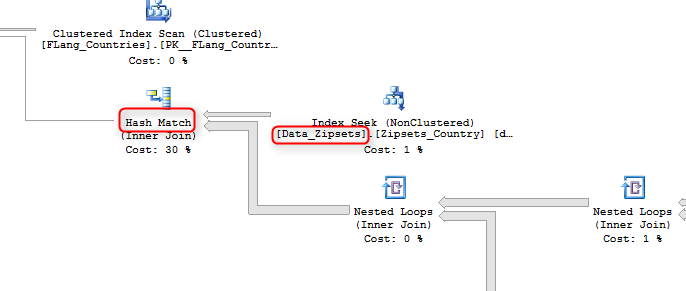
Not sure why SQL Server decided to do a Hash Match, which usually is slower than a Loop.



You can force it to do a Loop by using “INNER LOOP JOIN”, highlighted in yellow background. But understanding that, by doing this, you are telling SQL Server to always use Loop join between Data\_Zipsets and Data\_Lanes.

USE tbc\_chemours;

DECLARE @user\_id int =5

DECLARE @lang\_id int = 1

SELECT  co.country\_id

      , co.country\_name

FROM    FLang\_Countries co

WHERE   co.language\_id = @lang\_id

        AND EXISTS ( SELECT     1

                     FROM       Data\_Zipsets dz WITH ( NOLOCK )

                     INNER LOOP JOIN       Data\_Lanes dl WITH ( NOLOCK ) ON dz.zipset\_id = dl.orig\_zipset\_id

                     JOIN       Data\_Rates dr WITH ( NOLOCK ) ON dr.lane\_id = dl.lane\_id

                     JOIN       Data\_Contract\_Mode\_Map dcmm ON dr.contract\_id = dcmm.contract\_id

                     JOIN       Adm\_User\_Mode\_Map aumm ON aumm.mode\_id = dcmm.mode\_id

                                                          AND aumm.user\_id = @user\_id

                     JOIN       Fmk\_Modes fm ON dcmm.mode\_id = fm.mode\_id

                                                AND fm.rate\_upload\_template\_id = 2

                     WHERE      dz.country\_id = co.country\_id )

ORDER BY co.country\_name;

----------------------------------------------------------------------------------------------------------------------

References: <https://www.mssqltips.com/sqlservertip/2917/sql-server-join-hints/>

**LOOP JOIN**  
Query has a small table on the left side of the join   
One or both tables are indexed on the JOIN predicate

**HASH JOIN**  
Tables are fairly evenly-sized or are large  
Indexes practically irrelevant unless filtering on additional WHERE clauses, good for heaps  
Arguably most versatile form of join

**REMOTE JOIN**  
Same as hash join, but good where right side is geographically distant  
Only suitable for INNER JOINs  
Not suitable for local tables, will be ignored.

**MERGE JOIN**  
Tables are fairly even in size  
Works best when tables are well-indexed or pre-sorted  
Uses very efficient sort algorithm for fast results  
Unlike hash join, no memory reallocation, good for parallel execution