

EOA - EVI Only Access

Die Möglichkeiten der SQL Query Engine von IBM DB2 for i wurden wesentlich weiter entwickelt. Eine kürzlich erfolgte Erweiterung liefert einen neuen Weg, womit die SQE unter Verwendung von EVI (Encoded Vector Index) und einer Index-Implementierung in der Tabelle QAQQINI eine Abfrage ausführen kann. Diese neue Fähigkeit wird als EVI Only Access (EOA) bezeichnet. Der Zugriff auf die gewünschten Daten erfolgt dann nur noch über den Index.

Was ist und was kann EOA?

Die Verwendung von EVIs (Encoded Vector Index) zielt zunächst auf die Auswahl von Schlüssel (WHERE-Anweisung) und Aggregation ab. Mit der Erweiterung von EOA können EVIs nun auch für die Projektion von Spaltenwerten (SELECT list) verwendet werden. Dieser neue Ansatz mit EOA wird bei der Verwendung des Index Advisors von System i Navigator sichtbar. Der Wert REASON_TYPE wird dabei auf 'I8' gesetzt.

Das geänderte Zeitverhalten bei Verwendung von EOA zeigt das folgende Beispiel, das ich aus einem IBM-Dokument entnommen habe. Dabei wird die Auswirkung auf das Zeitverhalten unter Verwendung eines EVIs für *fld1* und ein anderer EVI für *fld2* in der folgenden Abfrage untersucht:

```
SELECT COUNT(DISTINCT(fld1)) WHERE fld2 < 'value'
```

Aus den verschiedenen Tests entstand das folgende Diagramm:

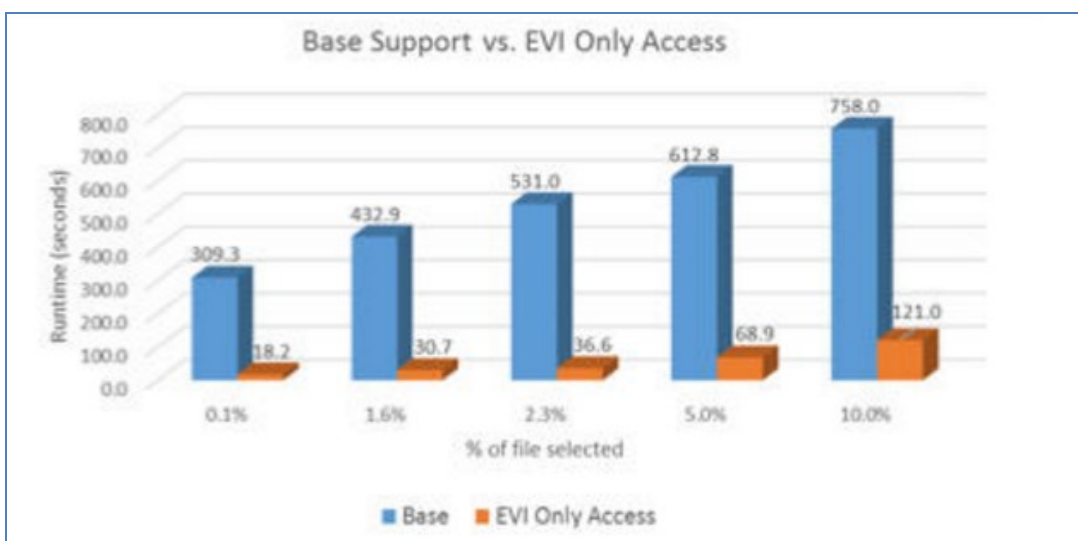


Abbildung 1: Vergleich zwischen EVI-Datenzugriff und Nicht-EVI-Datenzugriff

Die Implementierung von EOA

Die Aktivierung von EOA erfolgt über einen entsprechenden Eintrag in der Tabelle QAQQINI. Damit kann festgelegt werden, ob die SQE die Verwendung von EVIs für EOA in Betracht ziehen soll oder nicht. Der Name des Kontrolleintrags ist **ALLOW_EVI_ONLY_ACCESS**. Die Option kann folgende Werte erhalten:

1. ***DEFAULT** – In IBM i 7.1 ist der Default-Wert *NO. In IBM i 7.2 ist der Defaultwert *YES.
2. ***YES** – EOA kann durch die SQE berücksichtigt werden.
3. ***NO** – EOA kann durch die SQE nicht berücksichtigt werden.

Wie Sie sehen, ist das standardmäßige Verhalten unterschiedlich zwischen IBM i 7.1 und IBM i 7.2. Warum ist das so? Viele Anwender mit IBM i 7.1 ziehen es vor, dass die Operationen und Implementierungen der Abfragen unverändert bleiben. Daher wurde der Default-Wert bei diesem Release auf *NO gesetzt. Dieser kann durch Änderung der entsprechenden QAQQINI mit **ALLOW_EVI_ONLY_ACCESS** und dem Wert ***YES** überschrieben werden. Bei IBM i 7.2 ist keine Änderung erforderlich. Voraussetzung ist die Installation der korrekten DB2 PTF-Gruppe.

Die DB2 PTF-Gruppe, mit der die Unterstützung für EOA ermöglicht wurde, zeigt die folgende Tabelle:

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

Abbildung 2: DB2 PTF Gruppen-Stufen (Levels), mit denen SOA aktiviert wurden

Damit SOA verwendet werden kann, muss der PTF-Stand der DB2 PTF-Gruppe mindestens 9 (bei IBM i 7.2) oder 38 (bei IBM i 7.1) sein.

Die Option für SOA kann mit der folgenden SQL-Anweisung eingefügt werden, falls sie nicht vorhanden ist:

```
INSERT INTO STAR1GA.QAQQINI
VALUES ('ALLOW_EVI_ONLY_ACCESS',
       '*DEFAULT',
       'Query option controls whether or not SQE can consider costing and
using EVIs for EOA. *DEFAULT - In IBM i 7.1, *NO is the default value. In
IBM i 7.2, *YES is the default value.');
```

Abbildung 3: SQL-Anweisung für ALLOW_EVI_ONLY_ACCESS

Tabelle	Spalten	Integritätsbedingungen über Schlüssel	Integritätsbedingungen über Fremdschlüssel	Prüfintegritätsbedingungen	Gespeicherte Abfrage	Part	
Spaltenname	Systemname	Datentyp	Länge	Kann Nullwerte entha...	Standardwert	Text	Überschrift 1
QQPARM	QQPARM	VARCHAR	256	Nein	"	Abfr.optionsparameter	Parameter
QQVAL	QQVAL	VARCHAR	256	Nein	"	Wert Abfrageoptionspara...	Parameterwert
QQTEXT	QQTEXT	VARGRAPHIC	1000	Ja	Standarddatentyp	Abfr.optionstext	Abfr.option

Abbildung 4: Struktur der Tabelle QAQQINI

EVI RRN Probe (probe=Untersuchung)

Bei Ausführung von Projekten zur Ermittlung von Spaltenwerten wird eine Operation Table Probe verwendet, um die Zeile einer Tabelle über deren Zeilennummer abzurufen. Diese Zeilennummer wird der Table Probe Zugriffsmethode durch eine andere Operation verfügbar gemacht, die eine Zeilennummer für die Tabelle erstellt. Mit der neuen EOA-Methode kann der Wert einer Spalte nun über den EVI mit Hilfe der EVI Relative Record Number (RRN)-Probe ermittelt werden. Die EVI RRN-Probe ist eine Index Only Access-Methode, die zur Ermittlung von ausgewählten Spalten dient, indem der Wert über einen EVI anstatt mit Hilfe einer Table Probe aus der Tabelle ermittelt wird. Das Abrufen eines Wertes über einen EVI sollte bessere I/O-Werte liefern, als der wahlfreie I/O-Zugriff, der mit einer Table Probe-Operation verbunden ist.

Diese Methode wird in Verbindung mit den folgenden Techniken angewandt: Radix Index-Probe, Radix Index-Scan oder EVI-Probe. Mit diesen Techniken werden zunächst die Zeilen selektiert. Dann wird die relative Satz-Nummer (RRN) der selektierten Zeile dazu benutzt, um mit Hilfe der EVIs selektierte Werte aufzufinden, die mit der Indexauswahl nicht zur Verfügung gestellt wurden. Die EVI RRN-Probe kann mehrere EVIs verwenden, um ausgewählte Werte zu finden.

In der folgenden Tabelle finden Sie eine Übersicht über die Eigenschaften der neuen EVI RRN-Probe:

Beschreibung	Der Encoded Vektor Index (EVI) kann rasch mit Hilfe der RRNs untersucht werden, die durch den zugrunde liegenden Index-Zugriff verfügbar gemacht werden.
Vorteile	<ul style="list-style-type: none"> ermöglicht den Zugriff auf alle Daten der EVI Index Schlüsselwerte und vermeidet dadurch die Table Probe. bietet bessere Paging-Eigenschaften als eine Table-Probe.
Überlegungen	<ul style="list-style-type: none"> Nur einfache Schlüssel EVIs werden bei dieser Implementierung berücksichtigt. Für alle selektierten Spalten muß ein einfacher Spalten-EVI existieren. Der Speicherbereich des zugeordneten Pools muss ausreichend groß sein, um die EVIs aufnehmen zu können.
Voraussichtliche Anwendung	Wenn eine Tabelle viele Zeilen hat, die Anzahl der ausgewählten Spalten klein ist im Vergleich zur gesamten Spaltenzahl der Tabelle und die Abfrage eine Table Probe zum Auslesen der Spalten benötigt.
Beispiel SQL-Abfrage	<pre>SELECT QUANTITY, ORDERPRIORITY, EXTENDEDPRICE FROM ITEM_FACT WHERE SHIPMODE = 'TRUCK' ORDER BY ORDERDATE DESC LIMIT 50 OFFSET 0;</pre>
Beispiele für Index-Erstellung	<pre>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);</pre>
Datenbank-Monitor und Plan-Cache Verwendungsanzeige	A QQRID 3001 index used record for each EVI with QQRCOD = 'I8'


SMP parallel möglich	YES
Wird auch genannt als	Table probe, preload
Icon für Visual Explain	

Abbildung 5: Übersicht über die Eigenschaften der neuen EVI RRR-Probe

Meine Testumgebung

Als Schema für die Beispiele verwende ich das bekannte STAR-Schema. Bei mir heißt diese Bibliothek. STAR1GA. Diese enthält neben anderen Tabellen die Datei ITEM_FACT mit 6.001.215 Sätzen.

Ausprägungen für WHERE-Abfragen und für EVIs:

SHIPMODE = 7 Ausprägungen

`select count(distinct shipmode) from star1ga.item_fact → 7`

```
Anfang auf Zeile . . .
.....+.....1
SHIPMODE
SHIP
RAIL
MAIL
FOB
REG AIR
AIR
TRUCK
```

Abbildung 6: Ausprägungen der Spalte SHIPMODE

QUANTITY = 50 Ausprägungen

`select count(distinct quantity) from star1ga.item_fact → 50`

ORDERPRIORITY = 5 Ausprägungen

`select count(distinct orderpriority) from star1ga.item_fact → 5`

EXTENDEDPRICE = 913.278 Ausprägungen

`select count(distinct EXTENDEDPRICE) from star1ga.item_fact`

SQL-Abfrage ohne und mit Aktivierung von EOA

In den beiden folgenden Abbildungen werden die Abläufe mit Visual Explain dargestellt. Im ersten Fall wurde EOA de-aktiviert mit

```
UPDATE qaqqini SET QQVAL='*NO' WHERE QQPARM='ALLOW_EVI_ONLY_ACCESS';
```

The screenshot shows the Visual Explain interface for a query. The query plan on the left consists of four steps: Table Scan, Temporary Sorted List, Sorted List Scan, and Final Select. The right pane displays the following information:

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_CUSTK EY_00001 4
Statement Text	SELECT ITEM_FACT_1.ORDERDATE, ITEM_FA... TEST

The statement text at the bottom of the window is: `SELECT QUANTITY, ORDERPRIORITY FROM ITEM_FACT WHERE SHIPMODE = 'sdf' ORDER BY ORDERDATE DESC LIMIT 50 OFFSET 0`

Abbildung 7: SQL-Abfrage ohne EOA

Im zweiten Fall wurde EOA auf aktiv gesetzt mit

```
UPDATE qaqqini SET QQVAL='*YES' WHERE QQPARM='ALLOW_EVI_ONLY_ACCESS';
```

The screenshot displays a query execution plan for the following SQL query:

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

The execution plan consists of the following steps:

- Temporary Index**: Initial step at the bottom.
- Index Probe**: Labeled with `*NONE.MTI(14)`. An arrow with `2.514E7` points to the next step.
- EVI Random IO**: Labeled `STAR100G.EVI over ORDERPRIORITY`. An arrow with `2.514E7` points to the next step.
- Final Select**: The final step at the top.

On the right side, the **Additional Index Info** table is shown for the EVI index:

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	STAR1000
Member of Index Used	EVI_0001
Long Name of Index Used	EVI over ORDERPRIORITY
Long Library of Index Used	STAR1000
Name of Table Being Queried	ITEM_FACT
Library of Table Being Queried	STAR1000
Member of Table Being Queried	ITEM_FACT
Long Name of Table Being Queried	ITEM_FACT
Long Library of Table Being Que...	STAR1000
Index Only Access	Yes
Estimated Time Information (Sta...	
Processing Time(ms)	4,096
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 ex...	

Abbildung 8: SQL-Abfrage mit EOA

Vorschau:

Falls es weitere Neuigkeiten bei der Entwicklung von EOA, werde ich Sie darüber informieren.

Bis dahin wünsche ich Ihnen weiterhin viel Spaß beim Vermehren Ihrer Fertigkeiten.

Den Autor Theo Bär erreichen Sie unter EDV-Beratung Theo Bär - Ringmauerweg 1 - 69250 Schönau -
Tel. (+49) 6228 912 630 - e-Mail info@edv-baer.com