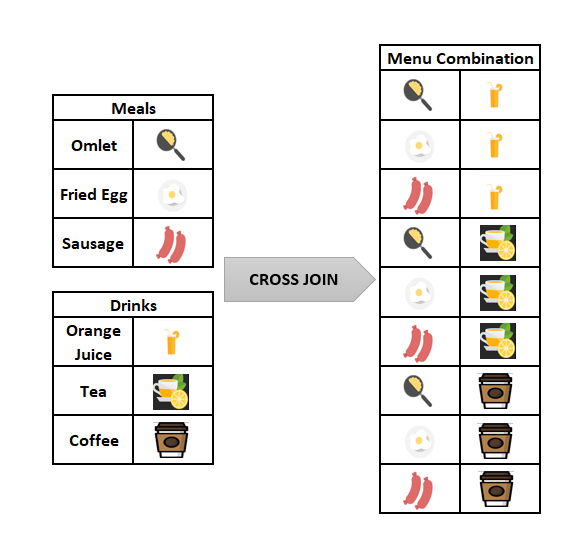
Introduction

The CROSS JOIN is used to generate a paired combination of each row of the first table with each row of the second table. This join type is also known as cartesian join.

Suppose that we are sitting in a coffee shop and we decide to order breakfast. Shortly, we will look at the menu and we will start thinking of which meal and drink combination could be more tastier. Our brain will receive this signal and begin to generate all meal and drink combinations.

The following image illustrates all menu combinations that can be generated by our brain. The SQL CROSS JOIN works similarly to this mechanism, as it creates all paired combinations of the rows of the tables that will be joined.



*“Please don’t worry, even if you feel a bit hungry now, you can eat whatever you want after reading our article.”*

The main idea of the CROSS JOIN is that it returns the Cartesian product of the joined tables. In the following tip, we will briefly explain the Cartesian product;

***Tip:****What is the Cartesian Product?*

*The Cartesian Product is a multiplication operation in the set theory that generates all ordered pairs of the given sets. Suppose that, A is a set and elements are {a,b} and B is a set and elements are {1,2,3}. The Cartesian Product of these two A and B is denoted AxB and the result will be like the following.*

*AxB ={(a,1), (a,2), (a,3), (b,1), (b,2), (b,3)}*

Syntax

The syntax of the CROSS JOIN in SQL will look like the below syntax:

| 1  2  3  4  5 | SELECT ColumnName\_1,         ColumnName\_2,         ColumnName\_N  FROM [Table\_1]       CROSS JOIN [Table\_2] |
| --- | --- |

Or we can use the following syntax instead of the previous one. This syntax does not include the CROSS JOIN keyword; only we will place the tables that will be joined after the FROM clause and separated with a comma.

| 1  2  3  4 | SELECT ColumnName\_1,         ColumnName\_2,         ColumnName\_N  FROM [Table\_1],[Table\_2] |
| --- | --- |

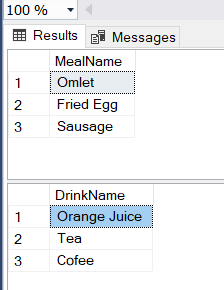
The resultset does not change for either of these syntaxes. In addition, we must notice one point about the CROSS JOIN. Unlike the INNER JOIN, LEFT JOIN and FULL OUTER JOIN, the CROSS JOIN does not require a joining condition.

SQL CROSS JOIN example:

In this example, we will consider the breakfast menu example again, which we mentioned in the earlier part of the article. Firstly, we will create the two-sample tables which contain the drink and meal names. After then, we will populate them with some sample data.

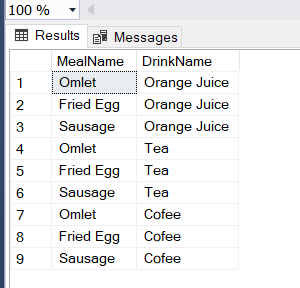
Through the following query, we will perform these two-steps:

| 1  2  3  4  5  6  7  8  9  10 | CREATE TABLE Meals(MealName VARCHAR(100))  CREATE TABLE Drinks(DrinkName VARCHAR(100))  INSERT INTO Drinks  VALUES('Orange Juice'), ('Tea'), ('Cofee')  INSERT INTO Meals  VALUES('Omlet'), ('Fried Egg'), ('Sausage')  SELECT \*  FROM Meals;  SELECT \*  FROM Drinks |
| --- | --- |

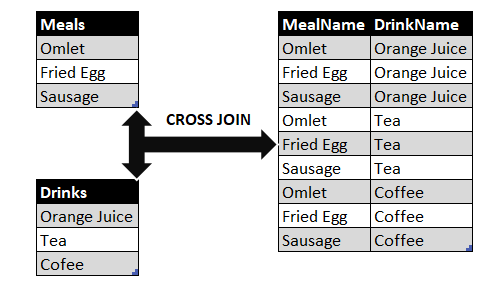


The following query will join the Meals and Drinks table with the CROSS JOINkeyword and we will obtain all of the paired combinations of the meal and drink names.

| 1  2 | SELECT \* FROM Meals  CROSS JOIN Drinks |
| --- | --- |

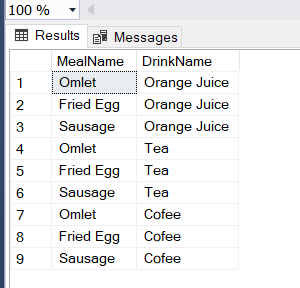


The below image illustrates the working principle of the CROSS JOIN.



At the same time, we can use the following query in order to obtain the same result set with an alternative syntax without CROSS JOIN.

| 1  2 | SELECT \* FROM Meals  ,Drinks |
| --- | --- |

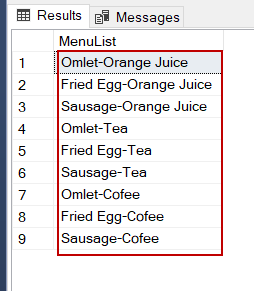


***Tip:****The resultset row count will equal to multiplication of tables row counts that will be joined. For the breakfast menu example, the Meals table row count is 3 and the Drinks table row count is 3, so the resultset row count can find with the following calculation.*

***3****(Meals table row count) x****3****(Drinks table row count) =****9****(Resultset row count)*

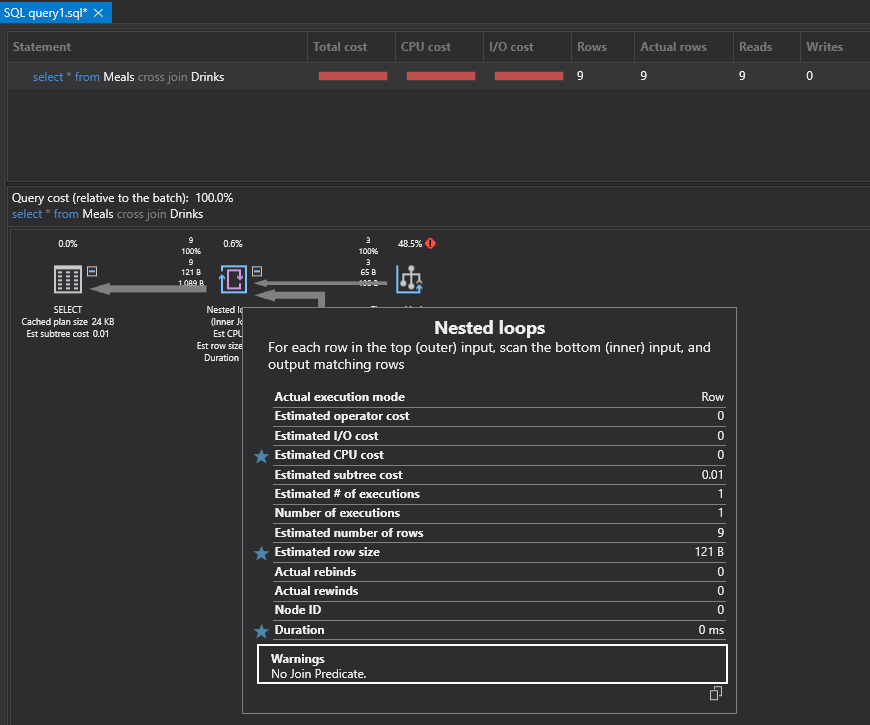
[CONCAT\_WS](https://docs.microsoft.com/en-us/sql/t-sql/functions/concat-ws-transact-sql?view=sql-server-ver15) function will help to concatenate the column expressions. Thus, we can create a more meaningful breakfast menu resultset.

| 1  2  3 | SELECT CONCAT\_WS('-',MealName,DrinkName) AS MenuList  FROM Meals CROSS JOIN  Drinks |
| --- | --- |



SQL CROSS JOIN and Performance Considerations

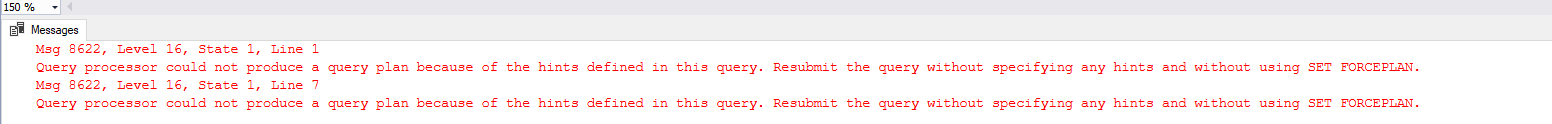
The SQL queries which contain the CROSS JOIN keyword can be very costly. We try to say that these queries have a high potential to consume more resources and can cause performance issues. For the following query, we will analyze the execution plan with [ApexSQL Plan](https://www.apexsql.com/sql-tools-plan.aspx). In the generated actual execution plan, we will see a Nested loops operator and when we hover over the mouse on this operator, the detail pop-up window will appear.



In this window, a warning message leaps to our eyes. “No Join Predicate” message specifies that this query can be faced with performance problems. For this reason, the query optimizer warns us about this potential problem. Briefly, when we decide to use the CROSS JOIN in any query, we should consider the number of the tables that will be joined. Such as, when we CROSS JOIN two tables and if the first one contains 1000 rows and the second one contains 1000 rows, the row count of the resultset will be 1.000.000 rows.

***Tip:****CROSS JOIN can only be implemented with Nested Loops, so the following queries will return an error if we force Query Optimizer to use other join types.*

| 1  2  3  4  5  6  7  8  9 | SELECT \* FROM Meals  CROSS JOIN Drinks  OPTION (MERGE  JOIN )    GO    SELECT \* FROM Meals  CROSS JOIN Drinks  OPTION (HASH  JOIN ) |
| --- | --- |



Conclusion

In this article, we learned SQL CROSS JOIN basics with details and we also mentioned the performance considerations of the CROSS JOIN. When CROSS JOIN is used for tables that have a high number of rows, it might affect the performance negatively.