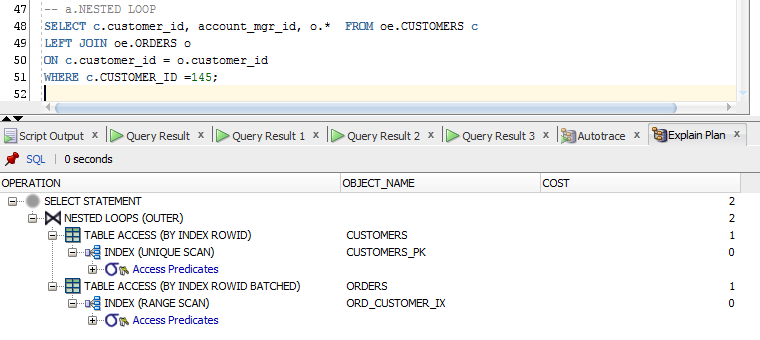
Oracle Join Methods

**2. Nested Loop**

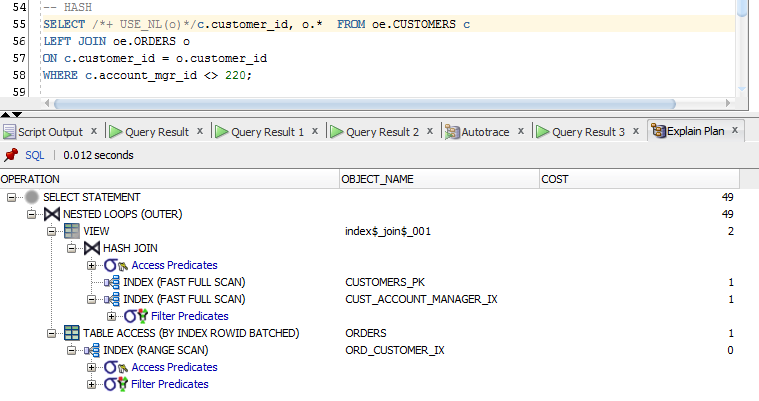
Nested loop is a join method that is used for small sources of data and there is an index in the condition or select clause.

Nested loop uses cycles to find the appropriate value in the 2nd value from the value from the 1st table. In our example it uses customer\_id from Customers and moves to for cycle in the table orders to find the same customer\_id. It also uses PK key – Customer\_ID.



Forced Nested Loop

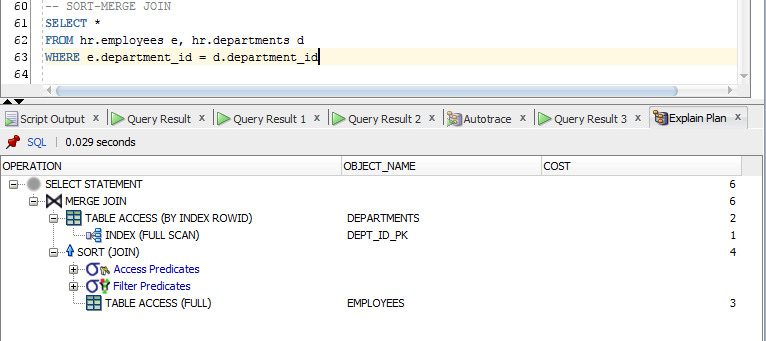
We can force to do nested loop when oracle uses Hash Join, but it only provides worse cost.



\* Account\_mgr\_id – index

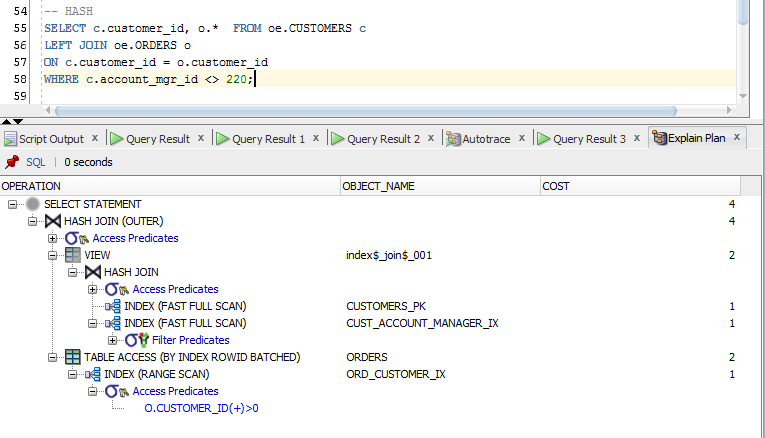
**3. Merge Sorted Join**

Merge sorted join sorts both datasets and then join rows. In our example for every department\_id from employees, which is the sorted table, Oracle search the appropriate row from department\_id from departments. It can cause better performance because of “hard” SORT operation of the first table.



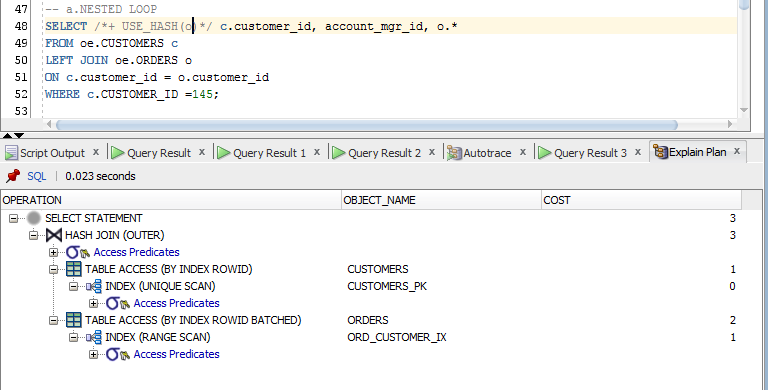
**4. Hash Join**

Hash join is also used for large subsets of data. Hash join puts data from the small table – orders in hash and uses large table to find appropriate rows in the handled orders table.



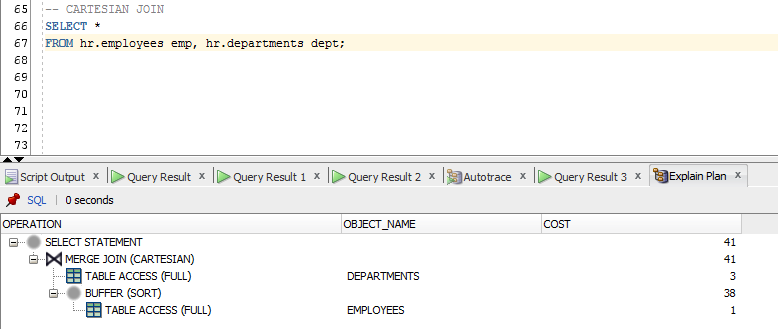
\* Account\_mgr\_id – index

Forced Hash Join



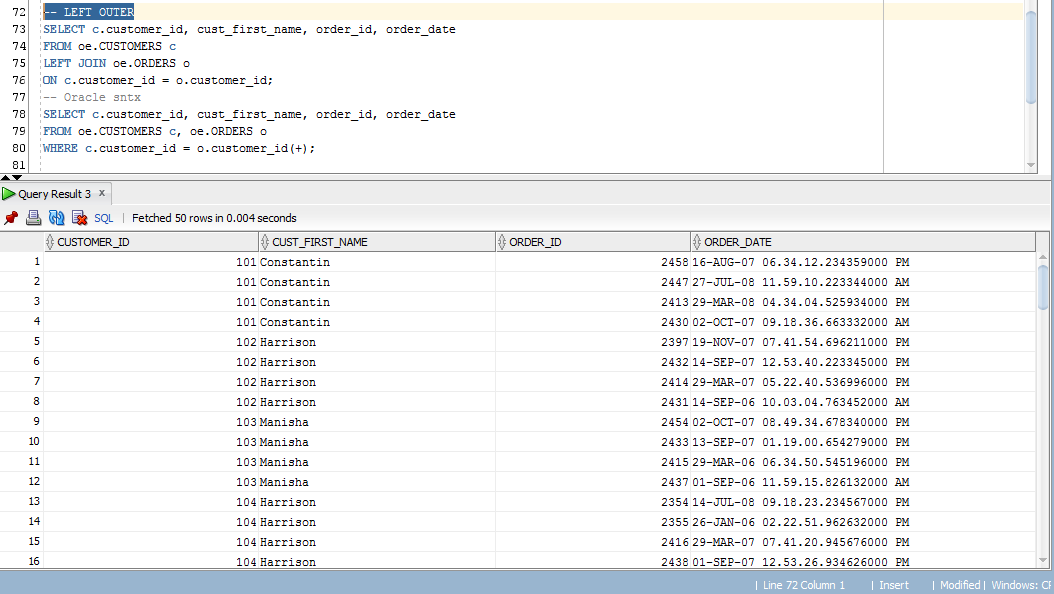
**5. Cartesian Join**

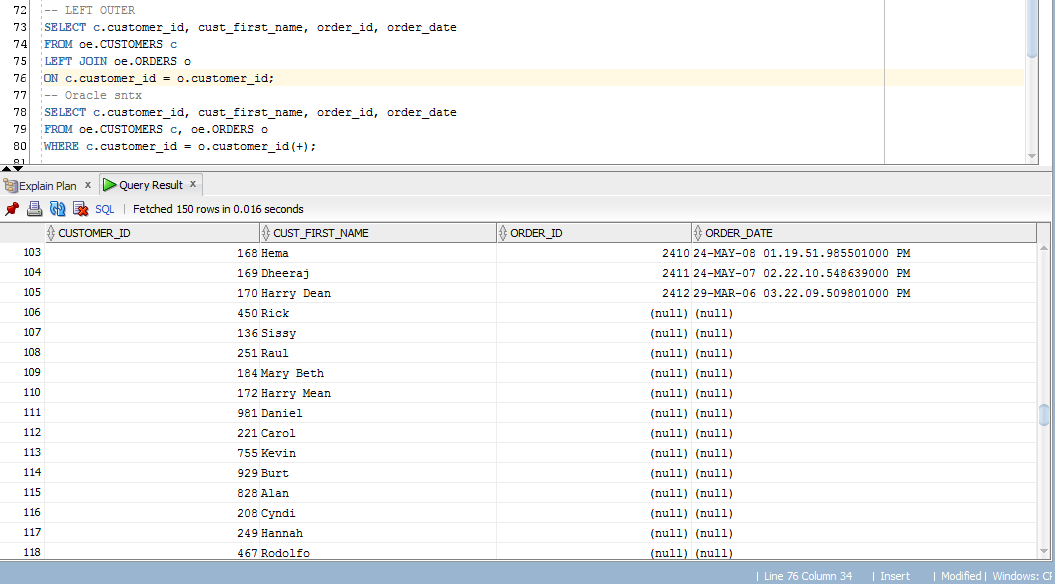
Cartesian join appears when there is a need to relate two tables without appropriate conditions. Our example shows that there is no field used in where clause that helps to relate tables Employees and Departments.



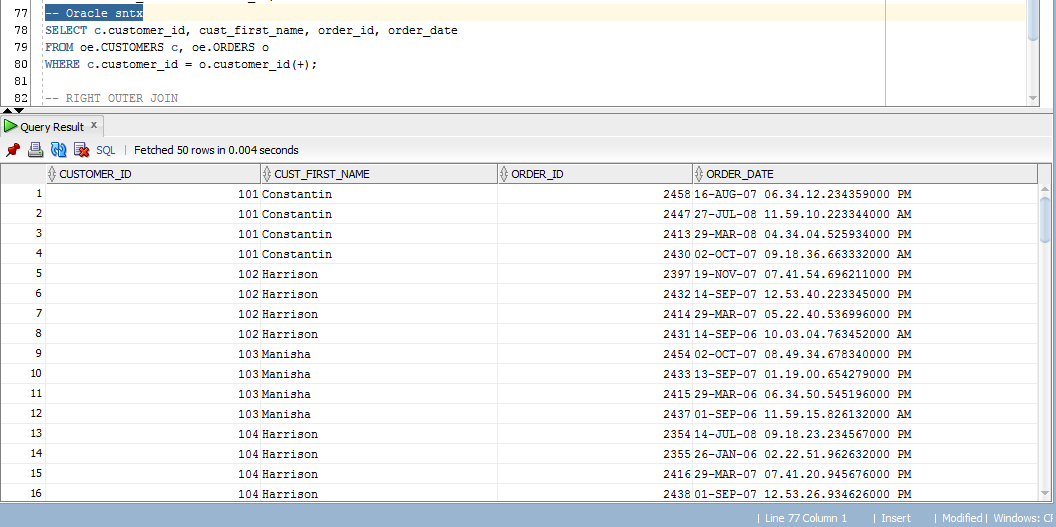
**LEFT OUTER JOIN**

Left join simply adds values from table in LEFT JOIN clause to the table in FROM clause and if the value is null it just turn “null” in that row. In our example we have orders that are join to customers, and there is amount of customers with null orders and with the help of LEFT JOIN we see “null” in such records.



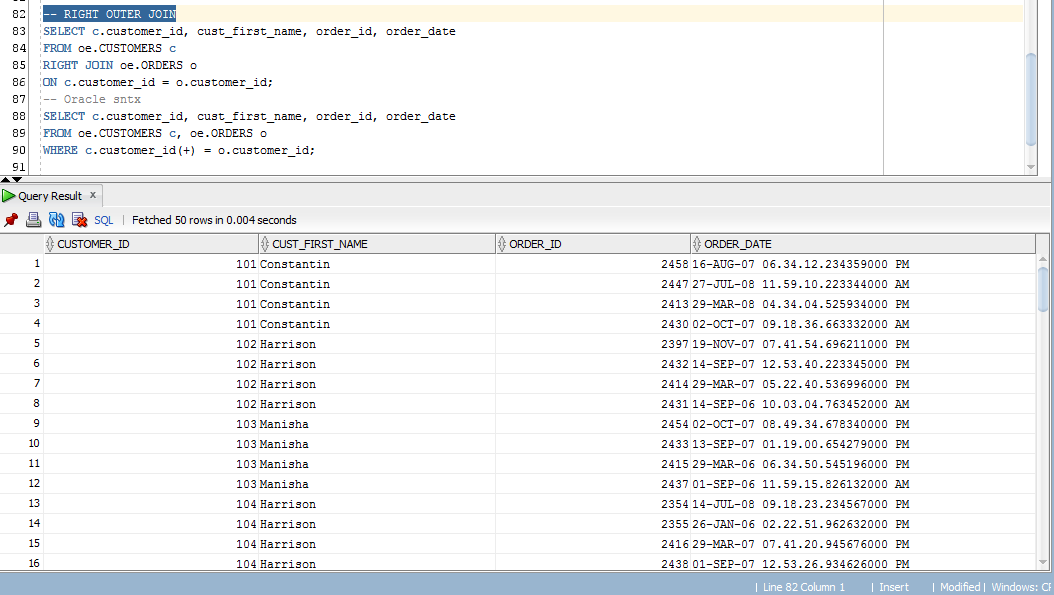


+ Oracle sntx

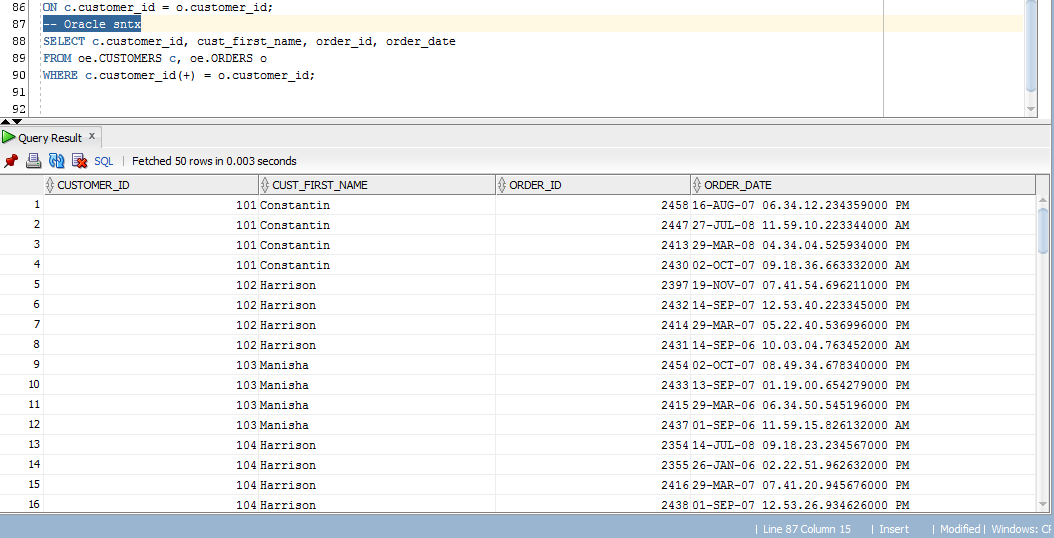


**RIGHT OUTER JOIN**

Right join is the same thing as Left join but the difference is that table in FROM clause joines to the table in RIGHT JOIN clause.

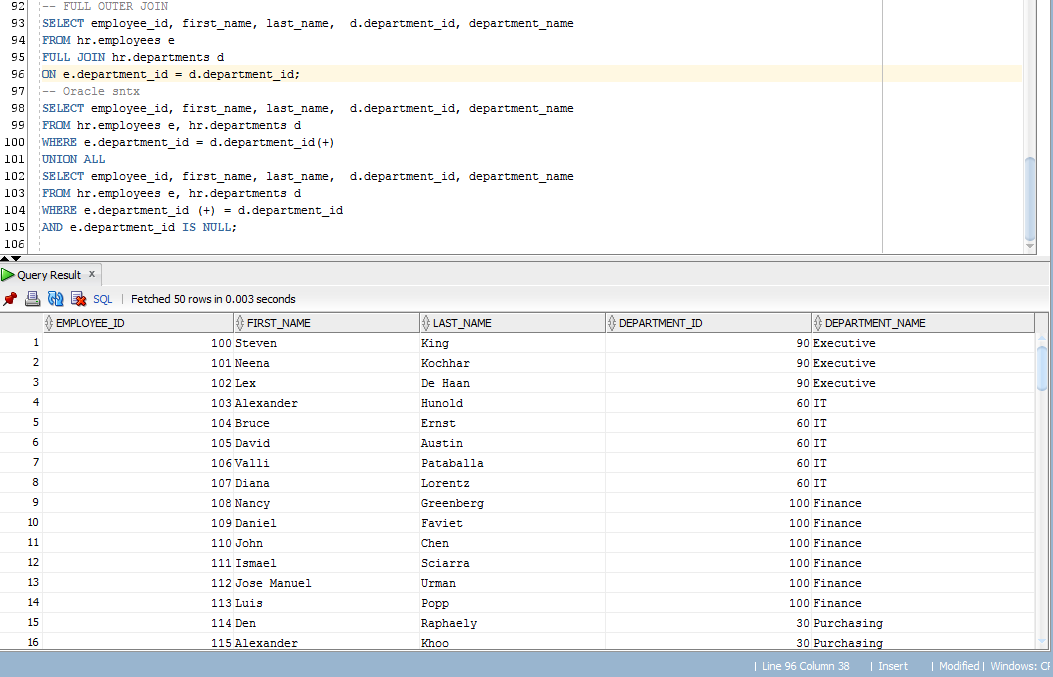


+ Oracle sntx

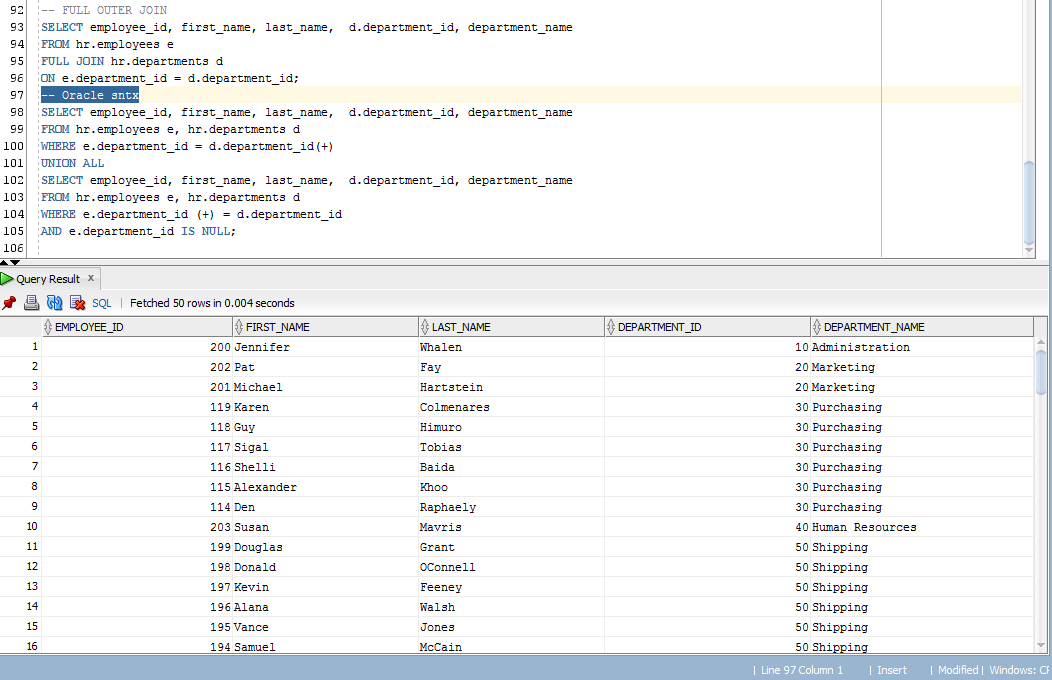


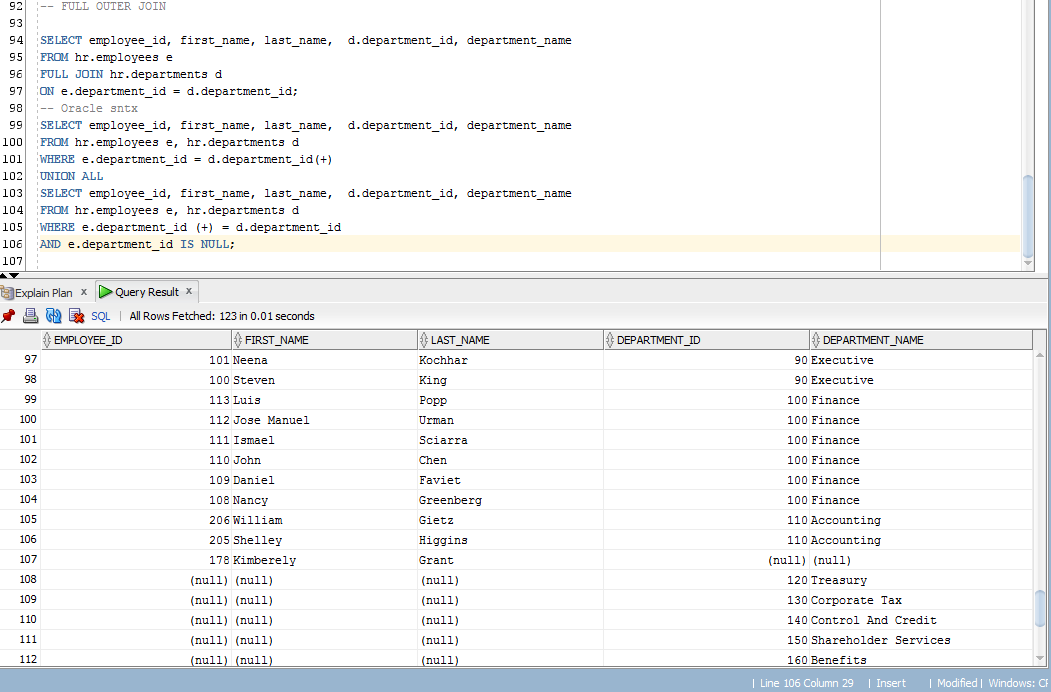
**FULL OUTER JOIN**

Full outer join makes it possible to see all values from both tables with null if they have no matches.

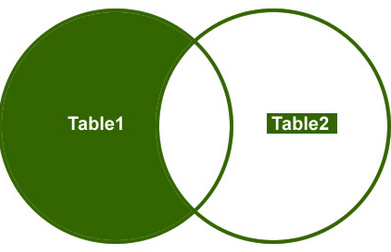


+ Oracle syntax





In both variants we get count number of all records = 123. One more important thing is that we should use e.department id = null so as not to get already used values in union.



**Task 8**

|  |  |  |
| --- | --- | --- |
| Table “A” | Table “B” | Join type description |
| Small Table w/o index on join field | Small Table w/ index on join field | Nested Loop  - small tables & index |
| Small Table w/o index on join field | IOT | Merge join  - similar tables by amount of records |
| Small Table with index on join table | Large Table | Nested Loop  ! Condition with PK = appropriate value  Small Table is limited by 1 value, so Nested Loop could be used effective to go in one cycle through all large table |
| Medium table without index on join field | Medium table without index on join field | Hash Join  - using hash without sort as it is too costly |
| Any table  without join field | Any table  without join field | Merge Join Cartesian  - no join field |
| Large table with index on join field | Large table with index on join field | Hash Join  - using hash on one table so as not to use sort & large set of data |
| Large table without index on join field | Small table without index on join field | Hash Join  - move small table to hash and turn to it from the large table & no index & large set of data |
| Large table without index on join field | Small table with index on join field | Hash Join  - move small table to hash and turn to it from the large table & no index & large set of data |
| Large table with index on join field | Small table with index on join field | Hash Join  - move small table to hash and turn to it from the large table & no index & large set of data |
| Large table with index on join field | Small table with index on join field | Merge Join  !Small table ordered by PK  - one table is sorted by its PK so there is no need to turn to manual sort (that is used in Merge Join).This table can be easily proceed by large table in merge join method |

\* few examples was made with the help with organized small/large tables, which are presented in the script in the folder