



Mastering Rapid Power BI Development with ChatGPT

Pedro Reis

Ricardo Santos

Impact of AI on Knowledge Workers

Study on AI at BCG with 758 Professionals



25%

Faster to complete tasks by
using AI

40%

Higher quality results using AI

Source: Navigating the Jagged Technological Frontier: Field Experimental Evidence of the Effects of AI on Knowledge Worker Productivity and Quality

[Harvard Business School Technology & Operations Mgt. Unit Working Paper No. 24-013](#)

Who can this help?

- SQL and BI Professionals need to tweek some M or DAX
- Power BI / Citizen Developers / SME Lacking coding expertise
- Professionals lacking UI / UX skills or in need of inspiration
- Occasional usage for productivity or direction pointer

- If you don't have co-copilot capacity
- If you don't have multi-modal capabilities



About the speakers



Pedro Reis

- Microsoft MVP, MCT
- Expert Leader Data & AI
- Leader of Fabric Power BI Portugal Group
- www.fabricbi.net
(training / consulting)



Ricardo Castro Santos

- Expert Leader - BI & Big Data
- @Data & AI Platform team
- Natixis in Portugal – Group BPCE
- Vice-President @ APBI – Portuguese Association for Business Intelligence



pedroreis@reispmonmicrosoft.com



<https://www.linkedin.com/in/pedrogoncalvesreis/>



rjcsantos@hotmail.com



<https://www.linkedin.com/in/rickcsantos>

Agenda and content slides

- 1) Generate a Data Model
- 2) Generate Artificial Data
- 3) Generate Power Query code
- 4) Generate DAX Measures
- 5) Enhance Report Theme
- 6) Publication & Administration w/ Cmdlets
- 7) Key Takeaways / The Future



Test OK



**Test OK
with iterations or
limitations**

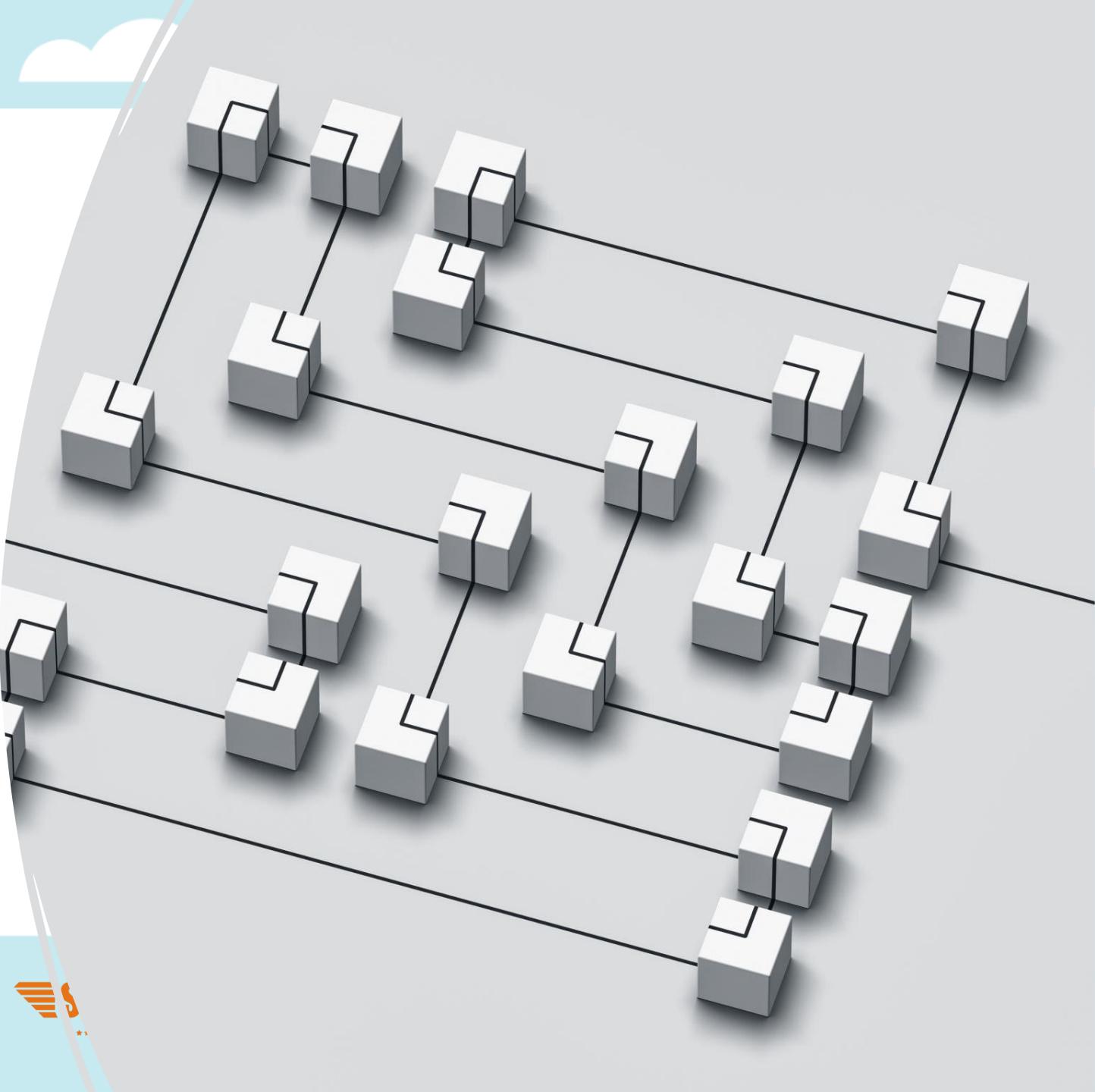


Test KO



1) Generate a Data Model

Generate a Data Model for a specific industry



Generate a Data Model



How can I help you today?

Suggest some names
for my cafe-by-day, bar-by-night business

Plan a trip
to experience Seoul like a local

Show me a code snippet
of a website's sticky header

Brainstorm names
for an orange cat we're adopting from the shelter

Message ChatGPT...



ChatGPT can make mistakes. Consider checking important information.



Generate a Data Model



Prompt:

I manage a {data services} firm.

- *Generate a data model for this company*
- *Prefix the Fact tables with FCT_ and the Dimension tables with DIM_*
- *List the relationships between the tables in a matrix format*

Table Format:

Table Type	Table Name	Related Tables
Factual	Projects	Employees, Clients, Services, Locations, Date
Factual	Invoices	Projects, Clients, Date
Factual	TimeEntries	Projects, Employees, Date
Dimension	Employees	Projects, TimeEntries
Dimension	Clients	Projects, Invoices
Dimension	Services	Projects
Dimension	Locations	Projects
Dimension	Date	Projects, Invoices, TimeEntries

* "X" will indicate a direct relationship or foreign key reference.

* "-" will indicate no direct relationship.

	DIM_Consultant	DIM_Client	DIM_Project	DIM_Skill	FCT_Invoice	FCT_Timesheet
DIM_Consultant	-	-	-	X	-	X
DIM_Client	-	-	X	-	X	-
DIM_Project	-	X	-	-	X	X
DIM_Skill	X	-	-	-	-	-
DIM_Time	-	-	-	-	X	X
FCT_Invoice	-	X	X	-	-	-
FCT_Timesheet	X	-	X	-	-	-





2) Generate Artificial Data

Generate CSV Files

Generate Py Script which generates CSV Files

Generate T-SQL Script which generates and populates Database



Generate Data - Produce CSV



entity (Dimension table).

In this matrix:

- **FCT_Sales** has relationships with all four dimensions: It records sales transactions, which involve client details, the services sold, the employees who made the sale, and the time when the sale happened.
- **FCT_ServiceUsage** also relates to all dimensions except for DIM_Services: It tracks how clients use services over time, which employees are involved in providing these services, and when these services are used.
- **FCT_ClientInteractions** is related to DIM_Clients, DIM_Employees, and DIM_Time: It logs interactions with clients, the employees involved in these interactions, and the time of these interactions.

This BUS matrix is a simplified representation and can be expanded or modified based on the actual processes and data flows within your firm. It's essential to tailor this matrix to reflect the unique interactions between your business processes and data entities for a more accurate and useful data model.



Message ChatGPT...

I



ChatGPT can make mistakes. Consider checking important information.



Generate Artificial Data - Python Script

Generate sample CSV files for each of these tables. Assume the following:

- Dates range from 2019 to end of 2023
- There was growth in 2019, a poor year in 2021, and a big recovery in 2022.
- 50 employees with random realistic American names
- 30 Customers with fictitious corporate like names



ChatGPT

The sample CSV files have been generated for your data model:

- **Employees Dimension Table:** DIM_Employees.csv
- **Customers Dimension Table:** DIM_Customers.csv
- **Sales Fact Table:** FCT_Sales.csv

These files contain sample data based on the criteria you provided. [→]

◀ 2 / 2 ▶ 📁 ⬆️ ⬇️ 🔍

Was this response better
or worse?

⬆️
Better

⬇️
Worse

≡
Same

×

Is this conversation helpful so far? ⬆️ ⬇️ ×



Message ChatGPT...



ChatGPT can make mistakes. Consider checking important information.

Generate Artificial Data - Python Script

```
    sales_records.append(record)
df_sales = pd.DataFrame(sales_records)
df_sales.to_csv(output_directory + 'FCT_Sales.csv', index=False)

# Generate all data
create_dimension_data()
generate_sales_data()

print('CSV files have been generated and saved to the specified directo
```

To execute the script:

1. Copy the script into a text file and save it with a ` `.py` extension, for example, ` `generate_csv.py` `.
2. Open a command prompt or terminal window.
3. Navigate to the directory where you saved the ` `.py` file.
4. Run the script by typing ` `python generate_csv.py` ` and pressing Enter.

Remember that this script uses dummy data for demonstration purposes. You would need to populate the CSV files with actual  from your business processes.



 Message ChatGPT...



ChatGPT can make mistakes. Consider checking important information.



Generate Artificial Data - Python Script - Results

A
1 CustomerID,CustomerName
2 1,Jennifer Robert Technologies
3 2,John Elizabeth Technologies
4 3,Elizabeth Robert Co
5 4,Mary Robert LLC
6 5,Linda Robert Inc
7 6,Robert Elizabeth Technologies
8 7,Michael Robert Inc
9 8,Linda Jennifer Co
10 9,Mary Michael LLC
11 10,Robert Elizabeth LLC
12 11,Patricia Elizabeth Systems
13 12,Linda Robert Systems
14 13,Patricia Robert Co
15 14,James Linda Systems
16 15,John Michael Co

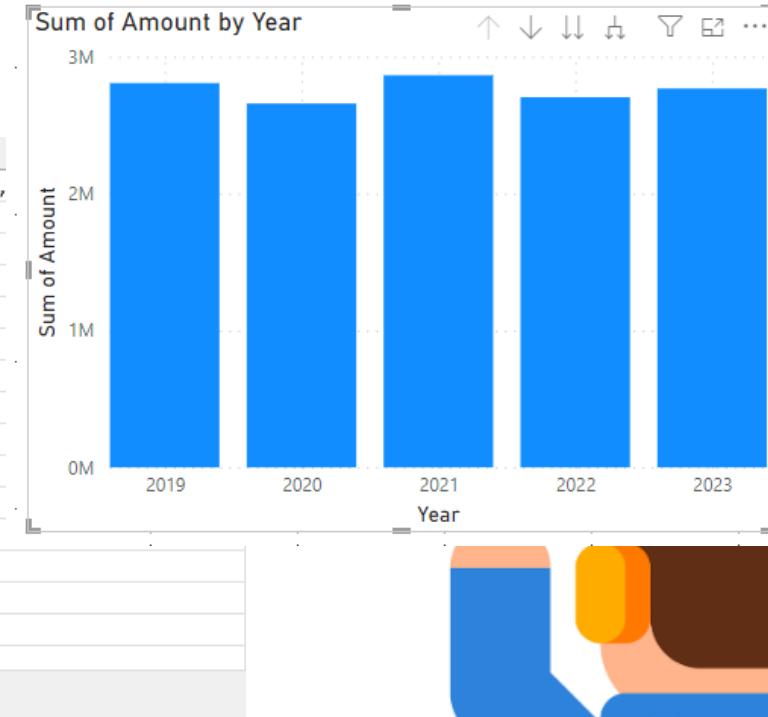
DIM_Customers

A
1 EmployeeID,EmployeeName
2 1,Mary Brown
3 2,Linda Garcia
4 3,Jennifer Wilson
5 4,Mary Miller
6 5,Robert Jones
7 6,Mary Wilson
8 7,Michael Smith
9 8,Linda Garcia
10 9,Jennifer Rodriguez
11 10,Jennifer Rodriguez
12 11,James Smith
13 12,Robert Williams
14 13,Michael Jones
15 14,Patricia Williams
16 15,Linda Smith

DIM_Employees

A
1 DateKey,Amount,CustomerID,EmployeeID,
2 2019-01-01,3172.1981533106064,29,12,8,1
3 2019-01-01,1717.2053373594313,14,25,7,1
4 2019-01-01,4616.179799681728,17,27,5,2
5 2019-01-02,3506.566111399132,20,24,5,4
6 2019-01-02,1042.6612732185104,1,44,1,3
7 2019-01-02,2280.75555440332,10,41,9,2
8 2019-01-02,4413.48576106266,6,49,7,2
9 2019-01-02,3847.43395063937,30,44,8,1
10 2019-01-03,1906.475193067163,1,35,8,5
11 2019-01-03,651.5581633629162,5,25,8,3
12 2019-01-04,1453.6289421357617,16,1,5,3
13 2019-01-04,4377.463815404791,5,3,3,4
14 2019-01-05,604.8937728054532,1,43,3,1
15 2019-01-05,4611.91458213251,1,28,1,3
16 2019-01-05,3293.839512315117,9,24,2,4

FCT_Sales



Generate Artificial Data - T-SQL Script

```
'EmployeeID': random.choice(range(1, number_of_employee
    }
    interaction_records.append(record)
df_client_interactions = pd.DataFrame(interaction_records)

# Save the dataframes to CSV files
df_sales.to_csv(output_directory + 'FCT_Sales.csv', index=False)
df_service_usage.to_csv(output_directory + 'FCT_ServiceUsage.csv',
df_client_interactions.to_csv(output_directory + 'FCT_ClientInterac

# Generate all data
create_dimension_data()
generate_fact_data()

print('All CSV files have been generated and saved to the specified dir
```

This script now includes all the Dimension and Fact tables we initially discussed. You can add more tables as per your model or adjust the script to better fit your data structure and content. Remember to run this script in your local environment.



Message ChatGPT



ChatGPT can make mistakes. Consider checking important information.

Generate Artificial Data - T-SQL Script

```
INSERT Sample data into DIM_Location
-- Repeat the following block for the number of locations you have
INSERT INTO dbo.DIM_Location (LocationID, LocationName)
VALUES (1, 'Location 1'),
       (2, 'Location 2'),
       -- ...
       (5, 'Location 5');
GO

-- Insert sample data into FCT_Sales, FCT_ServiceUsage, FCT_ClientInter
-- Use a similar approach as DIM_Time to generate and insert data for t

-- Don't forget to set the identity_insert to ON when inserting explici
```

This script creates the tables and provides examples of how you might insert data. In practice, you would likely use a more sophisticated method for generating and inserting data, especially for the Fact tables, which could have a large number of records that correlate with actual business activities. The actual data insertion would depend on your business logic and the source of your data.



Message ChatGPT...



ChatGPT can make mistakes. Consider checking important information.

Generate Artificial Data - T-SQL Script

The screenshot shows the Microsoft SQL Server Management Studio (SSMS) interface. On the left, the Object Explorer window displays the database structure for 'ConsultingDemoDB'. In the center-right, a query editor window titled 'SQLQuery2.sql' contains the following T-SQL script:

```
-- Create FCT_ServiceUsage table
CREATE TABLE dbo.FCT_ServiceUsage (
    UsageID INT IDENTITY(1,1) PRIMARY KEY,
    DateKey DATE,
    UsageTime INT, -- Assuming usage time is an integer value like minutes
    CustomerID INT,
    EmployeeID INT,
    ServiceID INT,
    FOREIGN KEY (DateKey) REFERENCES dbo.DIM_Time(DateKey),
    FOREIGN KEY (CustomerID) REFERENCES dbo.DIM_Customers(CustomerID),
    FOREIGN KEY (EmployeeID) REFERENCES dbo.DIM_Employees(EmployeeID),
    FOREIGN KEY (ServiceID) REFERENCES dbo.DIM_Services(ServiceID)
);
GO
```

The 'Messages' pane at the bottom of the query editor shows the command completed successfully with a completion time of 2024-01-05T11:44:54.5481158+00:00.

Generate Artificial Data - T-SQL Script

The screenshot shows the Microsoft SQL Server Management Studio (SSMS) interface. The title bar indicates the current session is "SQLQuery3.sql - fabricbisqldbserver.database.windows.net.ConsultingDemoDB (pedroreis@reispn.onmicrosoft.com (85))". The Object Explorer sidebar shows the database structure, including the "ConsultingDemoDB" database which contains tables like "dbo.DIM_Customers", "dbo.DIM_Employees", and "dbo.DIM_Location". The main query editor window displays a T-SQL script for generating artificial data:

```
-- Insert data into DIM_Location
SET IDENTITY_INSERT dbo.DIM_Location ON;
SET @Counter = 1;
WHILE @Counter <= 5 -- Assuming you want 5 locations
BEGIN
    INSERT INTO dbo.DIM_Location (LocationID, LocationName)
    VALUES (@Counter, 'Location ' + CAST(@Counter AS VARCHAR(50)));
    SET @Counter = @Counter + 1;
END;
SET IDENTITY_INSERT dbo.DIM_Location OFF;
GO

-- Continue with the rest of the insert statements without needing IDENTITY_INSERT
-- For tables where you're not inserting explicit values for the identity columns
```

The "Messages" pane at the bottom of the editor shows numerous error messages related to the `IDENTITY_INSERT` statement:

- Msg 8106, Level 16, State 1, Line 4
Table 'dbo.DIM_Employees' does not have the identity property. Cannot perform SET operation.
- Msg 137, Level 15, State 1, Line 17
Must declare the scalar variable "@Counter".
- Msg 137, Level 15, State 2, Line 18
Must declare the scalar variable "@Counter".
- Msg 137, Level 15, State 2, Line 21
Must declare the scalar variable "@Counter".
- Msg 137, Level 15, State 2, Line 22
Must declare the scalar variable "@Counter".
- Msg 137, Level 15, State 1, Line 28
Must declare the scalar variable "@Counter".
- Msg 137, Level 15, State 2, Line 30
Must declare the scalar variable "@Counter".
- Msg 137, Level 15, State 2, Line 33
Must declare the scalar variable "@Counter".
- Msg 137, Level 15, State 2, Line 34
Must declare the scalar variable "@Counter".

The status bar at the bottom of the SSMS window shows "Ready", "In 53", "Col 1", "Ch 4", and "INS".

Generate Artificial Data - T-SQL Script

The screenshot shows the Microsoft SQL Server Management Studio (SSMS) interface. The title bar indicates the file is named `SQLQuery3.sql` and is connected to the database `ConsultingDemoDB`. The Object Explorer on the left shows the database structure, including tables like `dbo.DIM_Client`, `dbo.FCT_Invoice`, and `dbo.FCT_Sales`. The main pane displays a T-SQL script:

```
***** Script for SelectTopNRows command from SSMS *****/
SELECT TOP (1000) [InvoiceID]
    ,[ClientID]
    ,[ProjectID]
    ,[Date]
    ,[Amount]
    ,[Status]
FROM [dbo].[FCT_Invoice]
```

The Results tab shows the output of the query, displaying 17 rows of generated invoice data:

	InvoiceID	ClientID	ProjectID	Date	Amount	Status
1	1	7	91	2021-11-03	4150.00	Paid
2	2	4	23	2021-03-07	1257.00	Paid
3	3	28	123	2022-12-04	2703.00	Paid
4	4	16	128	2021-04-23	4990.00	Unpaid
5	5	23	6	2019-06-29	3053.00	Paid
6	6	7	162	2022-02-10	4228.00	Paid
7	7	5	146	2020-02-10	4628.00	Paid
8	8	24	106	2021-08-03	4838.00	Unpaid
9	9	6	41	2020-08-27	3495.00	Paid
10	10	4	4	2019-07-03	3866.00	Paid
11	11	22	142	2021-06-11	2073.00	Unpaid
12	12	13	164	2019-11-28	728.00	Unpaid
13	13	7	36	2022-12-12	2347.00	Paid
14	14	27	50	2020-07-07	2579.00	Unpaid
15	15	2	67	2019-05-14	2743.00	Paid
16	16	28	123	2019-03-16	1347.00	Paid
17	17	24	73	2019-06-20	2166.00	Paid

A message at the bottom of the results pane states "Query executed successfully." and "fabricbisqldbserver.database.windows.net".



2) Generate Artificial Data



- Generate CSV Files
- Generate Py Script which generates CSV Files
- Generate T-SQL Script which generates and populates Database

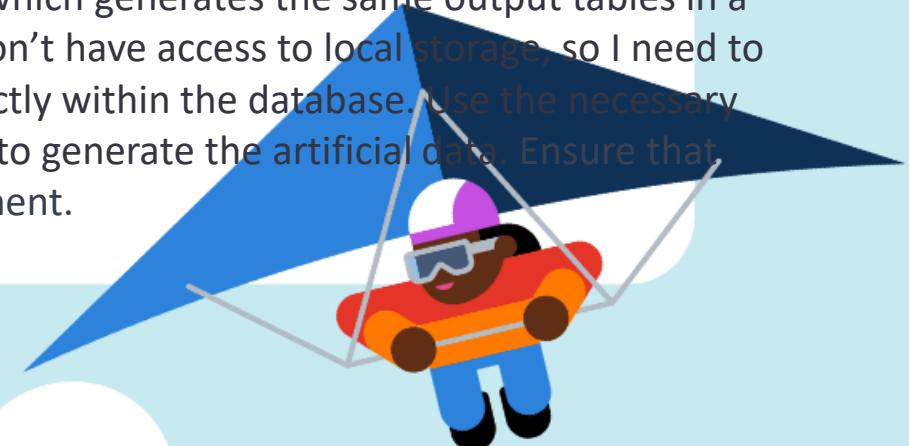
1. Text files (.txt)
2. CSV files (.csv)
3. JSON files (.json)
4. Python scripts (.py)
5. LaTeX documents (.tex)
6. Markdown files (.md)
7. XML files (.xml)

Generate sample **CSV files** for each of these tables. Assume the following:

- Dates range from 2019 to end of 2023
- There was growth in 2019, a poor year in 2021, and a big recovery in 2022.
- 50 employees with random realistic American names
- 30 Customers with fictitious corporate like names

Generate a **Python** script which generates the CSV files for every table in the model and save them on c:\Consultancy Project\

Generate a **T-SQL** script which generates the same output tables in a SQL Server Database. I don't have access to local storage, so I need to generate the output directly within the database. Use the necessary randomization functions to generate the artificial data. Ensure that primary keys auto-increment.



2) Generate Artificial Data



If the output is not complete,
request completion

AG i need the full t-sql script for all tables

Certainly. Here's the full T-SQL script for all the tables:

```
sql

-- Create DIM_Skill table
CREATE TABLE DIM_Skill (
    SkillID INT PRIMARY KEY,
    SkillName NVARCHAR(255),
    SkillDescription NVARCHAR(1000)
);

INSERT INTO DIM_Skill (SkillID, SkillName, SkillDescription)
VALUES
```

Create a SQL Database

[Azure SQL Database free offer - Azure SQL Database | Microsoft Learn](#)

New Database

Select a page

General Options Configure SLO

Script Help

Database name: ConsultingDemoDB

Object Explorer

Connect Databases System Databases ConsultancyDemoDB Database Diagrams Tables Views External Resources Synonyms Programmability Query Store Extended Events Storage Security ConsultingServices FabricBISQLDB Security Integration Services Catalogs

SQLQuery1.sql - fa...microsoft.com (81)*

```
-- Create DIM_Skill table
CREATE TABLE DIM_Skill (
    SkillID INT IDENTITY(1,1) PRIMARY KEY,
    SkillName NVARCHAR(255),
    SkillDescription NVARCHAR(1000)
);

-- Insert into DIM_Skill (SkillName, SkillDescription)
VALUES
('Data Analysis', NEWID()),
('Machine Learning', NEWID()),
('Data Visualization', NEWID()),
('Database Management', NEWID()),
('ETL', NEWID()),
('Cloud Computing', NEWID()),
('Big Data', NEWID());

-- Create DIM_Constant table
CREATE TABLE DIM_Constant (
    ConsultantID INT IDENTITY(1,1) PRIMARY KEY,
    FirstName NVARCHAR(255),
    LastName NVARCHAR(255),
    Email NVARCHAR(255),
    Phone NVARCHAR(20),
    HireDate DATE,
```

99 %

Messages

(7 rows affected)

(1 row affected)



3) Generate Power Query code

- Generate Calendar
- Create Custom Columns
- Create Custom Functions
- Translate Calendar



Generate Dim_Date



You

Generate a DIM_DATE table for Power Query.

This table should have Year, Month (Number Short Month, Long Month), Day Columns.

The range of dates in this table should be the 1st of January 2023, until the 31st of December of the current year.



ChatGPT

The DIM_DATE table for Power Query, spanning from the 1st of January 2023 until the 31st of December of the current year, has been generated with the following columns:

- Year
- Month (Number)
- Short Month (e.g., Jan)
- Long Month (e.g., January)
- Day

Here's how the first few entries look:

Year	Month	Short Month	Long Month	Day
2023	1	Jan	January	1
2023	1	Jan	January	2
2023	1	Jan	January	3
2023	1	Jan	January	4
2023	1	Jan	January	5 [→]

Generate Dim_Date

AG You

Generate a DIM_DATE table for Power Query.

This table should have Year, Month (Number, Short Month, Long Month), Day Columns. The range of dates in this table should be the 1st of January 2023, until the 31st of December of the current year.

Return the M code in a format that I can directly paste into a new blank query in Power BI.

ChatGPT

To create a DIM_DATE table in Power Query for use in Power BI with the specified columns and date range, you can use the following M code. This script will generate a table starting from January 1st, 2023, to December 31st of the current year, including columns for Year, Month Number, Short Month Name, Long Month Name, and Day.

Please copy and paste the following code into a new blank query in Power BI:

```
m
let
    StartDate = #date(2023, 1, 1),
    EndDate = #date(Year(DateTime.LocalNow()), 12, 31),
    DurationDays = Duration.Days(EndDate - StartDate) + 1,
    DateList = List.Dates(StartDate, DurationDays, #duration(1, 0, 0, 0)),
    TableFromList = Table.FromList(DateList, Splitter.SplitByNothing(), null, null, ExtraValues.Error),
    RenamedColumns = Table.RenameColumns(TableFromList,{{"Column1", "Date"}}),
    InsertedYear = Table.AddColumn(RenamedColumns, "Year", each Date.Year([Date])),
    InsertedMonthNumber = Table.AddColumn(InsertedYear, "Month Number", each Date.Month([Date])),
    InsertedShortMonthName = Table.AddColumn(InsertedMonthNumber, "Short Month", each Date.Text([Date], "ShortMonthName")),
    InsertedLongMonthName = Table.AddColumn(InsertedShortMonthName, "Long Month", each Date.Text([Date], "LongMonthName")),
    InsertedDay = Table.AddColumn(InsertedLongMonthName, "Day", each Date.Day([Date])),
    RemovedColumns = Table.RemoveColumns(InsertedDay, {"Date"})
in
RemovedColumns
```

The screenshot shows the Power Query Editor interface. The 'Advanced editor' tab is selected. In the main pane, there is an error message: '[Expression.Error] The import Year matches no exports. Did you miss a module reference?'. Below the error message, it says 'isRecoverable: True' and 'isExpected: True'. To the right, the 'Query settings' pane shows the 'Name' field set to 'Query' and the 'Applied steps' pane lists several steps including 'RemovedColumns' at the bottom.

Generate Dim_Date

AG

You

[Expression.Error] The import Year matches no exports. Did you miss a module reference?

DetailsIsRecoverable: True
isExpected: True

ChatGPT

It seems there was an error with the use of `Year(DateTime.LocalNow())` in the provided M code, likely due to a syntax issue or a misunderstanding of how to correctly use the `Year` function. Let's correct that part of the script to ensure it properly calculates the end date using the current year. The correct approach is to use `DateTime.LocalNow()` to get the current date and time, and then extract the year part using the `Date.Year` function. Here's the revised version of the M code:

```
m
let
    StartDate = #date(2023, 1, 1),
    CurrentYear = Date.Year(DateTime.LocalNow()),
    EndDate = #date(CurrentYear, 12, 31),
    DurationDays = Duration.Days(EndDate - StartDate) + 1,
    DateList = List.Dates(StartDate, DurationDays, #duration(1, 0, 0, 0)),
    TableFromList = Table.FromList(DateList, Splitter.SplitByNothing(), null, null, false),
    RenamedColumns = Table.RenameColumns(TableFromList, [{"Column1", "Date"}]),
    InsertedYear = Table.AddColumn(RenamedColumns, "Year", each Date.Year([Date])),
    InsertedMonthNumber = Table.AddColumn(InsertedYear, "Month Number", each Date.MonthNumber([Date])),
    InsertedShortMonthName = Table.AddColumn(InsertedMonthNumber, "Short Month", each Date.Name([Month Number], "Short")),
    InsertedLongMonthName = Table.AddColumn(InsertedShortMonthName, "Long Month", each Date.Name([Month Number], "Long")),
    InsertedDay = Table.AddColumn(InsertedLongMonthName, "Day", each Date.Day([Date])),
    RemovedColumns = Table.RemoveColumns(InsertedDay, {"Date"})
in
    RemovedColumns
```

This corrected code snippet specifically addresses the issue with the `Year` function by first calculating the current year through `Date.Year(DateTime.LocalNow())`, then using this value to define the `EndDate`. This should resolve the import error and allow the script to run successfully in Power BI.

The screenshot shows the Power Query Editor interface. The ribbon tabs include Home, Transform, Add column, View, and Help. The Home tab is selected. The Queries [1] pane shows a single query named "Table.RemoveColumns(InsertedDay, {"Date"})". The main area displays a table with five columns: Year, Month Number, Short Month, Long Month, and Day. The data consists of 20 rows from January 1, 2023, to January 20, 2023. The "RemovedColumns" step in the Applied steps pane is highlighted with a red border, indicating it is the last step in the query.

	Year	Month Number	Short Month	Long Month	Day
1	2023	1	jan	janeiro	1
2	2023	1	jan	janeiro	2
3	2023	1	jan	janeiro	3
4	2023	1	jan	janeiro	4
5	2023	1	jan	janeiro	5
6	2023	1	jan	janeiro	6
7	2023	1	jan	janeiro	7
8	2023	1	jan	janeiro	8
9	2023	1	jan	janeiro	9
10	2023	1	jan	janeiro	10
11	2023	1	jan	janeiro	11
12	2023	1	jan	janeiro	12
13	2023	1	jan	janeiro	13
14	2023	1	jan	janeiro	14
15	2023	1	jan	janeiro	15
16	2023	1	jan	janeiro	16
17	2023	1	jan	janeiro	17
18	2023	1	jan	janeiro	18
19	2023	1	jan	janeiro	19
20	2023	1	jan	janeiro	20



Generate Dim_Date (Improved Prompt)

You

Generate a DIM_DATE table for Power Query.

This table should have Year, Month (Number Short Month, Long Month), Day Columns.
The range of dates in this table should be the 1st of January 2023, until the 31st of December of the current year.

Return the M code in a format that I can directly paste into a new blank query in Power BI, ensuring syntax accuracy and proper function calls to avoid common errors such as misreferencing modules or functions.

ChatGPT

To create a 'DIM_DATE' table in Power Query for Power BI that includes columns for Year, Month (Number, Short Month, Long Month), and Day, covering the date range from January 1st, 2023, to December 31st of the current year, you can use the M code snippet provided below. This M code creates a list of dates for the specified range and then expands that list into a table. For each date, it extracts the year, the numeric month, the abbreviated month name, the full month name, and the day.

```
m
let
    StartDate = #date(2023, 1, 1),
    EndDate = #date(Date.Year(DateTime.LocalNow()), 12, 31),
    DateList = List.Dates(StartDate, Duration.Days(EndDate - StartDate) + 1, #duration(1,0,0,0)),
    ConvertToListOfRecords = List.Transform(DateList, (Date) =>
        Record.FromList(
            {Date.Year(Date), Date.Month(Date), Text.Start(Date.ToText(Date, "MMM"), 3), Date.ToText(Date, "MMMM")}, {"Year", "MonthNumber", "ShortMonth", "LongMonth", "Day"})
    ),
    Convert.ToTable = Table.FromRecords(ConvertToListOfRecords)
in
    Convert.ToTable
```

Here's a breakdown of how this works:

The screenshot shows the Power Query Editor interface. On the left, the 'Queries [1]' pane shows a single query named 'Query'. The main area displays a table with seven columns: Year, MonthNumber, ShortMonth, LongMonth, and Day. The data consists of 7 rows, each representing the 1st of January 2023. The 'Advanced editor' pane at the bottom shows the M code used to generate this table.

ABC Year	ABC MonthNumber	ABC ShortMonth	ABC LongMonth	ABC Day
1 2023	1 jan	jan	janeiro	1
2 2023	1 jan	jan	janeiro	2
3 2023	1 jan	jan	janeiro	3
4 2023	1 jan	jan	janeiro	4
5 2023	1 jan	jan	janeiro	5
6 2023	1 jan	jan	janeiro	6
7 2023	1 jan	jan	janeiro	7

```
1 let
2     StartDate = #date(2023, 1, 1),
3     EndDate = #date(Date.Year(DateTime.LocalNow()), 12, 31),
4     DateList = List.Dates(StartDate, Duration.Days(EndDate - StartDate) + 1, #duration(1,0,0,0)),
5     ConvertToListOfRecords = List.Transform(DateList, (Date) =>
6         Record.FromList(
7             {Date.Year(Date), Date.Month(Date), Text.Start(Date.ToText(Date, "MMM"), 3), Date.ToText(Date, "MMMM")}, {"Year", "MonthNumber", "ShortMonth", "LongMonth", "Day"})
8     ),
9     Convert.ToTable = Table.FromRecords(ConvertToListOfRecords)
10
11 in
12     Convert.ToTable
13
```



Custom Function

You

Generate a DIM_DATE table for Power Query.

This table should have Year, Month (Number Short Month, Long Month), Day Columns.
It should also have a Portuguese Holiday column with 1 for holidays and 0 for non holidays.
Finally it should have a column to mark if it's a working day or not (working day must be non holiday and non weekend).

The range of dates in this table should be the 1st of January 2023, until the 31st of December of the current year.

First generate the complete list of Portuguese Holidays between the requested dates in a format which can be used inside M code.

Then use this list and return the complete M code in a format that I can directly paste into a new blank query in Power BI, ensuring syntax accuracy and proper function calls to avoid common errors such as misreferencing modules or functions.

ChatGPT

For this task, I'll first generate the list of Portuguese public holidays for the years 2023 and 2024, considering the range you've specified. Then, I'll provide the M code for creating the "DIM_DATE" table in Power Query for Power BI.

Portuguese Public Holidays for 2023 and 2024

Let's start by listing the Portuguese public holidays for these two years, taking into account that some holidays are fixed, and others, like Easter-related holidays, vary each year.

Fixed Holidays:

- New Year's Day: January 1

- Freedom Day: April 25

- Labour Day: May 1

- Portugal Day: June 10

- Assumption Day: August 15

- Republic Day: October 5

- All Saints' Day: November 1

- Restoration of Independence Day: December 1

- Immaculate Conception: December 8

- Christmas Day: December 25

Variable Holidays (Easter-related for 2023 and 2024):

- 2023: Good Friday is on April 7, Easter Sunday is on April 9, and Corpus Christi is on June 8.

- 2024: Good Friday is on March 29, Easter Sunday is on March 31, and Corpus Christi is on May 23.

M Code for DIM_DATE Table

Here's the M code to generate the "DIM_DATE" table, incorporating the holiday list and the logic for working days. This code is designed to be pasted into a new blank query in Power BI.

```
let
    StartDate = #date(2023, 1, 1),
    EndDate = #date(2024, 12, 31),
    Source = List.Dates(StartDate, Number.From(EndDate) - Number.From(StartDate) + 1, #duration(1, 0, 0, 0)),
    TableFromList = Table.FromList(Source, Splitter.SplitByNothing(), {"Date"}, null, ExtraValues.Error),
    ChangedType = Table.TransformColumnTypes(TableFromList,{{"Date", type date}}),
    AddedYear = Table.AddColumn(ChangedType, "Year", each Date.Year([Date])),
    AddedMonthNumber = Table.AddColumn(AddedYear, "Month (Number)", each Date.Month([Date])),
    AddedShortMonth = Table.AddColumn(AddedMonthNumber, "Short Month", each Text.Start(Date.ToText([Date]), "MMM")),
    AddedLongMonth = Table.AddColumn(AddedShortMonth, "Long Month", each Date.ToText([Date], "MMMM")),
    AddedDay = Table.AddColumn(AddedLongMonth, "Day", each Date.Day([Date])),
    // Define Portuguese Holidays
    Holidays = [
        #date(2023, 1, 1), #date(2023, 4, 7), #date(2023, 4, 9), #date(2023, 5, 1), #date(2023, 6, 8),
        #date(2023, 6, 10), #date(2023, 8, 15), #date(2023, 10, 5), #date(2023, 11, 1), #date(2023, 12, 1), #date(2023, 12, 8), #date(2023, 12, 25),
        #date(2024, 1, 1), #date(2024, 3, 29), #date(2024, 5, 1), #date(2024, 5, 23),
        #date(2024, 6, 10), #date(2024, 8, 15), #date(2024, 10, 5), #date(2024, 11, 1), #date(2024, 12, 1), #date(2024, 12, 8), #date(2024, 12, 25)
    ],
    AddedHoliday = Table.AddColumn(AddedDay, "Portuguese Holiday", each if List.Contains(Holidays, [Date]) then 1 else 0),
    AddedWorkingDay = Table.AddColumn(AddedHoliday, "Day of Week", each Date.DayOfWeek([Date])),
    AddedWorkingDay = Table.AddColumn(AddedWorkingDay, "Working Day", each if [Portuguese Holiday] = 0 and [Day of Week] < 5 then 1 else 0),
    RemovedColumns = Table.RemoveColumns(AddedWorkingDay, {"Day of Week"})
in
RemovedColumns
```



Cumulative Function

AG You

Add a Column with the cumulative working days for each year. This count must reset on every year and start again from 1 on the first working day.

Return the complete M code in a format that I can directly paste into a new blank query in Power BI, ensuring syntax accuracy and proper function calls to avoid common errors such as misreferencing modules or functions.

AG You

Based on the "Working Day" Column, add a Column with the cumulative working days for each year. This count must reset on every year and start again from 1 on the first working day.

Return the complete M code in a format that I can directly paste into a new blank query in Power BI, ensuring syntax accuracy and proper function calls to avoid common errors such as misreferencing modules or functions.

① [Expression.Error] The column 'Weekday' of the table wasn't found.

Details

isRecoverable: True

isExpected: True

	ABC 123 Date	ABC 123 Month (Number)	ABC 123 Short Month	ABC 123 Long Month	ABC 123 Day	ABC 123 Portuguese Holiday	ABC 123 Working Day	ABC 123 Cumulative Working Days
359	12/24/2023							
360	12/26/2023		12 dez	dezembro	26	0	1	360
361	12/27/2023		12 dez	dezembro	27	0	1	361
362	12/28/2023		12 dez	dezembro	28	0	1	362
363	12/29/2023		12 dez	dezembro	29	0	0	363
364	12/30/2023		12 dez	dezembro	30	0	0	364
365	12/31/2023		12 dez	dezembro	31	0	1	365
366	1/1/2024		1 jan	janeiro	1	1	0	1
367	1/2/2024		1 jan	janeiro	2	0	1	2
368	1/3/2024		1 jan	janeiro	3	0	1	3
369	1/4/2024		1 jan	janeiro	4	0	1	4



Column Translation



You

Translate my column headers to French



ChatGPT

To translate your column headers into French, I'll provide the translations for each of the column headers used in the previous M script:

- 'Date' -> 'Date'
- 'Year' -> 'Année'
- 'MonthNumber' -> 'NuméroMois'
- 'ShortMonth' -> 'MoisCourt'
- 'LongMonth' -> 'MoisLong'
- 'Day' -> 'Jour'
- 'IsHoliday' -> 'EstFérié'
- 'Weekday' -> 'JourDeLaSemaine'
- 'IsWorkingDay' -> 'EstJourOuvré'
- 'CumulativeWorkingDays' -> 'JoursOuvrésCumulés'

Here's how the script would look with the column headers translated to French:

```
m          copy code

let
    StartDate = #date(2023, 1, 1),
    EndDate = #date(2024, 12, 31),
    NumberOfWorkDays = Duration.Days(EndDate - StartDate) + 1,
    DateRange = List.Dates(StartDate, NumberOfWorkDays, #duration(1,0,0,0)),
    Source = Table.FromList(DateRange, Splitter.SplitByNothing(), {"Date"}),
    InsertedYear = Table.AddColumn(Source, "Année", each Date.Year([Date])),
    InsertedMonthNumber = Table.AddColumn(InsertedYear, "NuméroMois", each Date.Month),
    InsertedShortMonth = Table.AddColumn(InsertedMonthNumber, "MoisCourt", each Date.
    InsertedLongMonth = Table.AddColumn(InsertedShortMonth, "MoisLong", each Date.ToText("MM"))
    InsertedDay = Table.AddColumn(InsertedLongMonth, "Jour", each Date.Day([Date])),
    InsertedHoliday = Table.AddColumn(InsertedDay, "EstFérié", each if List.Contains(
        InsertedIsWorkingDay = Table.AddColumn(InsertedHoliday, "JourDeLaSemaine", each Date.D
    InsertedIsWorkingDay = Table.AddColumn(InsertedWeekday, "EstJourOuvré", each if [Date].
    // Adjusted calculation for cumulative working days
    AddedCumulativeWorkingDays = Table.Group(InsertedIsWorkingDay, {"Année"}, {"AllData"
    ExpandCumulativeWorkingDays = Table.TransformColumns(AddedCumulativeWorkingDays,
        CumulativeColumn = List.Accumulate(t[EstJourOuvré], (0), (state, current) =>
            CorrectedCumulativeColumn = if List.First(CumulativeColumn) = 0 then List.Skip(
                AddColumn = Table.FromColumns(Table.ToColumns(t) & {CorrectedCumulativeColumn
            in AddColumn}),
            ExpandedTable = Table.ExpandTableColumn(ExpandCumulativeWorkingDays, "AllData", [
                in
                ExpandedTable
```

The screenshot shows the Power Query Editor interface. At the top, there is a formula bar with the M script. Below it is a preview pane showing a table with 18 rows and 13 columns. The columns are labeled: Année, Date, NuméroMois, MoisCourt, MoisLong, Jour, EstFérié, JourDeLaSemaine, EstJourOuvré, and JoursOuvrésCumulés. The preview data shows dates from January 1, 2023, to January 18, 2023, with corresponding month numbers and names in Portuguese. To the right of the preview is a 'Query settings' pane with sections for 'Properties' (Name: Query) and 'Applied steps'. A red dashed box highlights the 'ExpandedTable' step under 'Applied steps', indicating it is the last step in the process.





4) Generate DAX

Create Standard Measures

Create Custom Measures



Generate Measure using Tabular Editor



You

Generate C# code for Tabular Editor to Create measures from columns



ChatGPT

Tabular Editor is a tool primarily used for working with Tabular Models in Microsoft Analysis Services, Power BI, and Power Pivot. It allows for scripting and automation through the use of C# or the Tabular Object Model (TOM). Below is an example of C# code that you can use to create measures from columns using Tabular Editor:

csharp

[Copy code](#)

```
using System;
using System.Linq;
using TabularEditor;

class Program
{
    static void Main()
    {
        // Path to your Tabular Model (Power BI or SSAS Tabular)
        string modelPath = @"C:\Path\To\Your\Model.pbix";

        // Open the model
        Model model = new Model();
        model.Open(modelPath);

        // Specify the table and columns for which you want to create m
        string tableName = "YourTableName";
        string[] columnNames = { "Column1", "Column2" };
    }
}
```

Please note the following:

1. Make sure to replace `"`C:\Path\To\Your\Model.pbix`"`, `"`YourTableName`"`, `"`Column1`"`, and `"`Column2`"` with the actual path to your Power BI model and the specific table and columns for which you want to create measures.
2. The code uses the `SUM` aggregation as an example. You can modify the formula in the `AddMeasure` method to suit your specific requirements.
3. Ensure that you have the Tabular Editor library referenced in your project.
4. This code is a simple example, and you may need to adapt it based on your specific scenario and requirements.

Always make sure to backup your model before making any significant changes, and thoroughly test the code in a safe environment.

Generate Measure using Tabular Editor



How can I help you today?

Explain options trading
if I'm familiar with buying and selling stocks

Brainstorm content ideas
for my new podcast on urban design

Message ChatGPT...

Tell me a fun fact
about the Golden State Warriors

Brainstorm names
for an orange cat we're adopting from the shelter

Última modificação da apresentação: 12 de janeiro



ChatGPT can make mistakes. Consider checking important information.



Generate Measure using Tabular Editor

Expression Editor C# Script

Samples + X | 100 %

```
1 // Iterate through all tables in the model
2 foreach (var table in Model.Tables)
3 {
4     // Iterate through all columns in the table
5     foreach (var column in table.Columns)
6     {
7         // Check if the column data type is numeric
8         if (column.DataType == DataType.Int64 ||
9             column.DataType == DataType.Decimal ||
10            column.DataType == DataType.Double)
11        {
12            // Define the measure name
13            string measureName = "SUM " + column.Name;
14
15            // Define the DAX expression for the measure
16            string measureExpression = "SUM(" + column.DaxObjectName + ")";
17
18            // Create the measure
19            var measure = table.AddMeasure(measureName);
20            measure.Expression = measureExpression;
21        }
22    }
23}
```



- ▼ Sales
 - > Partitions
 - SUM SalesOrderLineKey 1
 - SUM ResellerKey
 - SUM CustomerKey 1
 - SUM ProductKey 1
 - SUM OrderDateKey
 - SUM DueDateKey
 - SUM ShipDateKey
 - SUM SalesTerritoryKey 1
 - SUM Order Quantity
 - SUM Unit Price
 - SUM Extended Amount
 - SUM Product Standard Cost
 - SUM Total Product Cost
 - SUM Sales Amount
 - SUM Unit Price Discount Pct
 - SUM CurrencyKey
 - SUM SalesOrderLineKey 2
 - SUM ResellerKey
 - SUM CustomerKey 2
 - SUM ProductKey 2
 - SUM OrderDateKey
 - SUM DueDateKey
 - SUM ShipDateKey
 - SUM SalesTerritoryKey 2
 - SUM Order Quantity 1
 - SUM Unit Price 1
 - SUM Extended Amount 1
 - SUM Product Standard Cost 1
 - SUM Total Product Cost 1
 - SUM Sales Amount 1
 - SUM Unit Price Discount Pct 1
 - SUM CurrencyKey 3
 - Order Quantity
 - Unit Price
 - Extended Amount
 - Product Standard Cost
 - Total Product Cost
 - Sales Amount
 - Unit Price Discount Pct
 - > Reseller
 - > Currency Rate
 - > Currency
 - > Translations

- ▼ Sales
 - > Partitions
 - SUM Order Quantity
 - SUM Unit Price
 - SUM Extended Amount
 - SUM Product Standard Cost
 - SUM Total Product Cost
 - SUM Sales Amount
 - SUM Unit Price Discount Pct
 - SUM CurrencyKey
 - SUM SalesOrderLineKey 2
 - SUM ResellerKey
 - SUM CustomerKey 2
 - SUM ProductKey 2
 - SUM OrderDateKey
 - SUM DueDateKey
 - SUM ShipDateKey
 - SUM SalesTerritoryKey 2
 - SUM Order Quantity 1
 - SUM Unit Price 1
 - SUM Extended Amount 1
 - SUM Product Standard Cost 1
 - SUM Total Product Cost 1
 - SUM Sales Amount 1
 - SUM Unit Price Discount Pct 1
 - SUM CurrencyKey 3
 - Order Quantity
 - Unit Price
 - Extended Amount
 - Product Standard Cost
 - Total Product Cost
 - Sales Amount
 - Unit Price Discount Pct
 - > Reseller
 - > Currency Rate
 - > Currency
 - > Translations

Create DAX measure based on the date



How can I help you today?

Compare business strategies
for transitioning from budget to luxury

Write a Python script
to automate sending daily email reports

Brainstorm names
for an orange cat we're adopting from the shelter

Write a thank-you note
to our babysitter for the last-minute help

I
Message ChatGPT...

ChatGPT can make mistakes. Consider checking important information.

Create DAX measure based on the date

Expression Editor | C# Script

Samples | + | X | P | F5 | 100 %

```
1 // Get reference to the Sales table
2 var salesTable = Model.Tables["Sales"];
3
4 // Create Year To Date (YTD) measure
5 var ytdMeasure = salesTable.AddMeasure("Sales YTD", "CALCULATE(SUM(Sales[Amount]), DATESYTD(Calendar[Date]))");
6
7 // Create Month To Date (MTD) measure
8 var mtdMeasure = salesTable.AddMeasure("Sales MTD", "CALCULATE(SUM(Sales[Amount]), DATESMTD(Calendar[Date]))");
9
10
11
12
13
14
```

 Sales YTD
 Sales MTD
 Order Quantity
 Unit Price
 Extended Amount
 Product Standard Cost
 Total Product Cost
 Sales Amount
 Unit Price Discount Pct

 Sales YTD
 Sales MTD
 Order Quantity
 Unit Price
 Extended Amount
 Product Standard Cost
 Total Product Cost
 Sales Amount
 Unit Price Discount Pct



Create DAX measure rolling sum for the year

specified.



You

remove the Save changes and simplify the code



ChatGPT

Sure, here's the simplified version of the code without saving changes:

```
csharp
// Specify the name of the Sales table
var salesTableName = "Sales";

// Specify the name of the Sales Amount column
var salesAmountColumnName = "Sales Amount";

// Create Year to Date (YTD) measure
model.Tables[salesTableName].AddMeasure(
    name: "Sales YTD",
    expression: "TOTALYTD(SUM('Sales'["
        + salesAmountColumnName + "]), 'Date'[Date])"
);

// Create Month to Date (MTD) measure
model.Tables[salesTableName].AddMeasure(
    name: "Sales MTD",
    expression: "TOTALMTD(SUM('Sales'["
        + salesAmountColumnName + "]), 'Date'[Date])"
);
```

This code directly adds the measures to the specified Sales table without saving the changes explicitly.

Make sure to execute this script within the Advanced Scripting feature of Tabular Editor.



Message ChatGPT...



ChatGPT can make mistakes. Consider checking important information.

Create DAX measure rolling sum for the year

Translation: (No translation) Filter

Expression Editor C# Script

```
1 // Specify the name of the Sales table
2 var salesTableName = "Sales";
3
4 // Specify the name of the Sales Amount column
5 var salesAmountColumnName = "Sales Amount";
6
7 // Specify the name of the rolling SUM measure
8 var rollingSumMeasureName = "Rolling Sales Amount";
9
10 // Specify the number of periods for the rolling sum
11 var rollingSumPeriods = 3; // Change this according to your requirement
12
13 // Define the expression for the rolling SUM measure
14 var rollingSumExpression = "CALCULATE(SUM('" + salesTableName + "'[" + salesAmountColumnName +
15     "]), DATESINPERIOD('Date'[Date], LASTDATE('Date'[Date]), -" + rollingSumPeriods + ", MONTH))";
16
17 // Create the rolling SUM measure
18 Model.Tables[salesTableName].AddMeasure(
19     name: rollingSumMeasureName,
20     expression: rollingSumExpression
21 );
22
```

-  Sales YTD
-  Sales MTD
-  Rolling Sales Amount
-  Order Quantity
-  Unit Price
-  Extended Amount
-  Product Standard Cost
-  Total Product Cost
-  Sales Amount
-  Unit Price Discount Pct
-  Descaller





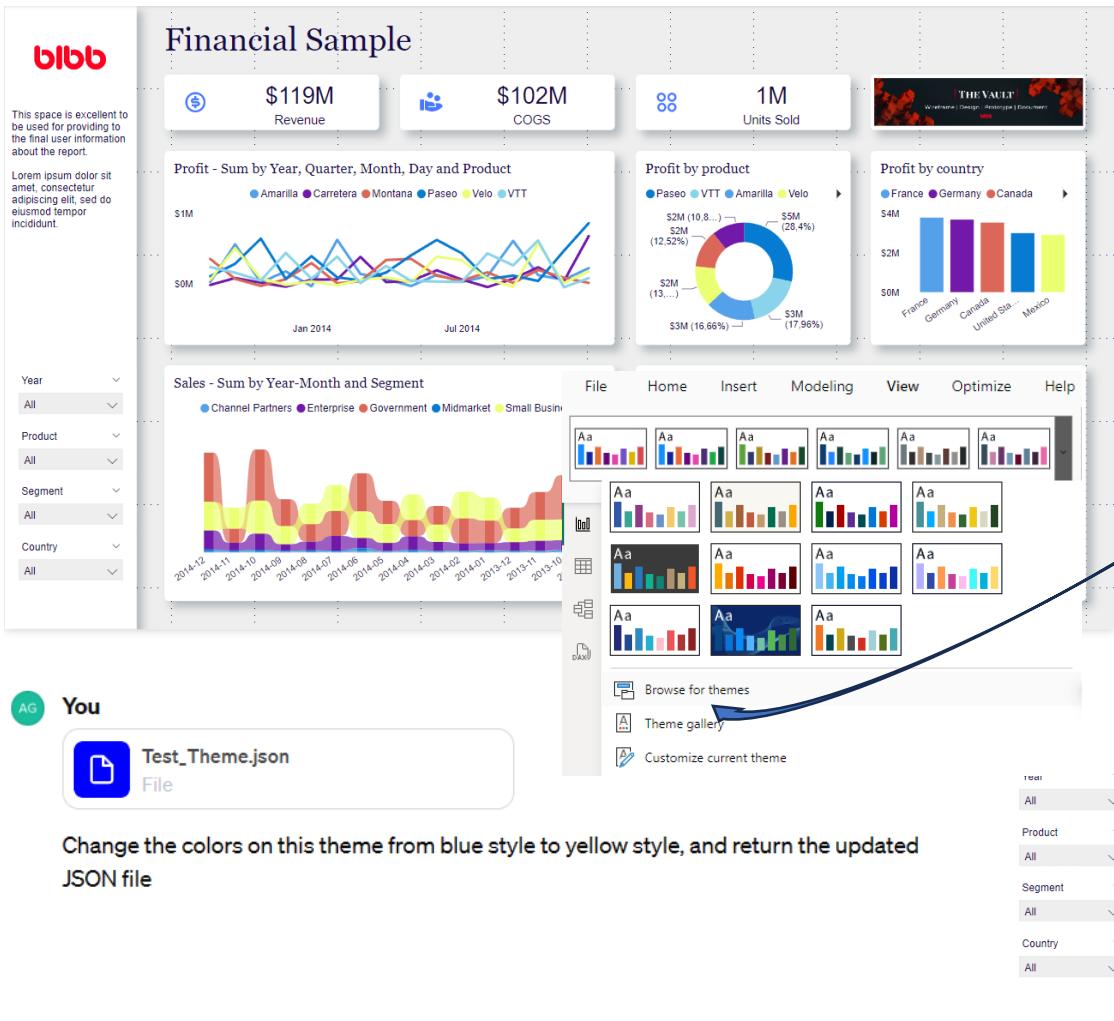
5) Enhance Report Theme & Imaging

Update Theme Colors

Add new Objects to Theme



Adjust Report Theme Objects Code

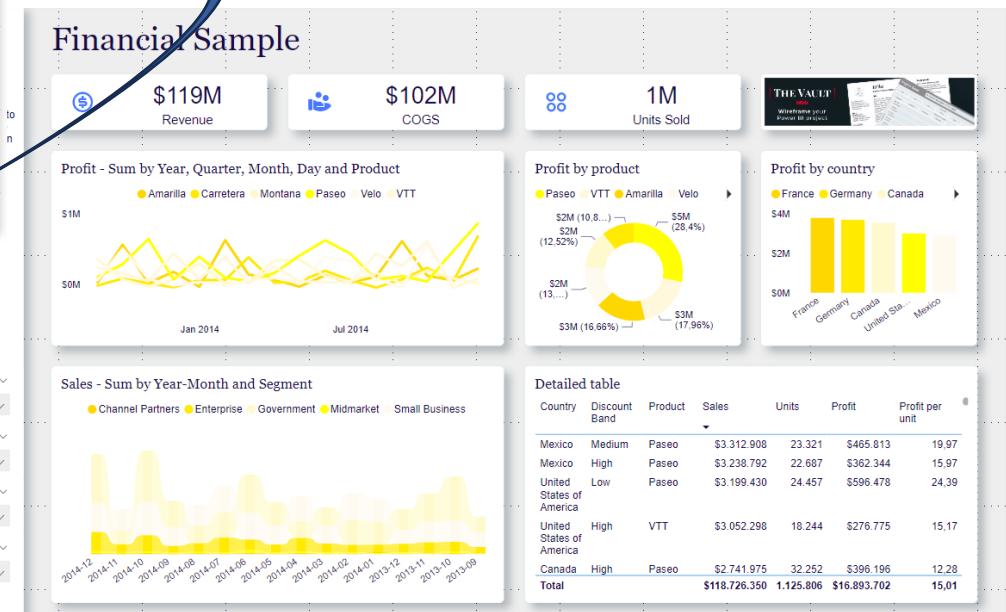


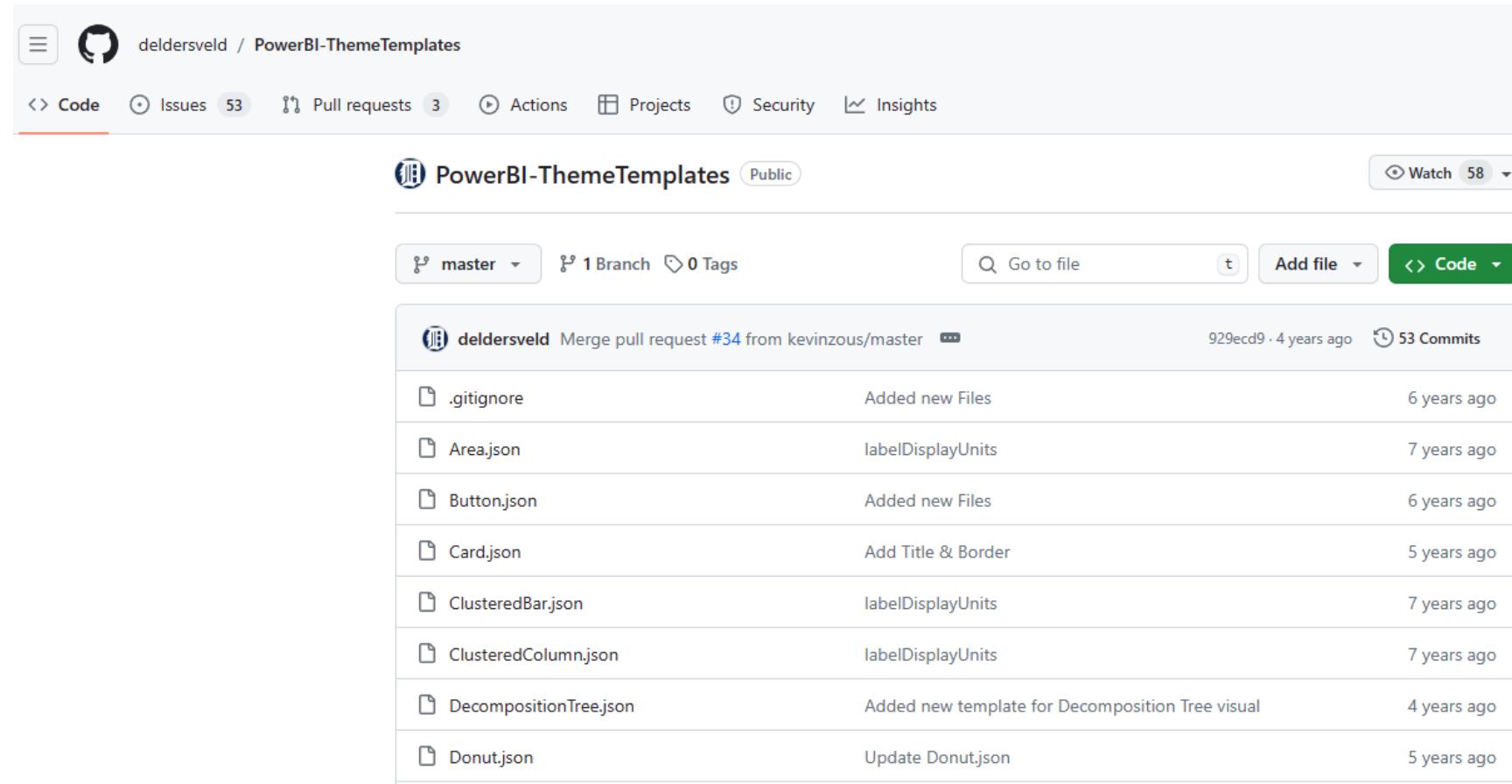
Change the colors on this theme from blue style to yellow style, and return the updated JSON file

AG You
can't download the file, please share a downloadable link

 ChatGPT
Certainly! Here is the downloadable link for the updated Power BI theme file with a yellow color style:

[Download Updated Yellow Style Theme](#)





A screenshot of a GitHub repository page for 'PowerBI-ThemeTemplates'. The repository is public and has 58 stars. It contains 1 branch and 0 tags. The master branch has 53 commits. The commits are listed below:

Author	Commit Message	Date
deldersveld	Merge pull request #34 from kevinzous/master	929ecd9 · 4 years ago
	.gitignore	Added new Files
	Area.json	labelDisplayUnits
	Button.json	Added new Files
	Card.json	Add Title & Border
	ClusteredBar.json	labelDisplayUnits
	ClusteredColumn.json	labelDisplayUnits
	DecompositionTree.json	Added new template for Decomposition Tree visual
	Donut.json	Update Donut.json

Source: [GitHub - deldersveld/PowerBI-ThemeTemplates: Snippets for assembling Power BI Themes](https://github.com/deldersveld/PowerBI-ThemeTemplates)

Adjust Report Theme Objects Code



You

The theme lacks a definition for the table visual. Use this one , updated to match the yellow colors and return the JSON theme file including the table visual.

```
{  
  "name": "TableTemplate",  
  "visualStyles": {  
    "tableEx": {  
      "*": {  
        "grid": [  
          "gridVertical": true,  
          "gridVerticalColor": { "solid": { "color": "#FFB8DD" } },  
          "gridVerticalWeight": 2,  
          "gridHorizontal": true,  
          "gridHorizontalColor": { "solid": { "color": "#FFB8DD" } },  
          "gridHorizontalWeight": 2,  
        ]  
      }  
    }  
  }  
}
```



ChatGPT

The Power BI theme file has been updated to include the table visual with yellow color styles. You can download the updated theme file using the link below:

[Download Updated Theme with Table Visual \[?\]](#)



This space is excellent to be used for providing to the final user information about the report.
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt.

Year
All
Product
All
Segment
All
Country
All

Financial Sample



\$119M

Revenue



\$102M

COGS



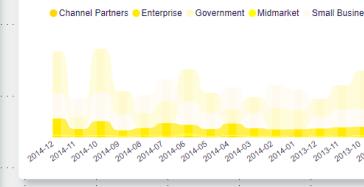
88 1M

Units Sold

Profit - Sum by Year, Quarter, Month, Day and Product



Sales - Sum by Year-Month and Segment



Financial Sample



\$119M

Revenue



\$102M

COGS



88 1M

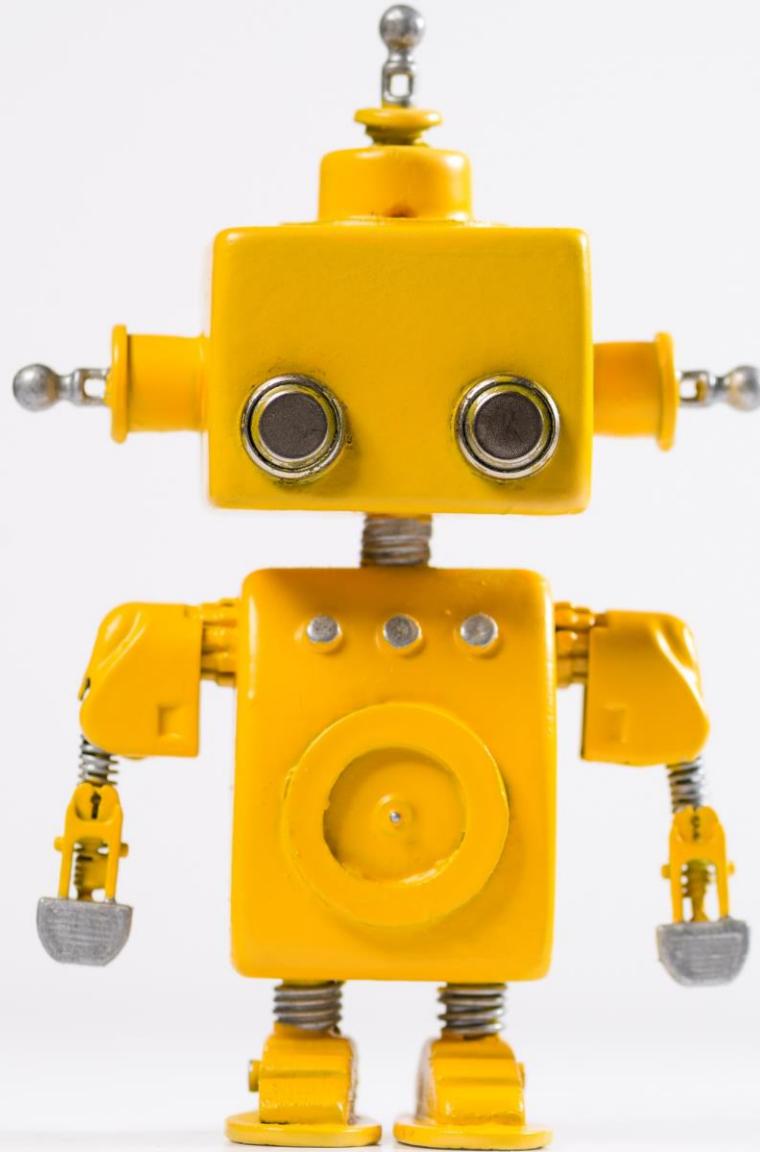
Units Sold

Profit by product



6) Publication & Administration

- List PBI Workspace Artifacts
- Deployment
- Backup Models and Reports
- Deploy to Prod WorkSpace
- Refresh Dataset



List PowerBI Workspace Artifacts using PowerBI cmdlets for PowerShell

The screenshot shows the ChatGPT interface. At the top, there's a circular icon with a swirl pattern and the text "How can I help you today?". Below this, there are four rectangular cards with rounded corners, each containing a prompt:

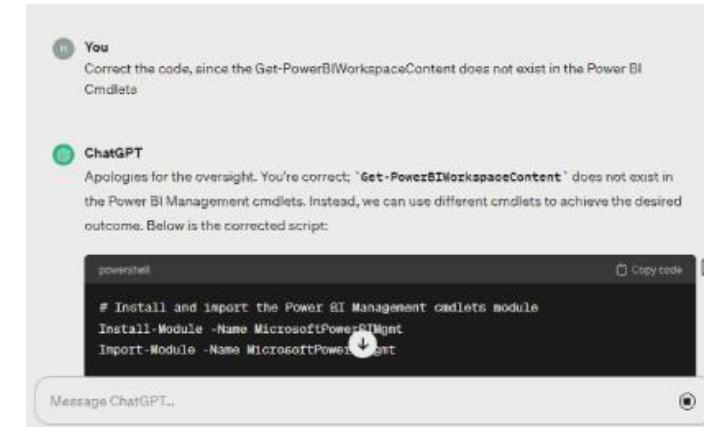
- Help me pick**
a gift for my dad who loves fishing
- Brainstorm names**
for an orange cat we're adopting from the shelter
- Plan an itinerary**
for a literary tour of England, visiting famous authors' ...
- Make a content strategy**
for a newsletter featuring free local weekend events

At the bottom, there's a large input field with the placeholder "Message ChatGPT..." and an upward arrow icon. Below the input field, a small note says "ChatGPT can make mistakes. Consider checking important information."

List PowerBI Workspace Artifacts using PowerBI cmdlets for PowerShell



```
Windows PowerShell ISE
File Edit View Tools Debug Add-ons Help
Untitled1.ps1* X
1 # Install and import the Power BI Management cmdlets module
2 Install-Module -Name MicrosoftPowerBIMgmt
3 Import-Module -Name MicrosoftPowerBIMgmt
4
5 # Authenticate user
6 Connect-PowerBIServiceAccount
7
8 # Get workspace information
9 $workspace = Get-PowerBIWorkspace -Scope Individual -Name "FabricBI"
10
11 # Get datasets in the workspace
12 $datasets = Get-PowerBIDataset -WorkspaceId $workspace.Id
13
14 # Get reports in the workspace
15 $reports = Get-PowerBIReport -WorkspaceId $workspace.Id
16
17 # Get dashboards in the workspace
18 $dashboards = Get-PowerBIDashboard -WorkspaceId $workspace.Id
19
20 # Display the artifacts
21 Write-Host "Datasets in FabricBI Workspace:"
22 foreach ($dataset in $datasets) {
23     Write-Host "Name: $($dataset.Name), Type: Dataset"
24 }
25
26 Write-Host "Reports in FabricBI Workspace:"
27
28 Write-Host "Dashboards in FabricBI Workspace:"
29 foreach ($dashboard in $dashboards) {
30     Write-Host "Name: $($dashboard.DisplayName), Type: Dashboard"
31 }
32
33 Environment : Public
34 TenantId    : 1c6ebc60-5ab4-42f0-be8c-2fc60286bd97
35 UserName    : Ricardo.Santos@onmicrosoft.com
36
37 Datasets in FabricBI Workspace:
38 Name: Fabric_PBI_PT2023, Type: Dataset
39 Name: Fabric_PT_LK, Type: Dataset
40 Name: Fabric_PT_LKH, Type: Dataset
41 Name: Regional Sales Sample, Type: Dataset
42 Reports in FabricBI Workspace:
43 Name: Regional Sales Sample, Type: Report
44 Dashboards in FabricBI Workspace:
```



```
Environment : Public
TenantId    : 1c6ebc60-5ab4-42f0-be8c-2fc60286bd97
UserName    : Ricardo.Santos@onmicrosoft.com

Datasets in FabricBI Workspace:
Name: Fabric_PBI_PT2023, Type: Dataset
Name: Fabric_PT_LK, Type: Dataset
Name: Fabric_PT_LKH, Type: Dataset
Name: Regional Sales Sample, Type: Dataset

Reports in FabricBI Workspace:
Name: Regional Sales Sample, Type: Report

Dashboards in FabricBI Workspace:
```

PS C:\WINDOWS\system32\WindowsPowerShell\v1.0>

Publish Local PowerBI Report to Workspace

```
# Get reports in the workspace
$reports = Get-PowerBIReport -WorkspaceId $workspace.Id

# Get dashboards in the workspace
$dashboards = Get-PowerBIDashboard -WorkspaceId $workspace.Id

# Display the artifacts
Write-Host "Datasets in FabricBI Workspace:"
foreach ($dataset in $datasets) {
    Write-Host "Name: $($dataset.Name), Type: Dataset"
}

Write-Host "Reports in FabricBI Workspace:"
foreach ($report in $reports) {
    Write-Host "Name: $($report.Name), Type: Report"
}

Write-Host "Dashboards in FabricBI Workspace:"
foreach ($dashboard in $dashboards) {
    Write-Host "Name: $($dashboard.DisplayName), Type: Dashboard"
}
```

In this script:

- We use `Get-PowerBIDataset`, `Get-PowerBIReport`, and `Get-PowerBIDashboard` cmdlets to retrieve datasets, reports, and dashboards, respectively, within the specified workspace.
- Then, we loop through each collection and display the name and type of each artifact.

Make sure to replace `"**FabricBI**"` with the name of your actual workspace. When you run this script, it will prompt you to log in with your Power BI credentials if you haven't already authenticated.

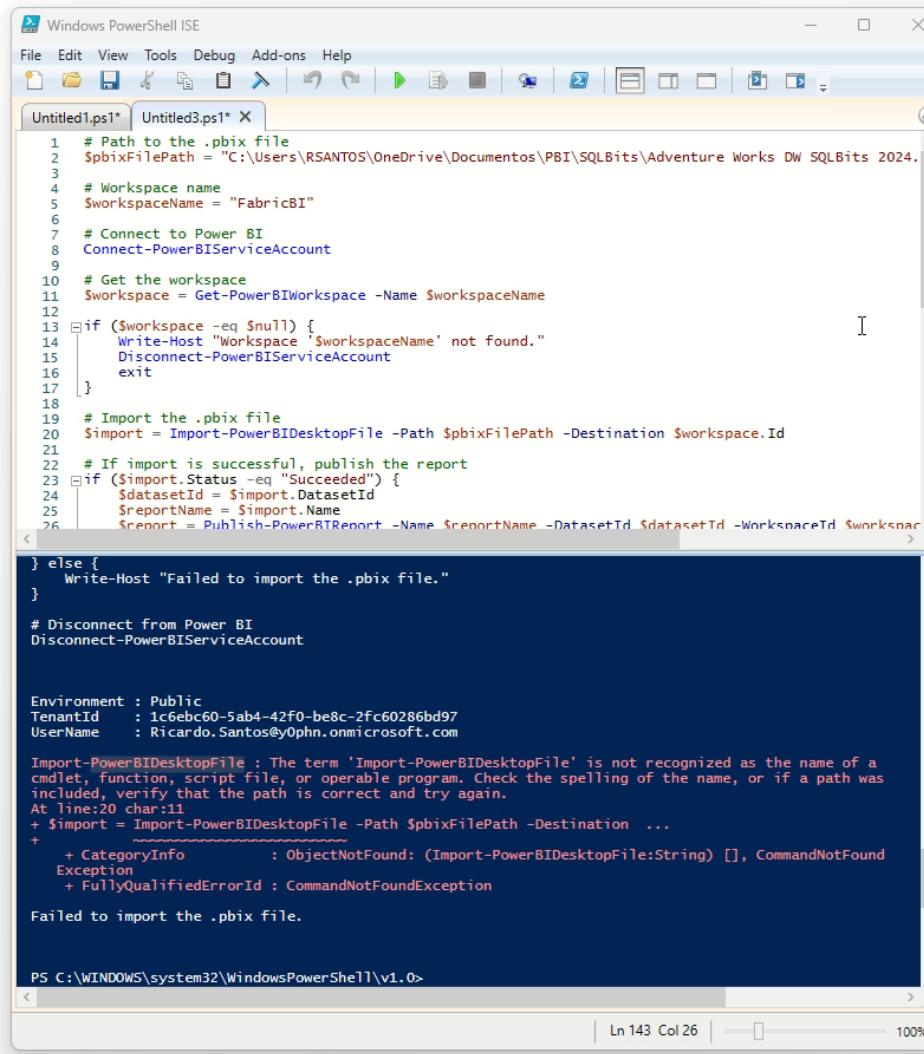


Message ChatGPT...



ChatGPT can make mistakes. Consider checking important information.

Publish Local PowerBI Report to Workspace - Iterations



The screenshot shows a Windows PowerShell ISE window with two tabs: Untitled1.ps1* and Untitled3.ps1*. The Untitled1.ps1* tab contains the following PowerShell script:

```
1 # Path to the .pbix file
2 $pbixFilePath = "C:\Users\RSANTOS\OneDrive\Documentos\PBI\SQLBits\Adventure Works DW SQLBits 2024.pbx
3
4 # Workspace name
5 $workspaceName = "FabricBI"
6
7 # Connect to Power BI
8 Connect-PowerBIServiceAccount
9
10 # Get the workspace
11 $workspace = Get-PowerBIWorkspace -Name $workspaceName
12
13 if ($workspace -eq $null) {
14     Write-Host "'$workspaceName' not found."
15     Disconnect-PowerBIServiceAccount
16     exit
17 }
18
19 # Import the .pbix file
20 $import = Import-PowerBIDesktopFile -Path $pbixFilePath -Destination $workspace.Id
21
22 # If import is successful, publish the report
23 if ($import.Status -eq "Succeeded") {
24     $datasetId = $import.DatasetId
25     $reportName = $import.Name
26     $report = Publish-PowerBIRReport -Name $reportName -DatasetId $datasetId -WorkspaceId $workspace.Id
}
27
28 } else {
29     Write-Host "Failed to import the .pbix file."
30 }
31
32 # Disconnect from Power BI
33 Disconnect-PowerBIServiceAccount
34
35
Environment : Public
TenantId    : 1c6ebc60-5ab4-42f0-be8c-2fc60286bd97
UserName    : Ricardo.Santos@ohn.onmicrosoft.com
36
37 Import-PowerBIDesktopFile : The term 'Import-PowerBIDesktopFile' is not recognized as the name of a cmdlet, function, script file, or operable program. Check the spelling of the name, or if a path was included, verify that the path is correct and try again.
At line:20 char:11
+ $import = Import-PowerBIDesktopFile -Path $pbixFilePath -Destination ...
+ ~~~~~
+ CategoryInfo          : ObjectNotFound: (Import-PowerBIDesktopFile:String) [], CommandNotFoundException
+ FullyQualifiedErrorId : CommandNotFoundException
38
39 Failed to import the .pbix file.
40
41
```

The bottom pane of the ISE shows the error output for the command Import-PowerBIDesktopFile, which is not recognized as a cmdlet.

Publish Local PowerBI Report to Workspace - Publish



```
Windows PowerShell ISE
File Edit View Tools Debug Add-ons Help
Untitled1.ps1* Untitled3.ps1* X
5 $workspaceName = "FabricBI"
6
7 # Connect to Power BI
8 Connect-PowerBIServiceAccount
9
10 # Get the workspace
11 $workspace = Get-PowerBIWorkspace -Name $workspaceName
12
13 if ($workspace -eq $null) {
14     Write-Host "Workspace '$workspaceName' not found."
15     Disconnect-PowerBIServiceAccount
16     exit
17 }
18
19 # Create new report in the workspace
20 $newReport = New-PowerBIReport -WorkspaceId $workspace.Id -Path $pbixFilePath -Name "Adventure Works DW"
21
22 if ($newReport -ne $null) {
23     Write-Host "Report created successfully in workspace '$workspaceName'."
24 } else {
25     Write-Host "Failed to create report."
26 }
27
28 # Disconnect from Power BI
29 Disconnect-PowerBIServiceAccount
30

Write-Host "Workspace '$workspaceName' not found."
Disconnect-PowerBIServiceAccount
exit
}

# Create new report in the workspace
$newReport = New-PowerBIReport -WorkspaceId $workspace.Id -Path $pbixFilePath -Name "Adventure Works DW"
if ($newReport -ne $null) {
    Write-Host "Report created successfully in workspace '$workspaceName'."
} else {
    Write-Host "Failed to create report."
}
# Disconnect from Power BI
Disconnect-PowerBIServiceAccount

Environment : Public
TenantId    : 1c6ebc60-5ab4-42f0-be8c-2fc60286bd97
UserName   : Ricardo.Santos@y0phn.onmicrosoft.com

Report created successfully in workspace 'FabricBI'.

PS C:\WINDOWS\system32\WindowsPowerShell\v1.0> |
```

You
correct the code since the Import-PowerBIDesktopFile does not exist, and use New-PowerBIReport

ChatGPT
Apologies for the confusion. Let's adjust the script to use 'New-PowerBIReport' instead of 'Import-PowerBIDesktopFile'. Here's the corrected script:

```
powershell
# Path to the .pbix file
$pbixFilePath = "C:\Users\Ricardo.Santos@y0phn.onmicrosoft.com\OneDrive\Documentos\PBI\SQLBits\Adventure Works DW.pbix"
```

```
Environment : Public
TenantId    : 1c6ebc60-5ab4-42f0-be8c-2fc60286bd97
UserName   : Ricardo.Santos@y0phn.onmicrosoft.com

Datasets in FabricBI Workspace:
Name: Fabric_PBI_PT2023, Type: Dataset
Name: Fabric_PT_LK, Type: Dataset
Name: Fabric_PT_LKH, Type: Dataset
Name: Regional Sales Sample, Type: Dataset
Name: Adventure Works DW SQLBits 2024, Type: Dataset
Reports in FabricPT Workspace:
Name: Regional Sales Sample, Type: Report
Name: Adventure Works DW SQLBits 2024, Type: Report
Dashboards in FabricPT Workspace:
PS C:\WINDOWS\system32\WindowsPowerShell\v1.0> |
```

Manage DevOps in PRO (Automatically backup all reports)

```
# Create local folder if it doesn't exist
if (-not (Test-Path $localFolderPath)) {
    New-Item -Path $localFolderPath -ItemType Directory | Out-Null
}

# Backup each report to .pbix file with today's date
foreach ($report in $reports) {
    $localPbixFileName = $report.Name + "_" + $dateToday + ".pbix"
    $localPbixFilePath = Join-Path -Path $localFolderPath -ChildPath $localPbixFileName
    if (Test-Path $localPbixFilePath) {
        Remove-Item $localPbixFilePath
    }
    Export-PowerBIReport -Id $report.Id -OutFile $localPbixFilePath -Format PBIX -IncludeData
    Write-Host "Report '$($report.Name)' backed up successfully to '$localPbixFileName'"
}

# Disconnect from Power BI
Disconnect-PowerBIServiceAccount
```

In this script:

- We define the parameters such as the workspace name, local parent folder path, and today's date.
- We connect to Power BI using `'Connect-PowerBIServiceAccount'`.
- We retrieve the workspace and all reports within it.
- We create a local folder named with today's date if it doesn't exist.
- For each report, we export it to a .pbix file with today's date, and if the file already exists, we replace it by removing it first.
- Finally, we disconnect from Power BI using `'Disconnect-PowerBIServiceAccount'`.

Ensure you have the necessary permissions to access reports in the workspace and write files to the local folder. Adjust the workspace and local folder paths as needed.

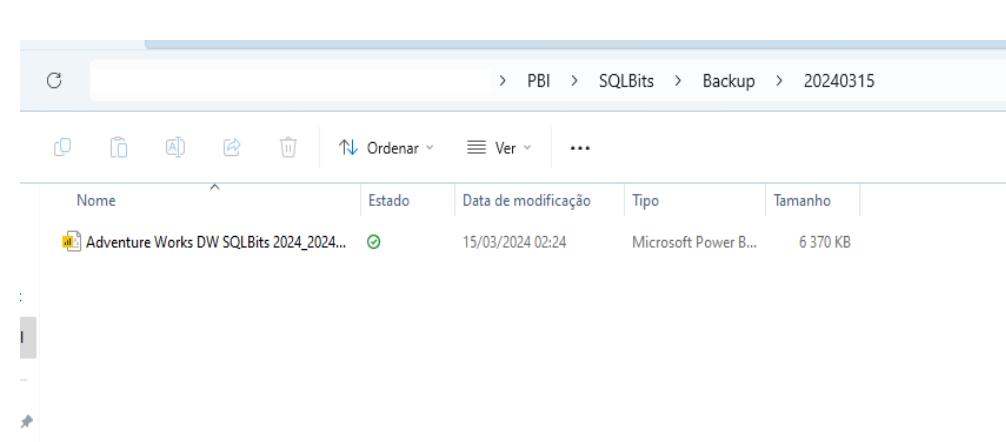
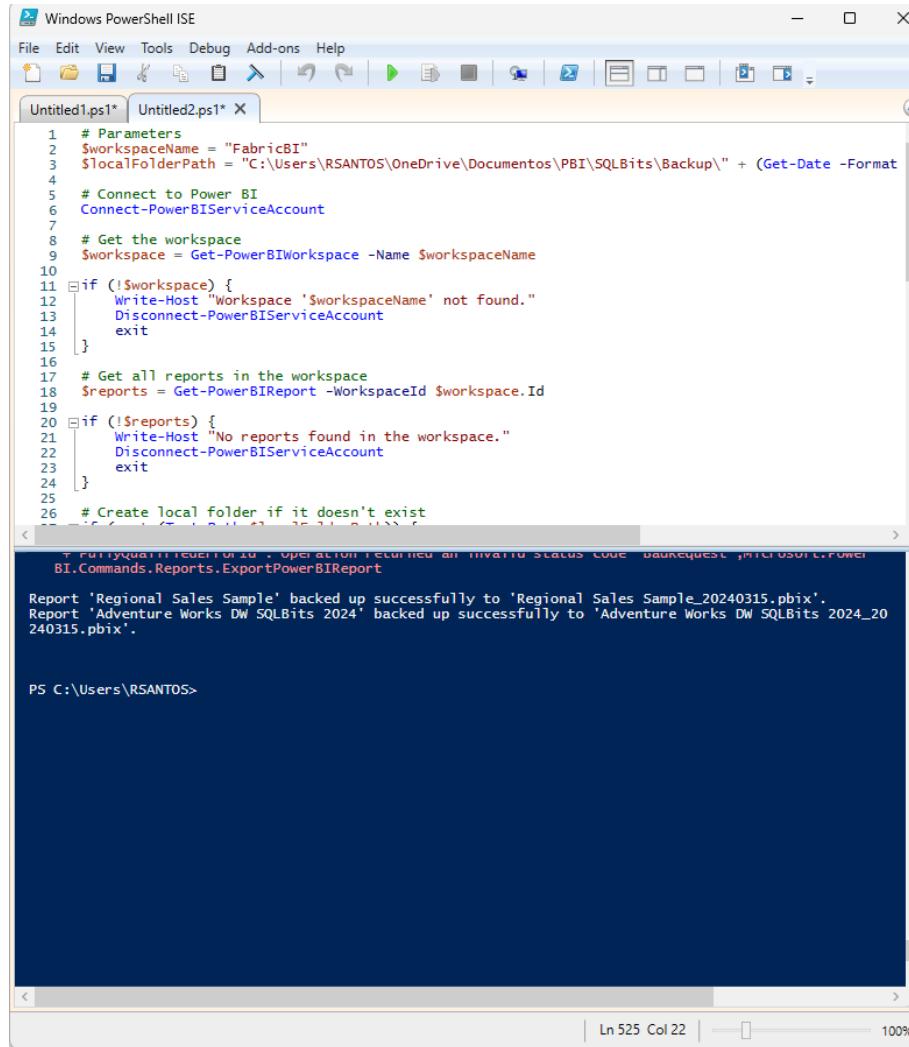


Message ChatGPT...



ChatGPT can make mistakes. Consider checking important information.

Manage DevOps in PRO (Automatically backup all reports)



```
Windows PowerShell ISE
File Edit View Tools Debug Add-ons Help
Untitled1.ps1* Untitled2.ps1* X
1 # Parameters
2 $workspaceName = "FabricBI"
3 $localFolderPath = "C:\Users\RSANTOS\OneDrive\Documentos\PBI\SQLBits\Backup\" + (Get-Date -Format
4
5 # Connect to Power BI
6 Connect-PowerBIServiceAccount
7
8 # Get the workspace
9 $workspace = Get-PowerBIWorkspace -Name $workspaceName
10
11 if (!$workspace) {
12     Write-Host "Workspace '$workspaceName' not found."
13     Disconnect-PowerBIServiceAccount
14     exit
15 }
16
17 # Get all reports in the workspace
18 $reports = Get-PowerBIReport -WorkspaceId $workspace.Id
19
20 if (!$reports) {
21     Write-Host "No reports found in the workspace."
22     Disconnect-PowerBIServiceAccount
23     exit
24 }
25
26 # Create local folder if it doesn't exist
27 $localFolderPath = "C:\Users\RSANTOS\OneDrive\Documentos\PBI\SQLBits\Backup\" + (Get-Date -Format
28
29 foreach ($report in $reports) {
30     $reportPath = Join-Path $localFolderPath $report.Name
31     if (-Not (Test-Path $reportPath)) {
32         New-Item -ItemType Directory -Path $reportPath
33     }
34     $report | Export-PowerBIReport -Path $reportPath
35 }
36
37 Write-Host "All reports backed up successfully to $localFolderPath"
38
39 PS C:\Users\RSANTOS>
```

The PowerShell script performs the following steps:

- Defines parameters: workspaceName ("FabricBI") and localFolderPath ("C:\Users\RSANTOS\OneDrive\Documentos\PBI\SQLBits\Backup").
- Connects to Power BI service account.
- Gets the workspace by name.
- If no workspace is found, exits.
- Gets all reports in the workspace.
- If no reports are found, exits.
- Creates a local folder if it doesn't exist.
- Exports each report to a local file using the Export-PowerBIReport cmdlet.
- Outputs a success message indicating all reports were backed up.

The resulting backup files are listed in the File Explorer:

Nome	Estado	Data de modificação	Tipo	Tamanho
Adventure Works DW SQLBits 2024_20240315.pbix	OK	15/03/2024 02:24	Microsoft Power B...	6 370 KB



Manage DevOps in PRO (without Deployment Pipelines)

```
}

Write-Host "Report '$reportName' exported to a local .pbix file with today's date."

# Publish report to the destination workspace using New-PowerBIReport with -ConflictAction CreateOrOverwrite
$publishedReport = New-PowerBIReport -Path $localPbixFilePath -Name $reportName -WorkspaceName $destinationWorkspaceName -ConflictAction CreateOrOverwrite

if ($publishedReport -eq $null) {
    Write-Host "Failed to publish the report to the destination workspace."
    Disconnect-PowerBIServiceAccount
    exit
}

Write-Host "Report '$reportName' redeployed successfully to '$destinationWorkspaceName'"

# Disconnect from Power BI
Disconnect-PowerBIServiceAccount
```

This script performs the following actions:

1. Connects to Power BI using the current user's credentials.
2. Gets the source ("FabricBI") and destination ("FabricBI Prod") workspaces.
3. Defines the local path to save the .pbix file with today's date.
4. Creates the local folder if it doesn't exist.
5. Exports the report to a local .pbix file with today's date.
6. Publishes the report to the destination workspace using `New-PowerBIReport` with `-ConflictAction CreateOrOverwrite`.
7. Disconnects from Power BI.

Ensure you have the necessary permissions to access, export, and publish reports in both workspaces.

Adjust the workspace and folder paths as needed.



Message ChatGPT...



ChatGPT can make mistakes. Consider checking important information.

Manage DevOps in PRO (without Deployment Pipelines)



```
Windows PowerShell ISE
File Edit View Tools Debug Add-ons Help
Untitled1.ps1* X
4/
48 }
49
50 Export-PowerBIReport -Id $report.Id -OutFile $localPbixFilePath
51
52 if (-not (Test-Path $localPbixFilePath)) {
53     Write-Host "Failed to export the report to a local .pbix file."
54     Disconnect-PowerBIServiceAccount
55     exit
56 }
57 Write-Host "Report '$reportName' exported to a local .pbix file with today's date."
58
59 # Publish report to the destination workspace using New-PowerBIReport with -ConflictAction CreateOrOverwrite
60 $publishedReport = New-PowerBIReport -Path $localPbixFilePath -Name $reportName -WorkspaceId $dest
61
62 if ($publishedReport -eq $null) {
63     Write-Host "Failed to publish the report to the destination workspace."
64     Disconnect-PowerBIServiceAccount
65     exit
66 }
67 Write-Host "Report '$reportName' redeployed successfully to '$destinationWorkspaceName'."
68
69 # Disconnect from Power BI
70 Disconnect-PowerBIServiceAccount
71
72 Write-Host "Report '$reportName' exported to a local .pbix file with today's date."
73
74 # Publish report to the destination workspace using New-PowerBIReport with -ConflictAction CreateOrOverwrite
75 $publishedReport = New-PowerBIReport -Path $localPbixFilePath -Name $reportName -WorkspaceId $destination
76
77 if ($publishedReport -eq $null) {
78     Write-Host "Failed to publish the report to the destination workspace."
79     Disconnect-PowerBIServiceAccount
80     exit
81 }
82 Write-Host "Report '$reportName' redeployed successfully to '$destinationWorkspaceName'."
83
84 # Disconnect from Power BI
85 Disconnect-PowerBIServiceAccount

Environment : Public
TenantId    : 1c6ebc60-5ab4-42f0-be8c-2fc60286bd97
UserName    : Ricardo.Santos@y0phn.onmicrosoft.com

Report 'Adventure Works DW SQLBits 2024' exported to a local .pbix file with today's date.
Report 'Adventure Works DW SQLBits 2024' redeployed successfully to 'FabricBI Prod'.

PS C:\Users\RSANTOS>

```



FabricBI Prod

+ New ↕ Upload Create app

Name
Adventure Works DW SQLBits 2024
Adventure Works DW SQLBits 2024

Manage DevOps in PRO (Refresh Datasource)

```
        exit
    }

    # Get all reports in the workspace
    $reports = Get-PowerBIReport -WorkspaceId $workspace.Id

    if (!$reports) {
        Write-Host "No reports found in the workspace."
        Disconnect-PowerBIServiceAccount
        exit
    }

    # Create local folder if it doesn't exist
    if (-not (Test-Path $localFolderPath)) {
        New-Item -Path $localFolderPath -ItemType Directory | Out-Null
    }

    # Backup each report to .pbix file with today's date
    foreach ($report in $reports) {
        $localPbixFileName = $report.Name + "_" + (Get-Date -Format 'yyyyMMdd') + ".pbix"
        $localPbixFilePath = Join-Path -Path $localFolderPath -ChildPath $localPbixFileName
        if (Test-Path $localPbixFilePath) {
            Remove-Item $localPbixFilePath
        }
        Export-PowerBIReport -Id $report.Id -OutFile $localPbixFilePath -Format PBIX -Inc
        Write-Host "Report '$($report.Name)' backed up successfully to '$localPbixFileName'
    }

    # Disconnect from Power BI
    Disconnect-PowerBIServiceAccount
```

In this script:

- We define the parameters such as the workspace name and local folder path with today's date.
- We connect to Power BI using `Connect-PowerBIServiceAccount`.

Message ChatGPT...



ChatGPT can make mistakes. Consider checking important information.

Manage DevOps in PRO (Refresh Datasource)

```
5 # Connect to Power BI
6 Connect-PowerBIServiceAccount
7
8 # Get the workspace
9 $workspace = Get-PowerBIWorkspace -Name $workspaceName
10
11 if (!$workspace) {
12     Write-Host "Workspace '$workspaceName' not found."
13     Disconnect-PowerBIServiceAccount
14     exit
15 }
16
17 # Get the dataset
18 $dataset = Get-PowerBIDataset -WorkspaceId $workspace.Id -Name $datasetName
19
20 if (!$dataset) {
21     Write-Host "Dataset '$datasetName' not found in the workspace."
22     Disconnect-PowerBIServiceAccount
23     exit
24 }
25 $datasetRefresh = Get-PowerBIDataset -WorkspaceId $workspace.Id | ` 
26     Where {$_.Name -eq $datasetName} | ForEach {$_.Id}
27
28 $MailFailureNotify = @{"notifyOption"="MailOnFailure"}
29
30 # Refresh the dataset
```

```
PS C:\Users\RSANTOS> C:\Users\RSANTOS\OneDrive\Documentos\PBI\Fabric_Summit\PWRSHLL
Environment : Public
TenantId    : 1c6ebc60-5ab4-42f0-be8c-2fc60286bd97
UserName    : Ricardo.Santos@y0phn.onmicrosoft.com

Dataset 'AdventureWorksSQLBITS2024_PROD' refreshed successfully.

PS C:\Users\RSANTOS>
```

```
WARNING: The Body parameter was null, the request may be invalid when Method parameter is Post.
Error occurred during dataset refresh: Ocorreram um ou mais erros. Encountered errors when invoking the command:
{
    "code": "ItemNotFound",
    "message": "Dataset 3a783e4f-9b39-44a9-9c21-fa143049acbf 38b285e9-ab67-4c12-90ed-a3e91283cccd b7bcdf64-54a2-4ed7-9f26-14980463a3ed 2d26d3d9-e901-4ef2-be4b-20a3b597d24e 17a0edb8-07b4-45d9-92ba-4909e3cba856 ba3c3021-f577-47d4-a2ae-9aca4b159d19 is not found! please verify datasetId is correct and user have sufficient permissions."
}
```

```
Environment : Public
TenantId    : 1c6ebc60-5ab4-42f0-be8c-2fc60286bd97
UserName    : Ricardo.Santos@y0phn.onmicrosoft.com

Dataset 'AdventureWorksSQLBITS2024_PROD' refreshed successfully.
```



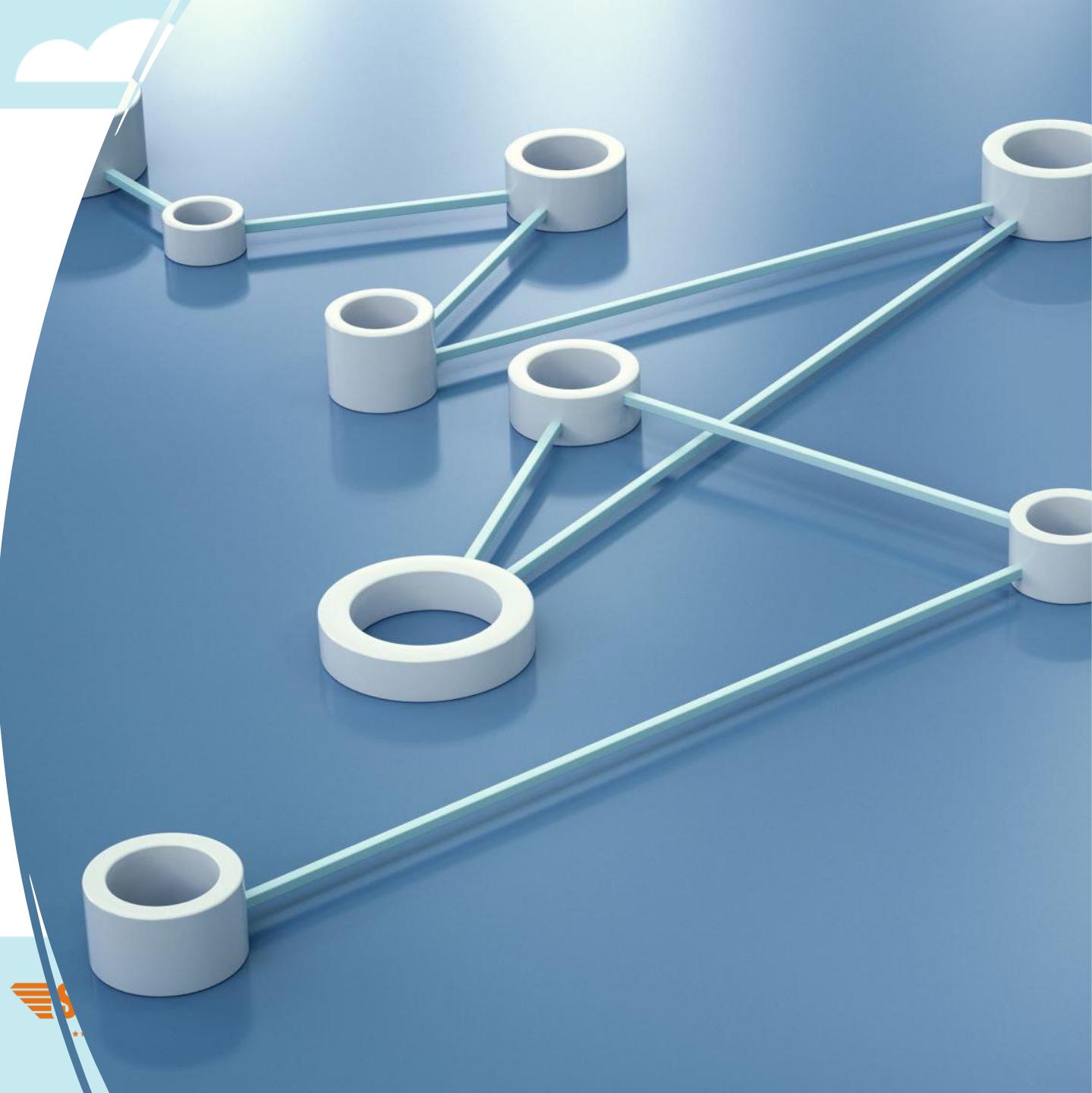


7) Key Takeaways / The Future

Summary Table

Key Takeaways

The Future



Results Summary

Test Area	Test #	Test Description	Test Result
Generate a Data Model	1.1	Creation of a new Data Model	OK
Generate Artificial Data	2.1	Generate CSV files	Not OK
	2.2	Generate Python Script	OK with Iterations
	2.3	Generate T-SQL Script	OK with iterations
Power Query	3.1	Generate Dim Date	OK
	3.2	Create Medium Complexity Custom Column	OK with Iterations
	3.3	Create Cumulative Function	OK with Iterations
	3.4	Translate M Script	OK
DAX	4.1	Generate Simple DAX Measures	OK
	4.2	Generate Sum for all numeric Columns	OK with Iterations
	4.2	Generate Date based Dax measure	OK with Iterations
	4.3	Generate complex measures	OK with Iterations
Report Theme & Imaging	5.1	Adjust Report Theme Colors	OK
	5.2	Adjust Report Theme Objects Code	OK
	5.3	Generate Images	Not OK
Cmdlets / PowerShell Scripts	7.1	List Power BI Workspace Artifacts	OK
	7.2	Publish to Workspace	OK with Iterations
	7.3	Republish to another Workspace	OK with Iterations
	7.4	Rebind Report	Not OK
	7.5	Backup Workspace	OK with Iterations
	7.6	Refresh Datasource	OK with Iterations

BCG Study Performance Distribution Results

Figure 2: Performance Distribution - Inside the Frontier

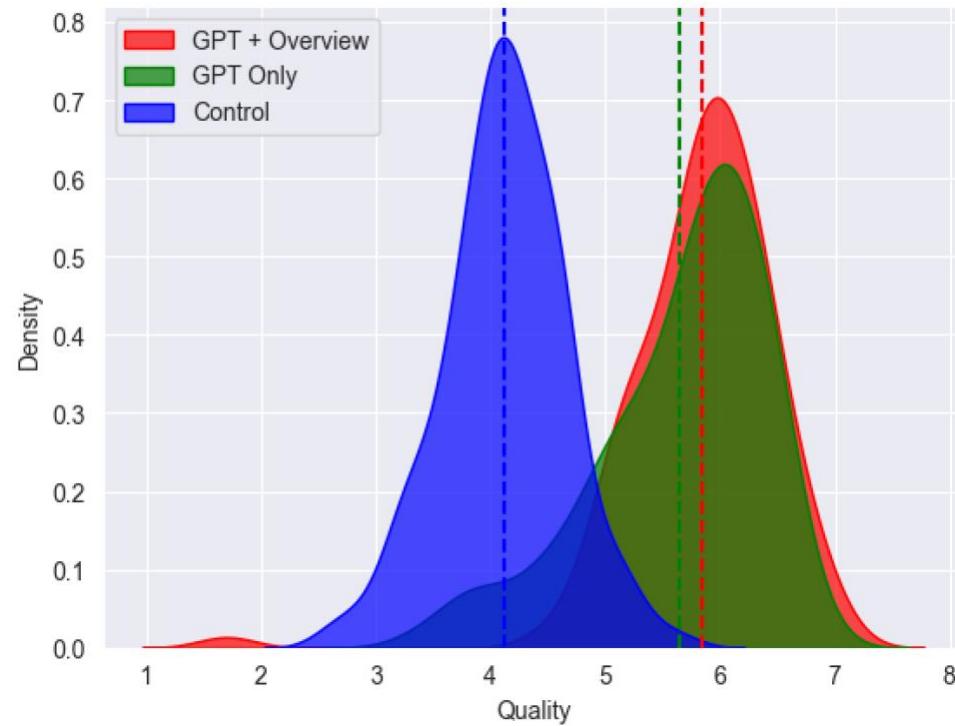
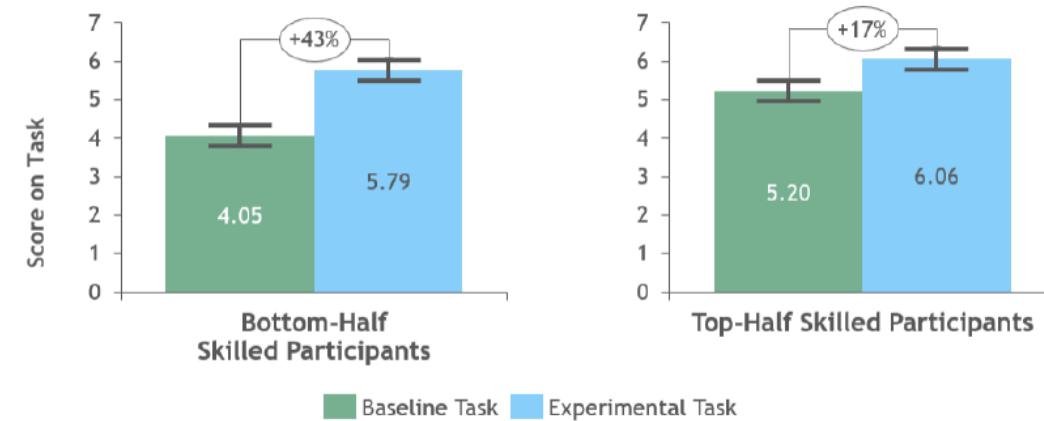


Figure 5: Bottom-Half Skills and Top-Half Skills - Inside the Frontier



Exam Scores by GPT 3.5 and GPT 4

Exam results (ordered by GPT-3.5 performance)

Estimated percentile lower bound (among test takers)

100% —

80% —

60% —

40% —

20% —

0% —

gpt-4 (no vision)
gpt-4
gpt3.5

AP Calculus BC
AMC 12
Codeforces Rating
AMC 10
AMC 10
Uniform Bar Exam
AP English Language
AP Chemistry
AP Physics 2
GRE Quantitative
USABO Semifinal 2020
AP Statistics
AP Macroeconomics
AP Microeconomics
GRE Writing
AP Biology
AP Economics
AP World History
SAT Math
AP US History
AP US Government
AP Psychology
AP Art History
AP Environmental Science

Exam



7) Key Takeaways / The Future

- Far from perfect but points in the correct direction and possible to get to a working version by iterating requests, output and errors.
- Works like a Junior level assistant, which confirms results of BCG study for Power BI also.
- Will tend to improve quickly (same way as test results improved from GPT 3.5 to GPT 4.0)
- PBIP Format should greatly enhance interaction capabilities
- Likely see it not only in DAX and Power Query, Report Design but also in Administration, UI/UX and much more





1) DAX with SVG and CSS

DAX with SVG

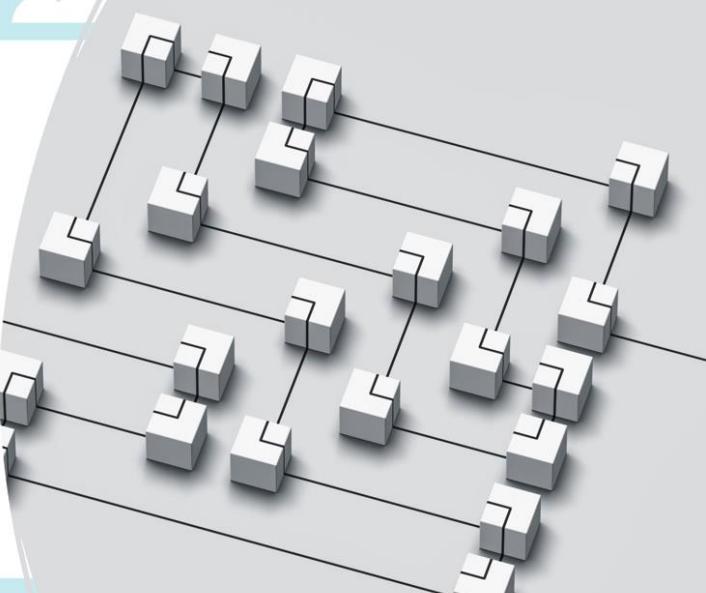
DAX with CSS on HTML visual



2) Data Modeling & Analysis

Analyze a Data Model

Analyze Data



3) Report Design Assessment

Analyze report design

Rate report design

Provide recommendations



4) Troublesh ooting

Troubleshooting Problem

Understanding errors

Performance Optimization



5) Knowledge

Knowledge Assessment

Knowledge Improvement



Feedback



<https://sqlb.it/?12594>

Connect with us!



Pedro Reis

Data & Information Leader •
Enterprise-Level Data Strategy & Tran...



pedroreis@reispn.onmicrosoft.com



Ricardo Santos

Expert Leader Big Data & BI @ Big
Data & Architecture @ Groupe BPCE |...



rjcsantos@hotmail.com