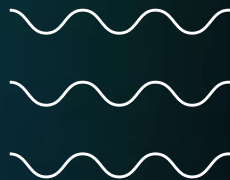
1er mars 2024

# Objectifs du projet



- 01 Créer une base de données en respect des normes réglementaires et des besoins clients
- 02 Gérer une base de données établissant des requêtes et en respect du RGPD

*2 tableaux Excel seront fournis pour gérer une base de données de contrats d'assurance immobilière*



# Dictionnaire des données



01

## Observation du contenu des colonnes

Examiner attentivement les données présentes dans chaque colonne des 2 tables pour comprendre la nature des informations qu'elles contiennent.

02

## Recherche des typologies de données

Déterminer les types de données appropriés pour chaque colonne en fonction du contenu, par exemple des types tels que VARCHAR, INT, DATE, etc. pour représenter différentes informations.

03

## Définition des contraintes

Les contraintes de la clé primaire et de la clé étrangères seront définies, tout comme les contraintes d'intégrité et de domaine.



# Dictionnaire des données

|             | Nom des colonnes        | Type de données | Taille | Clé            | Description  |
|-------------|-------------------------|-----------------|--------|----------------|--|
| CONTRAT.CSV | Contrat_ID              | INT             |        | Clé primaire   | Identifiant unique du contrat  |
|             | No_voie                 | VARCHAR         | 255    |                | Numéro dans la voie pour l'adresse du logement assuré  |
|             | B_T_Q                   | CHAR            | 1      |                | Bâtiment, Tour ou Quartier, s'il y en a un.  |
|             | Type_de_voie            | VARCHAR         | 5      |                | Type de voie pour l'adresse du logement assuré: rue, av (Avenue), rte (Route), ...   |
|             | Voie                    | VARCHAR         | 100    |                | Libellé de la voie pour l'adresse du logement assuré   |
|             | Code_dep_code_commune   | VARCHAR         | 100    | Clé secondaire | Concaténation du code département et code commune pour avoir une clé unique  |
|             | Code_postal             | VARCHAR         | 5      |                | Code postal pour l'adresse du logement assuré  |
|             | Commune                 | VARCHAR         | 100    |                | Libellé de la commune de l'adresse du logement   |
|             | Code_departement        | INT             |        |                | Code départemental   |
|             | Surface                 | INT             |        |                | Surface du local   |
|             | Type_local              | VARCHAR         | 100    |                | Type de local (appartement, maison, etc...).   |
|             | Occupation              | VARCHAR         | 100    |                | Occupation du local (propriétaire, locataire, etc.).   |
|             | Type_contrat            | VARCHAR         | 100    |                | Type de contrat d'assurance (résidence principale ou secondaire).  |
|             | Formule                 | VARCHAR         | 100    |                | Formule d'assurance (classique ou intégral).   |
| REGION.CSV  | Valeur_declaree_biens   | VARCHAR         | 100    |                | Valeur déclarée des biens assurés (4 Tranches de "0-25 000 €", "25 000 € - 50 000 €", "50 000 € - 100 000 €", "+ de 100 000 €"). |
|             | Prix_cotisation_mensuel | INT             |        |                | Montant mensuel de la cotisation d'assurance.  |
|             | Code_dep_code_commune   | VARCHAR         | 100    | Clé primaire   | Concaténation du code département et code commune pour avoir une clé unique  |
|             | reg_code                | INT             |        |                | Code identifiant une région géographique en France.  |
|             | reg_nom                 | VARCHAR         | 100    |                | Nom complet de la région géographique en France.   |
|             | aca_nom                 | VARCHAR         | 100    |                | Nom de l'académie associée à la région.  |
|             | dep_nom                 | VARCHAR         | 100    |                | Nom complet du département géographique en France.   |
|             | com_nom_maj_court       | VARCHAR         | 100    |                | Nom de la commune en majuscules et sous une forme abrégée.   |
|             | dep_code                | VARCHAR         | 5      |                | Code identifiant un département géographique en France.  |
|             | dep_nom_num             | VARCHAR         | 100    |                | Nom numérique du département   |

Il est important de créer un dictionnaire des données détaillé qui documente toutes ces informations pour assurer une compréhension claire de la structure de la base de données et garantir l'intégrité et la qualité des données.

# Schéma relationnel normalisé en 3NF



01

## Définition

Le schéma relationnel en troisième forme normale (3NF) définit la façon dont les données sont organisées et stockées.

02

## Recherche des typologies de données

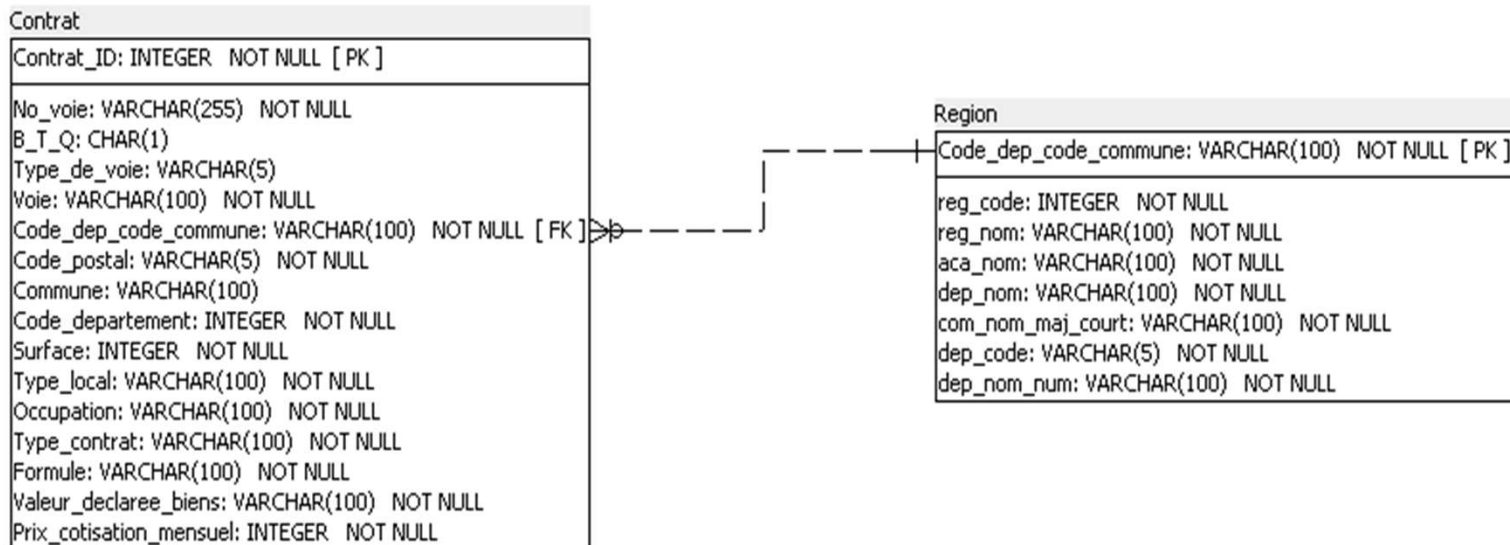
Pour respecter la 3NF, chaque colonne doit dépendre uniquement de la clé principale de la table, et non d'une autre colonne non clé. En respectant la 3NF, on s'assure d'une conception de base de données efficace, évitant la redondance excessive et garantissant l'intégrité des données.

03

## Utilisation du logiciel SQL POWER ARCHITECT

Ce logiciel sera utilisé pour faire correspondre le schéma relationnel normalisé avec le dictionnaire des données.

# Schéma relationnel normalisé en 3NF



Le logiciel permet de générer le code SQL qui permet de créer la base de données directement dans Mysql.

# Création et chargement des données

01

Installation du logiciel MySQL

02

Création de la base de données Projet3

03

Création des 2 tables Contrat et Région

04

Chargement des données des 2 tables

05

Création de la jointure entre les 2 tables

```
mysql> SELECT COUNT(*) AS nombre_de_lignes  
-> FROM contrat;  
+-----+  
| nombre_de_lignes |  
+-----+  
|          30335 |  
+-----+  
1 row in set (0.03 sec)
```

```
mysql> SELECT COUNT(*) AS nombre_de_lignes_region  
-> FROM region;  
+-----+  
| nombre_de_lignes_region |  
+-----+  
|          38916 |  
+-----+  
1 row in set (0.01 sec)
```

# Requêtes SQL

**Requête 1 :** Lister les numéros de contrats (contrat\_ID) avec leur surface pour la commune de Caen.

```
mysql> SELECT contrat_ID, surface FROM contrat
-> WHERE commune = 'caen';
```

| contrat_ID | surface |
|------------|---------|
| 103791     | 35      |
| 103792     | 99      |
| 103793     | 40      |
| 103794     | 20      |

4 rows in set (0.02 sec)

**Requête 2 :** Lister les numéros de contrats avec le type de contrat et leur formule pour les maisons du département 71.

```
mysql> SELECT contrat_ID, Type_contrat, formule
-> FROM contrat
-> WHERE Type_local = 'maison' AND code_departement = '71';
```

| contrat_ID | Type_contrat         | formule   |
|------------|----------------------|-----------|
| 114768     | Residence principale | Integral  |
| 114779     | Residence principale | Classique |
| 114782     | Residence principale | Classique |
| 114812     | Residence principale | Integral  |

4 rows in set (0.03 sec)

**Requête 3 :** Lister le nom des régions de France.

```
mysql> SELECT DISTINCT reg_nom
-> FROM region;
```

| reg_nom                    |
|----------------------------|
| Grand Est                  |
| Auvergne-Rhône-Alpes       |
| Occitanie                  |
| Provence-Alpes-Côte d'Azur |
| Normandie                  |
| Nouvelle-Aquitaine         |
| Centre-Val de Loire        |
| Hauts-de-France            |
| Bourgogne-Franche-Comté    |
| Bretagne                   |
| Corse                      |
| Pays de la Loire           |
| Ile-de-France              |
| Guadeloupe                 |
| Martinique                 |
| Guyane                     |
| La Réunion                 |
| Collectivités d'outre-mer  |
| Mayotte                    |

19 rows in set (0.04 sec)



# Requêtes SQL

**Requête 4 :** combien existe-t-il de contrats sur les résidences principales ?

```
mysql> SELECT COUNT(*) AS Contrat_ID
-> FROM Contrat
-> WHERE Type_contrat = 'residence principale';
```

| Contrat_ID |
|------------|
| 25620      |

1 row in set (0.03 sec)

**Requête 5 :** Quelle est la surface moyenne des logements avec un contrat à Paris ?

```
mysql> SELECT ROUND(AVG(Surface), 1) AS surface_moyenne_arrondie
-> FROM Contrat
-> WHERE Commune LIKE 'Paris%';
```

| surface_moyenne_arrondie |
|--------------------------|
| 50.7                     |

1 row in set (0.02 sec)

**Requête 6 :** Quels sont les 5 contrats qui ont les surfaces les plus élevées ?

```
mysql> SELECT Contrat_ID, Surface
-> FROM Contrat
-> ORDER BY Surface DESC
-> LIMIT 5;
```

| Contrat_ID | Surface |
|------------|---------|
| 104211     | 815     |
| 105463     | 742     |
| 130878     | 595     |
| 100822     | 570     |
| 109872     | 559     |

5 rows in set (0.02 sec)

**Requête 7 :** Quel est le prix moyen de la cotisation mensuelle ?

```
mysql> SELECT ROUND(AVG(Prix_cotisation_mensuel), 2) AS prix_moyen_cotisation
-> FROM Contrat;
```

| prix_moyen_cotisation |
|-----------------------|
| 19.33                 |

1 row in set (0.01 sec)

# Requêtes SQL

**Requête 8 :** Quel est le nombre de contrats pour chaque catégorie de prix de la valeur déclarée des biens ?

```
mysql> SELECT
-> CASE
-> WHEN Valeur_declaree_biens >= 0 AND Valeur_declaree_biens < 25000 THEN '0-24999'
-> WHEN Valeur_declaree_biens >= 25000 AND Valeur_declaree_biens < 50000 THEN '25000-49999'
-> WHEN Valeur_declaree_biens >= 50000 AND Valeur_declaree_biens < 100000 THEN '50000-99999'
-> WHEN Valeur_declaree_biens >= 100000 THEN '100000+'
-> END AS categorie_prix,
-> COUNT(*) AS nombre_de_contrats
-> FROM Contrat
-> GROUP BY categorie_prix
-> ORDER BY categorie_prix <> '100000+', MAX(Valeur_declaree_biens) DESC;
```

| categorie_prix | nombre_de_contrats |
|----------------|--------------------|
| 100000+        | 104                |
| 50000-99999    | 696                |
| 25000-49999    | 6815               |
| 0-24999        | 22720              |

4 rows in set, 65535 warnings (0.07 sec)

**Requête 9 :** Quel est le classement des 10 départements où le prix moyen de la cotisation est le plus élevé ?

```
mysql> SELECT r.dep_nom AS nom_departement, ROUND(AVG(c.Prix_cotisation_mensuel), 2) AS prix_moyen_cotisation
-> FROM Contrat c
-> JOIN Region r ON c.Code_dep_code_commune = r.Code_dep_code_commune
-> GROUP BY r.dep_nom
-> ORDER BY prix_moyen_cotisation DESC
-> LIMIT 10;
```

| nom_departement   | prix_moyen_cotisation |
|-------------------|-----------------------|
| Paris             | 35.34                 |
| Hauts-de-Seine    | 26.49                 |
| Allier            | 20.00                 |
| Val-de-Marne      | 19.73                 |
| Rhône             | 18.96                 |
| Vosges            | 18.10                 |
| Alpes-Maritimes   | 18.12                 |
| Als               | 17.95                 |
| Charente-Maritime | 17.30                 |
| La Réunion        | 17.25                 |

10 rows in set (0.07 sec)

**Requête 10 :** Quel est le nombre de contrats avec des formules "Intégral" pour la région Pays de la Loire ?

```
mysql> SELECT COUNT(*) AS nombre_contrats_integral
-> FROM Contrat c
-> JOIN Region r ON c.Code_dep_code_commune = r.Code_dep_code_commune
-> WHERE r.reg_nom = 'Pays de la Loire'
-> AND c.Formule = 'Intégral';
```

| nombre_contrats_integral |
|--------------------------|
| 561                      |

1 row in set (0.04 sec)

**Requête 11 :** Quelle est la liste des communes ayant au moins 150 contrats ?

```
mysql> SELECT
-> 'Paris' AS nom_commune,
-> COUNT(*) AS total_contrats
-> FROM Contrat c
-> JOIN Region r ON c.Code_dep_code_commune = r.Code_dep_code_commune
-> WHERE r.dep_nom = 'Paris'
-> GROUP BY nom_commune
-> HAVING total_contrats >= 150
-> UNION
-> SELECT
-> r.com_nom_maj_court AS nom_commune,
-> COUNT(*) AS total_contrats
-> FROM Contrat c
-> JOIN Region r ON c.Code_dep_code_commune = r.Code_dep_code_commune
-> WHERE r.dep_nom != 'Paris'
-> GROUP BY nom_commune
-> HAVING total_contrats >= 150;
```

| nom_commune | total_contrats |
|-------------|----------------|
| Paris       | 5198           |
| TOULOUSE    | 187            |
| BORDEAUX    | 302            |
| GRENOBLE    | 220            |
| NANTES      | 291            |
| LILLE       | 161            |
| NICE        | 387            |
| TOULON      | 170            |
| COURBEVOIE  | 163            |

9 rows in set (0.06 sec)

**Requête 12 :** Quel est le nombre de contrats pour chaque région ?

```
mysql> SELECT
-> r.reg_nom AS nom_region,
-> COUNT(*) AS nombre_contrats
-> FROM Contrat c
-> JOIN Region r ON c.Code_dep_code_commune = r.Code_dep_code_commune
-> GROUP BY nom_region
-> ORDER BY nombre_contrats DESC;
```

| nom_region                 | nombre_contrats |
|----------------------------|-----------------|
| Ile-de-France              | 13474           |
| Provence-Alpes-Côte d'Azur | 3287            |
| Auvergne-Rhône-Alpes       | 2972            |
| Nouvelle-Aquitaine         | 2097            |
| Occitanie                  | 1837            |
| Hauts-de-France            | 1327            |
| Pays de la Loire           | 1138            |
| Bretagne                   | 945             |
| Normandie                  | 898             |
| Grand Est                  | 806             |
| Centre-Val de Loire        | 804             |
| Bourgogne-Franche-Comté    | 402             |
| Corse                      | 247             |
| Martinique                 | 60              |
| Guyane                     | 37              |
| La Réunion                 | 4               |

16 rows in set (0.04 sec)

# Bilan du projet



01

Installation du logiciel MySQL très difficile

02

Chargement des données chronophage en raison de données incohérentes (encoding, utilisation notepad++...)

03

Orientation vers de mauvaises solutions lors des recherches sur Google, Stackoverflow ou ChatGPT...



04

Parti de zéro en matière de connaissance technique sur SQL

05

Finalement, réussite du projet et maîtrise du logiciel SQL



## Conclusion

Ce projet m'aura permis de maîtriser SQL, un puissant outil de gestion de bases de données relationnelles.

Mais cela m'aura surtout permis d'appréhender le métier de programmeur, où celui-ci doit souvent solutionner seul, des problèmes informatiques complexes et chronophages, à l'aide de moteurs de recherche sur internet.

