

Compte rendu S.A.E R1.04

Phase 1 :

1. Description de la méthode utilisée pour obtenir la table demandée :
 - I. Création de la base de données
 - II. Création de la première table nommé vgsales
2. Analyse détaillé des données colonne par colonne :
 - I. Description de chaque colonne
 - II. Requêtes :
 - Champs vides
 - Données N/A ou Unknown
 - Redondance

Phase 2 :

1. Schéma de la base de données
2. Créations des tables
3. Insertion des données :
 - Requête d'insertion de données dans la table "Jeux"
 - Requête qui enlève les années "N/A" en valeur "NULL" et défini la colonne "Année" de type entier
 - Requête d'insertion de données dans la table "VENTES"
 - Requête permettant de supprimer la table vgsales

Phase 3 :

1. Installation de SQLPad sur les ordinateurs de l'IUT
2. L'ensemble des requêtes SQL et leurs résultats et les graphiques:
 - 1 - Affichez la liste des plateformes ayant plus de 20M de jeux vendus en Europe depuis 2010
 - 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
 - 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
 - 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
 - 5 - Proposez au moins 2 requêtes supplémentaires d'analyse des données qui vous semblent intéressantes. Créez les graphiques adéquats correspondants

Phase 1 :

1. Description de la méthode utilisée pour obtenir la table demandée :

I. Création de la base de données :

Nous avons utilisé le fichier CSV en lien ci-dessous. Ce fichier contient des données sur les ventes de jeux vidéo :

<https://www.kaggle.com/datasets/gregorut/videogamesales>.

Une fois sur la page, il suffit de procéder au téléchargement du fichier.

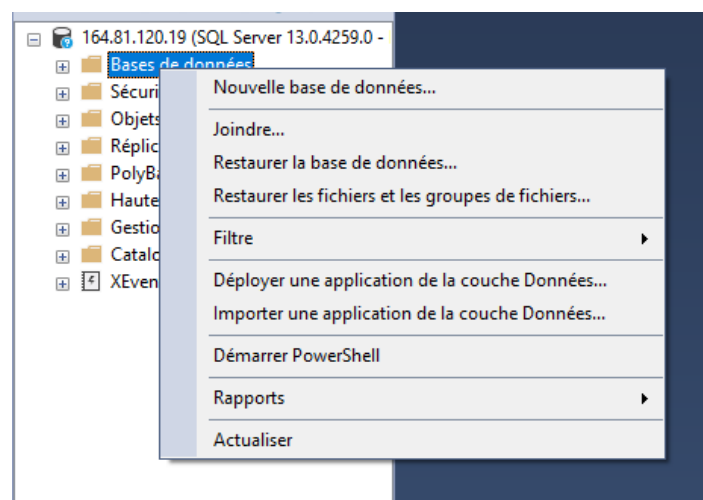
Nous travaillerons ensuite avec le logiciel Microsoft SQL Server Management Studio.

Pour commencer, ouvrez le logiciel et connectez-vous au serveur qui hébergera la base de données. Les informations de connexion à utiliser dans ce cas sont :

- Nom du serveur : 164.81.120.19
- Identifiant : BUT1
- Mot de passe : IUT

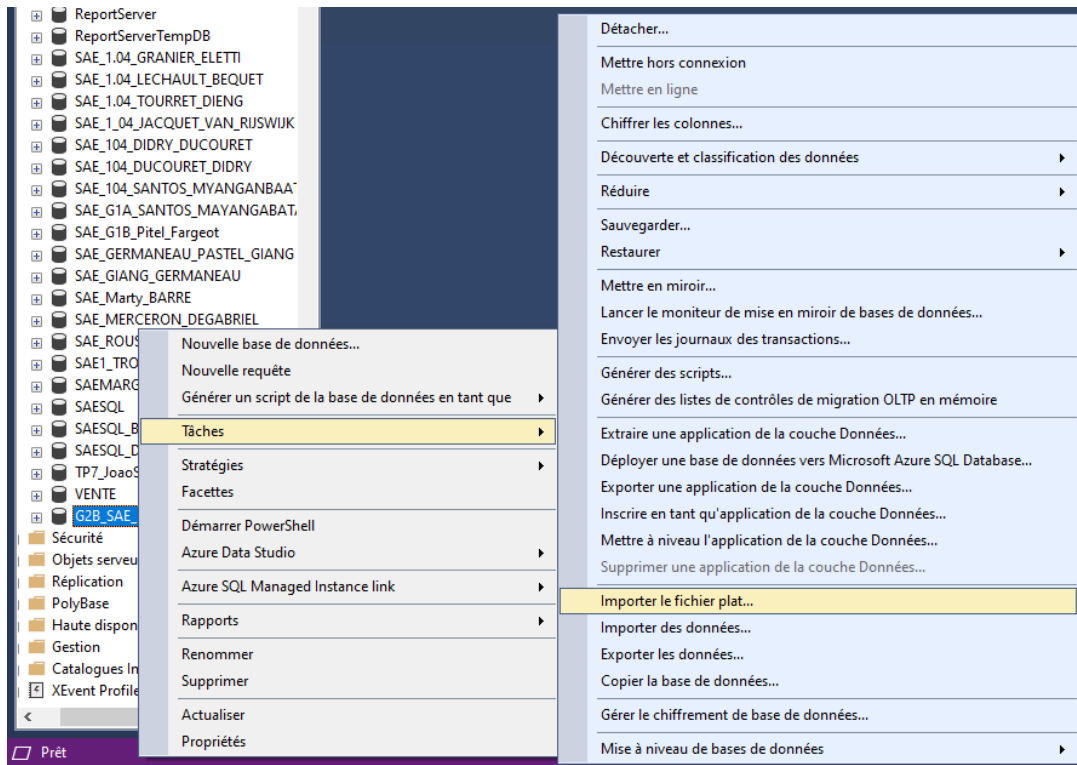
Il faut s'assurer que le type de serveur sélectionné est bien "Moteur de base de données" et que l'authentification choisie est celle de SQL Server.

Une fois connecté, vous pouvez créer une nouvelle base de données en effectuant un clic droit sur le dossier "Bases de données", puis en sélectionnant "Nouvelle base de données". Nous avons ensuite nommé notre base de données "G2B_SAE1.04_BETTON_BASTIDE_SACRISTAN".

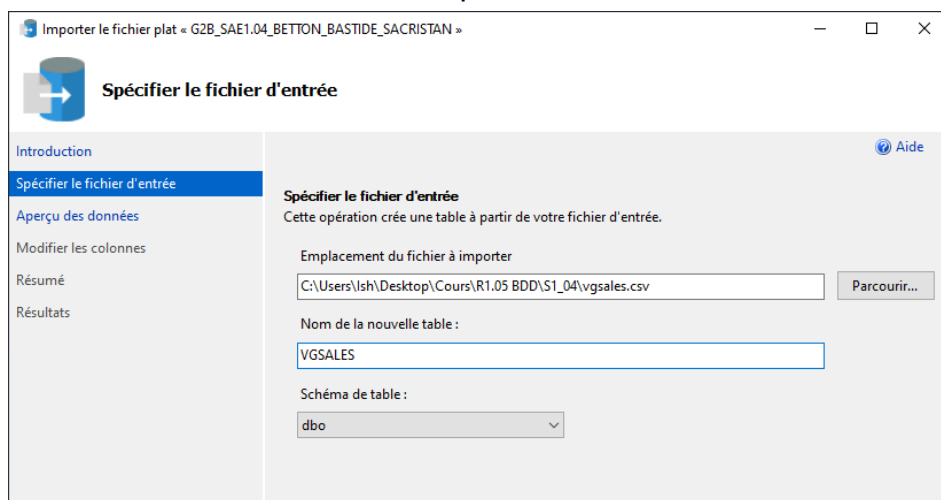


I. Création de la première table nommée vgsales

Ensuite, nous avons inséré les données brutes du fichier CSV sous forme de chaînes de caractères dans une unique table pour pouvoir analyser ses données. Pour cela, il faut faire un clic droit sur notre base de données, puis sélectionner “importer le fichier plat” dans l’onglet “Tâches” :



Ensuite, il faudra sélectionner le fichier CSV précédemment téléchargé et le nommer “vgsales” avec le schéma de table correspondant dans notre cas “dbo”.



Nous pouvons obtenir un aperçu des données de cette table :

Importer le fichier plat « G2B_SAE1.04_BETTON_BASTIDE_SACRISTAN »

Aperçu des données

Introduction
Spécifier le fichier d'entrée
Aperçu des données
Modifier les colonnes
Résumé
Résultats

Aperçu des données
Cette opération a analysé la structure du fichier d'entrée pour générer l'aperçu ci-dessous pour les 50 premières lignes.

Rank	Name	Platform	Year	Genre	Publish
1	Wii Sports	Wii	2006	Sports	Nintend
2	Super Mario Br...	NES	1985	Platform	Nintend
3	Mario Kart Wii	Wii	2008	Racing	Nintend
4	Wii Sports Resort	Wii	2009	Sports	Nintend
5	Pokemon Red/...	GB	1996	Role-Playing	Nintend
6	Tetris	GB	1989	Puzzle	Nintend
7	New Super Mar...	DS	2006	Platform	Nintend
8	Wii Play	Wii	2006	Misc	Nintend
9	New Super Mar...	Wii	2009	Platform	Nintend
10	Duck Hunt	NES	1984	Shooter	Nintend
11	Nintendogs	DS	2005	Simulation	Nintend
12	Mario Kart DS	DS	2005	Racing	Nintend
13	Pokemon Gold...	GB	1999	Role-Playing	Nintend
14	Wii Fit	Wii	2007	Sports	Nintend
15	Wii Fit Plus	Wii	2009	Sports	Nintend
16	Kinect Adventu...	X360	2010	Misc	Microso
17	Grand Theft Au...	PS3	2013	Action	Take-Tw

☒ Utilisez la détection de type de données riche qui peut être mieux adaptée. Toutefois, les cellules avec des valeurs anormales peuvent être supprimées.

< Précédent Suivant > Annuler

Après avoir cliqué sur "Suivant", il a été nécessaire de modifier les types des colonnes créées par défaut. Toutes les colonnes étaient initialement définies en varchar(50), mais la colonne "nom" a dû être changée en varchar(200) afin de prendre en compte des valeurs dont la taille dépasse 50 caractères. De plus, nous avons mis en paramètres pour toutes les autres colonnes la valeur NOT NULL :

Modifier les colonnes

Introduction
Spécifier le fichier d'entrée
Aperçu des données
Modifier les colonnes
Résumé
Résultats

Modifier les colonnes
Cette opération a généré le schéma de table suivant. Vérifiez si le schéma est correct et, dans le cas contraire, apportez-y les modifications nécessaires.

Nom de la colonne	Type de données	Clé primaire	Autoriser les valeurs Null
Rank	varchar(50)	<input type="checkbox"/>	<input type="checkbox"/>
Name	nvarchar(200)	<input type="checkbox"/>	<input type="checkbox"/>
Platform	nvarchar(50)	<input type="checkbox"/>	<input type="checkbox"/>
Year	varchar(50)	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Genre	nvarchar(50)	<input type="checkbox"/>	<input type="checkbox"/>
Publisher	nvarchar(50)	<input type="checkbox"/>	<input type="checkbox"/>
NA_Sales	varchar(50)	<input type="checkbox"/>	<input type="checkbox"/>
EU_Sales	varchar(50)	<input type="checkbox"/>	<input type="checkbox"/>
JP_Sales	varchar(50)	<input type="checkbox"/>	<input type="checkbox"/>
Other_Sales	varchar(50)	<input type="checkbox"/>	<input type="checkbox"/>
Global_Sales	varchar(50)	<input type="checkbox"/>	<input type="checkbox"/>

Il faut ensuite cliquer sur suivant et notre table sera créée. Un écran de configuration s'affichera et il montrera les différentes colonnes de la table .



VGSALES	
	Rank
	Name
	Platform
	Year
	Genre
	Publisher
	NA_Sales
	EU_Sales
	JP_Sales
	Other_Sales
	Global_Sales

2. Analyse détaillé des données colonne par colonne :

I. Description de chaque colonne :

- Rank : Classement du jeu par rapport aux ventes globales
- Name : Nom du jeu
- Platform : Plateforme de la version des jeux
- Year : Année de publication du jeu
- Genre : Genre du jeu
- Publisher : Éditeur du jeu
- NA_Sales (en millions) : Ventes en Amérique du Nord
- EU_Sales (en millions) : Ventes en Europe
- JP_Sales (en millions) : Ventes au Japon

- Other_Sales : Ventes dans les autres parties du monde
- Global_Sales (en millions) : Ventes globales

II. Requêtes :

- Champs vides :

Il faut maintenant vérifier s'il y a des champs vides.

Requête :

```
SELECT *  
FROM vgsales  
WHERE Rank = '' OR Name = '' OR Platform = '' OR Year = ''  
      OR Genre = '' OR Publisher = '' OR Na_Sales = ''  
      OR EU_Sales = '' OR JP_Sales = '' OR Other_Sales = ''  
      OR Global_Sales = '';
```

Retourne :

Résultats

Messages

Rank	Name	Platform	Year	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales	Other_Sales	Global_Sales
------	------	----------	------	-------	-----------	----------	----------	----------	-------------	--------------

Nous pouvons en conclure qu'il n'existe pas de champs vides.

- Données N/A ou Unknown :

Nous pouvons constater la présence de données "N/A" et "Unknown" (Non Applicable , Inconnu). Pour vérifier cela on exécute la requête suivante :

Requête :

```
SELECT * FROM vgsales  
WHERE (Year = 'N/A') OR (Publisher = 'N/A') OR (Publisher =  
'Unknown');
```

Retourne :

Résultats		Messages								
Rank	Name	Platform	Year	Genre	Publisher	NA_Sales	EU_Sales	JP_Sales	Other_Sales	Global_Sales
1	Madden NFL 2004	PS2	N/A	Sports	Electronic Arts	4.26	0.26	0.01	0.71	5.23
2	FIFA Soccer 2004	PS2	N/A	Sports	Electronic Arts	0.59	2.36	0.04	0.51	3.49
3	LEGO Batman: The Videogame	Wii	N/A	Action	Warner Bros. Interactive Entertainment	1.86	1.02	0	0.29	3.17
4	wwe Smackdown vs. Raw 2006	PS2	N/A	Fighting	N/A	1.57	1.02	0	0.41	3
5	Space Invaders	2600	N/A	Shooter	Atari	2.36	0.14	0	0.03	2.53
6	Rock Band	X360	N/A	Misc	Electronic Arts	1.93	0.34	0	0.21	2.48
7	Frogger's Adventures: Temple of the Frog	GBA	N/A	Adventure	Konami Digital Entertainment	2.15	0.18	0	0.07	2.39
8	LEGO Indiana Jones: The Original Adventures	Wii	N/A	Action	LucasArts	1.54	0.63	0	0.22	2.39
9	Call of Duty 3	Wii	N/A	Shooter	Activision	1.19	0.84	0	0.23	2.26
10	Rock Band	Wii	N/A	Misc	MTV Games	1.35	0.56	0	0.2	2.11
11	Gran Turismo Concept 2001 Tokyo	PS2	2001	Racing	Unknown	0	1.1	0.42	0.33	1.84
12	Call of Duty: Black Ops	PC	N/A	Shooter	Activision	0.58	0.81	0	0.24	1.63
13	Rock Band	PS3	N/A	Misc	Electronic Arts	0.99	0.41	0	0.22	1.62
14	Triple Play 99	PS	N/A	Sports	N/A	0.81	0.55	0	0.1	1.46
15	LEGO Harry Potter: Years 5-7	Wii	N/A	Action	Warner Bros. Interactive Entertainment	0.76	0.47	0	0.13	1.36
16	LEGO Batman: The Videogame	PSP	N/A	Action	Warner Bros. Interactive Entertainment	0.57	0.46	0	0.28	1.32
17	Adventure	3600	N/A	Adventure	Atari	1.21	0.08	0	0.01	1.2

On peut observer ici un exemple où de nombreuses données présentent des valeurs inconnues, qui sont particulièrement fréquentes. Il sera donc important d'y prêter attention et de les traiter lors de la prochaine étape sinon cela pourrait créer des erreurs.

- Redondance :

Ensuite nous vérifierons la redondance d'informations avec cette requête :

Requête :

```
SELECT COUNT(*) AS Redondance, Rank, Name, Platform
FROM vgsales
GROUP BY Rank, Name, Platform
HAVING COUNT(*) > 1;
```

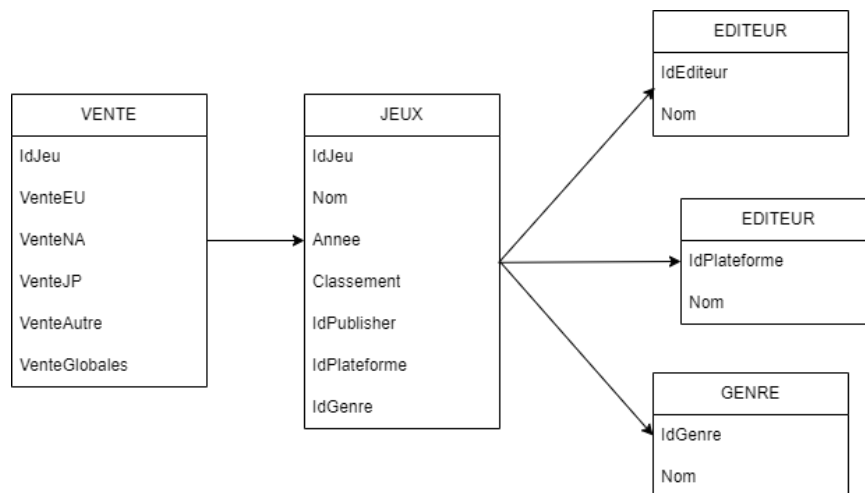
Retourne :

Résultats		Messages		
Redondance	Rank	Name	Platform	

Nous pouvons en conclure qu'il n'y a aucune redondance.

Phase 2 :

1. Schéma de la base de données :



Nous avons créé 5 tables en ayant réparti logiquement les données de la table vgsales. Pour ce faire nous avons séparé les tables “Jeux” et “Ventes” car ces deux tables contenaient un nombre conséquent de données. Après avoir créé les tables en s'appuyant sur ce schéma, il faudra insérer les données dans ces tables.

2. Créations des tables :

Voici les requêtes de la création des tables :

```
CREATE TABLE GENRE (  
    idGenre INT PRIMARY KEY,  
    nomGenre VARCHAR(15) NOT NULL  
);
```

```
CREATE TABLE EDITEUR (  
    idEditeur INT PRIMARY KEY,  
    nomEditeur VARCHAR(50) NOT NULL  
);
```

```
CREATE TABLE PLATEFORME (  
    idPlateforme INT PRIMARY KEY,
```

```
        nomPlateforme VARCHAR(10) NOT NULL
    );

CREATE TABLE JEUX (
    idJeu INT PRIMARY KEY,
    NomDuJeu VARCHAR(150) NOT NULL,
    Annee VARCHAR(5),
    Classement INT NOT NULL,
    idPublisher INT FOREIGN KEY REFERENCES EDITEUR(idEditeur),
    idPlateforme INT FOREIGN KEY REFERENCES
    PLATEFORME(idPlateforme),
    idGenre INT FOREIGN KEY REFERENCES GENRE(idGenre)
);

CREATE TABLE VENTES (
    idJeu INT FOREIGN KEY REFERENCES JEUX(idJeu) PRIMARY KEY,
    VenteEU DECIMAL(4,2) NOT NULL,
    VenteNA DECIMAL(4,2) NOT NULL,
    VenteJP DECIMAL(4,2) NOT NULL,
    VenteAutre DECIMAL(4,2) NOT NULL,
    VenteGlobales DECIMAL(4,2) NOT NULL
);
```

3. Insertion des données :

Nous allons maintenant insérer les données dans les tables créées précédemment.

Requêtes d'insertion de données dans les tables "Plateforme", "Genre" et "Editeur" :

```
INSERT INTO PLATEFORME (idPlateforme, NomPlateforme)
SELECT ROW_NUMBER() OVER (ORDER BY Platform ASC) AS idPlateforme,
Platform
FROM (SELECT DISTINCT Platform FROM vgsales) Platform;

INSERT INTO GENRE (idGenre, NomGenre)
SELECT ROW_NUMBER() OVER (ORDER BY Genre ASC) AS idGenre, Genre
FROM (SELECT DISTINCT Genre FROM vgsales) Genre;
```

```
INSERT INTO EDITEUR (idEditeur, NomEditeur)
SELECT ROW_NUMBER() OVER (ORDER BY Publisher ASC) AS idEditeur,
Publisher
FROM (SELECT DISTINCT Publisher FROM vgsales) Publisher;
```

Requête d'insertion de données dans la table "Jeux":

```
INSERT INTO JEUX(idJeu, NomDuJeu, Classement, idPublisher,
idPlateforme, idGenre, Annee)
SELECT
    ROW_NUMBER() OVER (ORDER BY v.name ASC) AS idJeu, v.name,
    v.rank, ed.idEditeur, pl.idPlateforme, g.idGenre, v.Year AS Annee

FROM vgsales AS v
INNER JOIN PLATEFORME AS pl ON v.Platform = pl.nomPlateforme
INNER JOIN EDITEUR AS ed ON v.publisher = ed.nomEditeur
INNER JOIN GENRE AS g ON v.Genre = g.nomGenre;
```

Requête qui enlève les années "N/A" en valeur "NULL" et définit la colonne "Année" de type entier :

```
UPDATE JEUX SET Annee = NULL WHERE Annee = 'N/A';
ALTER TABLE JEUX ALTER COLUMN Annee INT;
```

Requête d'insertion de données dans la table "VENTES" :

```
INSERT INTO VENTES (idJeu, VenteS, Vente P, VenteNA, Ventes Autre,
Ventes Globales)
SELECT j.iJe, v.EU_Sales, v.JP_Sales, v.NASales, v.Global Sales,
v.Other Sales
FROM JEUX AS j, vg sales AS v
WHERE j.Clasement = v.Rank;
```

Requête permettant de supprimer la table vgsales :

Une fois les insertions correctement effectuées, nous pouvons supprimer la table "vg sales", cependant cela peut être intéressant de la garder pour pouvoir ensuite vérifier ses résultats.

Requête :

```
DROP TABLE vgsales
```

Phase 3 :

1. Installation de SQLPad sur les ordinateurs de l'IUT :

Ouvrez le Gestionnaire de fichier et allez dans le répertoire
'/pédago-iut/INFO/Public/ENSEIGNEMENT/BUT1/R1_05/S1_04/SQLPad'

« Cliquez droit » sur le fichier 'install SQLPad_tar.sh' puis « Lancer dans Konsole »

Attendez la fin du démarrage, le message « !OK! » apparaît dans le terminal

Vous pouvez lancer SQLPad dans un navigateur : <http://localhost:3000>

login : admin

password : admin

Créez une nouvelle connexion en utilisant le driver SQL Server :

Host/Server/IP Address : 164.81.120.19

Database : « nom de votre base »

Database Username : BUT1

Database Password : IUT

2. L'ensemble des requêtes SQL et leurs résultats et les graphiques :

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

Requête :

```
SELECT PLATEFORME.en Plateforme
FROM PLATEFORME
JOIN (
    SELECT j.id Plateforme
    FROM JEUX as j, VENTES as v
    WHERE j.iJe = v.idJeur
    AND j.année >= 2010
    GROUP BY j.id Plateforme
    HAVING SUM(VenteEU) > 20
) as r
ON r.id Plateforme = PLATEFORME.en Plateforme;
```

Retourne :

	nomPlateforme
1	X360
2	3DS
3	Wii
4	WiiU
5	PS3
6	PS4
7	DS
8	XOne
9	PC

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 :

Requête :

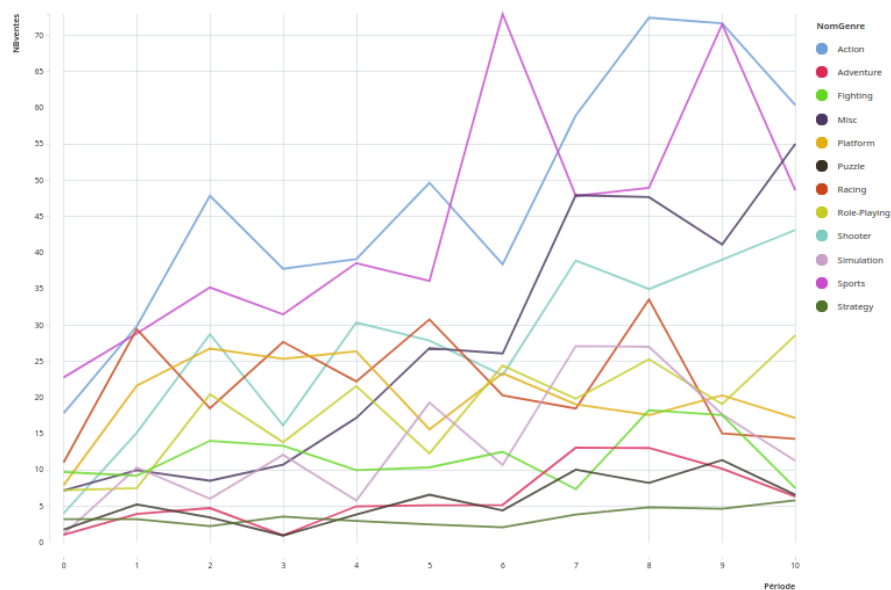
```
SELECT e.nomEditeur
FROM EDITEUR AS e
JOIN (
    SELECT SUM(v.Ventes Globales) AS total Vente, j.Id Publisher
    FROM VENTES AS v, JEUX AS j
    WHERE v.Jeu = j.idJeur AND j.id Plateforme = (
        SELECT id Plateforme FROM PLATEFORME WHERE Nom Plateforme
        = 'PS4'
    )
    GROUP BY j.IdPublisher
) AS r ON r.idPublisher = e.IdEditeur
WHERE totalVente > 2
ORDER BY total Vente 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 :

Requête :

```
SELECT SUM(VenteNA) AS ventes, RIGHT(Année, 1) AS Periode, Nom
Genre
FROM VENTES AS v, JEUX AS j, GENRE AS g
WHERE j.idJeu = v.adJei AND j.idGenre = g.idGenre AND Annee
BETWEEN 2000 AND 2009
GROUP BY Annee, Nom Genre
UNION
SELECT SUM(VenteNA) AS ventes, RIGHT(Année, 2) AS Periode, Nom
Genre
FROM VENTES AS v, JEUX AS j, GENRE AS g
WHERE j.iJe = v.idJeur AND j.id Genre = g.id Genre AND Annee =
2010
GROUP BY Annee, Nom Genre
ORDER BY Nom Genre;
```

Retourne :

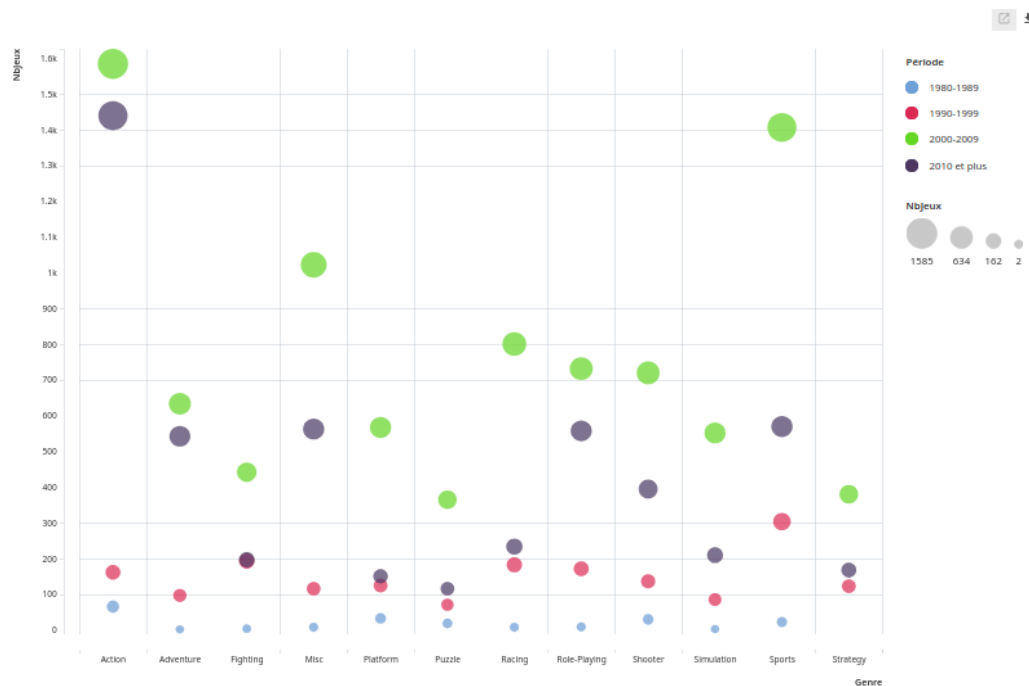


4 - Afficher le nombre de jeux édités en fonction du genre par tranche de 10 ans, soit 4 tranches : de 1980 à 1989, de 1990 à 1999, de 2000 à 2009, 2010 à maintenant. Pour cette requête vous construirez le graphique suivant :

Requête :

```
SELECT COUNT(j.idJeu) AS NbJeux, g.nomGenre AS Genre,
CASE
    WHEN j.Annee BETWEEN 1980 AND 1989 THEN '1980-1989'
    WHEN j.Annee BETWEEN 1990 AND 1999 THEN '1990-1999'
    WHEN j.Annee BETWEEN 2000 AND 2009 THEN '2000-2009'
    WHEN j.Annee >= 2010 THEN '2010 et plus'
END AS Periode
FROM JEUX AS j, GENRE AS g
WHERE j.idGenre = g.idGenre AND j.Annee IS NOT NULL
GROUP BY g.nomGenre,
CASE
    WHEN j.Annee BETWEEN 1980 AND 1989 THEN '1980-1989'
    WHEN j.Annee BETWEEN 1990 AND 1999 THEN '1990-1999'
    WHEN j.Annee BETWEEN 2000 AND 2009 THEN '2000-2009'
    WHEN j.Annee >= 2010 THEN '2010 et plus'
END;
```

Retourne :



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

Les deux graphiques que nous avons trouvé intéressants et présentés sont :

- Un graphique de l'évolution de ventes globales en fonction de l'année (première requête)
- Un graphique de la part de marché qu'occupe chaque plateforme entre 2000 et 2010 (seconde requête)

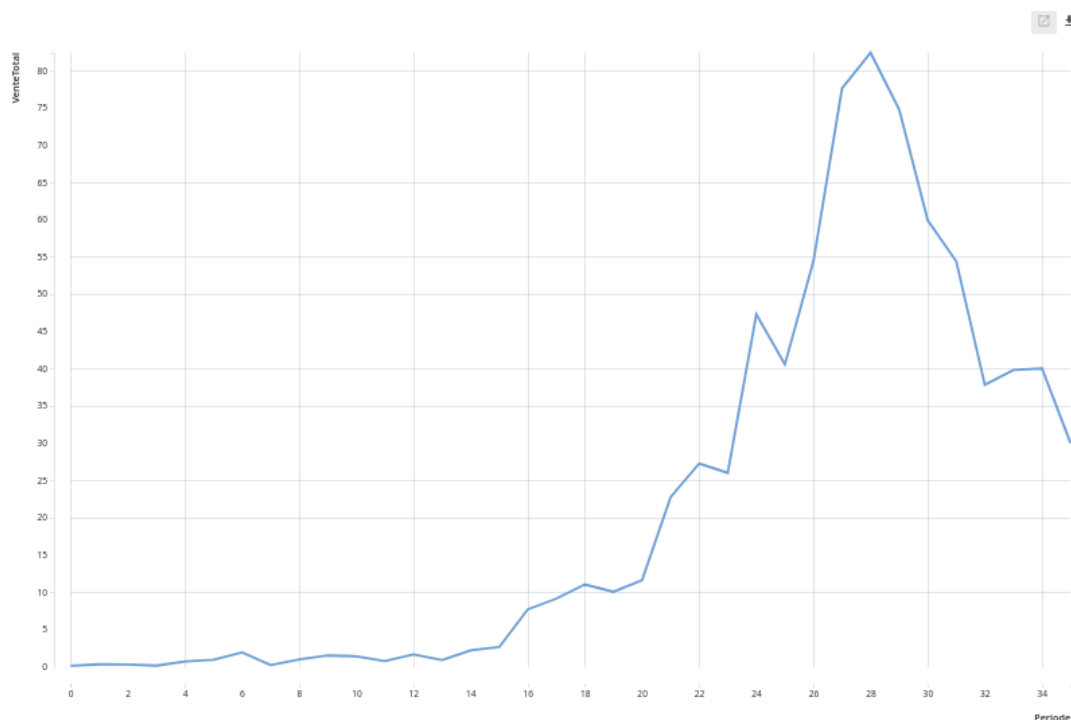
Première requête :

```
SELECT SUM(Ventes Globales) AS Vente Total, RIGHT(Année, 1) AS Periode
FROM VENTES AS v, JEUX AS j
WHERE j.iJe = v.idJeur AND Annee BETWEEN 1980 AND 1989 AND Annee IS NOT
NULL
GROUP BY Annee
UNION
SELECT SUM(Ventes Globales) AS Vente Total, RIGHT(Année, 1) + 10 AS
Periode
FROM VENTES AS v, JEUX AS j
WHERE j.iJe = v.idJeur AND Annee BETWEEN 1990 AND 1999 AND Annee IS NOT
NULL
GROUP BY Annee
UNION
```



```
SELECT SUM(Ventes Globales) AS Vente Total, RIGHT(Année, 1) + 20 AS  
Periode  
FROM VENTES AS v, JEUX AS j  
WHERE j.iJe = v.idJeur AND Annee BETWEEN 2000 AND 2009 AND Annee IS NOT  
NULL  
GROUP BY Annee  
UNION  
SELECT SUM(Ventes Globales) AS Vente Total, RIGHT(Année, 1) + 30 AS  
Periode  
FROM VENTES AS v, JEUX AS j  
WHERE j.iJe = v.idJeur AND Annee BETWEEN 2010 AND 2015 AND Annee IS NOT  
NULL  
GROUP BY Annee;
```

Retourne :



Seconde requête :

```
SELECT SUM(Ventes Globales) AS ventes, RIGHT(Année, 1) AS Période,  
Année, Nom Plateforme  
FROM VENTES AS v, JEUX AS j, PLATEFORME AS p  
WHERE j.idJeu = v.idJeu AND j.idPlateforme = p.idPlateforme AND Annee  
BETWEEN 2000 AND 2009
```

```
GROUP BY Annee, NomPlateforme
UNION
SELECT SUM(VenteGlobales) AS NBventes, RIGHT(Annee, 2) AS Période,
Annee, NomPlateforme
FROM VENTES AS v, JEUX AS j, PLATEFORME AS p
WHERE j.idJeu = v.idJeu AND j.idPlateforme = p.idPlateforme AND Annee =
2010
GROUP BY Annee, NomPlateforme;
```

Retourne :

