developerWorks®

Encoded-vector index (EVI) only access in IBM DB2 for i

Using encoded vector indexes to accelerate the projection of columns returned by an SQL query with DB2 for i

Scott L. Forstie

January 29, 2016

IBM® DB2® for i continues to enhance the SQL Query Engine (SQE) in many ways. One recent enhancement delivers a new way that SQE can use an encoded-vector index (EVI) to access the data needed by a query through an index implementation. This new capability is called *EVI only access* (EOA) where the data can be reached by *only accessing* an index. This article provides some of the foundational details you'll need to use *EOA*.

EVI only access

IBM DB2 for i continues to enhance the SQE in many ways. One recent, in November 2015, enhancement delivers a new way that SQE can use an EVI to access the data needed by a query through an index implementation. This new capability is called EVI only access (EOA) where the data can be reached by *only accessing* an index. This article provides some of the foundational details you'll need to use *encoded-vector index only access*.

EVIs are a patented technology invented by the IBM Research team and they're unique to IBM DB2 for i. What can an EVI do for you? Indexes are a permanent structure, maintained by the database to provide a fast answer to your question (get the data needed by a query). EVIs can be used as part of **selection**, where the resultant rows are identified. EVIs used for selection are generally considered more beneficial when the column has low cardinality (number of unique values). In 2012, IBM i 7.1 was enhanced to allow EVIs to **include and maintain aggregate** detail such as summary, average, and count values over important columns.

With this enhancement to support EOA, EVIs can be used for **projection** of column values. Projection of column values are the columns found between the SELECT and the FROM clauses in your SQL query. Where should you use EOA? As you might expect, I'll have to respond with "It depends". Anyone answering a performance-related question always takes this safe path. The reality of performance is that it truly does depend upon factors, some of which are client-specific. Like others, I'll lean on the "It depends", but also encourage you to get educated and then spend time evaluating your options.

Enabling EOA

A QAQQINI query option controls whether or not SQE can consider costing and using EVIs for EOA. The control is named **ALLOW_EVI_ONLY_ACCESS**, and it can be configured to be set to the following values:

1. ***DEFAULT** – In IBM i 7.1, *NO is the default value. In IBM i 7.2, *YES is the default value.
2. ***YES** – EOA is eligible to be considered by SQE
3. ***NO** – EOA is not eligible to be considered by SQE

As you can see, the default behavior differs between IBM i 7.1 and 7.2. Why is this? Because many IBM i 7.1 users prefer operations and query implementations to remain unchanged, we decided to have the support set to off by default on that release. To override this decision, simply update your favorite QAQQINI file to use **ALLOW_EVI_ONLY_ACCESS** with *YES or update QUSRSYS/QAQQINI to enable EOA across the entire partition. If you're using IBM i 7.2, you don't need to do anything to enable EOA, other than applying the DB2 PTF Group.

The DB2 PTF Group which included EOA enablement is shown in Table 1. To use EOA, your DB2 PTF Group level needs to match or exceed the levels in the table.

Table 1: DB2 PTF Group levels which enabled EOA

DB2 for i enhancement	IBM i 7.2	IBM i 7.1
EVI Only Access (EOA)	**SF99702 Level 9**	**SF99701 Level 38**

EVI RRN probe

When processing the projects of column values, a table probe operation is used to retrieve a specific row from a table based upon its row number. The row number is provided to the table probe access method by some other operation that generates a row number for the table. With the new EOA capability, the column value can now be retrieved from the EVI using an *EVI relative record number (RRN) probe*. The EVI RRN probe is an index only access method that is used to provide selected columns by retrieving the value from the EVI instead of using a table probe to access the table. Retrieving the value from the EVI should provide better I/O characteristics than the random I/O associated with a table probe operation.

This access method is used in conjunction with a radix index probe, radix index scan, or EVI probe operation. The radix index probe, radix index scan, or EVI probe operation is used to select the rows and then the RRN of the selected row is used to probe into EVIs to retrieve any selected values that were not provided by the index used for selection. The EVI RRN probe can access multiple EVIs to provide selected values.

Table 2 provides a summary of the new EVI RRN probe with its associated advantages, considerations, examples, and other information.

Table 2: EVI RRN probe attributes

Data access method	EVI RRN probe

Description	The encoded-vector index is quickly probed based on the RRNs provided by the underlying index access.
Advantages	• Provides the potential to extract all the data from the EVI index key values, thus eliminating the need for a table probe • Provides better paging characteristics than a table probe
Considerations	• Only single-key EVIs are considered for this implementation • All selected columns must have a single column EVI created • The EVIs must fit in the query's fair share of optimizer memory
Likely to be used	• When the table row size is wide, the number of select columns is small compared to the number of columns in the table and the query requires a table probe to retrieve columns

Example SQL statement	`SELECT QUANTITY, ORDERPRIORITY, EXTENDEDPRICE`
	` FROM ITEM_FACT`
	`  WHERE SHIPMODE = 'TRUCK'`
	`  ORDER BY ORDERDATE DESC`
	`LIMIT 50`
	`OFFSET 0;`

Example CREATE INDEX SQL statements	`CREATE ENCODED VECTOR ITEM_FACT_QUANTITY_EVI`
	`   ON ITEM_FACT ( QUANTITY ASC )`
	`   WITH 65537 DISTINCT VALUES;`
	
	`CREATE ENCODED VECTOR ITEM_FACT_ORDERPRIORITY_EVI`
	`   ON ITEM_FACT ( ORDERPRIORITY ASC )`
	`   WITH 65537 DISTINCT VALUES;`
	
	`CREATE ENCODED VECTOR ITEM_FACT_EXTENDEDPRICE_EVI`
	`   ON ITEM_FACT ( EXTENDEDPRICE ASC )`
	`   WITH 65537 DISTINCT VALUES;`
	
	`CREATE INDEX ITEM_FACT_SHIPMODE_RADIX`
	`   ON ITEM_FACT ( SHIPMODE ASC );`

Database monitor and plan cache record indicating use	A QQRID 3001 index used record for each EVI with QQRCOD = 'I8'
SMP parallel enabled	Yes
Also referred to as	Table probe, preload
Visual Explain icon	

Prior to encoded vector index only access, the recommendation had been to only create EVIs for column with low cardinality (small number of distinct values). This recommendation has now changed. EVI RRN probe can be used for columns with high cardinality (large number of distinct values). However, when creating the EVI, the WITH integer DISTINCT VALUES clause should be used to set the initial width of the EVI vector appropriately and to minimize maintenance time if the database manager needs to use a wider vector. Refer to the CREATE INDEX statement in SQL reference for more details.

Table 3 shows the three database Navigator product options. The information and examples within this article were built using the Navigator product found within IBM i Access for Windows. Some refer to this product as System i Navigator, while others call it IBM i Navigator. In this article, the term IBM i Navigator is used.

Table 3: Database Navigator product overview

Products	IBM i Access for Windows	IBM i Access Client Solutions	IBM i Navigator

Aliases	System i Navigator or IBM i Navigator	ACS	Navigator for i
Where does it run?	Windows PC Install	Anywhere Java can be used	Browser Served from IBM i 7.1 and 7.2
Recent service level?	IBM i Access Windows Service Pack 7.1 – SI57907	Version 1.1.5.0	IBM HTTP Server for i PTF Group: 7.2 - SF99713 Level 12 7.1 - SF99368 Level 37
Best of breed features, for the database user	Run SQL scripts Visual Explain Many other features	Run SQL scripts SQL Performance Center	PDI Perspectives OmniFind administration
Webpage to watch	**ibm.com**/systems/power/software/i/access/windows_sp.html	**ibm.com**/systems/power/software/i/access/solutions_support.html	www-912.ibm.com/s_dir/SLINE003.NSF/PTFbyNumber/SF99713 www-912.ibm.com/s_dir/SLINE003.NSF/PTFbyNumber/SF99368
Next (planned) update	June, 2016	July, 2016 → Version 1.1.6.0	New PTF Group every 2 to3 months

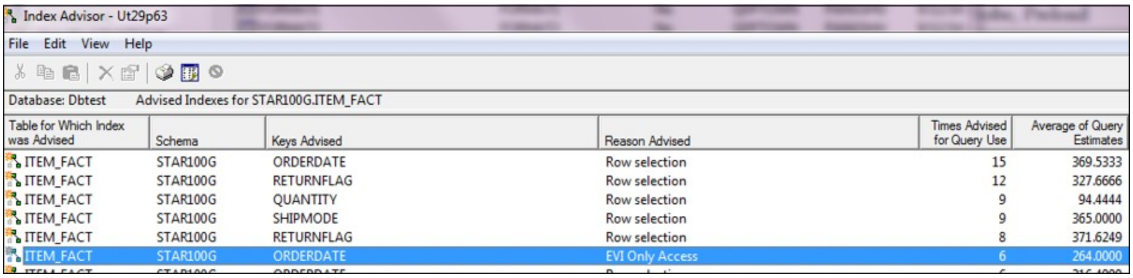
Index advice for EVI RRN probe plans

Index Advisor may advise EVI indexes for use with the EVI RRN probe plan. To limit excessive index advice, SQE won't consider advising EOA unless there's at least one single column EVI over one or more of the projection columns for the query.

The EVI RRN probe plan advice requires one or more indexes to be created as a dependent set. If any of the EVI indexes are missing, the optimizer won't be able to cost and choose these dependent indexes for implementation of the EVI RRN probe.

The Reason Advised column will have reason code 'I8' if you query QSYS2/SYSIXADV directly. If you use IBM i Navigator to examine the index advice, you'll see something similar to the one shown in Figure 1.

Figure 1. Index advice and EOA



Database: Dbtest   Advised Indexes for STAR100G.ITEM_FACT

Table for Which Index was Advised	Schema	Keys Advised	Reason Advised	Times Advised for Query Use	Average of Query Estimates
ITEM_FACT	STAR100G	ORDERDATE	Row selection	15	369.5333
ITEM_FACT	STAR100G	RETURNFLAG	Row selection	12	327.6666
ITEM_FACT	STAR100G	QUANTITY	Row selection	9	94.4444
ITEM_FACT	STAR100G	SHIPMODE	Row selection	9	365.0000
ITEM_FACT	STAR100G	RETURNFLAG	Row selection	8	371.6249
ITEM_FACT	STAR100G	ORDERDATE	EVI Only Access	6	264.0000

The advice is on a per table basis for the query.

Indexes are advised only when the following statements are true:

- At least one of the columns in the select list of the query needs to have an existing single column EVI.
- The number of columns that do not match to an existing EVI must be less than 20.

- All columns must be eligible to be an index key.

When Index Advisor shows a highly dependent advice, use of the exact match capability from Show Statements to find the query in the plan cache is helpful. Once found, use Visual Explain to discover the dependent index advice specific to that query.

EVI RRN probe plan advice example

In the following example, assume that the following indexes already exist.

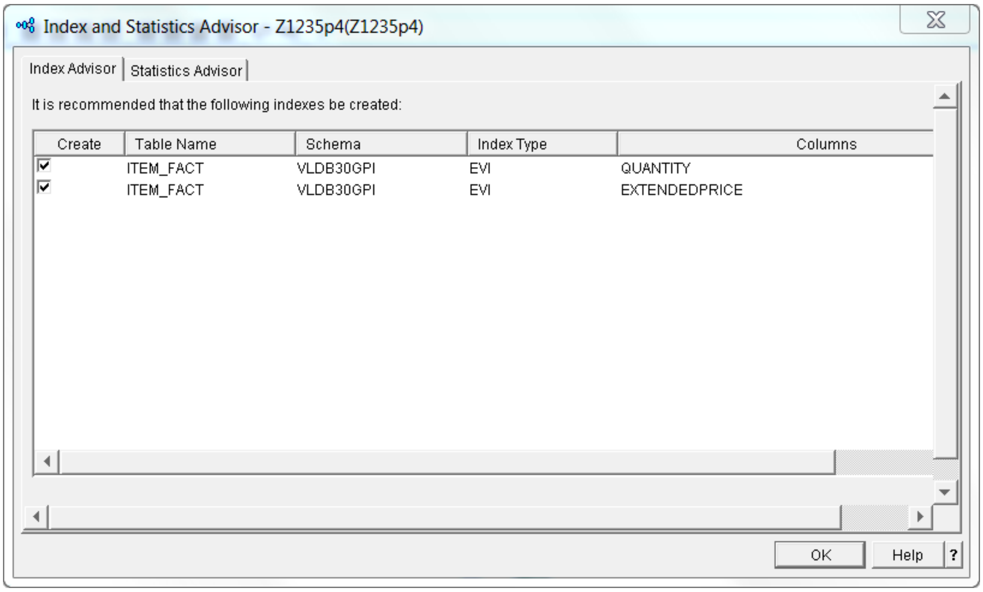
- A radix index over the column ORDERDATE
- An EVI index over the column ORDERPRIORITY

```
SELECT QUANTITY, ORDERPRIORITY, EXTENDEDPRICE FROM ITEM_FACT
  WHERE SHIPMODE = 'TRUCK'
  ORDER BY ORDERDATE DESC
LIMIT 50
OFFSET 0;
```

The Index Advisor will advise an EOA style EVI index over all over missing columns: QUANTITY and EXTENDEDPRICE. While EOA style index advice appears in the IBM i Navigator Index Advisor, if you have a specific query, you should use the Index Advisor of Visual Explain to quickly identify the missing indexes.

Figure 2 shows the EVI advice generated for the query, when examining the advice using Visual Explain.

Figure 2. EOA and index advice within Visual Explain



It is recommended that the following indexes be created:

Create	Table Name	Schema	Index Type	Columns
☑	ITEM_FACT	VLDB30GPI	EVI	QUANTITY
☑	ITEM_FACT	VLDB30GPI	EVI	EXTENDEDPRICE

Whenever you are wondering why an existing index was not used by SQE, you can look in two places to gain some additional insight.

1. **Analyze:** If you capture an SQL performance monitor, you can use IBM i Navigator or IBM i Access Client Solutions to analyze the monitor. If you drill into the statements detail for SQE,

you can access the *Indexes Considered* action to see the indexes considered by SQE and a basic explanation of why the index was not used. Figure 3 shows only that reason code 20. A verbose reason might appear in a future IBM i release.

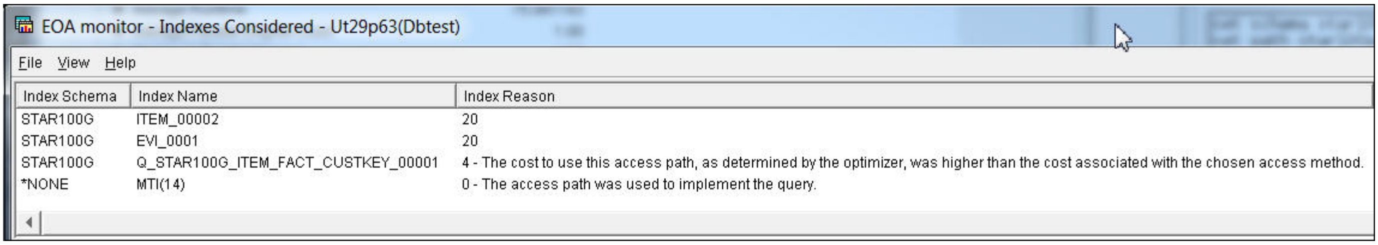
2. **Visual Explain:** The same information about indexes considered can be found under the **Information About the Plan Performed** section and the **List of Indexes Optimized** detail. Figure 4 shows that the EVIs are not used by the optimizer for reason 20.

With EOA support, the indexes considered detail has been extended to include two addition reason codes:

- **Reason code 20**: The access path cannot be used because Encoded Vector Index does not fit in memory.
  User response: Consider using a private pool, as discussed in the next section.
- **Reason code 21**: The access path cannot be used because not all referenced columns had an Encoded Vector Index.
  User response: Create single column EVIs over all referenced columns.

Note: Refer to the QQ1000 column of the QQQ3007 table within IBM Knowledge Center for explanations of reasons 1 to 19.

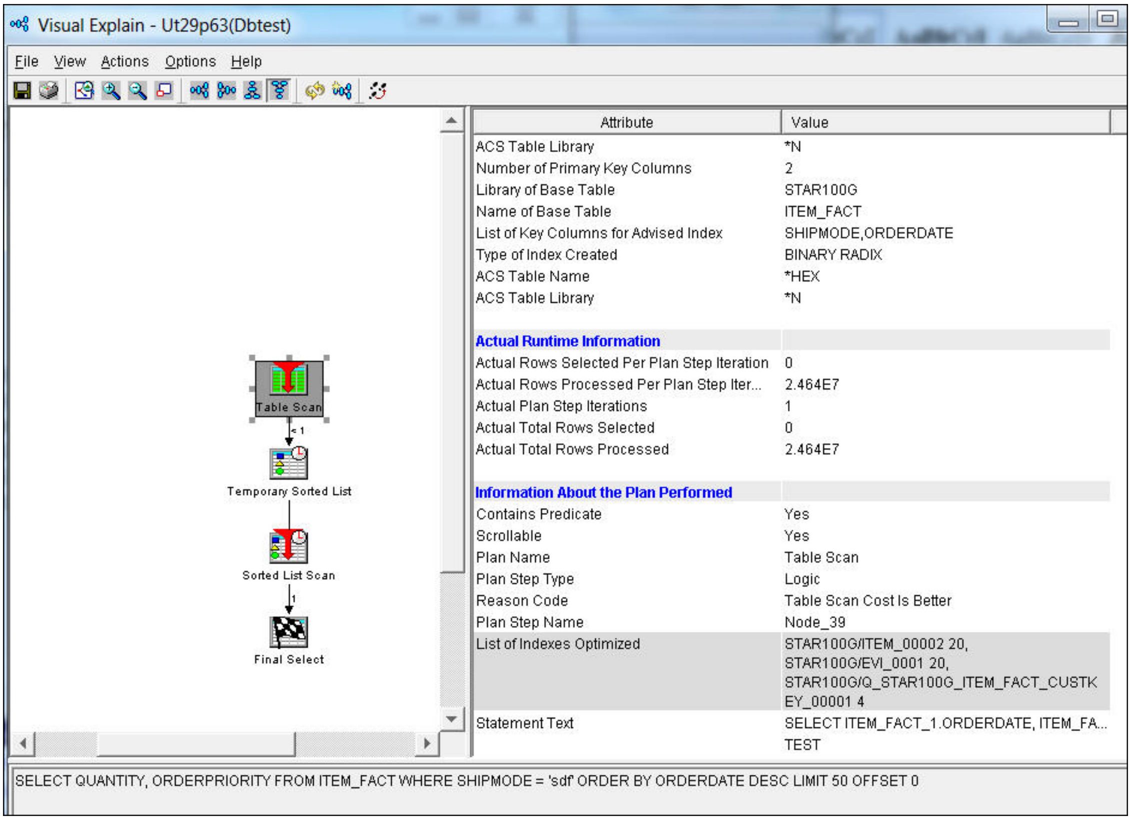
Figure 3. IBM i Navigator analyze indexes considered



EOA monitor - Indexes Considered - Ut29p63(Dbtest)

Index Schema	Index Name	Index Reason
STAR100G	ITEM_00002	20
STAR100G	EVI_0001	20
STAR100G	Q_STAR100G_ITEM_FACT_CUSTKEY_00001	4 - The cost to use this access path, as determined by the optimizer, was higher than the cost associated with the chosen access method.
*NONE	MTI(14)	0 - The access path was used to implement the query.

Figure 4. Visual Explain indexes considered and not used



Attribute	Value
ACS Table Library	*N
Number of Primary Key Columns	2
Library of Base Table	STAR100G
Name of Base Table	ITEM_FACT
List of Key Columns for Advised Index	SHIPMODE,ORDERDATE
Type of Index Created	BINARY RADIX
ACS Table Name	*HEX
ACS Table Library	*N
Actual Runtime Information	
Actual Rows Selected Per Plan Step Iteration	0
Actual Rows Processed Per Plan Step Iter...	2.464E7
Actual Plan Step Iterations	1
Actual Total Rows Selected	0
Actual Total Rows Processed	2.464E7
Information About the Plan Performed	
Contains Predicate	Yes
Scrollable	Yes
Plan Name	Table Scan
Plan Step Type	Logic
Reason Code	Table Scan Cost Is Better
Plan Step Name	Node_39
List of Indexes Optimized	STAR100G/ITEM_00002 20, STAR100G/EVI_0001 20, STAR100G/Q_STAR100G_ITEM_FACT_CUSTKEY_00001 4
Statement Text	SELECT ITEM_FACT_1.ORDERDATE, ITEM_FA... TEST

```
SELECT QUANTITY, ORDERPRIORITY FROM ITEM_FACT WHERE SHIPMODE = 'sdf' ORDER BY ORDERDATE DESC LIMIT 50 OFFSET 0
```

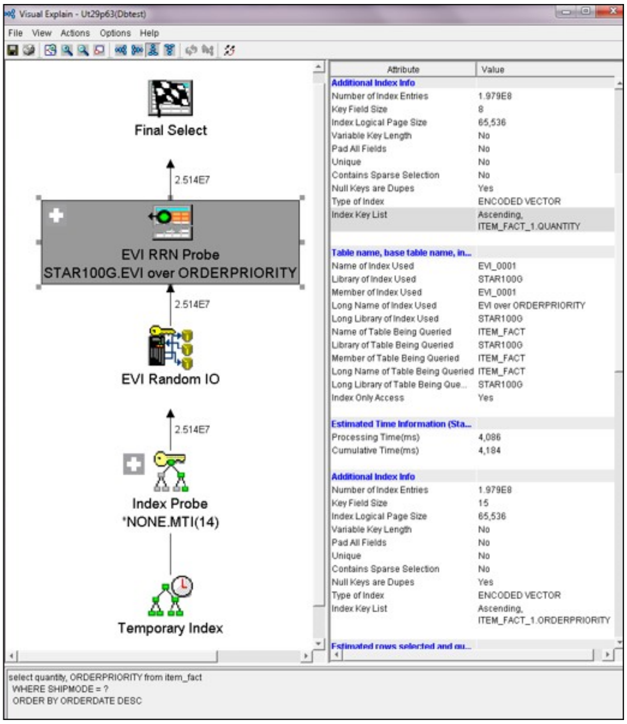
Fair share of memory

Because indexes can be very large, SQE has taken a conservative approach when considering whether to use EOA. If the memory impact for an EOA implementation would exceed the user's fair share of memory, SQE will avoid using EOA.

To mitigate this issue, you can configure a private storage pool and use the QAQQINI MEMORY_PREFERENCE option to direct SQE to use that pool for your query. This is a useful tuning technique for queries beyond EOA, but it is a critical success factor for those who need to use EOA with large data. See the end of this article for a pointer to memory preference detail.

Figure 5 shown a screen capture from Visual Explain containing the query implementation after we have created the advised EVI indexes and mitigated the *fair share memory* issue.

Figure 5. Visual Explain of an EOA implementation



Attribute	Value
Additional Index Info	
Number of Index Entries	1.979E8
Key Field Size	8
Index Logical Page Size	65,536
Variable Key Length	No
Pad All Fields	No
Unique	No
Contains Sparse Selection	No
Null Keys are Dupes	Yes
Type of Index	ENCODED VECTOR
Index Key List	Ascending, ITEM_FACT_1.QUANTITY
Table name, base table name, in...	
Name of Index Used	EVI_0001
Library of Index Used	STAR100G
Member of Index Used	EVI_0001
Long Name of Index Used	EVI over ORDERPRIORITY
Long Library of Index Used	STAR100G
Name of Table Being Queried	ITEM_FACT
Library of Table Being Queried	STAR100G
Member of Table Being Queried	ITEM_FACT
Long Name of Table Being Queried	ITEM_FACT
Long Library of Table Being Que...	STAR100G
Index Only Access	Yes
Estimated Time Information (Sta...	
Processing Time(ms)	4,086
Cumulative Time(ms)	4,184
Additional Index Info	
Number of Index Entries	1.979E8
Key Field Size	15
Index Logical Page Size	65,536
Variable Key Length	No
Pad All Fields	No
Unique	No
Contains Sparse Selection	No
Null Keys are Dupes	Yes
Type of Index	ENCODED VECTOR
Index Key List	Ascending, ITEM_FACT_1.ORDERPRIORITY
Estimated rows selected and qu...	

```
select quantity, ORDERPRIORITY from item_fact
 WHERE SHIPMODE = ?
 ORDER BY ORDERDATE DESC
```

German football and EOA

These two topics have more in common than you might imagine. To demonstrate how EOA might be used, we need to look no further than the German Bundesliga football league. In this fictional example, we have a very large database table consisting of one row for each fan. The row gives us information such as the fan's name, email address, and of course, the team for which the fan root, cheer and encourage through thick and thin, ups and downs, wins and losses.

The query for this example is quite simple. Find the email addresses of all the fans of the Eintracht club, based in Frankfurt. Figure 5 contains a visual explanation of EOA in action.

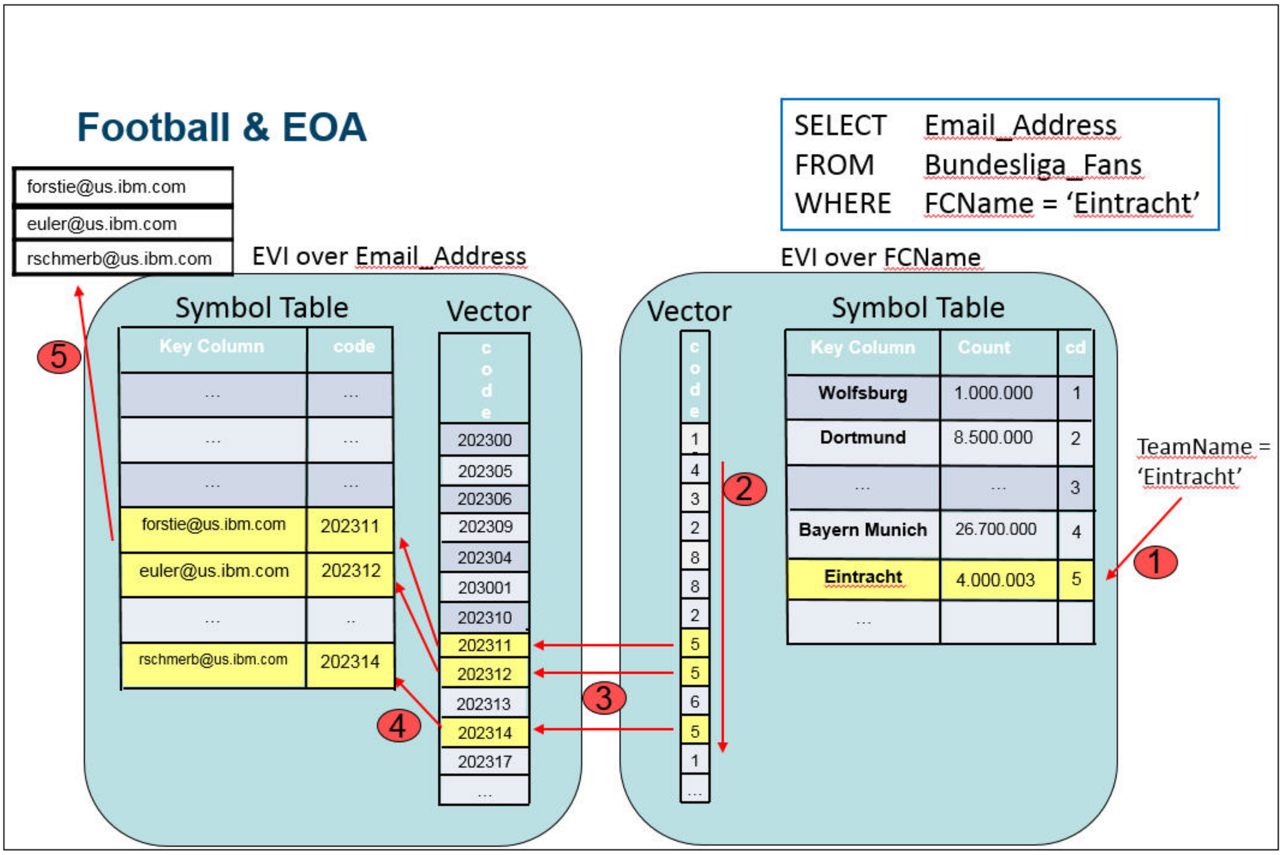
We begin our examination at red circle numbered 1 in Figure 5. Using a traditional EVI built over the column FCName (that is the football club name), we are able to efficiently find the rows in the table that matches our selection criteria. The traditional EVI has maintained a symbol table containing the various football club names and the associated symbol table code for that value. Because there are only 18 football clubs in Bundesliga, we can conclude that the cardinality of FCName is very low.

At the red circle numbered 2, we see SQE using the EVI symbol table code (5), scanning the EVI vector and identifying the rows that match 'Eintracht'. Well done SQE, cheers from the crowd!

At the red circle numbered 3, we see EOA appear. The vector scan in the previous step has not only identified the selection rows, but because the vector is constructed in a relative row order, we can use that insight to quickly identify the same rows in vector of any associated EVIs. In the example, such an EVI is built over the Email_Address column. EOA can access column values starting with the EVI vector and moving up to the EVI symbol table. As shown in the example, this is the opposite order of processing, as compared to using EVIs for selection of rows.

At red circle numbered 4, we see that the EVI vector code is used to find the column value in the EVI symbol table. This EOA style EVI could have a very high level of cardinality and still be useful. It is important to understand this point, because EVIs were traditionally targeted to lower cardinality selection of data. The other consideration is that if you know that your EOA style EVI index will have higher cardinality, you should use the `WITH xx DISTINCT VALUES` so you can start with a wide EVI vector. While DB2 for i can expand the vector size as cardinality pushes it wider, it is better to start with the widest vector if you anticipate many unique values to appear in the column.

Figure 6. Bundesliga football and EOA



forstie@us.ibm.com
euler@us.ibm.com
rschmerb@us.ibm.com

```
SELECT   Email_Address
FROM     Bundesliga_Fans
WHERE    FCName = 'Eintracht'
```

Key Column	code
…	…
…	…
…	…
forstie@us.ibm.com	202311
euler@us.ibm.com	202312
…	..
rschmerb@us.ibm.com	202314

code
202300
202305
202306
202309
202304
203001
202310
202311
202312
202313
202314
202317
…

code
1
4
3
2
8
8
2
5
5
6
5
1
…

Key Column	Count	cd
Wolfsburg	1.000.000	1
Dortmund	8.500.000	2
…	…	3
Bayern Munich	26.700.000	4
Eintracht	4.000.003	5
…		

Finally, at red circle numbered 5, our query returns the results by accessing only two encoded-vector indexes. DB2 for i and SQE has used EOA to "**ein Tor schießen"**, which translates to "score a goal"**!**

Summary

The goal of this article is to raise awareness to EOA and provide of the basic knowledge to help you get started. Indexes aren't free, so you probably should not just go ahead and build single column EVIs over the columns of your favorite table. Instead, look for an under-performing query and take EOA for a test drive.

Resources

If you're responsible for the indexing strategy, you should consider using a couple of resources to help you along the way.

- QAQQINI memory preference by pool name
- Database performance and query optimization
- BOOM! Another IBM i 7.1 Technology Refresh: (EVI aggregates)
- IBM DB2 for i – indexing methods and strategies white paper
- DB2 for i SQL Performance Workshop