DP-500 Certification Study Guide

Data Loading And Transformation Using Power Query, Building And Optimizing Tabular Data Models In Power BI Using External Tools

Potential Problem Areas

- Slow data source
- Connector design
- Gateway installation
- Complex transformations
- Multiple queries pull data from same data source
- One query is referenced by more queries
- Steps that allow Query folding
- Removing , renaming and filtering using WHERE type functionality
- GROUP BY, JOIN, UNION ALL (Same source)
- Adding custom columns with simple logic
- Pivot and Unpivot operations
- Steps that do not allow Query folding
- Merging, Appending on different sources
- Adding custom columns w/ complex logic and adding Index Columns
- Settings in Power BI Desktop
- Disable data privacy firewall
- Disable background analysis

Troubleshooting through Query Diagonistics

Potential Improvements

1. Add proper indexes
2. Create persistent objects in databases
3. Remove un-necessary rows / columns
4. Use Query Folding
5. Use GROUP BY and summarize to raise the grain of Fact Table
6. Optimize column data types with preference for numeric data
7. Disable Power Query query load setting in PBI Desktop
8. Disable Auto Date/ Time in PBI Desktop
9. Preference for custom columns in PQ instead of calculated columns in model

Performance Analyzer

- DAX query
- Direct Query
- Visual display
- Evaluated parameters
- Others

DAX Studio

- DAX formatting
- Query plans
- Server Timings
- Vertipaq Analyzer
- FE and SE performance

Tabular Editor 2

- Create calculation groups
- Create perspectives
- Best Practice Analyzer
- Define / edit measures, format strings, detail row settings
- Add dataset roles for RLS and OLS

Potential Improvements

1. Create new calculation group in TE2
2. Add placeholder using SELECTEDMEASURE()
3. Create calculation items based on placeholder measure
4. Save the model in TE2
5. Use calculation groups in visuals
6. Precedence - Order of evaluation
7. Order property defines the sort order of calc. group

Create Aggregations

1. Original Fact table in DirectQuery
2. Data types of columns must match between original fact table and aggregated table
3. Aggregated tables are hidden by default
4. Aggregation Precedence defines which table to use if query can be satisfied from multiple aggregations
5. For RLS to work, relationships must be defined on aggregated tables as well

Incremental Refresh And Hybrid Tables

1. Data source should support query folding
2. Create RangeStart and RangeEnd parameters in PQ
3. Filter the date column of Fact table using these parameters
4. Set import and refresh ranges on the Fact table
5. If interested in real time, then select DQ or Hybrid Table
6. In Hybrid table, historical data is import partition and most recent data is in DQ partition
7. Hybrid table only works for Premium subscription
8. Hybrid tables cannot have calculated columns
9. TE2 can be used to create partitions in which historical data is DQ and most recent data is import partition

Row/ Object Level Security And Perspectives

- Row level Security (RLS)
- Add DAX table filters for roles in PBI Desktop
- Validate functionality of roles in PBI Desktop
- Role mapping and adding members to roles in PBI Service
- Object Level Security (OLS)
- Perform all steps for RLS
- Set value of column (s) or table (s) to None in TE2
- Create perspective in TE2 and measures to be included in perspective
- Use perspective with Personalized Visuals in PBI Desktop and Service

Choose proper storage mode for data modeling using Star schema

Import Model

- **Benefits**
- Supports all Power BI data source types
- Supports all M queries and DAX functions
- Supports calculated tables
- Best query performance due to caching
- **Limitations**
- Model Size, 1 GB -> Pro, 10 GB -> Premium
- Refresh, 8 / day -> Pro, 48 / day -> Premium
- Data sovereignty concerns

DirectQuery

- What to avoid
- Complex PQ transformations and DAX
- Relationships on calculated columns
- **Limitations**
- All Data Sources not supported
- All PQ transformations not supported
- Analytic performance dependent on source system

Composite Model

- It comprises more than one source group
- Allows using Dual storage mode
- All one-to-many intra source group relationships are regular
- All cross island relationships and many-to-many relationships are limited
- Common scenario to set dimensional tables in Dual storage mode for better performance

Data Modeling

Best Practices

1. Try making dimensions short and wide
2. Try making facts long and narrow
3. Avoid DAX calculated columns
4. Limit the use of bi-directional filtering in building relationships
5. Avoid many-to-many relationships

DAX programming

Best Practices

1. Use variables to improve readability
2. Use calculation groups

ENTERPRISE DNA

enterprisedna.co

Prepared By ABU BAKAR NISAR ALVI

DP-500 Certification Study Guide

Querying And Transforming Data Using Azure Synapse Analytics SQL Pools And Apache Spark Pool

Azure Data Lake Storage Gen 2

- **CSV files -> Comma Separated Values**
- **JSON files -> Key / value pairs**
- Both CSV and JSON suitable for storing data in Staging (Bronze) layer in raw form
- **Parquet files**
- Support both horizontal and vertical compression
- Suitable format to store data in Transform (Silver) and Data Model (Gold) layer

Apache Spark Pools

- Support Python, Scala, Java and SQL
- Dataframe is most commonly used data structure
- %% is a magic operator
- %%pyspark is used for Python, %% spark is used for Scala and %% SQL is used for SQL
- StructType and StructField constructs used to declare schema for dataframe in python
- createOrReplaceTempView used to create a temporary view for using SQL in Python script

Code Snippet For Loading Files In Python

```
%%pyspark
df = spark.read.load('abfss://adisdp500@adlsdp500.dfs.core.windows.net/
csv/*.csv' , format = 'csv')
```

Code Snippet For Loading Files In Scala

```
%%pyspark
val df = spark.read.format('csv').option("header" , "true"),
load("abfss://adlsdp500@adlsdp500.dfs.core.windows.net/csv/*.csv")
```

Code Snippet For Querying Top 100 Records From JSON

```
SELECT
  TOP 100 *
FROM
   OPENROWSET(
     BULK 'https://adlsdp500.dfs.core.windows.net/adlsdp500/json/**' ,
     FORMAT = 'CSV' ,
     FIELDTERMINATOR = '0x0b' ,
     FIELDQUOTE = '0x0b' ,
     ROWTERMINATOR = '0x0b'
   ) WITH
     (
        Doc NVARCHAR (MAX)
     )
AS result
```

Azure Synapse Serverless SQL Pools

- Non infrastructure needed, pay only for consumption
- Data is stored in ADLS and Scaling is automatic
- Ideal for data exploration and transformation on ADLS, CosmosDB, Dataverse and Logical DWH
- For querying files, OPENROWSET is most important function
- BULK parameter contains the URL to the filepath in ADLS
- FORMAT parameter specified the type of data being queried
- HEADER_ROW parameter instructs the query engine to use the first row of data as headers
- FIELDTERMINATOR specifies the character used to separate field values in a row. ROWTERMINATOR specifies the character used to signify the end of row of data
- WHERE clause used with filepath to query partitioned data
- OPENJSON returns JSON objects in Tabular form
- JSON_VALUE extracts a scalar value from JSON string
- JSON_QUERY extracts object or array from JSON string

Querying Top 100 Records Of CSV Files

```
SELECT
  TOP 100 *
FROM
   OPENROWSET(
     BULK 'https://adlsdp500.dfs.core.windows.net/adlsdp500/csv/**' ,
     FORMAT = 'CSV' ,
     PARSER-VERSION = '2.0'
   ) AS [ result ]
```

Querying Top 100 Records W/Schema

```
SELECT
  TOP 100 *
FROM
   OPENROWSET(
     BULK 'https://adlsdp500.dfs.core.windows.net/adlsdp500/csv/**' ,
     FORMAT = 'CSV' ,
     PARSER-VERSION = '2.0')
WITH
(
    SalesOrderNumber VARCHAR(10) COLLATE Latin1_General_100_BIN2_UTF8,
    SalesOrderLineNumber INT,
    OrderDate DATE,
    CustomerName VARCHAR(25) COLLATE Latin1_General_100_BIN2_UTF8,
    EmailAddress VARCHAR(50) COLLATE Latin1_General_100_BIN2_UTF8,
    Item VARCHAR(30) COLLATE Latin1_General_100_BIN2_UTF8,
    Quantity INT,
    UnitPrice DECIMAL(18,2),
    TaxAmount DECIMAL(18,2)
)
AS[result]
```

Azure Synapse Dedicated SQL Pools

- User is responsible for provisioning infrastructure and scaling
- User pays for both compute and storage
- Data is stored in internal data warehouse
- Ideal for querying that involves multiple concurrent users and for creating persistent database objects (tables, matrialized views)
- In a direct query scenario it is much better use a dedicated SQL pool than a serverless SQL pool
- For storing dimension tables, better to use replicated distribution for smaller sized tables and hash distribution for larger tables
- For storing fact tables, better to use hash distribution with columnstore index
- For staging tables, better to use round-robin distribution
- Dedicated pools do not support foreign key and unique constraints
- When loading new data into the data warehouse, it is good to rebuild the table columnstore indexes and update statistics

Creating External Data Source

```
CREATE EXTERNAL DATA SOURCE files
WITH
(
LOCATION = 'https://adlsdp500.dfs.core.windows.net/
adlsdp500/')
```

Creating External File Format

```
CREATE EXTERNAL FILE FORMAT CsvFormat
   WITH (
      FORMAT_TYPE = DELIMITEDTEXT,
      FORMAT_OPTIONS(
         FIELD_TERMINATOR = ' , ' ,
         STRING_DELIMITER ' " '
      )
   );
GO
```

Creating External Table

```
CREATE EXERTNAL TABLE dbo.orders
(
SalesOrderNumber VARCHAR(10),
SalesOrderLineNumber INT,
OrderDate DATE,
CustomerName VARCHAR(25),
EmailAddress VARCHAR(50),
Item VARCHAR(30),
Quantity INT,
UnitPrice DECIMAL(18,2),
TaxAmount DECIMAL(18,2)
)
WITH
(
   DATA_SOURCE = files,
   LOCATION = 'csv/*.*',
   FILE_FORMAT = CsvFormat
);
GO
-- query the table
SELECT * FROM dbo.orders;
```

Using SQL PREDICT Function

- Use VARBINARY(MAX) when storing the model
- Model can only be stored in ONNX format which is a hexadecimal string
- DATA parameter is specified in a table format which can be a table, table alias, CTE alias, view or table-valued function
- PREDICT generates a predicted value or scores based on the stored model

Selecting Data And Predicting Scores

```
SELECT d.* , p.Score
FROM
PREDICT
(
MODEL = @model,
DATA = dbo.mytable AS d
)
WITH (Score FLOAT) AS p;

DECLARE @model VARBINARY(max) =
(SELECT model FROM scoring_model WHERE
model_name = 'ScoringModelV1');

INSERT INTO loan_applications (c1, c2, c3, c4, score)
SELECT d.c1, d.c2, d.c4, p.score
FROM
PREDICT
(
MODEL = @model,
DATA = dbo.mytable AS d
)
WITH (score FLOAT) AS p;)
```

DP-500 Certification Study Guide

Querying And Transforming Data Using Azure Synapse Analytics
SQL Pools And Apache Spark Pool

Native Visuals In Spark Notebooks

- The **display function** allows to turn SQL queries and Apache Spark dataframes and RDDs into rich data visualizations.
- The **display function** can be used on dataframes or RDDs created in PySpark, Scala, Java, R, and .NET.
- We can use *display(df,summary = true)* to check the statistics summary of a given Apache Spark DataFrame

Azure Synapse SQL Results Pane

- Search capability for number and text
- Export results in CSV, JSON, XML and image format
- Charting options to plot line, scatter and column charts with labels for category, data and legends

Code Snippet For Dataframes

Converting RDD to DF

```
val df = rdd.to.DF()
```

Creating DF from CSV file

```
csvDf = spark.read.csv("/path/to/file.csv")
```

Creating DF from JSON

```
jsonDf = spark.read.json("/path/to/file.json")
```

Creating DF from schema

```
data = [("Adam","Smith","Male","CA"),
("Brenda","Jones","Female","FL")]
schema = ["firstname", "lastname", "gender", "state"]
df = spark.createDataFrame(data = data, schema = schema)
```

Write DF to CSV

```
DP500_df.write.csv('DP500_df', mode = 'overwrite')
```

Write DF to Parquet

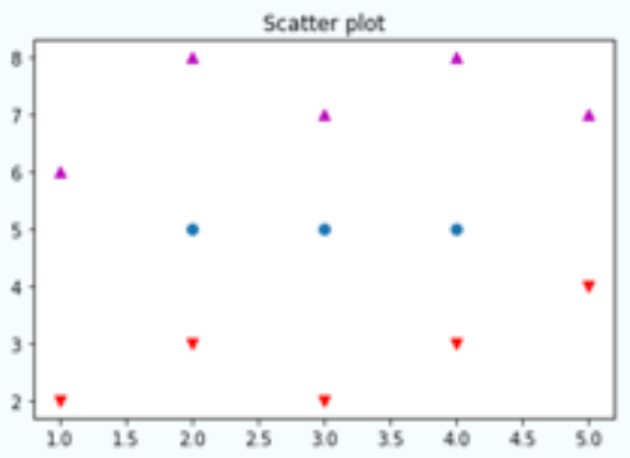
```
DP500_df.write.parquet('abfss://<<StorageAccountFileSystem>>@
<<StorageAccount>>.dfs.core.windows.net/DP500/DP500_df',
mode='overwrite')
```

Write DF to View

```
DP500_df.createOrReplaceTempView('DP500_df')
```

Scatter plot

```
Import matplotlib.pyplot as plt
x1 = [2,3,4]
y1 = [5,5,5]
x2 = [1,2,3,4,5]
y2 = [2,3,2,3,4]
y3 = [6,8,7,8,7]
plt.scatter(x1,y1)
plt.scatter(x2,y2,marker = 'v' , color = 'r')
plt.scatter(x2,y3,marker = '^', color ='m')
plt.title("Scatter plot")
plt.show()
```



Create And Import A Custom Theme

- Custom colors do not stick when a report theme is changed
- Foreground and FirstLevelAccent are the same
- .PBIT file is a better way to share the custom themes

Personalized Visual In A Report Or Dashboard

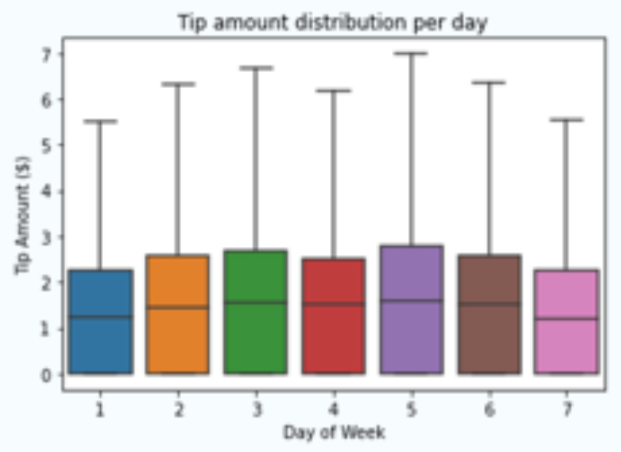
- This option is only for report viewers and requires a Pro or Premium license
- Option can be enabled directly from the Power BI Desktop by the developer
- To Keep personalized visuals persistent, personal bookmarks and can be used in Power BI Service

Automatic Page Refresh

- Supported only in DirectQuery storage mode, or Mixed mode containing at least one DirectQuery data source
- Fixed Interval -> set the desired interval (ranging from 1 second to X days)
- Change detection works only in Premium
- Change direction works on only one measure per dataset

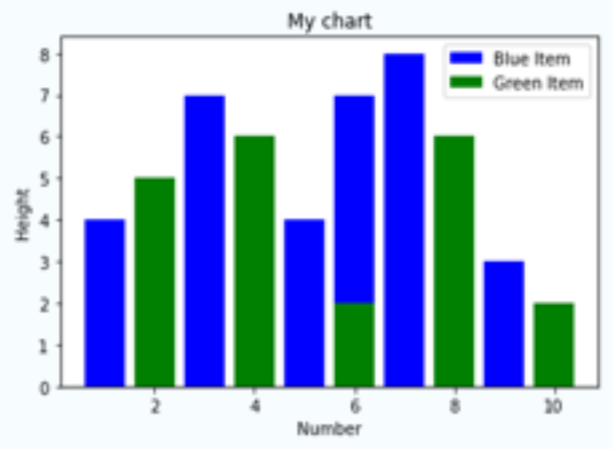
Boxplot

```
ax = sns.boxplot(x="day of week" , y = "tipAmount"
, data = sampled_taxi_pd_df, showfliers = False)
ax.set_title('Tip amount distribution per day')
ax.set_xlabel('Day of Week')
ax.set_ylabel('Tip Amount ($)')
plt.show()
```



Bar chart (stacked)

```
x1 = [1,3,4,5,6,7,9]
y1 = [4,7,2,4,7,8,3]
x2 = [2,4,6,8,10]
y2 = [5,6,2,6,2]
plt.bar(x1, y1, label = "Blue Item" , color = 'b')
plt.bar(x2, y2, label = "Green Item", color = 'g')
plt.plot()
plt.xlabel("Number")
plt.ylabel("Height")
plt.title("My chart")
plt.legend()
plt.show()
```



Using R And Python Custom Visuals

- Cannot use more than 150.000 rows for plotting
- Require a Pro or PPU license to render in reports
- Cannot be used as source for cross-filtering and highlighting

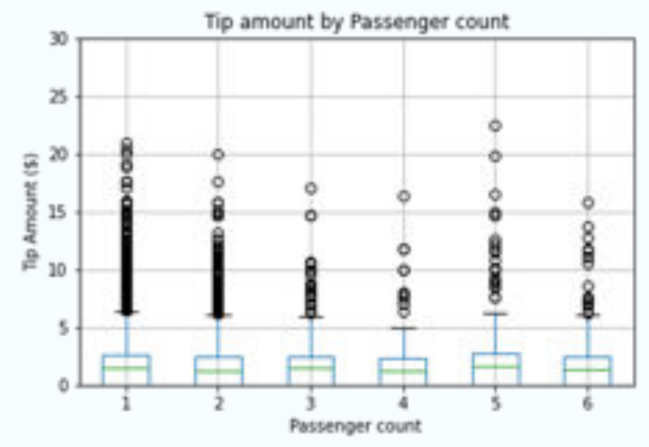
Power BI Reports For Accessibility

- Ensure that color contrast between the report elements is at least 4.5:1 as per WCAG standard
- Supplement color with texts or icons
- Avoid using tooltips as a method for transmitting important information

XMLA Endpoints For Connecting And Querying Datasets

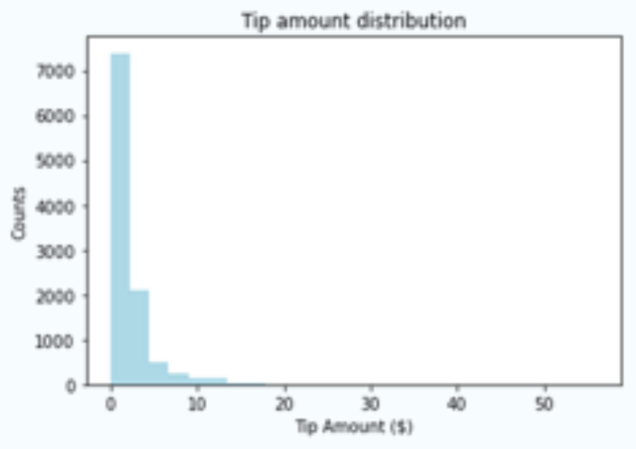
- XMLA Read/Write requires some kind of Premium license
- Power BI Dataset in workspace is a database in the Analysis Services instance
- Use the connection string of the dataset to connect with it through external tool like Tabular Editor or SSMS
- Once changes have been made to the dataset through the XMLA endpoint, the dataset cannot be used to download the PBIX file

```
ax2 = sampled_taxi_pd_df.boxplot(column =
['tipAmount'] ,
by = ['passengerCount']
ax2.set_title('Tip amount by Passenger count')
ax2.set_xlabel('Passenger count')
ax2.set_ylabel('Tip Amount ($)')
ax2.set_ylim(0,30)
plt.suptitle('')
plt.show()
```



Histogram

```
ax1 = sampled_taxi_pd_df['tipAmount'].plot(kind = 'hist' ,
bins = 25, facecolor = 'lightblue')
ax1.set_title('Tip amount distribution')
ax1.set_xlabel('Tip Amount ($)')
ax1.set_ylabel('Counts')
plt.suptitle('')
plt.show()
```



Paginated Reports In Power BI

- Paginated Reports are created in Power BI Report Builder
- Paginated Report requires Premium license, lately added to Pro as well
- Paginated Report visual in Power BI works in DirectQuery mode
- Paginated reports are idela for creating pixelperfect Sales invoices, receipts etc.

ENTERPRISE DNA

enterprisedna.co

Prepared By **ABU BAKAR NISAR ALVI**

DP-500 Certification Study Guide

Querying And Transforming Data Using Azure Synapse Analytics SQL Pools And Apache Spark Pool

Roles In Microsoft Purview

- Purview **Data Reader Role**: Has access to the Microsoft Purview governance portal and can read all content in Microsoft Purview except for scan bindings.
- Purview **Data Source Administrator Role**: Doesn't have access to the Microsoft Purview governance portal. Can manage all aspects of scanning data into Microsoft Purview, but doesn't have read or write access to content in Microsoft Purview beyond those tasks related to scanning.
- Purview **Data Curator Role**: Has access to the Microsoft Purview governance portal and can read all content in Microsoft Purview except for scan bindings. Can edit information about assets, can edit classification definitions and glossary terms, and can apply classifications and glossary terms to assets

Microsoft Purview - Power BI Integration

- To perform a scan, the user must be both a Data Source Administrator and a Data Reader in Microsoft Purview AND the user must be a Power BI admin.
- Step 1 : Create a Microsoft Purview account (same tenant)
- Step 2 : Register the Power BI tenant in Microsoft Purview
- Step 3 : From Azure portal, validate if Microsoft Purview account Network is set to public access
- Step 4 : Authenticate to Power BI tenant and create a new security group in Azure Active Directory
- Step 5 : Configure Power BI tenant and associate the security group with Power BI tenant
- Step 6 : Create scan for same-tenant Power BI using Azure IR and Managed Identity

Microsoft Purview - Azure Synapse Analytics Integration

- Step 1 : Create a Microsoft Purview account (same tenant)
- Step 2 : Configure data access for Microsoft Purview, Purview requires Reader role for the Azure Synapse Workspace resource in the Azure subscription. Purview requires membership of the Storage Blob Data Reader role for the Azure Storage account hosting the Azure Data Lake Storage Gen2 container. Purview requires membership of the db_datareader fixed database role in each database
- Step 3 : Register and scan data sources
- Data Map = Data assets + Lineage + Classifications + Business Context
- Collections are a way of grouping data assets into logical categories (collections) to simplify management and discovery of assets within the catalog
- Classifying data is important to understand its level of confidentiality,
- Custom classifications can also be defined
- Government - covers attributes such as government identity cards, driver license numbers, passport numbers,
- Financial - covers attributes such as bank account numbers or credit card numbers
- Personal - personal information such as a person's age, date of birth, email address, phone number etc
- Security - attributes like passwords that may be stored
- Miscellaneous - attributes not covered in the other categories

Power BI Licensing Options

Free license	Pro license	Premium per user
User cannot share or distribute content with other users	Can leverage all content creation and distribution features (primary choice for PBI report developers)	Can leverage all content creation, distribution and premium features
Capabilities - Access to My Workspace only - Use Publish to Web - Export to Power point, Excel, CSV - Cannot consume content created by Pro or PPU licenses*	**Capabilities** - All features of the free tier plus - Workspaces - All sharing options - Publish Apps - Email subscriptions - Analyze in Excel - Usage of Dataflows, Paginated reports, Shared Datasets, monitoring and some governance features - Model size limit is 1 GB - Scheduled refresh up to 8 times / 24 hrs	**Capabilities** - All features of the Pro license plus - Model size limit is 100 GB - Scheduled refresh up to 48 times / 24 hrs - Full features of dataflows (enhanced compute engine, computed entity) - XMLA endpoint Read / Write - AI functionalities in Power BI - Datamarts - Enhanced Automatic page refresh - Large storage format dataset - Deployment pipelines

Power BI Gateway Options

Features	Personal Mode	Standard Mode	VNet
Numbers of users supported	One	Multiple	Multiple
Managed by	Customer (User)	Customer (Admin)	Microsoft
Support of Data Refresh	Yes	Yes	Yes
Support for Azure Active Directory (Single Sign-On) DirectQuery PBI dataset	No	No	Yes
Support for standard DirectQuery + source Single Sign-On	No	Yes	Yes
Live Connection	No	Yes	Yes
Python & R visuals	Yes	No	No

Monitoring and Audit
Usage metrics tenant
Usage metric reports
Audit log
Activity log
REST APIs / PowerShell cmdlets

PowerShell Cmdlet
.Name MicrosoftPowerBIMgmt
Connect.PowerBIServiceAccount
Get-PowerBIWorkspace
New-PowerBIWorkspace
Get-PowerBIGroup
New-PowerBIReport
Remove-PowerBIReport
Invoke-PowerBIRestMethod

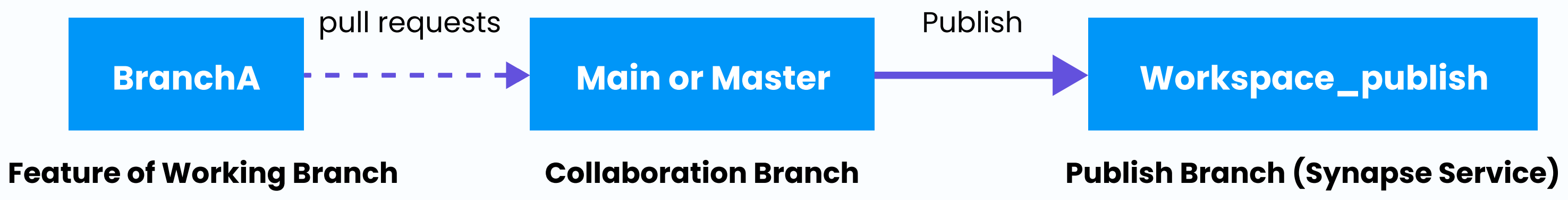
Power BI Dataflows

Premium Features
Enhanced Compute Engine
DirectQuery connectivity
Computed Entities
Linked Entities
Incremental Refresh

Power BI - Azure Data Lake Storage Gen 2 Integration

- **Pre-requisites for ADLS Gen 2 Storage Account**
- Must have Owner permission at the storage account layer. Administrator must assign himself owner permission
- The storage account must be created with the Hierarchical Namespace (HNS) enabled
- The storage account must be created in the same Azure Active Directory tenant as the Power BI tenant
- The user must have Storage Blob Data Owner role, Storage Blob Data Reader role, and an Owner role at the storage account level (scope should be this resource and not inherited)
- **Pre-requisites for Power BI**
- The Power BI workspace tenant region should be the same as the storage account region
- Connecting ADLS Gen 2 with Power BI tenant -> Only Power BI admin can do
- Connecting ADLS Gen 2 with workspace -> Only Workspace admin can do
- Connecting ADLS Gen 2 with workspace -> No dataflows in the workspace before connecting
- Create Power BI dataflows writing back to connected ADLS account -> Min. Workspace Contributor role
- Consume Power BI dataflow -> Min. Workspace Viewer role

Source Control Repository In Azure Synapse Analytics


BranchA
pull requests
Main or Master
Publish
Workspace_publish
Feature of Working Branch
Collaboration Branch
Publish Branch (Synapse Service)

Deployment Pipelines In Power BI

- Must have Premium license & must be admin of workspace.
- Step 1: Create a deployment pipeline
- Step 2: Assign a workspace
- Step 3: Create deployment rules
- Step 4: Deploy content from one stage to another
- Auto-binding across pipelines automatically recognizes connections between items across different pipelines based on their dependencies

Miscellaneous

- For version control of .PBIX -> OneDrive, Sharepoint, Git
- For version control of .BIM -> Git
- Impact analysis on external datasets cannot be performed in existing workspace
- Promoted datasets are endorsed by content owners, while certified datasets are endorsed by authorized reviewers
- The status code of a REST API call in Power BI Desktop by using Web.Contents function with Headers option
- Interactive operations are prioritized over background operations when deciding on dataset eviction in PBI Capacity

Prepared By ABU BAKAR NISAR ALVI