

## DB2 Virtual Indexes : how Click4DB can automate the process ?

To solve performance issues, one of the solution is to create a new index. But the design of the best index can be time consuming due to number of candidate fields or number of different queries using the related table. Moreover, the creation of an index is CPU consuming and it can be disruptive; so it is best to be sure of its usage in advance.

Virtual index is a nice way to “give a new index a try” before you create it.

### Find candidate columns

First you will have to make an inventory of candidate columns

### Create Virtual Indexes

As a 5 fields INDEX can lead to 325 fields combinations, imagine how you will find the best index among the 325 candidates?

Don't worry ! **Click4DB** will do the job.

**Click4DB** can create Virtual Indexes from 1 up to 5 different columns. You can decide if the firstkeycard and fullkeycard columns must be calculated or estimated based on catalog statistics. If your statistics values are correct in your non production DB2, estimated keycards will give you a realistic result.

### Comparison between Calculated & Estimated cardinalities

	Calculated KEYCARDS	Estimated KEYCARDS
Virtual index creation time	Linked to table size <sup>(1)</sup>	< 10 seconds
CPU consumption	Linked to table size	Minimum
Data accuracy	Maximum	Relative to formulas used <sup>(2)</sup>
Number of time the base table is read	1 <sup>(3)</sup>	0

(1) For a 100.000 rows tables it takes about 30 seconds.

(2) The values estimated can be smart enough to give a correct view to the optimizer.

(3) Only from V11 , usage of the OLAP function CUBE.

### Explain and check indexes usage

Now, you can explain all the dynamic queries and packages using the table; The PLAN\_TABLE will indicates the best candidate indexes.

If more than one index is used, you can apply your own criteria's :

- Index only
- Highest matchcols
- Avoiding sort

## Create index and REBIND

After this quick and simple process, the index creation can be done and the packages rebound to take advantages of the new available index.

### Time consumption evaluation

Task	Who ?	Time required
FIND fields candidate fields	You	Related to information collected <sup>(1)</sup>
CREATE virtual indexes	<a href="#">Click4DB</a>	10 sec <sup>(2)</sup>
EXPLAIN to show index usage	DB2 Optimizer	5 min.
Select summary information from plan_table	<a href="#">Click4DB</a>	10 sec
CREATE the index to resolve performance problem	You	Related to table size <sup>(3)</sup>
REBIND packages	DB2 Optimizer	Related to the number of packages <sup>(3)</sup>
CHECK access path for the table	<a href="#">Click4DB</a>	1 min.

- (1) [Click4DB](#) can make selection based on DSN\_PREDICAT\_TABLE , DSN\_SORTKEY\_TABLE if they are available on your system.
- (2) Based on a simulation of firstkeycard and fullkeycard (no data access)
- (3) The time required for this this action is always necessary and not improvable

## Summary

DB2 Virtual Indexes is a very nice feature to solve a performance problem. However, creating all potential Virtual Indexes can be time consuming. [Click4DB](#) automates this step and makes it fast and easy to realize.



Of course this solution must be always challenged with query rewrite or other tuning actions.