

BIENVENUE AU



Année universitaire 2022-2023

PRICewise PREDICTION



GROUPE



Ghassen El Abed



Farah boukari



Youssef zghal



Ghada Dhaoui

L'OBJECTIF DE PROJET

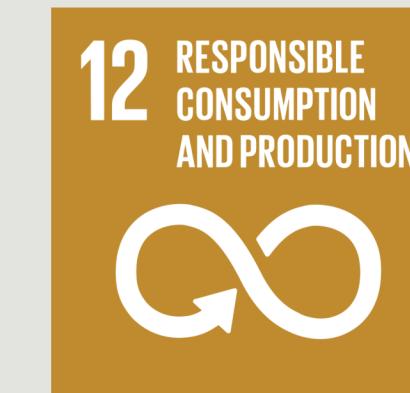
La prédition de la tendance des prix de l'immobilier dans les prochaines années.

PROBLÉMATIQUE

Les fluctuations imprévisibles des prix immobiliers rendent difficile pour les acheteurs potentiels de déterminer la valeur réelle des maisons, entraînant des décisions d'achat peu éclairées.

SOLUTION PROPOSÉE

Le croisement des données qui va être exploité au niveau de ce projet BI va nous aider à **élaborer des tableaux de bords autonomes et développer un modèle prédictif de manière à donner aux acteurs de domaine de l'immobilier la possibilité d'assurer une prise de décision avec une grande flexibilité.**



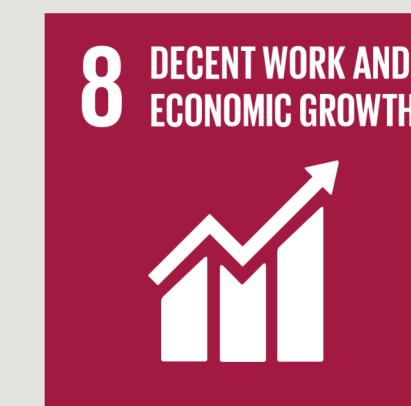
Connaitre le prix reel de immobilisations



Evaluer l'impact énergétique des biens immobiliers proposés



infrastructures durables



croissance économique due au cout raisonnable des logement



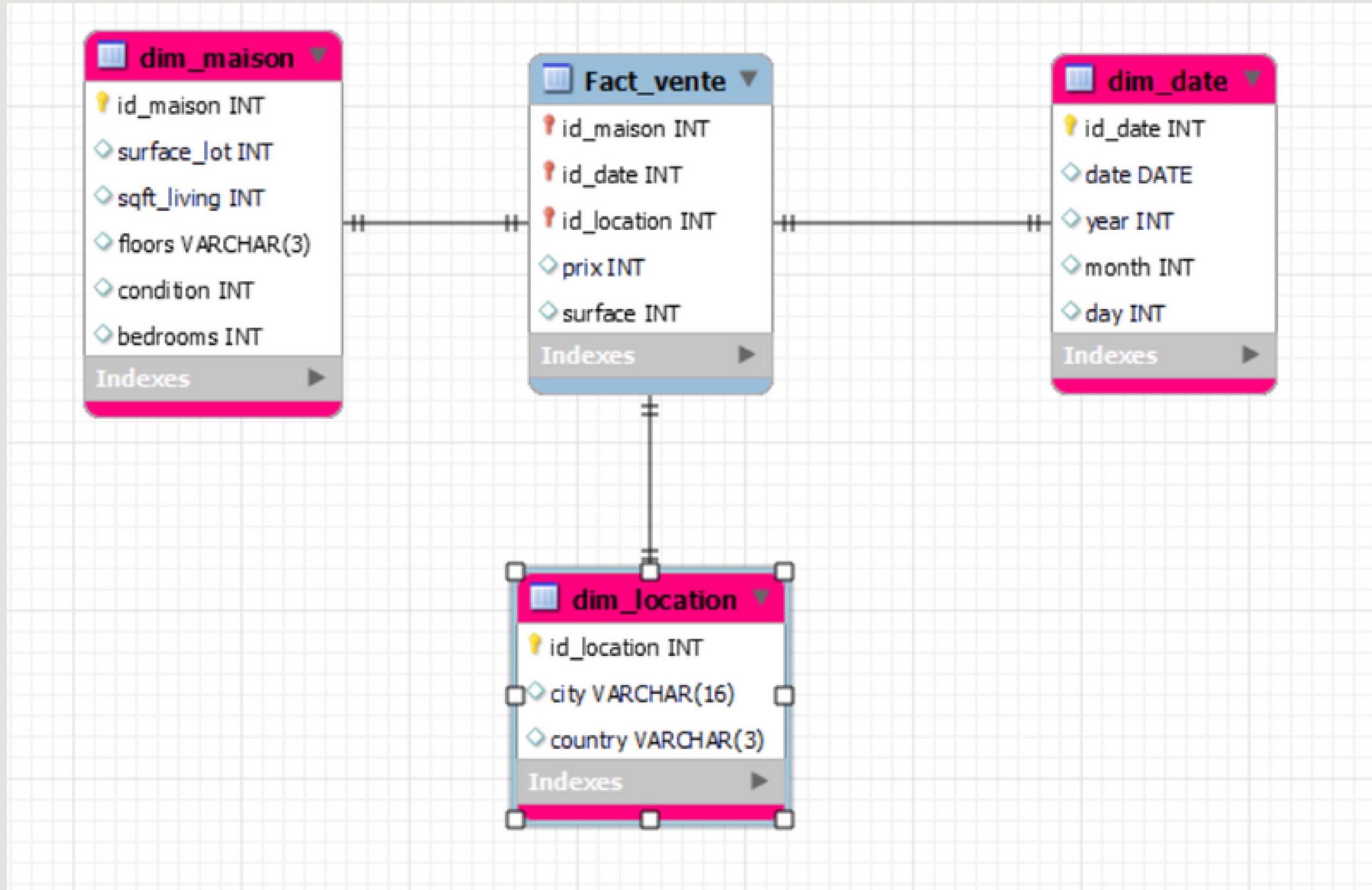
idée innovante grace a l'intellegence artificielle

DESCRIPTION DATASET

DATASET CAPTURE

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
date	price	bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfront	view	condition	sqft_above	sqft_basement	yr_built	yr_renovated	street	city	statezip	country
02-May-14	313000	3	1,5	1340	7912	1,5	0	0	3	1340	0	1955	2005	18810 Densmore Av	Shoreline	WA 98133	USA
02-May-14	2384000	5	2,5	3650	9050	2	0	4	5	3370	280	1921	0	709 W Blaine St	Seattle	WA 98119	USA
02-May-14	342000	3	2	1930	11947	1	0	0	4	1930	0	1966	0	26206-26214 143rd	Kent	WA 98042	USA
02-May-14	420000	3	2,25	2000	8030	1	0	0	4	1000	1000	1963	0	857 170th Pl NE	Bellevue	WA 98008	USA
02-May-14	550000	4	2,5	1940	10500	1	0	0	4	1140	800	1976	1992	9105 170th Ave NE	Redmond	WA 98052	USA
02-May-14	490000	2	1	880	6380	1	0	0	3	880	0	1938	1994	522 NE 88th St	Seattle	WA 98115	USA
02-May-14	335000	2	2	1350	2560	1	0	0	3	1350	0	1976	0	2616 174th Ave NE	Redmond	WA 98052	USA
02-May-14	482000	4	2,5	2710	35868	2	0	0	3	2710	0	1989	0	23762 SE 253rd Pl	Maple Valley	WA 98038	USA
02-May-14	452500	3	2,5	2430	88426	1	0	0	4	1570	860	1985	0	46611-46625 SE 129	North Bend	WA 98045	USA
02-May-14	640000	4	2	1520	6200	1,5	0	0	3	1520	0	1945	2010	6811 55th Ave NE	Seattle	WA 98115	USA
02-May-14	463000	3	1,75	1710	7320	1	0	0	3	1710	0	1948	1994	Burke-Gilman Trail	Lake Forest P	WA 98155	USA
02-May-14	1400000	4	2,5	2920	4000	1,5	0	0	5	1910	1010	1909	1988	3838-4098 44th Ave	Seattle	WA 98105	USA
02-May-14	588500	3	1,75	2330	14892	1	0	0	3	1970	360	1980	0	1833 220th Pl NE	Sammamish	WA 98074	USA

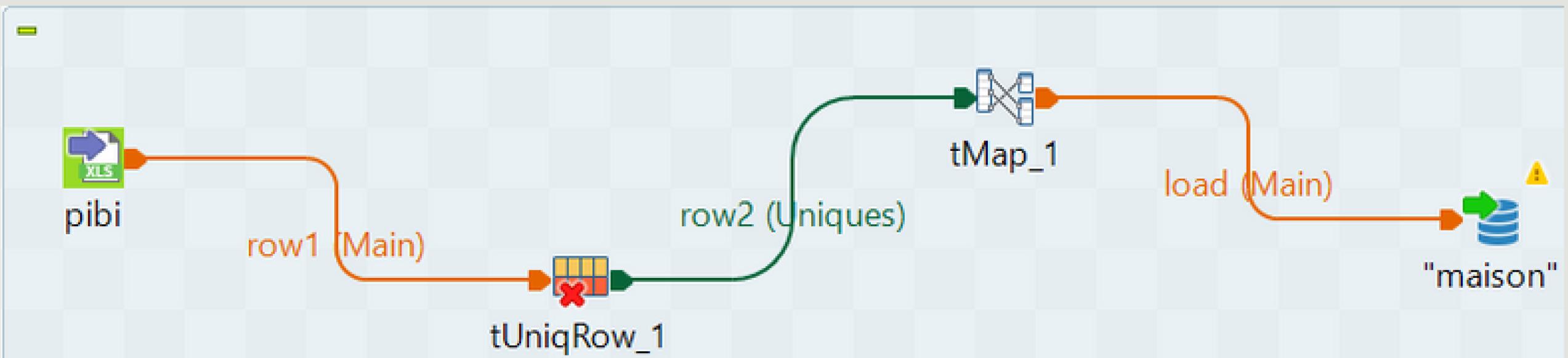
CONCEPTION DE L'ENTREPOT DE DONNÉES



INTEGRATION TALEND



DIMENSION MAISON



Query Query History

```

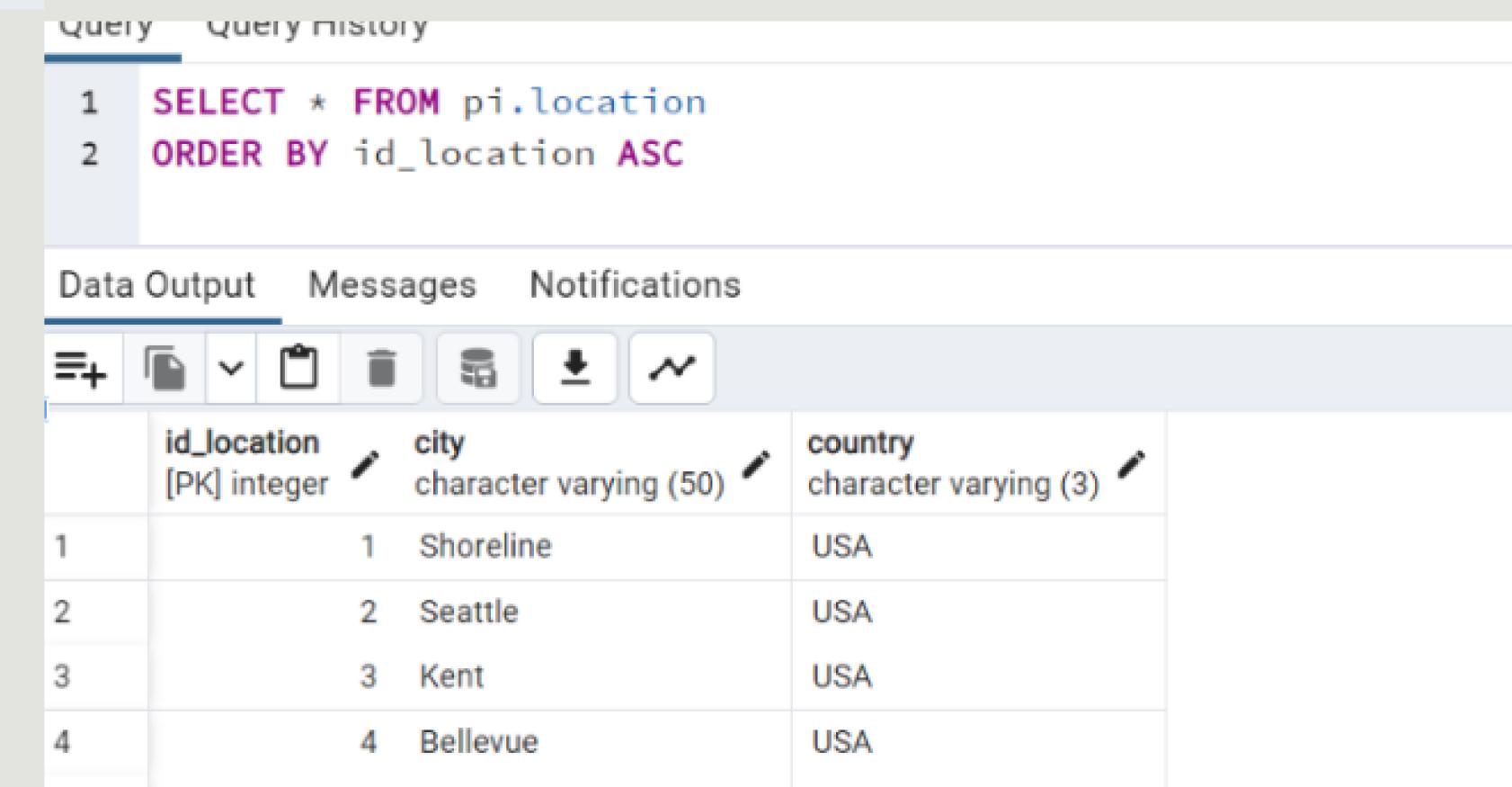
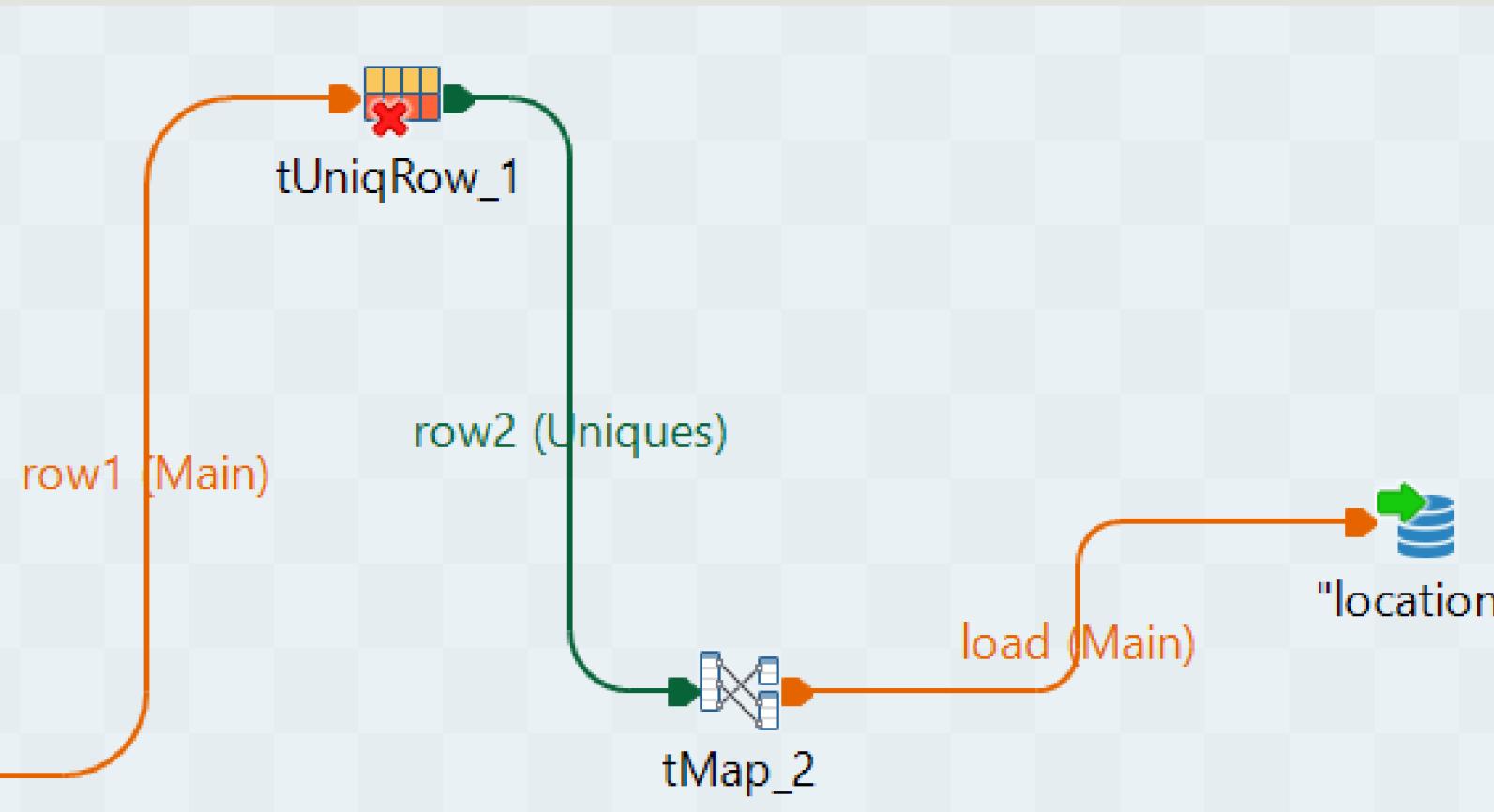
1 SELECT * FROM pi.maison
2 ORDER BY id_maison ASC
  
```

Data Output Messages Notifications



	id_maison [PK] integer	bedrooms double precision	sqft_living double precision	sqft_lot double precision	condition double precision
1	1	3	1340	7912	3
2	2	5	3650	9050	5
3	3	3	1930	11947	4
4	4	3	2000	8030	4
5	5	4	1940	10500	4

DIMENSION LOCATION



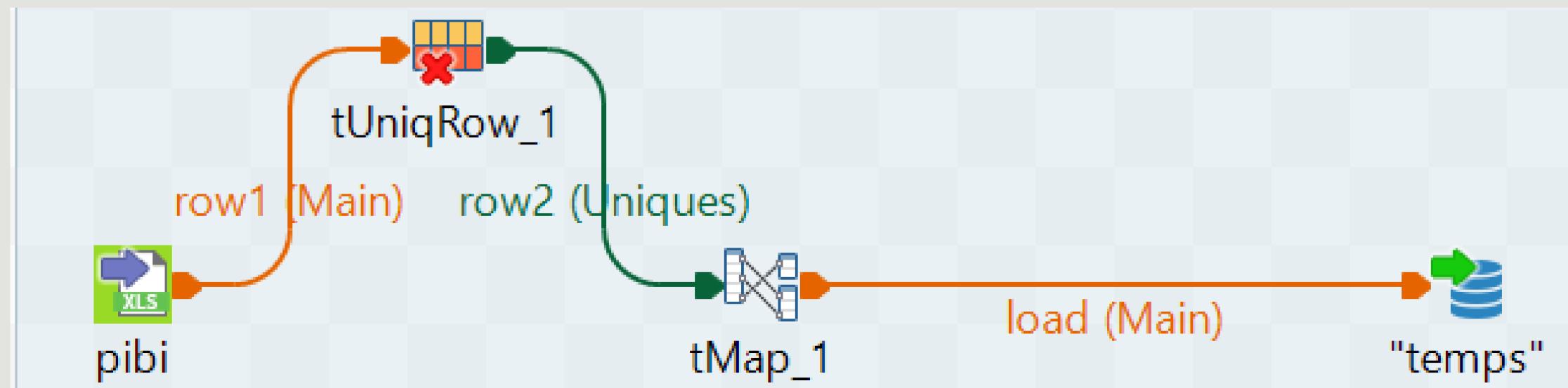
The screenshot shows a database interface with the following details:

- Query:**

```
1 SELECT * FROM pi.location
2 ORDER BY id_location ASC
```
- Data Output:** A table showing the data loaded into the 'location' table.
- Table Headers:** id_location [PK] integer, city character varying (50), country character varying (3)
- Table Data:**

	id_location	city	country
1	1	Shoreline	USA
2	2	Seattle	USA
3	3	Kent	USA
4	4	Bellevue	USA

DIMENSION TEMPS



```
1 SELECT * FROM pi.temps
2 ORDER BY id_temps ASC
```

Data Output Messages Notifications



	id_temps [PK] integer	date date	annee integer	jours integer	mois integer
1		1 2014-05-02	2014	2	4
2		2 2014-05-03	2014	3	4
3		3 2014-05-04	2014	4	4
4		4 2014-05-05	2014	5	4

Query Query History

```
1 SELECT * FROM pi.vente
2
```

Data Output Messages Notifications

≡+ ↻ 📄 🗑️ 🚶 ⏴ ⏵

	id_location integer	id_maison integer	id_temps integer	price numeric	surface numeric
1	1	4061	1	313000	1340
2	2	4253	1	2384000	3650
3	3	3951	1	342000	1930
4	4	4389	1	420000	2000

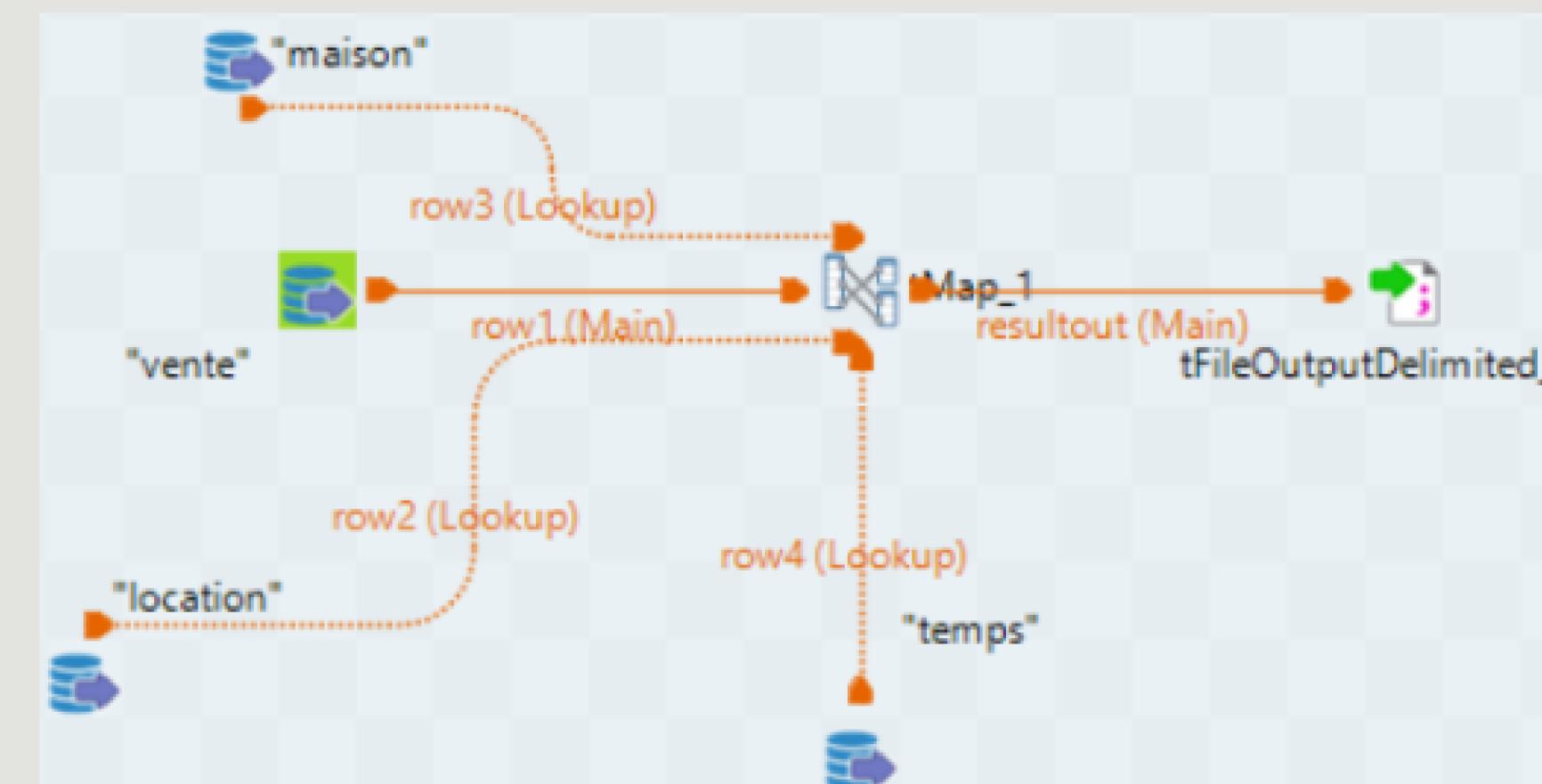


TABLEAU DE BORD



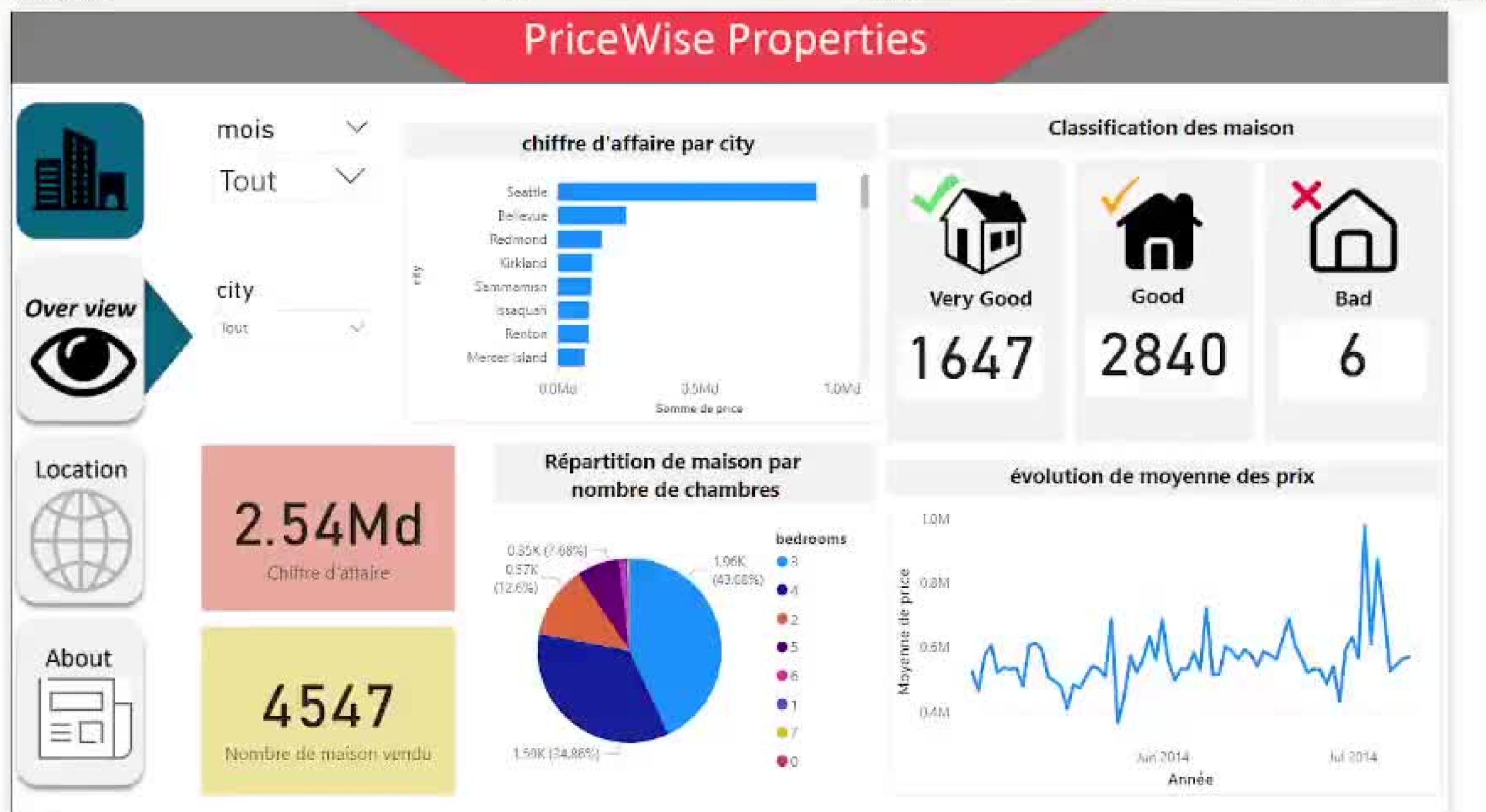
[Fichier](#) [Accueil](#) [Insérer](#) [Modélisation](#) [Afficher](#) [Optimiser](#) [Aide](#)


Obtenir les données [Centre de données](#) [SQL Server](#) [Entrer des données](#) [Sources récentes](#)
[Excel](#) [Données](#)

Transformer les données [Nouveau visuel](#) [Zone de texte](#) [Plus de visuels](#)
[Actualiser](#) [Insérer](#)

Nouvelle mesure [Confidentialité](#) [Publier](#)
mesure rapide [Calculs](#) [Confidentialité](#)

Partager



Visualisations

Générer un élément visuel

Filtres

Rechercher

pi_location
pi_maison
pi_temps
pi_vente

Valeurs

Ajouter des champs de don...

Extraire

Interraport

Garder tous les filtres

Ajouter des champs d'extr...

Activate Windows
Go to Settings to activate Windows.

Page 9

Page 6

Page 11

+

Data Mining



EVALUATION

localhost:8889/notebooks/PIBI-F-1.ipynb

Entrée []: *###evaluation*

Entrée [126]:

```
# Calculer les scores (Score) des modèles
rf_score = mean_squared_error(y_test, rf_pred, squared=False)
xgb_score = mean_squared_error(y_test, xgb_pred, squared=False)
svm_score = mean_squared_error(y_test, svm_pred, squared=False)
lr_score = mean_squared_error(y_test, lr_pred, squared=False)

# Créer un DataFrame pour les scores
scores = pd.DataFrame({
    'Modèle': ['Random Forest', 'XGBoost', 'SVM', 'Régression Linéaire'],
    'Score': [rf_score, xgb_score, svm_score, lr_score]
})

# Afficher le tableau des scores
print(scores)
```

	Modèle	Score
0	Random Forest	348985.088190
1	XGBoost	336744.554451
2	SVM	383036.513302
3	Régression Linéaire	336438.021018

EVALUATION

localhost:8889/notebooks/PIBI-F-1.ipynb

jupyter PIBI-F-1 Dernière Sauvegarde : il y a 2 minutes (auto-sauvegardé)

File Edit View Insert Cell Kernel Widgets Help

Connexion au noyau Non flable Python 3 (pykernel) ⚡

Entrée [129]:

```
# Affichage de la courbe comparative
plt.figure(figsize=(8, 6))
plt.bar(models, rmse_scores)
plt.xlabel('Modèle')
plt.ylabel('RMSE')
plt.title('Comparaison des performances des modèles')

# Recherche du modèle avec le score minimal (le plus efficace)
best_model = models[np.argmin(rmse_scores)]

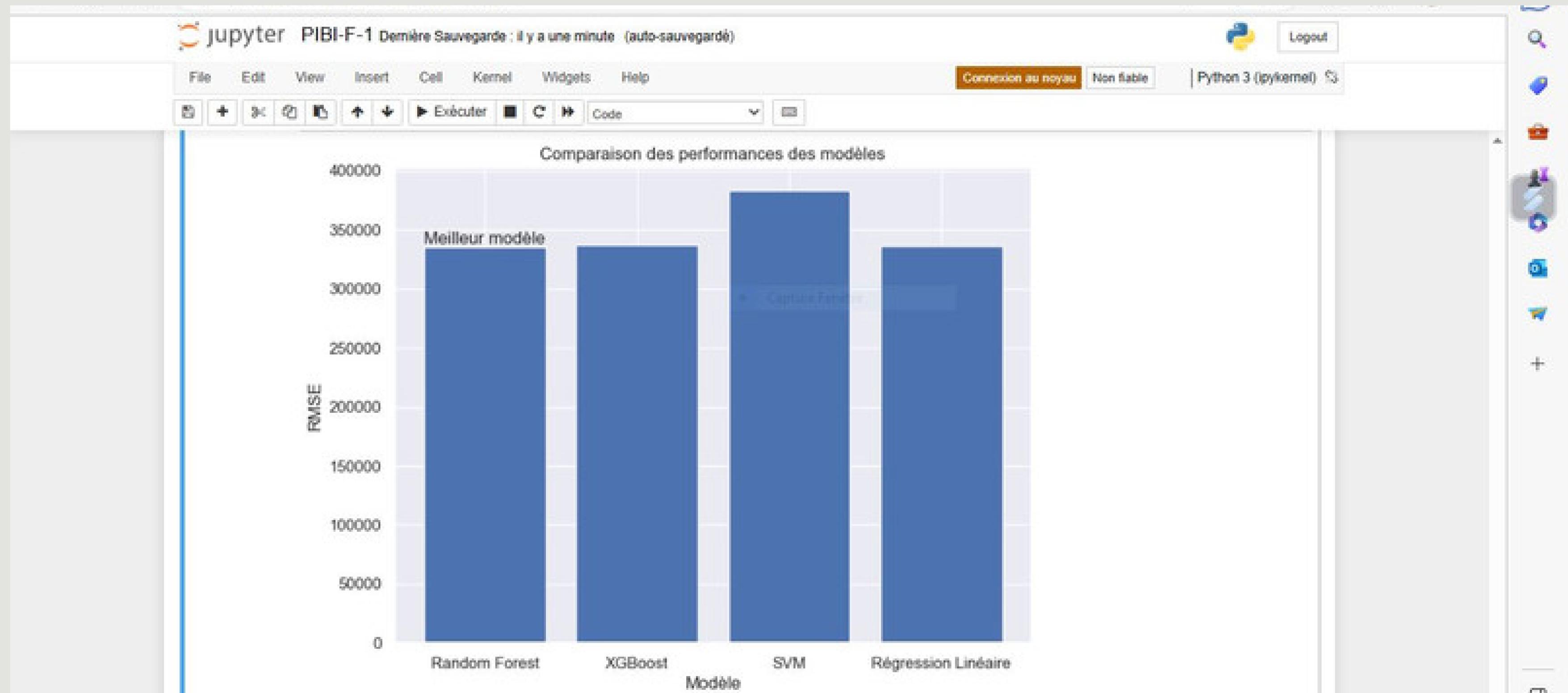
# Affichage de la légende indiquant le modèle le plus efficace
plt.text(best_model, np.min(rmse_scores), 'Meilleur modèle', ha='center', va='bottom')

# Affichage de la courbe
plt.show()
```

Comparaison des performances des modèles

Modèle	RMSE
Meilleur modèle	~320,000
Modèle 1	~340,000
Modèle 2	~380,000
Modèle 3	~310,000

EVALUATION



APPLICATION

Pricewise



House Price Prediction

State:

Choose a state



Number of Bedrooms:

Living Surface Area:

Predict

clideo.com

CONTACT US



Ghassen el abed

Ghassen.elabed@esprit.tn



Farah boukari

Farah.boukari@esprit.tn



Youssel Zghal

Youssef.zghal@esprit.tn



Ghada dhaoui

dhaoui.ghada@esprit.tn





THANK YOU