Lesson 20

Topic: Publishing and Sharing in Power BI
Prerequisites: Give a written brief explanation for all questions.

1. How does Power BI handle large datasets in the Online Service, and what is the role of Premium Capacity in this?

Power BI handles large datasets in the **Power BI Online Service** with several built-in mechanisms and architectural strategies. The **role of Premium Capacity** is central to enabling **larger scale**, **better performance**, and **advanced features**.

Handling Large Datasets in Power BI Online Service

By default (in **shared capacity**, i.e., Free or Pro workspaces):

- **Dataset size is limited** to **1 GB** per dataset after compression.
- **Scheduled refreshes** are limited (up to 8 times per day for Pro).
- **Memory and CPU resources** are shared among many users.
- **Performance may degrade** for large models or concurrent users.

To handle large datasets effectively, Power BI:

- Uses **columnar storage** (based on VertiPaq engine) for compression.
- Supports **incremental refresh** to update only recent data.
- Applies **query folding** (delegating filters to the data source).
- Caches queries using **semantic model (dataset) caching**.

---

2. Role of Premium Capacity

Power BI Premium Capacity provides **dedicated cloud resources**, which dramatically improves how large datasets are managed.

Key Advantages:

Feature	Description
Larger dataset sizes	Up to **400 GB** per dataset with **Premium Gen2** (vs 1 GB in Pro)
More refreshes	Up to **48 refreshes/day**, more frequent updates
Dedicated compute	Resources (CPU/RAM) are **not shared**, ensuring **consistent**

Feature	Description
	performance
Paginated Reports	Support for pixel-perfect reports ideal for printing
AI capabilities	Use of ML models, AutoML, cognitive services, etc.
Deployment pipelines	Enterprise-grade CI/CD and version control support
Incremental refresh & hybrid tables	Efficient refresh by updating only recent partitions
XMLA endpoint	Full access to models via SSMS or third-party tools (for dev/ops)

Summary

Aspect	Shared (Pro)	Premium Capacity
Max Dataset Size	1 GB	Up to 400 GB
Refresh Frequency	8/day	48/day
Performance	Shared CPU/memory	Dedicated CPU/memory
Cost	Lower (per-user)	Higher (per-capacity or per-user via Premium per user - PPU)
Advanced Features	Limited	Yes (AI, paginated reports, XMLA, etc.)

✅ **Conclusion**

Premium Capacity is essential for:

Enterprises managing **large volumes of data**,

Needing **faster, more frequent updates**,

Demanding **advanced features** and **enterprise-scale BI**.

2.What are the differences between Import mode, DirectQuery, and Live Connection in Power BI Service?

In **Power BI**, the way you connect to and interact with data depends on the **storage mode** you choose. The three primary modes—**Import**, **DirectQuery**, and **Live Connection**—have key differences in performance, flexibility, and real-time capabilities. Here's a detailed breakdown:

---

1. Import Mode

How it works:

Data is **copied and stored** in Power BI's compressed in-memory engine (VertiPaq).

Refreshes must be scheduled or triggered manually.

Pros:

✅ **Fastest performance** (optimized for analytics).

✅ Supports **all Power BI features** (DAX, calculated columns, Q&A, etc.).

✅ Works **offline** (once imported).

✅ Allows **data transformation** (Power Query).

Cons:

❌ **Not real-time** (data must be refreshed).

❌ **Storage limits** (depends on Power BI license).

❌ Higher memory usage for large datasets.

Best for:

✔ Static or frequently refreshed reports.

✔ Complex transformations or DAX-heavy models.

✔ Scenarios where speed is critical.

---

2. DirectQuery (DQ)

How it works:

Queries the source database in real-time (no data is stored in Power BI).

Only metadata (model structure) is stored.

Pros:

✅ **Near real-time data** (always current).

✅ No storage limits (data stays in the source).

✅ Good for **large datasets** (avoids import limits).

Cons:

❌ **Slower performance** (depends on source system speed).

❌ **Limited transformations** (some Power Query features disabled).

❌ **No calculated columns** (only measures).

❌ **Source system load** (each visual triggers a query).

Best for:

✔ Reports needing **live data** (e.g., stock prices, operational dashboards).

✔ Very large datasets that exceed import limits.

✔ When source systems can handle query loads.

---

3. Live Connection

How it works:

Connects **directly to an existing semantic model** (e.g., SQL Server Analysis Services, Power BI datasets, or Datamarts).

No data is imported—Power BI acts as a **visualization layer**.

Pros:

✅ **Ultra-fast for pre-aggregated models** (SSAS Tabular, Power BI Premium datasets).

✅ **Always in sync** (no refresh needed).

✅ Centralized governance (single source of truth).

Cons:

❌ **No Power Query transformations** (must be done in the source model).

❌ **Limited to the source model's capabilities** (no DAX changes in Power BI).

❌ Requires **pre-built data models** (SSAS, Power BI datasets).

Best for:

✔ Enterprise environments with **centralized data models**.

✔ When using **Analysis Services (SSAS)** or shared Power BI datasets.

✔ Reports needing **zero-latency** data.

Comparison Summary

Feature	Import Mode	DirectQuery (DQ)	Live Connection
Data Storage	In Power BI	In Source DB	In Semantic Model (SSAS/PBI Dataset)
Real-Time	No (scheduled refresh)	Yes (query on demand)	Yes (always live)
Performance	⚡ Fastest	Slower (depends on source)	⚡ Fast (if model optimized)
Transformations	Full Power Query	Limited	None (must be in source)
DAX/Calculations	Full support	Measures only	Depends on source model
Best For	Static reports, speed	Large/live data	Enterprise semantic models

---

When to Use Which?

Use Import → For most self-service reports needing speed & flexibility.

Use DirectQuery → For real-time analytics on large or frequently updated data.

Use Live Connection → When leveraging existing enterprise models (SSAS, shared datasets).

3. Explain deployment pipelines in Power BI Online. What stages do they include?

Deployment pipelines in Power BI Online are a **DevOps-style feature** that help teams manage the **lifecycle of Power BI content** (like reports, datasets, and dashboards) across **development, testing, and production environments** in a structured, controlled, and repeatable way.

What Are Deployment Pipelines?

A **deployment pipeline** allows BI teams to:

Develop content in a **development workspace**.

Test changes in a **test workspace**.

Publish stable versions to a **production workspace**.

Compare, **validate**, and **deploy** changes between environments.

This is crucial for:

Avoiding breaking changes in production.

Ensuring version control.

Supporting team collaboration and CI/CD-like workflows.

Stages in a Deployment Pipeline

Deployment pipelines in Power BI have **three default stages**:

Stage	Purpose
1. Development	Where authors build and design reports, datasets, and dashboards. Frequent updates are made here.
2. Test (or Test/Validation)	Content is reviewed and validated by testers or stakeholders. Permissions and data connections can be tested.
3. Production	Final, approved version of the content is used by end users. Only stable and verified reports are deployed here.

Key Features of Deployment Pipelines

✅ **Side-by-side comparison**: See differences between stages before deploying.

Selective deployment: Choose specific items to deploy (not all).

Parameter rules: You can configure different **parameters** (e.g., database connection strings) for each stage.

Data refresh control: Refresh datasets at each stage independently.

Integration with CI/CD: Integrate pipelines with APIs or tools like Azure DevOps.

Example Workflow

Develop a sales dashboard → Save it in the **Development** workspace.

Promote to **Test** → Test filters, refresh schedules, row-level security.

Approve and **deploy to Production** → End users consume it safely.

Licensing Requirement

Deployment pipelines require **Power BI Premium** or **Premium Per User (PPU)** licenses.

Pipelines only work with **workspaces assigned to Premium capacity**.

✅ **Summary**

Feature	Description
Stages	Development → Test → Production
Purpose	Manage and promote content safely
Benefits	Version control, testing before release, reliable updates
Requirement	Power BI Premium or Premium Per User

4. How can Power BI Service integrate with Microsoft Teams or SharePoint for collaboration?

Power BI Service integrates seamlessly with **Microsoft Teams** and **SharePoint** to enhance **collaboration**, **communication**, and **data-driven decision-making** across organizations.

---

1. Integration with Microsoft Teams

Power BI is tightly integrated with Microsoft Teams to enable **collaborative BI inside the chat and meeting environment**.

✅ **Key Integration Features:**

Capability	Description
Power BI App for Teams	Use Power BI within Teams without switching apps. Users can explore, create, and share reports from inside Teams.
Embed Reports in Teams Channels	You can embed a specific Power BI report directly into a Teams tab in any channel.
Chat with a Report	Discuss a report directly within the Teams chat using the **"Chat in Teams"** feature from Power BI Service.
Automated Notifications	Set up alerts or scheduled report updates that push to Teams channels via Power Automate.

How to Embed a Report in Teams:

Go to Power BI Service → Open report.

Click **“File” → “Embed report” → “Microsoft Teams”**.

Select the target Teams channel/tab.

---

2. Integration with SharePoint Online

Power BI can also integrate with **SharePoint Online** to embed and share reports across intranet portals and team sites.

✅ Key Integration Features:

Capability	Description
Embed Interactive Reports	Use the **Power BI web part** in modern SharePoint pages to embed fully interactive reports.
Secure Access Control	Respects Power BI workspace and RLS (row-level security) permissions.
Live Updates	The embedded reports are live – any update in the dataset reflects immediately.
No Code Required	Easily configured through SharePoint's web interface without any development work.

How to Embed in SharePoint:

Go to Power BI Service → Open the report.

Click **"File" → "Embed report" → "SharePoint Online"**.

Copy the link and paste it into a **Power BI web part** on your SharePoint site.

✅ Summary Table

Feature	Microsoft Teams	SharePoint Online
Embed Reports	✅ Yes (tabs & chats)	✅ Yes (web part)
Permissions Sync	✅ Teams + Power BI workspace	✅ SharePoint + Power BI workspace
Collaboration	Chat, calls, notifications	Portal-based access
Setup Required	Add Power BI app to Teams	Use modern SharePoint + web part
Real-Time Data	✅	✅

Bonus: Power Automate Integration

Both Teams and SharePoint can use **Power Automate flows** to:

Notify Teams when reports are updated.

Schedule data alerts.

Send SharePoint form data to Power BI.

5. What is the XMLA endpoint in Premium and how does it benefit developers or enterprise BI teams?

The **XMLA endpoint** (XML for Analysis endpoint) in **Power BI Premium** is a powerful feature that allows **external tools** to **connect directly to Power BI datasets**, enabling **advanced modeling, automation, and enterprise-level BI capabilities**.

What Is the XMLA Endpoint?

The **XMLA endpoint** is an **open communication interface** used by **Analysis Services** (the engine behind Power BI datasets). With Premium (or Premium Per User), Microsoft **exposes this endpoint** so you can treat a Power BI dataset like a traditional **tabular model in SSAS**.

Think of it like this: The XMLA endpoint makes your Power BI dataset behave like a SQL Server Analysis Services (SSAS) model—allowing deep integration with external BI and dev tools.

Access Levels

XMLA endpoints support two modes:

Mode	Description
Read	View metadata and run queries (e.g., via Excel or SSMS). Available to all workspaces.
Read/Write	Modify the model (e.g., add measures, partitions, update schema). **Only in Premium / PPU** workspaces.

Key Benefits for Developers and BI Teams

Benefit	Description
✅ **External Tool Access**	Use **SSMS**, **Tabular Editor**, **DAX Studio**, and **ALM Toolkit** to connect directly to Power BI models.
Advanced Modeling	Create complex measures, calculation groups, perspectives, translations, roles, and partitions **not available in Power BI Desktop**.
CI/CD and Automation	Integrate with **DevOps pipelines**, **deployment scripts**, and **version control** using tools like Tabular Editor or PowerShell.
Large-Scale Deployment	Manage and deploy datasets as **code** using the **TOM (Tabular Object Model)** and **TMSL (Tabular Model Scripting Language)**.
Performance Tuning	Profile and optimize DAX queries using **DAX Studio** connected via XMLA.
Security Management	Define and manage **row-level security (RLS)** and roles programmatically.
Data Refresh Partitions	Create **incremental refresh** strategies by scripting partition logic directly (e.g., for large models).

How to Connect via XMLA

Enable the XMLA endpoint in Power BI Admin Portal.

Get the workspace connection string:

bash
CopyEdi
powerbi://api.powerbi.com/v1.0/myorg/[WorkspaceName]

Open your tool (e.g., Tabular Editor).

Connect to the above workspace URL using your Power BI credentials.

✅ **Summary**

Feature	XMLA Endpoint
Available In	Power BI Premium & Premium Per User
Protocol	XMLA (same as SSAS tabular)
Tools Supported	SSMS, Tabular Editor, DAX Studio, ALM Toolkit
Use Cases	Modeling, scripting, CI/CD, RLS, performance tuning
Access Types	Read / Read-Write

6. Describe how usage metrics and audit logs work in Power BI Service.

In **Power BI Service**, **usage metrics** and **audit logs** are two key features that help organizations **monitor**, **analyze**, and **govern** how reports, dashboards, and datasets are being accessed and used.

---

1. Usage Metrics in Power BI

Usage metrics provide **built-in analytics** about how often your **reports and dashboards** are being used.

✅ **Key Features:**

Feature	Description
Pre-built report	Power BI automatically generates a **usage metrics report** for each report or dashboard.
User tracking	See who viewed the report, how often, and when.
Date filters	Analyze usage over specific time periods (last 30 days, etc.).
Engagement data	Includes views, viewers, shares, export actions, etc.
Customizable	You can **save and customize** the metrics report like any other Power BI report.

How to Use:

Open a report or dashboard in Power BI Service.

Click **"…" > View usage metrics report**.

Explore or customize the generated report.

Notes:

Only available for **report/dataset creators** and users with at least **Contributor** role.

Not available in **personal workspaces**.

---

2. Audit Logs in Power BI (Microsoft 365 Compliance Center)

Audit logs provide **detailed activity tracking** across Power BI (and other Microsoft 365 services). They are designed for **IT admins**, **security**, and **compliance** needs.

✅ **Key Features:**

Feature	Description
User-level logging	Tracks exactly **who did what**, when, and on what object.
Governance-ready	Used for **security auditing**, compliance, and data governance.
Centralized in Microsoft 365	Access logs via the **Microsoft 365 compliance portal**.
Exportable	Can be exported or queried via PowerShell or API.

Common Tracked Events:

Event	Example
Viewed report	User opened a report
Exported data	Data exported to Excel or PDF
Shared report	User shared content with another
Dataset refreshed	Scheduled or manual refresh ran
Created/deleted report	User published or removed a report

How to Access Audit Logs:

Go to Microsoft 365 Compliance Center.

Select **Audit → Audit Search**.

Filter by **Power BI activities**.

Export or save the results.

Requirements:

Must be a **Global Admin**, **Audit Admin**, or **Compliance Admin**.

Audit log search must be **enabled** in the Microsoft 365 tenant.

Comparison: Usage Metrics vs. Audit Logs

Feature	Usage Metrics	Audit Logs
Purpose	User engagement & report popularity	Security, compliance, and admin auditing
Level	Report/Dashboard	Tenant-wide activities
Who Can Use	Report owners	Microsoft 365 Admins
Tools	Power BI UI	Microsoft 365 Compliance Center, PowerShell
Export	Yes (Power BI report)	Yes (CSV, PowerShell)

✅ **Summary**

Usage metrics help report owners understand how users interact with their content.

Audit logs give admins full visibility into user actions for compliance and governance.

Both are essential tools for **managing, securing, and optimizing** your Power BI environment.

7.How do you manage workspace access and permissions for different users?

Managing **workspace access and permissions** in Power BI Service is essential for **data security**, **collaboration**, and **governance**. Workspaces act as containers for reports, dashboards, datasets, and dataflows—and **access control** determines what users can do with that content.

---

1. Types of Roles in a Power BI Workspace

Each workspace in Power BI (especially in **modern workspaces**) supports the following roles:

Role	Permissions
Admin	Full control – add/remove members, publish, delete, assign roles, manage settings.
Member	Create, edit, publish content, and manage datasets. Cannot delete workspace or manage roles.
Contributor	Publish and edit content but cannot manage dataset permissions or others' content.
Viewer	Read-only access – can view reports and dashboards but **can't edit or publish**.

---

2. How to Assign Roles in a Workspace

Go to **Power BI Service** → **Workspaces**.

Open the target workspace.

Click on **"Access"** or **Settings > Permissions**.

Use the **"+ Add people or groups"** option to assign:

Azure AD users

Security groups

Microsoft 365 groups

Assign one of the 4 roles (Admin, Member, Contributor, Viewer).

---

3. Managing Dataset Permissions Separately

While workspace roles control overall access, **datasets** can have **separate build permissions**:

Give users **"Build" permission** if they need to:

Create reports on top of the dataset.

Analyze data in Excel.

Use the dataset in other workspaces (composite models).

To manage:

Click on the dataset → **"Manage Permissions"**.

Add users or groups with **Read/Build** rights.

---

4. Best Practices for Workspace Access

Practice	Recommendation
✅ Use security groups	Easier to manage access at scale.
Avoid using personal workspaces	Use shared workspaces for production reports.
Apply RLS (Row-Level Security)	Control what data users can see based on roles.
Use Viewer role for consumers	Prevent unintentional edits by non-authors.
Separate dev/test/prod	Use **deployment pipelines** or separate workspaces for each environment.

---

5. Auditing Access

To monitor access:

Use **Power BI Admin portal** (for tenant-level view).

Use **audit logs** in Microsoft 365 Compliance Center to track who accessed what and when.

✅ **Summary**

Task	How
Add users to workspace	Use **"Access"** and assign roles
Set dataset permissions	Use **"Manage Permissions"** on datasets
Restrict data views	Implement **Row-Level Security (RLS)**
Best for many users	Assign **security groups**
Monitoring access	Use **Audit Logs** and **Admin Portal**

8. How can data governance be enforced in Power BI Service?

Enforcing **data governance** in **Power BI Service** is critical for ensuring **data quality**, **security**, **compliance**, and **trust** across an organization. Power BI provides several **built-in governance tools** and integrates with **Microsoft 365** security and compliance features to support enterprise data governance.

---

1. Workspace and Access Management

Strategy	Tools
Role-based access control	Use **workspace roles** (Admin, Member, Contributor, Viewer)
Centralize with security groups	Assign **Azure AD groups** to workspaces and datasets
Separate environments	Use **dedicated workspaces** for Dev, Test, and Production
Deployment pipelines	Control promotion of content between environments

---

2. Dataset Governance and Certification

Feature	Purpose
Endorsement (Certified/Promoted)	Tag **trusted datasets** so users know which ones are verified
Sensitivity labels	Apply **Microsoft Purview Information Protection** labels (e.g., Confidential, Public)
Dataset ownership	Assign clear dataset owners to ensure accountability
Data lineage view	Visualize how datasets, reports, and dataflows are connected for impact analysis

---

3. Data Security and Compliance

Security Measure	Description
Row-Level Security (RLS)	Restrict data access **within a dataset** based on user identity
Audit logs	Track all activities (viewing, sharing, publishing) in **Microsoft 365 Compliance Center**
Data loss prevention (DLP) policies	Prevent sharing or exporting sensitive data outside the organization
Sensitivity labels	Encrypt and classify data using Microsoft Purview, inherited across services like Excel, Teams
Bring Your Own Key (BYOK)	Control data encryption using your own keys (Premium only)

---

4. Monitoring and Admin Oversight

Tool	Purpose
Power BI Admin Portal	Manage tenant settings, audit usage, assign capacities
Usage Metrics	View report-level engagement stats
Power BI REST API / PowerShell	Automate governance tasks (access review, content inventory)
Purview or Microsoft Defender for Cloud Apps	Monitor and classify Power BI assets for security policies

---

5. Governance Policies via Tenant Settings

In the **Power BI Admin Portal**, tenant-level governance policies can be enforced:

Setting	Example
Export control	Disable data export, copy, and print
Sharing restrictions	Limit external sharing or publish to web
Data refresh policy	Control how often datasets can refresh
Workspace creation	Allow only certain users or groups to create workspaces

---

✅ Summary: Key Governance Pillars

Governance Area	Tools & Features
Access Control	Workspace roles, security groups, RLS
Content Trust	Endorsement, sensitivity labels, data lineage
Security & Compliance	Purview labels, audit logs, DLP policies
Monitoring	Admin portal, usage metrics, APIs
Policy Enforcement	Tenant settings, deployment pipelines

Enforcing **data governance** in **Power BI Service** is critical for ensuring **data quality**, **security**, **compliance**, and **trust** across an organization. Power BI provides several **built-in governance tools** and integrates with **Microsoft 365** security and compliance features to support enterprise data governance.

---

1. Workspace and Access Management

Strategy	Tools
Role-based access control	Use **workspace roles** (Admin, Member, Contributor, Viewer)
Centralize with security groups	Assign **Azure AD groups** to workspaces and datasets
Separate environments	Use **dedicated workspaces** for Dev, Test, and Production
Deployment pipelines	Control promotion of content between environments

---

2. Dataset Governance and Certification

Feature	Purpose
Endorsement (Certified/Promoted)	Tag **trusted datasets** so users know which ones are verified
Sensitivity labels	Apply **Microsoft Purview Information Protection** labels (e.g., Confidential, Public)
Dataset ownership	Assign clear dataset owners to ensure accountability
Data lineage view	Visualize how datasets, reports, and dataflows are connected for impact analysis

---

3. Data Security and Compliance

Security Measure	Description
Row-Level Security (RLS)	Restrict data access **within a dataset** based on user identity
Audit logs	Track all activities (viewing, sharing, publishing) in **Microsoft 365 Compliance Center**
Data loss prevention (DLP) policies	Prevent sharing or exporting sensitive data outside the organization
Sensitivity labels	Encrypt and classify data using Microsoft Purview, inherited across services like Excel, Teams
Bring Your Own Key (BYOK)	Control data encryption using your own keys (Premium only)

---

4. Monitoring and Admin Oversight

Tool	Purpose
Power BI Admin Portal	Manage tenant settings, audit usage, assign capacities
Usage Metrics	View report-level engagement stats
Power BI REST API / PowerShell	Automate governance tasks (access review, content

Tool	Purpose
	inventory)
Purview or Microsoft Defender for Cloud Apps	Monitor and classify Power BI assets for security policies

---

5. Governance Policies via Tenant Settings

In the **Power BI Admin Portal**, tenant-level governance policies can be enforced:

Setting	Example
Export control	Disable data export, copy, and print
Sharing restrictions	Limit external sharing or publish to web
Data refresh policy	Control how often datasets can refresh
Workspace creation	Allow only certain users or groups to create workspaces

---

✅ Summary: Key Governance Pillars

Governance Area	Tools & Features
Access Control	Workspace roles, security groups, RLS
Content Trust	Endorsement, sensitivity labels, data lineage
Security & Compliance	Purview labels, audit logs, DLP policies
Monitoring	Admin portal, usage metrics, APIs
Policy Enforcement	Tenant settings, deployment pipelines

9.What are the limitations of Row-Level Security when using DirectQuery or Live Connection?

Row-Level Security (RLS) in Power BI allows you to restrict data access **at the row level** based on **user identity**, but when using **DirectQuery** or **Live Connection**, RLS behaves differently and has several important **limitations**.

Key Difference: DirectQuery vs Live Connection

Mode	Description
DirectQuery	Power BI sends queries **directly to the data source** in real time.
Live Connection	Power BI connects to an external Analysis Services (like SSAS or AAS) model and **does not manage the model itself**.

---

1. Limitations of RLS with DirectQuery

✅ Supported (with conditions), but:

Limitation	Explanation
Performance impact	RLS adds **WHERE clauses** to each query → this can significantly **slow down** DirectQuery performance, especially on large datasets.
Limited transformation support	Some **Power Query transformations** (e.g., M functions) aren't compatible with RLS in DirectQuery mode.
No RLS for composite models with multiple DirectQuery sources	RLS may not work properly when combining multiple DirectQuery sources.
Caching disabled	RLS + DirectQuery prevents some **query caching** → slower performance.
RLS must be defined in Power BI	Not inherited from the source system — RLS must be **manually implemented** in Power BI Desktop.

---

2. Limitations of RLS with Live Connection

❌ Power BI RLS does not apply to Live Connections. RLS is handled entirely in the source system.

Limitation	Explanation
❌ No RLS control in Power BI	You **cannot define RLS** in Power BI for Live Connected models (e.g., SSAS Tabular, Azure AS).
RLS must be defined in SSAS/AAS	All row-level access control is managed using **roles and permissions in Analysis Services**, not Power BI.
Can't customize roles in Power BI	Users must be assigned to roles in the source model – Power BI does not manage this.
No dynamic RLS in Power BI	You can't use USERNAME() or USERPRINCIPALNAME() in Power BI Desktop for Live Connected models – these must be in the source model logic.

✅ Workarounds and Best Practices

Scenario	Recommendation
DirectQuery RLS needed	Use **simplified roles**, optimize source performance, limit user base
Complex security with Live Connection	Define RLS in **SSAS Tabular/Azure Analysis Services** using **dynamic roles**
Mixed models	Use **composite models** carefully, as RLS with multiple sources can fail
Need RLS and performance	Consider **import mode** for sensitive data, apply RLS with better control

---

✅ Summary Table

Feature	DirectQuery	Live Connection
Define RLS in Power BI	✅ Yes	❌ No

Feature	DirectQuery	Live Connection
Performance impact	High	Depends on source
Works with DAX filters	✓ Yes	✓ Yes (in SSAS)
Works with USERNAME()	✓ Yes	✓ In SSAS only
Centralized security management	✗ No (managed in Power BI)	✓ Yes (in SSAS/AAS)

10.Explain how you can refresh a dataset via Power Automate or REST API.

You can refresh a Power BI dataset **programmatically** using either **Power Automate** (no-code/low-code) or the **Power BI REST API** (developer-focused). Both options are useful for triggering refreshes based on events (like file uploads, form submissions, or schedule-based workflows).

---

1. Refresh Dataset via Power Automate

Power Automate provides a **native Power BI connector** that lets you trigger dataset refreshes easily.

✓ Steps:

Go to Power Automate.

Create a new flow (e.g., **Scheduled**, **Button**, or **Automated** trigger).

Add the **"Refresh a dataset"** action from the **Power BI connector**.

Choose:

Workspace: where the dataset is stored

Dataset: to be refreshed

(Optional) Add steps to **wait** or **notify** on refresh success or failure.

Use Cases:

After a file is uploaded to SharePoint or OneDrive

After a Microsoft Form is submitted

On a recurring schedule (e.g., every morning at 6 AM)

---

2. Refresh Dataset via Power BI REST API

The REST API allows developers to automate dataset refreshes via **scripts**, **apps**, or **CI/CD pipelines**.

✅ Endpoint:

http
CopyEdit
POST https://api.powerbi.com/v1.0/myorg/groups/{groupId}/datasets/{datasetId}/refreshes

Requirements:

Register an app in **Azure AD** (get a client ID).

Grant it permissions: **Dataset.ReadWrite.All**.

Authenticate using OAuth2 (get access token).

Use tools like **Postman**, **PowerShell**, or **Python** to make the call.

Sample PowerShell:

powershell
CopyEdit
$accessToken = "your-access-token"
$groupId = "workspace-guid"
$datasetId = "dataset-guid"

Invoke-RestMethod -Uri
"https://api.powerbi.com/v1.0/myorg/groups/$groupId/datasets/$datasetId/refreshes" `
 -Method Post `
 -Headers @{Authorization = "Bearer $accessToken"}

Monitor Refresh Status:

You can also call:

http
CopyEdit
GET https://api.powerbi.com/v1.0/myorg/groups/{groupId}/datasets/{datasetId}/refreshes

To check status, history, and errors.

✅ Summary Table

Feature	Power Automate	Power BI REST API
Skill Level	Low-code (for analysts)	Dev-level (for engineers)
Trigger Types	Button, scheduled, file uploaded, etc.	Scripted, event-driven apps
Setup	Power BI connector	OAuth token, app registration
Monitoring	Built-in actions & flows	API calls for status checks
Best For	Business users, workflows	DevOps, CI/CD, automation pipelines