

Requête 1 :

Nombre total d'appartements vendus au 1er semestre 2020 : 31 376.

SQL File 5* x SQL File 6* SQL File 7* SQL File 10* SQL File 11* SQL File 13* SQL File 12* SQL File 14* SQL File 15* SQL File 16*

Limit to 1000 rows

```
1 SELECT count(*) AS 'Nombre total d'appartements vendus au 1er semestre 2020'
2 FROM `mutation`
3 JOIN `bien` ON `mutation`.ID_Bien = `bien`.ID_Bien
4 WHERE bien.typeBien = 'Appartement' and `mutation`.`dateTran` BETWEEN '2020-01-01' AND '2020-06-30'
5
```

< >

Result Grid Filter Rows: Export: Wrap Cell Content:

Nombre total d'appartements vendus au 1er semestre 2020
31376

Result Grid
Form Editor

Requête 2 :
Proportion des ventes
d'appartements par le nombre de
pièces :

0 pièce : 0.10%
1 pièce : 21.48%
2 pièces : 31.18%
3 pièces : 28.57%
4 pièces : 14.21%
5 pièces : 3.55%
6 pièces : 0.65%
7 pièces : 0.17%
8 pièces : 0.05%
9 pièces : 0.03%
10 pièces : 0.01%
11 pièces : 0 %

SQL File 5* SQL File 6* x SQL File 7* SQL File 10* SQL File 11* SQL File 13* SQL File 12* SQL File 14* SQL File 15* SQL File 16*

Limit to 1000 rows

```
1 select bien.nbrePiPrin as 'nombre de pièce', round(count(mutation.ID_Mutation) / (select count(mutation.ID_Mutation)
2 From Mutation
3 JOIN `bien` ON `mutation`.ID_Bien = `bien`.ID_Bien`
4 Where bien.typeBien = 'Appartement')*100,2) AS 'Proportion des ventes'
5 FROM `mutation`
6 JOIN `bien` ON `mutation`.ID_Bien = `bien`.ID_Bien`
7 WHERE bien.typeBien = 'Appartement'
8 GROUP BY bien.nbrePiPrin
9 ORDER BY bien.nbrePiPrin;
10
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: [IA](#)

	nombre de pièce	Proportion des ventes
▶	0	0.10
	1	21.48
	2	31.18
	3	28.57
	4	14.21
	5	3.55
	6	0.65
	7	0.17
	8	0.05
	9	0.03
	10	0.01
	11	0.00

requete 3 :

Liste des 10 départements où le prix du mètre carré est le plus élevé :

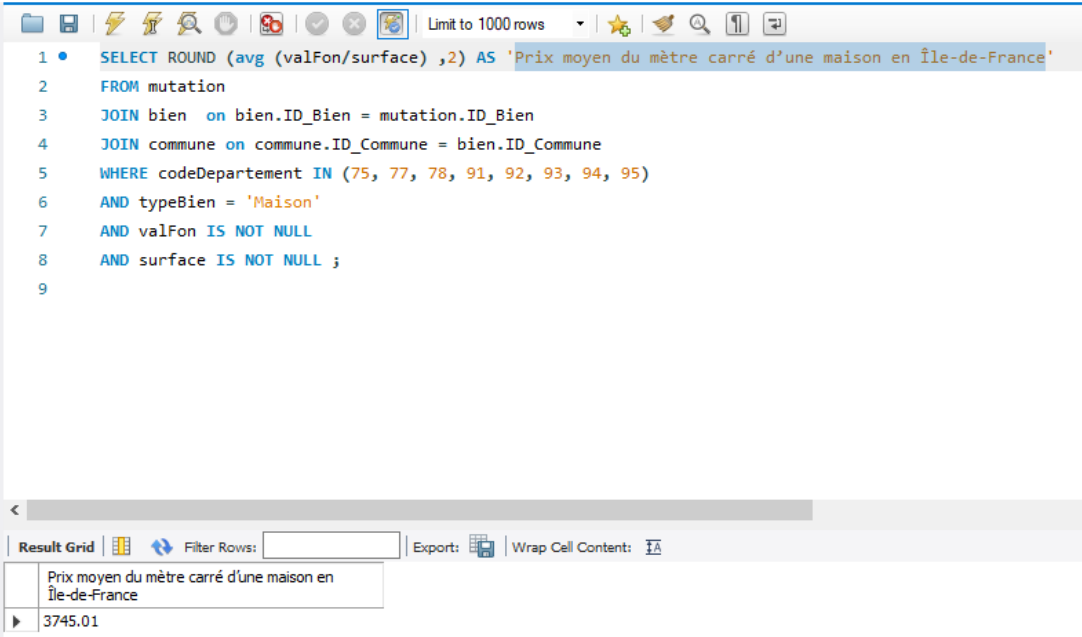
75,92,94,6,74,93,78,69,33,83.

```
1 • SELECT ROUND (AVG ( mutation.valFon/bien.surface) , 2) AS 'Prix mètre carré', codeDepartement
2     FROM bien
3     JOIN mutation on bien.ID_Bien = mutation.ID_Bien
4     JOIN commune ON bien.ID_Commune = commune.ID_Commune
5     WHERE surface != 0 AND codeDepartement IS NOT NULL
6     GROUP BY codeDepartement
7     ORDER BY ROUND (AVG ( mutation.valFon/bien.surface), 2) DESC
8     LIMIT 10;
9
```

Result Grid			Filter Rows:	Export:	Wrap Cell Content:	Fetch rows:
	Prix mètre carré	codeDepartement				
▶	12045.42	75				
	7219.39	92				
	5340.51	94				
	4696.83	6				
	4667.13	74				
	4336.7	93				
	4225.25	78				
	4059.31	69				
	3764.14	33				
	3718.08	83				

Requête 4:

Prix moyen du mètre carré d'une maison en Île-de-France : 3745.01€.



The screenshot shows a SQL query editor with a toolbar at the top. The query is as follows:

```
1 • SELECT ROUND (avg (valFon/surface) ,2) AS 'Prix moyen du mètre carré d'une maison en Île-de-France'
2 FROM mutation
3 JOIN bien on bien.ID_Bien = mutation.ID_Bien
4 JOIN commune on commune.ID_Commune = bien.ID_Commune
5 WHERE codeDepartement IN (75, 77, 78, 91, 92, 93, 94, 95)
6 AND typeBien = 'Maison'
7 AND valFon IS NOT NULL
8 AND surface IS NOT NULL ;
9
```

Below the query editor, there is a "Result Grid" section. It includes a "Filter Rows:" input field, an "Export:" button, and a "Wrap Cell Content:" checkbox. The result grid displays the following data:

	Prix moyen du mètre carré d'une maison en Île-de-France
▶	3745.01

Requête 5:

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

Bien numéro :

32251, 21816, 29777, 32409, 29828, 29500, 31949, 32111, 29331, 29491.

```
1 • SELECT
2   bien.ID_Bien,
3   surface,
4   typeBien,
5   valFon,
6   codeDepartement
7 FROM mutation
8 JOIN bien on bien.ID_Bien = mutation.ID_Bien
9 JOIN commune on commune.ID_Commune = bien.ID_Commune
10 WHERE bien.typeBien = 'Appartement'
11 ORDER BY valFon DESC LIMIT 10 ;
12
```

Result Grid					
Filter Rows:					
Export: Wrap Cell Conte					
	ID_Bien	surface	typeBien	valFon	codeDepartement
►	32251	9.1	Appartement	9000000	75
	21816	64	Appartement	8600000	91
	29777	20.55	Appartement	8577710	75
	32409	42.77	Appartement	7620000	75
	29828	253.3	Appartement	7600000	75
	29500	139.9	Appartement	7535000	75
	31949	360.95	Appartement	7420000	75
	32111	595	Appartement	7200000	75
	29331	122.56	Appartement	7050000	75
	29491	79.38	Appartement	6600000	75

Requête 6:

Taux d'évolution du nombre de ventes entre le premier et le second trimestre 2020 = 3.67%.

```
1 • with
2
3 table1 as(
4     select count(ID_Mutation) as Premier_trimestre
5     from mutation
6     where dateTran between '2020-01-01' AND '2020-03-31'),
7
8 table2 as(
9     SELECT COUNT(ID_Mutation) as Second_trimestre
10    FROM mutation
11   WHERE dateTran between '2020-04-01' AND '2020-06-30')
12
13 SELECT ((Second_trimestre-Premier_trimestre)/Premier_trimestre*100) as 'Taux d'évolution du nombre de ventes entre le premier et le second
14 trimestre de 2020.'
15
16 from table1, table2 ;
17
18
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

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

3.6781







Requête 7:

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

```
1 • with
2 ventetrimestre1 as (
3   select commune, count(ID_Mutation) AS vente_1ertrimestre_2020
4   FROM mutation
5   JOIN bien on bien.ID_Bien = mutation.ID_Bien
6   JOIN commune on commune.ID_Commune = bien.ID_Commune
7   WHERE dateTran between '2020-01-01' AND '2020-03-31'
8   GROUP BY commune),
9 ventetrimestre2 as (
10  select commune, count(ID_Mutation) AS vente_2emetrimestre_2020
11  FROM mutation
12  JOIN bien on bien.ID_Bien = mutation.ID_Bien
13  JOIN commune on commune.ID_Commune = bien.ID_Commune
14  WHERE dateTran between '2020-04-01' AND '2020-06-30'
15  GROUP BY commune)
16
17 SELECT commune AS "Commune", vente_1ertrimestre_2020 AS "Vente trimestre 1",
18 vente_2emetrimestre_2020 AS "Vente trimestre 2",
19 round(((vente_2emetrimestre_2020 - vente_1ertrimestre_2020) / vente_1ertrimestre_2020 * 100), 2) AS "Taux d'evolution des ventes entre le premier et le second trimestre de 2020 "
20 from ventetrimestre1
21 join ventetrimestre2 using (commune)
22 WHERE round (((vente_2emetrimestre_2020 - vente_1ertrimestre_2020) / vente_1ertrimestre_2020 * 100), 2) > 20
23 :
```

	Commune	Vente trimestre 1	Vente trimestre 2	Taux d'évolution des ventes entre le premier et le second trimestre de 2020
►	PAU	3	78	2500.00
	CAVAILLON	1	17	1600.00
	L ISLE SUR LA SORGUE	1	13	1200.00
	LE DEVOLUY	1	11	1000.00
	MORLAIX	1	11	1000.00
	LYON 8EME	5	53	960.00
	RONCHIN	1	9	800.00
	LYON 7EME	7	63	800.00
	CHALON-SUR-SAONE	3	26	766.67
	ROUEN	12	98	716.67
	LUCE	1	8	700.00
	LYON 3EME	11	78	609.09
	RETHEL	1	7	600.00
	LAMBALLE-ARMOR	1	7	600.00



SQL File 5* SQL File 6* SQL File 7* SQL File 10* SQL File 11* SQL File 13* SQL File 12* SQL File 14* x SQL File 16*

 Limit to 1000 rows     

```
1 • WITH
2 prix_m2_appartement_t2 AS (select round (avg(valFon /surface),2) AS PrixM2AppartementT2
3   from mutation
4   JOIN bien on mutation.ID_Bien = bien.ID_Bien
5   WHERE typebien = "appartement" AND nbrePiPrin =2),
6 prix_m2_appartement_t3 AS (select round (avg(valFon /surface),2) AS PrixM2AppartementT3
7   from mutation
8   JOIN bien on mutation.ID_Bien = bien.ID_Bien
9   WHERE typebien = "appartement" AND nbrePiPrin =3)
10 SELECT ROUND (((PrixM2AppartementT3 - PrixM2AppartementT2) / PrixM2AppartementT2) * 100, 2)
11 AS "Différence en pourcentage du prix au mètre carré entre un appartement de 2 pièces et un appartement de 3 pièces :"
12 FROM prix_m2_appartement_t2, prix_m2_appartement_t3;
13
```

Requête 8 :

Différence en pourcentage du prix au mètre carré entre un appartement de 2 pièces et un appartement de 3 pièces : **12.31%**.

Result Grid		Filter Rows: <input type="text"/>	Export: 	Wrap Cell Content: 
	Différence en pourcentage du prix au mètre carré entre un appartement de 2 pièces et un appartement de 3 pièces :			
	-12.31			

```

1 • WITH
2   ValeurFonciereParVille
3   as ( SELECT codeDepartement, commune, avg(valFon) as valFonMoyenne
4   FROM mutation
5   JOIN bien on mutation.ID_Bien = bien.ID_Bien
6   JOIN commune on commune.ID_Commune = bien.ID_Commune
7   WHERE codeDepartement IN (6, 13, 33, 59, 69)
8   GROUP BY commune)
9   SELECT codeDepartement AS "Departement", commune, round(valFonMoyenne,2) AS "valeurs foncières moyennes pour le top 3 des communes des
10  départements 6, 13, 33, 59 et 69"
11  FROM (
12  SELECT codeDepartement, commune, valFonMoyenne,
13  rank () OVER ( PARTITION BY codeDepartement ORDER BY valFonMoyenne DESC) AS classement
14  FROM ValeurFonciereParVille) AS classementparville
15  WHERE classement <=3;

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: [IA](#)

	Departement	commune	valeurs foncières moyennes pour le top 3 des communes des départements 6, 13, 33, 59 et 69
6		SAINT-JEAN-CAP-FERRAT	968750
6		EZE	655000
6		MOUANS-SARTOUX	476898.09
13		GIGNAC-LA-NERTHE	330000
13		SAINT SAVOURNIN	314425
13		CASSIS	313416.88
33		LEGE-CAP-FERRET	549500.64
33		VAYRES	335000
33		ARCACHON	307435.93
59		BERSEE	433202
59		CYSOING	408550
59		HALLUIN	322250
69		VILLE SUR JARNIOUX	485300
69		LYON 2EME	455217.27
69		LYON 6EME	426968.25

Requête 9 :

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