

Speech 2—Part 2 Transcript: The role of DB2 in Web 2.0 and in the IOD World

Slide 1: Cover

Welcome to the speech, The role of DB2 in Web 2.0 and in the Information on Demand World. This is the second speech in the DB2 on Campus lecture series and its Part 2 of 2 parts. My name is Raul Chong and I'm the DB2 on Campus Program Manager.

Slide 2: Agenda

In Part 1 of this speech, we covered IBM's direction and then we explained how DB2 fits into this direction. Then we talked about the role of DB2 in the Information on Demand world and the role of DB2 in Web 2.0. In this second part of this speech, we'll start with What is pureXML?

Slide 3: Why use XML with Databases?

Before talking about pureXML, we need to provide some background information. So let's start with the question, "Why use XML with databases?" Well, databases are a very robust technology and they are very good to store large volumes of data in general and to retrieve them very quickly. So, databases are good for persistency, recovery, performance, scalability, etc. All of these reasons that make relational databases very good software also apply to XML data. The 2nd reason why XML is better at storing databases is integration and that means integrating relational data with hierarchical data together.

Slide 4: Integration: Relational and hierarchical data

So let me give you a short example here. Let's say I work as a medical researcher at a hospital and I'm trying to find a cure of cancer, and let's say I've been doing this for the past 25 years where I've been storing information about my patients in a table, where I have relational data like the name of the patient, the sex of the patient, the age, how many years this patient has been drinking coffee, how many years this patient has been smoking, and I'm also storing the DNA of each patient on the last column. And I'm storing this information as an XML document, as explained in Part 1 because XML is good to store unstructured or semi-structured type of information like DNA. Now let's assume that all cancer patients or most of my cancer patients have a gene X with a value of 987.

So what I can do with DB2 9 is to do some sort of integration where I can integrate relational data like coffee and smoking, which are some factors that may or may not cause cancer, and this gene X with a value of 987, which seems to appear in most cancer patients. Now as you can see as well from these charts, I am representing the XML document as a tree because an XML document is hierarchical and a tree is hierarchical in nature.

Slide 5: SQL with XQuery/XPath

So running the query that is shown in this slide we could perform this integration. The first two lines and the last four lines are just SQL as usual where I'm selecting the name from the patient's table and I'm doing some filtering based on the sex, the age, coffee, and smoking. The third and the fourth line is what are new with DB2 version 9. This is a call to a function that is part of the SQL 2006 standard. The function is called **XMLexist** and it allows me to perform some testing

in the XML document. So, basically here, on this 3rd and 4th line, what I'm doing is I'm using XPath and I am navigating the tree, which is the XML document to test that gene x is equal to 987. So basically, I am combining in this query SQL with XPATH or XQUERY, or I'm basically combining or mixing the relational model with the hierarchal model of XML.

Slide 6: See a DEMO: “Think of the possibilities: DB2 & pureXML”

Now I explain these examples in more detail using DB2, the Control Center, and the Command Editor in another series, which I called "Think of the possibilities—DB2 on pureXML" and Episode 1 talks about this demo. It's called "How pureXML may help to find the cure of cancer." So I invite you to take a look at that other series and that first episode.

Slide 7: Databases with XML

Moving on in this presentation, let's talk about how XML can be stored in databases. There are two classifications: the first one is to store XML in what is called "XML-enabled databases" and I call this the *old technology*. The 2nd way is to store them using native XML databases and I call this the *new technology*. DB2, starting with version 9, is a native XML database.

Slide 8: 1) XML-Enabled Databases: Two Main Options

So let me first explain the old technology, which is “XML-enabled databases.” In the old technology, there were two main options for storing XML. The first option is called CLOB or varchar, and what that means basically is that you will be storing your XML document as a string in a varchar or clob column of a table in a database. So basically, you know, an XML document is hierarchical in nature, and when you store it as a string, you're basically flattening this tree structure of this XML document. And then what's going to happen is if you want to retrieve your XML document, let's say you're doing XPath and XQuery and you want to get part of that XML document, you may have to first get the string and then do the parsing. So your application will have to do the parsing while it needs to build the tree, and then after it builds the tree, it can navigate the tree to get what it wants. Of course, I'm simplifying the explanation here. There's SAX parsing and DOM parsing. I'm basically explaining what DOM parsing is, which is building the tree. And those topics are not going to be covered in this presentation, so I'm just simplifying the explanation here.

Now the other option within the old technology is called *shredding* or *decomposition*, and what that means, is you basically have at the top an XML document and then you're going to shred it or decompose it into little parts so that each part has some part of the XML document. Now the problem of this method is, if you want to get your XML document back, you probably need to join these tables, and an SQL join operation is expensive. Now, in this example, we only have 3 tables, but assuming you had a more complex XML document that may require 50 tables, 60 tables, and so on, so doing a join operation on 50 tables or 60 tables is really expensive. The other problem with this method is that if you want to make a change to the XML document, sometimes that change may not be propagated to the tables one to one. You may have to create two more tables or three more tables depending on the change that you make. So in both of these methods or options within the old technology there is a problem in terms of flexibility and in terms of performance. In the case of the first option, your application will have to pay the price because it has to do the parsing. And then you can do the navigation of the XML, but first you have to do the parsing.

Slide 9: 2) Native XML Databases

Now let me explain to you the *new technology*, which are [uses] native XML databases and which is what DB2 version 9 uses. So with this new technology, in DB2, what we do is create a table as usual. We can create relational columns, we can use relational columns, but now we will want to store XML documents. We create a column and we use a datatype XML like in this example. The datatype column is using the XML data type. Now I'm sure you've seen in other relational data servers; they also have a data type called XML. The difference is on the implementation. What do they actually do with this XML datatype? I will explain to you what IBM DB2 does when you work with this XML datatype. So when you are inserting an XML document into this column, DB2 at that time will do the parsing. That means it will build the tree or the hierarchical nature of your XML document and it will store it like that in your database so it will be persistent. So what that means is that this is done ahead of time, at insert time, when you are working with the XML document at query time. You don't need to build a tree any more. The tree was built before. Therefore you save time already there. There is a performance improvement there. Now, in addition to this, we also build some internal indexes, and we also let you build some XML indexes yourself. So this would possibly or potentially allow the DB2 engine to not bring the entire tree to memory, but just what it needs to process the query. The XQuery that you're doing or the XPath that you're doing saves in terms of performance.

Back to Slide 10: 1) XML-Enabled Databases: Two Main Options

And if we compare it to the first option of the old technology, here, what we were doing is, we were flattening the tree and we were storing the XML as a string and then we had to build a tree at runtime in order to parse it.

Back to Slide 11: 2) Native XML Databases

But, in this new technology, with native XML databases in DB2, we don't need to build a tree at runtime. The tree was build ahead of time when we inserted it into the database. So that's the first main characteristic of DB2 native XML support or we also call it pureXML as I will explain in a few minutes. Now what is the second characteristic of this native XML support? Well for this let me show you with a graph like the following.

Demo in Paint

So let's say what I'm drawing here is a relational engine. And as far as I know, most of the vendors today have only a relational engine. Of course they may have bought another company that has an XML engine, but these are two separate products. So let's assume I have this engine, of another vendor, which is not DB2. And now what they say is that they support XPath and XQuery, which is the language that you use for navigating an XML document. So if they're running XPath or XQuery and they have a relational engine; a relational engine cannot understand XPath and XQuery. So probably what they will have to do is they will have to map it to SQL. Probably it will have to go like this: they have the XPath, they have to map it to SQL, and then the SQL will go down to the relational engine, which can understand SQL. Now, of course, as well, this means that the information is stored as a table. So, as you can see from this chart we have that they are trying to do an XPath or XQuery and they first have to map it as SQL, which goes to the relational engine, and they store all the information as a table. Now this mapping means there will be degradation in performance.

Now let me explain how DB2 works with pure XML. In DB2 what we have is two interfaces to the engine: one of the interfaces was a relational part, and the other part is going to be an XML part. Now, if you are doing XPath or XQuery with DB2 you don't need to map it to SQL. What we just need to do is we go directly to the XML engine. So it goes directly to the XML engine or interface to the engine, which is native. And then with this, we will go and access the storage, which as I explained to you before, is stored as a tree, persistent, it's stored like this in DB2. So let me just put it here to complete the chart. So, these things here—all the way from here—all the way here—is what we called DB2 pureXML. So, we have again two parts or two main characteristics of DB2 for XML or native XML, which is, one is the fact that we store the XML document as a tree in the hierarchical way, and the second one is that we have a native XML interface to the engine. So there is no mapping. That is what is called "native". There is no mapping required; we don't need to convert this to SQL. So that's native XML support.

Back to ppt

Now let's go back to the presentation.

Slide 12: XML in DB2

In this other chart I explain exactly what I just mentioned the fact that we go from a client to a server and we can use SQL to go through the relational interface of the engine and access relational data. We can also use SQL with XML extensions and that will access XML data and we can also use XQuery with the XML interface and it can access XML or relational data. So there are 4 ways in which we can access data now with DB2 version 9.

Slide 13: pureXML™ Support... More details

Now I've been talking about pureXML and I've also been talking about native XML. What is the difference? Well, basically there is no difference, it's the same thing. What happens is if you go to Google or any search engine and you type "native XML" you may find hits from manuals of other database vendors because they say that they have native XML since they have an XML data type. But in the case of IBM DB2 perspective, we don't consider that they have the same support that we have. We don't think that they really have native XML support, and I already explained the two things that come with native XML, pureXML, support. So basically what IBM decided to do is (that) we decided to create a new term and we called it pureXML to differentiate ourselves from these other vendors. So, because, as I said before, if you go to Google and type "native XML" you may find things from other vendors, and if you were a user you would be confused because you would say, "Oh, ok, Oracle also provides native XML support, SQL server also provides native XML support, DB2 also provides native XML support." However, it is actually a different type of support. DB2 provides a different type of support than these other vendors and that's why we decided to call our support pureXML, which we believe is really what we should define as native XML having a native XML interface or engine and having or storing the information as a tree.

Now we also made some changes to the APIs to provide support for XML in languages like COBOL, C, Java, etc. etc. We provide, as I said before, support for creating your own XML indexes. We provide support for XML schema, and DTD repositories in terms of performance and scalability.

Slide 14: DB2 9: Feedback from partners/clients

What our customers and partners are saying or are seeing huge improvements in the manipulation of XML. This is an example of a company called Storebrand, which did a test using the old technology and the new technology and saw huge improvements.

Slide 15: Awards

In this chart you can also see that DB2 9 is being recognized by the industry as a good product. We won the 2007 CODiE award.

Slide 16: XQuery embedded in SQL

This is also a slide showing some more explanation or an example of the things you could do with XQuery and SQL. You can combine SQL with XQuery, so at the top level you can have SQL and embedded you can also have XQuery.

Slide 17: SQL embedded in XQuery

You can also do the opposite where at the top level you can have XQuery and embedded in red in this example you can have SQL. So there are many things you can do with pureXML in DB2 9. There are a lot of powerful combinations you can perform.

Slide 18: More details for pureXML™?

If you want to know about pureXML I recommend that you take a look at Lesson 15 of this DB2 on Campus lecture series, which is called “pureXML, SQL, XML and XQuery.”

Slide 19: Agenda

So let's move on to the next section of this presentation, which is “What is DB2 Express-C?”

Slide 20: DB2 Data Serving Solutions

What you can see in this chart are the different editions of DB2. Now, DB2 Express-C is the core of DB2. The C stands for community. So we created DB2 Express-C to give it away for free to the community. It's not that we're using different code and we are packaging that and saying that this is DB2 Express-C and giving it away to the public; we are actually using the same code that we use for other editions, which are fee-based. So DB2 Express-C is the core, and then when you add some more functionality to that it becomes DB2 Express Edition; when you add more functionality to that, it becomes DB2 Workgroup Server Edition and so on and so forth. We'll talk more about those other editions in other sessions of these DB2 on campus lecture series. Now it is important again to know that because we are giving... because DB2 Express-C is the core of DB2, basically what we're saying if you develop an application for DB2 Express-C, that same application, with no modifications, will work on all the other editions because, as I said before, it's the same code, it's the core that we're giving to you away for free.

Slide 21: DB2 Express-C—Jan 30, 2006

So, DB2 Express-C was announced, was available starting January 30, of 2006 and the motto that we use is it's free to build, deploy, distribute, with no limits. What it means, is free to build if you are developing an application and you need to connect to a database or data server, there you have DB2 Express-C, which you can use for free.

- **It's free to deploy:** what that means is that if you are working in production or in a commercial setting and you need a database, let's say it's your internet company and you need the database, you can use DB2 Express-C in production or in a commercial setting, and it's free.
- **It's free to distribute:** that means, let's say you are developing your own application and you want to embed a database as part of this application, well you can embed DB2 Express-C as part of your application and then you can distribute your application, and you can sell your application to other people. And even though you're distributing the code of DB2 as part of your application, you don't need to pay anything to IBM, it is free. So, potentially, you could develop a very simple interface or a very simple application on top of DB2 where DB2 is doing most of the work and you can sell your application, and even though you are making money, and including DB2 in your application, you don't need to pay anything to IBM. The little asterisk here (*) just means that you need to register when you do that because we just want to keep track of the statistics for our own internal purposes.
- And finally, there are **no limits:** no limits meaning that no database has limits comparing to other data server vendors, which have about a 4 GB (gigabyte) database size limit. There's no limit on the number of instances per server, there is no user limit, and with DB2 Express-C, we're also including pureXML, so everything that you get with DB2 Express-C is free.

The latest version in DB2 is DB2 9.5. In terms of support, you can get some free support from the community. We have a forum that you can access through the main web site of DB2 Express-C, which is www.ibm.com/db2/express and from there you can get support from the community, and it's also monitored by members of the DB2 Express-C community team. In terms of the skills, everything that you learn about DB2 Express-C will also apply to other editions of DB2.

Slide 22: DB2 Express-C 12 Months License and Subscription

So this is what I just talked about, it's free to build, deploy, distribute, pureXML is included, community assistance, which is for free, and you can download or get a free DVD from the main DB2 Express-C web site. In order to get DB2, we also offer, in the case where let's say you have a company and you need support, you have... this is a critical database that you have, and you're using it in production for your company and you need 24x7 support from IBM, we will also provide that, and we also provide SQL Replication and High Availability with that, with a 12-month license. This is costing now about \$2,995 US dollars per year per server and this is, as I said before, in the case that you want support from IBM.

So we provide two types of licenses: the first one on the left is the one that is free and unwarranted, and the one on the right is the one where you have to pay but you can get support from IBM and it also has a couple of other features, which are SQL Replication and High Availability. You can read more about this from the main web site of DB2 Express-C, which is www.ibm.com/db2/express.

Slide 23: Application Development Freedom

We also talk about application development freedom when we talk about DB2 because we provide all the drivers and extensions that you need to work with all of these languages.

Slide 24: Agenda

Finally, we move on to the last section of this presentation, which is two approaches for free application development.

Slide 25: J2EE – Free development platforms

Since most of my audience is university students or the open source community, I picked two free applications, free development platforms because these are more appealing to the open source or university community. The first one is maybe you want to develop J2EE applications and you can use Linux as your operating system. You could use Eclipse as your IDE; you can use Apache Derby for your data server or DB2 Express-C. Now I included here DB2 Express-C in parenthesis because, so far, all the things that I mentioned, Linux, Eclipse, Apache derby are open source, which means you can see the code and they are free. In the case of DB2 Express-C, it is not open source you cannot see the code, but, it is free so that's why I include it in this chart in parenthesis. Most people today when they use a data server they just want to take advantage of what the data server can do for their application; they are not interested that much in modifying the engine of the data server itself.

You can also have an application server for free and I included WAS-CE also in parenthesis WAS-CE means WebSphere Application Server—Community Edition. I put it in parenthesis again because it is not open source, but it is free. And with all of these you can also develop also your Web Services. So, all of these are free development platforms. You can use them now and pretty much all of them were donated by IBM. So IBM is a company that supports the open source community.

Slide 26: Web 2.0 Languages: PHP & Ruby on Rails

Now, the second way to develop or the second free platforms are using what I call the Web 2.0 languages. And again, I picked these two because my presentations are mainly for students or the open source community. So here I mention PHP and Ruby on Rails; both of them are open source and free. What we have created are a couple of packages to make your learning of these languages more easily. So, for example, we have ZenCor for IBM; this is actually a package developed by a partner of IBM called Zen. They have developed this package, which are called ZenCor for IBM, which includes PHP, DB2 Express-C and all the drivers and extensions that you need to work with PHP. The nice thing of using this, as I said before, is that you can download it and use it for free from the website that is mentioned here and, everything, once you download and install it, you are ready to go and you can start working with PHP. So, you don't need to go to one web site to get PHP; you don't need to go to another web site to get DB2; you don't need to go to another web site to get extensions or drivers that you need. Everything is packaged together and you're ready to go. We also have another package called Web 2.0 Starter Toolkit for IBM DB2, which includes some demos of applications that you can develop using PHP and DB2 Express-C and other web 2.0 technologies. So this is also a free package and you can download it from the link that is mentioned in this slide.

And finally we have another package, the Startup Toolkit for DB2 on Rails. This is a package also free to download and to use, which comes with Ruby on Rails and DB2 Express-C and all the other drivers and extensions that you need to work with Ruby on Rails and DB2. So you can get it for free from this link and it's a very good and exciting way to get started on this new and exciting technology, which is Ruby on Rails.

Slide 27: Agenda

So, finally we will talk very briefly about how to contact us.

Slide 28: Contacts

You can either send me an email or maybe you will be interested in looking at other videos or taking a look at the blogs from the sources that are mentioned in this slide.

Slide 29: The DB2 on Campus Lecture series

And here is a summary of the different speeches and courses that are part of the DB2 on campus lecture series.

Slide 30: The DB2 Express-C Web site

Again, everything, or the only one web site that you really need to remember is www.ibm.com/db2/express, which is the main web site of DB2 Express-C.

Slide 31: In Summary

So in summary, in the first part we first talked about IBM's direction and the role of DB2 and in this second part we talked about pureXML, what is DB2 Express-C, and two approaches for free application development.

Slide 32: Thank you! What's Next?

So I thank you very much for listening to this speech and if you are still interested about learning more about DB2, I encourage you to listen to the 3rd speech on this DB2 on Campus lecture series, which is: "Your opportunity—Ten steps to career success." Thank you.