Praveen Madupu
Mb: +91 98661 30093
Sr. Database Administrator

1️⃣ What is a Query Store?

- **Query Store** is a SQL Server feature (introduced in **SQL Server 2016**) that **captures a history of executed queries, their execution plans, and runtime statistics**.
- Think of it as a **"flight recorder"** for query performance — it stores:
  - Which queries ran
  - What execution plans were used
  - How long they took
  - How performance changed over time
- **Why it's useful**:
  - Identify performance regressions (e.g., query runs slower after an index change).
  - Force SQL Server to use a known good execution plan.
  - Troubleshoot intermittent performance issues.

2️⃣ How Query Store Works

- Data is stored **inside the user database** (not tempdb).
- Two main stores:
  1. **Plan Store** → Keeps query text and execution plans.
  2. **Runtime Stats Store** → Keeps aggregated execution statistics over time.

3️⃣ Enabling Query Store

By default, the Query Store is **OFF** before SQL Server 2022. From 2022, it's **ON by default** for new databases.

```
ALTER DATABASE YourDatabase

SET QUERY_STORE = ON;


ALTER DATABASE YourDatabase

SET QUERY_STORE (OPERATION_MODE = READ_WRITE);
```

Praveen Madupu
Mb: +91 98661 30093
Sr. Database Administrator

4️⃣ Configuring Query Store

```
ALTER DATABASE YourDatabase

SET QUERY_STORE = ON

(

    OPERATION_MODE = READ_WRITE,

    CLEANUP_POLICY = (STALE_QUERY_THRESHOLD_DAYS = 30),

    DATA_FLUSH_INTERVAL_SECONDS = 900,

    MAX_STORAGE_SIZE_MB = 100, INTERVAL_LENGTH_MINUTES = 60

);
```

Key settings:

- **STALE_QUERY_THRESHOLD_DAYS** → Keep query history for X days.
- **MAX_STORAGE_SIZE_MB** → Storage limit inside the database.
- **INTERVAL_LENGTH_MINUTES** → Granularity for runtime stats.

5️⃣ Viewing Query Store Data

Find Top Resource-Consuming Queries

```
SELECT TOP 10

    qsrs.avg_duration,

    qsrs.last_execution_time,

    qt.query_sql_text,

    qsp.query_plan

FROM sys.query_store_runtime_stats qsrs

JOIN sys.query_store_plan qsp

    ON qsrs.plan_id = qsp.plan_id

JOIN sys.query_store_query qsq

    ON qsp.query_id = qsq.query_id

JOIN sys.query_store_query_text qt

    ON qsq.query_text_id = qt.query_text_id

ORDER BY qsrs.avg_duration DESC;
```

Praveen Madupu
Mb: +91 98661 30093
Sr. Database Administrator

Identify Queries with Multiple Plans

```
SELECT qt.query_sql_text, COUNT(DISTINCT qsp.plan_id) AS plan_count

FROM sys.query_store_query_text qt

JOIN sys.query_store_query qsq

    ON qt.query_text_id = qsq.query_text_id

JOIN sys.query_store_plan qsp

    ON qsq.query_id = qsp.query_id

GROUP BY qt.query_sql_text

HAVING COUNT(DISTINCT qsp.plan_id) > 1;
```

💡 Useful for spotting **plan regressions**.

6️⃣ Forcing a Specific Execution Plan

If a query is running slower due to a bad plan choice, you can force the good plan:

```
EXEC sp_query_store_force_plan @query_id = 42, @plan_id = 7;
```


To unforce:

```
EXEC sp_query_store_unforce_plan @query_id = 42, @plan_id = 7;
```

7️⃣ Removing Query Store Data

- Clear all data: `ALTER DATABASE YourDatabase SET QUERY_STORE CLEAR;`
- Disable Query Store: `ALTER DATABASE YourDatabase SET QUERY_STORE = OFF;`

8️⃣ Real-World Usage Scenario

Problem:

A stored procedure suddenly runs 10× slower after an index rebuild.

Without Query Store:

You'd struggle to find the old plan.

With Query Store:

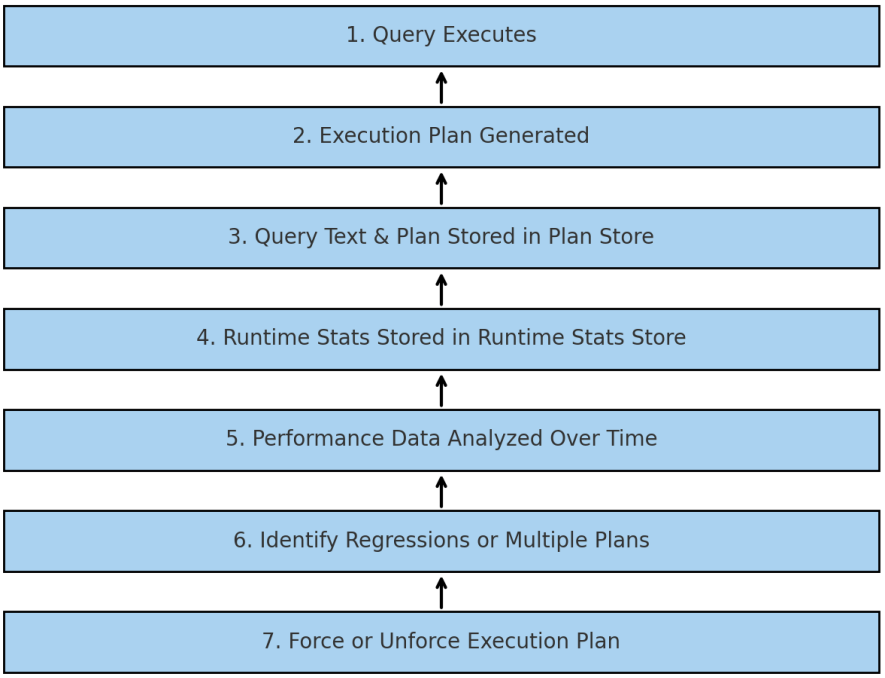
1. Check historical performance for the stored procedure.
2. Identify the good plan from before the index rebuild.
3. Force SQL Server to use that plan until you fix statistics or indexing.

Praveen Madupu
Mb: +91 98661 30093
Sr. Database Administrator

9 Query Store in SQL Server Versions

SQL Server Version	Query Store Behavior
2016	First introduced; manual enablement required.
2017	Enhanced DMVs, adaptive query plans support.
2019	Works with **Accelerated Database Recovery**; more metadata views.
2022	**Enabled by default** for all databases; integrates with **read replicas** in Always On AG.

SQL Server Query Store Workflow



Here's the **Query Store workflow diagram** — it shows how SQL Server captures queries, stores plans, tracks performance, and lets you force good plans when regressions occur.