

Notice du Projet 3

Frédéric La Rosa

Nettoyer la base de donnée

Utilisation d'Excel et de Kutools

Supprimer toutes les colonnes vides avec un en-tête

Renommer la feuille « **Tableau** »

ALT + F11 dans **Excel** pour ouvrir **VBA**

Cliquer **Insérer > Module**

Coller code VBA :

```
Sub deleteblankcolwithheader()  
'updateby Extendoffice  
Dim xEndCol As Long  
Dim I As Long  
Dim xDel As Boolean  
On Error Resume Next  
xEndCol = Cells.Find("*", SearchOrder:=xlByColumns,  
SearchDirection:=xlPrevious).Column  
If xEndCol = 0 Then  
    MsgBox "There is no data on "" & ActiveSheet.Name & "" .", vbExclamation,  
"Kutools for Excel"  
    Exit Sub  
End If  
Application.ScreenUpdating = False  
For I = xEndCol To 1 Step -1  
    If Application.WorksheetFunction.CountA(Columns(I)) <= 1 Then  
        Columns(I).Delete  
        xDel = True  
    End If  
Next  
If xDel Then  
    MsgBox "All blank column(s) with only a header row have been deleted.",  
vbInformation, "Kutools for Excel"  
Else  
    MsgBox "There are no Columns to delete as each one has more data (rows) than just  
a header.", vbExclamation, "Kutools for Excel"  
End If  
Application.ScreenUpdating = True  
End Sub
```

Exécuter le code (**F5**), valider boîte de dialogue qui rappelle la commande

Cliquer sur **OK**

Ajouter le préfixe « **0** » pour toutes les valeurs **Code département** < ou = 9

Mettre les valeurs de la colonne **Code département** au format **Texte**

Filtrer colonne **Code département** de **1** à **9** hormis **2a** et **2b**

Onglet **Kutools** > **Texte** > **Ajouter Texte**

Ajouter « **0** » **avant le premier caractère**

Retirer le filtre de la colonne

Importer les données « **nom_département** »

Télécharger fichier **departements-france.csv** sur **data.gouv.fr**

<https://static.data.gouv.fr/resources/departements-de-france/20200425-135513/departements-france.csv>

Créer le dictionnaire des données

Utilisation d'Excel

Numéro	Code propriété	Signification	Type	Observation
1	num_disp	Num disposition	Numérique	SMALLINT NOT NULL
2	date_mut	Date mutation	Date	jj-mm-aaaa NOT NULL
3	val_fonciere	Valeur fonciere	Monétaire	DECIMAL(11,2) NULL
4	num_voie	Numéro de voie	Numérique	SMALLINT NULL
5	compl	B/T/Q	Texte	CHAR(1) NULL
...

Modéliser la base de donnée

*Utilisation de **SQL Power Architect***

Respecter les règles de normalisation :

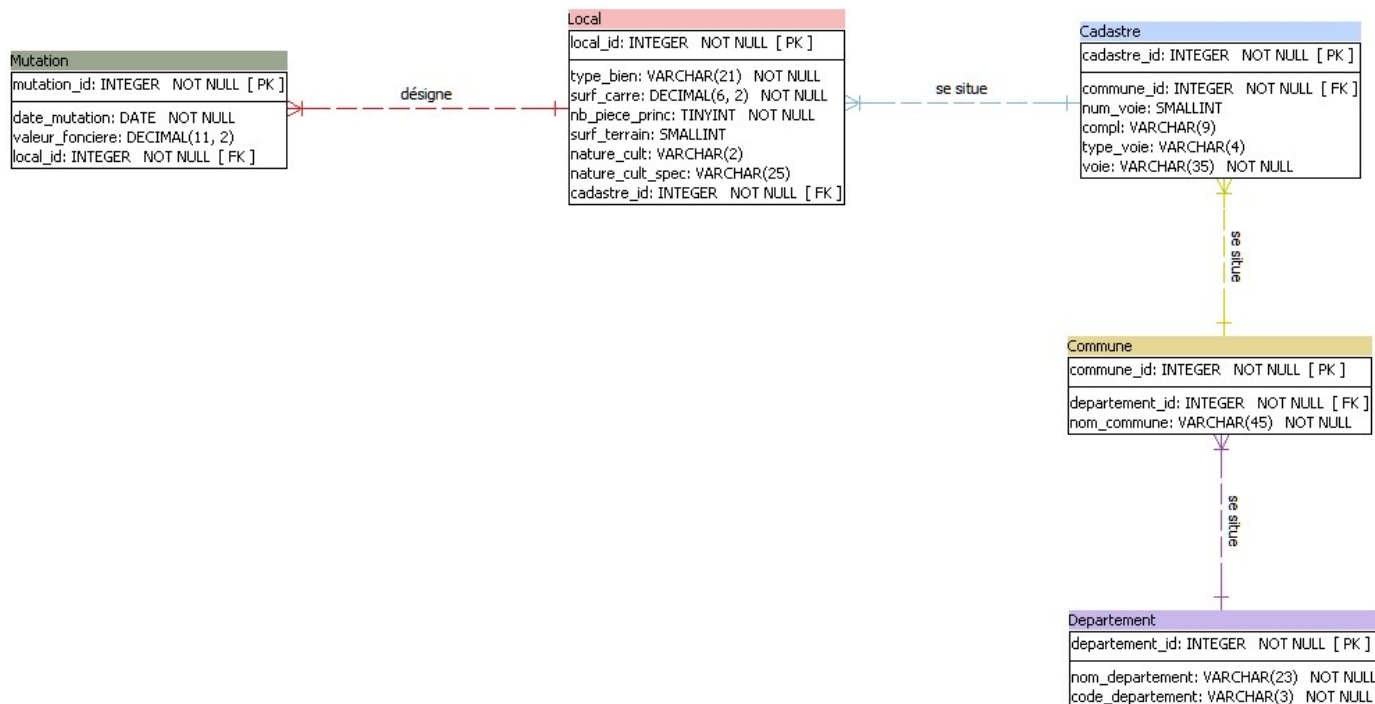
1FN: Données atomiques

2FN: Les attributs non clés ne doivent pas dépendre que d'une partie de la clé

3FN: les attributs non clés doivent dépendre de la clé primaire, si certains ne dépendent pas QUE de la clé primaire => on crée une table

Indiquer les cardinalités :

Exemple : un local se situe dans 1 et 1 seule cadastre; un cadastre peut contenir 1 ou plusieurs (n) local



Répartir les données en tables et fichiers CSV

Utilisation d'Excel

Retirer les doublons avec la fonction RechercheV

(ATTENTION jointure sur la gauche)

Copier > coller les colonnes correspondant aux tables « **Mutation** », « **Local** », « **Cadastre** », « **Commune** », « **Département** » dans des feuilles séparées

Pour chaque table, une feuille « **nom_table_doublon** » traiter les doublons de la base de donnée

Insérer une nouvelle colonne « **Identifiant** » en **A1** pour chaque table

Insérer une fonction CONCATENER en A2 reprenant les attributs de la table

A2					
					=CONCATENER(D2&"_"&C2&"_"&B2)
		A	B	C	D
1		identifiant	num_disposition	date_mutation	val_fonciere
2		56000_43864_1	1	43864	56000

Insérer un **Tableau croisé dynamique** avec « **identifiant** » en **Lignes** et **Valeurs**

Copier la colonne « **Etiquettes de lignes** » > coller dans une nouvelle feuille « **nom_table** »

Dans la feuille « **nom_table** » insérer à la suite les attributs composants l'identifiant

En B1 placer la colonne « **table_id** » ou insérer une colonne « **id** » si clé artificielle (incrémenter la colonne)

En **C2, (D2), (E2) ...** Rechercher les valeurs pour chaque attribut et donner une condition de retour de valeur « » si case vide

Exemple : Feuille « **mutation** »

```
=SI(RECHERCHEV(A2;mutation_doublon!A:D;2;FAUX)=0;"";RECHERCHEV(A2;mutation_doublon!A:D;2;FAUX))
```

Si une clé artificielle doit être créer dans une table

Insérer une colonne temporaire « **Colonne_x** » suivi d’une colonne « **table_id** » avant les attributs de la table dans la feuille « **Tableau** »

Insérer fonction **CONCATENER** reprenant les attributs de la table

Exemple :

```
=CONCATENER([@voie]&"_"&[@type_voie]&"_"&[@compl]&"_"&[@num_voie]))
```

Insérer fonction **RechercheV** dans la ligne de « **table_id** » ou « **nom_attribut_manquant** » pour joindre la valeur manquante

=RECHERCHEV([@Colonne1];cadastre!A:F;2;FAUX)						
Colonne1	cadastre_id	num_voie	compl	type_voie	voie	
0,00 CENTRALE_RUE_A_190	26539	190	A	RUE	CENTRALE	
0,00 DU CHATEAU_RUE_347	22903	347		RUE	DU CHATEAU	

Nettoyer les « **table_doublon** » et les colonnes « **identifiant** »

ATTENTION faire un **copier > collage spécial** de toutes les colonnes à conserver sinon la valeur des attributs sera perdue dans le nettoyage

Enregistrer « **Tableau** » et « **noms_tables** » dans x fichiers CSV pour x table

Solution alternative : utilisation de Power Query

Créer et insérer des colonnes index représentant les clés étrangères

Pour chaque table, sélectionner les colonnes utiles

Exporter le résultat dans une nouvelle feuille

Copier-coller la table dans un nouveau classeur et l’enregistrer sous fichier CSV

Création de la database et des tables avec un SGDB-R

Utilisation de l'extension **MySQL** dans l'éditeur **Visual Studio Code**

Création de la database « laplace_immo »

```
CREATE DATABASE laplace_immo CHARACTER SET 'utf8' ;
```

Création des tables

```
--- J'utilise la Database laplace_immo
USE laplace_immo;

--- Je nettoie ma Database en retirant les tables si elles existent
DROP TABLE IF EXISTS mutation;
DROP TABLE IF EXISTS local;
DROP TABLE IF EXISTS cadastre;
DROP TABLE IF EXISTS commune;
DROP TABLE IF EXISTS departement;

--- Je crée mes tables en respectant mon dictionnaire de données

-- Je crée ma table departement
CREATE TABLE departement (
  id INT AUTO_INCREMENT NOT NULL,
  nom_departement VARCHAR(23) NOT NULL,
  code_departement VARCHAR(3) NOT NULL,
  PRIMARY KEY (id)
);

-- Je crée ma table commune
CREATE TABLE Commune (
  id INT NOT NULL,
  nom_commune VARCHAR(45) NOT NULL,
  departement_id INT NOT NULL,
  PRIMARY KEY (id),
  FOREIGN KEY (departement_id) REFERENCES departement(id)
);

-- Je crée ma table cadastre
CREATE TABLE cadastre (
  id INT AUTO_INCREMENT NOT NULL,
  num_voie SMALLINT,
  compl VARCHAR(9),
  type_voie VARCHAR(4),
  voie VARCHAR(35) NOT NULL,
  commune_id INT NOT NULL,
  PRIMARY KEY (id),
  FOREIGN KEY (commune_id) REFERENCES commune(id)
);
```

```

-- Je crée ma table local
CREATE TABLE local (
  id INT AUTO_INCREMENT NOT NULL,
  type_local VARCHAR(21) NOT NULL,
  surf_carre DECIMAL(6, 2) NOT NULL,
  nb_piece_princ INTEGER NOT NULL,
  surf_terrain SMALLINT,
  nature_cult VARCHAR(2),
  nature_cult_spec VARCHAR(25),
  cadastre_id INT NOT NULL,
  PRIMARY KEY (id),
  FOREIGN KEY (cadastre_id) REFERENCES cadastre(id)
);

-- Je crée ma table mutation
CREATE TABLE mutation (
  id INT AUTO_INCREMENT NOT NULL,
  date_mutation DATE NOT NULL,
  valeur_fonciere DECIMAL(11, 2) NULL,
  local_id INT NOT NULL,
  PRIMARY KEY (id),
  FOREIGN KEY (local_id) REFERENCES local(id)
);

--- J'importe les data nécessaires à chacune de mes tables

-- J'importe les data departement.csv
LOAD DATA LOCAL INFILE "C:/ProgramData/MySQL/MySQL Server
8.0/Data/laplace_immo/Departement.csv" INTO TABLE departement FIELDS TERMINATED By ';'
lines terminated by '\r\n' IGNORE 1 LINES;

-- J'importe les data commune.csv
LOAD DATA LOCAL INFILE "C:/ProgramData/MySQL/MySQL Server
8.0/Data/laplace_immo/Commune.csv" INTO TABLE commune FIELDS TERMINATED By ';' lines
terminated by '\r\n' IGNORE 1 LINES;

-- J'importe les data cadastre.csv
LOAD DATA LOCAL INFILE "C:/ProgramData/MySQL/MySQL Server
8.0/Data/laplace_immo/Cadastre.csv" INTO TABLE cadastre FIELDS TERMINATED By ';' lines
terminated by '\r\n' IGNORE 1 LINES;

-- J'importe les data local.csv
LOAD DATA LOCAL INFILE "C:/ProgramData/MySQL/MySQL Server 8.0/Data/laplace_immo/Local.csv"
INTO TABLE local FIELDS TERMINATED By ';' lines terminated by '\r\n' IGNORE 1 LINES;

-- J'importe les data mutation.csv
LOAD DATA LOCAL INFILE "C:/ProgramData/MySQL/MySQL Server
8.0/Data/laplace_immo/Mutation.csv" INTO TABLE mutation FIELDS TERMINATED By ';' lines
terminated by '\r\n' IGNORE 1 LINES;

```


Exécuter les requêtes SQL

Utilisation de l'extension **MySQL** dans l'éditeur **Visual Studio Code**

1. Nombre total d'appartements vendus au 1er semestre 2020 ?

```
SELECT COUNT(type_local)
FROM local
WHERE type_local = 'Appartement'
;
```

-- Résultat: 31378 appartements vendus au 1er semestre 2020

<input type="checkbox"/>		* COUNT(type_local) int	⬆
		Filter	
<input type="checkbox"/>	1	31378	

2. Proportion des ventes d'appartements par le nombre de pièces ?

```
SELECT
    type_local ,
    nb_piece_princ,
    (COUNT(type_local) * 100 / (SELECT COUNT(*) FROM local)) AS proportion_vente
FROM local
WHERE type_local = 'Appartement'
GROUP BY nb_piece_princ
ORDER BY nb_piece_princ DESC
;

-- Résultats de la proportion des ventes d'appartements,
-- ordonnés par le nbre de pièces du + grand au + petit:
-- 11 = 0.0029% ; 10 = 0.0059% ; 9 = 0.0234% ; 8 = 0.0498% ;
-- 7 = 0.1580% ; 6 = 0.5970% ; 5 = 3.2603% ; 4 = 13.0528% ;
-- 3 = 26.2402% ; 2 = 28.6312% ; 1 = 19.7226% ; 0 = 0.0878%
```

		* type_local int	* nb_piece_princ varchar(21)	* proportion_vente decimal(6,2)
		Filter	Filter	Filter
	1	Appartement	11	0.0029
	2	Appartement	10	0.0059
	3	Appartement	9	0.0234
	4	Appartement	8	0.0498
	5	Appartement	7	0.1580
	6	Appartement	6	0.5970
	7	Appartement	5	3.2603
	8	Appartement	4	13.0528
	9	Appartement	3	26.2402
	10	Appartement	2	28.6312
	11	Appartement	1	19.7226
	12	Appartement	0	0.0878

3. Liste des 10 départements où le prix du mètre carré est le plus élevé ?

```
SELECT
    CONCAT(nom_departement,"_", "(" , code_departement, ")") AS departement,
    ROUND(AVG(valeur_fonciere / surf_carre)) AS prix_m2
FROM mutation
JOIN local
    ON mutation.local_id = local.id
JOIN cadastre
    ON local.cadastre_id = cadastre.id
JOIN commune
    ON cadastre.commune_id = commune.id
JOIN departement
    ON commune.departement_id = departement.id
GROUP BY code_departement
ORDER BY prix_m2 DESC
LIMIT 10
;
```

-- Résultats classés par département,
-- ordonnés par prix moyen arrondi du + cher au - cher:
-- Paris_(75) = 12 084€ ; Hauts-de-Seine_(92) = 7301€ ; Val-de-Marne_(94) = 5428€
-- Haute-Savoie_(74) = 4781€ ; Alpes-Maritimes_(6) = 4756€ ; Seine-Saint-Denis_(93) = 4386€
-- Yvelines_(78) = 4276 € ; Rhone_(69) = 4100€
-- Corse-du-Sud_(2A) = 4063€ ; Gironde_(33) = 3807€

	* type_local int	* nb_piece_princ varchar(21)	* proportion_vente decimal(6,2)
	Filter	Filter	Filter
1	Appartement	11	0.0029
2	Appartement	10	0.0059
3	Appartement	9	0.0234
4	Appartement	8	0.0498
5	Appartement	7	0.1580
6	Appartement	6	0.5970
7	Appartement	5	3.2603
8	Appartement	4	13.0528
9	Appartement	3	26.2402
10	Appartement	2	28.6312
11	Appartement	1	19.7226
12	Appartement	0	0.0878

4. Prix moyen du mètre carré d'une maison en Île-de-France ?

```
SELECT
    type_local,
    code_departement AS ile_de_france,
    ROUND(AVG(valeur_fonciere / surf_carre)) AS prix_m2
FROM mutation
JOIN local
    ON mutation.local_id = local.id
JOIN cadastre
    ON local.cadastre_id = cadastre.id
JOIN commune
    ON cadastre.commune_id = commune.id
JOIN departement
    ON commune.departement_id = departement.id
WHERE code_departement IN (75, 77, 78, 91, 92, 93, 94, 95)
AND type_local = "Maison"
;
```

-- Résultat: le prix moyen (arrondi) du mètre carré d'une maison en Île-de-France est de 3765€

	* departement int	* prix_m2 date
	Filter	Filter
1	Paris_(75)	12084
2	Hauts-de-Seine_(92)	7301
3	Val-de-Marne_(94)	5428
4	Haute-Savoie_(74)	4781
5	Alpes-Maritimes_(6)	4756
6	Seine-Saint-Denis_(93)	4386
7	Yvelines_(78)	4276
8	Rhone_(69)	4100
9	Corse-du-Sud_(2A)	4063
10	Gironde_(33)	3807

5. Liste des 10 appartements les plus chers avec le département et le nombre de mètres carrés ?

```
SELECT
    type_local,
    FORMAT(valeur_fonciere, "###.###.###.###"),
    CONCAT(nom_departement,"_", "(" , code_departement, ")") AS departement,
    surf_carre
FROM mutation
JOIN local
    On mutation.local_id = local.id
JOIN cadastre
    ON local.cadastre_id = cadastre.id
JOIN commune
    ON cadastre.commune_id = commune.id
JOIN departement
    ON commune.departement_id = departement.id
WHERE type_local = "Appartement"
GROUP BY valeur_fonciere
ORDER BY valeur_fonciere DESC
LIMIT 10
;
-- Résultats: Appartement 9 000 000€ Paris_(75) 9m2 ; Appartement 8 600 000€ Essonne_(91)
64m2
-- Appartement 8 577 713€ Paris_(75) 20m2 ; Appartement 7 620 000€ Paris_(75) 42m2
-- Appartement 7 600 000€ Paris_(75) 253m2 ; Appartement 7 535 000€ Paris_(75) 139m2
-- Appartement 7 420 000€ Paris_(75) 360m2 ; Appartement 7 200 000€ Paris_(75) 595m2
-- Appartement 7 050 000€ Paris_(75) 122m2 ; Appartement 6 600 000€ Paris_(75) 79m2
```

	* type_local int	* FORMAT(valeur_fonc varchar(21)	* departement decimal(6,2)	* surf_carre int
	Filter	Filter	Filter	Filter
1	Appartement	9,000,000	Paris_(75)	9.00
2	Appartement	8,600,000	Essonne_(91)	64.00
3	Appartement	8,577,713	Paris_(75)	20.00
4	Appartement	7,620,000	Paris_(75)	42.00
5	Appartement	7,600,000	Paris_(75)	253.00
6	Appartement	7,535,000	Paris_(75)	139.00
7	Appartement	7,420,000	Paris_(75)	360.00
8	Appartement	7,200,000	Paris_(75)	595.00
9	Appartement	7,050,000	Paris_(75)	122.00
10	Appartement	6,600,000	Paris_(75)	79.00

6. Taux d'évolution du nombre de ventes entre le premier et le second trimestre de 2020 ?

WITH

```
1er_trimestre AS (  
  SELECT  
    COUNT(id) AS 1er_trim_ventes  
  FROM mutation  
  WHERE date_mutation BETWEEN '2020-01-01' AND '2020-03-31'),
```

```
2nd_trimestre AS (  
  SELECT  
    COUNT(id) AS 2nd_trim_ventes  
  FROM mutation  
  WHERE date_mutation BETWEEN '2020-04-01' AND '2020-06-30')
```

```
SELECT  
  ROUND(((2nd_trim_ventes - 1er_trim_ventes) / 1er_trim_ventes * 100),2) AS  
taux_evolution  
FROM 1er_trimestre, 2nd_trimestre
```

TH

```
table1 AS (  
  SELECT COUNT(id) AS 1er_trim_ventes  
  FROM mutation  
  WHERE date_mutation BETWEEN '2020-01-01' AND '2020-03-31'),
```

```
table2 AS (  
  SELECT COUNT(id) AS 2nd_trim_ventes  
  FROM mutation  
  WHERE date_mutation BETWEEN '2020-04-01' AND '2020-06-30')
```

```
SELECT ROUND(((2nd_trim_ventes - 1er_trim_ventes) / 1er_trim_ventes * 100),2) AS  
taux_evolution  
FROM table1, table2  
;
```

-- Résultat: le taux d'évolution est en baisse de -3.70% entre le 1er et le 2nd trimestre 2020

* taux_evolution int		
Filter		
	1	3.68

7. Liste des communes où le nombre de ventes a augmenté d'au moins 20% entre le premier et le second trimestre de 2020 ?

```
WITH
1er_trimestre AS (
    SELECT
        COUNT(mutation.id) AS 1er_trim_ventes,
        nom_commune
    FROM mutation
    JOIN local
        ON mutation.local_id = local.id
    JOIN cadastre
        ON local.cadastre_id = cadastre.id
    JOIN commune
        ON cadastre.commune_id = commune.id
    WHERE date_mutation BETWEEN '2020-01-01' AND '2020-03-31'
    GROUP BY nom_commune),

2nd_trimestre AS (
    SELECT
        COUNT(mutation.id) AS 2nd_trim_ventes,
        nom_commune
    FROM mutation
    JOIN local
        ON mutation.local_id = local.id
    JOIN cadastre
        ON local.cadastre_id = cadastre.id
    JOIN commune
        ON cadastre.commune_id = commune.id
    WHERE date_mutation BETWEEN '2020-04-01' AND '2020-06-30'
    GROUP BY nom_commune)

SELECT
    nom_commune,
    ROUND(((2nd_trim_ventes - 1er_trim_ventes) / 1er_trim_ventes * 100),2)
        AS taux_evolution
FROM 1er_trimestre
LEFT JOIN 2nd_trimestre USING(nom_commune)
WHERE ROUND(((2nd_trim_ventes - 1er_trim_ventes) / 1er_trim_ventes * 100),2) >= 20
ORDER BY taux_evolution DESC
;

-- Résultats: les communes qui ont eu un taux d'évolution >= 20 % entre le 1er et le 2nd
trimestre
-- PAU = 2500 % ; CAVAILLON = 1600 % ; L'ISLE SUR LA SORGUE = 1200 %
-- LE DEVOLUY = 1000 % ; MORLAIX = 1000 % ; ... 574 communes au total
```



* nom_commune
int



* taux_evolution
varchar(45)



	Filter	Filter
1	PAU	2500.00
2	CAVAILLON	1600.00
3	L ISLE SUR LA SORGUE	1200.00
4	LE DEVOLUY	1000.00
5	MORLAIX	1000.00
6	LYON 8EME	960.00
7	RONCHIN	800.00
8	LYON 7EME	800.00
9	CHALON-SUR-SAONE	766.67
10	ROUEN	716.67
11	LUCE	700.00
12	LYON 3EME	609.09
13	RETHEL	600.00
14	LAMBALLE-ARMOR	600.00
15	LAMORLAYE	600.00
16	LONS	600.00
17	MACON	600.00
18	PERRAY-EN-YVELINES (LE)	600.00
19	ORANGE	600.00
20	BELLEVILLE-EN-BEAUJOLAIS	500.00
21	VETRAZ-MONTHOUX	500.00
22	ORCIERES	400.00
23	VERNON	400.00
24	NOGENT-LE-ROI	400.00
25	PONT ST ESPRIT	400.00
26	SAINT CYR SUR LOIRE	400.00
27	VOREPPE	400.00

8. Différence en pourcentage du prix au mètre carré entre un appartement de 2 pièces et un appartement de 3 pièces ?

```
WITH
2_pieces AS (
    SELECT
        ROUND(AVG(valeur_fonciere / surf_carre)) AS prix_m2_appart_2_pieces
    FROM mutation
    JOIN local
        ON mutation.local_id = local.id
    JOIN cadastre
        ON local.cadastre_id = cadastre.id
    JOIN commune
        ON cadastre.commune_id = commune.id
    WHERE nb_piece_princ = 2 AND type_local = 'appartement'
),

3_pieces AS (
    SELECT
        ROUND(AVG(valeur_fonciere / surf_carre)) AS prix_m2_appart_3_pieces
    FROM mutation
    JOIN local
        ON mutation.local_id = local.id
    JOIN cadastre
        ON local.cadastre_id = cadastre.id
    JOIN commune
        ON cadastre.commune_id = commune.id
    WHERE nb_piece_princ = 3 AND type_local = 'appartement'
)

SELECT
    ROUND(AVG((prix_m2_appart_2_pieces - prix_m2_appart_3_pieces) /
prix_m2_appart_3_pieces * 100),2)
        AS pourcentage_difference
FROM 2_pieces, 3_pieces
;
-- Résultat: il y a 14.51 % de différence entre le prix d'un appartement 2 pièces et d'un
3 pièces
```

* pourcentage_differer	
int	
Filter	
1	14.51

9. Les moyennes de valeurs foncières pour le top 3 des communes des départements 6, 13, 33, 59 et 69 ?

```
WITH
valeur_moyenne_par_ville AS (
    SELECT
        code_departement,
        nom_departement,
        nom_commune,
        AVG(mutation.valeur_fonciere) AS valeur_fonciere
    FROM mutation
    JOIN local
        ON mutation.local_id = local.id
    JOIN cadastre
        ON local.cadastre_id = cadastre.id
    JOIN commune
        ON cadastre.commune_id = commune.id
    JOIN departement
        ON commune.departement_id = departement.id
    WHERE code_departement IN (6,13,33,59,69)
    GROUP BY code_departement, nom_commune)
SELECT
    CONCAT(nom_departement,"_", "(" , code_departement, ")") AS departement,
    nom_commune AS commune,
    FORMAT(ROUND(valeur_fonciere), '###.###.###') AS prix_moyen
FROM (
    SELECT
        code_departement,
        nom_departement,
        nom_commune,
        valeur_fonciere,
        RANK() OVER (PARTITION BY code_departement ORDER BY valeur_fonciere DESC) AS
rang
    FROM valeur_moyenne_par_ville) AS resultat_top_trois
WHERE rang <= 3
;
-- Résultats: Top 3 des communes pour les départements 6, 13, 33, 59 et 69
-- Alpes-Maritimes_(6) :
-- 1) SAINT-JEAN-CAP-FERRAT = 968 750 € - 2) EZE = 655 000 € - 3) MOUANS-SARTOUX = 476
898 €
-- Bouches-du-Rhone_(13) :
-- 1) GIGNAC-LA-NERTHE = 330 000 € - 2) SAINT SAVOURNIN = 314 425 € - 3) CASSIS = 313 417
€
-- Gironde_(33) :
-- 1) LEGE-CAP-FERRET = 549 501 € - 2) VAYRES = 335 000 € - 3) ARCACHON = 307 436 €
-- Nord_(59) :
-- 1) BERSEE = 433 202 € - 2) CYSOING = 408 550 € - 3) HALLUIN = 322 250 €
-- Rhone_(69)
-- 1) VILLE SUR JARNIOUX = 485 300 € - 2) LYON 2eme = 455 217 € - 3) LYON 6eme = 426 968 €
```


	* departement int	* commune date	prix_moyen decimal(11,2)
	Filter	Filter	Filter
1	Bouches-du-Rhone_(13)	GIGNAC-LA-NERTHE	330,000
2	Bouches-du-Rhone_(13)	SAINT SAVOURNIN	314,425
3	Bouches-du-Rhone_(13)	CASSIS	313,417
4	Gironde_(33)	LEGE-CAP-FERRET	549,501
5	Gironde_(33)	VAYRES	335,000
6	Gironde_(33)	ARCACHON	307,436
7	Nord_(59)	BERSEE	433,202
8	Nord_(59)	CYSOING	408,550
9	Nord_(59)	HALLUIN	322,250
10	Alpes-Maritimes_(6)	SAINT-JEAN-CAP-FERRAT	968,750
11	Alpes-Maritimes_(6)	EZE	655,000
12	Alpes-Maritimes_(6)	MOUANS-SARTOUX	476,898
13	Rhone_(69)	VILLE SUR JARNIOUX	485,300
14	Rhone_(69)	LYON 2EME	455,217
15	Rhone_(69)	LYON 6EME	426,968