

# Requête n° 1 :

	Besoin	Requête	Résultat
1	Lister les numéros de contrats (contrat_ID) avec leur surface pour la commune de Caen	<pre>SELECT contrat_id AS ID_Contrat, surface AS Surface FROM contrats c WHERE commune = 'CAEN';</pre>	<pre># ID_Contrat Surface 1 103791 35 2 103792 99 3 103793 40 4 103794 20</pre>

**SELECT contrat\_id, surface** : Définition de la projection. On spécifie les colonnes que l'on souhaite inclure dans le résultat. Dans ce cas, nous voulons les valeurs de la colonne "contrat\_id" et de la colonne "surface". L'utilisation des alias AS "ID\_Contrat" et AS "Surface" attribuent un nom convivial aux colonnes résultantes.

**FROM contrats c** : Cette partie indique la table à partir de laquelle nous voulons récupérer les données. On utilise l'alias "c" pour la table "contrats", ce qui permet de se référer à cette table en utilisant cet alias tout au long de la requête.

**WHERE commune = 'CAEN'** : Ajout d'une restriction à la requête. On demande à la requête de ne récupérer que les enregistrements où la valeur de la colonne "commune" est égale à 'CAEN'. Cela signifie que nous obtiendrons uniquement les contrats associés à la commune de CAEN.

## Requête n° 2 :

	Besoin	Requête	Résultat
2	Lister les numéros de contrats (contrat_ID), avec le type de contrat et leur formule pour les maisons du département de la Saône-et-Loire (Département 71)	<pre>SELECT contrat_id AS ID_Contrat, type_contrat AS Type_de_contrat, formule AS Formule FROM contrats c WHERE type_local = 'Maison' AND code_departement = 71;</pre>	<pre># ID_Contrat Type_de_contrat Formule 1 114768 Residence principale Integral 2 114782 Residence principale Classique 3 114812 Residence principale Integral 4 114779 Residence principale Classique</pre>

**SELECT contrat\_id, type\_contrat, formule** : Définition de la projection. On spécifie les colonnes que l'on souhaite inclure dans le résultat. Dans ce cas, nous voulons les valeurs des colonnes "contrat\_id", "type\_contrat" et "formule".

**FROM contrats c** : Cette partie indique la table à partir de laquelle nous voulons récupérer les données. On utilise l'alias "c" pour la table "contrats".

**WHERE type\_local = 'Maison' AND code\_departement = 71** : Ajout d'une restriction à la requête. On spécifie deux conditions liées par l'opérateur logique "AND". La première condition demande à la requête de ne sélectionner que les enregistrements où la colonne "type\_local" a la valeur 'Maison'. La deuxième condition demande à la requête de ne sélectionner que les enregistrements où la colonne "code\_departement" a la valeur 71.

## Requête n° 3 :

	Besoin	Requête	Résultat
3	Lister le nom des régions de France	<i>SELECT DISTINCT reg_nom AS Nom_des_regions FROM regions;</i>	# Nom_des_regions 1 Auvergne-RhôneAlpes 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 Corse 10 Bourgogne-FrancheComté 11 Bretagne 12 Pays de la Loire 13 Île-de-France 14 Guadeloupe 15 Martinique 16 Guyane 17 La Réunion 18 Collectivités d'outremer 19 Mayotte

**SELECT DISTINCT reg\_nom** : Définition de la projection. On spécifie la colonne que l'on souhaite inclure dans le résultat. Dans ce cas, nous voulons les valeurs distinctes de la colonne "reg\_nom".

**FROM regions** : Cette partie indique la table à partir de laquelle nous voulons récupérer les données. Dans ce cas, nous sélectionnons la table "regions".

**DISTINCT** : L'ajout du mot-clé DISTINCT avant le nom de la colonne indique que nous ne voulons que les valeurs uniques de cette colonne dans le résultat. Si des valeurs identiques apparaissent plusieurs fois dans la colonne elles ne seront incluses qu'une seule fois dans le résultat.

## Requête n° 4 :

	Besoin	Requête	Résultat
4	Combien existe-t-il de contrats sur les résidences principales ?	<pre>SELECT COUNT(contrat_id) AS Nombre_de_contrats FROM contrats WHERE type_contrat = 'Residence principale';</pre>	<pre># Nombre_de_contrats 1 25620</pre>

**SELECT COUNT(contrat\_id)** : On spécifie ici que nous voulons compter le nombre d'enregistrements. Le compte est basé sur la colonne "contrat\_id". L'expression COUNT(contrat\_id) compte le nombre de valeurs non nulles dans la colonne "contrat\_id".

**FROM contrats** : Cette partie indique la table à partir de laquelle nous voulons récupérer les données. Dans ce cas, nous sélectionnons la table "contrats".

**WHERE type\_contrat = 'Residence principale'** : Ajout d'une restriction à la requête. On spécifie que nous voulons uniquement compter les enregistrements où la colonne "type\_contrat" a la valeur 'Residence principale'. Cela signifie que seuls les contrats de résidence principale seront pris en compte dans le comptage.

## Requête n° 5 :

	Besoin	Requête	Résultat
5	Quelle est la surface moyenne des logements avec un contrat à Paris ?	<pre>SELECT AVG(surface) AS Surface_moyenne FROM contrats WHERE code_departement = 75;</pre>	# Surface_moyenne 1 51.77

**SELECT AVG(surface)** : On spécifie ici que nous voulons calculer la moyenne (AVG) des valeurs dans la colonne "surface".

**FROM contrats** : Cette partie indique la table à partir de laquelle nous souhaitons calculer la moyenne. Dans ce cas, nous effectuons le calcul à partir de la table "contrats".

**WHERE code\_departement = 75** : Ajout d'une restriction à la requête. On spécifie que nous voulons uniquement calculer la moyenne pour les enregistrements où la colonne "code\_departement" a la valeur 75. Cela signifie que seuls les contrats du département 75 seront inclus dans le calcul de la moyenne de la surface.

## Requête n° 6 :

	Besoin	Requête	Résultat
6	Quels sont les 5 contrats qui ont les surfaces les plus élevées ?	<pre>SELECT contrat_id AS ID_Contrat, surface AS Surface FROM contrats ORDER BY surface DESC LIMIT 5;</pre>	<pre># ID_Contrat Surface 1 104211 815 2 105463 742 3 130878 595 4 100822 570 5 109872 559</pre>

**SELECT contrat\_id , surface** : Définition de la projection. On spécifie les colonnes que l'on souhaite inclure dans le résultat. Dans ce cas, nous voulons les valeurs des colonnes "contrat\_id" et "surface".

**FROM contrats** : Cette partie indique la table à partir de laquelle nous voulons récupérer les données. Dans ce cas, nous sélectionnons la table "contrats".

**ORDER BY surface DESC** : Cette partie ordonne les résultats par ordre décroissant de la colonne "surface".

**LIMIT 5** : Cette partie limite le résultat à seulement 5 enregistrements. Seuls les 5 premiers enregistrements après le tri par ordre décroissant de la colonne "surface" seront inclus dans le résultat.

## Requête n° 7 :

	Besoin	Requête	Résultat
7	Quel est le prix moyen de la cotisation mensuelle ?	<i>SELECT ROUND(AVG(prix_cotisation_mensuel), 2) AS Cotisation_moyenne FROM contrats</i>	# Cotisation_moyenne 1 19.33

**SELECT ROUND(AVG(prix\_cotisation\_mensuel), 2) :** On spécifie ici que nous voulons calculer la moyenne (AVG) des valeurs dans la colonne "prix\_cotisation\_mensuel", puis arrondir le résultat à deux décimales à l'aide de la fonction ROUND.

**FROM contrats :** Cette partie indique la table à partir de laquelle nous voulons récupérer les données. Dans ce cas, nous sélectionnons la table "contrats".

## Requête n° 8 :

	Besoin	Requête	Résultat
8	Nombre de contrats pour chaque catégorie de prix de la valeur déclarée des biens ?	<pre> SELECT CASE WHEN Valeur_declaree_biens ='0-25000' THEN '0-25000' WHEN Valeur_declaree_biens ='25000-50000' THEN '25000-50000' WHEN Valeur_declaree_biens ='50000-100000' THEN '50000-100000' WHEN Valeur_declaree_biens ='100000+' THEN '100000+' END AS Categorie, COUNT(contrat_id) AS Nombre_de_contrats FROM contrats GROUP BY Categorie; </pre>	# Categorie Nombre_de_contrats 1 0-25000 22720 2 100000+ 104 3 25000-50000 6815 4 50000-100000 696

**SELECT CASE WHEN Valeur\_declaree\_biens = '0-25000' ... END AS Categorie, COUNT(contrat\_id) :** On utilise la clause CASE pour définir des conditions et assigner des catégories aux enregistrements en fonction de la valeur de la colonne "Valeur\_declaree\_biens". Ensuite, on utilise COUNT(contrat\_id) pour compter le nombre de contrats dans chaque catégorie.

**FROM contrats :** Cette partie indique la table à partir de laquelle nous voulons récupérer les données. Dans ce cas, nous sélectionnons la table "contrats".

**GROUP BY Categorie :** On utilise GROUP BY pour regrouper les résultats en fonction de la colonne "Categorie" que nous avons définie à l'aide de la clause CASE. Cela signifie que nous obtiendrons des totaux pour chaque catégorie distincte.



## Requête n° 9 :

	Besoin	Requête	Résultat
9	Classement des 10 départements où le prix moyen de la cotisation est le plus élevé ?	<pre>SELECT code_departement AS Departement, ROUND(AVG(prix_cotisation_mensuel), 2) AS Cotisation_moyenne FROM contrats GROUP BY code_departement ORDER BY Cotisation_moyenne DESC LIMIT 10;</pre>	<pre># Departement Cotisation_moyenne 1 75 36.4 2 92 26.27 3 94 19.82 4 78 18.88 5 69 18.46 6 1 18.24 7 6 18.14 8 17 17.32 9 74 17.16 10 2A 17.07</pre>

**SELECT code\_departement, ROUND(AVG(prix\_cotisation\_mensuel), 2) :** On spécifie ici que nous voulons sélectionner deux colonnes dans le résultat. La première colonne est "code\_departement", et la deuxième colonne est la moyenne arrondie de "prix\_cotisation\_mensuel".

**FROM contrats :** Cette partie indique la table à partir de laquelle nous voulons récupérer les données. Dans ce cas, nous sélectionnons la table "contrats".

**GROUP BY code\_departement :** On utilise GROUP BY pour regrouper les résultats par département, ce qui signifie que les calculs (comme la moyenne) seront effectués pour chaque département distinct.

**ORDER BY Cotisation\_moyenne DESC :** Cette partie trie les résultats par ordre décroissant de la colonne "Cotisation\_moyenne".

**LIMIT 10 :** Cette partie limite le résultat aux 10 premiers départements après le tri.

## Requête n° 10 :

	Besoin	Requête	Résultat
10	Quel est le nombre de contrats avec des formules "Integral" pour la région Pays de la Loire ?	<pre>SELECT COUNT(contrat_id) AS Nombre_de_contrats FROM contrats c JOIN regions r ON c.code_postal = r.code_postal WHERE c.formule = 'Integral' AND r.reg_nom = 'Pays de la Loire';</pre>	# Nombre_de_contrats 1 590

**SELECT COUNT(contrat\_id)** : On spécifie ici que nous voulons compter le nombre de contrats. La fonction COUNT(contrat\_id) compte le nombre de valeurs non nulles dans la colonne "contrat\_id".

**FROM contrats c** : Cette partie indique la table à partir de laquelle nous voulons récupérer les données. Dans ce cas, nous sélectionnons la table "contrats".

**JOIN regions r ON c.code\_postal = r.code\_postal** : Cette partie effectue une jointure entre les tables "contrats" et "regions" en utilisant la colonne "code\_postal" comme critère de jointure. Cela signifie que les enregistrements seront associés en fonction des codes postaux correspondants.

**WHERE c.formule = 'Integral' AND r.reg\_nom = 'Pays de la Loire'** : Ajoute d'une restriction à la requête. On spécifie que nous voulons uniquement compter les contrats ayant la formule "Integral" (de la table "contrats") et qui sont associés à la région "Pays de la Loire" (de la table "regions").

# Requête n° 11 :

	Besoin	Requête	Résultat
11	Liste des communes ayant eu au moins 150 contrats ?	<pre>SELECT commune AS Communes, COUNT(contrat_id) AS Nombre_de_contrats FROM contrats GROUP BY commune HAVING COUNT(commune) &gt;= 150 ORDER BY COUNT(commune) DESC;</pre>	<pre># Communes Nombre_de_contrats 1 PARIS 18 515 2 PARIS 17 468 3 PARIS 15 407 4 PARIS 16 394 5 NICE 387 6 PARIS 11 381 7 PARIS 20 302 8 BORDEAUX 302 9 NANTES 291 10 PARIS 19 266 11 PARIS 10 263 12 PARIS 12 252 13 PARIS 14 222 14 GRENOBLE 220 15 PARIS 09 204 16 TOULOUSE 187 17 TOULON 170 18 COURBEVOIE 163 19 LILLE 161 20 PARIS 03 159</pre>

**SELECT commune, COUNT(contrat\_id) :** On spécifie ici que nous voulons sélectionner deux colonnes dans le résultat. La première colonne est "commune", et la deuxième colonne est le nombre de contrats pour chaque commune. La fonction COUNT(contrat\_id) compte le nombre de valeurs non nulles dans la colonne "contrat\_id".

**FROM contrats :** Cette partie indique la table à partir de laquelle nous voulons récupérer les données. Dans ce cas, nous sélectionnons la table "contrats".

**GROUP BY commune :** On utilise GROUP BY pour regrouper les résultats par commune, ce qui signifie que les calculs (comme le compte) seront effectués pour chaque commune distincte.

**HAVING COUNT(commune) >= 150 :** Cette partie filtre les résultats en utilisant la clause HAVING. On spécifie que nous voulons uniquement inclure les communes ayant un nombre de contrats d'au moins 150.

**ORDER BY COUNT(commune) DESC :** Cette partie trie les communes par ordre décroissant du nombre de contrats (COUNT(commune)). Cela signifie que les communes avec le plus grand nombre de contrats apparaîtront en premier.

## Requête n° 12 :

	Besoin	Requête	Résultat
12	Quel est le nombre de contrats pour chaque région ?	<pre> SELECT r.reg_nom as Nom_region, COUNT(DISTINCT c.contrat_id) AS Nombre_de_contrats FROM contrats c JOIN regions r ON c.code_departement = r.dep_code GROUP BY r.reg_nom </pre>	<pre> # Nom_region Nombre_de_contrats 1 Auvergne-RhôneAlpes 3056 2 Bourgogne-FrancheComté 287 3 Bretagne 947 4 Centre-Val de Loire 598 5 Corse 247 6 Grand Est 768 7 Guyane 37 8 Hauts-de-France 1182 9 La Réunion 17 10 Martinique 73 11 Normandie 822 12 Nouvelle-Aquitaine 2046 13 Occitanie 1601 14 Pays de la Loire 1196 15 Provence-Alpes-Côte d'Azur 3279 16 Île-de-France 14179 </pre>

**SELECT r.reg\_nom, COUNT(DISTINCT c.contrat\_id) :** On spécifiez ici que nous voulons sélectionner deux colonnes dans le résultat. La première colonne est "reg\_nom" de la table "regions", et la deuxième colonne est le nombre de contrats distincts pour chaque région. L'utilisation de DISTINCT avec COUNT permet de compter uniquement les contrats distincts (uniques).

**FROM contrats c :** Cette partie indique la table à partir de laquelle nous voulons récupérer les données. Dans ce cas, nous sélectionnons la table "contrats".

**JOIN regions r ON c.code\_departement = r.dep\_code :** Cette partie effectue une jointure entre les tables "contrats" et "regions" en utilisant la colonne "code\_departement" de la table "contrats" et la colonne "dep\_code" de la table "regions" comme critères de jointure. Cela signifie que les enregistrements seront associés en fonction des codes de département correspondants.

**GROUP BY r.reg\_nom :** On utilise GROUP BY pour regrouper les résultats par nom de région (colonne "reg\_nom"). Cela signifie que les calculs seront effectués pour chaque région distincte.