

Validation SQL des mesures

Ce document contient les requêtes SQL permettant de vérifier les résultats obtenus dans Power BI, ainsi qu'un espace réservé pour insérer les captures d'écran des résultats MySQL.

1. Dette Initiale Totale

DAX équivalent : `SUM(Dette[capital_initial])`

Requête SQL

```
SELECT SUM(capital_initial) AS dette_initiale_totale  
FROM dataset;
```

Résultat MySQL

Result Grid	
▶	dette_initiale_totale 40214100949.82003

2. Dette Totale Restante

DAX équivalent : `SUM(Dette[capital_restant])`

Requête SQL

```
SELECT SUM(capital_restant) AS dette_totale_restante  
FROM dataset;
```

Résultat MySQL

Result Grid	
	Filter Rows
dette_totale_restante	38041898619.62003

3. Nombre d'emprunts

DAX équivalent : `COUNTROWS(Dette)`

Requête SQL

```
SELECT COUNT(*) AS nb_emprunts  
FROM dataset;
```

Résultat MySQL

Result Grid	
nb_emprunts	610

4. Nombre de prêteurs distincts

DAX équivalent : `DISTINCTCOUNT(Dette[preteur])`

Requête SQL

```
SELECT COUNT(DISTINCT organisme_preteur_ou_chef_de_file) AS nb_preteurs  
FROM dataset;
```

Résultat MySQL

Result Grid	
nb_preteurs	61

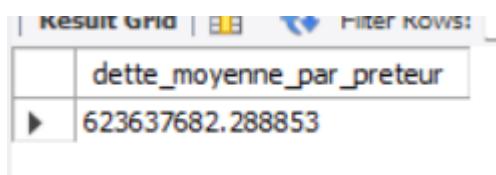
5. Dette Moyenne par Prêteur

DAX équivalent : DIVIDE([Dette Totale Restante], [Nb Prêteurs])

Requête SQL

```
SELECT
    SUM(capital_restant) / COUNT(DISTINCT organisme_preteur_ou_chef_de_file)
    AS dette_moyenne_par_preteur
FROM dataset;
```

Résultat MySQL



dette_moyenne_par_preteur
623637682.288853

6. Part de Marché (%) par prêteur

DAX équivalent : SUM(capital_restant) / SUM(capital_restant ALL PRETEURS)

Requête SQL

```
SELECT
    organisme_preteur_ou_chef_de_file,
    SUM(capital_restant) AS montant,
    SUM(capital_restant) / (SELECT SUM(capital_restant) FROM dataset) * 100 AS
part_marche
FROM dataset
GROUP BY organisme_preteur_ou_chef_de_file
ORDER BY part_marche DESC;
```

Résultat MySQL

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

	organisme_preteur_ou_chef_de_file	montant	part_marche
▶	BEI	3490000000	9.174095212482479
	HSBC-Natixis	3400000000	8.937513960584651
	HSBC	3167623220.9999995	8.326669635164592
	HSBC-Natixis-SociétéGénérale	2375000000	6.243116369526043
	SOCIETE GENERALE	2130000000	5.5990896282486196
	DEXIA CREDIT LOCAL	1994091861.2	5.241830543577421
	BNP Paribas	1976328502.43	5.195136347402789
	Natixis	1806000000	4.7473971214164346
	HSBC-Natixis-Banca IMI	1600000000	4.205888922628072
	Goldman Sachs	1080000000	2.8389750227739476
	Dexia - Morgan Stanley	960000000	2.5235333535768425
	108 - HSBC-SG-CA	920000000	2.418386130511141
	Crédit Suisse	885822501.6000003	2.32854440430883
	Crédit Suisse-BNPP	857959961.8499999	2.255302687252073
	CREDIT FONCIER	818112254.31	2.1505557924179426
	Helaba	780000000	2.0503708497811846
	CCU	722151220.1	1.8000000000000002

7. Calcul du HHI (Herfindahl-Hirschman Index)

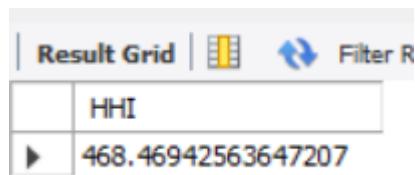
DAX équivalent : `SUMX(VALUES(preteur), (Part Marché %)2)`

Requête SQL

```

SELECT
SUM(part_pct * part_pct) AS HHI
FROM (
    SELECT
        organisme_preteur_ou_chef_de_file,
        (SUM(capital_restant) / (SELECT SUM(capital_restant) FROM dataset)) * 100 AS
part_pct
    FROM dataset
    GROUP BY organisme_preteur_ou_chef_de_file
) AS T;
```

Résultat MySQL



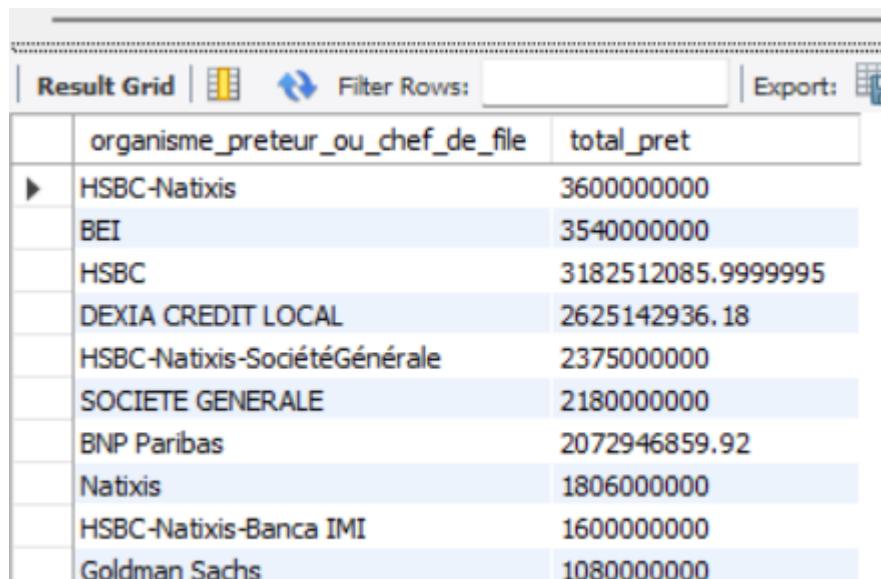
	HHI	
▶	468.46942563647207	

8. Top 10 prêteurs par montant total prêté

Requête SQL

```
SELECT
    organisme_preteur_ou_chef_de_file,
    SUM(capital_initial) AS total_pret
FROM dataset
GROUP BY organisme_preteur_ou_chef_de_file
ORDER BY total_pret DESC
LIMIT 10;
```

Résultat MySQL



	organisme_preteur_ou_chef_de_file	total_pret
▶	HSBC-Natixis	3600000000
	BEI	3540000000
	HSBC	3182512085.9999995
	DEXIA CREDIT LOCAL	2625142936.18
	HSBC-Natixis-SociétéGénérale	2375000000
	SOCIETE GENERALE	2180000000
	BNP Paribas	2072946859.92
	Natixis	1806000000
	HSBC-Natixis-Banca IMI	1600000000
	Goldman Sachs	1080000000

9. Nombre d'emprunts par prêteur

Requête SQL

```

SELECT
    organisme_presteur_ou_chef_de_file,
    COUNT(*) AS nb_emprunts
FROM dataset
GROUP BY organisme_presteur_ou_chef_de_file
ORDER BY nb_emprunts DESC;

```

Résultat MySQL

	organisme_presteur_ou_chef_de_file	nb_emprunts
▶	HSBC	68
	BEI	46
	DEXIA CREDIT LOCAL	38
	SOCIETE GENERALE	37
	C.D.C.	34
	Natixis	31
	Crédit Suisse	28
	BNP Paribas	26
	Helaba	23
	HSBC-Natixis	20
	S.I.A.A.P.	20
	HSBC-Natixis-SociétéGénérale	16
	92 - HSBC	15
	CREDIT FONCIER	15
	Goldman Sachs	14
	CREDIT AGRICOLE IDF	14
	Other Nat C.F	10

10. Évolution annuelle de la dette totale (capital initial)

Utilisée pour valider la courbe dans Power BI

Requête SQL

```

SELECT
    YEAR(date_d_emission) AS annee,
    SUM(capital_initial) AS dette_initiale_annuelle
FROM dataset
GROUP BY YEAR(date_d_emission)
ORDER BY annee;

```

Résultat MySQL

The screenshot shows a MySQL Workbench interface with a result grid titled 'Result Grid'. The grid displays two columns: 'annee' (Year) and 'dette_initiale_annuelle' (Annual Initial Debt). The data is sorted by year from 1996 to 2015. The debt values are listed in scientific notation. A status bar at the bottom indicates 'Result 39'.

	annee	dette_initiale_annuelle
	1996	541194011.18
	1998	15244901.72
	2001	378073562.74
	2002	1570000000
	2003	965000000
	2004	750000000
	2005	4619110176.019999
	2006	4100000000
	2007	3746912751.759999
	2008	3303288846.500005
	2009	2472435000
	2010	2720000000
	2011	2667000000
	2012	2820000000
	2013	3150000000
	2014	1880000000
	2015	2241500000

11. Evolution annuelle de la dette restante

Requête SQL

```
SELECT
    annee_de_publication,
    SUM(capital_restant) AS dette_restante
FROM dataset
GROUP BY annee_de_publication
ORDER BY annee_de_publication;
```

Résultat MySQL

| Result Grid | Filter Rows:

	annee	dette_restante
	1996	396367444.81000006
	1998	0
	2001	76224508.62
	2002	1340000000
	2003	890000000
	2004	600000000
	2005	4087602953.529999
	2006	3750000000
	2007	3393102501.599999
	2008	3303288846.500005
	2009	2472435000
	2010	2720000000
	2011	2667000000
	2012	2820000000
	2013	3150000000
	2014	1880000000
	2015	2241500000