



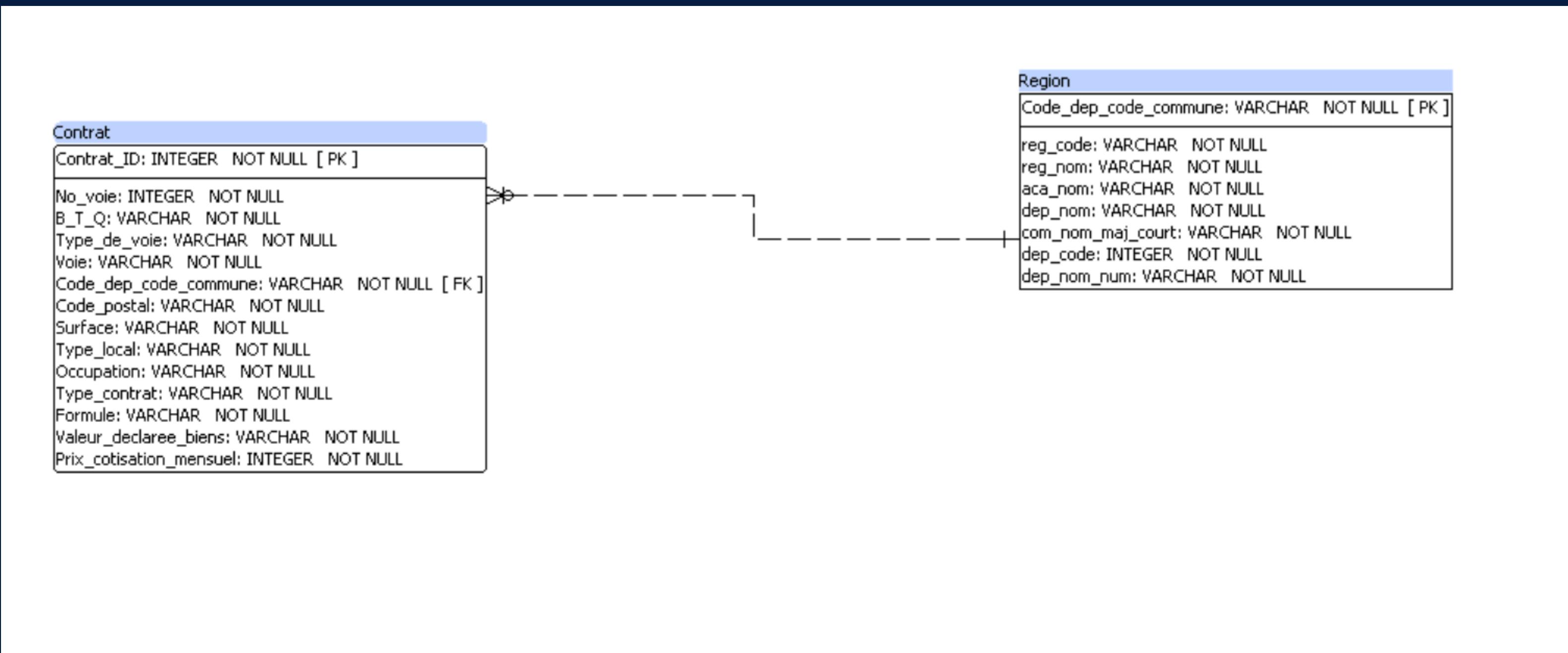
# SUPPORT DE PRÉSENTATION DU PROJET 3: REQUÊTEZ UNE BASE DE DONNÉES AVEC SQL

Présenté par : Aïda Gassama

# 1ère étape: Crédit à la création du Dictionnaire des données

	Nom des colonnes	Type de données	Taille	Clé	Description
CONTRAT.CSV	Contrat_ID	INTEGER		Clé primaire	Id unique pour les contrats
	No_voie	INTEGER			Numéro dans la voie pour l'adresse du logement assuré
	B_T_Q	VARCHAR	1		Indicateur éventuel de répétition pour l'adresse du logement assuré sur un caractère
	Type_de_voie	VARCHAR	15		Type de voie pour l'adresse du logement assuré: rue, av (Avenue), rte (Route), ...
	Voie	VARCHAR	50		Libellé de la voie pour l'adresse du logement assuré
	Code_dep_code_commune	VARCHAR	5	Clé secondaire	Concaténation du code département et code commune pour avoir une clé unique
	Code_postal	VARCHAR	10		Code postal pour l'adresse du logement assuré
	Surface	INTEGER			Superficie du logement en m²
	Type_local	VARCHAR	15		Type de logement
	Occupation	VARCHAR	15		Statut d'occupation du logement de l'utilisateur : locataire ou propriétaire
	Type_contrat	VARCHAR	50		Type de contrat
	Formule	VARCHAR	15		Type de formule ou d'offre choisie
REGION.CSV	Valeur_declaree_biens	VARCHAR	50		Plage de valeur déclarée des biens en euros, indiquée par des intervalles
	Prix_cotisation_mensuel	INTEGER			Montant mensuel de la cotisation en euros
	Code_dep_code_commune	VARCHAR	15	Clé étrangère	Concaténation du code département et code commune pour avoir une clé unique
	reg_code	VARCHAR	2		Code unique de 2 chiffres représentant l'identifiant d'une région
	reg_nom	VARCHAR	50		Nom de la région
	aca_nom	VARCHAR	50		Nom de la ville
	dep_nom	VARCHAR	50		Nom du département
	com_nom_maj_court	VARCHAR	50		Nom de la commune en majuscules abrégé ou simplifié
	dep_code	INTEGER			Code numérique du département, représenté par un entier
	dep_nom_num	VARCHAR	50		Nom du département suivi de son code entre parenthèses

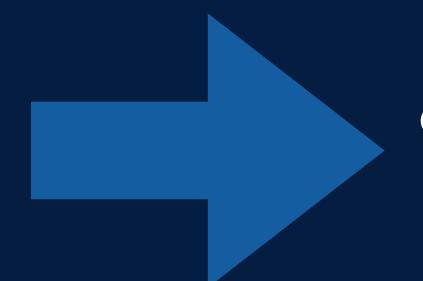
# 2ème étape Le Schéma Relationnel



# 3ème étape: Chargement de la base de données dans MySQL

The screenshot shows the MySQL Workbench interface with the 'region\_csv' tab selected. The query editor contains the SQL command: `SELECT * FROM client.region_csv;`. The result grid displays data from the 'region\_csv' table, which includes columns such as 'Code\_dep\_code\_communne', 'reg\_code', 'reg\_nom', 'aca\_nom', 'dep\_nom', 'com\_nom\_maj\_court', 'dep\_code', and 'dep\_nom\_num'. The data shows various regions and their corresponding codes and names, such as Grand Est, Reims, Marne, etc.

The screenshot shows the MySQL Workbench interface with the 'contrat\_csv' tab selected. The query editor contains the SQL command: `SELECT * FROM client.contrat_csv;`. The result grid displays data from the 'contrat\_csv' table, which includes columns like 'Contrat\_ID', 'No\_voie', 'B\_T\_Q', 'Type\_de\_voie', 'Voie', 'Code\_dep\_code\_communne', 'Code\_postal', 'Surface', 'Type\_local', 'Occupation', 'Type\_contrat', 'Formule', and 'Valeur\_de'. The data represents various contracts with details like address, type, and value.



- Crédation de la base de données puis chargement de la base de données à partir du fichier CSV sur MySQL
- Crédation des 2 tables: Region\_csv et Contrat\_csv

# 4ème étape: Les requêtes sur MySQL

The screenshot shows the MySQL Workbench interface with a query editor and a results grid. The query is:

```
1 •  SELECT Contrat_ID, Surface
2   FROM Contrat_csv
3  WHERE Code_postal = 14000;
```

The results grid displays the following data:

Contrat_ID	Surface
103792	99
103794	20
103791	35
103793	40

## Requête 1:

Lister les numéros de contrats (Contrat\_ID) avec leur surface pour la commune de Caen.

- **SELECT Contrat\_ID, Surface** : La requête sélectionne deux colonnes de la table : Contrat\_ID et Surface
- **FROM Contrat\_csv** : Indique que les données doivent être extraites de la table Contrat\_csv
- **WHERE Code\_postal = 14000** : La requête filtre les résultats pour ne retourner que les lignes où la valeur de la colonne Code\_postal est égale à 14000 (Caen)

The screenshot shows the MySQL Workbench interface with a query editor and a results grid. The query is:

```
1 •  SELECT Contrat_ID, Type_contrat, Formule
2   FROM contrat_csv
3  WHERE Type_local = 'Maison' AND SUBSTRING(Code_dep_code_commune, 1, 2) = '71';
```

The results grid displays the following data:

Contrat_ID	Type_contrat	Formule
102358	Residence principale	Classique
102395	Residence principale	Classique
114779	Residence principale	Classique
114782	Residence principale	Classique
102385	Residence secondaire	Integral
102411	Residence principale	Integral
114768	Residence principale	Integral
114812	Residence principale	Integral

## 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** : Permet de selectionner 3 colonnes de la table contrat\_csv
- **FROM contrat\_csv** : Indique que les données doivent être extraites de la table contrat\_csv.
- **WHERE Type\_local = 'Maison'** : La requête filtre les résultats pour ne retourner que les lignes où Type\_local est 'Maison'
- **AND SUBSTRING(Code\_dep\_code\_commune, 1, 2) = '71'** : Cette partie de la condition filtre les résultats pour ne retourner que les lignes où les deux premiers caractères de la colonne Code\_dep\_code\_commune sont '71'. Cela correspond au département 71 (Saône-et-Loire en France)

# Les requêtes sur MySQL

The screenshot shows the MySQL Workbench interface with a query editor and a results grid. The query is:

```
1 • SELECT DISTINCT r.reg_nom
2   FROM contrat_csv c
3   JOIN region_csv r ON substring(c.Code_dep_code_commune,1, 2) = r.dep_code;
```

The results grid displays the names of French regions:

reg_nom
Bourgogne-Franche-Comté
Bretagne
Auvergne-Rhône-Alpes
Pays de la Loire
Hauts-de-France
Ile-de-France
Corse

The screenshot shows the MySQL Workbench interface with a query editor and a results grid. The query is:

```
1 • SELECT Contrat_ID, Surface
2   FROM contrat_csv
3   ORDER BY CAST(Surface AS DECIMAL) DESC
4   LIMIT 5;
```

The results grid displays the top 5 contracts by surface area:

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

## Requête 3:

Lister le nom des régions de France

- **SELECT DISTINCT r.reg\_nom** : La requête sélectionne la colonne reg\_nom (nom de la région) de la table region\_csv. DISTINCT garantit que chaque région n'apparaît qu'une seule fois dans les résultats, même si elle est associée à plusieurs contrats
- **FROM contrat\_csv c** : La table principale utilisée est contrat\_csv, à laquelle on donne l'alias c pour simplifier les références
- **JOIN region\_csv r ON SUBSTRING (c.Code\_dep\_code\_commune,1, 2) = r.dep\_code** : Une jointure est effectuée avec la table region\_csv (r). Elle est basée sur les deux premiers caractères de la colonne Code\_dep\_code\_commune dans contrat\_csv, qui représentent le code départemental

## Requête 4 :

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

- **SELECT Contrat\_ID, Surface** : La requête sélectionne deux colonnes de la table : Contrat\_ID et Surface
- **FROM Contrat\_csv** : Indique que les données doivent être extraites de la table Contrat\_csv
- **ORDER BY CAST(Surface AS DECIMAL) DESK** : ORDER BY trie les résultats en fonction de la colonne Surface
- **CAST(Surface AS DECIMAL)** convertit la colonne Surface en un type numérique (DECIMAL) pour s'assurer que le tri se fait correctement
- **DESC** indique que le tri est en ordre décroissant (du plus grand au plus petit)
- **LIMIT 5** : Restreint les résultats aux 5 premiers enregistrements après le tri

# Les requêtes sur MySQL

The screenshot shows the MySQL Workbench interface. The SQL editor contains the following query:

```
1 •   SELECT AVG(CAST(prix_cotisation_mensuel AS DECIMAL)) AS Prix_moyen_cotisation
2   FROM contrat_csv;
```

The results grid shows one row with the value 19.3287.

The screenshot shows the MySQL Workbench interface. The SQL editor contains the following query:

```
1 •   SELECT
2     CASE
3       WHEN valeur_declaree_biens BETWEEN 0 AND 25000 THEN '0 - 25 000 €'
4       WHEN valeur_declaree_biens BETWEEN 25001 AND 50000 THEN '25 001 - 50 000 €'
5       ELSE '50 001 € et plus'
6     END AS categorie_prix,
7     COUNT(*) AS Nombre_contrats
8   FROM contrat_csv
9   GROUP BY categorie_prix;
```

The results grid shows three categories with their respective contract counts:

categorie_prix	Nombre_contrats
0 - 25 000 €	29535
25 001 - 50 000 €	696
50 001 € et plus	104

## Requête 5 :

Quel est le prix moyen de la cotisation mensuelle ?

- **SELECT** : permet de sélectionner des données
- **AVG** : est une fonction d'agrégation qui calcule la moyenne des valeurs dans une colonne
- **AS Prix\_moyen\_cotisation**: donne un alias à la colonne qui va s'afficher
- **FROM** : Spécifie la table à partir de laquelle les données sont sélectionnées. Dans ce cas, la table est nommée **contrat\_csv**

## Requête 6 :

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

- **CASE**: est une expression conditionnelle en SQL qui permet de créer des branches logiques
- **WHEN valeur\_declaree\_biens BETWEEN 0 AND 25000 THEN '0 - 25 000 €'**: Si la valeur de valeur\_declaree\_biens est comprise entre 0 et 25 000, alors elle est catégorisée comme '0 - 25 000 €'
- **ELSE '50 001 € et plus'**: Si la valeur de valeur\_declaree\_biens ne correspond à aucune des plages précédentes, elle est catégorisée comme '50 001 € et plus'.
- **COUNT(\*)** compte le nombre total de lignes (contrats) dans chaque catégorie
- **GROUP BY** est utilisé pour regrouper les résultats par la colonne

# Les requêtes sur MySQL

The screenshot shows the MySQL Workbench interface with two panes. The top pane displays the SQL query:

```
1 SELECT COUNT(*) AS Nombre_formules_integral
2 FROM contrat_csv c
3 JOIN region_csv r ON c.Code_dep_code_commune = r.Code_dep_code_commune
4 WHERE r.reg_nom = 'Pays de la Loire' AND c.Formule = 'Integral';
```

The bottom pane shows the result grid with one row:

Nombre_formules_integral
589

The title bar of the window is "Result 7".

## Requête 7 :

Quel est le nombre de formules “integral” sur la région Pays de la Loire ?

The screenshot shows the MySQL Workbench interface with two panes. The top pane displays the SQL query:

```
1 • SELECT Contrat_ID, Type_contrat, Formule
2 FROM contrat_csv
3 WHERE Type_local = 'Maison' AND Code_dep_code_commune LIKE '71%';
```

The bottom pane shows the result grid with five rows:

Contrat_ID	Type_contrat	Formule
102358	Residence principale	Classique
102395	Residence principale	Classique
114779	Residence principale	Classique
114782	Residence principale	Classique

The title bar of the window is "contrat csv 8".

## Requête 8 :

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

# Les requêtes sur MySQL

The screenshot shows the MySQL Workbench interface. At the top, there's a toolbar with various icons. Below it, a query editor window displays the following SQL code:

```
1 • SELECT AVG(CAST(Surface AS DECIMAL)) AS Surface_moyenne  
2   FROM contrat_csv  
3 WHERE Code_postal LIKE '75%';
```

Below the query editor is a results grid titled "Result Grid". It contains one row with the column header "Surface\_moyenne" and the value "51.8232".

The screenshot shows the MySQL Workbench interface. At the top, there's a toolbar with various icons. Below it, a query editor window displays the following SQL code:

```
2 • SELECT  
3     LEFT(Code_dep_code_commune, 2) AS Departement,  
4     ROUND(AVG(Prix_cotisation_mensuel), 2) AS Prix_cotisation_mensuel  
5   FROM contrat_csv  
6  GROUP BY Departement  
7 ORDER BY Prix_cotisation_mensuel DESC  
8 LIMIT 10;
```

Below the query editor is a results grid titled "Result Grid". It has two columns: "Departement" and "Prix\_cotisation\_mensuel". The data is as follows:

Departement	Prix_cotisation_mensuel
75	36.4
92	26.24
94	19.82
78	18.89
69	18.49
17	17.32
74	17.15
2A	17.07
12	17
60	16.95

## Requête 9 :

Quelle est la surface moyenne des contrats à Paris ?

## Requête 10 :

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

# Les requêtes sur MySQL

The screenshot shows the MySQL Workbench interface with a query editor and a result grid. The query is:

```
1 • SELECT r.com_nom_maj_court AS Commune, COUNT(*) AS Nombre_contrats
2   FROM contrat_csv c
3   JOIN region_csv r ON c.Code_dep_code_commune = r.Code_dep_code_commune
4   GROUP BY r.com_nom_maj_court
5   HAVING COUNT(*) >= 150;
```

The result grid displays the following data:

Commune	Nombre_contrats
NICE	387
TOULOUSE	187
BORDEAUX	302
GRENOBLE	220
NANTES	291

## Requête 11 :

Liste des communes ayant eu au moins 150 contrats

The screenshot shows the MySQL Workbench interface with a query editor and a result grid. The query is:

```
1 • SELECT r.reg_nom AS Region, COUNT(*) AS Nombre_Contrats
2   FROM Contrat_csv c
3   JOIN region_csv r ON c.Code_dep_code_commune = r.Code_dep_code_commune
4   GROUP BY r.reg_nom;
```

The result grid displays the following data:

Region	Nombre_Contrats
Nouvelle-Aquitaine	2038
Centre-Val de Loire	598
Bourgogne-Franche-Comté	293
Bretagne	947
Pays de la Loire	1196
Ile-de-France	14177
Martinique	73

## Requête 12 :

Quel est le nombre de contrats pour chaque région ?