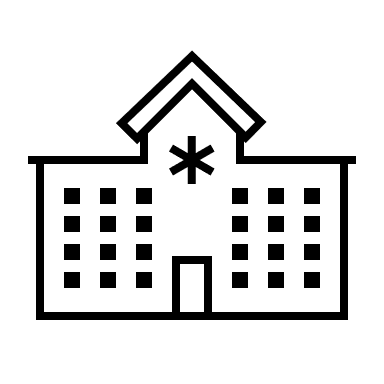
****

**Marché des assurances habitation**

**Document technique**

**2024**

*Julien Séré*

**SOMMAIRE :**

1. **Exploration du type de données**
2. **Types de données et contraintes**
3. **Nettoyage des données**
4. **Schéma relationnel**
5. **Code SQL**
6. **Base de données MySQL**
7. **Chargement de la base de données**
8. **Quelques requêtes SQL…**
9. **Exploration du type de données**
10. **Types de données et contraintes**

Les données ont été répertoriées et renseignées pour leurs types ainsi que leurs tailles.

Les types de données sont les suivantes :

**INTEGER :** entier long

**CHAR :** valeur alpha de longueur fixe

**VARCHAR :** valeur alpha de longueur maximale fixée (Les longueurs maximales sont ainsi renseignées dans la colonne taille

Ensuite, il est question de définir les clés primaires de chaque table :

Client\_ID : Clé primaire de la table contrat

Code\_dep\_code\_commune : Clé primaire de la table région

Code\_dep\_code\_commune est donc une clé étrangère dans la table contrat

Voici le dictionnaire actualisé des données :

Une image contenant texte, capture d’écran, logiciel, nombre

Description générée automatiquement

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Nom des colonnes** | **Type de données** | **Taille** | **Clé** | **Description** |
| **CONTRAT** | Contrat\_ID | INT |  | PRIMAIRE | Id unique pour les contrats |
|  | No\_voie | INT (NULL) |  |  | 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 | 100 |  | Libellé de la voie pour l'adresse du logement assuré |
|  | Code\_dep\_code\_commune | VARCHAR | 10 | ETRANGER | Concaténation du code département et code commune pour avoir une clé unique |
|  | Code\_postal | VARCHAR | 5 |  | Code postal pour l'adresse du logement assuré |
|  | Surface | INT |  |  | Surface du logement assuré |
|  | Type\_Local | VARCHAR | 15 |  | Type de logement assuré : maison ou appartement |
|  | Occupation | VARCHAR | 15 |  | Type de contrat : Propriétaire ou locataire |
|  | Type\_contrat | VARCHAR | 20 |  | Type de résidence : Résidence principale ou secondaire |
|  | Formule | VARCHAR | 10 |  | Formule du contrat d'assurance |
|  | Valeur\_declaree\_biens | VARCHAR | 20 |  | Valeur déclarée du logement assuré |
|  | Prix\_cotisation\_mensuel | INT |  |  | Valeur de la cotisation mensuelle client |
| **REGION** | Code\_dep\_code\_commune | VARCHAR | 10 | Clé primaire | Concaténation du code département et code commune pour avoir une clé unique |
|  | reg\_code | VARCHAR | 3 |  | Code régional |
|  | reg\_nom | VARCHAR | 50 |  | Nom de la région |
|  | aca\_nom | VARCHAR | 50 |  | Nom de la métropole |
|  | dep\_nom | VARCHAR | 50 |  | Nom du département |
|  | com\_nom\_maj\_court | VARCHAR | 50 |  | Nom de la commune |
|  | dep\_code | VARCHAR | 4 |  | Code départemental |
|  | dep\_nom\_num | VARCHAR | 50 |  | Concaténation du nom du département et code département |

Pour certaines colonnes, comme le N°Voie, il est possible également d’utiliser un type VARCHAR puisque le numéro de Voie sert à identifier une adresse et n’est pas destiné à une quelconque opération. Cependant, le dictionnaire étant déjà rempli par un type INTEGER, nous continuons avec un type INTEGER, en prenant soin d’attribuer la valeur NULL à toutes les cellules vides la colonne correspondante.

1. **Nettoyage des données**

Après avoir défini les types de données qui seront traitées. Il est nécessaire de passer en revue les tableurs de données afin de déceler des possibles erreurs, des cellules vides qui pourraient poser problèmes, ou enfin des valeurs manquantes.

On remarque lors des imports que l’import de la table CONTRAT comprend des erreurs liées aux contrats passés dans le département de La Réunion. En effet, des codes postaux ont été renseignés dans les Code\_dep\_commune ce qui a conduit à une erreur lors de l’identification à la clé primaire de la table Région (données partiellement inexistantes). Ainsi, il convient de veiller à corriger les données erronées en se référant aux codes communes de l’INSEE disponibles en ligne, ou bien en utilisant la table Région pour laquelle les codes sont renseignés.

1. **Schéma relationnel**

A l’aide de l’outil SQL Power Architect et du dictionnaire des données, nous pouvons déduire et construire le schéma relationnel des tables des données. Nous veillons bien à renseigner les contraintes pour chaque colonne et renseigner si celle-ci peut accepter les valeurs nulles.

Nous avons vu dans les données que la colonne N° voie contient des cellules vides, il convient alors de renseigner que cette colonne doit nécessairement accepter les valeurs nulles pour que l’import soit un succès.

Une image contenant texte, Police, reçu, capture d’écran

Description générée automatiquement

Schéma relationnel généré par Power A

L’association de tables est de type un à plusieurs puisqu’une région peut être associée à plusieurs contrats mais un contrat ne peut être associé qu’à une seule région.

Le code SQL peut donc être généré afin d’importer l’architecture des tables dans notre SGBD.

Pour ce projet, nous travaillons avec MySQL Workbench sur la database openclassrooms.

1. **Code SQL**

Voici le code SQL permettant de créer les tables et associations dans notre SGBD MySQL Workbench :

CREATE TABLE Region (

Code\_dep\_code\_commune VARCHAR(10) NOT NULL,

reg\_code VARCHAR(3) NOT NULL,

reg\_nom VARCHAR(50) NOT NULL,

aca\_nom VARCHAR(50) NOT NULL,

dep\_nom VARCHAR(50) NOT NULL,

com\_nom\_maj\_court VARCHAR(50) NOT NULL,

dep\_code VARCHAR(3) NOT NULL,

dep\_nom\_num VARCHAR(50) NOT NULL,

PRIMARY KEY (Code\_dep\_code\_commune)

);

CREATE TABLE Contrat (

Contrat\_ID INT NOT NULL,

No\_voie VARCHAR (3) NOT NULL

B\_T\_Q VARCHAR(1) NOT NULL,

Type\_de\_voie VARCHAR(4) NOT NULL,

Voie VARCHAR(100) DEFAULT 26 NOT NULL,

Code\_dep\_code\_commune\_contrat VARCHAR(10) NOT NULL,

Code\_postal VARCHAR(5) NOT NULL,

Surface VARCHAR(4) NOT NULL,

Type\_Local VARCHAR(15) NOT NULL,

Occupation VARCHAR(15) NOT NULL,

Type\_contrat VARCHAR(20) NOT NULL,

Formule VARCHAR(10) NOT NULL,

Valeur\_declaree\_biens VARCHAR(20) NOT NULL,

Prix\_cotisation\_mensuel INT NOT NULL,

PRIMARY KEY (Contrat\_ID)

);

ALTER TABLE Contrat ADD CONSTRAINT region\_contrat\_fk

FOREIGN KEY (Code\_dep\_code\_commune\_contrat)

REFERENCES Region (Code\_dep\_code\_commune)

ON DELETE NO ACTION

ON UPDATE NO ACTION;

1. **Base de données MySQL**
2. **Chargement de la base de données**

La base de données MySQL est désormais configuré et prête à l’emploi.

Les données sont chargées à l’aide de l’outil d’import fourni dans MySQL WorkBench « Table Import Data Wizard ».

Cet outil nous permet d’attribuer aisément les bonnes colonnes avec les bonnes données dans le tableur.

L’import peut également se faire via un code SQL dédié à l’importation :

LOAD DATA INFILE '/chemin/vers/monfichier.csv'

INTO TABLE nom\_de\_la\_table

FIELDS TERMINATED BY ',' -- Séparateur de colonnes (virgule ici)

ENCLOSED BY '"' -- Si les valeurs sont entourées de guillemets

LINES TERMINATED BY '\n' -- Séparateur de lignes (nouvelle ligne ici)

IGNORE 1 ROWS;

Nous voici avec les bases de données chargées :

Une image contenant texte, capture d’écran, logiciel

Description générée automatiquement

Une image contenant capture d’écran, texte

Description générée automatiquement

Nous avons ici la certitude que toutes les lignes ont bien été importé avec la fonction COUNT. Pour pouvoir manipuler et analyser les données, il peut être plus pratique de générer un aperçu du contenu des tables. Il est possible de générer un aperçu des lignes présentes dans chaque table avec la requête suivante :

SELECT \* FROM contrat

SELECT \* FROM region

Une image contenant texte, capture d’écran, nombre, Police

Description générée automatiquement

Aperçu de la table contrat

Une image contenant texte, capture d’écran, nombre

Description générée automatiquement

Aperçu de la table région

1. **Quelques requêtes SQL…**

Le but d'une requête SQL est de permettre l'interaction avec une base de données pour effectuer diverses opérations, telles que la récupération, l'insertion, la mise à jour ou la suppression de données. SQL est le langage standard pour interroger et manipuler des bases de données relationnelles. Ici, nous pouvons mettre en avant des tendances, des éléments forts, ou des éléments faibles concernant le marché de l’assurance sur lequel la société opère.

**Requête 5 : Quel est le prix moyen de la cotisation mensuelle ?**

Une image contenant texte, capture d’écran, ligne

Description générée automatiquement

**Requête 6 : Quel est le nombre de contrats pour chaque catégorie de prix de la valeur déclarée des biens ?**

Une image contenant texte, Police, logiciel, nombre

Description générée automatiquement