

Power BI Service, Connectivity Modes, Security & Capstone

Take a report from Desktop to the people who need it. Understand Import, DirectQuery, Live connection, composite models and Direct Lake, publish to the Service, schedule refresh with a gateway, share through workspaces and apps, secure data with row-level security, tune performance, work with Copilot and Fabric, and complete the Adventure Cycles capstone.

● Power BI class 4

✦ Power BI Service & Microsoft Fabric 2026

Advanced

🕒 150 min

By the end of this lesson you can

- Compare Import, DirectQuery, Live connection, Composite and Direct Lake modes and choose one
- Publish a report, configure scheduled refresh and a gateway, and share via workspaces, apps and links
- Implement and test row-level security (static and dynamic) with DAX filters
- Use Performance Analyzer and best practices to speed up a model
- Describe Microsoft Fabric, Copilot and governance basics, and deliver an end-to-end capstone

Datasets used

adventure_cycles/fact_sales.csv

adventure_cycles/dim_customer.csv

adventure_cycles/dim_territory.csv

adventure_cycles/dim_product.csv

Connectivity modes

How does Power BI get its data: **copy it** into the model (Import) or **ask the source each time** (DirectQuery)?

Import vs DirectQuery

Import mode

- Data is copied into the model (VertiPaq)
- Very fast visuals, full DAX
- Needs scheduled refresh; data is as old as the last refresh
- Size limits: 1 GB Pro per model (up to 10 GB+ in Premium)
- Default and best for most reports

DirectQuery

- No data stored; queries run on the source
- Near-real-time data, huge tables stay in the database
- Slower visuals; source must be fast
- Some DAX and Power Query features limited
- Use for very large or live operational data

Import copies data into Power BI's compressed in-memory model; DirectQuery sends a query to the source for every interaction.

Mode	Data stored in Power BI?	Refresh	Typical use
Import	Yes (compressed copy)	Scheduled (up to 8/day Pro, 48/day Premium)	Most reports; Excel, CSV, SQL
DirectQuery	No	Live per interaction	Large SQL databases, near-real-time dashboards
Live connection	No: connects to an existing semantic model (Analysis Services or a published Power BI dataset)	Live	One governed model shared by many reports
Composite model	Mix: some tables Import, some DirectQuery, plus other semantic models	Both	Large fact in DirectQuery, small dimensions imported (dual storage mode)
Direct Lake (Fabric)	Reads Delta tables in OneLake directly	Automatic on data change	Import-like speed with no copy; Fabric only

TIP A practical rule

Start with **Import**. Move to DirectQuery or Direct Lake only if data volume or freshness requirements force you to. Reduce the data first: only needed columns, aggregated tables, filtered date range.

Publishing and the Service

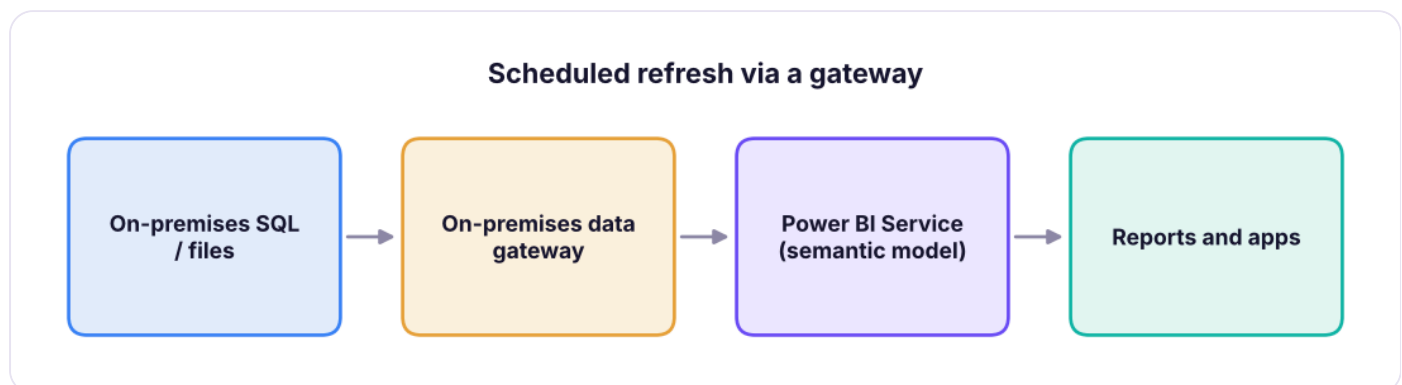
STEP BY STEP Publish a report

1. In Desktop: **Home** → **Publish** → sign in with a work/school account → choose a **workspace** (not "My workspace" if you want to share).
2. The Service now has a **semantic model** (the data and model) and a **report** (the visuals): the same objects in two items.
3. Open the report in the browser, check it, and **Share** with people or add it to an **app**.
4. Configure **scheduled refresh** (Settings → Semantic model → Scheduled refresh) with credentials; for on-premises sources install a **gateway**.

Workspaces, apps and sharing

Concept	Meaning
Workspace	A shared container for semantic models, reports, dashboards. Roles: Admin, Member, Contributor, Viewer
App	A packaged, read-only, curated set of reports/dashboards published to a wider audience (best way to distribute)
Share link	Give a person or group access to one report (they need appropriate licence)
Dashboard	One page of pinned tiles from several reports; supports alerts and mobile
Subscriptions & alerts	Email snapshots on a schedule; data alerts on card/gauge thresholds
Export / Analyze in Excel / Embed	PDF/PowerPoint export, live pivot tables on the model, secure embedding in portals
Deployment pipelines	Move content Dev → Test → Prod with controlled changes
Endorsement, sensitivity labels, lineage view	Governance: certified datasets, data classification, impact analysis

Refresh and gateways



A scheduled refresh for on-premises data goes through a gateway installed inside the company network.

- **Cloud sources** (Azure SQL, SharePoint Online, OneDrive files) refresh directly without a gateway.
- A **standard gateway** serves many users; a **personal gateway** is for one person's testing.
- **Incremental refresh** (right-click a table → Incremental refresh) loads only new or changed partitions (for example the last 7 days) instead of the whole fact table: essential for big tables. It needs `RangeStart` and `RangeEnd` parameters (Power Query).

- Refresh failures (expired credentials, changed schema, gateway offline) appear in the refresh history; enable failure notifications.

Row-level security (RLS)

RLS restricts the rows a user can see: a country manager sees only their country.

Static RLS

STEP BY STEP Create roles in Desktop

1. **Modeling** → **Manage roles** → **Create**. Name the role `India`.
2. Select the table `dim_territory` and add a DAX filter: `[country] = "India"`.
3. Create roles `UAE` and `Singapore` the same way. **Save**.
4. **View as** (Modeling → View as) → choose a role to test what that person sees.
5. Publish; in the Service: **Semantic model** → **Security** → add users or groups to each role.

Because filters flow from `dim_territory` to `fact_sales` through the relationship, the India role sees only India's sales everywhere:

Python

```
import pandas as pd
D = "adventure_cycles/"
f = pd.read_csv(D + "fact_sales.csv"); t = pd.read_csv(D + "dim_territory.csv"); m =
f.merge(t, on="territory_key")
role_view = m.groupby("country").agg(orders=("order_number", "count"), sales=
("sales_amount", "sum"))
role_view["sales"] = (role_view.sales / 1e6).round(1)
role_view.rename(columns={"sales": "sales (₹ M) visible to that role"}).assign(**{"all-
company total (₹ M)": round(f.sales_amount.sum() / 1e6, 1)})
```

OUTPUT

	orders	sales (₹ M) visible to that role	all-company total (₹ M)
country			
India	6135	406.9	594.6
Singapore	1487	96.7	594.6
UAE	1378	91.0	594.6

3 rows × 3 columns

Dynamic RLS

One role for everyone, driven by a **security table** mapping user emails to territories:

DAX

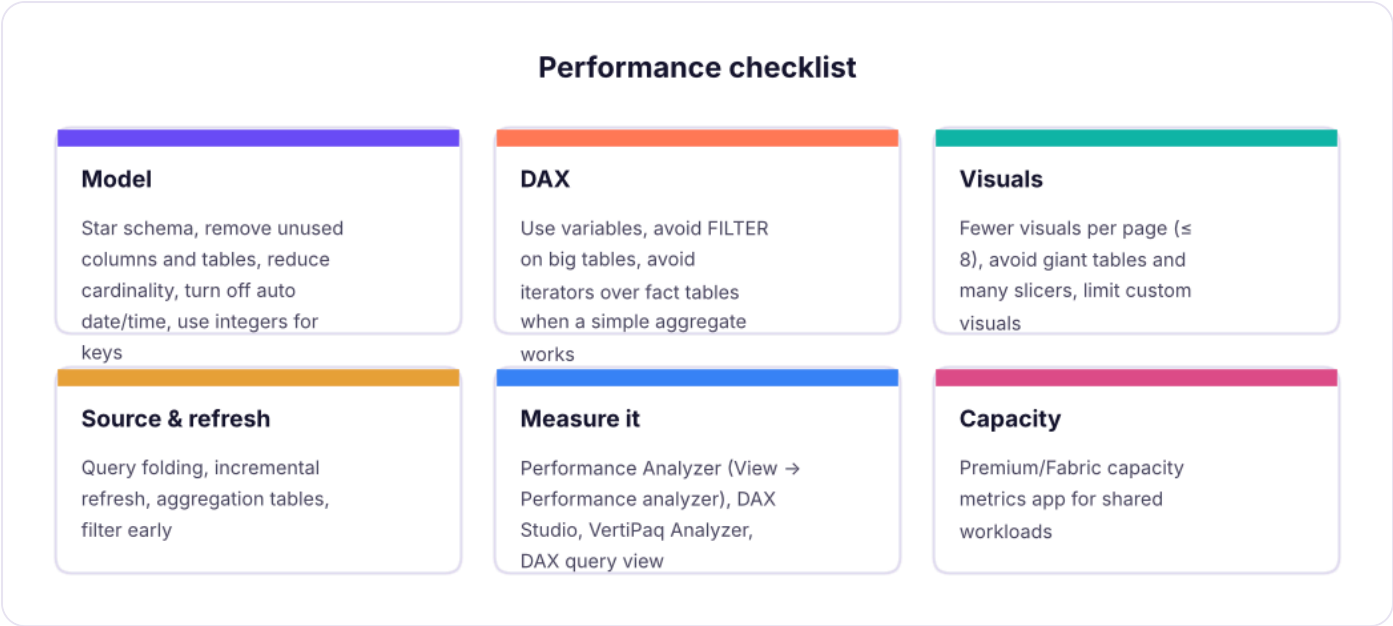
```
-- Table 'UserSecurity'(user_email, territory_key); role filter on dim_territory:
[territory_key]
IN CALCULATETABLE ( VALUES ( UserSecurity[territory_key] ),
                    UserSecurity[user_email] = USERPRINCIPALNAME () )
```

USERPRINCIPALNAME() returns the signed-in user's email. Add a person to the table and they get access automatically: no role edits.

WATCH OUT RLS pitfalls

- RLS applies to **Viewers** only. Workspace Admins, Members and Contributors can see all data.
- RLS does not hide column names or the existence of tables; use **object-level security** for that.
- Filters must flow along relationships: apply RLS to a **dimension**, not the fact, where possible.
- RLS with DirectQuery depends on the source's behaviour; test with *View as*.
- Do not store secrets or personal data you do not need in the model. In India, handling personal data is governed by the DPDP Act 2023.

Performance tuning



Where Power BI report time goes, and where to look.

Performance Analyzer (View → Performance analyzer → Start recording → refresh visuals) shows, for each visual, the time spent in **DAX query**, **Visual display** and **Other**. If DAX dominates, optimise the measure or model; if display dominates, simplify the visual.

Practical wins:

Problem	Fix
File is huge	Remove columns you do not use; store IDs as integers; split date and time; use Whole number/Fixed decimal rather than text
Slow card or matrix	Rewrite measures with variables; avoid nested FILTER (ALL (bigTable))
Slow refresh	Query folding, incremental refresh, filter at source
Slow DirectQuery	Aggregations table (Import) in front of the big table; index the source
Many identical measures	Calculation groups or one base measure referenced by others

Microsoft Fabric, Copilot and AI

Microsoft Fabric unifies data engineering, data factory, data warehouse, data science, real-time analytics and Power BI over one data lake called **OneLake** (Delta/Parquet files). Power BI's semantic models can use **Direct Lake** over lakehouse tables. For data professionals: learn SQL (this track), Python (next track), and then Fabric notebooks, pipelines and lakehouses.

Fabric item	Role
Lakehouse / Warehouse	Stores data (files + SQL endpoint)
Dataflow Gen2	Power Query in the cloud
Notebook	Python/Spark for data science and engineering
Pipeline	Orchestrates data movement
Semantic model	The Power BI model (formerly "dataset")
Eventstream / Real-time	Streaming data

Copilot in Power BI (needs a Fabric or Premium capacity): *create a report page* from a prompt, *suggest measures* and *write DAX*, *summarise* a page or report, and *answer questions* about the data. **Prep data for AI** (add descriptions, synonyms, verified answers) improves the answers. Treat it as a fast assistant, not an authority.

Governance and good practice

1. **One certified semantic model per subject area**, many thin reports on top (live connection).
2. Name items consistently (Sales - Overview, Sales_Model); keep an **owner** and a description.
3. Version control: **PBIP** + Git, deployment pipelines, a test workspace.
4. Document measures (description property) and keep a measure dictionary.
5. Limit **who can export** data; use sensitivity labels.
6. Monitor usage (usage metrics report) and remove unused reports.
7. Train users: the best dashboard is useless if nobody trusts or understands it.

Capstone: Adventure Cycles performance review

CASE STUDY The brief

The sales director wants a 3-page Power BI report for the Adventure Cycles leadership team and country managers: **(1) Executive summary, (2) Products and customers, (3) Country detail (drill-through)**. It must show YTD and vs-last-year numbers, let managers filter by year, country and category, and each country manager must see **only their own country**.

Deliverables checklist

Step	What to produce
1. Requirements	Audience, questions, KPIs, refresh frequency, security
2. Data	Load the five CSV tables; clean in Power Query
3. Model	Star schema, relationships, marked date table, hierarchies (Lessons 1–2)
4. Measures	Sales, Profit, Margin %, Orders, AOV, YTD, LY, YoY %, Rank, Top 5 + Others (Lessons 3–4)
5. Report	Three pages, theme, tooltips, drill-through, bookmarks, mobile layout (Lesson 5)
6. Security	RLS roles by country, tested with <i>View as</i>
7. Publish	Workspace, scheduled refresh, app, subscriptions (this lesson)
8. Documentation	One-page "how to read this report" and a measure dictionary

The analysis the report must support: verify your page numbers against this independent calculation.

Python

```
m = f.merge(t, on="territory_key").merge(pd.read_csv(D + "dim_product.csv"),
on="product_key"); m["profit"] = m.sales_amount - m.total_cost
m["order_date"] = pd.to_datetime(m.order_date); m["year"] = m.order_date.dt.year
kpi = m.groupby("year").agg(sales=("sales_amount", "sum"), profit=("profit", "sum"),
orders=("order_number", "nunique"))
kpi["margin_%"] = (kpi.profit / kpi.sales * 100).round(1); kpi["YoY_sales_%"] =
(kpi.sales.pct_change() * 100).round(1)
kpi[["sales", "profit"]] = (kpi[["sales", "profit"]] / 1e6).round(1)
kpi
```

OUTPUT

	sales	profit	orders	margin_%	YoY_sales_%
year					
2023	204.2	72.2	3004	35.4	NaN
2024	200.4	70.8	3022	35.3	-1.9
2025	189.9	67.2	2974	35.4	-5.2

3 rows × 5 columns

Python

```
c = m.groupby("country").agg(sales=("sales_amount", "sum"), profit=("profit", "sum"))
c["margin_%"] = (c.profit / c.sales * 100).round(1); c["share_%"] = (c.sales /
c.sales.sum() * 100).round(1); c[["sales", "profit"]] = (c[["sales", "profit"]] /
1e6).round(1)
bike_share = (m[m.category == "Bikes"].sales_amount.sum() / m.sales_amount.sum() *
100).round(1)
print(c.to_string()); print(f"\nBikes are {bike_share}% of total sales; Components
{m[m.category == 'Components'].sales_amount.sum() / m.sales_amount.sum() * 100:.1f}%.")
```

OUTPUT

	sales	profit	margin_%	share_%
country				
India	406.9	144.1	35.4	68.4
Singapore	96.7	34.1	35.2	16.3
UAE	91.0	32.0	35.1	15.3

Bikes are 78.5% of total sales; Components 18.4%.

KEY TAKEAWAYS

What a good leadership summary says

1. **Sales are shrinking, not growing:** ₹204.2 M (2023) → ₹200.4 M (2024, -1.9%) → ₹189.9 M (2025, -5.2%), while the margin holds steady at about 35%. The problem is **volume or demand**, not pricing or cost.
2. **Bikes** generate 78.5% of sales and **India** 68.4%, so any decline in Indian bike sales moves the whole company: a concentration risk.
3. Singapore (16.3% of sales) and the UAE (15.3%) earn the same ~35% margin as India, so growth there is as profitable as at home.
4. Recommended next steps: find where the 2025 decline comes from (by category, country and month, using the YoY measures from Lesson 4), set recovery targets per country, and track them monthly in the report.

(Verify each statement against the tables above before you present it; the numbers, not the template, are the story.)

Summary

KEY TAKEAWAYS

- **Import** is fast and the default; **DirectQuery** keeps data in the source (slower, near-live); **Live connection** reuses a shared semantic model; **Composite** mixes modes; **Direct Lake** reads Fabric Delta tables directly.
- Publish to a **workspace**, schedule refresh (gateway for on-premises, incremental refresh for big facts), distribute through **apps**, and use subscriptions and alerts.
- **RLS** filters rows by role (static) or by the signed-in user (dynamic, `USERPRINCIPALNAME()`); it applies to Viewers and must be tested with *View as*.
- Tune performance with a star schema, lean columns, efficient DAX and **Performance Analyzer**.
- Fabric unifies the data stack; Copilot speeds up building; governance (certified models, pipelines, labels) keeps it trustworthy.

Quick quiz

6 questions · tap an answer to check it

1. Which mode stores a compressed copy of the data inside the Power BI model?

A Import

B DirectQuery

C Live connection

D Streaming

Answer: A. Import loads the data into the in-memory VertiPaq model.

2. When is an on-premises data gateway needed?

- A For any Excel file
- B To schedule refresh from data sources inside a company network**
- C To publish reports
- D To use slicers

Answer: B. Gateways let the cloud service reach on-premises sources.

3. A user with the Admin role in the workspace is also in an RLS role. What do they see?

- A Only the role's rows
- B All data, because RLS does not apply to Admin, Member and Contributor roles**
- C Nothing
- D Only their own data

Answer: B. RLS applies to Viewers; test with View as.

4. What does incremental refresh do?

- A Refreshes slicers only
- B Refreshes only new or changed partitions of a large table**
- C Deletes old data
- D Converts the model to DirectQuery

Answer: B. It speeds up refresh by loading only recent data.

5. Which tool shows time spent in DAX vs visual display?

A Model view

B Performance Analyzer

C Selection pane

D Bookmarks

Answer: B. Performance Analyzer breaks down each visual's load time.

6. Which function enables dynamic row-level security by user?

A TODAY ()

B SELECTEDVALUE ()

C USERPRINCIPALNAME ()

D RELATED ()

Answer: C. It returns the logged-in user's email for comparison with a security table.

EXERCISE Deploy and secure

(1) Create RLS roles for the three countries and test each with *View as*. (2) Create a dynamic RLS design using a `UserSecurity` table (write the table and the DAX filter). (3) List five changes you would make to a 2 GB model that refreshes in 45 minutes. (4) Write the checklist you would give a colleague before publishing a report to leadership.

Show solution (worked answer)

(1) Manage roles → filter `dim_territory[country]`; View as each role and confirm the totals equal the pandas numbers above. (2) Table: `user_email`, `territory_key`; filter on `dim_territory: [territory_key] IN CALCULATETABLE ( VALUES ( UserSecurity[territory_key] ), UserSecurity[user_email] = USERPRINCIPALNAME ( ) )`. (3) Remove unused columns; star schema and integer keys; incremental refresh on the fact; push transformations to the source (query folding); aggregate to the needed grain; disable auto date/time. (4) Numbers reconciled with source; filters and slicers tested; titles state insights; RLS tested; refresh scheduled and tested; owner and description added; performance checked; accessibility and mobile layout checked; app published with notes.

INTERVIEW CORNER Questions you may be asked

Q: Import vs DirectQuery? Import copies and compresses data into the model (fast, needs refresh, size limits); DirectQuery queries the source live (fresh, handles huge data, slower, DAX limitations).

Q: What is row-level security and how do you implement it? Restricting the rows a user can see: define roles with DAX filters on tables (static), or use `USERPRINCIPALNAME` and a security table (dynamic); assign users in the Service; test with *View as*.

Q: How do you improve a slow Power BI report? Measure with Performance Analyzer; fix the model (star schema, fewer columns), optimise DAX, simplify visuals, use aggregations/incremental refresh, and reduce data volume.

Q: What is a semantic model and why use live connections? The data model (tables, relationships, measures); multiple reports can connect live to one governed model so everyone uses the same definitions.

Practise more online

Free, trusted places to test yourself on this topic. Links open the original site in a new tab.