In Chrome:

To find plan performance regression in SQL Server Query Store and force the last good query plan

<https://www.google.com/search?q=how+to+find+plan+performance+regression+in+query+store+and+force+last+good+query+plan&oq=how+to+find+plan+performance+regression+in+query+store+and+force+last+good+query+plan&gs_lcrp=EgZjaHJvbWUyBggAEEUYOTIHCAEQIRiPAjIHCAIQIRiPAjIHCAMQIRiPAtIBCjY3NDkyajBqMTWoAgiwAgE&sourceid=chrome&ie=UTF-8>

To find plan performance regression in SQL Server Query Store and force the last good query plan, you can use the Query Store views within SQL Server Management Studio to identify queries with significantly worsened execution times compared to previous executions, then use the FORCE\_PLAN option to explicitly specify the plan handle of the previously optimal plan you want to use.

Steps:

* **Identify potential regressions:**
  + **Access Query Store views:** In SSMS, navigate to the "Query Store" section and use system views like *sys.dm\_db\_query\_store\_plan\_stats* or *sys.dm\_db\_query\_store\_query* to analyze query execution data.
  + **Compare execution times:** Look for queries where the average or recent execution time has significantly increased compared to previous execution periods.
  + **Analyze plan changes:** Check the "Plan ID" column to see if the query is now using a different execution plan than before.
* **Find the "last good" plan:**
  + **Filter by plan handle:** For the identified regressed query, use the plan handle from a previous execution with a good performance to target the desired plan.
  + **Review plan details:** Use *sys.dm\_exec\_query\_plan* to analyze the specific operations and cost of different plans associated with the query.
* **Force the last good plan:**
  + Use FORCE\_PLAN option: When executing the query, add the OPTION (FORCE\_PLAN = <plan\_handle>) clause, replacing <plan\_handle> with the plan handle of the desired execution plan.
  + **Example query:**

Code

SELECT \* FROM yourTable WHERE ... OPTION (FORCE\_PLAN = <plan\_handle>);

Important considerations:

* **Query Store configuration:**

Ensure the Query Store is enabled and configured appropriately to capture sufficient historical data.

* **Plan stability:**

While forcing a plan can resolve immediate performance issues, consider the underlying cause of the regression and potentially optimize the database schema or query to avoid future plan regressions.

* **Monitoring and review:**

Regularly monitor query performance and review any forced plans to ensure they are still optimal and not causing new issues.