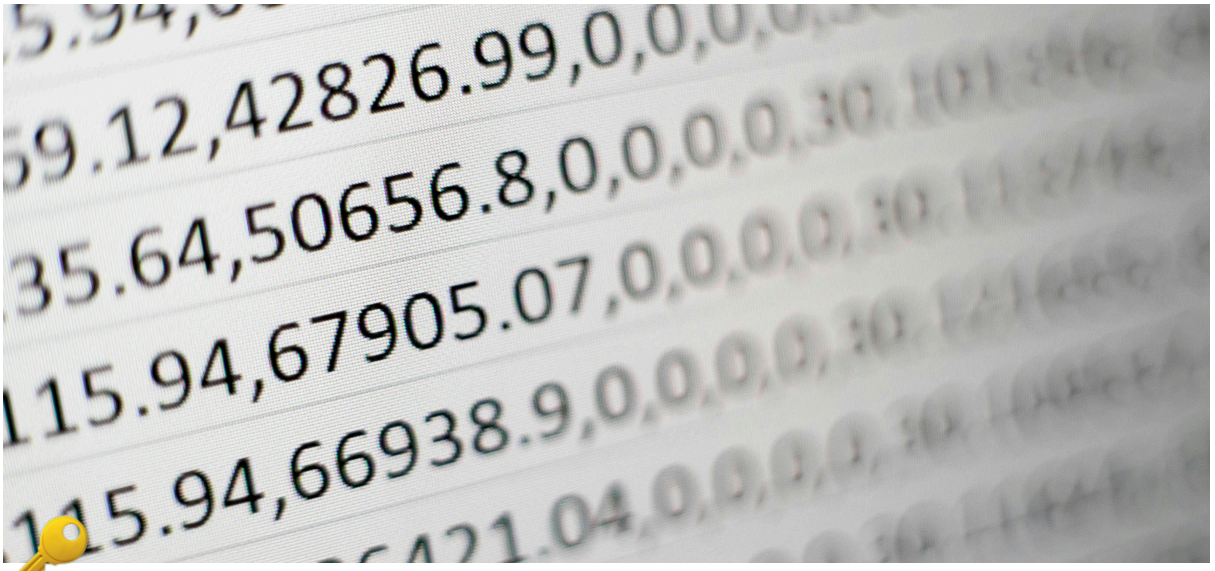


🔑

Video Games Sales Analytics

Tech Stack	AWS S3 · Amazon Athena (ODBC) · AppSource custom visuals · DAX · Power BI Desktop & Service
Brief Summary	A hands-on Power BI project that connects to **Amazon Athena** using both the **Simba ODBC driver** and the **native Athena connector**, models & cleans data in **Power Query**, and publishes a polished **four-page** report with advanced UX (bookmarks, slicers, small multiples) and custom visuals (Radar chart from AppSource).
Link	https://github.com/khanhmdinh/khanhmdinh.github.io/tree/main/05_Video%20Games%20Project

🌟 **Navigation bar**

Table of Contents

Summary

Source Architecture

- **Data lake**: Amazon S3 (tables cataloged by **AWS Glue**).
- **Query layer**: **Amazon Athena** (serverless SQL over S3).
- **Access**: **IAM** user/role with least-privilege to S3, Glue, and Athena (workgroup).
- **Desktop connectivity**:
  - **Power BI → Get Data → Amazon Athena** (native connector)
  - **Power BI → Get Data → ODBC** using **Simba Athena ODBC Driver** (DSN-based)

Deliverables

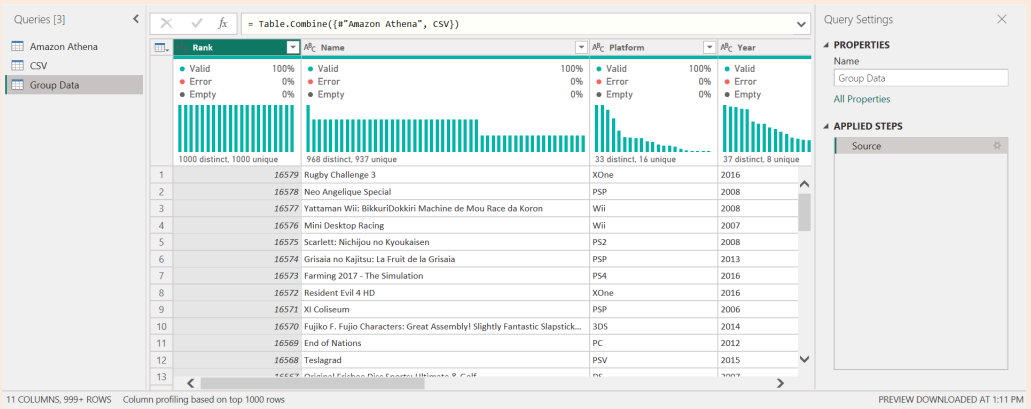
1. **Page 1 — Bookmark UI**: interactive views controlled by bookmarks (toggle KPIs, metric sets, or dimensional cuts).
2. **Page 2 — Slicer UI**: classic filter experience; table rebuilt via **Unpivot** to enable slicer filtering across attributes.
3. **Page 3 — Insights**: includes **Radar chart** for multi-metric brand/product comparison.
4. **Page 4 — Trends**: **Small multiples** line chart for per-segment time series.

Data Assessment & Cleaning Tools

▼ Dataset Information

Column Name	Definition
Rank	The ranking of the game based on sales & popularity.
Name	The title of the video game.
Platform	The gaming system or console on which the game was released.
Year	The year the game was released.
Genre	The category or type of the game (e.g., action, adventure, etc.).
Publisher	The company that published or distributed the game.
NA_Sales	The sales of the game in North America (in millions of units).
EU_Sales	The sales of the game in Europe (in millions of units).
JP_Sales	The sales of the game in Japan (in millions of units).
Other_Sales	The sales of the game in other regions outside NA, EU, and JP.
Global_Sales	The total worldwide sales of the game (in millions of units).

▼ Cleaning & Appending Data imported from Amazon Athena & CSV (Power Query)



Source Tables & Target Schema

Each source contains **11 business columns**, standardized to the following names and types:

Column	Type	Notes
Rank	Whole Number	Positional rank of the title
Name	Text	Game title
Platform	Text	Hardware/platform
Year	**Text (raw)**	Left as text initially due to non-numeric values

Column	Type	Notes
`Genre`	Text	Game genre
`Publisher`	Text	Publisher name
`NA_Sales`	Decimal Number	North America sales
`EU_Sales`	Decimal Number	Europe sales
`JP_Sales`	Decimal Number	Japan sales
`Other_Sales`	Decimal Number	Rest of world
`Global_Sales`	Decimal Number	Total sales

> ***Why Year as Text initially?***
>
> *Source values include non-numeric strings; converting too early produces errors. I normalize* `Year` *after appending.*

Data Quality Checks

- **Schema check:** `Table.ColumnNames` equality across sources before append.
- **Sales coherence:** Validate `Global_Sales ≈ NA_Sales + EU_Sales + JP_Sales + Other_Sales` (allow small rounding deltas).
- **Nulls & outliers:** Review `Publisher`, `Genre`, `Platform` for blanks and rare categories.

▼ Data Cleaning & Troubleshooting (Year Column)

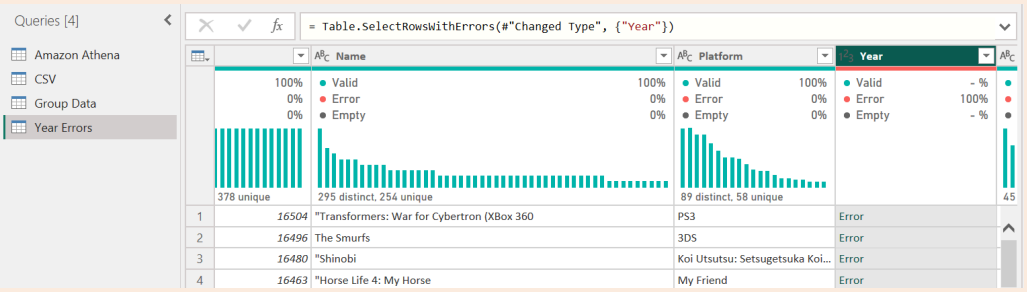
Profile the Data

Power Query ▸ View

- Turn on **Column quality**, **Column distribution**, **Column profile**.
- Switch **Column profiling** from *Top 1000 rows* to **Entire dataset** (bottom-left toggle).

Key observations

- `Rank` shows **Distinct = Unique**, so every value occurs once → a **reliable primary key** candidate.
- Text columns (`Name`, `Platform`, `Genre`, `Publisher`) are correctly typed as *Text*.
- `Year` still typed as *Text* (by design from Phase 1). Converting directly to a number produces **~2% errors**.



🔍

Isolate the `Year` Conversion Errors

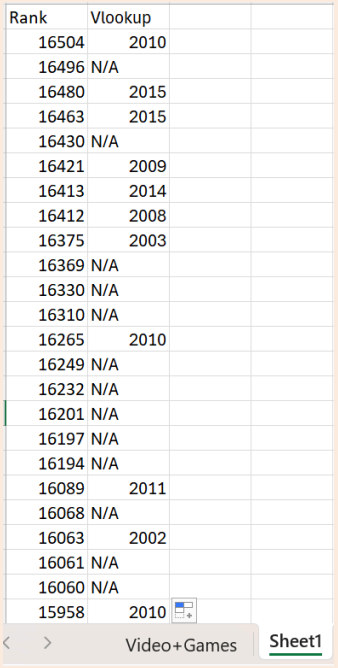
- **Duplicate** the `Group Data` query → rename to `Year Errors`.
- Back in `Group Data`, select `Year` ▸ **Remove Errors** (to keep only valid records in the main stream).
- In `Year Errors`, select **Home ▸ Keep Rows ▸ Keep Errors** to retain **only** the problematic records for analysis. This split gives you:
  - `Group Data` → rows with valid, convertible `Year`.
  - `Year Errors` → rows where `Year` threw a conversion error (e.g., text values like "NA", "Not Available", etc.).

Validate Against the Source (Root-Cause Check)

> *To confirm whether the error was introduced during ingestion or exists in the source:*

- Export `Rank` values from `Year Errors`.
- Use **VLOOKUP** in the original **Video Games** source file to retrieve the corresponding `Year`.
- Findings typically show:
  - Many **valid numeric years** present in source (ingestion/typing issue).
  - A smaller subset with **true** `NA` (no year available at source).

> ***Result:*** *We will repair valid years and keep explicit flags for genuine missing years.*



Rank	Vlookup
16504	2010
16496	N/A
16480	2015
16463	2015
16430	N/A
16421	2009
16413	2014
16412	2008
16375	2003
16369	N/A
16330	N/A
16310	N/A
16265	2010
16249	N/A
16232	N/A
16201	N/A
16197	N/A
16194	N/A
16089	2011
16068	N/A
16063	2002
16061	N/A
16060	N/A
15958	2010

Video+Games Sheet1

Build a "Year Fixed" Helper Table

- `Year` → **Replace "NA" with 0** → convert to *Whole Number*

Appending Final Table for Reporting

Append the repaired rows back to the cleaned main table:

- **Home ▸ Append Queries ▸ Append as New**
  - Table 1: `Group Data` (valid years)
  - Table 2: `Year Fixed` (repaired error rows)
- Rename the result to `Final Table`.

Validation

- **Row count:** *16,598* (expected end-to-end total after merge).
- **Schema:** still **11 columns** (no accidental splits).
- **Types:**

- `Rank` → Whole Number
- `Year` → **Whole Number** (0 for unknown)
- All sales → Decimal Number
- Others → Text

- **Quality checks:**
  - `Year` has **0 errors** (some 0s by design).
  - `NA_Sales` contains nulls (to be addressed in a later step).
  - `Global_Sales` ≈ sum of regional sales (allow rounding differences).

▼ Data Cleaning: Unpivot + Slicer-Driven Radar Chart

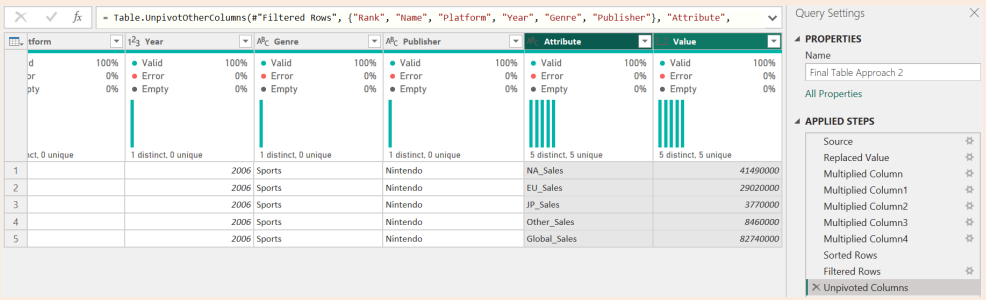
> ***Goals:***
>
> - Reshape regional sales from **columns → rows** (Unpivot) for a **slicer-friendly** model.
> - Drive **one Radar chart** with a **single-select region slicer**.

1. Model housekeeping

Model view → delete all auto-detected relationships (not needed for this page).

2. Build the slicer-ready table

- **Power Query**: duplicate **Final Table** → rename **Final Table (Approach 2)**.
- Select `NA_Sales, EU_Sales, JP_Sales, Other_Sales, Global_Sales` → **Transform ▸ Unpivot Columns**.
  - Output columns: **Attribute** (region label) and **Value** (sales in **units**).



- **Close & Apply**; remove any new relationships Power BI created.

3. QA & usability checklist

✅ **Final Table (Approach 2)** retains the original 11 columns, plus the two unpivoted fields (**Attribute**, **Value**).

✅ `Value` column is **numeric** (units), no unexpected nulls (remember we handled `NA_Sales` nulls earlier).

✅ `Attribute` slicer shows exactly the **five** region choices.

✅ Slicer **Single select** is enforced.

✅ Radar chart updates correctly when switching regions.

✅ No stray relationships re-created by auto-detect.

4. Why this works

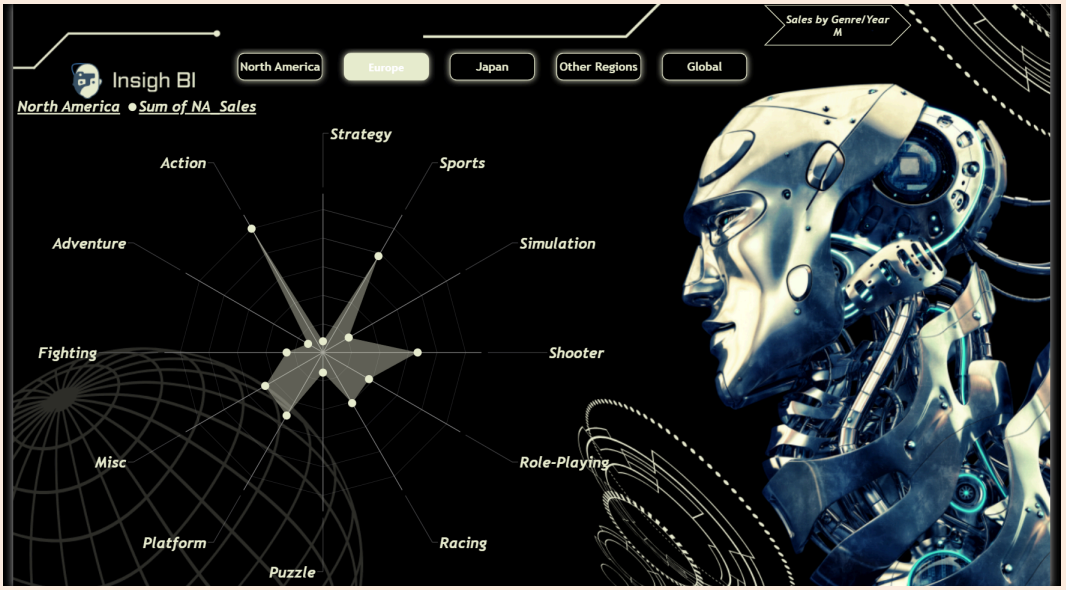
Pros: One chart + one slicer = simple UX; scales if new regions arrive (no extra visuals/bookmarks).

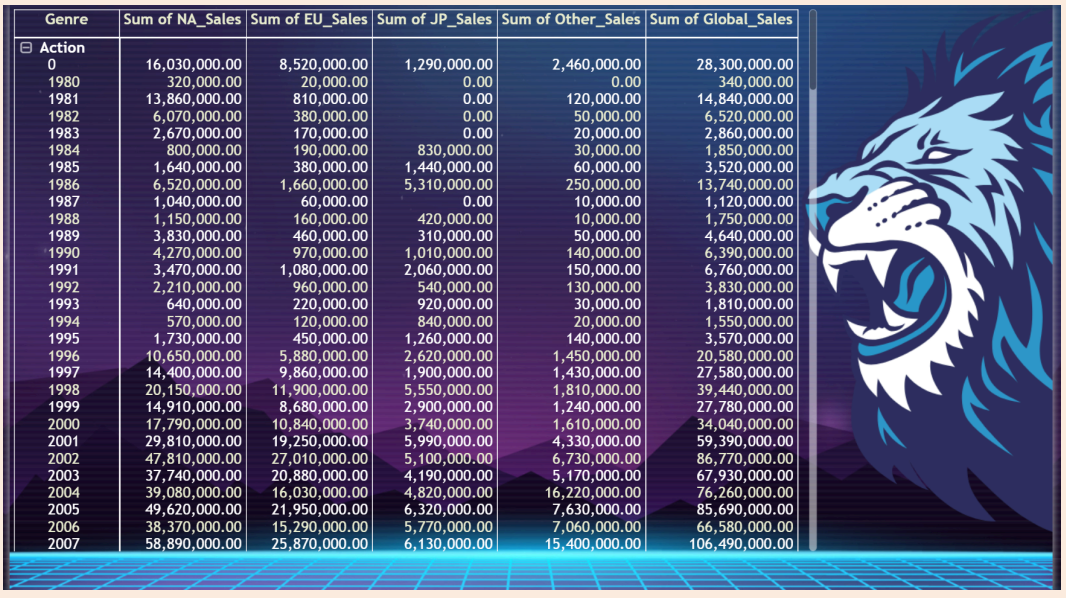
Trade-offs: row count grows (~×5). Optimize/import mode as needed.

Project Showcase

📌 Detailed Report (For More Information)

View the Live Dashboard: https://app.powerbi.com/reportEmbed?reportId=14a2b680-d93f-41e6-b924-4667a7d456eb&autoAuth=true&ctid=9f40849d-a657-43a5-85dc-4bd96886bad5





Genre	Sum of NA_Sales	Sum of EU_Sales	Sum of JP_Sales	Sum of Other_Sales	Sum of Global_Sales
⊟ Action					
0	16,030,000.00	8,520,000.00	1,290,000.00	2,460,000.00	28,300,000.00
1980	320,000.00	20,000.00	0.00	0.00	340,000.00
1981	13,860,000.00	810,000.00	0.00	120,000.00	14,840,000.00
1982	6,070,000.00	380,000.00	0.00	50,000.00	6,520,000.00
1983	2,670,000.00	170,000.00	0.00	20,000.00	2,860,000.00
1984	800,000.00	190,000.00	830,000.00	30,000.00	1,850,000.00
1985	1,640,000.00	380,000.00	1,440,000.00	60,000.00	3,520,000.00
1986	6,520,000.00	1,660,000.00	5,310,000.00	250,000.00	13,740,000.00
1987	1,040,000.00	60,000.00	0.00	10,000.00	1,120,000.00
1988	1,150,000.00	160,000.00	420,000.00	10,000.00	1,750,000.00
1989	3,830,000.00	460,000.00	310,000.00	50,000.00	4,640,000.00
1990	4,270,000.00	970,000.00	1,010,000.00	140,000.00	6,390,000.00
1991	3,470,000.00	1,080,000.00	2,060,000.00	150,000.00	6,760,000.00
1992	2,210,000.00	960,000.00	540,000.00	130,000.00	3,830,000.00
1993	640,000.00	220,000.00	920,000.00	30,000.00	1,810,000.00
1994	570,000.00	120,000.00	840,000.00	20,000.00	1,550,000.00
1995	1,730,000.00	450,000.00	1,260,000.00	140,000.00	3,570,000.00
1996	10,650,000.00	5,880,000.00	2,620,000.00	1,450,000.00	20,580,000.00
1997	14,400,000.00	9,860,000.00	1,900,000.00	1,430,000.00	27,580,000.00
1998	20,150,000.00	11,900,000.00	5,550,000.00	1,810,000.00	39,440,000.00
1999	14,910,000.00	8,680,000.00	2,900,000.00	1,240,000.00	27,780,000.00
2000	17,790,000.00	10,840,000.00	3,740,000.00	1,610,000.00	34,040,000.00
2001	29,810,000.00	19,250,000.00	5,990,000.00	4,330,000.00	59,390,000.00
2002	47,810,000.00	27,010,000.00	5,100,000.00	6,730,000.00	86,770,000.00
2003	37,740,000.00	20,880,000.00	4,190,000.00	5,170,000.00	67,930,000.00
2004	39,080,000.00	16,030,000.00	4,820,000.00	16,220,000.00	76,260,000.00
2005	49,620,000.00	21,950,000.00	6,320,000.00	7,630,000.00	85,690,000.00
2006	38,370,000.00	15,290,000.00	5,770,000.00	7,060,000.00	66,580,000.00
2007	58,890,000.00	25,870,000.00	6,130,000.00	15,400,000.00	106,490,000.00

🌟 **Navigation bar**