Cheatsheet – Day 1

Platform, ELT Ingestion, Delta Fundamentals

Unity Catalog Hierarchy

```
Metastore
  └── Catalog
        └── Schema (Database)
              ├── Table (Managed / External)
              ├── View
              ├── Function
              └── Volume (Managed / External)
```

Cluster Types

Type	Purpose
All-Purpose	Interactive dev, notebooks
Job (Jobs Compute)	Automated job execution
SQL Warehouse	SQL queries, BI dashboards

Access Mode: Single User (full libs) vs Shared (governed, Unity Catalog enforced)

Reading Files

```sql
-- CSV with explicit schema
SELECT * FROM read_files('/path/to.csv',
    format => 'csv',
    header => 'true',
    sep => ',');

-- JSON
SELECT * FROM read_files('/path/to.json', format => 'json');
```

```python
df = spark.read.format("csv") \
    .option("header", "true") \
    .option("inferSchema", "true") \
    .load("/path/to.csv")
```

Creating Tables (CTAS)

```sql
CREATE TABLE catalog.schema.my_table
AS SELECT * FROM read_files('/path', format => 'csv', header => 'true');
```

Temp Views & SQL

```python
df.createOrReplaceTempView("my_view")
result = spark.sql("SELECT * FROM my_view WHERE col > 10")
```

Common Transformations

```
df.select("col1", "col2")
df.filter(col("price") > 0)
df.withColumn("total", col("qty") * col("price"))
df.withColumnRenamed("old", "new")
df.drop("unwanted_col")
df.dropDuplicates(["id"])
df.na.fill({"col": "default"})
```

Delta Lake Basics

- **Format:** Parquet + JSON transaction log (`_delta_log/`)
- **ACID:** Atomic, Consistent, Isolated, Durable
- **Default:** All tables in Databricks are Delta by default

```
DESCRIBE HISTORY my_table;
SELECT * FROM my_table VERSION AS OF 3;
SELECT * FROM my_table TIMESTAMP AS OF '2025-01-01';
RESTORE TABLE my_table TO VERSION AS OF 3;
```

Key Exam Points

- Unity Catalog is **deny-by-default**
- `USE CATALOG` + `USE SCHEMA` required before `SELECT`
- Managed table: data deleted on DROP
- External table: data survives DROP
- Volumes: manage non-tabular files (CSV, JSON, images)
- Delta transaction log: one JSON file per commit