PASS SQLSATURDAY

Move forward your data strategy

Andrea Benedetti, Microsoft

Own your career with interactive learning built by community and guided by data experts. Get involved. Get ahead.

Explore your PASS community

PASS MARATHON

Free online webinar events

PASS LOCAL GROUPS

Local user groups around the world

PASS SUMMIT

Connect with the global data community

PASS VIRTUAL GROUPS

Online special interest user groups

PASS.org

PASS CONNECTOR INSIGHTS

Learning on-demand and delivered to you

PASS VOLUNTEERS

Get involved

Missed PASS Summit 2019?

Get the Recordings

Download all PASS Summit sessions on Data Management, Analytics, or Architecture for only $399 USD

More options available at
PASSstuff.com

We are covering all bases to ensure our community can continue reaching new and exciting heights. Plans are underway for the in-person event you all know and love along with a new venture, a new opportunity: a PASS Summit 2020 Virtual Event.

Find out more at PASS.org/summit

PASS SQLSATURDAY

Thank you to our Global Sponsors and Supporters

Microsoft Azure

DELL Technologies

Google Cloud

IDERA

Quest

SentryOne

vmware

VIRTUAL PASS SQLSATURDAY PORDENONE | 30 MAY 2020

PASS SQLSATURDAY

Thank you to our Local Sponsors and Supporters

Microsoft

This event was sponsored by Microsoft

Learn more about SQL Server 2019 today:

-Get free training: aka.ms/sqlworkshops

-Download the SQL19 eBook: aka.ms/sql19_ebook

Andrea Benedetti

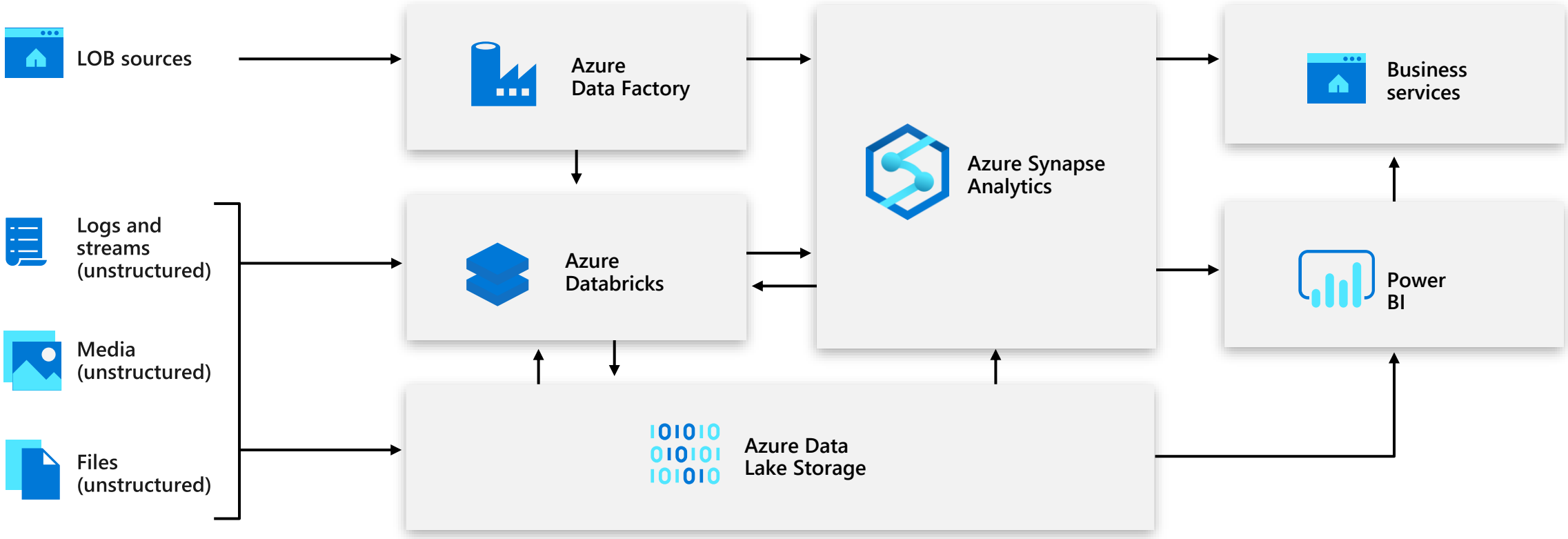
https://twitter.com/anBenedetti

https://github.com/anbened

https://www.linkedin.com/in/abenedetti/

Andrea Benedetti

The modern data warehouse



Azure Synapse Analytics

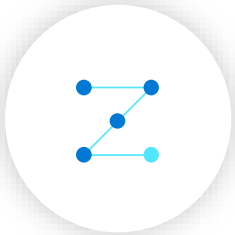
Limitless analytics service with unmatched time to insight

Limitless Scale

Industry-leading performance on demanding workloads

Powerful Insights

Integration with Power BI and Azure Machine Learning



Unified Experience

Data prep and management, big data, and AI in a single workspace

Unmatched Security

Fine-grained access controls and layers of protection

Power BI

Import

Great for small data sources and personal data discovery

Fine for CSV files, spreadsheet data and summarized OLTP data

DirectQuery

The enterprise solution

Avoid data movement

Delegate query work to the back-end source; take advantage of Azure Synapse's advanced features

Composite Models & Aggregation Tables

Why choose? Import and DirectQuery in a single model

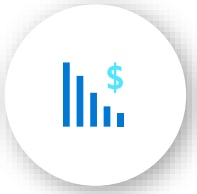
Keep summarized data local; get detail data from the source

Power BI + Azure Synapse Analytics

Industry leading combination for Enterprise Business Intelligence in the Cloud

Azure Synapse

Performance



Security

Workload Management

Power BI

In-Memory

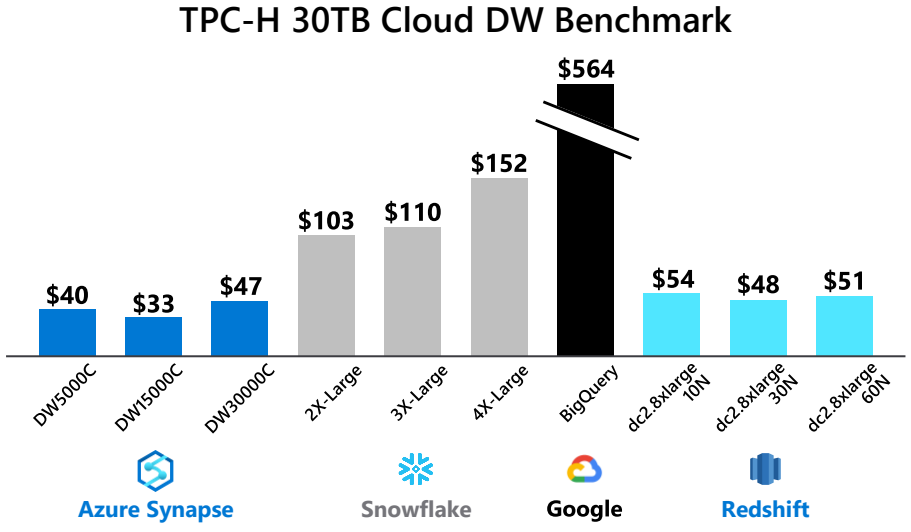
DirectQuery

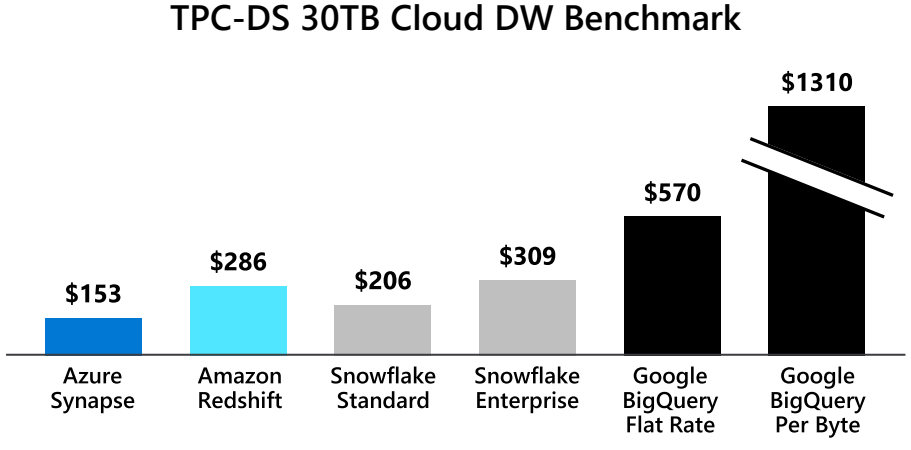
Composite Models

Azure Synapse performance advantage

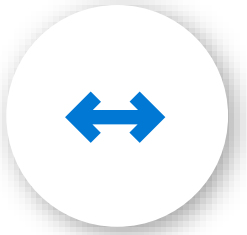
Overview

- Azure Synapse's industry leading price-performance comes from leveraging the Azure ecosystem and core SQL Server engine improvements to produce massive gains in performance.
- These benefits require no customer configuration and are provided out-of-the-box for every data warehouse
  - **Gen2 adaptive caching** – using non-volatile memory solid-state drives (NVMe) to increase the I/O bandwidth available to queries
  - **Azure FPGA-accelerated networking enhancements** – to move data at rates of up to 1GB/sec per node
  - **Instant data movement** – leverages multi-core parallelism in underlying SQL Servers to move data efficiently between compute nodes
  - **Query Optimization** – ongoing investments in distributed query optimization





Performance Optimized Storage



Elastic Architecture

Columnar Storage

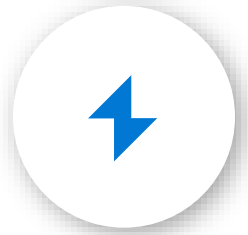
Columnar Ordering

Table Partitioning

Secondary Indexes

Hash Distribution

Materialized Views



Resultset Cache

Scaling to Petabytes

Materialized Views

- Transactional consistently to data modification
- Automatic Query Optimizer matching

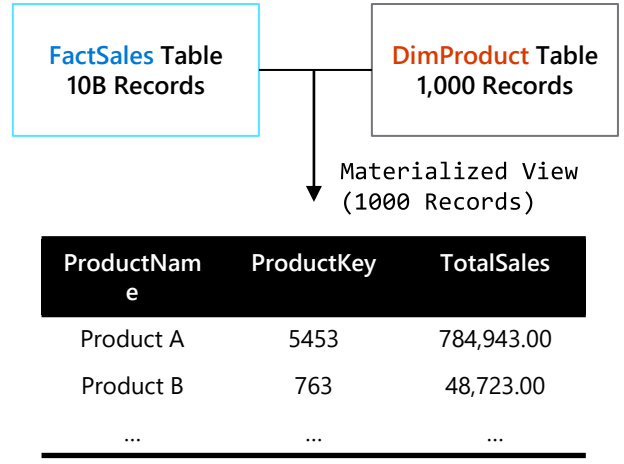
```
CREATE MATERIALZIED VIEW vw_ProductSales
WITH (DISTRIBUTION = HASH(ProductKey))
AS
SELECT
  ProductName
  ProductKey,
  SUM(Amount) AS TotalSales
FROM
  FactSales fs
INNER JOIN DimProduct dp ON fs.prodkey =
dp.prodkey
GROUP BY
  ProductName,
  ProductKey
```

Scaling to Petabytes

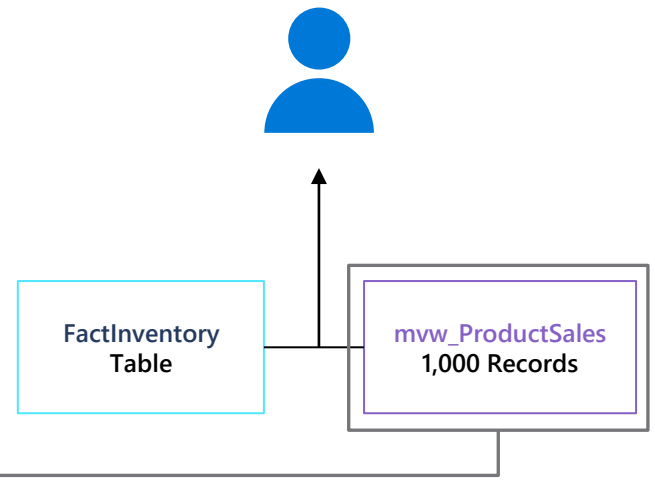
Materialized Views

- Transactional consistently to data modification
- Automatic Query Optimizer matching

```
CREATE MATERIALZIED VIEW mvw_ProductSales
WITH (DISTRIBUTION = HASH(ProductKey))
AS
SELECT
  ProductName
  ProductKey,
  SUM(Amount) AS TotalSales
FROM
  FactSales fs
INNER JOIN DimProduct dp
  ON fs.prodkey = dp.prodkey
GROUP BY
  ProductName,
  ProductKey
```




```
SELECT
  <COLUMNS>
FROM FactSales fs
INNER JOIN
(
  SELECT
    ProductName
    ProductKey,
    SUM(Amount) AS TotalSales
  FROM
    FactSales fs
  INNER JOIN DimProduct dp
  GROUP BY
    ProductName,
    ProductKey ) ps
INNER JOIN FactInventory
GROUP BY …
```

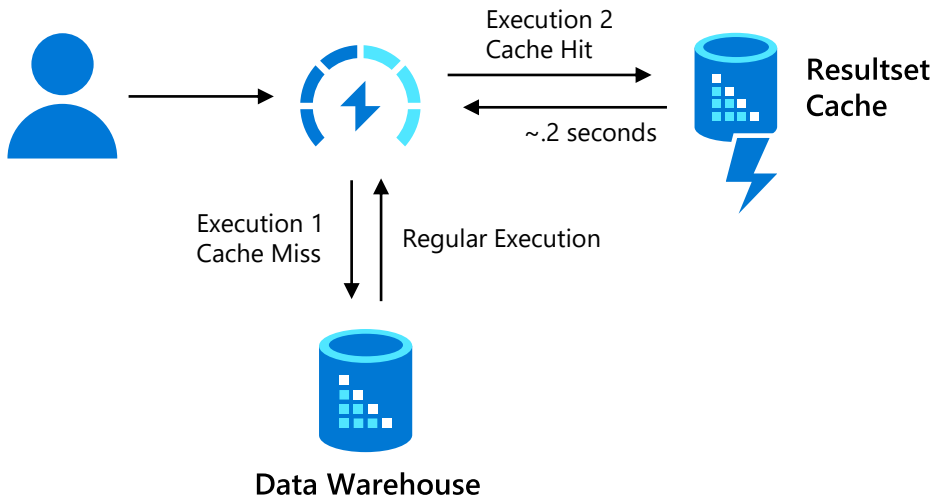


Scaling to Petabytes

Result set cache

- Automatic query matching
- Implicit creating from query activity
- Resilient to cluster elasticity

```
SELECT
  ProductName
  ProductKey,
  SUM(Amount) AS TotalSales
FROM
  Fact Sales
INNER JOIN DimProduct
GROUP BY
  ProductName,
  ProductKey
```



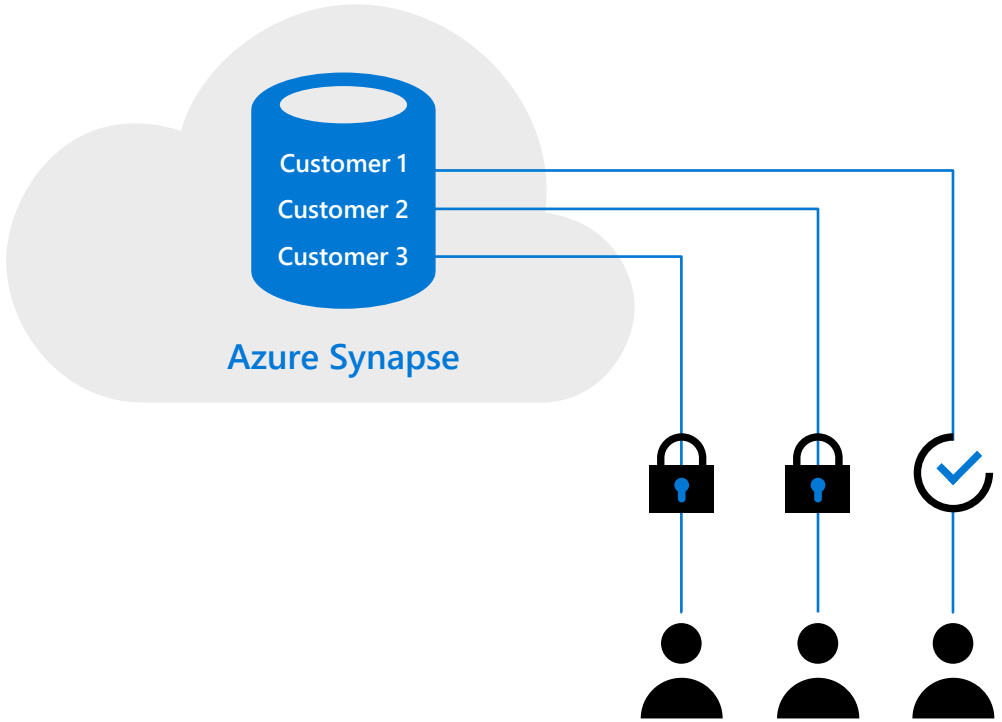
Complete security

Category	Feature	Azure Synapse
Data Protection	Data In Transit	**Yes**
	Data encryption at rest (Service & User Managed Keys)	**Yes**
	Data Discovery and Classification	**Yes**
Access Control	Native Row Level Security	**Yes**
	Table and View Security (GRANT / DENY)	**Yes**
	Column Level Security	**Yes**
	Dynamic Data Masking	**Yes**
Authentication	SQL Authentication	**Yes**
	Native Azure Active Directory	**Yes**
	Integrated Security	**Yes**
	Multi-Factor Authentication	**Yes**
Network Security	Virtual Network (VNET)	**Yes**
	SQL Firewall (server)	**Yes**
	Integration with ExpressRoute	**Yes**
Threat Protection	SQL Threat Detection	**Yes**
	SQL Auditing	**Yes**
	Vulnerability Assessment	**Yes**

Row-level security (RLS)

Overview

- Fine grained access control of specific rows in a database table.
- Help prevent unauthorized access when multiple users share the same tables.
- Eliminates need to implement connection filtering in multi-tenant applications.
- Administer via SQL Server Management Studio or SQL Server Data Tools.
- Easily locate enforcement logic inside the database and schema bound to the table.



Row-level security

Creating policies

- Filter predicates silently filter the rows available to read operations (SELECT, UPDATE, and DELETE).
- The following examples demonstrate the use of the CREATE SECURITY POLICY syntax

```
-- The following syntax creates a security policy with a filter
predicate for the Customer table
CREATE SECURITY POLICY [FederatedSecurityPolicy]
ADD FILTER PREDICATE [rls].[fn_securitypredicate]([CustomerId])
ON [dbo].[Customer];


-- Create a new schema and predicate function, which will use
the application user ID stored in CONTEXT_INFO to filter rows.
CREATE FUNCTION rls.fn_securitypredicate (@AppUserId int)
RETURNS TABLE
WITH SCHEMABINDING
AS
RETURN (
SELECT 1 AS fn_securitypredicate_result
WHERE
DATABASE_PRINCIPAL_ID() = DATABASE_PRINCIPAL_ID('dbo') --
application context
AND CONTEXT_INFO() = CONVERT(VARBINARY(128), @AppUserId));
GO
```

Column-level security

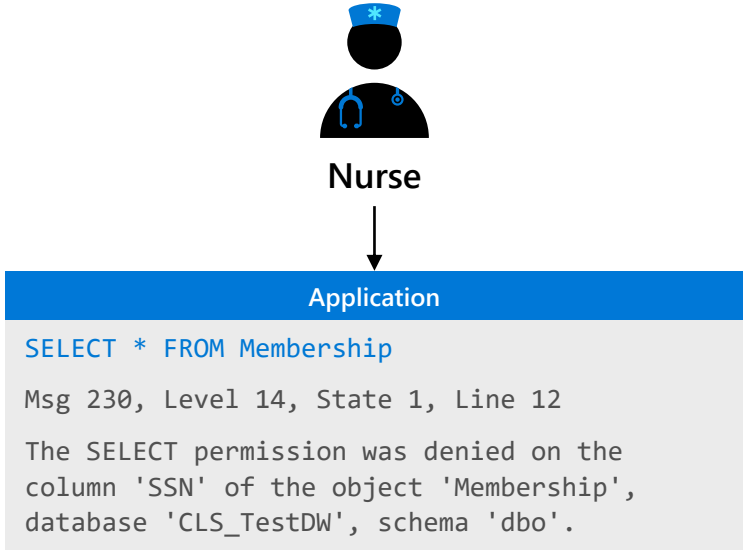
Overview

- Control access of specific columns in a database table based on customer's group membership or execution context.
- Simplifies the design and implementation of security by putting restriction logic in database tier as opposed to application tier.
- Administer via GRANT T-SQL statement.
- Both Azure Active Directory (AAD) and SQL authentication are supported.

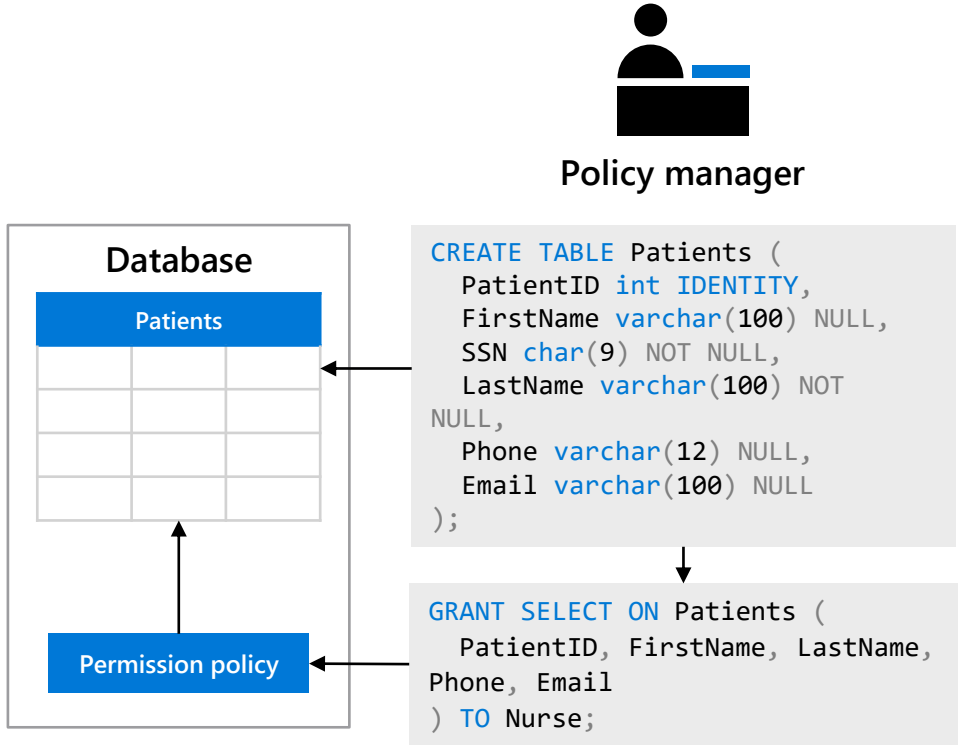
Column-level security

Three steps

1. Policy manager creates permission policy in T-SQL, binding the policy to the Patients table on a specific group.
2. App user (for example, a nurse) selects from Patients table.
3. Permission policy prevents access on sensitive data.



Queries executed as 'Nurse' will fail if they include the SSN column

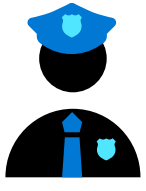


Allow 'Nurse' to access all columns except for sensitive SSN column

Dynamic Data Masking

Three steps

1. Security officer defines dynamic data masking policy in T-SQL over sensitive data in the Employee table. The security officer uses the built-in masking functions (default, email, random)
2. The app-user selects from the Employee table
3. The dynamic data masking policy obfuscates the sensitive data in the query results for non-privileged users



Security officer

```
ALTER TABLE [Employee]
ALTER COLUMN [SocialSecurityNumber]
ADD MASKED WITH (FUNCTION = 'DEFAULT()')

ALTER TABLE [Employee]
ALTER COLUMN [Email]
ADD MASKED WITH (FUNCTION = 'EMAIL()')

ALTER TABLE [Employee]
ALTER COLUMN [Salary]
ADD MASKED WITH (FUNCTION = 'RANDOM(1,20000)')

GRANT UNMASK to admin1
```

1

Business app

```
SELECT [First Name],
       [Social Security Number],
       [Email],
       [Salary]
FROM   [Employee]
```

2

Non-masked data (admin login)

	First Name	Social Security Num...	Email	Salary
1	LILA	758-10-9637	lila.barnett@comcast.net	1012794
2	JAMIE	113-29-4314	jamie.brown@ntlworld.com	1025713
3	SHELLEY	550-72-2028	shelley.lynn@charter.net	1040131
4	MARCELLA	903-94-5665	marcella.estrada@comcast.net	1040753
5	GILBERT	376-79-4787	gilbert.juarez@verizon.net	1041308

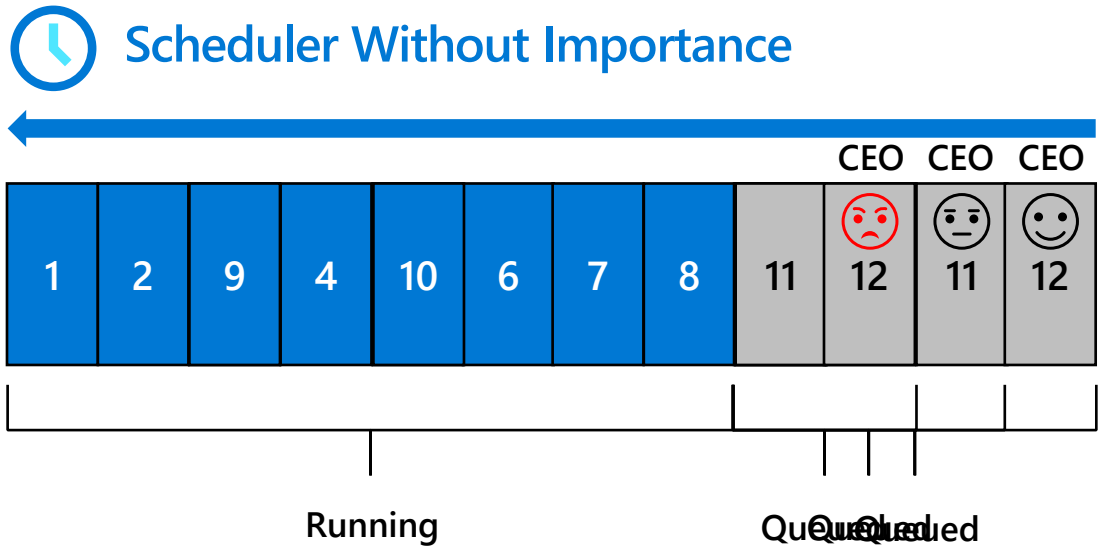
Masked data (admin1 login)

	First Name	Social Security Number	Email	Salary
1	LILA	XXX-XX-XX37	lXX@XXXX.net	8940
2	JAMIE	XXX-XX-XX14	jXX@XXXX.com	19582
3	SHELLEY	XXX-XX-XX28	sXX@XXXX.net	3713
4	MARCELLA	XXX-XX-XX65	mXX@XXXX.net	11572
5	GILBERT	XXX-XX-XX87	gXX@XXXX.net	4487

3

Workload Management

Workload Importance

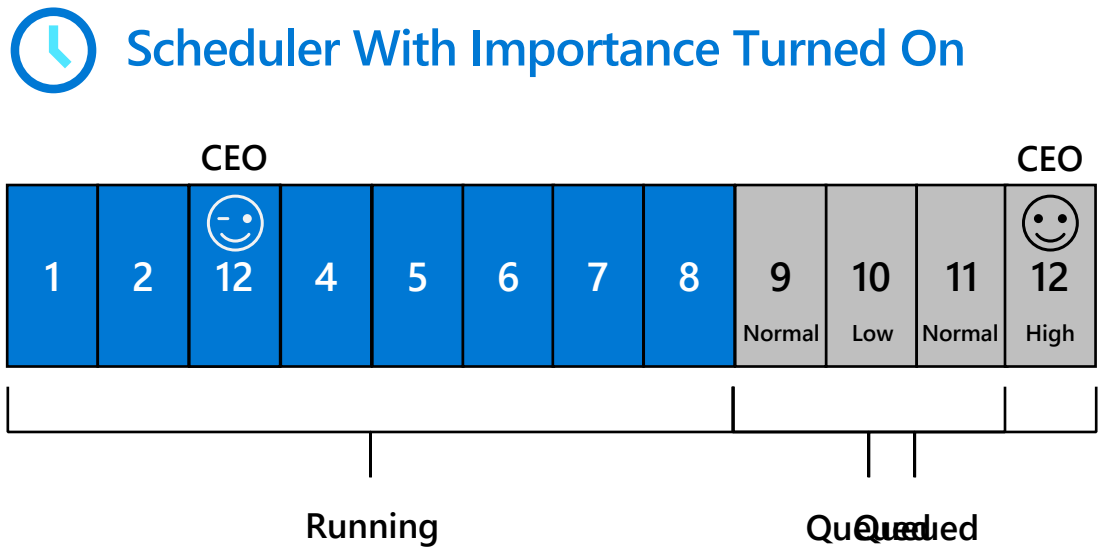


By default, workloads are run on a first-in first-out basis.

Workload Management

Workload Importance

```
CREATE WORKLOAD CLASSIFIER classifier_name
WITH
(
   WORKLOAD_GROUP = 'name' ,
   MEMBERNAME = 'security_account' [ [ , ]
   IMPORTANCE = { LOW | BELOW_NORMAL | NORMAL (default) | ABOVE_NORMAL | HIGH
}])
```



Power BI

Import

Great for small data sources and personal data discovery

Fine for CSV files, spreadsheet data and summarized OLTP data

DirectQuery

The enterprise solution

Avoid data movement

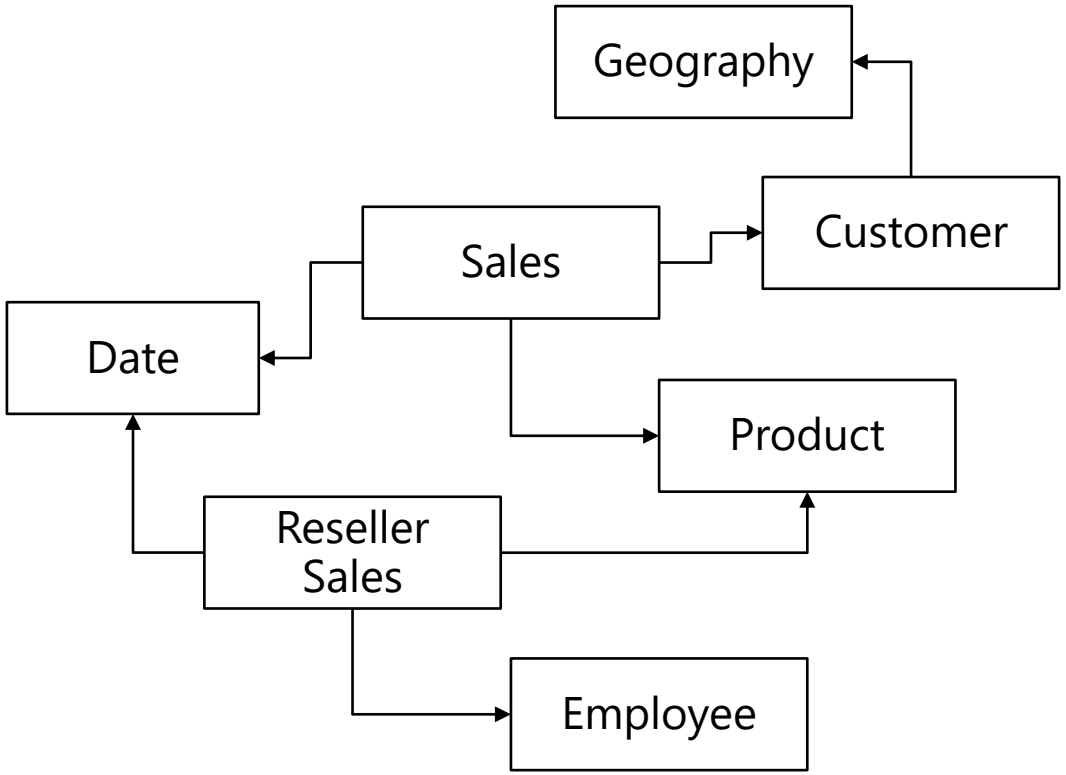
Delegate query work to the back-end source; take advantage of Azure Synapse's advanced features

Composite Models & Aggregation Tables

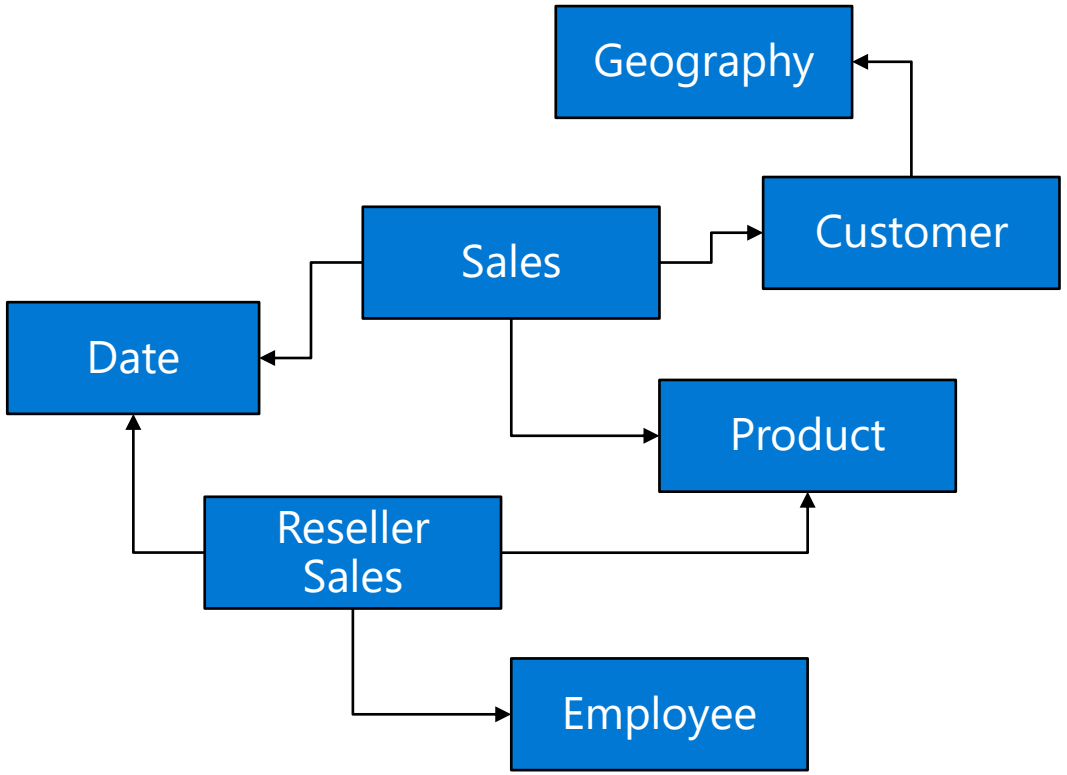
Why choose? Import and DirectQuery in a single model

Keep summarized data local; get detail data from the source

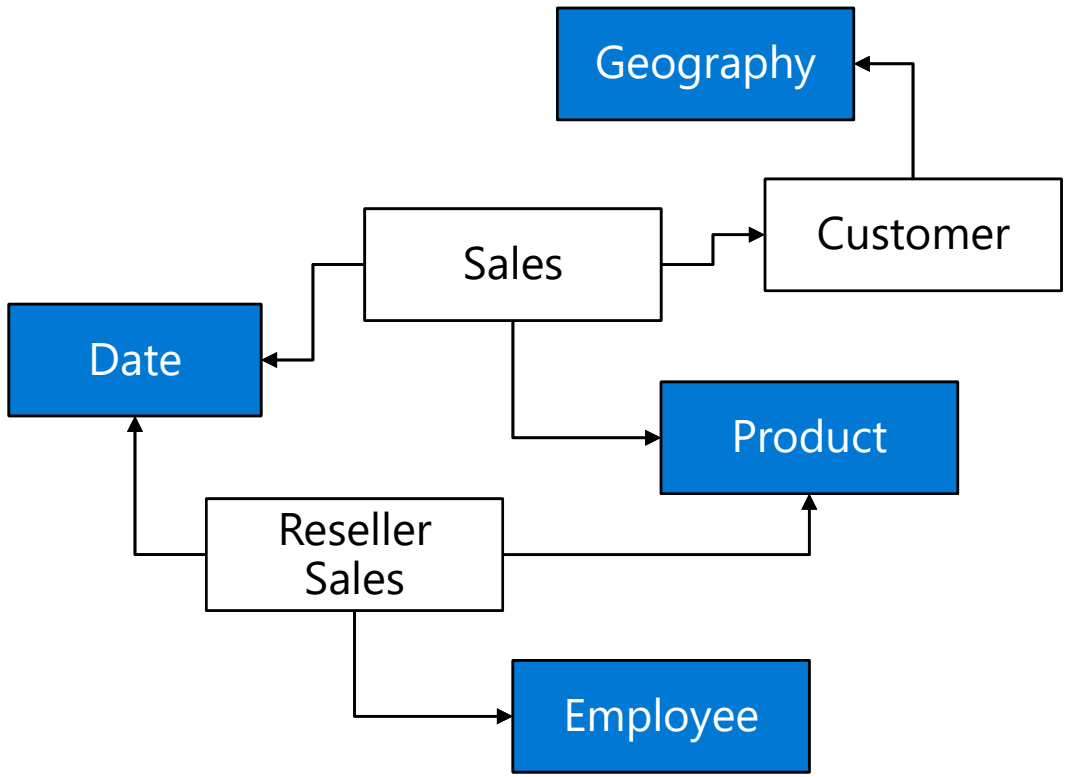
DirectQuery



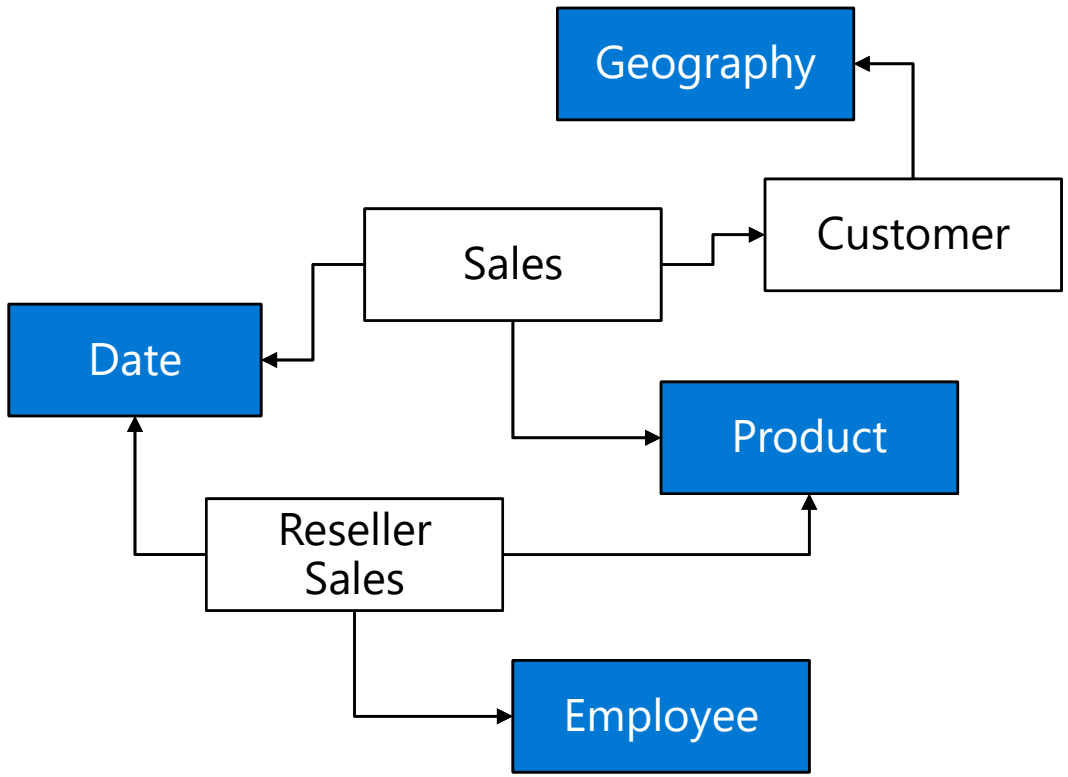
Import



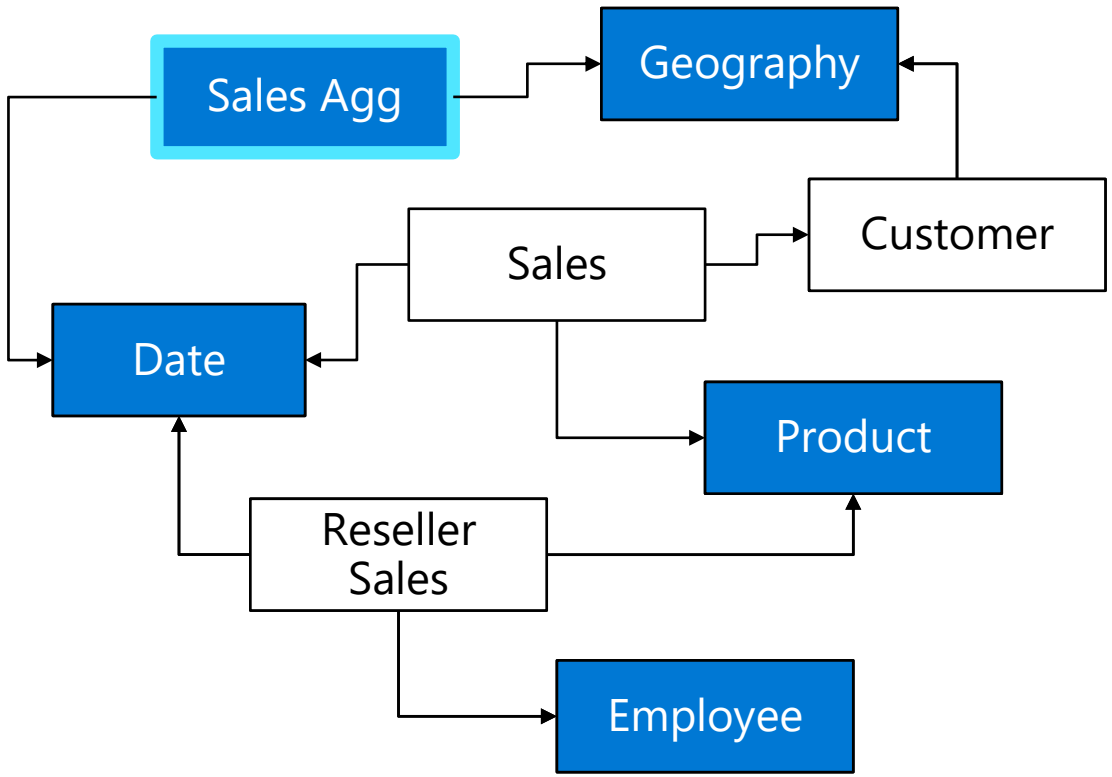
DirectQuery & Import



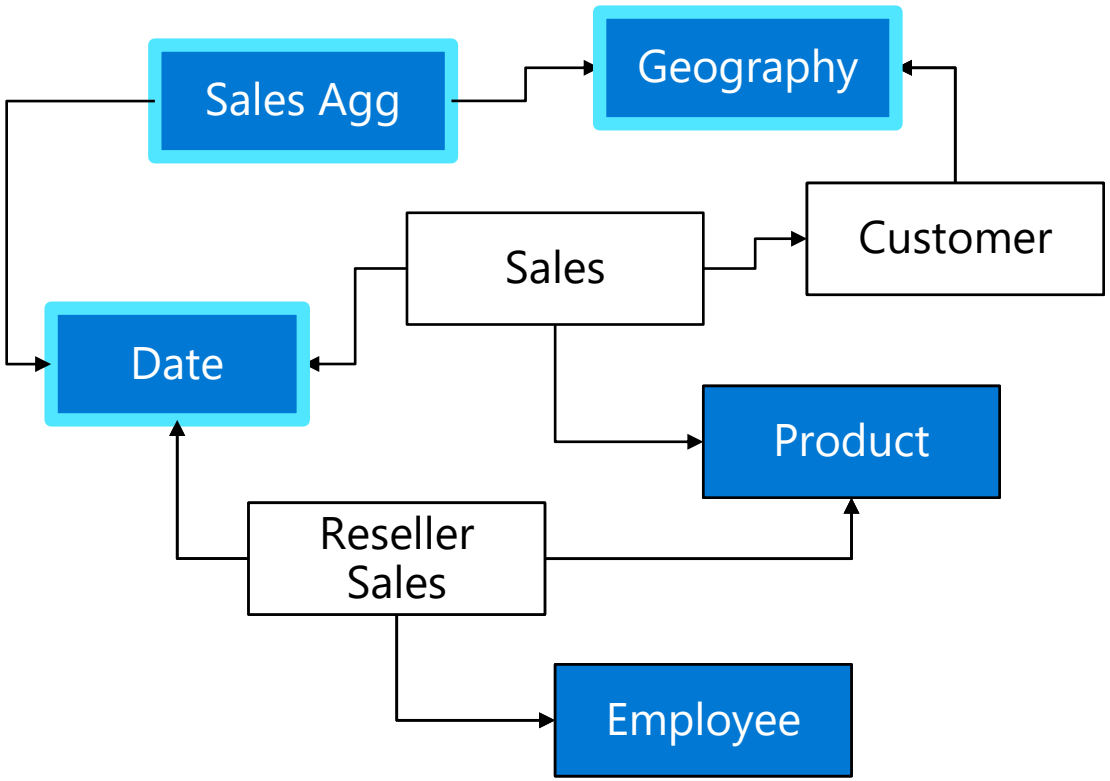
DirectQuery & Import



Aggregations



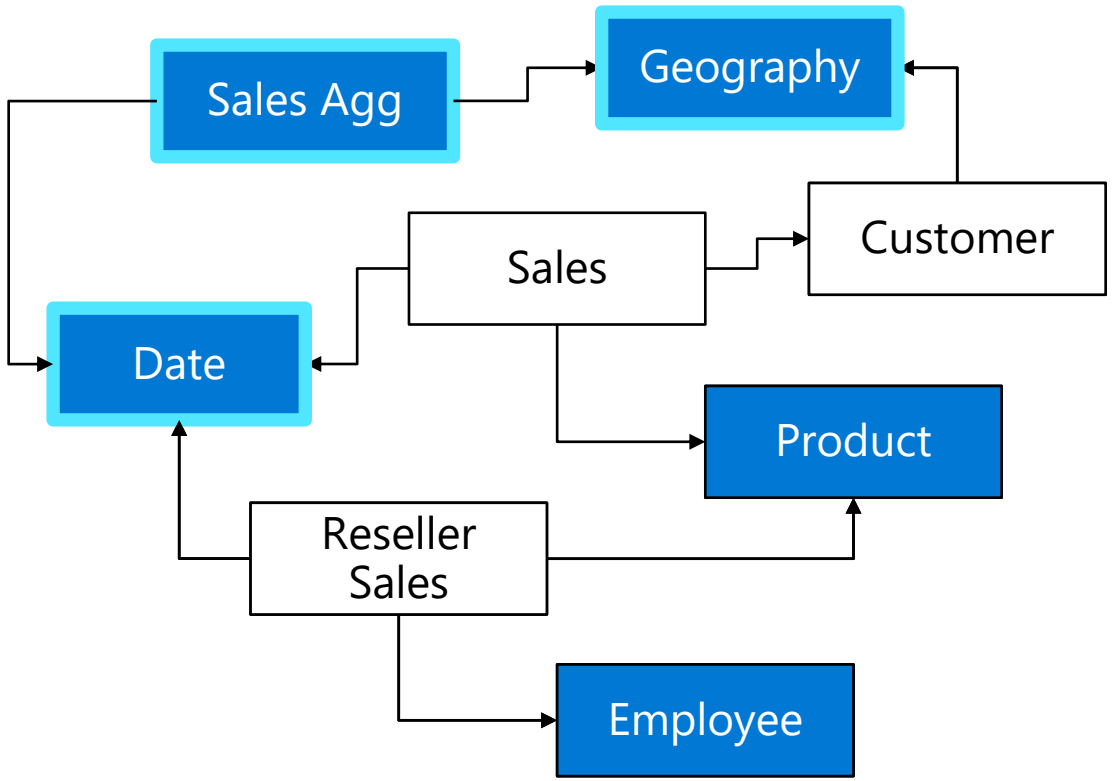
Aggregations




```
SummarizeColumns(
        Date[Year],
        Geography[City],
        "Sales", Sum(Sales[Amount])
)
```

Hits in-memory cache

Aggregations




```
SummarizeColumns(
        Date[Year],
        Geography[City],
        "Sales", Sum(Sales[Amount])
)
```

DirectQuery

```
SELECT [Year],
       [Name],
       SUM([Amount]) AS [Amount]
FROM   [Sales]
INNER JOIN [Date] ON ...
INNER JOIN [Customer] ON ...
GROUP BY [Year],
         [Name]
```

In preview

Azure Synapse + Power BI integration

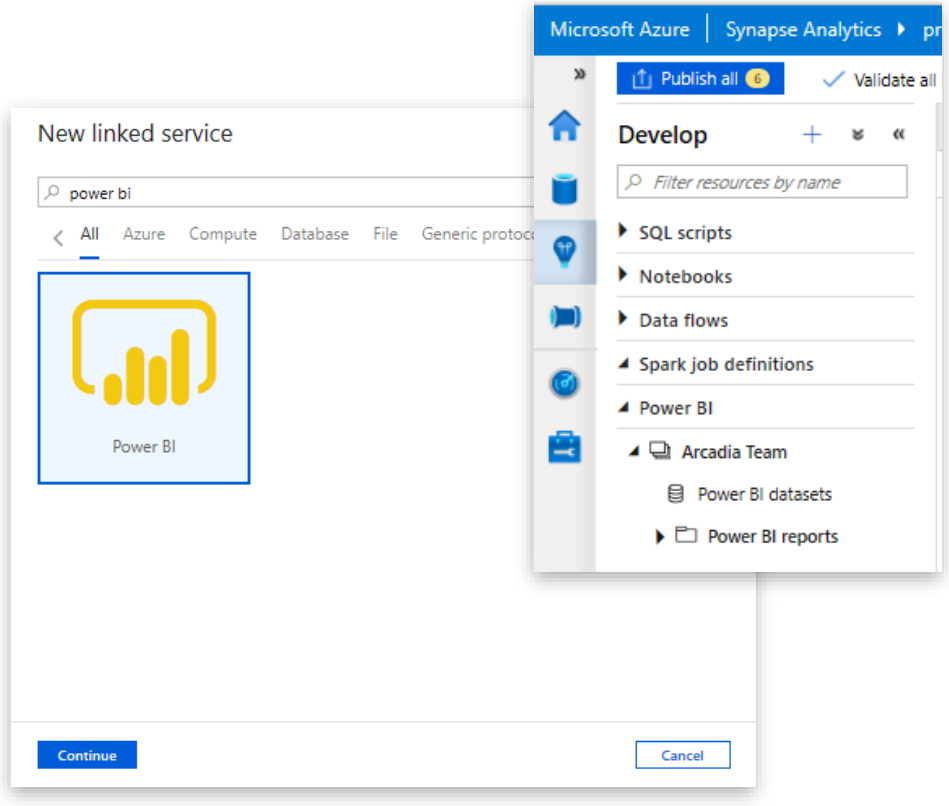
Build Power BI dashboards directly from Azure Synapse (preview)

Azure Synapse + Power BI integration

Develop Hub – Power BI

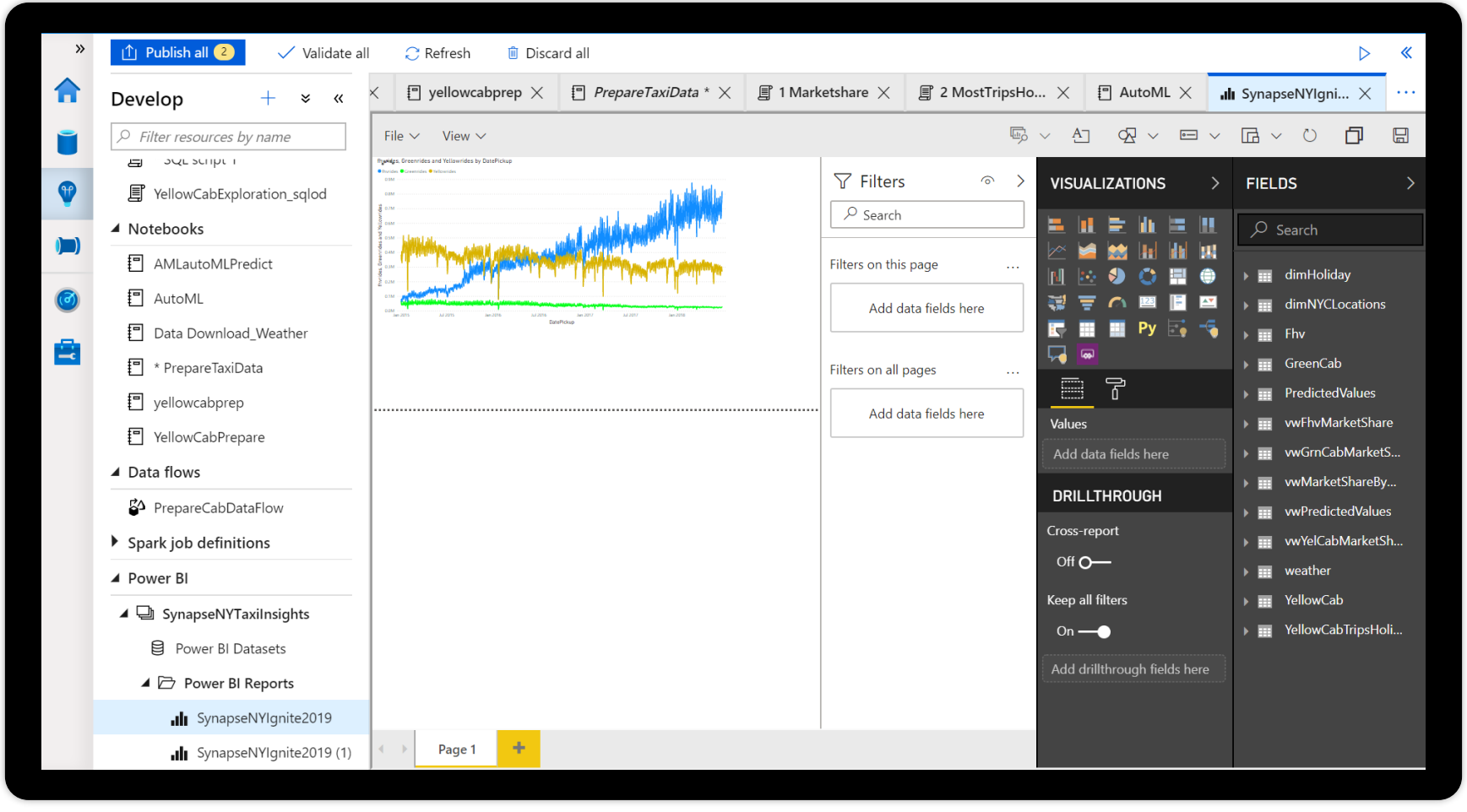
Overview

- Create Power BI reports in the workspace
- Provides access to published reports in the workspace
- Update reports real time from Synapse workspace to get it reflected on Power BI service
- Visually explore and analyze data



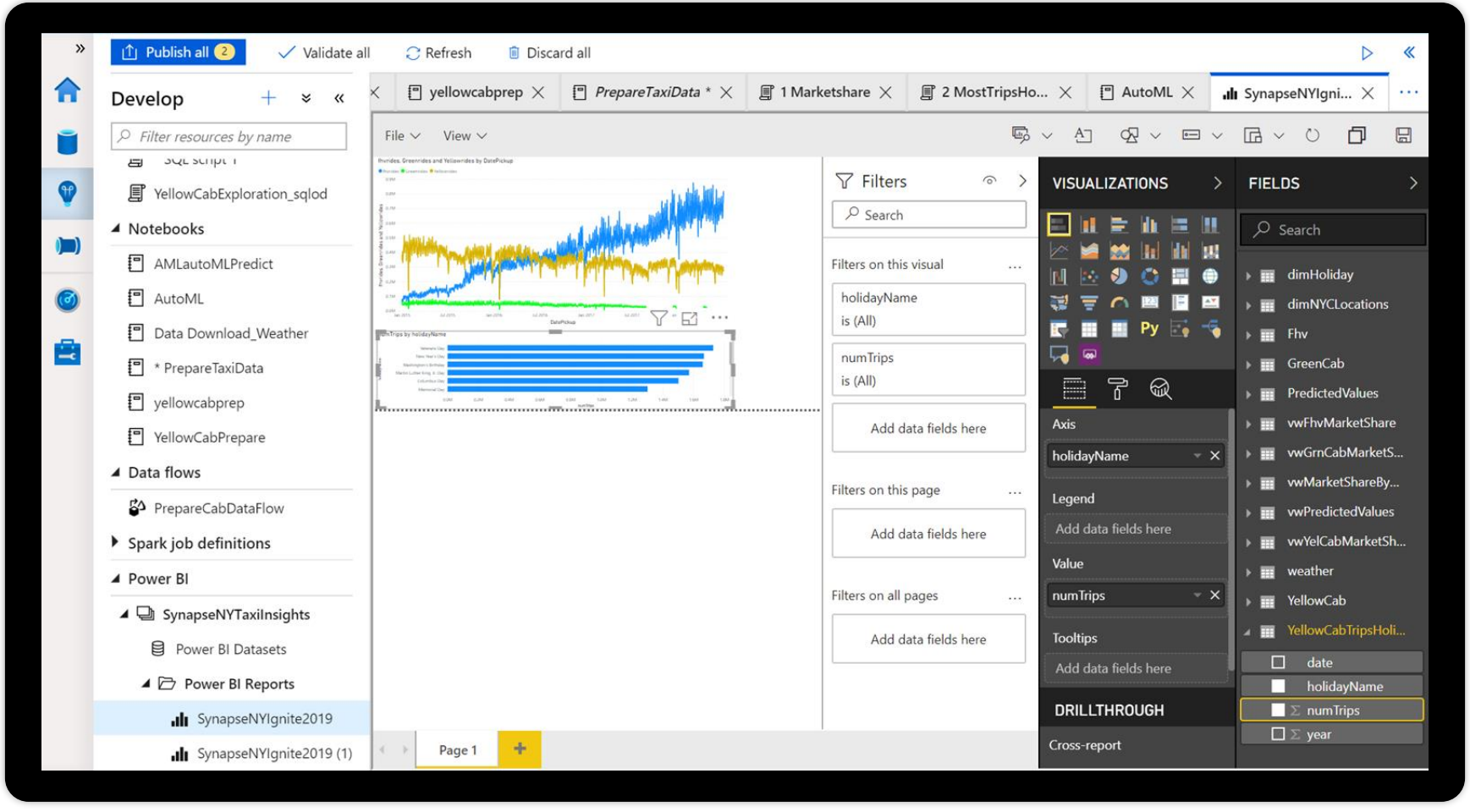
Develop Hub – Power BI

View published reports in Power BI workspace



Develop Hub – Power BI

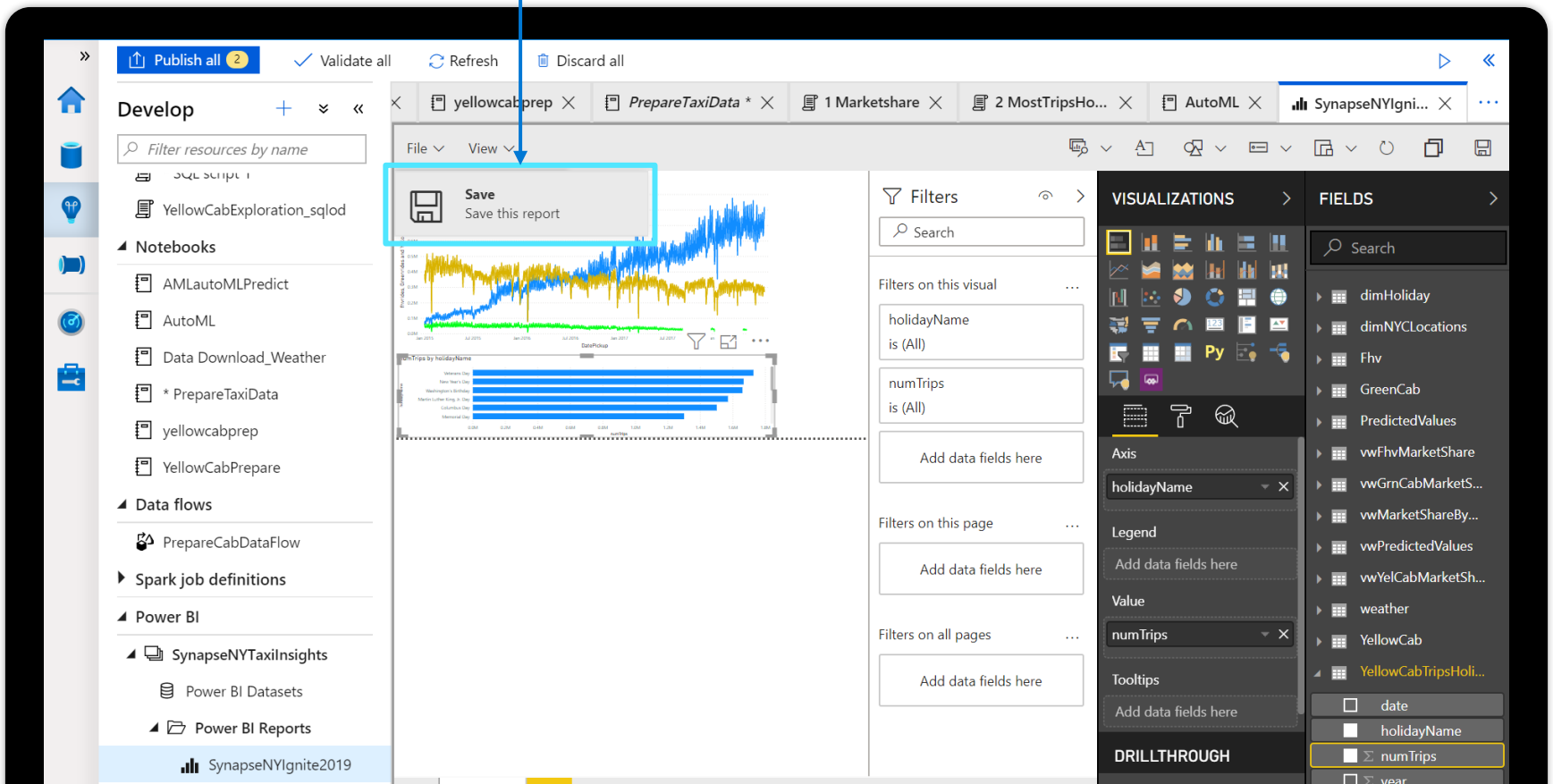
Edit reports in Synapse workspace



Develop Hub – Power BI

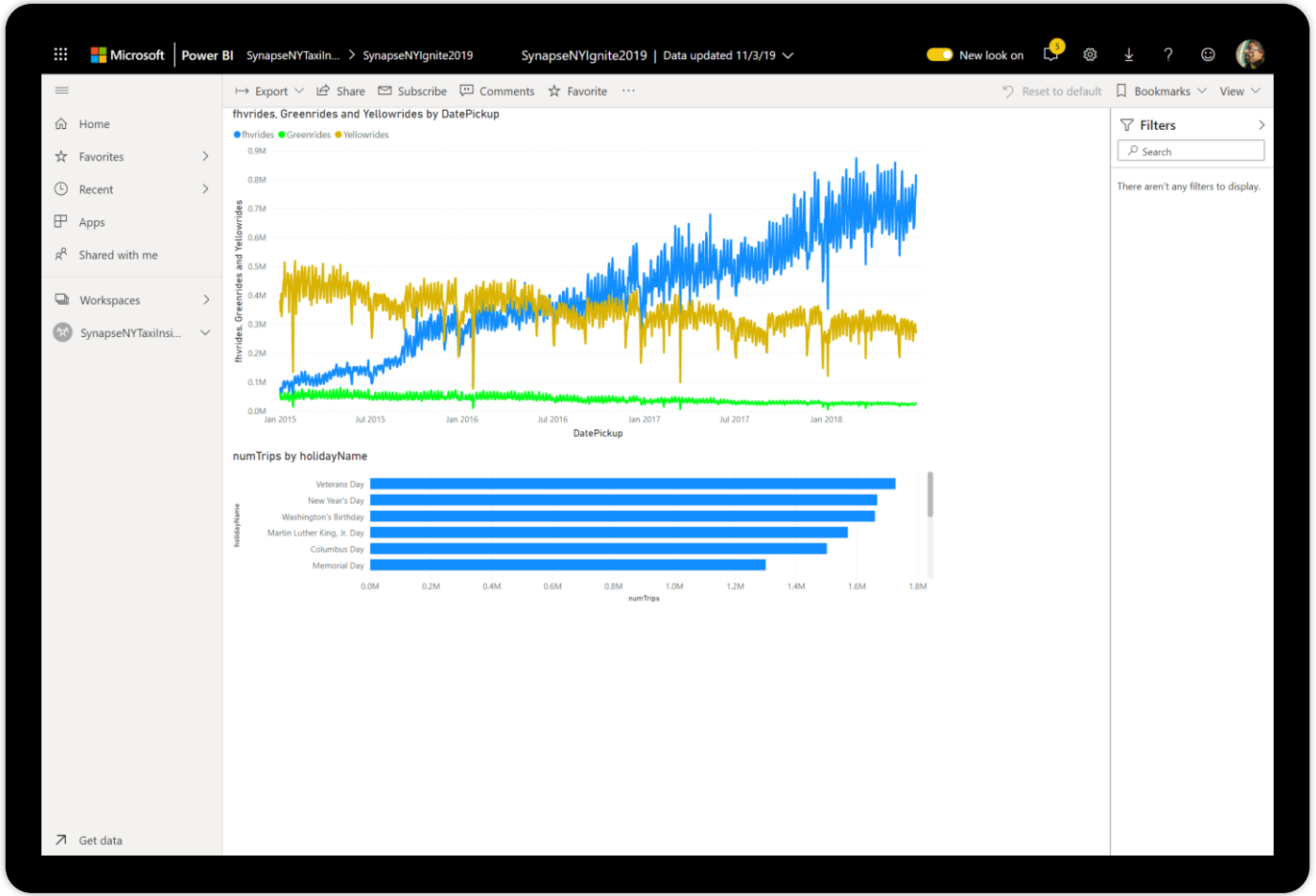
Publish edited reports in Synapse workspace to Power BI workspace

Publish changes by simple save report in workspace



Develop Hub – Power BI

Real-time publish on save




PASS