



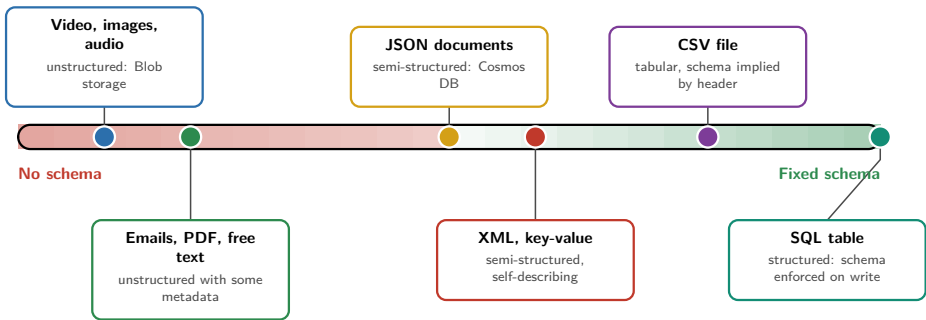
DP-900: Core Data Concepts

DP-900 domain 1, Describe core data concepts (25-30 %, as of 2026) | Structured, semi-structured and unstructured data, file formats, databases, transactional vs analytical workloads, data roles and services

DP-900 Domain 1 OLTP OLAP

Shapes of data

“How strictly is the structure defined before data is stored?”



Format	Layout	Best for	Notes
CSV / TSV	row, text	exchange, small extracts	no types, no compression, header row optional
JSON	document, text	APIs, documents, NoSQL	nested objects and arrays, field names repeated
XML	document, text	legacy integrations, SOAP	verbose tags, schemas (XSD)
Avro	row, binary	streaming, event serialisation	schema stored in the file; good for writes
Parquet	column, binary	analytics on data lakes	compression, reads only needed columns
ORC	column, binary	Hive / Hadoop analytics	similar to Parquet
Delta	Parquet + log	lakehouse tables (Fabric, Databricks)	ACID, time travel, MERGE

	columnar	not columnar
binary	Parquet ORC	Avro
not binary		CSV JSON XML

Columnar binary formats suit analytics; text formats suit exchange

Quantity	Formula
----------	---------

Columnar file size $S_p = \frac{S_c}{r}$

Data scanned by a query $D = S_p \times \frac{c}{C}$

where S_c = size of the same data as CSV (GB); r = compression ratio of Parquet vs CSV (ratio); S_p = Parquet size (GB); c = columns the query reads (count); C = columns in the table (assumed similar width) (count); D = data read by the query (GB)

Why columnar formats scan less data

Worked example: 100 million sales rows

100 M rows $\times$ 200 bytes = S_c = **20 GB** of CSV. With $r = 5$, S_p = **4 GB** of Parquet. A report that sums revenue by month reads $c = 3$ of $C = 20$ columns: $D = 4 \times 3 / 20 = \mathbf{0.6\ GB}$, against 20 GB for a full CSV scan: about 33 times less data, so faster queries and lower cost in engines that bill per scan.

📁 Data stores and workloads

File stores	Databases
Azure Blob Storage: objects of any format	Relational: Azure SQL, Azure Database for PostgreSQL / MySQL
Azure Data Lake Storage Gen2: Blob + hierarchical namespace (folders, POSIX ACLs) for analytics	Non-relational (NoSQL): Azure Cosmos DB (document, key-value, column-family, graph)
Azure Files: SMB / NFS file shares	Analytical: Fabric Warehouse, Synapse dedicated SQL pool, Azure Databricks SQL
Fabric OneLake: one logical lake per tenant, Delta tables	Real-time: Fabric Eventhouse (KQL database), Azure Data Explorer

Transactional (OLTP)	vs	Analytical (OLAP)
many small inserts and updates	Operations	large reads and aggregations
applications, thousands of users	Users	analysts, reports, data scientists
current state, normalised	Data	history, denormalised (star schema)
✓ ACID per transaction	Guarantee	consistent snapshots, bulk loads
slow on big scans	Query speed on aggregates	columnar, parallel, fast ✓
Azure SQL Database, Cosmos DB	Azure examples	Fabric Warehouse and lakehouse, Power BI

Mnemonic: **ACID**

- A Atomicity** a transfer debits and credits, or does neither
- C Consistency** constraints hold before and after
- I Isolation** concurrent transactions do not see each other's partial work
- D Durability** committed data survives a crash (log on disk)

📖 EXAM TIP

"Many concurrent small updates, must never lose a committed order" = OLTP with ACID.
"Historical data, aggregated for reports" = OLAP / analytical. A **data lake** stores raw files; a **data warehouse** stores modelled relational tables; a **lakehouse** is both on open Delta files.

👤 Data roles and services

Task	DBA	Data engineer	Data analyst
Back up and restore databases	R A	I	
Grant database permissions	R A		C
Build ingestion pipelines	C	R A	
Clean and model data in the lakehouse		R A	C
Design the Power BI semantic model		C	R A
Build reports and dashboards			R A
Tune slow queries	R A	C	I

R Responsible A Accountable C Consulted I Informed

Azure service	Main role	Purpose
Azure SQL Database / Managed Instance	DBA	managed relational OLTP databases
Azure Database for PostgreSQL / MySQL	DBA	managed open-source relational databases
Azure Cosmos DB	developer, DBA	globally distributed NoSQL
Azure Storage / ADLS Gen2	data engineer	files, data lake
Azure Data Factory, Fabric Data Factory	data engineer	pipelines and data flows (ETL / ELT)
Microsoft Fabric	engineer, analyst	lakehouse, warehouse, real-time, Power BI in one SaaS
Azure Databricks	engineer, scientist	Apache Spark platform, Delta Lake
Microsoft Purview	governance	data catalogue, lineage, classification
Power BI	data analyst	semantic models, reports, dashboards

YOU MAY HEAR
"JSON is unstructured"

BUSTED

✓ **ACTUALLY**
JSON is semi-structured
Records describe their own fields, which may vary.

YOU MAY HEAR
"A data lake replaces every database"

BUSTED

✓ **ACTUALLY**
Applications still need OLTP databases
Lakes and lakehouses serve analytics; transactions run on Azure SQL or Cosmos DB.

YOU MAY HEAR
"CSV is fine for big analytics"

BUSTED

✓ **ACTUALLY**
Columnar Parquet / Delta scan far less
No compression or types in CSV, every query reads every column.

YOU MAY HEAR
"Data analysts build ingestion pipelines"

BUSTED

✓ **ACTUALLY**
That is the data engineer's job
Analysts model and visualise data in Power BI.

Self-test and recap

Q1 Photos and videos are which kind of data?

A Unstructured.

Q2 Which format is columnar and compressed for analytics?

A Parquet (also ORC; Delta builds on Parquet).

Q3 Which format stores its schema and suits streaming serialisation?

A Avro.

Q4 Name the four ACID properties.

A Atomicity, consistency, isolation, durability.

Q5 Many small concurrent writes from an app: OLTP or OLAP?

A OLTP.

Q6 Store raw files of any type cheaply for analytics?

A Data lake (ADLS Gen2 or OneLake).

Q7 Tables with ACID on open files in a lake?

A Lakehouse with Delta Lake.

Q8 Who designs Power BI reports?

A Data analyst.

Q9 Data processed once a night in one run?

A Batch processing.

TOPIC

Core data concepts

HEADLINE 1

Structure is a spectrum

Why: unstructured → semi-structured → structured

HEADLINE 2

Columnar wins for analytics

Why: Parquet and Delta read only needed columns

HEADLINE 3

OLTP writes, OLAP reads

Why: ACID transactions vs aggregations

HEADLINE 4

Lake, warehouse, lakehouse

Why: files, modelled tables, both

HEADLINE 5

Three roles

Why: DBA, data engineer, data analyst