

# **R S InfoCon Inc.**

**- White Paper -**

## **Top 10 Reasons Why You Should Create Your Own SQL Indexes for JD Edwards**



**R S InfoCon, Inc.  
7071 S 13th St, Ste 205  
Oak Creek, WI 53154  
Tel: 262-995-7002  
Fax: 262-995-7082  
Email: [info@rsinfocon.com](mailto:info@rsinfocon.com)**

NOTICE: Proprietary

This material shall not be used, reproduced, copied, disclosed, transmitted, in whole or in part to a third party without the express consent of R S InfoCon Inc.

R S InfoCon Inc.

Tel: 262-995-7002

Email: [info@rsinfocon.com](mailto:info@rsinfocon.com)

## Overview

Over the years R S InfoCon consultants have come to discover that database administrators and application people seem to have an aversion to adding indexes to their systems. This appears to come from 'the old days', when machines were slow, cycles were expensive, and disk was expensive.

This observation is not limited to people in the 'i' world. It crosses over into the Microsoft and Oracle worlds as well.

## Feature

These observations have lead R S InfoCon consultants to come up with a 'Top Ten Reasons' why you should make your own SQL indexes in the JD Edwards EnterpriseOne world.

### **1. QBE**

All shops use QBE (Query by example). Even if you have QBE limited to existing indexes only, you probably have users complaining about how they want to get the data. JD Edwards didn't know in advance how you wanted to get at the data, so they shipped a set of indexes that helped performance for their way of using it. Users are VERY creative creatures. They do not really enjoy restrictions. But the thing they hate more than restrictions is wasting time trying to do something and then having to quit because it takes too long. New indexes solve both these problems. If you restrict QBE, you can give them more options by adding indexes. If you do not restrict QBE, then, by studying how the users use it, you can add the proper indexes to make their most painful selections much faster.

### **2. Row Security**

JD Edwards has offered you a very versatile method of controlling users to access your data. Unfortunately when JD Edwards decided what to ship to you, they had no way of knowing how you would want to set up security. This lack of foresight can cause the system to have to look at a large amount of data to decide how to fill a single screen. Row security is implemented by using a series of SQL 'BETWEEN' functions. If the item you are securing is not in the index the system gets to read the index, then read the record. In effect you get 2 reads, for the price of, well, 2 reads. If you would have had an index you would have effectively gotten 2 reads for the price of one.

NOTICE: Proprietary

This material shall not be used, reproduced, copied, disclosed, transmitted, in whole or in part to a third party without the express consent of R S InfoCon Inc.

R S InfoCon Inc.

Tel: 262-995-7002

Email: [info@rsinfocon.com](mailto:info@rsinfocon.com)

### **3. Dashboards**

CEO's and other managers love dashboards. They also know exactly what they want to see, and they want to see it NOW. Unfortunately as we all know, how they want to see it doesn't seem to conform to any known way that others want to see it. You are left designing how to get to the data yourself. If you try to do this without creating new indexes, even if they LOVE the results, they probably won't love the performance.

### **4. Custom Reports**

Everyone that does not have at least one custom report for their JD Edwards system raise your hand...Come on, anyone? If you have custom reports, by definition you are looking at something JD Edwards either did not anticipate, or decided was not worth it for them to do themselves. In either case why would you think that JD Edwards would have given you an efficient way to get at the data?

### **5. Modifications**

Many of the shops we have been at have implemented at least some modifications. Most of the time these are invoices, purchase orders, or work orders. There always seems to be a need for 'just that one extra piece of information'. The problem is that often the information that they want, doesn't seem to be in the files that the UBE or application has on hand. This means that you have to go get it yourself. In my experience, in at least 30% of these cases there is no efficient way to get that piece of information. The result is that the process that has been modified ends up dramatically slower.

### **6. 3rd party Add-ons**

There are a myriad of 3rd party add-on's to EnterpriseOne. From CreateForms, JetForms to Q-Software that provide extensions to the EnterpriseOne functionality. Each of these products gets at the data in ways that JD Edwards could not have possibly anticipated when they shipped your system. In order to make these add-ons fly you need to add indexes based on how you use them.

### **7. Ad-hoc query tools**

For those of you who raised your hand in #4, you probably don't have custom reports because you use 3rd party tools like Cognos or Crystal. These tools are great; they give you even more flexibility to get the data you want. The problem is that they give you even more ways to get at the data you want. This comes at the cost of forcing your system to go through loads more data. Just try writing a really cool report tying orders to your accounting (Account Master-F0901, General Ledger-F0911, Sales Header-F4201, Sales Detail-F4211), or even better against order history, with no new indexes. Give us a call next month, or next year when it finishes.

NOTICE: Proprietary

This material shall not be used, reproduced, copied, disclosed, transmitted, in whole or in part to a third party without the express consent of R S InfoCon Inc.

R S InfoCon Inc.

Tel: 262-995-7002

Email: [info@rsinfocon.com](mailto:info@rsinfocon.com)

### **8. What's a purge?**

OK, admit it. You keep telling your users that they need to purge that General Ledger-F0911 file. It's up to 10 million records. Their response, 'But we need 10 years of history...'. In our experience, most data more than a month old is never accessed again, except once at year end. Imagine if you could at least from an operational point of view ignore the 99% of the data that is 'old.' Indexes give you this ability, without having to purge.

### **9. Write once, update twice, read 100 (or 1000)**

OK, we haven't convinced you yet why you need to create your own indexes. Here is a technical reason to do so. The primary reason we hear that people don't want to add indexes is a fear of degrading performance.

Let's look at what actually happens in the life of an order line. First the order is taken. The line gets written (the order is taken), 1 index maintenance. The order line is then picked, packed, shipped, and invoiced (The index is maintained each time, total 5 index maintenances). You have maintained the index 6 times, and if you purge (see prior item), you get one more for 7. Also remember that indexes that have no fields changed don't get maintained.

Balance this against the times the record is read, you send an order confirmation (1), print a pick list(2), print the packing list (3), print the BOL (4), print the invoice (5), backorder report (6), product shipped report (7), daily sales report(8), weekly sales report(9), and the month end process (10). You now have 10 times you have read the record, on purpose. Now suppose you didn't have an index over the line number and status. Every time you need to update ANY lines status for this order you will get to read this line again just to eliminate it. These are what we call 'hidden' reads. They happen but you don't necessarily know they happened.

In reality we see records are actually updated a small fraction of the times they are accessed. It makes sense to bias your performance concerns towards making the reading more efficient instead of the write/update/delete more efficient.

### **10. Money**

As they say, time is money. The cost of disk and cycles (CPU) is going down, and shows no signs of stopping anytime soon. The cost of the people, is going up, and also shows no sign of slowing anytime soon. Well defined indexes reduce the time it takes the system to serve data up to your user community, reducing the costs of your business. You will also reduce the cost of the system itself to your business by reducing the amount of CPU needed to process your query, the amount of disk resources to access your data, and the amount of memory needed to store data while it is being evaluate

Now you have an idea of why R S InfoCon has written this white paper on the Top Ten Reasons why you should make your own SQL indexes in the JD Edwards EnterpriseOne world. We hope you take our advice and start to address these possible opportunities.

NOTICE: Proprietary

This material shall not be used, reproduced, copied, disclosed, transmitted, in whole or in part to a third party without the express consent of R S InfoCon Inc.

R S InfoCon Inc.

Tel: 262-995-7002

Email: [info@rsinfocon.com](mailto:info@rsinfocon.com)

---

This documentation is delivered as is. R S InfoCon, Inc. makes no warranty as to its accuracy or use. Any use of this documentation is at the risk of the user. Although we make every good faith effort to ensure accuracy, this document may include technical or other inaccuracies or typographical errors. R S InfoCon, Inc. reserves the right to make changes without prior notice.

R S InfoCon, Inc. and this publication are not affiliated with or endorsed by Oracle JD Edwards software

NOTICE: Proprietary

This material shall not be used, reproduced, copied, disclosed, transmitted, in whole or in part to a third party without the express consent of R S InfoCon Inc.

R S InfoCon Inc.

Tel: 262-995-7002

Email: [info@rsinfocon.com](mailto:info@rsinfocon.com)