

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

Power BI Online Service manages large datasets using data compression, incremental refresh, and optimized storage. Import mode is capped at 1 GB per dataset in shared capacity, but Premium increases this limit significantly. Premium capacity enables larger dataset sizes, up to 400 GB per dataset as of recent limits. It provides dedicated resources, reducing performance issues from noisy neighbors. Premium also supports advanced features like XMLA endpoint connectivity for enterprise-grade BI. It offers faster refresh rates, higher query concurrency, and larger model sizes. Incremental refresh helps handle billions of rows without reloading all data. The service manages partitions intelligently for performance optimization. Dataset eviction policies are relaxed in Premium compared to shared capacity. Overall, Premium is essential for enterprise scenarios with very large or complex data models.

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

Import mode stores a copy of the data inside the Power BI model, offering the fastest performance but limited by memory size. It supports full DAX functionality and offline use but requires scheduled refresh to stay updated. DirectQuery leaves the data in the source system and queries it live, ensuring always up-to-date results. DirectQuery can face performance issues depending on the source system responsiveness. It has limitations on DAX functions and calculated tables compared to Import. Live Connection connects directly to an Analysis Services model or Power BI dataset. In Live Connection, no data is imported; the model structure is defined by the source. It allows central semantic models but restricts creation of calculated columns or measures locally. Each mode balances performance, freshness, and flexibility differently. Choice depends on dataset size, governance requirements, and data latency needs.

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

Deployment pipelines in Power BI Online provide structured promotion of content across environments. They support DevOps practices by managing multiple versions of reports and datasets. Pipelines consist of three main stages: Development, Test, and Production. In Development, authors build and experiment with data models and reports. In Test, stakeholders validate accuracy, performance, and compliance. In Production, the finalized content is delivered to end users. Pipelines allow controlled promotion of updates from one stage to the next. They ensure consistency and reduce risk of errors during releases. Rules can map datasets to different data sources for each stage. This promotes agility while maintaining governance across the BI lifecycle.

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

Power BI integrates seamlessly with Microsoft Teams for collaborative analytics. Reports and dashboards can be pinned directly into Teams channels or chats. This allows users to access insights without leaving their collaboration environment. Notifications about report updates or comments can be shared in Teams. SharePoint Online supports embedding of Power BI reports in modern pages. This enables secure sharing of analytics across intranet portals. Both integrations respect Power BI Service security and RLS rules. Collaboration features encourage data-driven discussions among teams. Users can comment on reports and collaborate in real time. Integration ensures analytics are embedded where people work daily.

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

The XMLA endpoint in Premium provides open connectivity to Analysis Services tabular models. It allows external tools like SQL Server Management Studio (SSMS) or DAX Studio to connect directly. Developers can perform advanced model management tasks beyond the Power BI UI. It supports operations such as script-based deployments and metadata management. This enables integration with enterprise DevOps pipelines and CI/CD. It makes Power BI models interoperable with standard BI tooling. Data modeling, performance tuning, and debugging become more flexible. It empowers enterprise BI teams to apply traditional OLAP practices in Power BI. The endpoint supports both read and write operations, depending on licensing. Overall, it bridges Power BI with professional BI development environments.

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

Power BI usage metrics provide insights into report and dashboard adoption. Admins and creators can see how often content is viewed or shared. It helps identify popular reports and underutilized content. Audit logs, available in Microsoft 365 compliance center, provide detailed user actions. They track events like viewing, sharing, exporting, and deleting items. Audit logs help organizations meet compliance and monitoring requirements. Usage data supports capacity planning and user engagement strategies. Reports can be built on top of audit data for deeper analysis. Premium workspaces allow more granular usage reporting. Together, metrics and logs improve governance and operational visibility.

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

Workspaces in Power BI Service provide collaborative containers for BI content. Access is controlled using roles such as Viewer, Contributor, Member, and Admin. Viewer allows read-only access to reports and dashboards. Contributor permits report creation and dataset refresh but not permission management. Member has broader privileges including publishing content. Admin can manage permissions, settings, and connections. Permissions can be assigned to individuals or security groups. Azure Active Directory groups are often used for scalable management. Careful assignment ensures principle of least privilege is followed. This ensures both collaboration and governance are balanced in workspaces.

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

Data governance in Power BI Service involves policies, controls, and monitoring. Admins can enforce sensitivity labels to classify data. Information protection policies help prevent unauthorized sharing or downloads. Data loss prevention (DLP) rules can be integrated from Microsoft Purview. Row-Level Security ensures users see only the data relevant to them. Usage metrics and audit logs monitor compliance with policies. Capacity management helps allocate resources responsibly. Centralized certified datasets enforce a single source of truth. Workspaces can be structured to separate development and production. Governance balances security, compliance, and self-service analytics flexibility.

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

Row-Level Security (RLS) applies differently depending on storage mode. In Import mode, RLS filters are applied within the cached model. With DirectQuery, RLS is translated into SQL queries sent to the source. This can cause performance overhead on large datasets. Complex RLS rules may result in slow query response times. RLS in Live Connection depends on the underlying

Analysis Services model. If the source model defines roles, Power BI inherits them. Power BI cannot define new RLS rules on Live Connection datasets. This limits flexibility when consuming shared semantic models. Organizations must carefully design RLS to balance security and performance.

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

Power Automate provides an easy way to trigger dataset refreshes in Power BI. A flow can be created to run on a schedule or event-driven trigger. The Power BI connector includes an action to refresh a dataset in a workspace. Notifications can be configured to alert users on completion or failure. The REST API offers more granular programmatic control. Developers can trigger refreshes via scripts, custom apps, or pipelines. The API allows monitoring refresh status and history. It supports asynchronous refresh for large datasets. Both approaches help automate data freshness in enterprise solutions. They integrate well with DevOps and orchestration workflows.