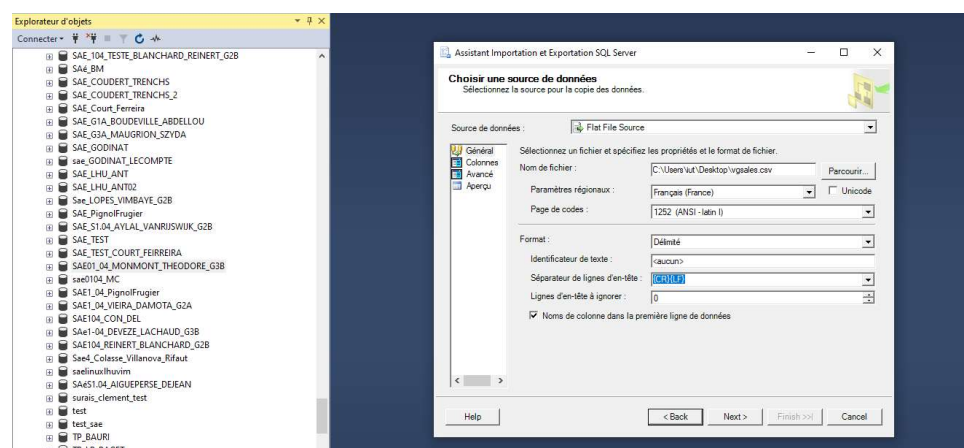
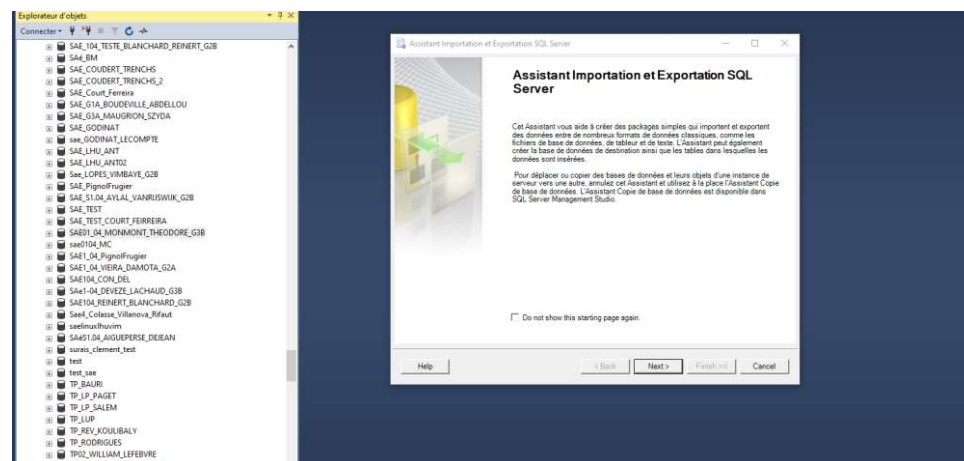


# Sommaire :

- Phase 1 : Importation et gestion de vgsales
- Création des tables et alimentation
- Requêtes

Nous avons importé le fichier vgsales a l'aide de l'outil d'import de management studio :



Ensuite, nous avons changé chaque valeur en varchar max pour bien inserer les données .

Le fichier importé , nous avons du chercher differents problemes puis les résoudre :

-Le premier problème à régler était un problème de décalage . Le rang a été incrémenté de 2 au lieu de 1 , et ceux 2 fois . Nous avons détecté ce problème à l'aide de la commande ROWNUMBER qui permet d'afficher les lignes , et nous avons vu que le rang ne correspondait pas au numéro de la ligne

declare @i int

set @i = 0

UPDATE vgsales set rank = @i, @i = @i + 1

pour régler le problème , nous avons déclaré une variable que l'on a incrémenté de 1 à chaque nouvelle ligne .

-certaine colonne avait la valeur null , nous avons donc changé les valeurs en 0

Phase 2 :

Nous avons donc crée 2 table , une jeux et une vente :

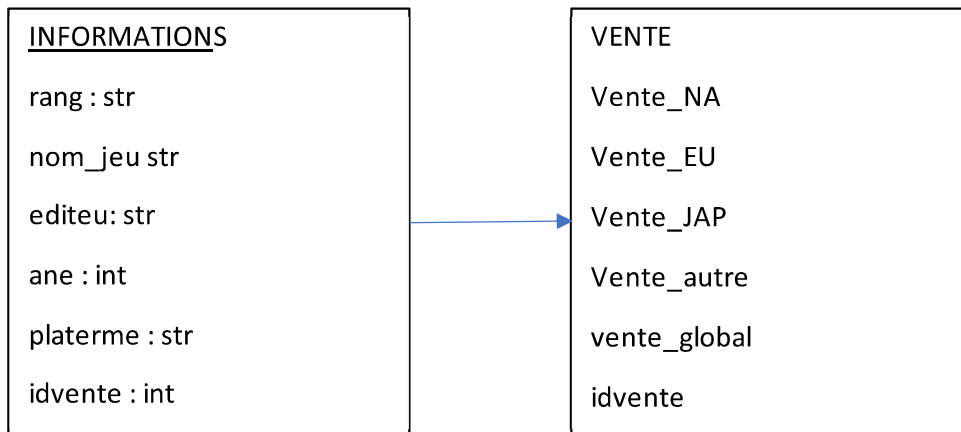


table Information:

```
CREATE TABLE INFORMATIONS(  
    rang int PRIMARY key,  
    nom_jeu varchar(150),  
    editeur VARCHAR(40),  
    annee smallint ,  
    genre varchar(30),  
    plateforme varchar(50),  
    idVente int FOREIGN KEY (idVente) REFERENCES VENTE(idVente) IDENTITY(1, 1)  
);
```

Nous avons utilisé comme clé primaire le rang du jeu .

Commande pour inserer les données :

```
insert into informations (rang, nom_jeu, editeur, annee, genre, plateforme) select rank, name,  
Publisher, year, Genre, Platform from vgsales
```

Table VENTE :

clé primaire faite avec IDENTITY

```
CREATE TABLE VENTE(  
    vente_NA decimal(4,2),  
    vente_EU DECIMAL(4,2),  
    vente_JAP DECIMAL(4,2),  
    vente_autre DECIMAL(4,2),  
    vente_global DECIMAL(4,2),  
    idVente int PRIMARY key not null IDENTITY(1,1)  
);
```

Ensuite , nous avons alimenter la table VENTES avec cette commande :

```
insert into vente (vente_NA, vente_EU, vente_JAP, vente_autre, vente_global) select NA_sales,  
EU_Sales, JP_Sales, Other_Sales, Global_Sales from vgsales
```

Nous avons donc inserer dans les differentes colonne des chaine de caractere qui se sont changer en decimal (2 , 2) comme nous l'avons choisi lors de la création de la table

Phase 3 :

1. Affichez la liste des plateformes ayant plus de 20M de jeux vendus en Europe depuis 2010.

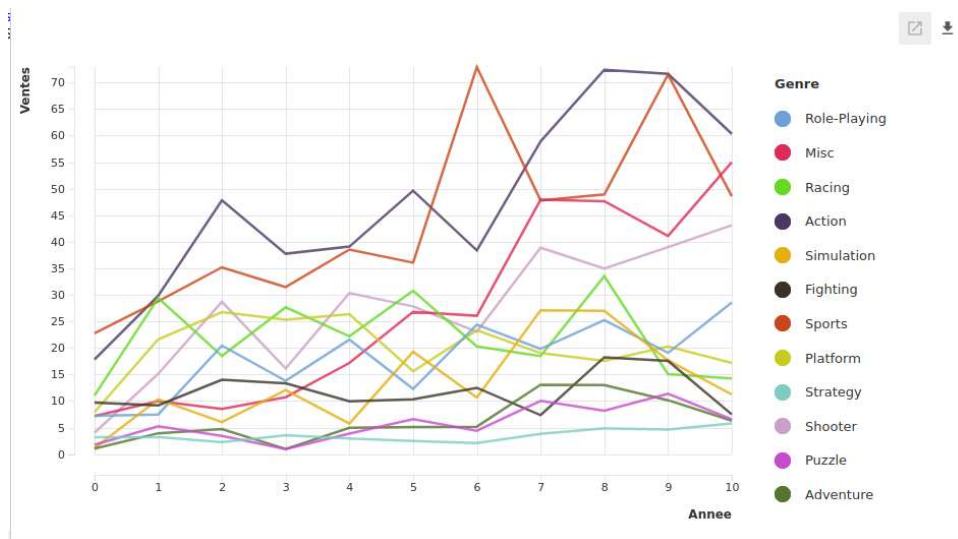
```
select plateforme from INFORMATIONS  
  
join vente on INFORMATIONS.idVente = vente.idVente  
  
where annee >= 2010  
  
group by plateforme  
  
HAVING sum(vente_EU) > 20
```

2 Affichez la liste des éditeurs de jeux PS4 dépassant les 2M de jeux vendus quelle que soit l'année, le résultat sera trié par nombre de jeux vendus du plus grand au plus petit.

```
select editeur, sum(vente_global) from INFORMATIONS  
  
join vente on INFORMATIONS.idVente = vente.idVente  
  
where plateforme = 'PS4'  
  
group by editeur  
  
having sum(vente_global) > 2  
  
order by sum(vente_global) desc
```

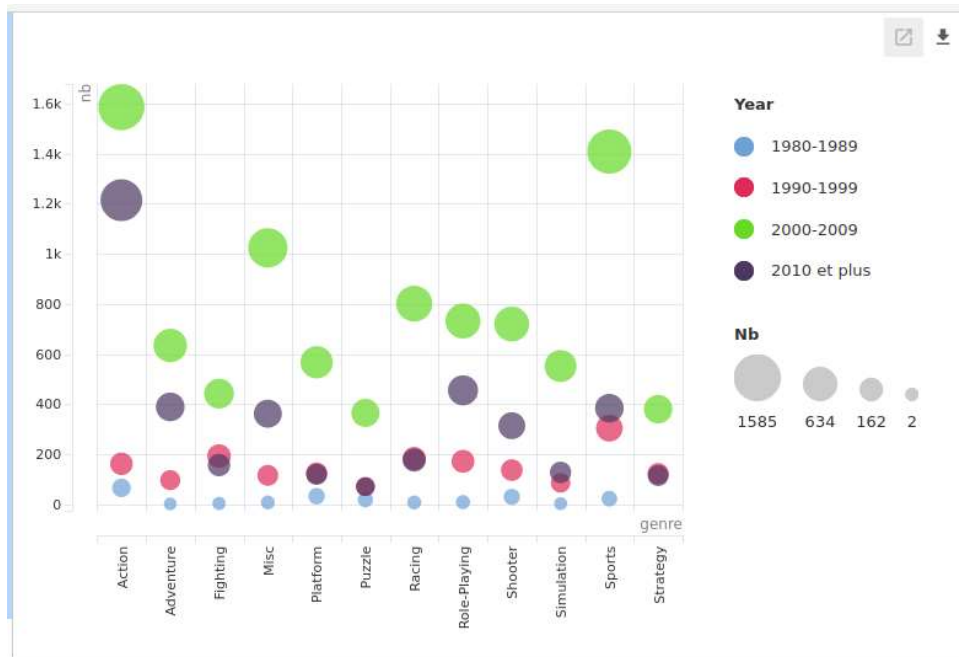
3 Évolution du nombre de ventes de jeux par genre en Amérique du Nord sur la période 2000 à 2010. Pour cette requête vous construirez le graphique suivant :

```
SELECT annee-2000 as annee,genre,SUM(vente_NA) as Ventes FROM VENTE  
JOIN INFORMATIONS ON VENTE.idVente = INFORMATIONS.idVente  
WHERE annee >= 2000 and annee <= 2010  
GROUP BY annee,genre -- ORDER BY annee-2000;
```



4 Afficher le nombre de jeux édités en fonction du genre par tranche de 10 ans, soit 4 tranches : de 1980 à 1989, de 1990 à 1999, de 2000 à 2009, 2010 à maintenant. Pour cette requête vous construirez le graphique suivant :

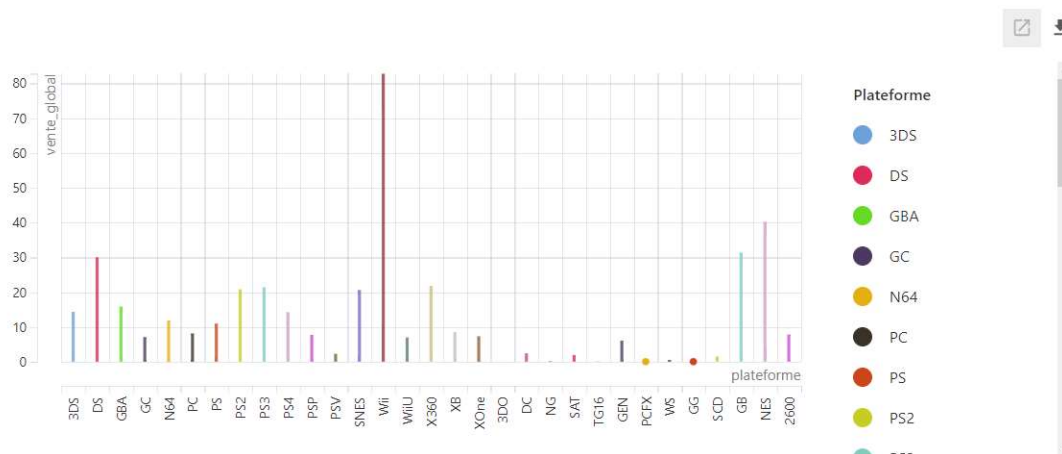
```
select SUM(nbventes) as nb, genre, year FROM
(select count(nom_jeu) as nbventes, genre, '1980-1989' as year
from INFORMATIONS -- where annee between 1980 and 1989
group by genre, annee) as t
group by genre, year
union
select SUM(nbventes) as nb, genre, year FROM
(select count(nom_jeu) as nbventes, genre, '1990-1999' as year
from INFORMATIONS
where annee between 1990 and 1999
group by genre, annee) as t
group by genre, year
union
select SUM(nbventes) as nb, genre, year FROM
(select count(nom_jeu) as nbventes, genre, '2000-2009' as year
from INFORMATIONS
where annee between 2000 and 2009
group by genre, annee) as t
group by genre, year
order by genre
select SUM(nbventes) as nb, genre, year FROM
(select count(nom_jeu) as nbventes, genre, '2010 et plus' as year
from INFORMATIONS
where annee > 2010
group by genre, annee) as t
group by genre, year
```



5 Proposez au moins 2 requêtes supplémentaires d'analyse des données qui vous semblent intéressantes. Créez les graphiques adéquats correspondants.

Premier graphique :

```
select plateforme, vente_global from INFORMATIONS
join VENTE on INFORMATIONS.idVente = VENTE.idVente
group by plateforme, vente_global
order by vente_global
```



deuxieme graphique :

```
select annee-2000 as annee, sum(vente_global) as ventes, editeur from INFORMATIONS
```

```
join VENTE on INFORMATIONS.idVente = VENTE.idVente
```

```
where annee > 2000 and annee <= 2015 and editeur = 'Nintendo'
```

```
group by annee, editeur
```

```
éorder by annee-2000
```

