

EXAMEN DE BUSINESS INTELLIGENCE

Membres du groupe 2 :

- **Eya BENALAYA**
- **Ibrahima CAMARA**
- **Samsidine DIATTA**
- **Papa Ba GAYE**
- **Oumar KANE**

Professeur :

Mr. BOLY

Table des matières

FORMULES ET VISUELS DES RAPPORTS DE L'EXERCICE 1	3
1er rapport	3
2 ^e me rapport	4
3 ^e me rapport	5
4 ^e me rapport	6
5 ^e me rapport	7
REPONSES AUX QUESTIONS DE L'EXERCICE 2	8
REQUETES DE L'EXERCICE 3	9

FORMULES ET VISUELS DES RAPPORTS DE L'EXERCICE 1

1er rapport

Citystate = `CONCATENATE`(`DimGeography`[City], `CONCATENATE`(" ",
`DimGeography`[StateProvinceName]))

SumofSalesAmount = `SUM`(`FactInternetSales`[SalesAmount])

Nom Etat Province
Tout

Montant Total Vente(s) par Ville

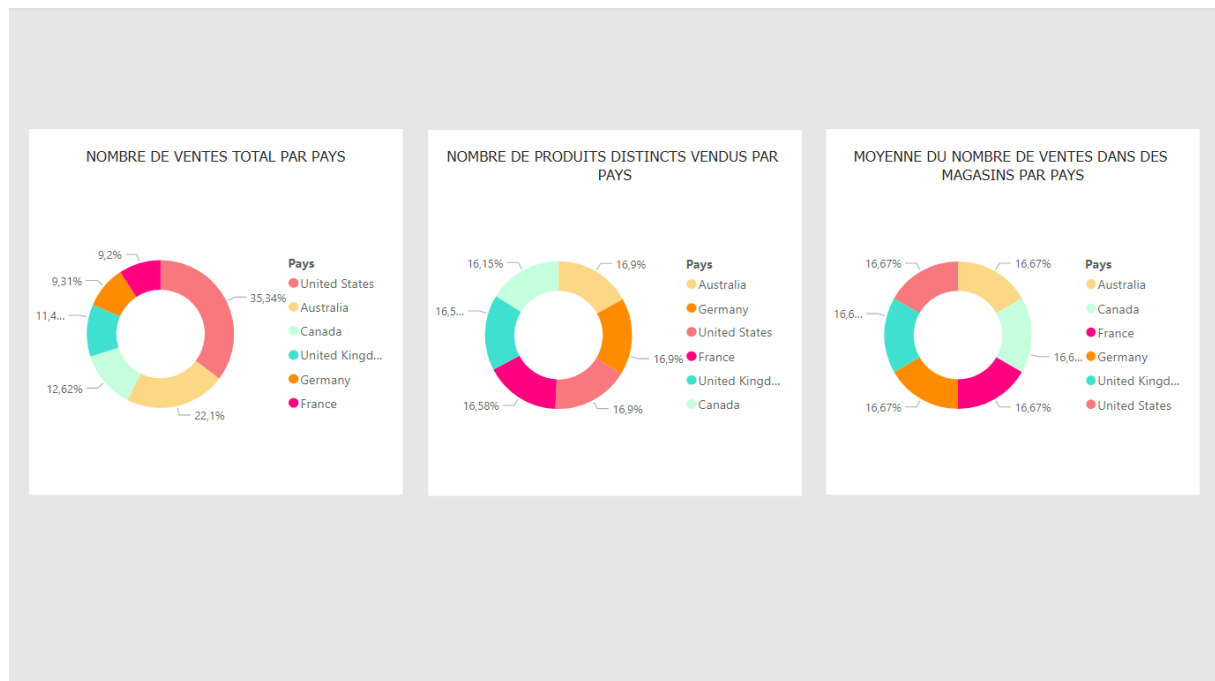
Ville_Etat	Somme montant vente
Byron Georgia	5 €
Chehalis Washington	5 €
Burbank Ohio	7 €
Carol Stream Illinois	7 €
Charlotte North Carolina	7 €
Baytown Texas	25 €
Clarkston Georgia	29 €
Chalk Riber Ontario	37 €
Birmingham Alabama	37 €
Bradenton Florida	39 €
Chantilly Virginia	40 €
Total	29 358 677 €

2^eme rapport

NumberOfSales = `COUNT`(FactInternetSales[OrderQuantity])

NumberOfUniqueProducts = `DISTINCTCOUNT`(FactInternetSales[ProductKey])

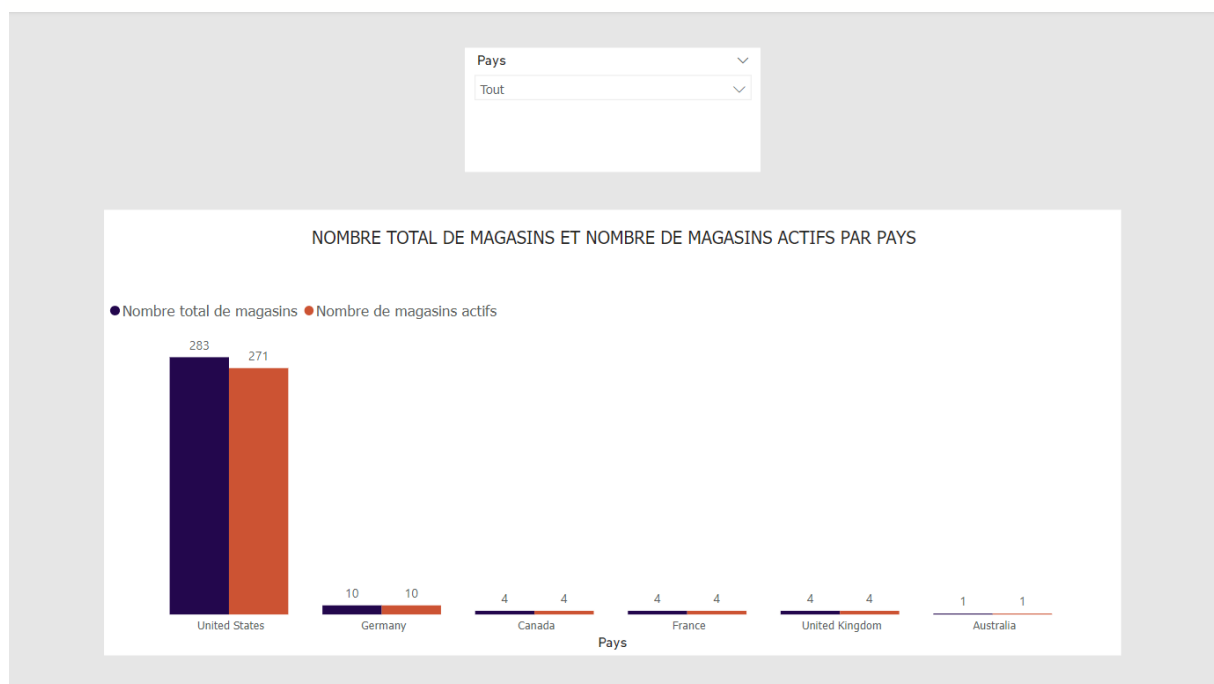
AverageSalesCountinStore = `AVERAGE`(FactInternetSales[OrderQuantity])



3^eme rapport

CountofAllStores = `COUNT(DimStore[StoreKey])` (from dimStore)

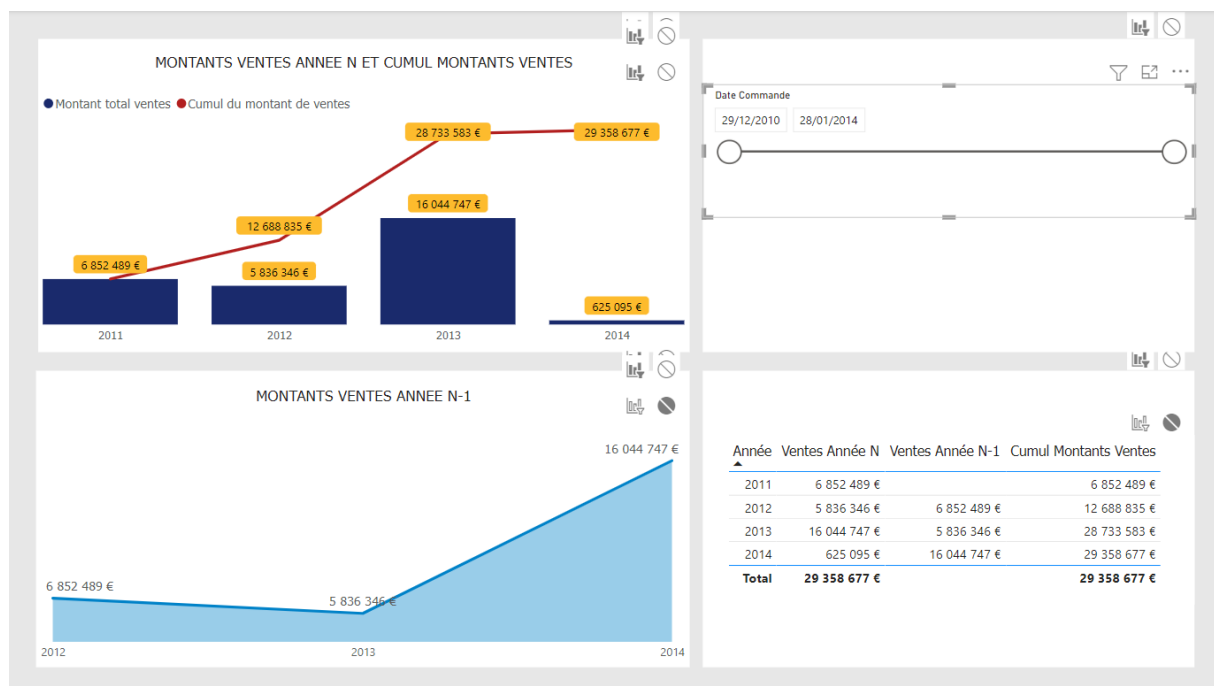
CountofActiveStore = `CALCULATE([CountofAllStores], DimStore[Status] = "On")`
) (from dimStore)



4^eme rapport

```
SumofSalesAmountToDate =
var DateEnd = MAX(DimDate[FullDateAlternateKey].[Date])
return
IF(FactInternetSales[SumofSalesAmount] > 0,
    CALCULATE(
        FactInternetSales[SumofSalesAmount],
        FILTER(ALLSELECTED(DimDate), DimDate[FullDateAlternateKey].[Date] <= DateEnd)))
```

```
SumofSalesAmountPrevYear =
IF(FactInternetSales[SumofSalesAmount]>0,
    CALCULATE(FactInternetSales[SumofSalesAmount],
        PREVIOUSYEAR(DimDate[FullDateAlternateKey].[Date])))
```



5^eme rapport

YearToDateofSales =

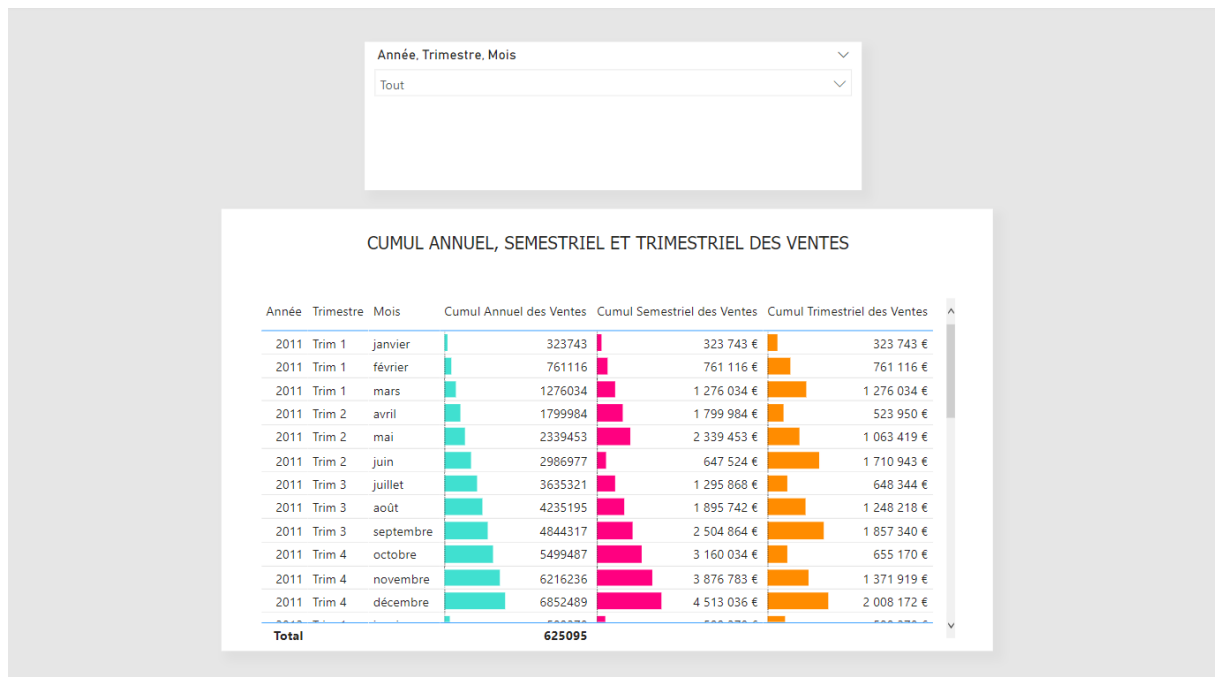
```
IF(
    FactInternetSales[SumofSalesAmount]>0,
    CALCULATE(FactInternetSales[SumofSalesAmount],
DATESYTD(DimDate[FullDateAlternateKey].[Date])))
```

SemestrialTotalofSales =

```
var day = MAX(DimDate[FullDateAlternateKey].[Date])
return
IF(
    FactInternetSales[SumofSalesAmount] > 0,
    IF(
        day >= DATE(year(day), 06, 01),
        CALCULATE(FactInternetSales[SumofSalesAmount],
DATESBETWEEN(DimDate[FullDateAlternateKey].[Date], DATE(YEAR(day), 06, 01), day)),
        CALCULATE(FactInternetSales[SumofSalesAmount],
DATESBETWEEN(DimDate[FullDateAlternateKey].[Date], DATE(YEAR(day), 01, 01), day))
    )
)
```

QuarterlyTotalofSales =

```
IF(
    FactInternetSales[SumofSalesAmount] > 0,
    CALCULATE(FactInternetSales[SumofSalesAmount],
DATESQTD(DimDate[FullDateAlternateKey].[Date])))
```



REPONSES AUX QUESTIONS DE L'EXERCICE 2

1. Le schéma en étoile car il comporte moins de jointures et donc les requêtes prennent moins de temps à s'exécuter.
2. Une table de faits comporte plus d'enregistrements qu'une table de dimension car :
 - Les tables de dimension doivent être créées en premier et se développent horizontalement (par colonne). Tous les enregistrements nécessaires sont généralement effectués lors de leur création.
 - La table de faits se développe verticalement (par ligne). Le nombre d'enregistrement ne cesse de croître.
3. Les tables de dimensions contiennent des paramètres qui correspondent aux informations faisant varier les mesures de l'activité qui sont au niveau de la table de faits.
4. La clé primaire est généralement un nombre que l'on incrémente de 1 à chaque nouvel enregistrement et qui l'identifie de manière unique alors que l'attribut CodeProduit (unique aussi) est de type chaîne de caractères et donc n'est pas incrémentale.

REQUETES DE L'EXERCICE 3

Question 1

```
-- Requête SQL de chargement de la table Produit à partir de la table BNNE.
INSERT INTO Produit (CodeProduit)
SELECT DISTINCT CodeProduit
FROM BNNE

-- Affichage de la table produit
SELECT *
FROM [Produit]
```

Question 2

```
-- Requête SQL de chargement de la table Vente à partir de la table BNNE.
--     INSERTION DE LA LIGNE NumLigne dans la table BNNE

ALTER TABLE BNNE add NumLigne INT Identity(1,1);
```

```
-- DECLARATION DES VARIABLES
DECLARE
@i INTEGER,
@taille INTEGER

-- INITIALISATION DES VARIABLES
SET @i = 1;
SET @taille = (SELECT count(*) FROM BNNE);

-- BOUCLE D'INSERTION
WHILE @i <= @taille
BEGIN
    BEGIN TRY
        INSERT INTO Vente (IdProduit, IdDate, IdClient, IdVendeur, PrixUnitaire,
Quantité)
        SELECT IdProduit, IdDate, IdClient, IdVendeur, PrixUnitaire, Quantité
        FROM BNNE, Produit, Vendeur, Client, Date
        WHERE BNNE.CodeProduit = Produit.CodeProduit
```

```

        AND BNNE.nomClient = Client.nomClient
        AND BNNE.date = Date.date
        AND BNNE.matricule = Vendeur.matricule
        AND BNNE.NumLigne = @i
    END TRY
    BEGIN CATCH
        PRINT 'ERREUR POUR LA LIGNE ' + @i
    END CATCH
    SET @i = @i + 1
END

-- AFFICHAGE DE LA TABLE VENTE
SELECT *
    FROM [Vente]

```