DEPARTMENT OF COMPUTER SCIENCE, IT AND ANIMATION

Project Report

ON

"FROM BEATS TO BYTES: VISUALIZING GLOBAL MUSIC TRENDS WITH SPOTIFY DATA "

Submitted by

Mr. Unge Rushikesh Babasaheb (Roll No: -125)

BCA (Sci.) III Year VI Semester

Academic Year 2024-2025

Guided by

Ms. Pallavi G. Chaudhary

Submitted To

Department of Computer Science, IT and Animation

Deogiri College, Chhatrapati Sambhajinagar 431005

DEPARTMENT OF COMPUTER SCIENCE, IT AND ANIMATION

CERTIFICATE

It is the certified work of candidates of ***BCA(Sci.) III Year VI Sem*** *who have satisfactorily completed the project entitle* ***"From Beats To Bytes: Visualizng Global Music Trends With Spotify Data"*** *for the fulfillment of the Bachelor Degree from Dr. Babasaheb Ambedkar Marathwada University, Chhatrapati Sambhajinagar for the academic year 2024-25.*

Submitted By

Mr. Unge Rushikesh Babasaheb (Roll No: -125)

Project Guide	In-Charge	HOD
Ms. Pallavi G. Chaudhary	Dr. Rajashri A.Joshi	Dr. Ritesh A. Magre

Internal Examiner **External Examiner**

ACKNOWLEDGEMENT

Many people have contributed in some or the other forms to this project, although our name appears on the cover of this project. We could not have done this Case Study without the assistance or support of each of the following. We thank all of you.

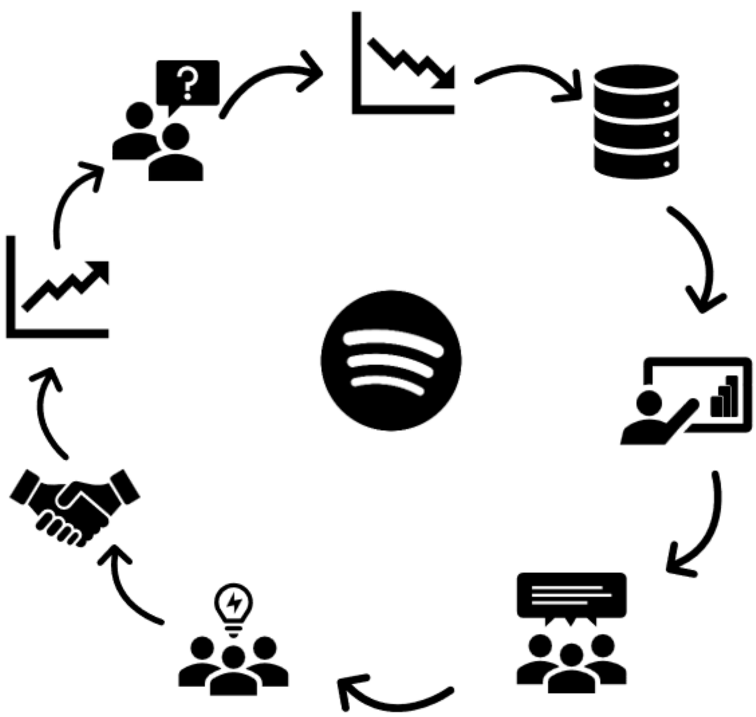
We wish to place on record our deep sense of gratitude towards our project guide **Ms. Pallavi G. Chaudhary** and In charge **Dr. Rajashri A. Joshi** for the constant motivation and valuable help through the project. We also want to express our gratitude towards our Head of the department **Dr. Ritesh A. Magre,** for his invaluable guidance, insightful advice and continuous encouragement. We also extend our thanks to other faculty members for their co-operation.

We also thank those who have directly as well as indirectly supported us for the completion of this Project.

Mr. Unge Rushikesh Babasaheb

INDEX

INTRODUCTION TO POWER BI DESKTOP



1. INTRODUCTION OF POWER BI DESKTOP

Power BI Desktop is a powerful, intuitive data analysis and visualization tool from Microsoft. It’s used by data professionals and business analysts to transform raw data into clear, actionable insights. This tool allows users to connect to a wide range of data sources, clean and transform data, create dynamic reports and dashboards, and share those insights with others. Whether you’re a beginner or a more experienced analyst, Power BI Desktop helps you easily work with data, no matter how complex it might be.

What is Power BI Desktop?

Power BI Desktop is part of the broader Power BI suite, which is a collection of tools designed to help organizations analyze, visualize, and share data. Power BI Desktop is the tool used for designing and creating reports and dashboards on a local machine before sharing them through Power BI Service, the cloud-based platform. It's free to use and does not require a subscription, making it accessible to a wide range of users.

At its heart, Power BI Desktop allows you to load data from a variety of sources, clean and transform it to meet your needs, and then visualize it using a wide array of charts, graphs, and tables. Once your report is complete, you can share it via the Power BI Service, which allows for easy collaboration and the sharing of reports with colleagues, stakeholders, or clients.

Key Features of Power BI Desktop

Power BI Desktop is packed with a variety of powerful features that cater to both novice users and experienced data analysts. These features allow users to manipulate and analyze data in ways that were once limited to highly specialized tools. Let's take a closer look at the major features that make Power BI Desktop so powerful.

1. Data Connectivity

One of the main strengths of Power BI Desktop is its ability to connect to a wide variety of data sources. These sources range from simple files like Excel and CSV, to more complex databases and web services.

Local Files: Power BI Desktop can easily import data from files like Excel, CSV, XML, and JSON.

Databases: It also allows you to connect to SQL Server, MySQL, PostgreSQL, Oracle, and other types of databases.

Cloud-based Sources: You can link Power BI Desktop to cloud services like Azure, Google Analytics, and Salesforce, bringing in data from online platforms.

Web APIs: It can pull data from web-based services using APIs, which is useful when you need to get data from a web-based system that doesn't have a native connector in Power BI.

These options give you the flexibility to bring data from practically any environment, allowing you to work with data in many different formats and from many different systems.

2. Power Query Editor

After data is loaded into Power BI, it often needs some cleaning and transformation before it can be analyzed or visualized. This is where Power Query Editor comes in.

The Power Query Editor is a tool within Power BI Desktop that allows you to manipulate your data in a variety of ways. For example, you can:

Filter Rows: Remove unwanted rows that may not be relevant to your analysis.

Change Data Types: Convert columns into the correct data type, such as turning a text-based column into a date or numeric column.

Merge Tables: Combine multiple data tables into one, whether they are from different sources or different queries.

Transform Data: Pivot or unpivot columns, split or combine columns, and perform other transformations.

With Power Query Editor, you don't need to know complex coding languages to clean your data; instead, you can perform all these tasks using a straightforward, visual interface.

3. Data Modeling

Power BI Desktop allows users to build data models that define how data from different tables is connected. This is crucial for ensuring that the relationships between datasets are understood and reflected accurately in your visualizations.

You can create relationships between tables through a drag-and-drop interface. These relationships are essential for working with multiple datasets and for making sure that your reports display accurate information.

Power BI uses a star schema approach to data modeling, where you have a central fact table that contains your core data (like sales data) and related dimension tables (like time, region, or products). This allows you to organize your data and analyze it from multiple perspectives, making it easier to perform complex calculations.

The Data Analysis Expressions (DAX) language is used to create calculated columns and measures in Power BI. DAX is a powerful formula language that allows you to perform custom calculations on your data, such as calculating growth percentages, aggregating sales by region, or comparing year-over-year data.

4. Visualizations

After transforming your data, you can start creating interactive and dynamic reports using Power BI's visualization tools. These visualizations help you display data in ways that are easy to understand and analyze.

Power BI provides a wide variety of built-in visualizations, including:

Charts: Bar charts, line charts, pie charts, area charts, and more.

Tables and Matrices: Display data in a tabular format, with options to group, sort, and filter the data.

Maps: Geographic data can be visualized on maps, helping to show location-based trends.

Cards and KPIs: These are used to show key performance indicators, such as total sales, profit margins, or other metrics.

Custom Visuals: If the built-in visualizations are not enough, users can import custom visuals from the Power BI marketplace or even create their own.

These visualizations are interactive, which means that clicking on one chart will automatically update and filter other visuals on the report, helping users to gain insights from different perspectives.

5. Slicers and Filters

Power BI provides filters and slicers that give users the ability to explore their data interactively. Slicers are a special type of filter that users can add to reports to allow them to slice and dice the data. For example, you can use a slicer to filter data by date range, region, product category, or any other dimension.

You can apply filters at the visual level, the page level, or the report level. This means you can control what data is visible across all visuals on a page or even across the entire report, making it easy to focus on the most important information.

6. Drill-Down and Drillthrough

With Power BI Desktop, you can create drill-down and drillthrough capabilities, which allow users to explore data in greater detail.

Drill-Down: Clicking on a data point (such as a bar on a chart) will allow you to "drill down" into a more granular level of data. For example, clicking on a region in a map could bring you to a breakdown of sales by city or product.

Drillthrough: Drillthrough allows you to right-click on a data point (such as a product) and see a detailed report filtered specifically for that item. It gives users the ability to focus on specific subsets of data without cluttering the report with unnecessary details.

7. Bookmarks and Storytelling

Power BI Desktop lets you create bookmarks, which are snapshots of a report’s state. This allows you to save different views of a report and switch between them quickly. Bookmarks are great for creating interactive presentations or stories that guide users through a set of insights or trends in the data.

For example, you could create a bookmark that shows high-level sales data, then another that drills down into a specific region, and another that highlights outliers or areas for improvement. Bookmarks provide a way to "walk" users through the report, emphasizing different aspects of the data.

8. Row-Level Security (RLS)

For organizations that deal with sensitive data, Power BI Desktop offers Row-Level Security (RLS). RLS allows you to restrict access to certain rows of data based on a user’s role. For example, you might want regional sales managers to only see data for their specific region.

By setting up roles and defining security filters, you can ensure that users only see the data that is relevant to them, making Power BI secure and customizable to your organization's needs.

9. Publishing and Sharing Reports

Once you’ve created a report in Power BI Desktop, you can publish it to the Power BI Service, which is the cloud-based platform that enables sharing and collaboration. Once in the Power BI Service, you can share the report with others, collaborate with teammates, and even schedule automatic data refreshes to keep your reports up-to-date.

Power BI reports are interactive by default, and users can filter and explore data as needed. The reports can be embedded in web pages, shared via email, or even integrated into other Microsoft applications like Teams, SharePoint, and PowerPoint.

Additionally, Power BI Service allows you to create dashboards, which provide an overview of key metrics from various reports in a single view. Dashboards can be shared with others, providing a high-level snapshot of your organization’s performance.

10. Scheduled Data Refresh

Power BI Desktop can be linked to various data sources that need to be updated periodically. Once your report is published to Power BI Service, you can schedule automatic data refreshes so that your reports always reflect the latest data.

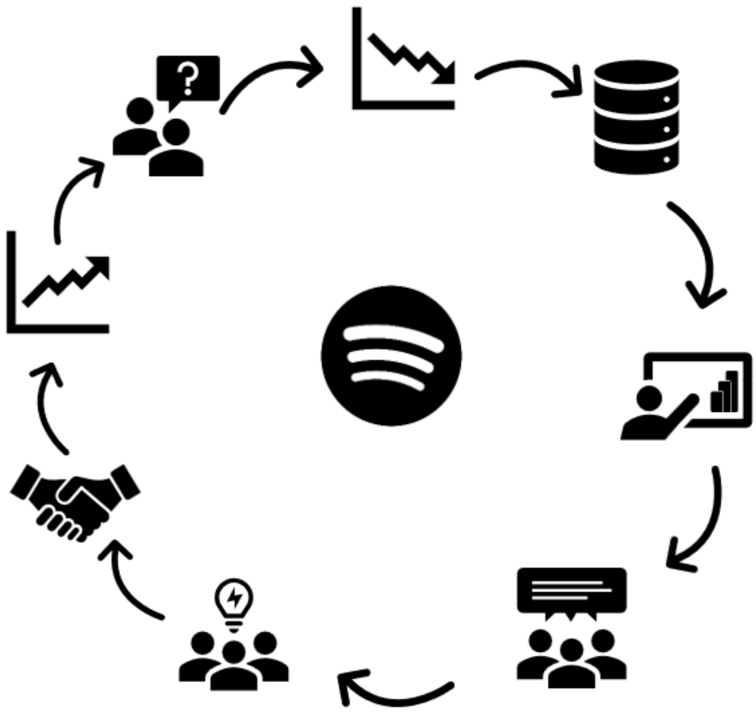
For example, if your sales data is updated daily or weekly, Power BI can refresh the report on a schedule, ensuring that anyone who views the report has access to the most current data without needing to manually update it.

11. Integration with the Microsoft Ecosystem

Power BI integrates seamlessly with other Microsoft tools and services, which makes it particularly powerful for organizations already using Microsoft products. You can import data from Excel files, share reports via Teams, and collaborate on reports within SharePoint.

Power BI also integrates with Azure, which allows for more advanced analytics and cloud-based data management. With Power BI's integration into the broader Microsoft ecosystem, users can work more efficiently and ensure their reports are connected to other business processes.

INTRODUCTION TO POWER BI SERVICE



2. INTRODUCTION OF POWER BI SERVICE

Power BI Service is a cloud-based platform developed by Microsoft that is specifically designed for sharing, collaborating, and distributing business intelligence (BI) reports and dashboards. While Power BI Desktop is used for creating and visualizing data reports, Power BI Service is where these reports are published and shared, allowing users to access, view, and collaborate on the data in an online environment. It plays a central role in providing a collaborative space for teams to work together, manage their reports, and ensure that everyone in the organization has access to the same data.

Once you have created your reports in Power BI Desktop, you can publish them to the Power BI Service. This allows users to access the reports through their web browser, making them easily shareable across teams and departments. Sharing reports is simple and can be done directly with colleagues or specific groups. You can control the access permissions, enabling users to either view or edit the reports, depending on the roles you assign to them.

Power BI Service allows you to organize your reports and dashboards into workspaces. These workspaces act as containers where related reports, dashboards, and datasets can be stored together. You can have personal workspaces for individual use or shared workspaces for team collaboration. When reports are placed in shared workspaces, team members can collaborate in real time, giving everyone a chance to interact with the data and provide input. Permissions can be set at various levels, allowing you to control who can view, edit, or manage content within each workspace.

An essential feature of Power BI Service is its ability to automatically refresh data. By setting up scheduled data refresh, you can ensure that your reports and dashboards always display the latest data without needing to manually update them. You can choose refresh intervals based on your needs, with options to refresh as frequently as every 30 minutes. This feature is essential for keeping reports up-to-date, especially when dealing with large datasets that change regularly.

For organizations that store data locally, Power BI Service offers the Data Gateway, which allows you to connect to on-premises data sources. This makes it possible to integrate data stored in local servers with your cloud-based Power BI Service, creating a hybrid solution. The gateway acts as a bridge between your on-premises data and the cloud, allowing Power BI to access and refresh the data stored on your internal servers.

Another key aspect of data management in Power BI Service is the ability to create dataflows. A dataflow is a series of steps that define how data is extracted, transformed, and loaded (ETL) into the cloud. Dataflows allow multiple reports to reuse the same set of data, which streamlines data management and reduces redundancy. This means that once you create a dataflow, other reports can reference this data, making data management more efficient across the entire organization.

Power BI Service also provides dashboards, which are collections of visualizations from one or more reports. These dashboards give users a high-level view of key metrics and performance indicators. Dashboards are customizable, so you can choose which visuals to include, making them an effective tool for executives and decision-makers who need to see an overview of the business. Additionally, Power BI Service supports real-time data streaming into dashboards, which is particularly useful for monitoring live data, such as sales numbers, website traffic, or operational metrics. This capability ensures that decision-makers always have access to up-to-the-minute information.

To make data accessible on the go, Power BI Service offers mobile apps for iOS, Android, and Windows. This means that users can access dashboards and reports from their smartphones or tablets, allowing them to stay informed and make decisions no matter where they are. With the mobile app, you can interact with the data, apply filters, and drill down into the visuals, just like on the desktop version.

One of the standout features of Power BI Service is its integration of advanced analytics and artificial intelligence (AI) tools. The Q&A feature in Power BI Service uses natural language processing (NLP) to let users ask questions in plain language about their data. For example, you can ask, “What were the total sales last quarter?” and Power BI will automatically generate the relevant visualization or calculation. This feature makes it much easier for non-technical users to interact with data and extract valuable insights without needing to create complex visualizations themselves.

Power BI Service also includes AI-powered insights, such as the Key Influencers feature, which helps you understand what factors are driving changes in your data. For instance, it can identify which variables most significantly impact sales or customer satisfaction. The Decomposition Tree visualization helps you break down data into categories, showing how specific factors contribute to the overall result. These features make it easier to spot trends and identify the drivers behind your data.

For organizations that are using machine learning models, Power BI can integrate with Azure Machine Learning to visualize and deploy machine learning models directly within Power BI reports. This integration enables businesses to leverage their existing machine learning models and display their results seamlessly within their Power BI reports.

Security and governance are also key concerns for organizations, and Power BI Service provides several features to ensure data protection. For example, Row-Level Security (RLS) allows you to control what data each user can see based on their role. This ensures that users only have access to the data that is relevant to them, even if they are viewing the same report. Additionally, Power BI Service supports data sensitivity labels, allowing organizations to classify and protect sensitive data within reports and datasets. This helps ensure that sensitive information is handled according to company policies and regulatory requirements.

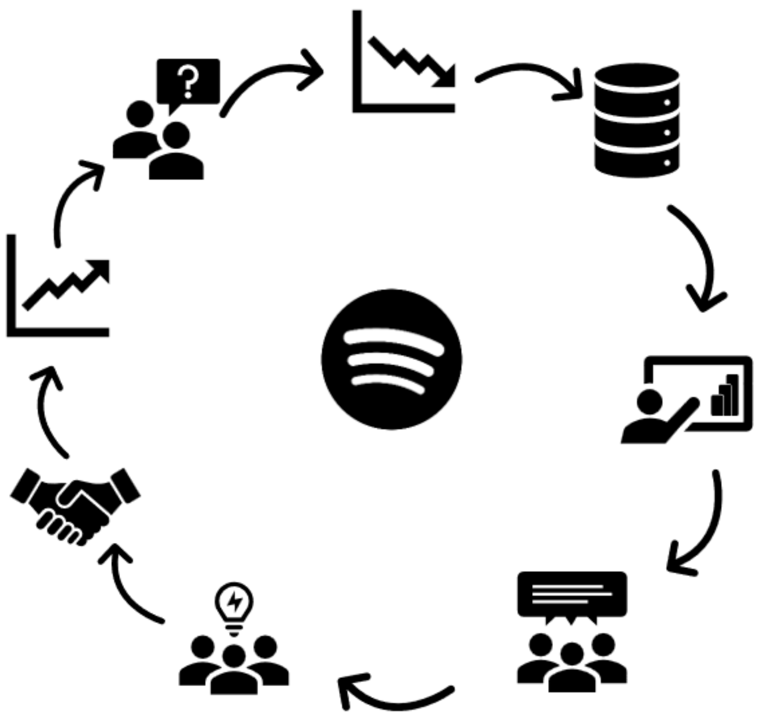
Power BI Service also provides features for controlling how data is shared externally. With data sharing policies, organizations can set rules for sharing reports and dashboards outside the organization, ensuring that data is only accessible by authorized users. This helps prevent unauthorized access to critical business data.

Another important feature in Power BI Service is the ability to use Power BI Apps and Content Packs. Power BI Apps are pre-packaged collections of dashboards, reports, and datasets designed for specific business areas, such as Finance, Marketing, or Sales. These apps can be shared across the organization, providing standardized views of data for everyone involved in those functions. Content Packs are similar but can be created by third-party vendors or internal teams. These packs typically contain reports and datasets integrated with external services like Salesforce, Google Analytics, or other cloud applications.

Regarding licensing, Power BI Service offers several options to suit different organizational needs. The Power BI Free version is available for individual use but comes with limited features, especially when it comes to sharing and collaboration. The Power BI Pro version is designed for teams and organizations, offering advanced features like sharing, collaboration, and data refresh scheduling. For larger organizations that need more advanced capabilities, Power BI Premium offers dedicated cloud resources, larger data storage, and enhanced AI features.

Finally, Power BI Service includes Power BI Embedded, which enables businesses to embed Power BI reports and dashboards directly into custom applications or websites. This feature is especially useful for providing business intelligence capabilities to external clients or for integrating analytics into enterprise systems

INTRODUCTION TO DATA SOURCES



3. INTRODUCTION TO DATA SOURCES

Power BI, a powerful business intelligence tool from Microsoft, is designed to help users visualize and analyze data from a wide variety of sources. A critical part of the Power BI ecosystem is its ability to connect to and integrate with different data sources. The platform provides a flexible and scalable way to work with data, enabling users to create detailed and insightful reports and dashboards.

When we talk about data sources in Power BI, we are referring to the various systems, databases, and files from which Power BI can pull data to be analyzed and visualized. Power BI allows users to connect to both on-premises (local) data sources and cloud-based data sources, offering flexibility in how and where data is stored. Understanding the different types of data sources and how Power BI connects to them is crucial for anyone looking to get the most out of the tool.

Types of Data Sources in Power BI

Power BI supports a broad range of data sources. These can be categorized into different groups based on whether they are stored on-premises or in the cloud, and whether the data is structured or unstructured.

1. **On-Premises Data Sources** include data that is stored locally within an organization's infrastructure. These typically include relational databases like SQL Server, which store data in tables and columns, making it ideal for analysis in Power BI. Additionally, Power BI can connect to Microsoft Excel files, commonly used for storing smaller datasets or simple tables. Microsoft Access, which is used by many small businesses for data management, is also a supported data source. Power BI also supports CSV files, which are a popular way to export tabular data. For more complex data environments, Power BI can connect to Oracle databases, enabling users to run SQL queries to retrieve data for reporting and visualization.

2. **Cloud-Based Data Sources** are hosted on remote servers and offer advantages such as scalability and flexibility. Examples include Azure, Microsoft’s cloud platform, which provides a range of data services like Azure SQL Database, Azure Data Lake, and Azure Blob Storage. These services can be accessed directly from Power BI. Cloud platforms like Google Analytics and Salesforce also integrate with Power BI. Google

Analytics provides website traffic data, while Salesforce offers customer relationship management data. Power BI also connects to Amazon Redshift, a data warehouse service from Amazon Web Services, which is used for large-scale data analysis. Additionally, Power BI supports Google BigQuery, a serverless data warehouse solution from Google Cloud that can handle vast amounts of data, making it ideal for big data analytics.

3. **Web and API-Based Data Sources** allow Power BI to pull in data from the web or through APIs. For example, Power BI can scrape data from websites, a process called web scraping, which is useful when dealing with data presented in web pages. APIs allow Power BI to connect to external services, pulling data in real-time from systems that provide structured data in formats like JSON or XML. Custom APIs can also be created to connect Power BI to internal systems or third-party services that are not directly supported by Power BI.

4. **SaaS Data Sources** (Software as a Service) are another category of data sources in Power BI. These are services that provide cloud-based software, typically through a subscription model. Power BI integrates with platforms like Salesforce, a popular customer relationship management tool, Marketo, a marketing automation platform, and Zendesk, which provides customer service data. These integrations allow businesses to analyze their data stored within these platforms directly within Power BI.

Data Connectivity Modes in Power BI

Once you've selected the data source, Power BI gives you several ways to interact with that data. These modes are important because they determine how Power BI handles and queries data.

1. **Import Mode** is one of the most common ways to work with data in Power BI. In this mode, the data is copied from the data source and stored locally within the Power BI file (a .pbix file). This provides several advantages, including faster query performance, as the data is already stored within Power BI. However, the downside is that the data is static, meaning it doesn't update unless manually refreshed or scheduled for periodic updates.

2. **DirectQuery Mode** is another mode that allows Power BI to query data in real time from the data source. Instead of storing data locally, Power BI sends queries directly to the source every time the user interacts with a report or dashboard. This mode is useful when the data is constantly changing or when the dataset is too large to store locally. However,

performance can be slower compared to the Import mode, as the speed of the query depends on the performance of the data source.

3. **Composite Mode**, combines the strengths of both Import and DirectQuery modes. In this mode, users can work with both imported data and real-time queries in a single report. This flexibility allows users to maintain high performance with smaller datasets while still getting real-time access to large datasets.

Connecting to Different Data Sources in Power BI

Power BI has a set of built-in, native connectors that make it easy to connect to different data sources. These connectors are designed for specific services, applications, or databases, and they simplify the process of linking Power BI to these data sources. For instance, Power BI provides native connectors for SQL Server, Excel, CSV files, and SharePoint, among others. These connectors enable Power BI to automatically recognize the format of the data and help users establish the connection with minimal configuration.

For more complex or customized data sources, Power BI offers the Power Query Editor. This tool allows users to transform and clean the data before it is imported into Power BI. Power Query Editor is extremely versatile, allowing users to filter rows, change data types, merge datasets, and apply a variety of other transformations to prepare the data for analysis.

Power BI also supports OData Feeds, which are used to access structured data from systems like SharePoint, SAP, and Microsoft Dynamics. OData is an open data protocol that makes it easy to query and manipulate data over the web, and Power BI can connect to OData feeds to access this data for reporting.

Handling Different Data Formats

Power BI is compatible with a variety of data formats, which ensures that it can connect to almost any data source. Structured data, which is organized into tables and columns (like databases and spreadsheets), is the most common type of data used in Power BI. However, Power BI also supports semi-structured data (like JSON or XML) and even unstructured data. This flexibility makes Power BI a powerful tool for handling diverse datasets.

For file-based data, Power BI supports several formats, including CSV, Excel (.xlsx, .xls), and text files. These formats are widely used for storing and exporting data, and Power BI makes it easy to import data from them into the platform.

Real-Time Data and Streaming

One of the key features of Power BI is its ability to work with real-time data. This is especially useful for dashboards that require up-to-the-minute data. Power BI can stream data from a variety of sources, such as Azure Stream Analytics, IoT devices, and the Power BI REST API. Real-time streaming ensures that the data in your reports and dashboards is constantly updated, giving you immediate insights into business performance, sales, or other critical metrics.

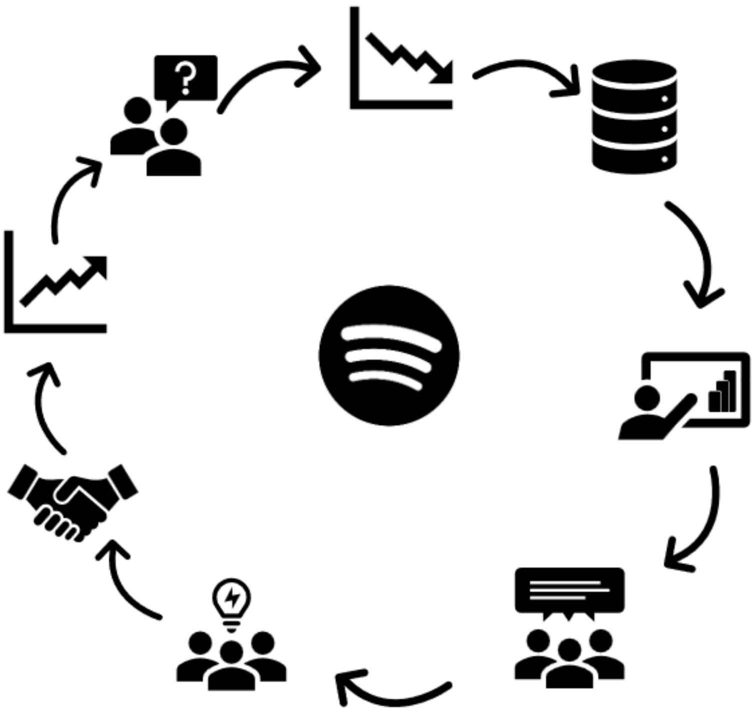
Power BI also supports push data, which allows external applications to send data directly to Power BI using the Power BI REST API. This is particularly useful when data is constantly changing, such as in the case of operational dashboards or sensor data.

Data Security and Privacy

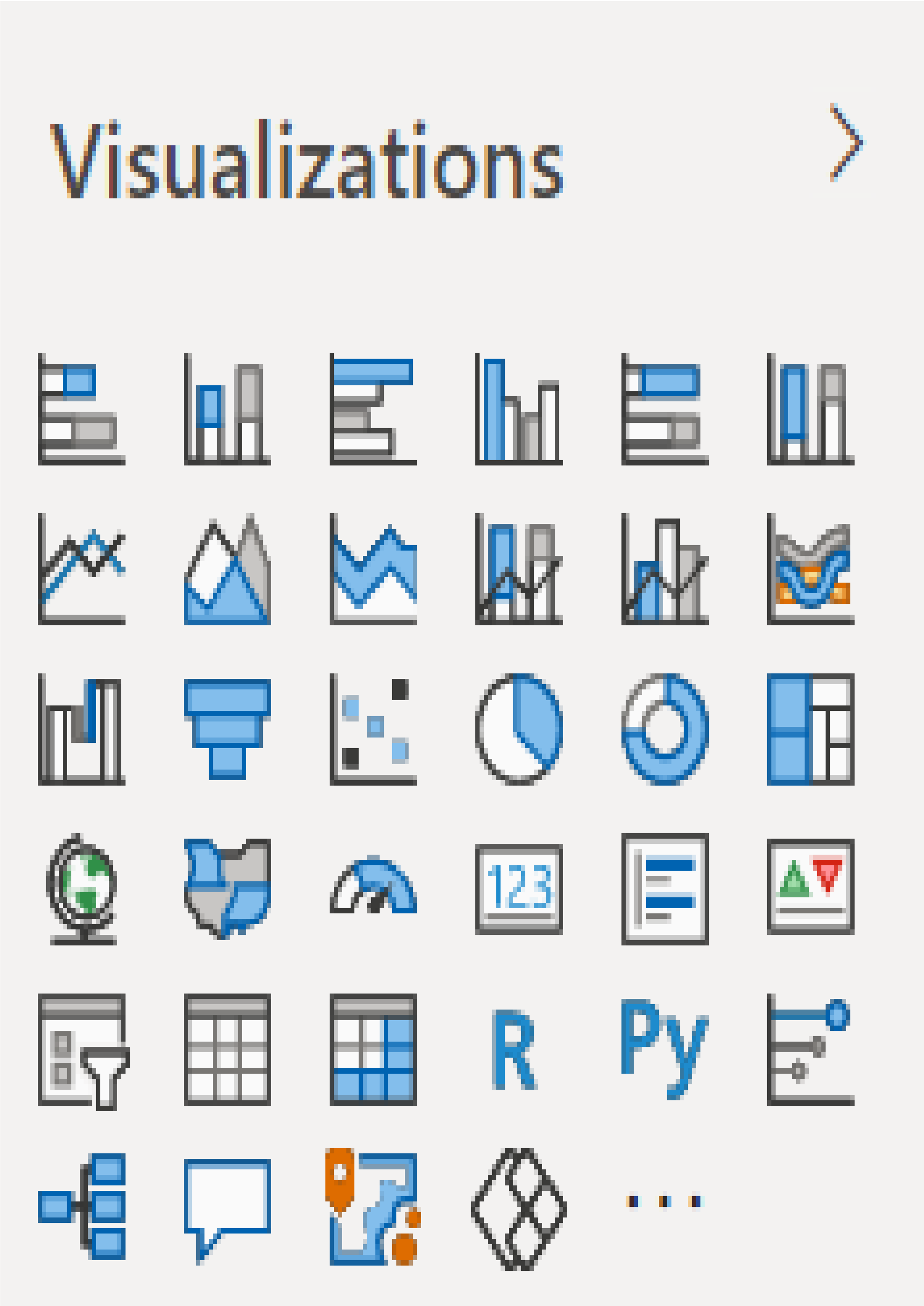
When working with sensitive data, security is a top priority. Power BI provides several features to protect data and ensure it is handled securely. Row-Level Security (RLS) is a feature that allows users to apply filters to data based on roles or user attributes. This ensures that users only see the data they are authorized to view, making it a critical tool for maintaining data privacy.

Additionally, Power BI includes data privacy settings, which help prevent unauthorized access or data leakage when combining data from multiple sources. By setting appropriate privacy levels, you can ensure that sensitive data is handled according to your organization's policies

INTRODUCTION TO VISUALS



4. INTRODUCTION TO VISUALS



In Power BI, visuals are a crucial aspect of turning raw data into understandable and actionable insights. Visuals represent the data in graphical forms like charts, graphs, tables, and other interactive elements, enabling users to see patterns, trends, and outliers clearly. These visuals help simplify complex data sets and make it easier to understand, analyze, and communicate insights.

Power BI offers a wide range of visualizations, from basic bar charts to more complex and interactive maps and gauge charts. These visuals can be tailored to suit the needs of the report, with customization options available for color schemes, data labels, and interactivity. In this way, users can create reports that not only present data but also provide an engaging and informative experience for anyone viewing the report or dashboard.

Types of Visuals in Power BI

Power BI provides several types of visuals that cater to different kinds of data analysis and insights. These visuals can be broadly classified into basic visuals, specialized visuals, and advanced visuals. Each type serves a distinct purpose and can be customized for specific needs.

- **Basic visuals:** include simple charts and graphs that are commonly used for comparing values, showing trends, and displaying proportions. Bar and column charts are among the most basic and popular types of visuals. Bar charts display data using horizontal bars, while column charts use vertical bars. Both are effective for comparing values across different categories. A stacked bar or column chart is often used to show part-to-whole relationships within the data, and a clustered bar or column chart is used to compare values across multiple categories side-by-side. Line and area charts are also used to display trends over time, with line charts connecting data points with a line, and area charts filling the area below the line to show cumulative data. Pie and donut charts are circular visuals that show the proportion of different categories within a whole. Donut charts are similar to pie charts but have a hole in the center. Tables and matrix visuals are used to display data in grid formats, with tables showing raw data and matrices offering the ability to include hierarchical columns and rows for more complex comparisons.

- **Specialized visuals:** include charts and visuals that are used to represent specific types of data or performance metrics. Cards and KPIs (Key Performance Indicators) are used for displaying single values or performance metrics, such as total sales or revenue. Cards show a single number, while KPIs display a value with a comparison to a target, indicating how well the data performs against predefined goals. A treemap is a hierarchical visualization that represents data as nested rectangles, where the size of each rectangle is proportional to the measure, and the color can represent categories. Funnel charts visualize the stages of a process, like a sales pipeline, showing the decrease in values at each stage. Waterfall charts display how a value changes incrementally over time, making them useful for tracking profit and loss.

- **Advanced visuals:** offer more sophisticated ways to visualize data. Gauge and dial charts are often used to track progress toward a goal, such as a sales target, and provide an easy-to-read visual representation of performance. Scatter and bubble charts are used to display relationships between multiple variables, with each point representing a data record. In a bubble chart, the size of the bubble can represent an additional measure. Map visuals are used to represent geospatial data, such as sales by region or customer distribution. These can include filled maps, which color regions based on data values, or shape maps, which are customizable to show specific geographic regions or custom shapes. Slicers, which are a type of filter, allow users to interactively select values to display in the report, making the report more dynamic and enabling users to explore data in more detail.

Customization of Visuals:

One of the strengths of Power BI is its ability to allow extensive customization of visuals. This customization is crucial for creating reports that are tailored to the needs of the user or the audience. Power BI offers various formatting options that enable users to adjust the appearance of visuals to better communicate the data.

Formatting options: in Power BI include the ability to change colors, data labels, and legends. Users can change the color scheme of a visual to align with an organization's branding or to improve readability. For example, using contrasting colors can make it easier to distinguish between different categories in a bar chart. Data labels can be added to visuals to show the exact values, making it simpler for viewers to interpret the data at a glance. Legends and titles can also be customized to clarify what different colors or patterns represent in the visuals, ensuring that the report is easy to understand.

Conditional formatting: is another powerful feature in Power BI that allows users to apply formatting rules based on the values in the data. For instance, color gradients can be used to show variations in data, such as using green for high values and red for low values, helping users quickly spot trends or areas of concern.

Tooltips and drill-through: are two interactivity features that enhance the user experience. Tooltips provide additional context or details when users hover over a data point on a visual. This feature is useful for providing extra information without cluttering the visual. Drill-through functionality allows users to right-click on a specific data point and navigate to another report page that shows more detailed data related to that point, facilitating deeper analysis.

Interactivity: is a fundamental feature of Power BI visuals. By default, visuals are interactive, meaning that when a user clicks on one part of a visual, other visuals on the report page will update accordingly. This allows for dynamic exploration of the data. Cross-filtering and cross-highlighting are key interactivity features. Cross-filtering enables users to click on a specific part of one visual (such as a bar in a bar chart) and filter the data in other visuals on the page based on that selection. Cross-highlighting works similarly but highlights the data related to the selected item across all visuals, making it easier to understand the relationships between different sets of data.

Advanced Visuals with Custom Visuals:

In addition to the built-in visuals, Power BI users can also explore a vast collection of custom visuals created by third-party developers and the community through the Power BI Visuals Marketplace. These custom visuals provide additional functionality and more specialized features that may not be available with the default visuals.

For example, custom maps provide enhanced geographic visualizations, such as detailed or specialized maps for users who need more advanced mapping features than the standard map visuals offer. Gantt charts, which are useful for project management, display tasks over time, with start and end dates shown on a timeline. Bullet charts are another popular custom visual. These compact alternatives to gauges display a single measure along with comparison markers, such as targets or thresholds, in a linear format, offering a more space-efficient way to represent performance.

The marketplace also includes more complex visuals, like heatmaps that visualize data density or intensity, funnel charts for more complex stage-based visualizations, and word clouds for showing the frequency of words in text data.

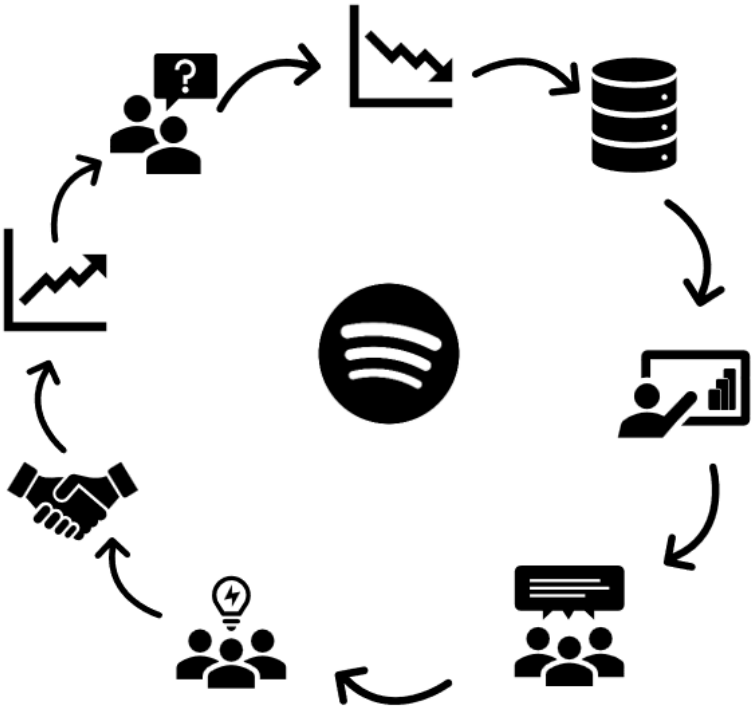
Choosing the Right Visual:

Selecting the right visual is key to creating effective reports. The choice of visual should be

guided by the data you are working with and the story you want to tell.
For comparison of quantities across categories, simple visuals like bar charts or column charts are ideal. When you want to show trends over time, line charts are typically the best choice. For visualizing relationships between two or more variables, scatter charts or bubble charts are appropriate. When working with location-based data, map visuals are the most effective choice, as they help identify trends or outliers in geographic distributions.

For key metrics, visuals like cards or KPI charts are perfect for showcasing single values, such as total revenue or number of units sold. To create more interactivity in a report, slicers and filters allow users to control the data displayed across other visuals, making it easier to explore the data dynamically.

INTRODUCTION TO MICROSOFT SQL SERVER



5. INTRODUCTION TO MICROSOFT SQL SERVER

Microsoft SQL Server is a widely used relational database management system (RDBMS) that is integral to the way organizations store, manage, and process data. It serves as a robust platform for handling large datasets and executing complex queries, which makes it a core component of many enterprise data management strategies. Power BI, Microsoft's business intelligence and data visualization tool, offers seamless integration with SQL Server, allowing users to easily transform, analyze, and visualize data stored in SQL Server databases. This integration provides organizations with the ability to create detailed, interactive reports and dashboards that can turn raw data into actionable insights.

Power BI's ability to connect with SQL Server opens up various possibilities for reporting and analysis. Whether you're working with operational data, financial reports, or complex business analytics, Power BI allows for multiple ways to connect to SQL Server databases and transform that data into meaningful visualizations. These visualizations enable organizations to make data-driven decisions, enhance operational efficiency, and improve overall performance.

Connecting Power BI to Microsoft SQL Server

Power BI provides several methods for connecting to Microsoft SQL Server, each catering to different reporting needs and performance considerations. The main connection methods include DirectQuery mode, Import mode, and Dataflows. Each approach has its own strengths and is suited for different types of data and reporting requirements.

1. **DirectQuery Mode** is a mode where Power BI does not import data into its memory but instead queries the SQL Server database live. This mode ensures that the data in the report is always up to date because Power BI queries the database each time a report is accessed. However, this approach can have performance limitations, especially when dealing with complex queries or large datasets. DirectQuery mode is best suited for real-time reporting where data changes frequently, such as operational dashboards that need to reflect the most recent transactions.

2. **Import Mode** involves loading data from the SQL Server into Power BI's internal data model during the refresh process. Once the data is loaded, Power BI stores it in its memory, which allows for faster querying and analysis. This mode is ideal for smaller datasets or when fast performance is needed for reports and dashboards. However, the data in Power BI may become outdated, requiring periodic refreshes to reflect the most recent changes in the SQL Server database.

3. **Dataflows** provide a more advanced option for connecting to SQL Server, especially when there is a need to perform ETL (Extract, Transform, Load) operations before the data enters Power BI. Using dataflows, you can clean, shape, and transform the data in Power BI's Power Query Editor, which helps ensure that only the relevant and necessary data is imported into Power BI. This method is useful when you want to automate the preparation of data before it is used for analysis and visualization.

Steps to Connect Power BI to SQL Server

To connect Power BI to a SQL Server database, the first step is to open Power BI Desktop. From there, click on the "Home" tab and then select "Get Data." In the subsequent dialog box, choose the option to connect to a "SQL Server." Once selected, you will need to enter the necessary connection details for the SQL Server, including the server name or IP address of the server and, optionally, the specific database you want to connect to. You can also choose the type of authentication you wish to use, whether it's Windows Authentication or SQL Server Authentication, based on how the SQL Server is configured.

After establishing the connection, you can select whether you want to use DirectQuery or Import mode, depending on your reporting needs. In the next step, Power BI will display a list of tables and views available in the database, allowing you to choose the ones you want to work with. Once the data is loaded, you can use Power BI’s built-in Power Query Editor to perform any necessary transformations, such as cleaning the data, filtering it, or merging it with other data sources, before you begin building your reports and dashboards.

Transforming Data from SQL Server in Power BI

Once you’ve connected to SQL Server, transforming the data to make it suitable for analysis is a key part of the process. Power BI offers an intuitive tool called Power Query Editor, which allows you to perform various data transformation tasks.

For example, you can filter and sort data in Power Query to ensure that only the most relevant data is included in your reports. Filtering data can involve operations like selecting only records from a specific time period or geographic region. Sorting data helps to organize it more effectively, allowing you to quickly spot trends and patterns.

You can also aggregate data using Power BI’s grouping features, such as summing up sales by region, calculating averages, or performing other statistical operations. Aggregation is particularly useful when you need to consolidate data into high-level summaries or key performance indicators (KPIs).

Another important feature of Power BI is the ability to merge queries, which is similar to performing SQL JOIN operations. Merging allows you to combine multiple data tables based on common columns or keys, enabling you to create relationships between different data sources. This is crucial when you need to integrate data from various tables, such as combining sales data with customer information.

In addition to merging, Power BI also allows you to pivot and unpivot data. Pivoting transforms columns into rows, which can make the data more suitable for analysis in certain scenarios. Conversely, unpivoting data can convert columns into a more usable, analysis-friendly format by bringing multiple column values into a single column with corresponding identifiers.

Visualizing Data from SQL Server in Power BI

After transforming the data, the next step is to build visualizations that communicate your insights effectively. Power BI provides a variety of visualization options, each of which can be customized to suit the nature of the data and the story you want to tell.

Charts and graphs are among the most common types of visualizations in Power BI. Bar charts and column charts are ideal for comparing values across categories, while line charts are used to show trends over time. Pie charts can be used to display proportions or percentages within a whole. These chart types help make numerical data more understandable and allow you to quickly spot trends, outliers, and comparisons.

Tables and matrices are often used. Tables display raw data in a grid format, while matrices allow for hierarchical data analysis by grouping data in rows and columns. These types of visuals are particularly useful when you need to present detailed information in a structured manner.

If your data includes geographic information, Power BI offers maps as a way to visualize this data geographically. By using map visuals, you can plot data points based on coordinates such as longitude and latitude, making it easier to analyze location-based data, such as sales by region or customer distribution.

Cards and KPIs are used for highlighting key metrics. Cards can display single values, such as total sales, and KPIs can be used to track progress against targets or goals, such as performance against sales quotas or profit margins.

Refreshing Data from SQL Server

To ensure your reports and dashboards in Power BI remain up to date, it's important to refresh the data regularly. Power BI offers two ways to refresh data connected to SQL Server.

Scheduled refresh allows you to set up automatic refreshes at specific intervals, such as daily, weekly, or monthly. This ensures that the data in your reports remains current without the need for manual intervention. Scheduled refresh is especially useful for reports that need to be updated regularly with new data from SQL Server.

Alternatively, you can perform a manual refresh at any time to pull in the most recent data from SQL Server. This feature is helpful when you need to immediately update a report with the latest data without waiting for the next scheduled refresh.

Security and Permissions

When connecting Power BI to SQL Server, security is a critical consideration. Power BI provides several features to ensure that your data is accessed securely.

Authentication is one of the first steps in securing the connection. Power BI can authenticate using either Windows Authentication, which integrates with Active Directory for seamless login, or SQL Server Authentication, which uses a specific username and password for access.
For more granular control over who can access specific data, Row-Level Security (RLS) is available. RLS allows you to restrict access to data based on the user's role, ensuring that users can only view the data relevant to them. For example, a sales manager in one region might only be able to see data for their region, while a corporate executive could access data from all regions.

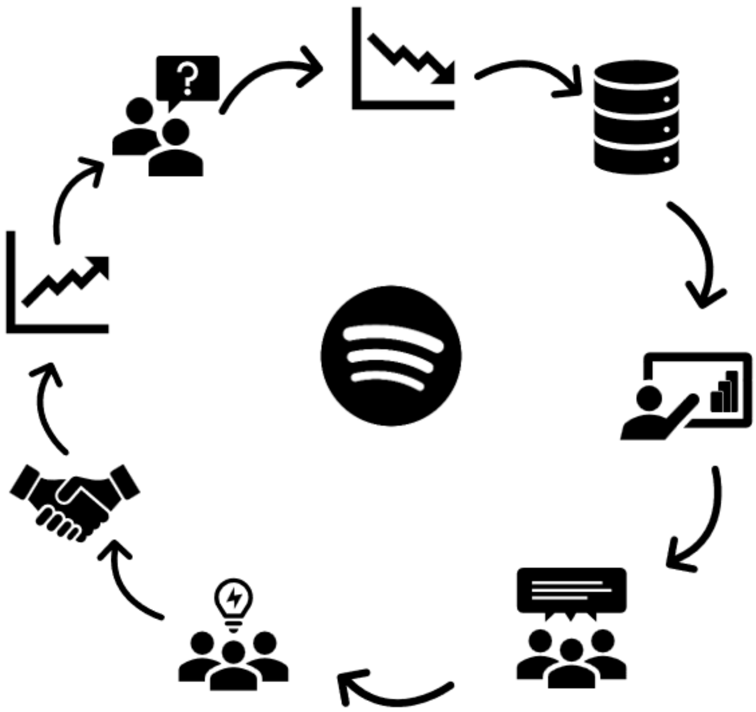
If your SQL Server is hosted on-premises, firewall and data gateway configurations may be required. The data gateway acts as a bridge between your on-premises SQL Server and Power BI, ensuring that the data is securely transmitted while maintaining compliance with organizational security policies.

Use Cases for SQL Server Integration in Power BI

The integration of SQL Server with Power BI can serve a variety of business needs.

- **Real-time data analysis** is one of the most powerful use cases. By using Power BI's DirectQuery mode, businesses can access live data from SQL Server, ensuring that reports and dashboards reflect the most current data. This is ideal for operational reporting, where timely data is crucial for decision-making.
- For organizations dealing with **historical data reporting**, Power BI's Import mode is particularly beneficial. By importing large datasets from SQL Server, businesses can perform in-depth analysis and reporting without worrying about performance issues. This approach is well-suited for periodic reports, such as quarterly financial statements or annual performance reviews.
- **Business intelligence dashboards** are another common use case. By integrating SQL Server data with Power BI, organizations can create interactive dashboards that provide valuable insights into key business metrics, trends, and performance indicators. These dashboards help decision-makers understand their business environment more clearly and take actions based on data-driven insights.

INTRODUCTION TO SQL SERVER MANAGEMENT STUDIO



6. INTRODUCTION TO SQL SERVER MANAGEMENT STUDIO

SQL Server Management Studio (SSMS) is a comprehensive integrated environment designed for database administrators (DBAs) and developers to manage SQL Server instances and databases effectively. It is a powerful tool used for configuring, managing, and administering SQL Server databases, handling tasks such as writing and executing queries, managing security, performing backups, and monitoring database performance. SSMS provides a user-friendly graphical user interface (GUI) that simplifies many of the complex operations that would typically require command-line interactions or scripting. It can be used to manage databases ranging from small, standalone instances to large-scale, enterprise-level systems.

Key Features of SQL Server Management Studio

One of the core features of SSMS is its ability to run T-SQL queries. T-SQL, or Transact-SQL, is Microsoft's extension of SQL that allows users to perform various operations on SQL Server databases. Through the query window in SSMS, users can write and execute SQL queries that retrieve, insert, update, or delete data. More complex SQL scripts can also be created and executed to manage database objects or automate various database tasks. SSMS includes functionality for viewing query execution plans, which can be crucial for optimizing query performance and ensuring efficient database operations.

Object Explorer within SSMS is another significant feature that enables users to interact with the hierarchical structure of SQL Server. The Object Explorer provides a clear view of all the databases, tables, views, stored procedures, and other objects within SQL Server. It allows for easy navigation through the structure, offering options to expand and collapse objects to reveal their relationships and dependencies. With a simple right-click on any object, users can perform administrative tasks such as creating, modifying, or deleting database objects.

SQL Server Profiler is another powerful tool integrated into SSMS. This tool captures and monitors SQL Server events in real time, helping DBAs to track queries and identify performance bottlenecks. By capturing slow-running queries and identifying high-resource-consuming operations, DBAs can analyze workloads and optimize the system's overall performance.

Security management in SSMS is also robust. It provides administrators the ability to manage security at both the server and database levels. DBAs can create and manage SQL Server logins and users, assign roles and permissions to ensure secure access to database objects, and set authentication modes such as Windows Authentication or SQL Server Authentication. Additionally, SSMS supports Row-Level Security (RLS), which allows for granular control over who can access specific data, ensuring that users can only view the data that is relevant to their roles.

In terms of backup and restore functionalities, SSMS provides tools to ensure data protection and disaster recovery. The backup options include full backups, differential backups, and transaction log backups. Users can also schedule automatic backups using the SQL Server Agent, ensuring that critical data is consistently backed up. In the event of a failure or data corruption, SSMS allows for quick restoration of databases from backups.

Components of SQL Server Management Studio

SSMS is built with several key components, each designed to make database management more efficient and user-friendly. The Object Explorer is one of the most important components, as it allows users to interact with the database objects in a graphical interface. It helps to visualize the database structure and access various database elements, such as tables, views, stored procedures, and user-defined functions. Users can perform administrative tasks by right-clicking on objects and selecting the appropriate options.

The Query Window is where SQL scripts are written and executed. It provides an area to write queries, execute them, and view the results. The query window also includes features such as syntax highlighting, which makes it easier to read and debug T-SQL code. Users can also see the query execution plans to analyze the performance of the queries and optimize them accordingly.

Another essential component is the Template Explorer, which provides a collection of pre-written templates for common tasks like creating tables, views, stored procedures, and more. These templates serve as a starting point for users, helping them quickly generate T-SQL code for frequently used operations.

The Solution Explorer is also available in SSMS, allowing users to organize SQL Server projects such as scripts or database projects. It provides a way to manage and structure complex SQL Server development tasks, including version control and code sharing among team members.

Common Tasks in SQL Server Management Studio

One of the most common tasks in SSMS is database creation. To create a new database, users can simply right-click on the "Databases" node in the Object Explorer and select "New Database." They are then prompted to provide the database name and configure settings, including the primary file and log file locations for data storage. This process is simple and intuitive, helping users set up databases quickly.

Writing and running queries is another fundamental task in SSMS. To execute a query, users open a new query window, select the database they want to query, and write the appropriate T-SQL code. After writing the query, users click "Execute" to run it, and the results are displayed in the results pane. This functionality is crucial for retrieving data, updating records, or performing any other data manipulation.

backup and restore operations, users can easily back up a database by right-clicking on the database in the Object Explorer and choosing "Tasks > Backup." The backup process allows users to select the type of backup (full, differential, or transaction log) and specify the destination (disk or tape). To restore a database, users can right-click on the "Databases" node and select "Restore Database," where they can choose the backup file and restore the database to its desired state.

Managing indexes is another essential task for improving query performance. Indexes are critical for speeding up query execution. In SSMS, users can create, modify, or delete indexes by right-clicking on a table and selecting "Design." This opens the table in design view, where users can access the "Indexes/Keys" section to manage the indexes associated with the table.

Security and permissions management is a key aspect of database administration in SSMS. DBAs can manage user access by right-clicking on the "Security" node in the Object Explorer. This area allows them to add or modify logins, users, and roles, as well as assign permissions to database objects like tables and views. By doing so, administrators can ensure that only authorized users have access to sensitive data.

For performance monitoring, SSMS includes tools like the Activity Monitor, which provides real-time insights into SQL Server performance. It shows active sessions, blocked queries, and resource utilization. By using the Activity Monitor, DBAs can identify long-running queries or transactions that might be affecting performance, helping to optimize the server's operations.

Manage Integration Services solutions

SQL Server Management Studio (SSMS) can be used to manage and monitor running SSIS packages. You can organize packages into folders, run, import, export, and upgrade Integration Services packages. However, since SSIS 2012, the storage of packages has changed. They're no longer stored in the server's msdb database of the default instance but are now managed through the SSIS Catalog database (SSISDB). This means that you can no longer manage packages in the same way as you did in previous versions of SSIS. You can still use SSMS to manage the SSIS Catalog database, but you must use the Integration Services Catalogs node in Object Explorer.The latest version of SSMS provides an integrated environment for managing any SQL infrastructure. It also allows users to run SSIS packages stored in the SSIS Catalog from Object Explorer in SSMS.The Import and Export Wizard within SSMS can be used to create SSIS packages, which is a good starting point for learning about SSIS. However, you must use SQL Server Data Tools (SSDT) to create and manage your packages for more complex packages.

Advantages of SQL Server Management Studio

One of the major advantages of SSMS is its comprehensive management capabilities. It consolidates all the necessary tools for database administrators into one environment, making it easier to manage various aspects of SQL Server. These include query execution, database design, performance monitoring, security management, and more. The unified interface makes it easier for users to handle multiple tasks without needing to switch between different applications or tools.

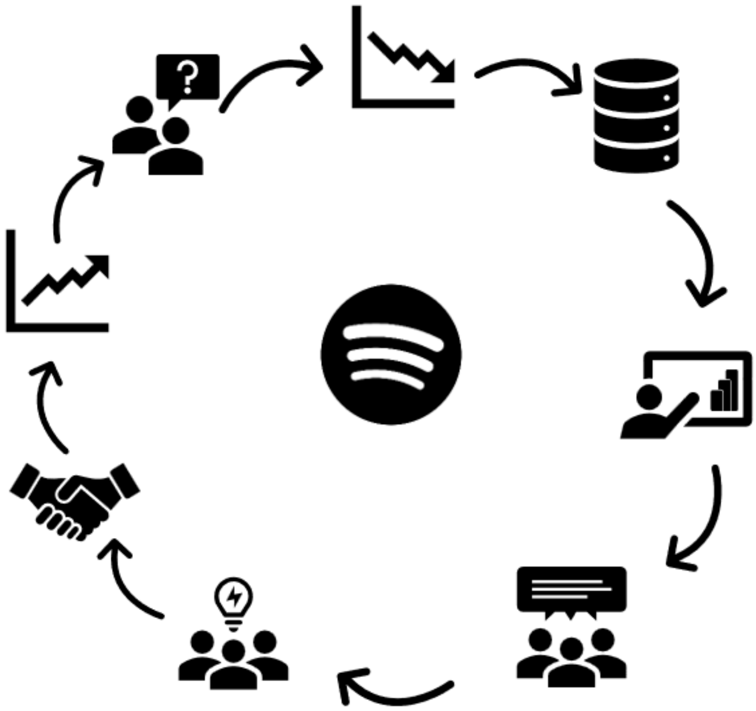
The user-friendly interface of SSMS is another significant benefit. The graphical interface simplifies many complex tasks, allowing even less experienced users to interact with SQL Server efficiently. The intuitive design of SSMS makes database management tasks more accessible, particularly for those who prefer not to rely solely on command-line interactions.

SSMS also provides substantial automation capabilities. The SQL Server Agent, for example, can be used to schedule jobs, automate backups, and run maintenance tasks, ensuring that these processes are handled without manual intervention. Automation helps database administrators save time and reduce the risk of errors caused by human oversight.

The customizability of SSMS is another major advantage. Users can tailor the environment to their specific needs by creating templates, writing scripts, and using extensions. This flexibility ensures that SSMS can be adapted to meet the specific requirements of different organizations or projects.

Finally, SSMS offers robust security management features. Administrators can manage user access and assign permissions effectively, ensuring that sensitive data remains secure. The ability to implement features like Row-Level Security (RLS) further enhances the control over who can access specific data within the database.

INTRODUCTION TO PROJECT



7. INTRODUCTION TO PROJECT

The music industry has undergone a massive transformation in recent years, with the rise of music streaming platforms like Spotify. As one of the largest streaming services globally, Spotify continuously tracks listener preferences, artist performances, and song popularity to enhance its offerings. However, raw data alone cannot provide meaningful insights unless it is carefully analyzed and visualized. This is where data analytics and visualization tools like Microsoft Power BI come into play.

This project focuses on analyzing Spotify's "Top 200 Songs" dataset, which is updated weekly and includes metrics such as track names, artists, genres, streaming counts, and more. By transforming this dataset into an interactive Power BI dashboard, the project aims to provide a comprehensive analysis of song performance, artist success, and genre popularity. The interactive dashboard will uncover hidden patterns and trends, revealing how songs rise to the top, what drives their success, and how different factors like release dates and genres impact their popularity.

The dashboard will cater to a wide range of users, including music industry analysts, marketing teams, producers, and even casual listeners who are curious about trending songs and artists. It will offer insights into questions such as:

- Which artists consistently perform well on the platform?

- What are the characteristics of songs that achieve high streaming counts?
- How do seasonal trends and genre preferences evolve over time?

By leveraging Power BI, the project will demonstrate how complex datasets can be transformed into actionable insights, helping stakeholders make informed decisions. This initiative showcases the immense potential of data visualization tools in revolutionizing how industries, particularly the music industry, interpret and utilize data.

This project not only enhances understanding of music industry trends but also highlights the potential of data visualization tools in other fields, such as e-commerce, education, and healthcare. Through this analysis, we aim to build a strong foundation for future projects that combine data science with user experience design to make analytics accessible and actionable for all stakeholders.

Objective

The project has several core objectives aimed at providing valuable insights and improving data analysis efficiency for Spotify and its stakeholders. These objectives are as follows:

1. Development of a Dynamic Dashboard

The primary goal is to create a visually appealing and interactive Power BI dashboard. This dashboard will analyze Spotify's top 200 songs dataset and present insights in an engaging format.

By offering dynamic visuals and filtering options, it will enable users to explore the data with ease.

2. Trend Analysis

The dashboard will help identify trends, such as top-performing artists, the most popular genres, and the influence of song duration on success. Users will be able to observe how these factors vary over time and across different regions.

3. Simplified Data Interpretation

Through visually intuitive representations like bar charts, pie charts, and line graphs, the project aims to reduce the complexity of raw data. This will allow stakeholders to quickly grasp key insights without requiring advanced technical expertise.

4. Demonstrating the Power of Data Visualization

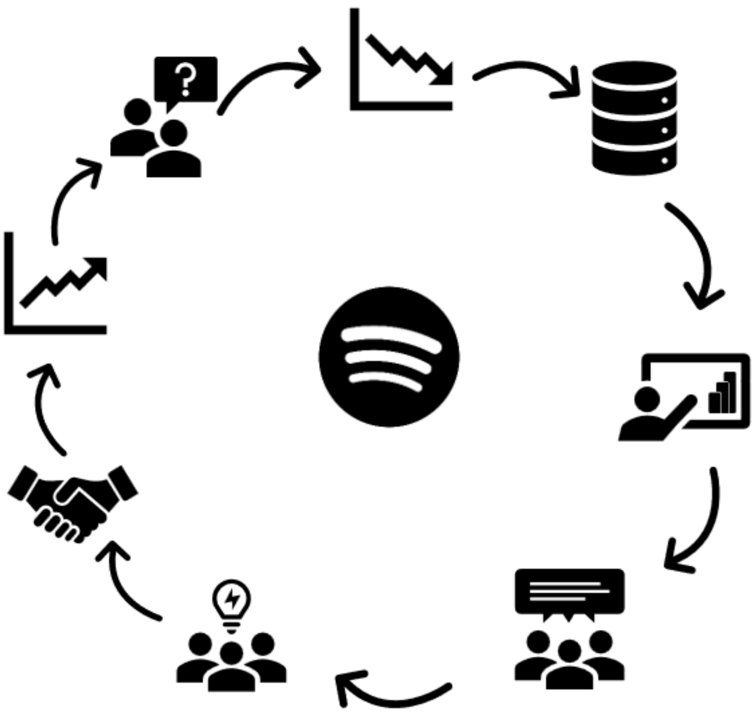
The project serves as a case study to highlight the effectiveness of modern data visualization tools. It will showcase how Power BI can transform large and complex datasets into meaningful insights, thereby encouraging the adoption of such tools in other domains.

Each of these objectives is designed to contribute to the overall goal of enhancing understanding and decision-making in the music industry, using Spotify's dataset as a foundation.

Another objective is to encourage the adoption of data-driven strategies within the music industry. By demonstrating the power of tools like Power BI, this project highlights how visualizations can simplify complex datasets and make insights more approachable for both technical and non-technical users.

Ultimately, this dashboard serves as a bridge between data and actionable strategies, helping stakeholders navigate the ever-changing landscape of the music industry.

SYSTEM ANALYSIS STUDY REPORT



8. SYSTEM ANALYSIS STUDY REPORT

A system analysis is an essential phase for evaluating and improving existing systems by understanding their components, identifying issues, and providing effective solutions to enhance performance. In the context of the Power BI with Spotify Data Analysis project, system analysis encompasses an in-depth examination of the data sources, understanding the user requirements, analyzing the data, pinpointing problems, and offering comprehensive solutions to improve the system's efficiency and effectiveness. This report outlines the system analysis in detail, providing an overview of the current system, its limitations, and potential improvements for the project.

1. Introduction

The purpose of this system is to analyze Spotify's music data using Microsoft Power BI to generate actionable insights into music trends, artist performance, track features, and user preferences. The system aims to achieve several objectives, including visualizing data trends over time, analyzing the popularity of tracks, artists, and genres, and providing useful insights that music curators, artists, analysts, and other stakeholders can use to make data-driven decisions.

The core objective of the analysis is to evaluate the current state of data availability and structure, using either Spotify's API or available CSV files. The system will focus on building an efficient Power BI report with interactive visuals and dashboards to provide detailed analysis of Spotify data, informing decision-making related to music trends, artist popularity, and listener behavior.

Spotify, as one of the world's leading music streaming platforms, offers an enormous dataset related to music tracks, artists, playlists, and user listening behaviors. This project will leverage publicly available data, either through Spotify's API or datasets from platforms such as Kaggle, to provide insights into music consumption trends, facilitating a deeper understanding of how listeners interact with music content.

2. System Overview

The existing system involves using the Spotify API or pre-available datasets to gather track, artist, and genre data, which is then processed using Power BI for analysis and visualization. Power BI, a business analytics tool from Microsoft, is integral to visualizing and analyzing this data. It will allow users to explore insights through interactive dashboards and charts.

Users of the system include data analysts, music curators, marketers, and decision-makers who seek insights into music trends and artist performance. The data flows from external sources (either through API or CSV datasets) into Power BI, which then processes and visualizes the data to present insights in a more user-friendly format.

Despite the strengths of the existing system, several issues exist. Data collection through the Spotify API or CSV files may be slow or inconsistent, requiring manual updates, which can lead to data integrity issues. Additionally, Spotify data often comes with gaps or inconsistencies, requiring thorough cleaning and transformation before analysis. Furthermore, integrating data from multiple sources poses challenges in ensuring accurate relationships across datasets. Lastly, while Power BI is a powerful tool, users without advanced analytical skills may find it challenging to interpret complex insights effectively.

3. Requirements Gathering

To build a robust system, both functional and non-functional requirements must be gathered. The functional requirements include the ability to import Spotify data from various sources, model and clean this data, and generate visualizations like bar charts, line charts, pie charts, and scatter plots. Users should also be able to filter the data based on attributes such as genre, artist, or year, with the option to drill down into specific trends or data points. Dashboards should be interactive, providing users with easy access to key insights about music trends, top tracks, and artist performance.

Non-functional requirements include ensuring the system is responsive and can handle large datasets without performance degradation. The system should adhere to data privacy regulations, especially when dealing with user data, and provide a user-friendly interface that accommodates both technical and non-technical users. Scalability is essential as the system should be able to accommodate growing data volumes as Spotify continuously updates its music catalog. Lastly, the system should be highly interactive, allowing users to drill deeper into the data based on specific queries or trends.

The user requirements further specify that music curators and marketers need real-time insights into trends and popular content to create curated playlists or campaigns. Artists want access to performance metrics to understand their audience engagement, while data analysts need a flexible and detailed system for ad-hoc analysis.

4. System Analysis

Data flow within this system starts with importing data from external sources, such as the Spotify

API or CSV files. This data is then transformed using Power BI's data modeling tools, where missing values are handled, and data types are normalized to create relationships between tables such as tracks, artists, and genres. Once the data is cleaned, it is loaded into Power BI's data model, where it is aggregated and analyzed. Users can then visualize the data through dashboards and reports, offering valuable insights.

The process flow involves several steps: first, the data is imported either through an API or a pre-exported CSV file. Next, Power BI cleanses the data using Power Query, removing duplicates and handling missing values. Then, a data model is created to establish relationships between entities like tracks, artists, and genres. Measures and calculations are defined, such as average popularity or total streams. Once the model is ready, interactive visualizations like bar charts, pie charts, and line graphs are created. The report is published to the Power BI Service for sharing and collaboration among stakeholders.

In terms of entities, the system includes entities like tracks, artists, and genres. These entities are related through foreign keys: for example, the Artist ID in the Tracks table links to the Artist entity. Similarly, the Genre ID in the Tracks table connects to the Genre entity. A detailed entity-relationship diagram (ERD) will illustrate these relationships to ensure accurate data modeling.

The existing system has several strengths, such as Power BI's advanced visualization capabilities and Spotify's access to a wealth of data on user behaviors and music trends. However, there are weaknesses in terms of the data collection process, which can be slow or inconsistent. Moreover, data quality issues may require manual effort to clean, especially when integrating data from different sources.

5. Proposed Solution or Improvements

To address the current system's limitations, several improvements are proposed. First, implementing a fully automated data pipeline will allow the system to extract data from the Spotify API in real-time, reducing the need for manual updates and ensuring up-to-date information. Power BI Embedded can be utilized to provide customized dashboards for different users, potentially integrating these reports into Spotify's internal tools or customer-facing platforms.

Further improvements include integrating the Spotify Web API to continuously pull live data, automating data cleaning and transformation processes, and providing advanced reporting features. For example, the system could track listener behavior or allow users to compare artist performance over time.

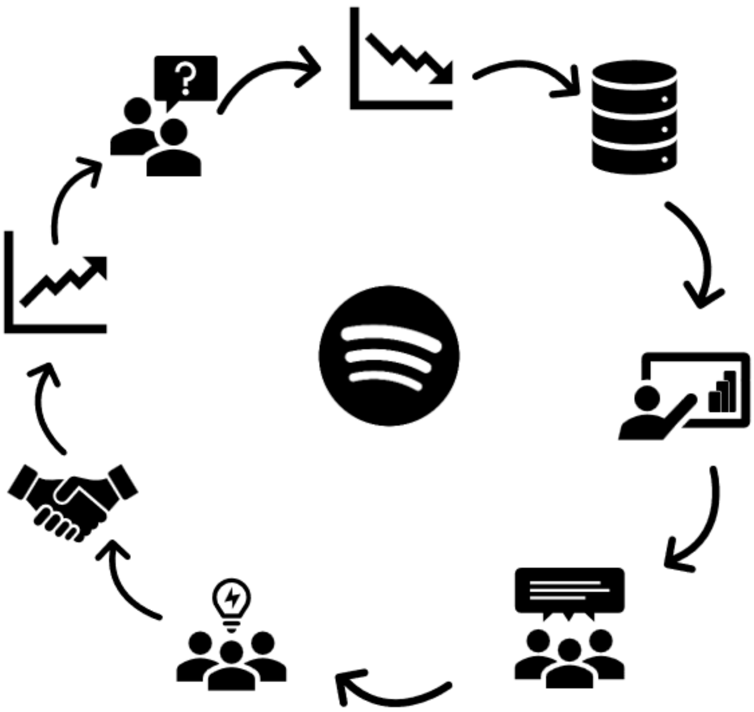
For the technology stack, external ETL (Extract, Transform, Load) tools like Power Automate or custom scripts using Python or Azure Data Factory could be used to automate data extraction and transformation. Cloud storage solutions, such as Azure Blob Storage, could be implemented to store historical Spotify data for more detailed and long-term analysis.

6. Cost-Benefit Analysis

Estimating the costs involves the Power BI licenses required for report generation, development costs for integrating the Spotify API and creating custom reports, and potential expenses for cloud storage to store large volumes of historical data. However, the expected benefits outweigh the costs. By providing enhanced insights into music trends and artist performance, the system will enable data-driven decisions in playlist curation, marketing, and artist promotion.

Additionally, automation will reduce manual efforts, saving time and resources, while scalability ensures the system can grow alongside increasing data volumes. The return on investment (ROI) is evident in terms of improved user engagement, optimized playlist strategies, and better marketing outcomes, all of which can drive revenue growth for Spotify and its partners.

SCOPE OF SYSTEM



9. SCOPE OF SYSTEM

Scope of the System

The scope of the Power BI with Spotify Data Analysis system encompasses all the activities involved in collecting, transforming, analyzing, and visualizing Spotify's music data to generate actionable insights. This system aims to provide a comprehensive view of music trends, track popularity, artist performance, and user listening behavior through interactive dashboards and reports.

At the core of this system lies the integration of Spotify's vast dataset, including information about tracks, artists, genres, and user interactions. By leveraging Microsoft Power BI, this system enables stakeholders to interact with data in real time and derive insights from a range of visualizations, including bar charts, line graphs, pie charts, and scatter plots. These visualizations aim to simplify the understanding of complex music consumption patterns, helping curators, marketers, data analysts, and artists make data-driven decisions.

Data Collection and Integration is a significant part of the system's scope. The system will either pull data from Spotify's public API or use datasets provided through third-party sources like Kaggle. It will encompass key features such as track popularity, genre trends, artist information, streaming counts, user preferences, and song features like tempo, danceability, and acousticness. This data will be aggregated over time, allowing for longitudinal analysis of how trends evolve. The scope includes managing both structured and unstructured data to ensure the correct data flow into Power BI.

Data Transformation and Modeling aspect of the system involves cleaning and structuring the raw data for analysis. The system will handle missing data, data duplication, inconsistencies, and incorrect formatting, ensuring that the datasets are accurate and ready for meaningful analysis. Additionally, relationships between different entities such as tracks, artists, and genres will be established through a structured data model, allowing for more advanced analysis across multiple dimensions. This part of the system ensures that the data is well-organized and ready to provide insights.

Visualization component of the system will come into play. Power BI's interactive dashboards and reports will serve as the primary tool for presenting the analyzed data. Users will have the ability to explore key insights on aspects like the most popular artists, tracks, and genres over specific periods, along with trends in user preferences. These visualizations will also allow for drill-down capabilities, meaning users can click on a particular trend or data point and get further details on specific attributes such as release date or region. By creating a dynamic dashboard, users will gain insights into seasonality, genre preferences, and track performance with ease.

User Interactivity element is also a crucial part of the system's scope. The system is designed with flexibility in mind, enabling users to filter data by various attributes like artist, genre, and time frame. This allows users to focus on specific data points and get more granular insights into user behavior and music performance trends. Users can explore how different factors, such as song length or release timing, influence the success of a track. This interactivity empowers users to perform ad-hoc analysis and explore trends that matter most to them.

Automation and Real-Time Data Integration. Currently, the system relies on importing static datasets, which can become outdated as new tracks are released. The scope of the system includes automating the data extraction process by integrating Spotify's Web API to pull real-time data. This approach ensures that the system is constantly updated with the latest music data, enabling users to work with fresh, real-time insights without relying on manual updates.

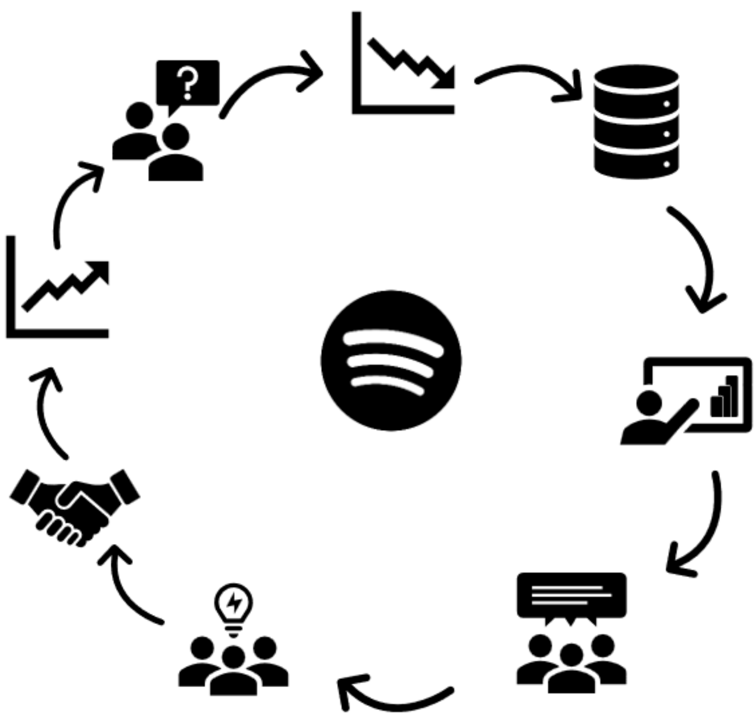
User Base for this system is diverse, consisting of music curators, artists, marketers, analysts, and casual users. Curators and marketers will rely on the insights to improve playlist curation and enhance music recommendations, while artists will benefit from knowing how their tracks are performing and understanding the dynamics of their audience engagement. Analysts and decision-makers will use the system for trend forecasting, understanding long-term shifts in music preferences, and exploring factors that influence the success of different genres or tracks.

System Maintenance and Scalability aspect is also an important consideration in its scope. As the Spotify music catalog expands and user preferences evolve, the system should be scalable to accommodate an increasing volume of data. The system will need to handle larger datasets over time and continue providing quick, responsive insights despite the growing scale. Furthermore, ongoing maintenance will be required to ensure data consistency, address any quality issues, and update the system for any changes to the Spotify API or data structures.

Security and Privacy, the scope of the system also includes ensuring that any user-related data, if involved, adheres to data protection regulations such as GDPR. The system must handle user information securely, respecting privacy policies and ensuring that any data used for analysis complies with legal requirements.

Overall, the scope of this system encompasses the full cycle of data from collection and transformation to analysis and visualization, all while ensuring a smooth and interactive user experience. The end result is a dynamic, scalable, and user-friendly system that empowers stakeholders in the music industry to make informed decisions based on robust data insights. Through this system, Spotify’s music data will be transformed into meaningful, actionable knowledge that can drive strategic decisions in music curation, marketing, and artist management

HARDWARE AND SOFTWARE REQUIREMENTS



10. HARDWARE AND SOFTWARE REQUIREMENTS

A) Hardware Requirements

To run Power BI and efficiently process the dataset from Spotify, having the right hardware setup is crucial to ensure smooth operation, especially when dealing with large datasets, data transformations, and complex visualizations. Below are the detailed hardware requirements for optimal performance:

1. **Processor:**
   - **Minimum Requirement:** AMD Ryzen 5 or Intel Core i5 (or equivalent)
     - These processors provide sufficient computational power for handling moderate-sized datasets, basic calculations, and visualizations in Power BI.
   - **Recommended:** AMD Ryzen 7 or Intel Core i7 (or equivalent)
     - These processors have more cores and threads, which are beneficial for faster data processing, especially when dealing with larger datasets and complex operations in Power BI.

2. **RAM:**
   - **Minimum Requirement:** 8 GB
     - 8 GB of RAM is enough to handle small to moderate datasets and allows Power BI to run efficiently for basic analysis and visualizations.
   - **Recommended:** 16 GB
     - 16 GB of RAM is ideal for optimal performance, particularly when working with large datasets, performing complex queries, or multitasking with other applications like Excel, SQL Server, and web browsers simultaneously. Larger datasets in Power BI, like the weekly updated Spotify data, require more memory for smooth performance.

3. **Storage:**
   - **Minimum Requirement:** 256 GB SSD (preferred) or equivalent HDD storage
     - An SSD (Solid State Drive) offers much faster read/write speeds compared to a traditional HDD, improving overall system performance, especially when processing large data files in Power BI.

- **Recommended:** 512 GB SSD or higher
  - A larger SSD provides more space for storing datasets, reports, and intermediate files. It also ensures faster loading times for Power BI and quicker data processing.

4. **Display:**
   - **Minimum Requirement:** HD (1920x1080) resolution or higher
     - Power BI requires a good display to show visualizations clearly. A minimum HD resolution ensures that charts, graphs, and dashboards are displayed without distortion.
   - **Recommended:** 4K (3840x2160) resolution
     - A 4K display is ideal for advanced users who need to work with multiple dashboards simultaneously or need to present detailed visualizations with clarity.

5. **Graphics:**
   - **Minimum Requirement:** Integrated graphics (Intel UHD, AMD Vega, etc.)
     - Integrated graphics are sufficient for typical Power BI visualizations. Power BI does not heavily rely on GPU performance, so integrated graphics should handle most of the data visualizations.
   - **Recommended:** Dedicated GPU (NVIDIA GTX/RTX or AMD Radeon)
     - A dedicated GPU can provide enhanced rendering for complex visualizations or large, intricate charts. It can be especially useful if you plan to work with advanced graphics or require smoother performance in multi-monitor setups.

6. **Input Devices:**
   - **Standard Requirement:** Standard keyboard and mouse for navigation
     - A simple keyboard and mouse are adequate for navigating Power BI's interface, creating dashboards, and interacting with the data.

B) Software Requirements

Power BI, combined with other essential software, is necessary for analyzing and visualizing Spotify's dataset. The software components are crucial for transforming data, building interactive dashboards, and performing analysis efficiently.

1. **Operating System:**

- **Minimum Requirement:** Windows 10 (or later)
  - Power BI is optimized for Windows operating systems, and newer versions like Windows 10 or 11 ensure compatibility with the latest Power BI features and provide system stability.
- **Recommended:** Windows 11
  - The latest version of Windows provides enhanced performance, security features, and better support for modern hardware.

2. **Web Browser:**
   - **Internet Explorer:** Version 11 (only if using older versions of Power BI).
   - **Google Chrome:** Latest version (preferred)
     - Chrome is the most widely supported browser for accessing Power BI's cloud services and interactive reports.
   - **Mozilla Firefox:** Version 106 or higher
     - Firefox can also be used as a reliable alternative for viewing Power BI dashboards and reports.

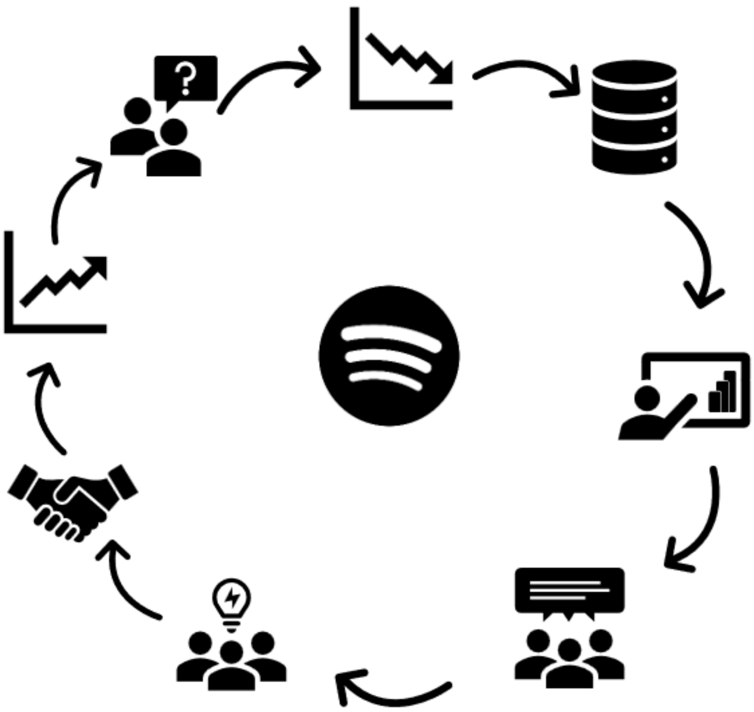
3. **Software Tools:**
   - **Power BI Desktop (June 2023 Version)**
     - Power BI Desktop is the primary tool for building reports and visualizations. The June 2023 version ensures that you have the latest features, updates, and security patches for working with Spotify's data and creating impactful dashboards.
   - **MS Office Excel (Version 2021)**
     - Excel is a valuable tool for data manipulation, initial cleaning, and analysis before importing data into Power BI. Version 2021 or later is recommended for compatibility with Power BI.
4. **Languages:**
   - **DAX (Data Analysis Expressions)**
     - DAX is a powerful formula language used in Power BI to create custom calculations, aggregate data, and build complex measures. Familiarity with DAX will allow for more dynamic and meaningful analysis of Spotify's dataset.

   - **SQL (Structured Query Language)**
     - SQL is essential for querying data stored in relational databases like Microsoft SQL Server or external data sources. Proficiency in SQL is

necessary for managing large datasets and extracting relevant data for Power BI.

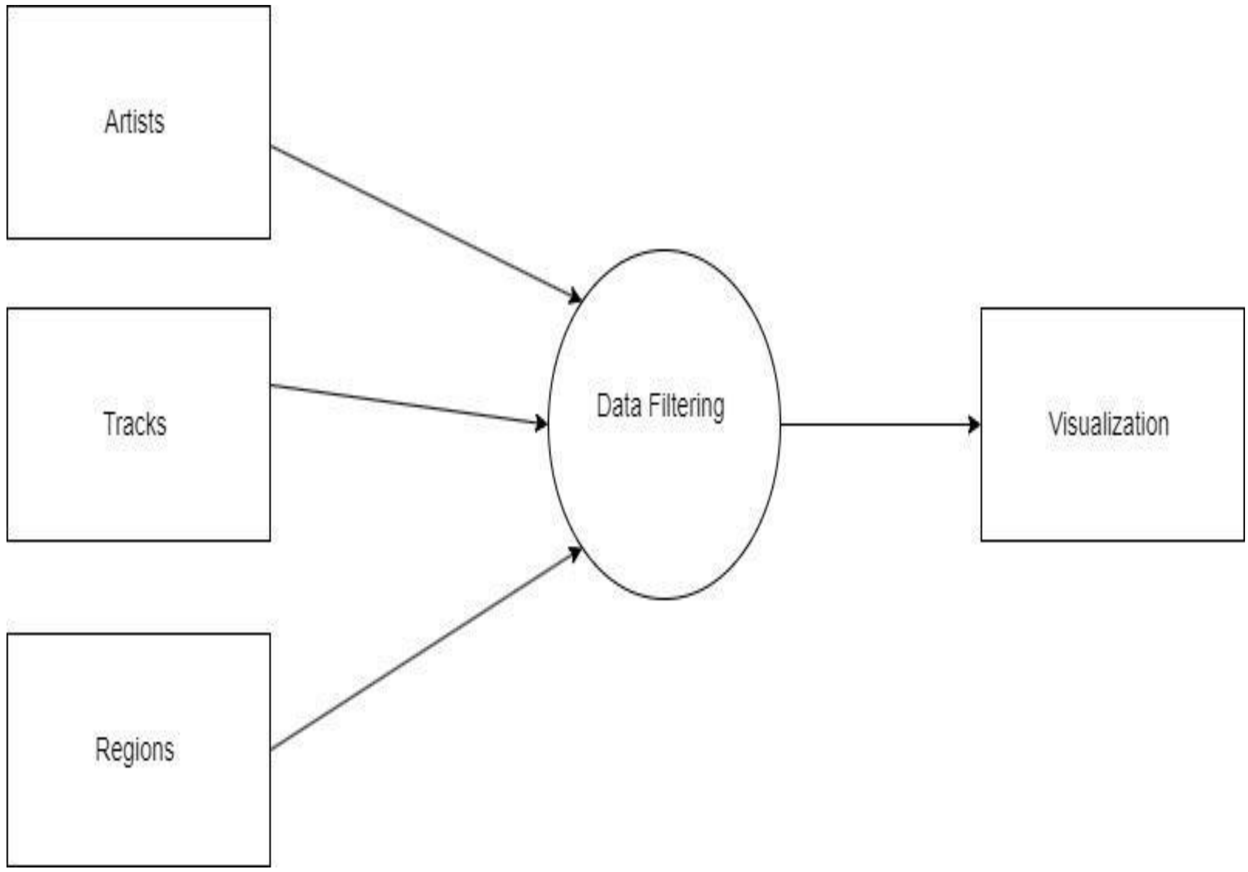
5. **Database Management Tools:**
   - **SQL Server Management Studio (SSMS) 19.0**
     - SSMS is used for managing Microsoft SQL Server databases, which may be involved in storing or querying Spotify's data. It provides an interface for performing queries, managing databases, and integrating data with Power BI.
   - **Microsoft SQL Server 19.0**
     - SQL Server is a relational database management system used for storing large volumes of structured data. It is useful for managing Spotify's data or other data sources that integrate into Power BI for detailed analysis and visualization.
6. **Additional Software:**
   - **Python/R (Optional)**
     - For advanced data manipulation, analysis, and integration of external data sources, Python or R can be used in conjunction with Power BI. Python can automate data processes and integrate machine learning models, while R can provide statistical analysis.
   - **Power Automate (Optional)**
     - Power Automate can help automate workflows, ensuring that new Spotify data is automatically extracted and loaded into Power BI without manual intervention. It can also trigger notifications and updates based on real-time data changes

DFD (DATA FLOW DIAGRAM)

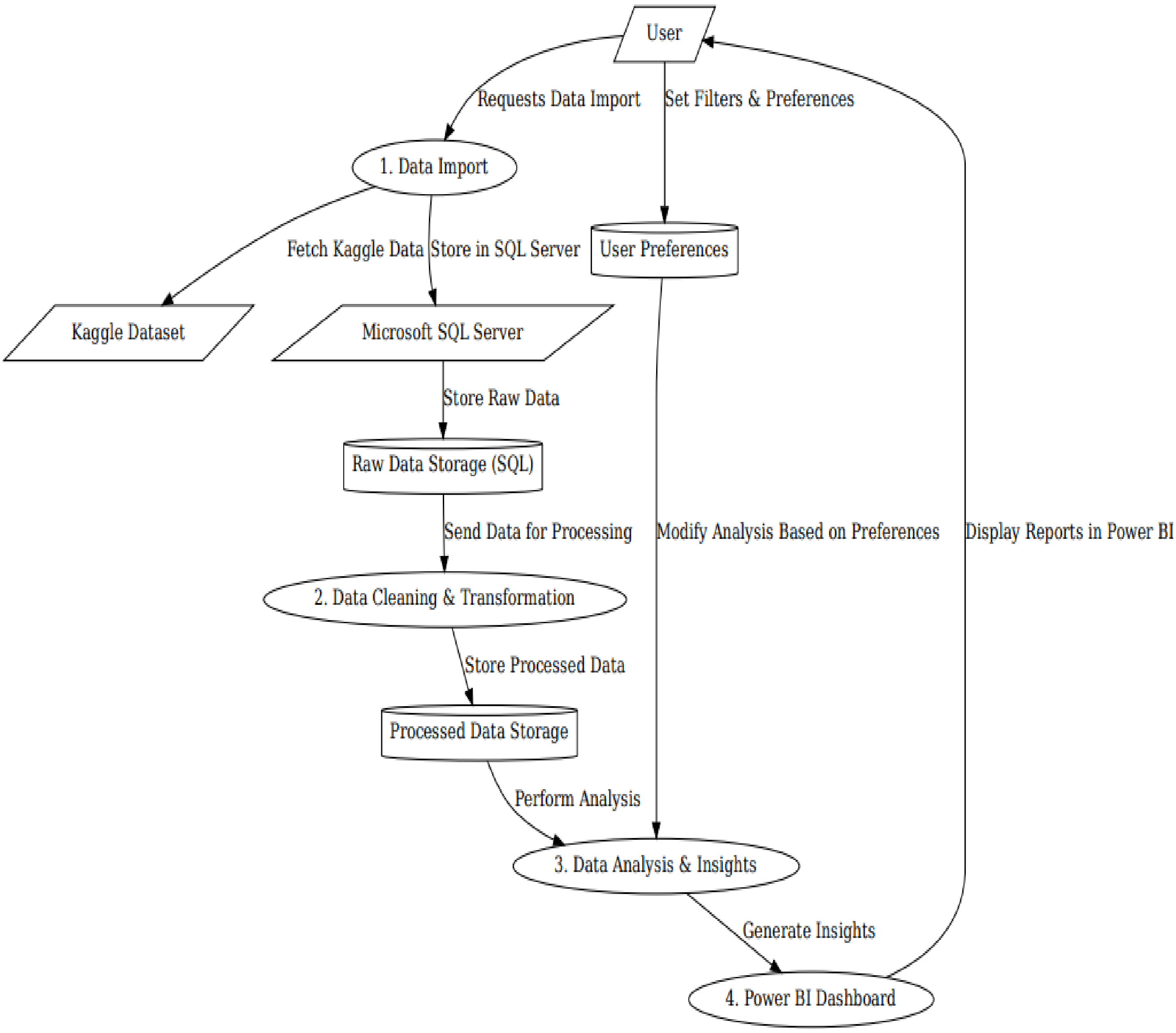


11. DFD (DATA FLOW DIAGRAM)

A} Level 0 DFD :-



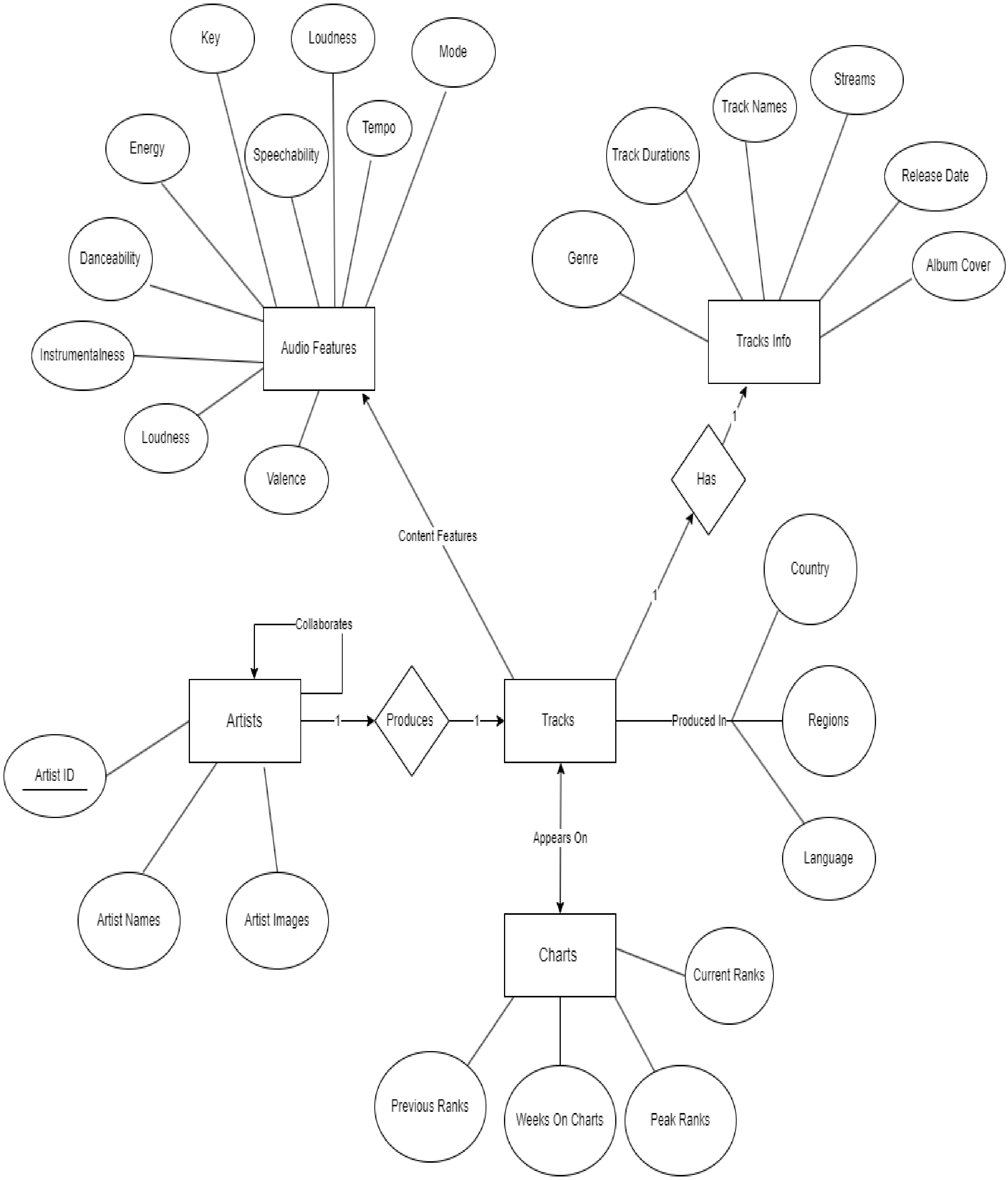
B} Level 1 DFD :-



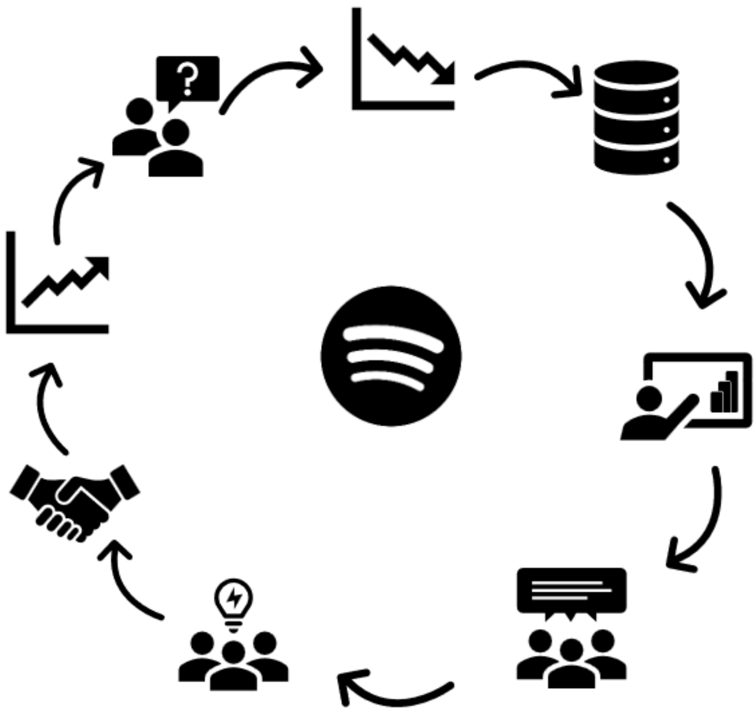
User
Requests Data Import
Set Filters & Preferences
1. Data Import
Fetch Kaggle Data
Store in SQL Server
User Preferences
Kaggle Dataset
Microsoft SQL Server
Store Raw Data
Raw Data Storage (SQL)
Send Data for Processing
Modify Analysis Based on Preferences
Display Reports in Power BI
2. Data Cleaning & Transformation
Store Processed Data
Processed Data Storage
Perform Analysis
3. Data Analysis & Insights
Generate Insights
4. Power BI Dashboard

ENTITY RELATIONSHIP (ER) DIAGRAM

12. ENTITY REALATIONSHIP (ER) DIAGRAM



FEASBILITY STUDY



13. FEASIBILITY STUDY

A Feasibility Study assesses the practicality and viability of a project by examining various factors that could impact its success. This includes technical feasibility**,** operational feasibility**,** financial feasibility**,** and legal and ethical considerations**.** For the Power BI with Spotify Data Analysis Project, we will analyze these aspects to determine if the project is viable and what resources are needed for its execution.

1. Technical Feasibility

1.1. Technology Requirements

The technical feasibility of the project hinges on the availability and compatibility of various tools and technologies required to carry out the analysis. The key technologies involved in this project are:

- **Power BI**: A business intelligence and data visualization tool that will be used for creating dashboards and generating insights.
- **Power BI Desktop**: Used to build reports locally.
- **Power BI Service**: Used for sharing and publishing reports to users.
- **Spotify Data**: The data required for analysis will come from Spotify's public datasets (available via CSV or the Spotify Web API).
- **Spotify Web API**: Provides real-time access to track, artist, and genre data.
- **Spotify Dataset**: CSV or pre-processed datasets available through public sources like Kaggle.
- **Data Sources**: Data will either be pulled from Spotify's API (real-time data) or downloaded from external repositories (e.g., Kaggle datasets).
- Data could include attributes like artist name, track popularity, genres, user behavior, and more.
- **Data Modeling**: Power BI will be used to create data models, relationships, and calculated fields.

1.2. Skills and Expertise

To execute this project, certain skills and expertise are required:

- **Power BI Skills**: Knowledge of Power BI's features (e.g., Power Query, DAX) for data cleaning, transformation, and visualization.
- **Spotify API Knowledge**: Understanding of how to connect Power BI to the Spotify Web API to pull live data, or manual data extraction if APIs are not viable.
- **Data Analysis**: Ability to clean, transform, and analyze data to extract meaningful insights.
- Since Power BI and Spotify's Web API are widely used tools, the technical skills required are accessible, and learning resources are available to bridge any gaps.

1.3. Infrastructure

- **Hardware and Software Requirements**:
  - Power BI Desktop (free), but for sharing and collaboration, Power BI Pro or Premium might be required.
  - A computer with internet access to use Power BI and download or access datasets via API.
- Cloud **Storage** may be required to store large datasets if the analysis involves handling significant volumes of data.

1.4. Integration and Compatibility

The integration between Power BI and the Spotify API is straightforward because Power BI can connect to RESTful APIs and external data sources (CSV files, Excel, etc.). However, developers may need to handle API rate limits and pagination when dealing with large datasets. Using CSV datasets can bypass these issues but limits real-time analysis.

2. Operational Feasibility

2.1. User Requirements

The system aims to meet the following user needs:

- **Music Curators**: Need insights to help with playlist curation and marketing strategies.
- **Artists**: Want to track their performance and engagement metrics.

- **Data Analysts**: Need the ability to perform ad-hoc analyses and generate insights from historical data.

By focusing on delivering interactive and user-friendly dashboards, Power BI's interface can fulfill these user requirements. It allows for filtering, drill-through, and customized reporting, making it easier for non-technical users to interpret the data.

2.2. Data Availability and Quality

- The quality of the data provided by Spotify, whether through the API or public datasets, is generally high. However, there may be challenges:
- **Missing or Incomplete Data**: Some datasets might have missing values, duplicates, or inconsistent data formats. This can be addressed through data cleaning and transformation within Power BI.
- **Data Granularity**: Data obtained via the Spotify API can be detailed and updated in real time, whereas static datasets might be limited to historical information, affecting the scope of analysis.

2.3. User Access and Training

- The system will be accessible via the Power BI Service. Power BI reports can be shared with authorized users across devices (desktop, mobile).
- A minimal training program or documentation will be required for users unfamiliar with Power BI. Given Power BI's intuitive interface, users should be able to navigate the dashboards with basic instructions.

3. Financial Feasibility

3.1. Cost Estimation

- **Power BI Licenses**:
- Power BI Desktop is free, but for sharing reports with others, Power BI Pro is required. The cost of Power BI Pro is around $10 per user per month.
- If there is a need for collaboration with a larger team, Power BI Premium can be considered, which is priced starting at $20 per user per month or higher based on organizational needs.

- **API Usage**:
- Spotify's Web API is free for a limited number of calls. For larger data pulls, it might require additional configuration, but for this project, the free tier of the API should be sufficient.
- **Data Storage Costs**:
- Depending on the volume of data, storage may be needed on cloud platforms (e.g., Azure Blob Storage or OneDrive). If using large datasets, there may be storage costs. However, for most small to medium-sized datasets, the cost is minimal.
- **Developer and Analyst Time**:
- Time will be needed to integrate the data, clean it, and develop the visualizations. An estimate of 20-40 hours of development work might be needed depending on the dataset size, complexity, and user feedback.

3.2. Return on Investment (ROI)

- The benefits of the project can be realized through the actionable insights it provides:
- **Improved Decision-Making**: Music curators, marketers, and analysts can make better data-driven decisions related to playlists, promotions, and artist engagement.
- **Increased Engagement**: Artists can track their performance, which can help them refine their strategies.
- **Operational Efficiency**: Automated reporting and real-time analytics reduce the need for manual data processing and reporting.
- The initial investment for tools and time is relatively low compared to the potential benefits in terms of operational efficiency and business insights.

4. Legal and Ethical Feasibility

4.1. Data Privacy and Compliance

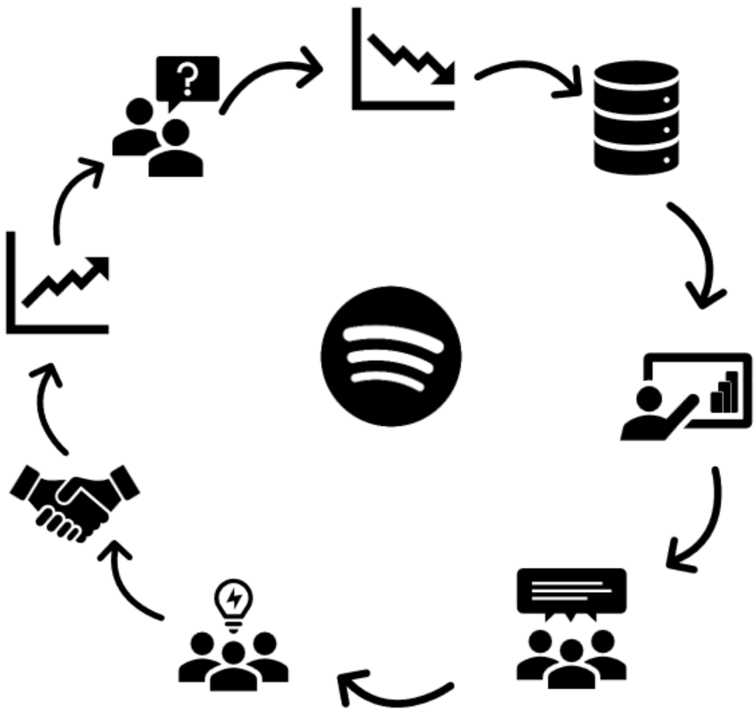
- **Data Privacy**: The project will use publicly available datasets or the Spotify API, which should not contain any personal data. However, it is essential to ensure that user privacy is respected when handling data.
- **Spotify's Terms of Service**: The Spotify API must be used according to Spotify's terms and conditions. Any commercial use of the data must comply with their usage policies.

- **Data Protection Laws**: The project must ensure compliance with GDPR (General Data Protection Regulation) if it is processing any user data related to individuals in the EU.

4.2. Ethical Considerations

- The project will only work with publicly available data, ensuring there are no ethical issues related to unauthorized data use.
- Transparency in data collection and reporting will be ensured to provide accurate and fair insights to all users.

DATA DICTIONARY



14. DATA DICTIONARY

A Data Dictionary is a documentation of all the fields (data elements) used in the project. It defines the data attributes, their meanings, types, and other relevant details. Below is a Data Dictionary for the Power BI with Spotify Data Analysis Project that includes the data attributes from the Spotify dataset and how they are utilized within the analysis.

1.Table: Tracks

This table contains information about individual tracks available on Spotify.

Field Name	Description	Data Type	Example
Track_ID	Unique identifier for the track.	Integer	12345
Track_Name	Name of the track.	String	"Blinding Lights"
Artist_ID	Unique identifier for the artist performing the track.	Integer	9876
Artist_Name	Name of the artist performing the track.	String	"The Weeknd"
Album_Name	Name of the album that the track belongs to.	String	"After Hours"
Genre	Genre of the track (e.g., pop, rock, hip-hop).	String	"Pop"
Release_Date	The release date of the track.	Date	2020-03-20
Duration	Duration of the track in seconds.	Integer	200 (seconds)
Popularity	Popularity score of the track (0 to 100).	Integer	88
Danceability	How suitable the track is for dancing (0 to 1 scale).	Decimal	0.85
Energy	Energy of the track (0 to 1 scale).	Decimal	0.75

Field Name	Description	Data Type	Example
Tempo	Tempo of the track in beats per minute (BPM).	Integer	120
Loudness	Overall loudness of the track in decibels (dB).	Decimal	-5.3
Valence	Musical positiveness of the track (0 to 1 scale).	Decimal	0.75

2. Table: Artists

This table contains information about artists who have tracks on Spotify.

Field Name	Description	Data Type	Example
Artist_ID	Unique identifier for the artist.	Integer	9876
Artist_Name	Name of the artist.	String	"The Weeknd"
Artist_Genre	Primary genre of the artist (e.g., pop, rock).	String	"Pop"
Popularity	Popularity score of the artist (0 to 100).	Integer	92
Followers	Number of followers the artist has on Spotify.	Integer	15000000
Country	The country where the artist is primarily based.	String	"Canada"

3. Table: Genres

This table provides information about different music genres that tracks can belong to.

Field Name	Description	Data Type	Example
Genre_ID	Unique identifier for the genre.	Integer	1
Genre_Name	Name of the genre (e.g., pop, rock, jazz).	String	"Pop"
Genre_Popularity	Popularity score of the genre (0 to 100).	Integer	85

4. Table: Playlists

This table contains information about playlists containing tracks.

Field Name	Description	Data Type	Example
Playlist_ID	Unique identifier for the playlist.	Integer	4567
Playlist_Name	Name of the playlist.	String	"Top Hits"
Creator_Name	Name of the creator or curator of the playlist.	String	"Spotify"
Track_Count	Number of tracks in the playlist.	Integer	100
Total_Streams	Total number of streams of all tracks in the playlist.	Integer	50000000
Playlist_Popularity	Popularity score of the playlist (0 to 100).	Integer	90

5. Table: User Listening Behavior (Optional)

This table would contain information about users' listening habits (if available).

Field Name	Description	Data Type	Example
User_ID	Unique identifier for the user.	Integer	123456
Track_ID	ID of the track the user listened to.	Integer	12345
Listen_Count	Number of times the user has listened to the track.	Integer	25
Device	The device the user used to listen (e.g., mobile, desktop).	String	"Mobile"
Listening_Time	Duration the user listened to the track (in seconds).	Integer	180
Timestamp	Time when the track was played.	DateTime	2025-02-26 12:34:56

6. Table: User Demographics (Optional)

This table would contain information about user demographics (if available).

Field Name	Description	Data Type	Example
User_ID	Unique identifier for the user.	Integer	123456
Age	Age of the user.	Integer	25
Country	The country where the user is located.	String	"USA"

Field Name	Description	Data Type	Example
Gender	Gender of the user.	String	"Male"
Subscription_Type	Type of Spotify subscription the user has (e.g., free, premium).	String	"Premium"

7. Relationships Between Tables

- **Tracks to Artists**: Each track is linked to one artist, but an artist can have multiple tracks.
  Track.Artist_ID → Artist.Artist_ID

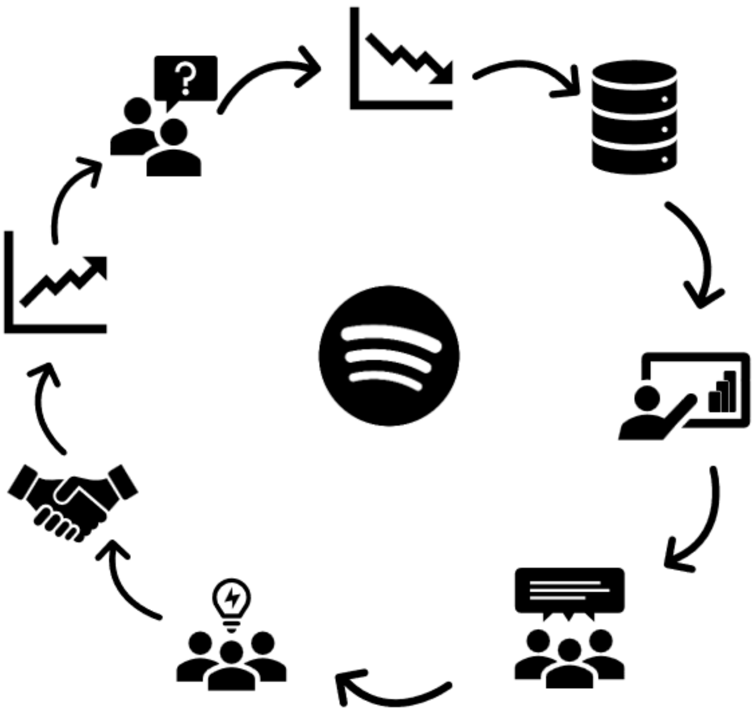
- **Tracks to Genres**: A track can belong to multiple genres. This would require a many-to-many relationship table if considering genre tags.
  Track.Genre (or Track.Genre_ID in a new table) → Genre.Genre_ID

- **Tracks to Playlists**: A track can appear in multiple playlists.
  Track.Track_ID → Playlist.Track_ID

- **Tracks to User Listening Behavior**: A track's listening behavior can be tracked per user.
  Track.Track_ID → User_Listen.Track_ID
  User_Listen.User_ID → User.User_ID

- **Artists to Playlists**: An artist’s tracks can appear in many playlists.
  Artist.Artist_ID → Playlist.Track_ID

PROBLEM STATEMENTS WITH SQL QUERIES AND OUTPUT



15. PROBLEM STATEMENTS WITH SQL QUERIES AND OUTPUT

PROBLEM STATEMENTS

In the highly competitive music streaming industry, understanding factors that drive artist popularity, track performance, and listener preferences is essential for artists, record labels, and streaming platforms. With millions of tracks available on platforms like Spotify, manually identifying trends and optimizing music recommendations is challenging.

This project aims to analyze Spotify data to extract meaningful insights into artist rankings, track popularity, and audio features such as danceability, tempo, and energy. By leveraging Power BI for data visualization, the analysis will uncover patterns that influence track success and listener engagement. The findings will help stakeholders make data-driven decisions**,** optimize music recommendations, and enhance audience reach through actionable insights.

SQL QUERIES

1. Total Tracks

   SELECT COUNT(*) AS total_tracks FROM cleaned_data;

	total_tracks
1	721468

2. Total Unique Artists

   SELECT COUNT(DISTINCT artist_names) AS total_artists FROM cleaned_data;

	total_artists
1	8794

3. Total Collaborations

   SELECT COUNT(*) AS total_collaborations FROM cleaned_data WHERE collab = 1;

	total_collaborations
1	439112

4. Tracks Achieving Peak Rank

SELECT COUNT(*) AS peak_rank_tracks FROM cleaned_data WHERE peak_rank = 1;

	peak_rank_tracks
1	71958

5. Total Streams

SELECT SUM(CAST(streams AS BIGINT)) AS total_streams FROM cleaned_data;

	total_streams
1	463574795229

6. Tracks by Year

SELECT YEAR(release_date) AS release_year, COUNT(*) AS track_count

FROM cleaned_data

GROUP BY YEAR(release_date);

	release_year	track_count
4	2004	1801
5	1981	14
6	2010	1445
7	1967	3
8	1987	944
9	2007	1249
10	2001	210
11	1984	806

7. Most Streamed Tracks

SELECT TOP 10 track_name, MAX(CAST(streams AS BIGINT)) AS max_streams

FROM cleaned_data

GROUP BY track_name

ORDER BY max_streams DESC;

	track_name	max_streams
3	In My Feelings	67499798
4	Glimpse of Us	57470492
5	God's Plan	54891573
6	DĂKITI	53344093
7	Blinding Lig...	52375259
8	Dance Monk...	52055226
9	Moscow Mule	51387060
10	First Class	50438229

8. Tracks per Artist

SELECT artist_names, COUNT(*) AS track_count

FROM cleaned_data

GROUP BY artist_names

ORDER BY track_count DESC;

	artist_names	track_count
7	Doja Cat	5925
8	Super Yei, Jone Quest, Sammy, Myke Towers, Lenn...	5850
9	The Neighbourhood	5539
10	ATB, Topic, A7S	4878
11	Billie Eilish	4589
12	J Balvin, KAROL G, Nicky Jam, Crissin, Totoy El Frio,...	4512
13	Harry Styles	4504
14	Maluma, BeĂ©le, Rauw Alejandro, Mambo Kingz, ...	4254

9. Streams by Genre

SELECT artist_genre, SUM(CAST(streams AS BIGINT)) AS total_streams

FROM cleaned_data

GROUP BY artist_genre;

	artist_genre	total_streams
1	rap	12251002739
2	musica mexicana	431643932
3	brooklyn indie	24882803
4	spiritual hip hop	10843878
5	uk alternative pop	6301725
6	british invasion	1181829
7	birmingham hip ...	1171533
8	viet remix	2891521

10. Tracks by Region

SELECT region, COUNT(*) AS track_count

FROM cleaned_data

GROUP BY region;

	region	track_count
1	Oceania	15288
2	Caribbean	16476
3	Europe	287583
4	Central A...	78650
5	North A...	24124
6	South A...	115258
7	Africa	30841
8	Middle E...	43521

11. Average Track Duration

```
SELECT AVG(CAST(duration AS FLOAT)) AS avg_duration
FROM cleaned_data;
```

	avg_duration
1	211875.092554348

12. Collaborations by Artist

```
SELECT artist_names, COUNT(*) AS collaboration_count
FROM cleaned_data
WHERE collab = 1
GROUP BY artist_names
ORDER BY collaboration_count DESC;
```

	artist_names	collaboration_count
4	Super Yei, Jone Quest, Sammy, Myke Towers, Lenn...	5850
5	ATB, Topic, A7S	4878
6	J Balvin, KAROL G, Nicky Jam, Crissin, Totoy El Frio,...	4512
7	Maluma, BeÃ©le, Rauw Alejandro, Mambo Kingz, ...	4254
8	Joel Corry, Jax Jones, Charli XCX, Saweetie	4116
9	Jay Wheeler, DJ Nelson, Myke Towers	3978
10	DÃ-melo Flow, Dalex, Sech, Justin Quiles, Arcangel...	3801
11	Joel Corry, MNEK	3734

13. Streams by Release Quarter

```
SELECT DATEPART(QUARTER, release_date) AS release_quarter,
SUM(CAST(streams AS BIGINT)) AS total_streams
FROM cleaned_data
GROUP BY DATEPART(QUARTER, release_date);
```

	release_quarter	total_streams
1	3	108975676750
2	1	108687648406
3	4	105965882271
4	2	139945587802

14. Tracks by Tempo Range

```
SELECT
CASE
WHEN tempo < 90 THEN 'Slow'
WHEN tempo BETWEEN 90 AND 120 THEN 'Moderate'
ELSE 'Fast'
END AS tempo_range, COUNT(*) AS track_count
FROM cleaned_data
GROUP BY
CASE
WHEN tempo < 90 THEN 'Slow'
WHEN tempo BETWEEN 90 AND 120 THEN 'Moderate'
ELSE 'Fast'
END;
```

	tempo_range	track_count
1	Slow	70069
2	Moderate	295930
3	Fast	355469

15. Top Genres by Year

```
SELECT YEAR(release_date) AS release_year, artist_genre, SUM(CAST(streams AS
BIGINT)) AS total_streams
FROM cleaned_data
GROUP BY YEAR(release_date), artist_genre
ORDER BY release_year, total_streams DESC;
```

	release_year	artist_genre	total_streams
4	1942	easy listening	79680289
5	1944	adult standards	10293389
6	1944	vocal jazz	2444553
7	1944	torch song	811469
8	1944	lounge	808245
9	1953	novelty	278419
10	1957	rockabilly	88125369
11	1957	rock-and-roll	82931044

16. Energy and Danceability Correlation

SELECT energy, danceability

FROM cleaned_data;

	energy	danceability
	0.547999973065491	0.870000004768372
2	0.695999979972839	0.760999977588654
3	0.731000006198883	0.519999980926514
4	0.467000007629395	0.771000027656555
5	0.736000001430511	0.811999976634979
6	0.824000000953674	0.852999985218048
7	0.453000009059906	0.657999992370605
8	0.680000007152557	0.796000003814697

17. Streams by Country

SELECT country, SUM(CAST(streams AS BIGINT)) AS total_streams

FROM cleaned_data

GROUP BY country

ORDER BY total_streams DESC;

	country	total_streams
1	Global	302210654739
2	United States	18364410440
3	Mexico	14192694281
4	Spain	11139636525
5	Brazil	10453281940
6	Germany	9848632018
7	Chile	7748071064
8	India	7601613875

18. Popular Languages

SELECT language, COUNT(*) AS track_count

FROM cleaned_data

GROUP BY language

ORDER BY track_count DESC;

	language	track_count
1	Spanish	228097
2	English	56362
3	Arabic	35389
4	Global	34273
5	German	33551
6	Greek	22189
7	French	19284
8	Dutch	18813

19. Weekly Stream Trends

```
SELECT week, SUM(CAST(streams AS BIGINT)) AS total_streams
FROM cleaned_data
GROUP BY week
ORDER BY week;
```

	week	total_streams
1	2016-12-29	645514543
2	2017-01-05	686039996
3	2017-01-12	676118219
4	2017-01-19	707662788
5	2017-01-26	683480757
6	2017-02-02	693969746
7	2017-02-09	681627591
8	2017-02-16	671727363
9	2017-02-23	689106338
10	2017-03-02	759202398

20. Top Artists with Peak Tracks

```
SELECT artist_names, COUNT(*) AS peak_track_count
FROM cleaned_data
WHERE peak_rank = 1
GROUP BY artist_names
ORDER BY peak_track_count DESC;
```

	artist_names	peak_track_count
1	Bad Bunny, Jhay Cortez	2878
2	Nio Garcia, Anuel AA, Myke Towers, Brray, Juanka	2575
3	The Weeknd	2377
4	Olivia Rodrigo	2342
5	Anuel AA, Daddy Yankee, KAROL G, Ozuna, J Ba...	2260
6	Bad Bunny, Tainy	2022
7	Bad Bunny	1935
8	KAROL G, Nicki Minaj	1854

21. Genre-Wise Avg Duration

```
SELECT artist_genre, AVG(CAST(duration AS FLOAT)) AS avg_duration
FROM cleaned_data
GROUP BY artist_genre;
```

	artist_genre	avg_duration
1	rap	217007.168843668
2	musica mexicana	229450.640256959
3	brooklyn indie	129979.692771084
4	spiritual hip hop	180893
5	uk alternative pop	408322.75
6	british invasion	185733
7	birmingham hip ...	159668
8	viet remix	196702.306666667

22. Longest Tracks

SELECT TOP 10 track_name, duration

FROM cleaned_data

ORDER BY duration DESC;

	track_name	duration
1	SAS PLUS / SAS PUSSY	1787030
2	SAS PLUS / SAS PUSSY	1787030
3	NĂ½tt lĂf	1203490
4	NĂ½tt lĂf	1203490
5	NĂ½tt lĂf	1203490
6	NĂ½tt lĂf	1203490
7	NĂ½tt lĂf	1203490
8	NĂ½tt lĂf	1203490

23. Streams Distribution by Danceability

SELECT danceability, CAST(streams AS BIGINT) AS streams

FROM cleaned_data;

	danceability	streams
1	0.870000004768372	2080139
2	0.760999977588654	1923270
3	0.519999980926514	1555631
4	0.771000027656555	1149499
5	0.811999976634979	1104997
6	0.852999985218048	1004471
7	0.657999992370605	870219
8	0.796000003814697	643830

24. Streams per Region

SELECT region, SUM(CAST(streams AS BIGINT)) AS total_streams

FROM cleaned_data

GROUP BY region

ORDER BY total_streams DESC;

	region	total_streams
1	Global	302210654739
2	Europe	57387436564
3	North America	35439691997
4	South America	34148024802
5	Asia	19660458498
6	Middle East	6133976351
7	Oceania	3662289157
8	Central America	3200523079
9	Africa	1203176760
10	Caribbean	528563282

25. Most Streamed Artists by Language

```
SELECT language, artist_names, SUM(CAST(streams AS BIGINT)) AS total_streams
FROM cleaned_data
GROUP BY language, artist_names
ORDER BY total_streams DESC;
```

	language	artist_names	total_streams
4	Global	Post Malon...	4151661670
5	Global	XXXTENTA...	3941464825
6	Global	Bad Bunny	3811207900
7	Global	Harry Styles	3702968543
8	Global	DJ Snake, ...	3584258080
9	Global	Nio Garcia,...	3483394908
10	Global	Anuel AA, D...	3304757690
11	Global	Olivia Rodri...	3119544427

26. Valence and Tempo Correlation

```
SELECT valence, tempo
FROM cleaned_data;
```

	valence	tempo
4	0.256000012159348	100.088996887207
5	0.395999997854233	91.9929962158203
6	0.693000018596649	100.050003051758
7	0.358000010251999	126.042999267578
8	0.709999978542328	179.923004150391
9	0.352999985218048	140.039993286133
10	0.335999995470047	127.96199798584
11	0.717000007629395	102

27. Monthly Track Releases

```
SELECT MONTH(release_date) AS release_month, COUNT(*) AS track_count
FROM cleaned_data
GROUP BY MONTH(release_date);
```

	release_month	track_count
1	9	61340
2	3	72460
3	12	43052
4	6	69878
5	7	56421
6	1	68494
7	10	58741
8	4	54399

28. Top Collaborators

```
SELECT artist_names, COUNT(*) AS collab_count
FROM cleaned_data
WHERE collab = 1
GROUP BY artist_names
ORDER BY collab_count DESC;
```

	artist_names	collab_count
10	DÃ-melo Flow, Dalex, Sech, Justin Quiles, Arcangel...	3801
11	Joel Corry, MNEK	3734
12	Sech, Daddy Yankee, J Balvin, ROSALÃA, Farruko	3705
13	Bad Bunny, Jhay Cortez	3616
14	Lost Frequencies, Calum Scott	3294
15	Wisin, Jhay Cortez, Anuel AA, Los Legendarios, Myk...	3095
16	Joel Corry, RAYE, David Guetta	3072
17	SAINt JHN, Imanbek	3024

29. Speechiness vs. Energy

```
SELECT speechiness, energy
FROM cleaned_data;
```

	speechiness	energy
148	0.0599000006914139	0.526000022888184
149	0.0272000003606081	0.717000007629395
150	0.025000000372529	0.501999974250793
151	0.305000007152557	0.611999988555908
152	0.0253999996930361	0.388999998569489
153	0.0283000003546476	0.800000011920929
154	0.28999999165535	0.508000016212463
155	0.0348000004887581	0.451999992132187

30. Total Streams by Key

```
SELECT [key], SUM(CAST(streams AS BIGINT)) AS total_streams
FROM cleaned_data
GROUP BY [key];
```

	key	total_streams
1	9	35005758713
2	3	15357092863
3	6	40284090809
4	7	48875274990
5	1	67031773228
6	10	25772180830
7	4	37743960710
8	5	37230398502

31. Artists with the Most Weeks on Chart

```
SELECT artist_names,
       SUM(weeks_on_chart) AS total_weeks
FROM cleaned_data
GROUP BY artist_names
ORDER BY total_weeks DESC;
```

Results | Messages

	artist_names	total_weeks
1	Bad Bunny	771266
2	Super Yei, Jone Quest, Sammy, Myke Towers, Lenny ...	669192
3	The Weeknd	484508
4	Riton, Nightcrawlers, Mufasa & Hypeman, Dopamine	295320
5	J Balvin, Bad Bunny	280692
6	Bad Bunny, Tainy	277596
7	Jay Wheeler, DJ Nelson, Myke Towers	270576
8	The Neighbourhood	237306
9	SAINt JHN, Imanbek	235988
10	Anuel AA, Daddy Yankee, KAROL G, Ozuna, J Balvin	230995
11	Joel Corry, MNEK	207920
12	Olivia Rodrigo	207719
13	Doja Cat, SZA	204572
14	Elton John, Dua Lipa, PNAU	201363
15	Tones And I	196804
16	The Chainsmokers, Coldplay	188374

32. Tracks with the Most Rank Changes

```
SELECT TOP 10 track_name,
       ABS(peak_rank - previous_rank) AS rank_change
FROM cleaned_data
ORDER BY rank_change DESC;
```

Results | Messages

	track_name	rank_change
1	Tan Bonita	201
2	REI | Mission 01	201
3	REI | Mission 01	201
4	When It Rains It Pours	201
5	Eventually	201
6	Red Eye (feat. TroyBoi)	201
7	Red Eye (feat. TroyBoi)	201
8	Love It When You Hate Me (feat. blackbear)	201
9	Love It When You Hate Me (feat. blackbear)	201
10	Questions	201

33. Albums with the Most Tracks

```
SELECT album_num_tracks,
       COUNT(*) AS album_count
FROM cleaned_data
GROUP BY album_num_tracks
ORDER BY album_count DESC;
```

Results | Messages

	album_num_tracks	album_count
1	1	333809
2	14	48688
3	16	35383
4	12	30584
5	10	29360
6	15	28487
7	11	27332
8	20	23637
9	18	19209
10	2	15331
11	13	13062
12	8	12418
13	17	12217
14	9	10519
15	7	8625
16	21	8237
17	22	[illegible]

34. Most Common Source of Streams

```
SELECT source,
       COUNT(*) AS stream_source_count
FROM cleaned_data
GROUP BY source
```

```sql
ORDER BY stream_source_count DESC;
```

	source	stream_source_count
1	Rimas Entertainment LLC	29774
2	Columbia	23960
3	Sony Music Latin	22982
4	Kemosabe Records/RCA Records	16631
5	Atlantic Records UK	13749
6	Republic Records	12000
7	Rich Music	11301
8	Atlantic Records	10833
9	Ministry of Sound Recordings	10551
10	Sony Music Latin/Duars Entertainment	9909
11	EMI	9570
12	Sony Music Entertainment	8720
13	UMLE - Latino	8424
14	Olivia Rodrigo PS	7863
15	SueÃ±os Globales, LLC	7684
16	Polydor Records	7128

35. Most Featured Artists in Collaborations

```sql
SELECT artist_individual,
       COUNT(*) AS collab_count
FROM cleaned_data
WHERE collab = 1
GROUP BY artist_individual
ORDER BY collab_count DESC;
```

	artist_individual	collab_count
1	Bad Bunny	10961
2	J Balvin	7767
3	Rauw Alejandro	7458
4	Myke Towers	6093
5	Doja Cat	5977
6	Anuel AA	4810
7	Ozuna	4416
8	Lenny TavÃ¡rez	4008
9	Dua Lipa	3992
10	Joel Corry	3974
11	SZA	3753
12	Sech	3604
13	Jhay Cortez	3376
14	Maria Becerra	3051
15	Jay Wheeler	2975
16	Elton John	2938

36. Distribution of Tracks by Loudness Level

```
SELECT
CASE
   WHEN loudness > -5 THEN 'Very Loud'
   WHEN loudness BETWEEN -10 AND -5 THEN 'Moderately Loud'
   ELSE 'Soft'
END AS loudness_category,
COUNT(*) AS track_count
FROM cleaned_data
GROUP BY
CASE
   WHEN loudness > -5 THEN 'Very Loud'
   WHEN loudness BETWEEN -10 AND -5 THEN 'Moderately Loud'
   ELSE 'Soft'
END;
```

Results | Messages

	loudness_category	track_count
1	Soft	64793
2	Very Loud	265593
3	Moderately Loud	391082

37. Average Tempo per Genre

```
SELECT artist_genre,
      AVG(tempo) AS avg_tempo
FROM cleaned_data
GROUP BY artist_genre
ORDER BY avg_tempo DESC;
```

Results | Messages

	artist_genre	avg_tempo
1	vbs	204.243003845215
2	japanese punk rock	194.957000732422
3	j-punk	194.957000732422
4	czech punk	189.281997680664
5	finnish choir	188.011993408203
6	japanese pop punk	186.966499328613
7	rock brasiliense	186.00700378418
8	brazilian rock	186.00700378418
9	japanese viral pop	184.40251658513
10	broadway	183.855773925781
11	tunisian pop	182.763045601223
12	barnmusik	182.661281444408
13	pop paraguayo	181.964004516602
14	torch song	180.886993408203
15	chinese metalcore	180.069000244141
16	salsa peruana	179.891400146484
17	vegan indie	179.626182036503

38. Countries with the Most Unique Artists

```
SELECT country,
       COUNT(DISTINCT artist_names) AS unique_artists
FROM cleaned_data
GROUP BY country
ORDER BY unique_artists DESC;
```

Results | Messages

	country	unique_artists
1	Global	961
2	Germany	684
3	Switzerland	624
4	South Africa	591
5	Finland	568
6	Austria	551
7	Luxembourg	500
8	Estonia	498
9	Norway	493
10	Netherlands	487
11	Lithuania	482
12	Sweden	472
13	Latvia	452
14	Ukraine	427
15	Cyprus	412
16	Greece	409
17	France	406

39. Most Popular Genres in Different Regions

```
SELECT region,
       artist_genre,
       COUNT(*) AS genre_count
FROM cleaned_data
GROUP BY region, artist_genre
ORDER BY region, genre_count DESC;
```

Results | Messages

	region	artist_genre	genre_count
1	Africa	pop	4426
2	Africa	amapiano	4212
3	Africa	moroccan pop	2562
4	Africa	rap maroc	2201
5	Africa	afro soul	1821
6	Africa	nigerian pop	1688
7	Africa	south african pop	1528
8	Africa	arabic hip hop	648
9	Africa	dance pop	600
10	Africa	rai	498
11	Africa	rap	420
12	Africa	afro dancehall	407
13	Africa	afropop	395
14	Africa	afro r&b	316
15	Africa	pop urbaine	315
16	Africa	south african house	313
17	Africa	canadian pop	263

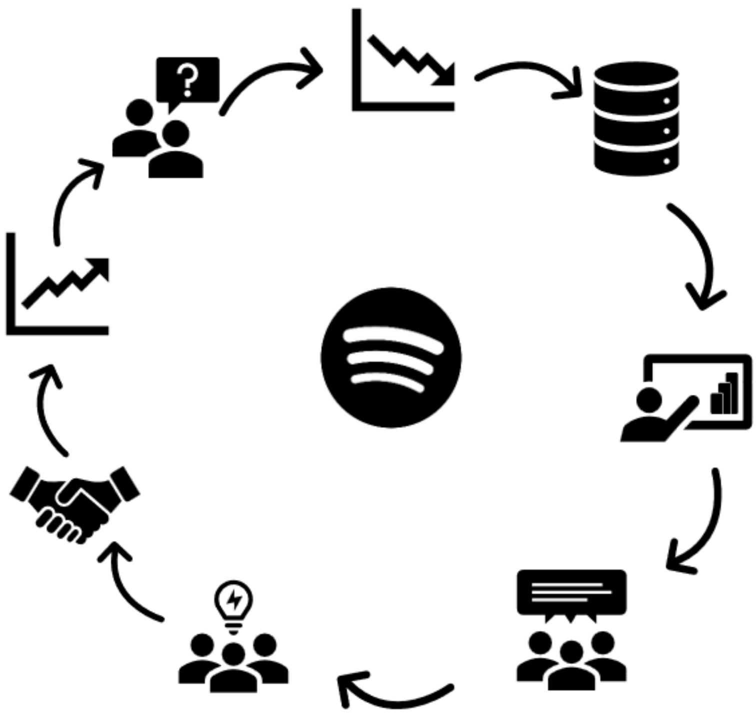
40. Tracks with the Highest Acousticness

```
SELECT TOP 10 track_name,
       acousticness
FROM cleaned_data
ORDER BY acousticness DESC;
```

Results | Messages

	track_name	acousticness
1	Yarning Many - Skit	0.995000004768372
2	Yarning Many - Skit	0.995000004768372
3	let it snow	0.994000017642975
4	let it snow	0.994000017642975
5	let it snow	0.994000017642975
6	let it snow	0.994000017642975
7	Intro	0.994000017642975
8	let it snow	0.994000017642975
9	let it snow	0.994000017642975
10	let it snow	0.994000017642975

DAX FUNCTIONS



16. DAX FUNCTIONS

DAX QUERY IMPLEMENTATION TO REPLACE VALUES

Advanced Editor

cleaned_data

```
// Replace numeric values in 'key' column with musical representations
#"Replaced Key Values" = Table.TransformColumns(
    dbo_cleaned_data,
    {
        {"key", each if _ = 0 then "C"
        else if _ = 1 then "C♯/D♭"
        else if _ = 2 then "D"
        else if _ = 3 then "D♯/E♭"
        else if _ = 4 then "E"
        else if _ = 5 then "F"
        else if _ = 6 then "F♯/G♭"
        else if _ = 7 then "G"
        else if _ = 8 then "G♯/A♭"
        else if _ = 9 then "A"
        else if _ = 10 then "A♯/B♭"
        else if _ = 11 then "B"
        else null,
        type text}
    }
),

// Replace boolean values in 'mode' column with Major/Minor
#"Replaced Mode Values" = Table.TransformColumns(
    #"Replaced Key Values",
    {
        {"mode", each if _ = true then "Major"
        else if _ = false then "Minor"
        else null,
        type text}
    }
),

// Replace numeric values in 'collab' column with YES/NO
#"Replaced Collab Values" = Table.TransformColumns(
    #"Replaced Mode Values",
    {
        {"collab", each if _ = 1 then "YES"
        else if _ = 0 then "NO"
        else null,
```

✓ No syntax errors have been detected.

DAX Query OUTPUT

key	mode	1.2
A♯/B♭	Minor	
G	Minor	
F♯/G♭	Minor	
F	Minor	
E	Minor	
D	Major	
A	Minor	
A♯/B♭	Minor	
E	Major	
D♯/E♭	Minor	
F	Minor	
C♯/D♭	Minor	
B	Minor	
D♯/E♭	Major	
G	Major	
C♯/D♭	Major	
C♯/D♭	Minor	
A	Major	
A	Minor	
E	Minor	
G	Major	
A♯/B♭	Minor	
C♯/D♭	Minor	
C♯/D♭	Minor	
D	Major	
B	Major	
C♯/D♭	Minor	

1. Count Of Total Collabs

   TotalCollaborations =

   COUNTROWS(FILTER(cleaned_data, cleaned_data[collab] = "YES"))

2. Count Of non Collabs

   NoCollabCount =

   CALCULATE(

   COUNTROWS('cleaned_data'),

   'cleaned_data'[collab] = "No"

)

collab
NO
NO
NO
NO
YES
YES
YES
YES
YES
YES
YES
YES

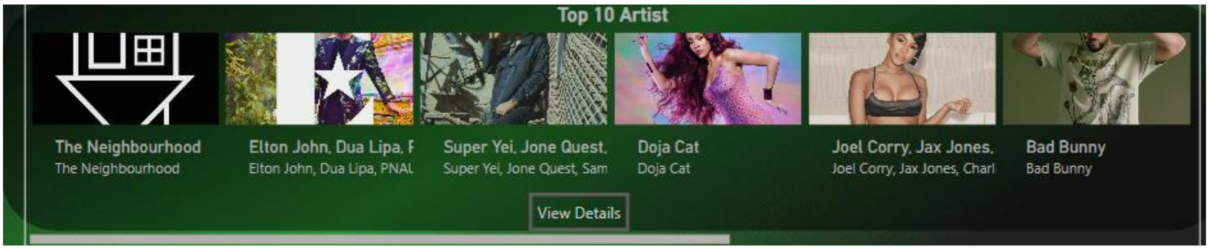
3. CALCULATED COLUMN TO CONVERT ARTIST IDs of spotify app to HTTPS

```
Converted Artist URL =
VAR ArtistID = SUBSTITUTE(cleaned_data[artist_id], "spotify:artist:", "")
RETURN
IF(
NOT(ISBLANK(ArtistID)),
"https://open.spotify.com/artist/" & ArtistID & "?utm_source=embed_player",
BLANK()
)
```

Converted Artist URL
https://open.spotify.com/artist/1mcTU81TzQhprhouKaTkpq?utm_source=embed_player
https://open.spotify.com/artist/1mcTU81TzQhprhouKaTkpq?utm_source=embed_player
https://open.spotify.com/artist/1mcTU81TzQhprhouKaTkpq?utm_source=embed_player
https://open.spotify.com/artist/1mcTU81TzQhprhouKaTkpq?utm_source=embed_player
https://open.spotify.com/artist/1mcTU81TzQhprhouKaTkpq?utm_source=embed_player
https://open.spotify.com/artist/1mcTU81TzQhprhouKaTkpq?utm_source=embed_player
https://open.spotify.com/artist/1mcTU81TzQhprhouKaTkpq?utm_source=embed_player
https://open.spotify.com/artist/1mcTU81TzQhprhouKaTkpq?utm_source=embed_player
https://open.spotify.com/artist/1mcTU81TzQhprhouKaTkpq?utm_source=embed_player
https://open.spotify.com/artist/1mcTU81TzQhprhouKaTkpq?utm_source=embed_player
https://open.spotify.com/artist/1mcTU81TzQhprhouKaTkpq?utm_source=embed_player
https://open.spotify.com/artist/1mcTU81TzQhprhouKaTkpq?utm_source=embed_player
https://open.spotify.com/artist/1mcTU81TzQhprhouKaTkpq?utm_source=embed_player
https://open.spotify.com/artist/1mcTU81TzQhprhouKaTkpq?utm_source=embed_player
https://open.spotify.com/artist/1mcTU81TzQhprhouKaTkpq?utm_source=embed_player
https://open.spotify.com/artist/1mcTU81TzQhprhouKaTkpq?utm_source=embed_player
https://open.spotify.com/artist/1mcTU81TzQhprhouKaTkpq?utm_source=embed_player
https://open.spotify.com/artist/1mcTU81TzQhprhouKaTkpq?utm_source=embed_player
https://open.spotify.com/artist/1mcTU81TzQhprhouKaTkpq?utm_source=embed_player
https://open.spotify.com/artist/1mcTU81TzQhprhouKaTkpq?utm_source=embed_player

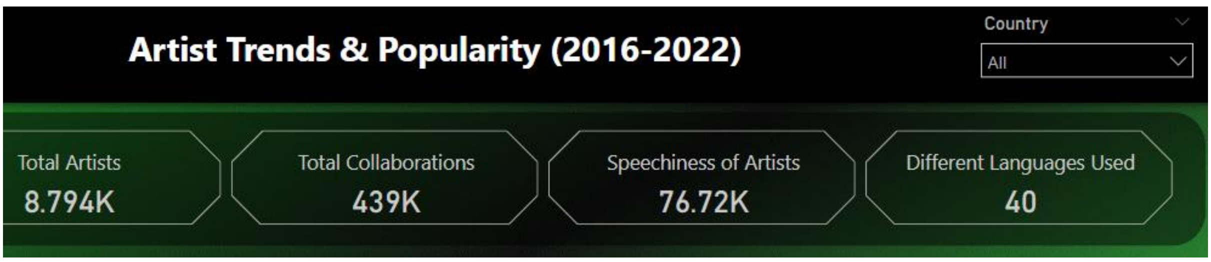
4. TOP ARTIST URL RETURNING MEASURE THROUGH BUTTON

```
Top_Aggregated_Artist_URL =
VAR GroupedData =
SUMMARIZE(
cleaned_data,
cleaned_data[artist_names],
cleaned_data[Converted Artist URL],
"TotalPeakRank", SUM(cleaned_data[peak_rank])
)

VAR MaxPeakRank =
MAXX(GroupedData, [TotalPeakRank])

VAR TopArtist =
FILTER(GroupedData, [TotalPeakRank] = MaxPeakRank)

RETURN
MAXX(TopArtist, cleaned_data[Converted Artist URL])
```



5. SLICER

COUNTRY



6. Average Streams per Track =
   DIVIDE(
   SUM(cleaned_data[streams]),
   DISTINCTCOUNT(cleaned_data[track_name]),

7. Most Streamed Genre =
   CALCULATE(
   MAXX(
   TOPN(
   1,
   SUMMARIZE(
   cleaned_data,
   cleaned_data[artist_genre],
   "Total Streams", SUM(cleaned_data[streams])
   ),
   [Total Streams], DESC
   ),
   cleaned_data[artist_genre]
)
)

8. Top Track Streams =
   MAXX(
   TOPN(1, 'cleaned_data', 'cleaned_data'[streams], DESC),
   'cleaned_data'[streams]
)

9. Average Danceability =
   AVERAGE(cleaned_data[danceability])

10. Average Energy Level =
    AVERAGE(cleaned_data[energy])

11. Average Tempo =
    AVERAGEX(
    FILTER(cleaned_data, cleaned_data[rank] <= 10),

```
cleaned_data[tempo]
)
```

12. Highest Valance Track =

```
MAXX(
TOPN(
1,
cleaned_data,
cleaned_data[valence], DESC
),
cleaned_data[track_name]
)
```

13. Longest Listening Time =

```
SUMX(
cleaned_data,
cleaned_data[streams] * cleaned_data[duration] / 3600
)
```

14. Most Recommended Artist =

```
MAXX(
TOPN(
1,
SUMMARIZE(
cleaned_data,
cleaned_data[artist_names],
"Total Collaborative Streams", SUM(cleaned_data[streams])
),
[Total Collaborative Streams], DESC
),
cleaned_data[artist_names]
)
```

15. Most Streamed Genre =

```
CALCULATE(
MAXX(
```

```
TOPN(
1,
SUMMARIZE(
cleaned_data,
cleaned_data[artist_genre],
"Total Streams", SUM(cleaned_data[streams])
),
[Total Streams], DESC
),
cleaned_data[artist_genre]
)
)
```

16. Top Engagement Track =

```
MAXX(
TOPN(
1,
SUMMARIZE(
cleaned_data,
cleaned_data[track_name],
"Engagement Score", SUM(cleaned_data[streams]) *
AVERAGE(cleaned_data[duration])
),
[Engagement Score], DESC
),
[track_name]
)
```

Calculated Columns

1. Calculated Column for conversion of links to https

```
Converted URL =
VAR TrackID = SUBSTITUTE(cleaned_data[uri], "spotify:track:", "")
RETURN
IF(
NOT(ISBLANK(TrackID)),
"https://open.spotify.com/track/" & TrackID & "?utm_source=embed_player",
```

```
BLANK()
)
```

Converted URL
https://open.spotify.com/track/2BEcap3xJtmiPaHKyfQi4e?utm_source=embed_player
https://open.spotify.com/track/2BEcap3xJtmiPaHKyfQi4e?utm_source=embed_player
https://open.spotify.com/track/2BEcap3xJtmiPaHKyfQi4e?utm_source=embed_player
https://open.spotify.com/track/2BEcap3xJtmiPaHKyfQi4e?utm_source=embed_player
https://open.spotify.com/track/2BEcap3xJtmiPaHKyfQi4e?utm_source=embed_player
https://open.spotify.com/track/2BEcap3xJtmiPaHKyfQi4e?utm_source=embed_player
https://open.spotify.com/track/2BEcap3xJtmiPaHKyfQi4e?utm_source=embed_player
https://open.spotify.com/track/2BEcap3xJtmiPaHKyfQi4e?utm_source=embed_player
https://open.spotify.com/track/2BEcap3xJtmiPaHKyfQi4e?utm_source=embed_player
https://open.spotify.com/track/2BEcap3xJtmiPaHKyfQi4e?utm_source=embed_player
https://open.spotify.com/track/2BEcap3xJtmiPaHKyfQi4e?utm_source=embed_player
https://open.spotify.com/track/2BEcap3xJtmiPaHKyfQi4e?utm_source=embed_player
https://open.spotify.com/track/2BEcap3xJtmiPaHKyfQi4e?utm_source=embed_player
https://open.spotify.com/track/2BEcap3xJtmiPaHKyfQi4e?utm_source=embed_player
https://open.spotify.com/track/2BEcap3xJtmiPaHKyfQi4e?utm_source=embed_player
https://open.spotify.com/track/2BEcap3xJtmiPaHKyfQi4e?utm_source=embed_player
https://open.spotify.com/track/2BEcap3xJtmiPaHKyfQi4e?utm_source=embed_player
https://open.spotify.com/track/2BEcap3xJtmiPaHKyfQi4e?utm_source=embed_player
https://open.spotify.com/track/2BEcap3xJtmiPaHKyfQi4e?utm_source=embed_player
https://open.spotify.com/track/2BEcap3xJtmiPaHKyfQi4e?utm_source=embed_player

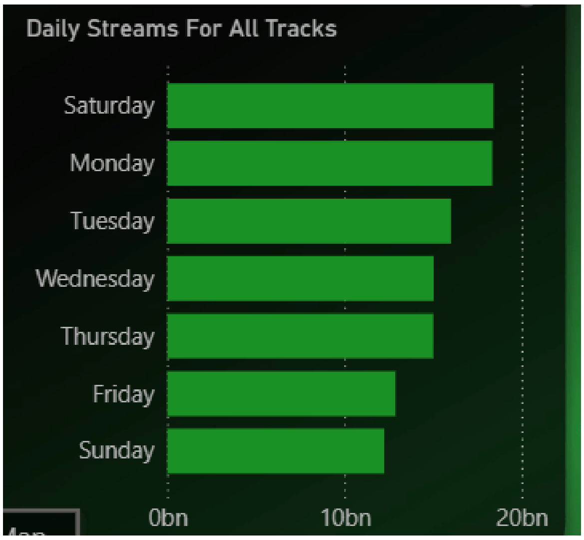
2. Created Calculated Column to create days

```
DayNames =
SWITCH(
    TRUE(),
    cleaned_data[week].[Day] = 1, "Sunday",
    cleaned_data[week].[Day] = 2, "Monday",
    cleaned_data[week].[Day] = 3, "Tuesday",
    cleaned_data[week].[Day] = 4, "Wednesday",
    cleaned_data[week].[Day] = 5, "Thursday",
    cleaned_data[week].[Day] = 6, "Friday",
    cleaned_data[week].[Day] = 7, "Saturday",
    "Unknown"  -- Default case for unexpected values
)
```

DayNames
Monday
Saturday
Wednesday
Tuesday
Monday
Sunday
Saturday
Thursday
Monday
Saturday
Tuesday
Friday
Tuesday
Thursday
Wednesday
Monday
Saturday

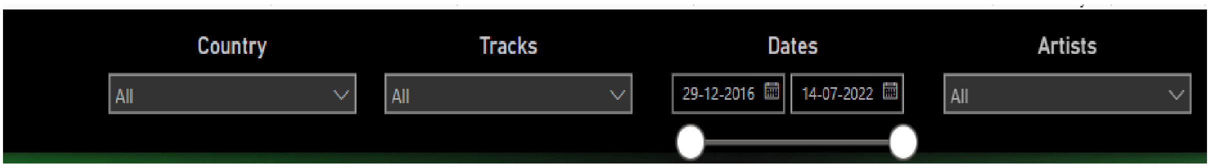
3. Created Measure to Select values from Day names Column

```
Dayname =
SWITCH(
    TRUE(),
    SELECTEDVALUE(cleaned_data[week].[Day]) = 1, "Sunday",
    SELECTEDVALUE(cleaned_data[week].[Day]) = 2, "Monday",
    SELECTEDVALUE(cleaned_data[week].[Day]) = 3, "Tuesday",
    SELECTEDVALUE(cleaned_data[week].[Day]) = 4, "Wednesday",
    SELECTEDVALUE(cleaned_data[week].[Day]) = 5, "Thursday",
    SELECTEDVALUE(cleaned_data[week].[Day]) = 6, "Friday",
    SELECTEDVALUE(cleaned_data[week].[Day]) = 7, "Saturday",
    BLANK()
)
```

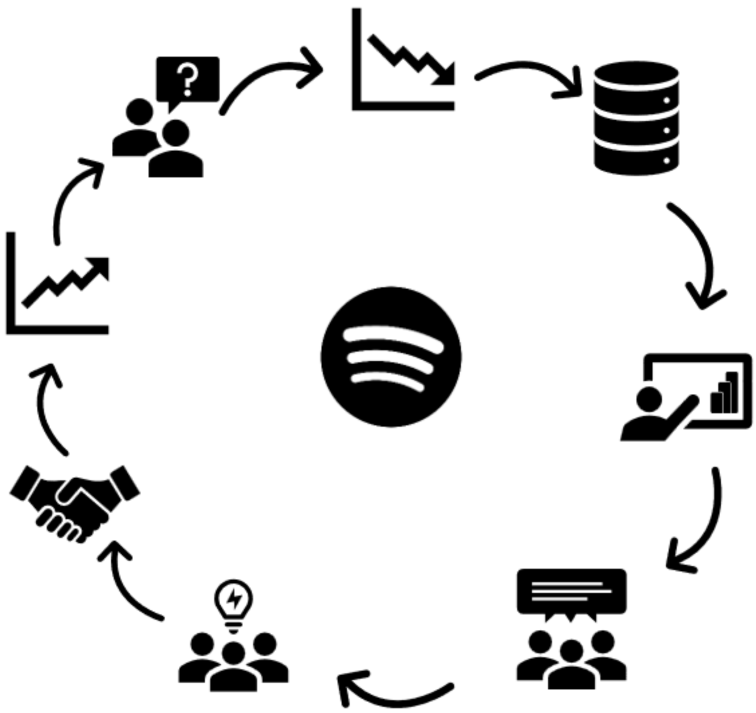


SLICERS

1) BY COUNTRY
2) Tracks
3) Dates (BETWEEN SLIDER)
4) BY ARTIST NAMES

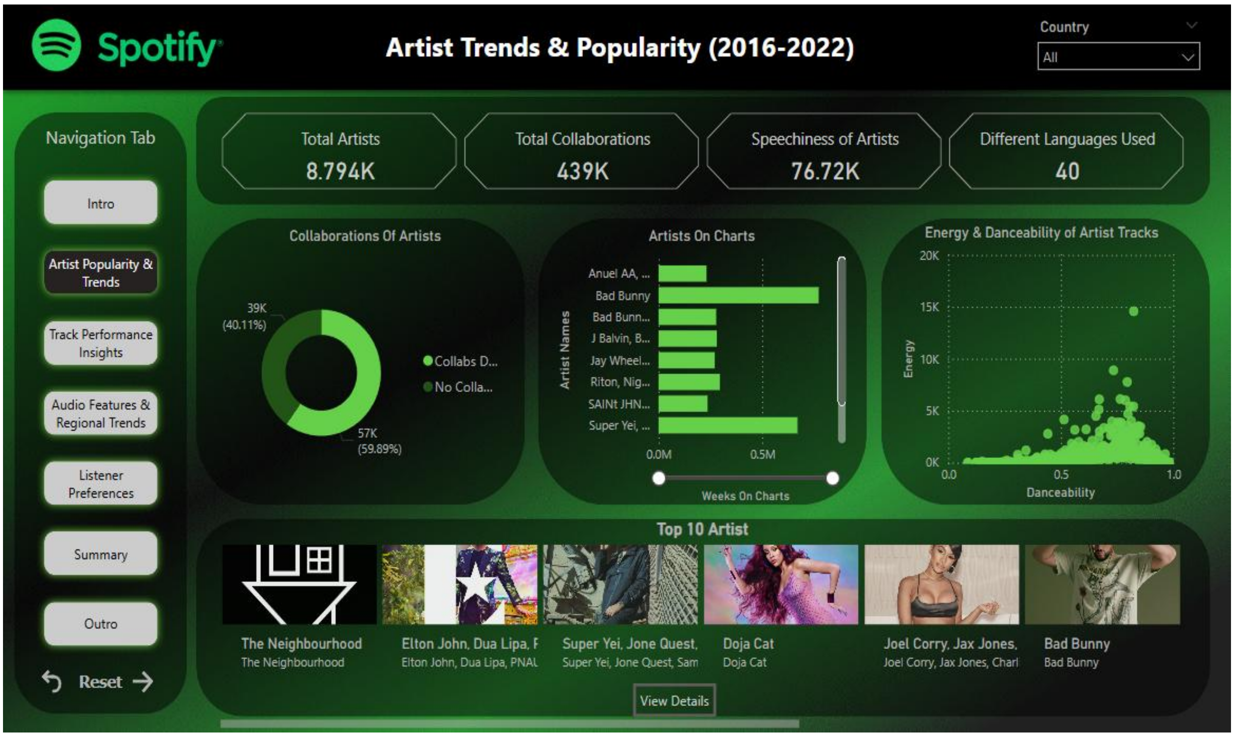


OUTPUT

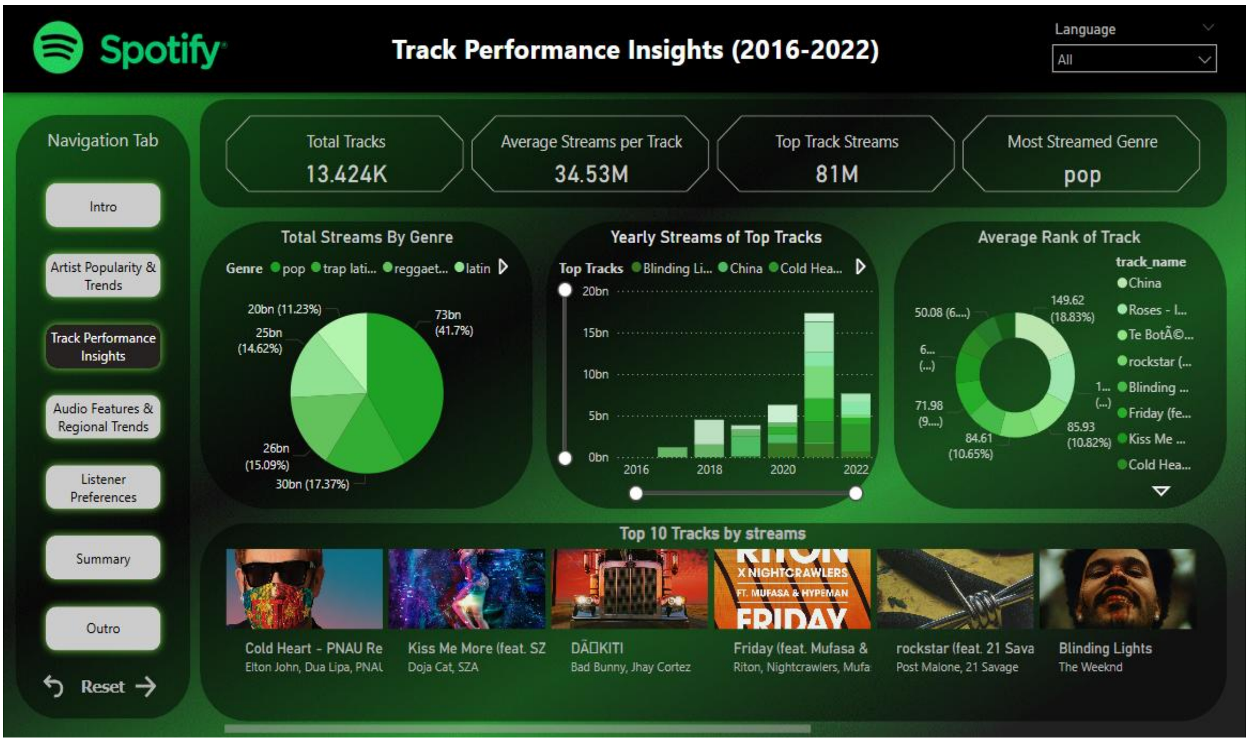


17.OUTPUT

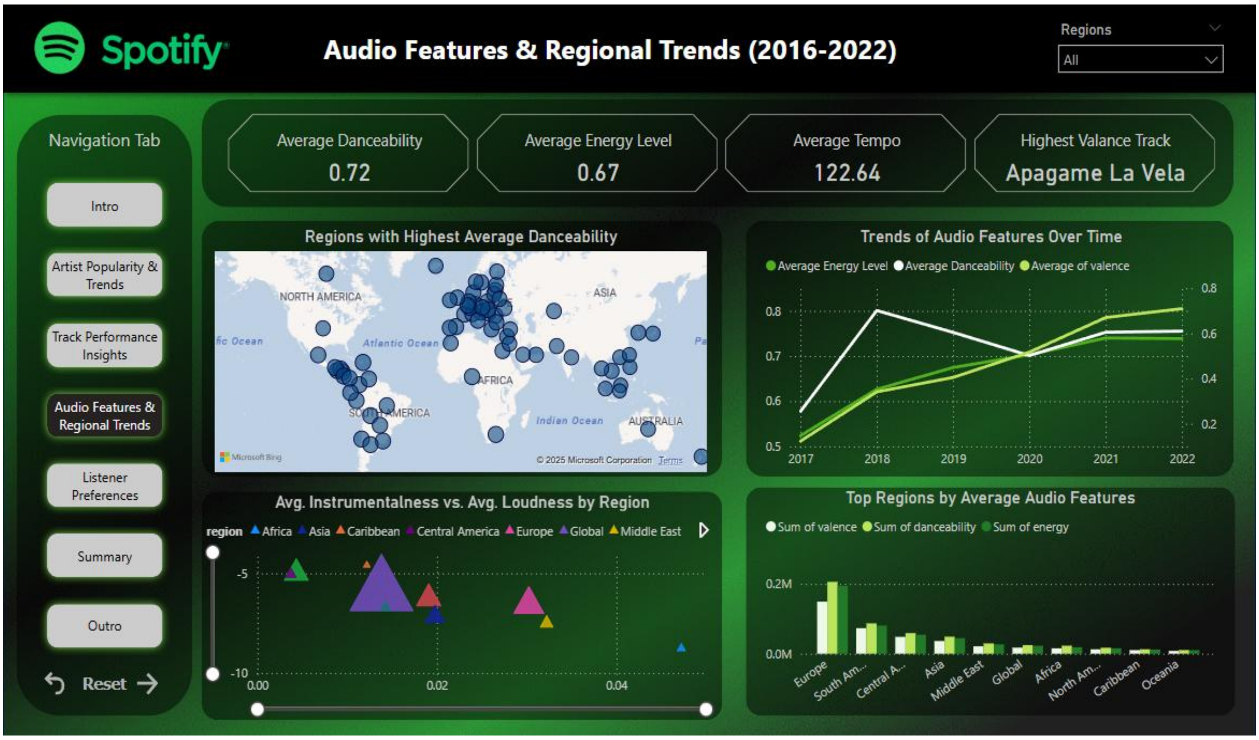
PAGE 1: ARTIST POPULARITY AND TRENDS



PAGE 2 : TRACK PERFORMANCE INSIGHTS



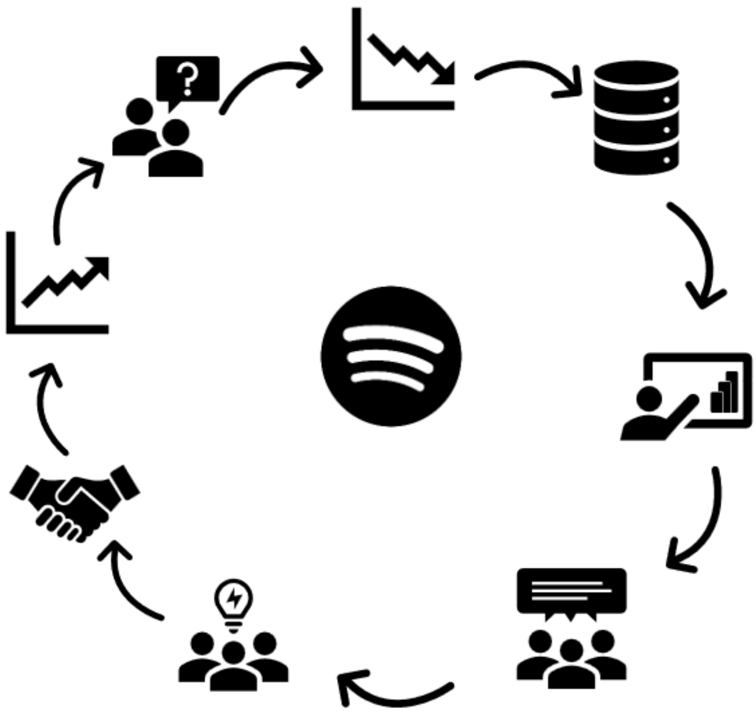
PAGE 3 : AUDIO FEATURES AND REGIONAL TRENDS



PAGE 4 : LISTENER PREFERRENCES

PAGE 5 : SUMMARY

CONCLUSION



18.CONCLUSION

The Power BI with Spotify Data Analysis Project has proven to be a significant and insightful initiative, offering a deeper understanding of various elements within the Spotify music ecosystem through data visualization and analytics. The project successfully leveraged Power BI, a powerful business intelligence tool, alongside Spotify's rich data sources, providing stakeholders with valuable insights into music trends, artist performance, track popularity, and user behavior.

By analyzing track popularity, the project has demonstrated the potential of data to assist curators, artists, and marketers in making informed decisions about which songs to promote or include in playlists. The project has shown how metrics such as Danceability, Energy, and Tempo can be used to identify characteristics of trending tracks. These metrics are essential for understanding what makes a track successful and how it resonates with different audiences. This information is particularly valuable for curators, as it enables them to curate playlists that align with current trends, ensuring they remain relevant and engaging for users.

Furthermore, the project's analysis of artist performance, including factors like popularity, follower count, and genre-specific data, has provided artists with insights into their audience's preferences. This data helps artists and record labels make more informed decisions about their marketing strategies, collaborations with other artists, or focus on specific regions or demographics. The ability to track performance metrics in real time allows artists to gauge the effectiveness of their promotional efforts and adjust strategies accordingly.

The analysis of user behavior, although optional, has opened the door for deeper insights into individual preferences and how different genres or artists perform among different user segments. This kind of data analysis can help improve Spotify's recommendation algorithms, allowing them to personalize music suggestions based on user behavior. Understanding user listening habits also enables Spotify to refine its algorithm, ensuring that users are exposed to the music they are most likely to enjoy, ultimately increasing user engagement and retention.

One of the most impactful aspects of the project was its focus on playlist optimization. By analyzing the relationships between tracks, artists, and playlists, the project revealed how playlists can drive track popularity and user engagement. Curators can now optimize their playlists based on factors such as track popularity, genre mix, and user behavior. This has a direct impact on user satisfaction, as well-curated playlists are more likely to resonate with listeners, leading to higher levels of user engagement.

In terms of data visualization, the use of Power BI proved to be invaluable. The interactive dashboards created allowed users to explore different aspects of the data in real time, providing them with dynamic filters, drill-through features, and slicers to gain deeper insights. These interactive features made the data more accessible and easier to understand, even for users with minimal technical expertise. The ability to interact with the data in this way empowered users to ask specific questions and generate personalized insights tailored to their needs. Power BI's custom reporting capabilities also allowed for the creation of tailored reports focused on key metrics for different stakeholders, such as marketers, artists, and curators. This made it easy to share insights across teams, ensuring that the right information was available at the right time for decision-making.

The ease of access to the reports via Power BI Service further enhanced the project's value. Stakeholders could access the reports on any device, whether desktop or mobile, ensuring that insights were always available for timely decision-making. This accessibility made the project more practical, as it catered to the needs of users on the go, allowing them to stay informed regardless of their location.

On the technical side, the integration of Spotify's public datasets and its API with Power BI was seamless. Power BI's ability to connect to external data sources, such as APIs and CSV files, made it straightforward to import data and analyze it in real time. The data transformation and modeling capabilities provided by Power BI, particularly through Power Query and DAX, ensured that raw data was cleaned, structured, and analyzed effectively. The successful integration of these technologies demonstrated how powerful tools like Power BI and Spotify's API could be combined to create a dynamic and real-time data analysis solution.

The project's technical feasibility was apparent in the ease with which the integration between Power BI, Spotify's data, and the necessary APIs was achieved. The minimal complexity of the implementation made it clear that the project could be executed with relatively low costs and required little technical effort. This is especially significant given that Power BI is an affordable and user-friendly tool, making it accessible to a wide range of users with varying levels of technical expertise.

From an operational perspective, the project had a tangible impact on decision-making.

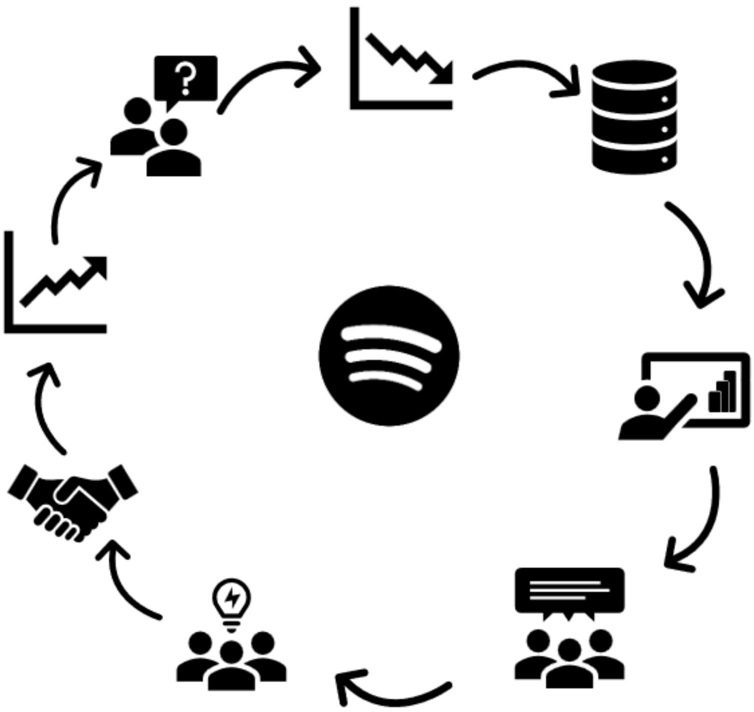
Stakeholders now have the ability to make data-driven decisions in a variety of contexts. Marketers can prioritize tracks and artists based on real-time performance data, curators can optimize playlists for maximum engagement, and artists can track their performance against industry benchmarks. The project empowered these stakeholders with insights that enable them to refine their strategies and improve their outcomes.

The business value of the project extends beyond Spotify itself. By analyzing trends and user behavior, businesses within the music industry can identify emerging genres, target specific audience segments, and adapt more quickly to shifting music preferences. This analysis allows Spotify to improve its service offering and better cater to its users' needs. Additionally, third-party marketing and promotional teams can leverage these insights to refine their campaigns and better engage with their target audiences.

However, the project did face some challenges. One significant issue was data quality, particularly when dealing with incomplete or inconsistent data. Raw data from APIs often requires preprocessing to ensure that it is accurate and usable. This challenge was overcome using Power BI’s data transformation features, which allowed for the effective cleansing and alignment of datasets. Another challenge arose from the limitations of the Spotify Web API, which imposes restrictions on the amount of data that can be pulled at once. This required careful handling of API rate limits and pagination. In some cases, static datasets were used as an alternative, though this limited the real-time aspect of the analysis. Despite these challenges, the project remained flexible, and the team was able to adapt to these limitations by adjusting the data pull strategy.

Scalability also posed a challenge, particularly when dealing with large datasets or when more real-time features were incorporated into the analysis. As the dataset grew or as more real-time features were integrated, it became apparent that more advanced data storage solutions, such as cloud storage or SQL-based databases, would

BIBLIOGRAPHY



19.BIBLIOGRAPHY

 Microsoft Power BI Documentation

- Microsoft. (n.d.). *Power BI Documentation*. Retrieved from https://docs.microsoft.com/en-us/power-bi/

 Spotify for Developers

- Spotify. (n.d.). *Spotify Web API Reference*. Retrieved from https://developer.spotify.com/documentation/web-api/

 DAX Guide

- SQLBI. (n.d.). *DAX Guide*. Retrieved from https://dax.guide/

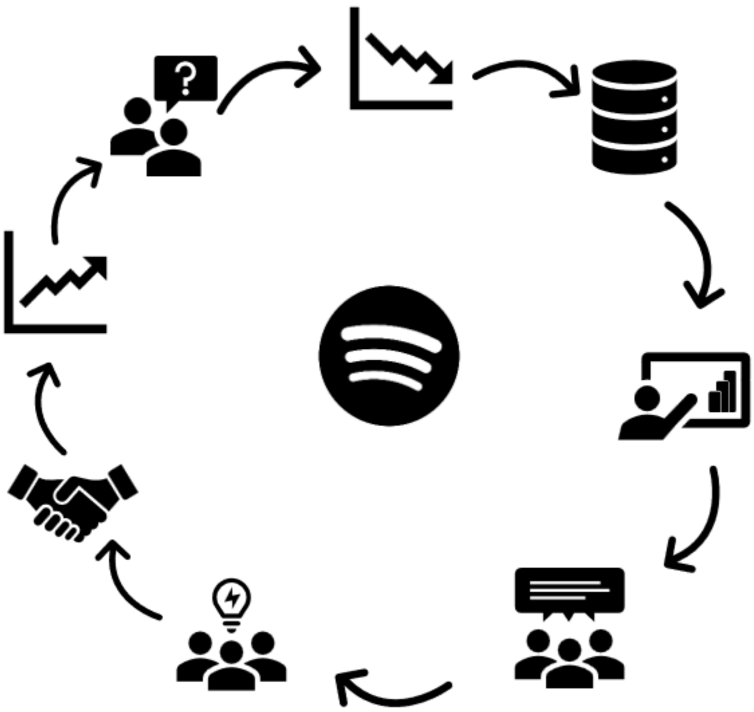
 Data Science for Business

- Provost, F., & Fawcett, T. (2013). *Data Science for Business: What You Need to Know about Data Mining and Data-Analytic Thinking*. O'Reilly Media.

 Power Query Documentation

- Microsoft. (n.d.). *Power Query Documentation*. Retrieved from https://docs.microsoft.com/en-us/power-query/

WEB REFERENCES



20.WEB REFERENCES

1. https://www.geeksforgeeks.org/power-bi-tutorial/
2. https://www.tutorialspoint.com/power_bi/index.htm
3. https://chatgpt.com/
4. https://gemini.google.com/?hl=en-IN