Question-1

How the Power BI Service handles “big” data

Storage modes fit the shape of your data

Import (VertiPaq, in-memory): fastest for most reporting; compresses data heavily but is constrained by dataset (semantic model) limits. Use with incremental refresh and partitioning so only recent partitions refresh

-

DirectQuery / Live connection: keeps data in the source and queries on demand; pair with **aggregations** to cache common rollups for speed.

- **Composite & Hybrid tables:** mix Import for history with DirectQuery for the hot/latest slice in the same table.
- **Direct Lake (Fabric):** queries parquet in OneLake **without** importing; practical limits are tied to your capacity’s memory, not a fixed file size.

Where Premium (incl. PPU/Fabric capacities) changes the game

Bigger models: Premium supports **Large semantic model** storage so models can grow well **beyond 10 GB** after publish/refresh (Desktop upload itself still has a 10 GB ceiling). Pro/shared doesn’t support growing past small limits. Microsoft Learn

Higher refresh headroom:

- Shared capacity: up to **8** scheduled refreshes/day and **2-hour** max refresh duration.
- Premium capacity/PPU: up to **48** scheduled refreshes/day and **5-hour** duration (and you can orchestrate more via REST/XMLA).

 Dedicated compute = consistency: Your workspace runs on **dedicated capacity** (isolation for CPU/RAM), enabling more **concurrency**, **parallel refresh**, and fewer throttles.

 Enterprise features: XMLA endpoints (read/write) for pro-grade modeling (e.g., Tabular Editor, programmatic partitions, backups), **automatic aggregations**, paginated reports, deployment pipelines, etc. Microsoft Learn+2Microsoft Learn+2

 Direct Lake limits by SKU: In Fabric, the **max memory per Direct Lake semantic model** depends on your capacity (F-SKU); if you exceed it, paging degrades performance.

Question-2

Feature	Import Mode	DirectQuery	Live Connection
Data stored in PBI?	**Yes (in-memory)**	**No**	**No**

Feature	Import Mode	DirectQuery	Live Connection
Performance	Fastest (VertiPaq)	Depends on source	Depends on Analysis Services / semantic model
Refresh	Scheduled refresh (8 Pro / 48 Premium)	No (queries source directly)	Handled in source model
Model flexibility	Full modeling in Power BI	Some restrictions	Minimal (read-only model)
Best for	Fast reporting, compressed datasets	Large/real-time datasets	Centralized enterprise models

Question-3

Deployment Pipelines in Power BI Online

Deployment pipelines are a lifecycle management feature in the Power BI Service that help you move content (reports, dashboards, datasets, dataflows) across environments in a controlled way. They're designed for Dev → Test → Production workflows, much like software development lifecycles.

The Three Stages

1. **Development (Dev)**
    - Workspace where authors build and design reports, datasets, and dashboards.
    - Frequent changes and experiments.
    - Not meant for end users.
2. **Test (or QA)**
    - Workspace for validating content before production.
    - Data source rules can be applied (e.g., connect to a test database instead of production).
    - QA team and selected users review reports for correctness, performance, and formatting.
3. **Production (Prod)**
    - Workspace where content is published for business users.
    - Stable environment with minimal changes.
    - Only thoroughly tested and approved content should reach here.

Question-4

Integration with Microsoft Teams

1. Power BI App for Teams
    - You can install the Power BI app directly inside Teams.
    - Lets users browse, search, and open reports without leaving Teams.
    - Personal app view shows recent/favorite reports.
2. Embedding Reports in Teams Channels/Chats
    - Use the "+" tab in a channel or chat → choose *Power BI* → select a report.
    - Everyone with access sees the live report inside Teams.
3. Message Sharing
    - You can copy a link to a report visual and paste it into a Teams message → it shows up as a rich card preview with direct access.
4. Notifications & Collaboration
    - Teams can notify users when Power BI activity occurs (e.g., report shared, data alert triggered).
    - Users can discuss insights directly in Teams chats while looking at the report**.**

Integration with SharePoint

1. Embed Power BI Reports in SharePoint Online
    - Use the Power BI web part in modern SharePoint pages.
    - This embeds an interactive report (not just a static screenshot).
    - Users must have proper Power BI permissions to view it.
2. Secure Access
    - Report security follows Power BI service permissions (RLS, dataset permissions).
    - No need to duplicate security in SharePoint.
3. Document Libraries + Power BI
    - Data stored in SharePoint Lists or Document Libraries can be used as a data source for Power BI reports.
    - Useful for tracking project tasks, issues, or storing structured Excel data.

Question-5

- XMLA = XML for Analysis protocol (the same protocol SQL Server Analysis Services Tabular uses).
- In Premium/PPU workspaces, the Power BI semantic model (dataset) can be accessed via an XMLA endpoint.
- This makes a Power BI dataset behave like an Analysis Services Tabular model in the cloud.

In short → It lets developers connect to and manage Power BI datasets using external tools, not just the Power BI Service U

Key Benefits for Developers & Enterprise BI Teams

1. **Enterprise-grade modeling & management**
    - Use **Tabular Editor, ALM Toolkit, DAX Studio, SSMS** to connect directly to datasets.
    - Edit models, create measures, add calculation groups, manage partitions.
    - Perform source control and versioning of models (DevOps-friendly).
2. **Automation & Governance**
    - Automate refreshes, partitioning, and deployment via scripts.
    - Backup/restore datasets or migrate between workspaces.
    - CI/CD pipelines with tools like Azure DevOps.
3. **Advanced performance tuning**
    - Query performance analysis with DAX Studio.
    - Fine-grained control of partitions for **incremental refresh** beyond the UI.
    - Apply best-practice rulesets (via Tabular Editor) for consistent model design.
4. **Cross-tool ecosystem**
    - Since it uses the same protocol as Analysis Services, **Excel, SSRS, and third-party BI tools** can query the model directly.
    - This allows broader adoption of Power BI models across the enterprise.

Question-6

Usage Metrics in Power BI Service

What they are:
Built-in reports that show how reports and dashboards are being used.

Scope:

- **Available per report or dashboard (owners can see).**
    - Number of views over time
    - Unique users
    - Which users viewed content
    - Platforms used (web, mobile, etc.)

Audit Logs in Power BI Service

What they are:
A detailed log of user and admin activities (part of Microsoft 365 compliance/audit logging).

Scope:

- **Covers actions like:**
    - Viewing reports/dashboards
    - Sharing content
    - Exporting data
    - Publishing/deleting datasets
    - Changing permissions
- **Available for admins (Power BI Admins, Compliance/Audit admins).**

Question-7

In the new workspace experience (default today), there are four main roles:

1. **Viewer**
    - Can: View content, interact with reports/dashboards, export data (if allowed), use Analyze in Excel.
    - Cannot: Edit, publish, or share content.
    - Best for: End-users / consumers.
2. **Contributor**
    - Can: Create, edit, and publish content in the workspace. Refresh datasets, schedule refresh.
    - Cannot: Manage permissions or delete workspace.
    - Best for: Report builders / analysts.
3. **Member**

- Can: Do everything a Contributor can plus: share content, manage apps, approve requests.
- Cannot: Delete or upgrade the workspace.
- Best for: Power users / team leads.

4. **Admin**
    - Full control: Add/remove users, assign roles, publish/delete/update all content, manage refreshes, configure gateway connections.
    - Best for: Workspace owners / BI admins.

How to Manage Access

1. In the Power BI Service → Go to a **workspace** → **Settings → Permissions**.
2. Add users, security groups, or distribution lists.
3. Assign the appropriate role (Viewer, Contributor, Member, Admin).
4. Use **Azure Active Directory (AAD) groups** for easier management at scale.

Question-8

Power BI Service enforces data governance through a layered approach:

- Security (RLS, workspace roles, AAD groups)
- Compliance (sensitivity labels, tenant policies)
- Trust & Quality (certification, endorsement, pipelines)
- Monitoring (usage metrics, audit logs, Purview lineage)

Question-9

RLS in DirectQuery

Performance impact:

RLS filters are translated into WHERE clauses on the source system.

If the data source is slow, queries can be significantly slower, especially with many users or complex filters.

Limited modeling flexibility

Some advanced DAX expressions in RLS roles may not be supported, depending on the data source.

You may not be able to use certain calculated columns/tables because they aren’t materialized in DirectQuery mode.

Data source dependency

The effectiveness of RLS depends on how well the underlying database can handle security filtering (indexes, query optimization).

RLS in Live Connection

Defined in the source, not Power BI

When connecting live to Analysis Services (SSAS/ Azure AS) or another Power BI dataset, you cannot define RLS in Power BI Service.

Security roles must be created and managed in the source model.

Read-only model

Because Live Connection doesn't allow adding new tables or roles in Power BI, you're limited to whatever security definitions exist in the SSAS/semantic model**.**

Dependency on external admins

BI developers/analysts often rely on IT/DBA teams to configure RLS in SSAS, which can slow agility.

Question-10

Refresh via Power Automate

How it works: You can build a flow that triggers dataset refreshes without going into Power BI manually.

Steps:

1. In Power Automate, create a new flow.
2. Add a trigger (e.g., scheduled, when a file is updated in SharePoint, after ETL finishes, etc.).
3. Add the Power BI action → "Refresh a dataset".
4. Select Workspace and Dataset.
5. Optionally add a "Wait until refresh finished" action to handle success/failure.

Benefits:

Event-driven refresh (e.g., refresh after data load completes).

No need to rely only on fixed schedules in Power BI Service.

Easy for business users (low-code).

Refresh via REST API

How it works: Developers can call the Power BI REST API to trigger refreshes from scripts, applications, or DevOps pipelines.

Key endpoints:

POST https://api.powerbi.com/v1.0/myorg/groups/{groupId}/datasets/{datasetId}/refreshes → triggers a refresh.

GET …/refreshes → check refresh history/status.

Authentication:

Requires Azure AD token (OAuth 2.0).

Usually handled via Service Principal (app registration in Azure AD) or delegated user identity.

Authentication:

Requires **Azure AD token** (OAuth 2.0).

Usually handled via **Service Principal** (app registration in Azure AD) or delegated user identity.

Options:

Can specify whether the refresh is full or incremental (if configured).

Can set notification options (e.g., send completion status).

Benefits:

Fully automatable (scripts, CI/CD pipelines, Azure Functions).

Integrates with enterprise ETL/ELT processes (e.g., refresh Power BI dataset after Data Factory finishes).