

## **SAE 1.04 - Création d'une base de données**

### Compétence 4 :

Concevoir et mettre en place une base de données à partir d'un cahier des charges client

### Apprentissages critiques :

AC 1 : Mettre à jour et interroger une base de données relationnelle (en requêtes directes ou à travers une application)

AC 3 : Concevoir une base de données relationnelle à partir d'un cahier des charges

### Problématique professionnelle :

L'objectif est de montrer que dans le cadre d'un nouveau besoin il est nécessaire d'étudier l'existant, de le maîtriser et ensuite de proposer des extensions à la solution existante.

## Sommaire :

<b>Sommaire :</b> .....	<b>2</b>
<b>Insertion d'un fichier csv :</b> .....	<b>3</b>
<b>Création des nouvelles tables :</b> .....	<b>4</b>
<b>Commandes SQL :</b> .....	<b>6</b>
1 ère requête : .....	6
2 ème requête : .....	6
3 ème requête : .....	6
4 ème requête : .....	7
5 ème requête : .....	8
6 ème requête : .....	8

## Insertion d'un fichier csv :

Pour faire la bascule entre les données qui étaient sous format csv, nous avons utilisé un logiciel en ligne qui se charge de faire la conversion des données à notre place en nous renvoyant pour chaque ligne de données la commande d'insertion correspondante.

Exemple :

1,Wii Sports,Wii,2006,Sports,Nintendo,41.49,29.02,3.77,8.46,82.74

->

```
INSERT INTO SAE104 (Rank, Name, Platform, Year, Genre, Publisher, NA_Sales, EU_Sales, JP_Sales, Other_Sales, Global_Sales) VALUES ('1', 'Wii Sports', 'Wii', '2006', 'Sports', 'Nintendo', '41.49', '29.02', '3.77', '8.46', '82.74');
```

Dans la continuité, nous avons donc créé une première table afin de stocker l'intégralité de ces données.

```
CREATE TABLE SAE104 (  
    Rank      INT,  
    Name      VARCHAR(512),  
    Platform   VARCHAR(512),  
    Year       INT,  
    Genre      VARCHAR(512),  
    Publisher   VARCHAR(512),  
    NA_Sales   DECIMAL(10, 2),  
    EU_Sales   DECIMAL(10, 2),  
    JP_Sales   DECIMAL(10, 2),  
    Other_Sales DECIMAL(10, 2),  
    Global_Sales DECIMAL(10, 2)  
);
```

Cette table sert donc de gigantesque base de données qui va nous permettre de stocker toutes les données avant de les répartir dans des tables plus spécifiques qui serviront à rechercher des informations spécifiques plus rapidement.

<https://www.convertcsv.com/csv-to-sql.html>

## Création des nouvelles tables :

```
CREATE TABLE GENRE_LIST (  
    id INT PRIMARY KEY,  
    genre VARCHAR(50),  
)
```

```
CREATE TABLE PUBLISHER_LIST (  
    id INT PRIMARY KEY,  
    editeur VARCHAR(100),  
)
```

```
CREATE TABLE PLATFORM_LIST (  
    id INT PRIMARY KEY,  
    plateforme VARCHAR(50),  
)
```

```
CREATE TABLE YEAR_LIST (  
    id INT PRIMARY KEY,  
    annee VARCHAR(50),  
)
```

```
CREATE TABLE NAME_LIST (  
    id INT PRIMARY KEY,  
    nom VARCHAR(200),  
)
```

```
CREATE TABLE SALES_LIST (  
    id INT PRIMARY KEY,  
    vente_EU NUMERIC(5,2),  
    vente_NA NUMERIC(5,2),  
    vente_JP NUMERIC(5,2),  
    vente_Autre NUMERIC(5,2),  
    vente_Monde NUMERIC(5,2),  
)
```

```
INSERT INTO GENRE_LIST  
SELECT ROW_NUMBER() OVER(ORDER BY Genre) AS id, Genre FROM  
(SELECT DISTINCT Genre FROM SAE104) LISTE_DES_GENRES
```

```
INSERT INTO PUBLISHER_LIST  
SELECT ROW_NUMBER() OVER(ORDER BY Publisher) AS id, Publisher FROM  
(SELECT DISTINCT Publisher FROM SAE104) LISTE_DES_EDITEURS
```

```
INSERT INTO PLATFORM_LIST  
SELECT ROW_NUMBER() OVER(ORDER BY Platform) AS id, Platform FROM
```

```
(SELECT DISTINCT Platform FROM SAE104) LISTE_DES_PLATFORMES
```

```
INSERT INTO YEAR_LIST
```

```
SELECT ROW_NUMBER() OVER(ORDER BY Year) AS id, Year FROM  
(SELECT DISTINCT Year FROM SAE104) LISTE_DES_ANNEES
```

```
INSERT INTO NAME_LIST
```

```
SELECT ROW_NUMBER() OVER(ORDER BY Name) AS id, Name FROM  
(SELECT DISTINCT Name FROM SAE104) LISTE_DES_NOMS
```

```
INSERT INTO SALES_LIST
```

```
SELECT ROW_NUMBER() OVER(ORDER BY NA_Sales) AS id, EU_SALES, NA_SALES, JP_SALES,  
Other_SALES, Global_SALES FROM  
(SELECT DISTINCT EU_SALES, NA_Sales, JP_SALES, Other_SALES, Global_SALES FROM SAE104)  
LISTE_DES_VENTES
```

# Commandes SQL :

## 1 ère requête :

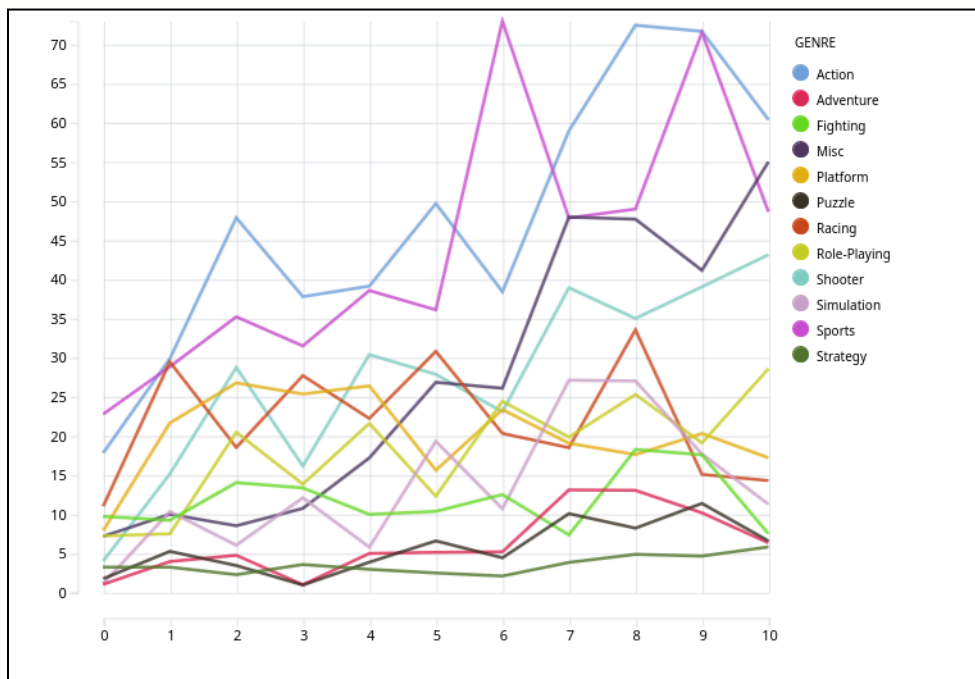
```
SELECT Platform
FROM SAE104
WHERE Year > '2010'
GROUP BY Platform
HAVING SUM(EU_Sales) > 20
```

## 2 ème requête :

```
SELECT Publisher, SUM(Global_Sales) AS Total_Global_Sales
FROM SAE104
WHERE Platform = 'PS4'
GROUP BY Publisher
HAVING SUM(Global_Sales) > 2
ORDER BY Total_Global_Sales DESC;
```

## 3 ème requête :

```
SELECT (Year - 2000) AS Annee, Genre, SUM(NA_Sales) AS Vente_Total
FROM SAE104
WHERE Year BETWEEN '2000' AND '2010'
GROUP BY (Year - 2000), Genre
ORDER BY (Year - 2000), Genre;
```

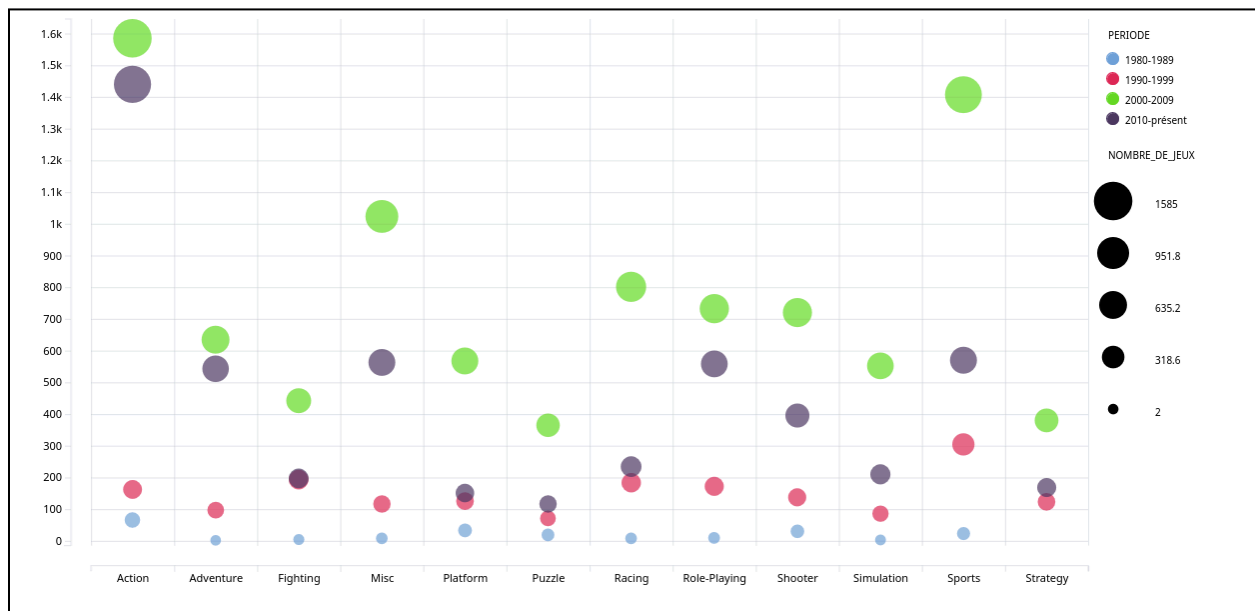


## 4 ème requête :

```

SELECT
  CASE
    WHEN Year BETWEEN 1980 AND 1989 THEN '1980-1989'
    WHEN Year BETWEEN 1990 AND 1999 THEN '1990-1999'
    WHEN Year BETWEEN 2000 AND 2009 THEN '2000-2009'
    WHEN Year >= 2010 THEN '2010-présent'
  END AS periode,
  genre,
  COUNT(*) AS nombre_de_jeux
FROM
  SAE104
WHERE
  ISNUMERIC(Year) = 1
GROUP BY
  CASE
    WHEN Year BETWEEN 1980 AND 1989 THEN '1980-1989'
    WHEN Year BETWEEN 1990 AND 1999 THEN '1990-1999'
    WHEN Year BETWEEN 2000 AND 2009 THEN '2000-2009'
    WHEN Year >= 2010 THEN '2010-présent'
  END,
  genre
ORDER BY
  periode, Genre;

```



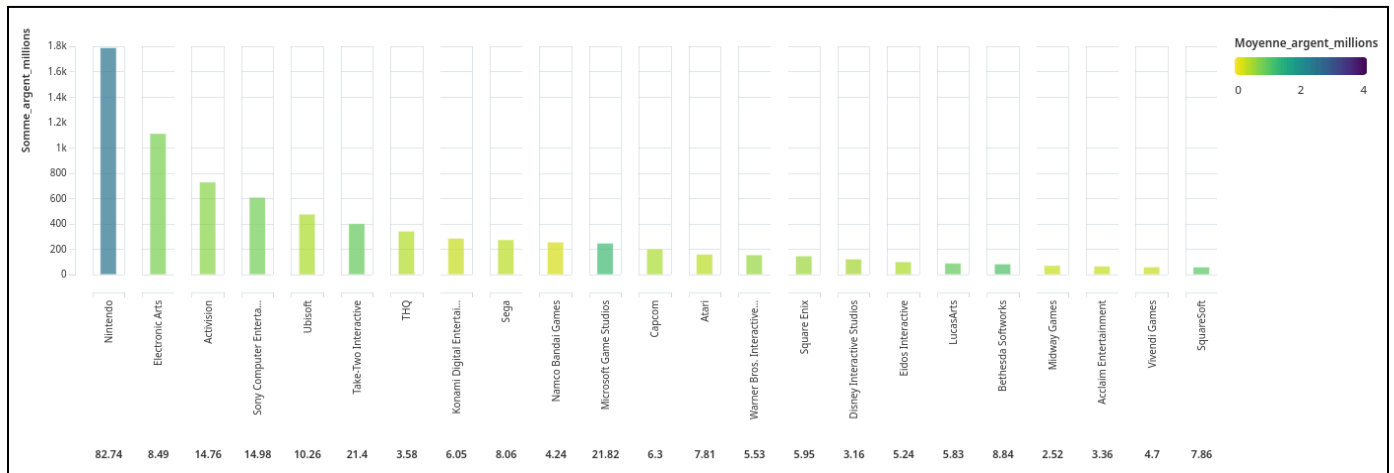
## 5 ème requête :

```

WITH TopJeu AS (
    SELECT
        Publisher,
        Name,
        Global_Sales,
        ROW_NUMBER() OVER (PARTITION BY Publisher ORDER BY Global_Sales DESC) AS
row_num
    FROM SAE104
)
SELECT
    Publisher,
    SUM(Global_Sales) AS Somme_argent_millions,
    AVG(Global_Sales) AS Moyenne_argent_millions,
    MAX(Global_Sales) AS Jeux_le_plus_rentable,
    -- Le jeu le plus vendu est celui avec row_num = 1
    (SELECT Name FROM TopJeu WHERE Publisher = SAE104.Publisher AND row_num = 1) AS
Jeu_le_plus_vendu,
    COUNT(Name) AS Nbr_jeux_publié

FROM SAE104
GROUP BY Publisher
ORDER BY Somme_argent_millions DESC;

```



## 6 ème requête :

```

SELECT Platform, Genre, COUNT(Global_Sales) AS Nbr_jeux_vendu
FROM SAE104
GROUP BY Platform, Genre
ORDER BY Nbr_jeux_vendu DESC

```



