مجلة جامعة تشرين للبحوث والدراسات العلمية _ سلسلة العلوم الهندسية المجلد (38) العدد (5) 2016
Tishreen University Journal for Research and Scientific Studies - Engineering Sciences Series Vol. (38) No. (5) 2016

مستقبل أنظمة قواعد المعطيات الأصيلة باستخدام لغة التأشير الموسعة كبديل لأنظمة قواعد المعطيات التقليدية

هاني محمد حسن*

(تاريخ الإيداع 6 / 6 / 2016. قُبِل للنشر في 6/ 9 / 2016)

□ ملخّص □

أصبح مصطلح قاعدة المعطيات العلائقية التقليدية مرادفاً لمصطلح قاعدة المعطيات، ولكن احتكار الشركات الكبرى المصنعة لأنظمة إدارة قواعد المعطيات أصبح يشكل هاجساً للعاملين في هذا الحقل نظراً للكلف العالية لهذه الأنظمة، مما وجه الأنظار نحو الاهتمام بالتقنية الصاعدة وهي أنظمة قواعد المعطيات الأصيلة باستخدام لغة التأشير الموسعة، حيث أنها تتمتع بخاصية المحمولية وأغلبها مجاني أو مفتوح المصدر، وبسبب زيادة الاعتماد على ملفات xml و لا سيما في التراسل بين التطبيقات المختلفة ووجود مجمعات لها مما استوجب التوجه نحو أنظمة قادرة على إدارتها وتنظيمها، وهذا فرض الحاجة إلى أنظمة قواعد المعطيات xml الأصيلة، وتهدف الدراسة إلى إجراء مقارنة بين إمكانيات أنظمة قواعد المعطيات العلائقية التقليدية وأنظمة قواعد المعطيات xml الأصيلة وفقاً لمعايير عديدة ، واستثمار التقنيتين في تطبيق عملي وإجراء الاختبارات المناسبة التي تعكس أثر استخدام كل من التقنيتين على أداء التطبيق المقترح وعرض النتائج وتقديم الاقتراحات المستقبلية.

الكلمات المفتاحية: قاعدة المعطيات العلائقية، قواعد المعطيات الأصيلة، لغة التأشير الموسعة، المحمولية، مجمعات.

* **ماجستير بهندسة البرمجيات– كلية الهندسة المعلوماتية – جامعة دمشق– سورية.**

مجلة جامعة تشرين للبحوث والدراسات العلمية _ سلسلة العلوم الهندسية المجلد (38) العدد (5) 2016
Tishreen University Journal for Research and Scientific Studies - Engineering Sciences Series Vol. (38) No. (5) 2016

The Future of Native XML Database Systems as an Alternative for the Traditional Database Systems

Hani Mohamad Hasan*

(Received 6 / 6 / 2016. Accepted 6 / 9 / 2016)

□ ABSTRACT □

The mentioning relational database system term has become a synonymous to database system, but the monopoly of big companies that work in database systems field has become an obsession for persons who work in this field, because of the high costs of this systems. For this reason the concerns turn towards the advanced technique : the native XML database systems which are free or most of them are open source systems because of the increasing dependency on XML files and particularly in transporting data between different applications and the availability of collections of related files. This has summoned towards a system to manage and organize them, for this reason the NXDs appeared. The aim of this study is making a comparison between the capabilities of RDBMS and NXDs in accordance to multiple standards , investing these two techniques in practical application , make the relevant tests which reflect the use of these techniques on the suggested application , display the results and give future suggestions.

Keywords : Relational database, Native Database system, collections, Extensible markup language.

*Master Degree in software engineering- Faculty of IT Engineering.

مقدمة:

بات احتكار الشركات الكبرى المصنعة لأنظمة إدارة قواعد المعطيات يشكل هاجساً للعاملين والباحثين في هذا الحقل نظراً للكلف العالية لهذه الأنظمة لذا صار من الضروري التوجه نحو البحث عن بدائل حقيقية تتمتع بخاصتين مهمتين: أن تكون أنظمة قواعد المعطيات محمولة وأن تكون الأنظمة مجانية أو مفتوحة المصدر ومن هنا بدأت قواعد المعطيات الأصيلة باستخدام XML تكتسب أهميتها ودورها ولاسيما التخزين في تطبيقات الانترنيت و استخدامها في تبادل البيانات كونها تتيح التخزين بصيغة معيارية وبآليات سهلة وغير معقدة التصميم ومدعومة بلغات استعلام متعددة الإمكانيات.

أهمية البحث وأهدافه:

الهدف من مشروع البحث هو إجراء دراسة مقارنة بين أنظمة إدارة قواعد المعطيات XML الأصيلة وأنظمة إدارة قواعد المعطيات التقليدية(وتحديداً النظام الذي سيستخدم في الدراسة هو النموذج العلائقي Relational Model كممثل عنها كونه الأكثر والأوسع انتشاراً وشهرة).

وتعرض الدراسة مقارنة معمقة بين إمكانيات أنظمة قواعد معطيات XML الأصيلة (Native XML DB) وبين إمكانيات أنظمة قواعد المعطيات التقليدية (RDBMs) وتجيب عن تساؤل مهم يتعلق بأنه هل يمكن الاستفادة من الصيغة النصية و المحمولية(Probability) التي توفرها XML لاستخدامها كصيغة نهائية لتمثيل قواعد المعطيات ضمن بيئة عمل حقيقية وبالتالي هل يمكن لها أن تكون بديلاً عن أنظمة قواعد المعطيات العلائقية؟ وهل يمكن لها القيام بمهام هذه الأنظمة وتضمن الأمن والمعالجة الفعالة والتخزين و العرض للبيانات المخزنة؟

و تجدر الإشارة إلى أن المقارنة المذكورة لا تكتفي بالمقارنات النظرية إنما سننتقل إلى المجال العملي التطبيقي للمقارنة وذلك من خلال بناء تطبيق متكامل باستخدام قواعد معطيات XML الأصيلة، ونفس التطبيق باستخدام قواعد المعطيات العلائقية وذلك على نفس المعطيات ثم تقييم الأداء و إجراء المقارنة المطلوبة باستخدام مجموعة من الأدوات البرمجية المتاحة و بشكل يعكس الفروقات ويعرض كل ما يمكن من التباينات التي يمكن ملاحظتها وعكسها من خلال التطبيق المعتمد وخاصة في مجال الأداء، وهنا لابد أن نشير إلى أن الفرق الأساسي هو مرتبط بالوضع أو نمط الاستعلامات التي سوف ننفذها وحجمها وعدد المستخدمين للنظام.

لذا في البداية سوف نتحدث عن التقنيتين المذكورتين بشكل عام ثم ننتقل إلى الخوض في تفاصيل المقارنات تبعاً لمعايير عديدة وننهي البحث بالجانب التطبيقي الذي سيبرز الفروقات متحدثين عن التقنيات والأنظمة المستخدمة وميزاتها.

طرائق البحث ومواده:

قواعد المعطيات العلائقية:

النموذج العلائقي في قواعد المعطيات هو من أكثر النماذج انتشاراً وهو نموذج شائع يعود العمل به لمرحلة السبعينيات و يعتمد على تخزين البيانات في جداول مختلفة، ترتبط مع بعضها البعض بعلاقات تحدد يدوياً من قبل مدير النظام، وفيها كل جدول هو مجموعة من السجلات وكل سجل هو مجموعة من الحقول وكل حقل هو ثنائية من (اسم الحقل، قيمة الحقل) وجميع الحقول التابعة لجدول معين تمتلك نفس العدد من الحقول.

النموذج العلائقي يُعرِّف قيود معينة وصريحة على الجداول، ويُعرِّف مجموعة من العمليات عليها ويستخدم لغة معيارية من أجل الاستعلام وهي لغة الاستعلام SQL وهي لغة قريبة للغة البشرية سهلة التعلم ومزودة بإمكانات عديدة تجعل استخدامها يسيراً.

معظم الإصدارات الحديثة من أنظمة قواعد المعطيات التقليدية أضافت قدرات وإمكانيات لاستضافة وإدارة ودعم وتخزين واسترجاع المعطيات من نمط XML، فهي تدعم XQuery و XPath وغيرها من التقنيات، فنظام ORACLE بإصداره الحديث أتاح تخزين غير محدود الحجم لعقد وثائق XML بعد أن حددها سابقاً بمقدار 64 كيلو بت.

قواعد معطيات XML الأصيلة:

السبب الأساسي في انتشار قواعد معطيات XML الأصيلة هو الاستخدام المتزايد في الاعتماد على XML في نقل البيانات، ولكن قبل الخوض في مفهوم قواعد معطيات XML الأصيلة، لابد لنا من التعريف بلغة التأشير XML التي أكسَبت بميزاتها أهمية كبيرة لقاعدة المعطيات التي تستخدمها.

لغة التأشير XML:

(Extensible Markup Language) XML: هي إطار عمل لتعريف لغة تأشير مرنة وملائمة للعمل عبر الويب ومطورة من قبل (W3C) World Wide Web Consortium وتتميز عن لغة التأشير HTML بأن المعرفات (tags) لا تتطلب تعريفاً مسبقاً حيث أنها تسمح بإنشاء معرفات وبشكل حر وبما يتلاءم مع التطبيق، مما يُكسبها مرونة كبيرة ويسمح لها بوصف المعطيات المخزنة وتركز على ماهية البيانات المخزنة وتجيب على تساؤل يتعلق بما هي البيانات المخزنة؟ أي What Data Is?

تُعتبر لغة XML أكثر لغات التأشير استخداماً، حيث تم دعمها على نحو واسع، وفرضت وجودها بما أتاح لها مترجماً خاصاً على كل المتصفحات في الويب، وأصبحت تُستخدم لتخزين البيانات، إذ تتألف وثيقة XML من عناصر عديدة وكل عنصر منها يحدد بعلامة ومعرف بداية و معرف نهاية ومحتوى بينهما، لا توجد كما ذكرنا أعلاه علامات معرفة مسبقاً، مما يتيح لمنشئ الوثيقة أن يُعرف علامات خاصة به باستخدام قواعد منطقية وسهلة الاستخدام، وإذا نظرنا لقاعدة المعطيات على أنها تجمع لبيانات عديدة يمكن اعتبار وثيقة XML بأنها قاعدة بيانات أو مخزن لها، حيث أن XML تعرف كعائلة من التكنولوجيات التي تمكننا من تبادل الرسائل بشكل مستقل عن بيئة التنجيز وهذا هو سبب انتشارها(Kay,2014).

وتتميز لغة التأشير XML بمجموعة من الخصائص أكسبتها الأهمية والانتشار الحالي:

- تدعم عدة صيغ لأنماط المعطيات وتسمح بتعريف الأنماط التي نريد بما فيها بيانات الوسائط المتعددة.
- مستقلة عن بيئة التنجيز مما جعلها صيغة معيارية لتبادل البيانات بين مختلف التطبيقات البرمجية المكتوبة بلغات مختلفة.
- البساطة: فهي لغة قريبة للغة الإنسان المحكية وبالتالي هي سهلة الفهم والقراءة، وسهلة المعالجة عن طريق الحاسوب والتطبيقات المختلفة وذلك لاحتوائها على العلامات والواصفات المختلفة، وبالتالي الهيكلية المناسبة لترجمة وفهم المحتوى مما يفتح آفاق جديدة لمحركات البحث الفعالة والتنقيب عن المعطيات في الويب.
- قابلية التوسع وإضافة معلومات كلما احتاج المطور ذلك والتحرر من عملية التقيد والالتزام بالبنى والهيكليات المسبقة التعريف مما يوفر مرونة قل نظيرها، ويمكن من إضافة المطلوب وتوسيع الوثيقة كلما تطلب الأمر ذلك.

- تصف نفسها بنفسها وذلك باستخدامها للواصفات والمعرفات والعلامات السهلة الفهم والقراءة فلا حاجة عندها لبنى ومعطيات إضافية لفهم الوثيقة ومعرفة محتواها فهي تعتمد هيكلية تعبر عن ذاتها وتشرح عن نفسها ومحتواها.
- البنية الشجرية التي تتيحها XML تجعلها مناسبة وملائمة لحفظ الكثير من البيانات التي يتوافق توصيفها مع هذه الهيكلية.

وبالعودة إلى مفهوم قواعد بيانات XML الأصيلة (NXDs):

فالتعريف الصوري لقواعد معطيات XML الأصيلة بأنها: تعرف نموذج (منطقي) لأجل وثيقة XML تسمح بتخزين وثائق XML واسترجاعها بناءً على هذا النموذج ويجب أن يتضمن النموذج المعتمد بالحد الأدنى على: عناصر، واصفات، أنماط وغيرها، وهي تعتمد وثيقة XML كوحدة أساسية في التخزين بينما هي الصف (row) في قواعد المعطيات العلائقية، ومن خلال هذا التعريف يمكن لنا أن نستنتج بأن NXDs مُصممة لتخزين وثائق XML كما هي مع آلية لإضافتها واسترجاعها، ونموذج التخزين يُمكن أن يختلف ليَدعم الطريقة التي سيتم فيها الاستعلام واسترجاع البيانات، والسبب الأساسي في انتشار هذه القواعد هو اعتماد وثائق XML كصيغة لتبادل المعطيات بين التطبيقات المختلفة، حيث تُخزن البيانات في القاعدة بصيغة مستندات XML كوسيلة للإرسال والاستقبال، مما جعل NXDs أكثر كفاءة وجودة من ناحية كلفة التحويل إلى صيغ أخرى تتلاءم والتطبيقات المستقبلة وأفضل لتخزين البيانات بصيغة XML، بينما يتطلب الأمر جهداً كبيراً في عمليات التبادل في الأنظمة التي تتطلب تحويل بياناتها إلى صيغة ملفات XML من أجل التراسل اللاحق.

من الجدير بالذكر أن قاعدة معطيات NXD تسمح بحفظ أي نوع من مستندات XML، بغض النظر عن هيكلية الوثيقة المخزنة وبدون فرض قيود مُسبقة على شكل وصيغة وثيقة XML.

يتم تجميع وثائق ومستندات XML ضمن حاويات (containers) أو مجمّعات (collections) والتي يمكن أن تجزأ لاحقاً إلى مجمعات جزئية، وتدار بوساطة نظام يسمح بالقيام بكافة العمليات المعتادة المعروفة والتي يتم إجراؤها على البيانات: كالاستعلام، والإضافة للبيانات والحذف منها والمناقلات بكافة أشكالها، كما يمكن لقاعدة المعطيات إدارة أكثر من مجمع (collection) في نفس الوقت أو تنجيز هرمية من المجمعات.

إن أنظمة NXDs ولأجل تسهيل عملية الاستعلام عن وثائق XML أوجدت مجموعة من النماذج ولغات الاستعلام ومنها Xpath وXQuery وسنتحدث عنها الآن ببعض التفصيل:

XPath:

هي لُغة لأجل عنونة أجزاء من مستندات XML وذلك بشكل بنية هرمية، وتدعم استخدام مجموعة من العمليات للاستعلام على أجزاء محددة من مستند XML، فهي تؤمن مجموعة من التوابع لأجل النفاذ للمعلومات المتواجدة في عُقد الوثيقة (document node)، حيث تعمل على البنية المنطقية لوثيقة XML، وتمثلها كبنية شجرية، والقيم التي يتم استرجاعها بوساطة الاستعلام هي مجموعة من العقد وأنماطها تختلف بحسب نوع العقدة.

Xquery:

هي محاولة لإيجاد لُغة استعلام تُوفر نفس الفعالية والإمكانات التي تؤمنها لغة الاستعلام SQL المستخدمة في الاستعلام ضمن قواعد المعطيات العلائقية.

XQuery لديها نظام أنماط معقد مبني باستخدام XML Schema Datatypes ويدعم المعالجة لعُقد المستند، وهي ليست مُصمَّمة لأجل التعاطي مع مستند XML كامل فقط، بل هي تستطيع المعالجة مع جزء من مستند XML ومع تتابع مختلف من المستندات ومع تتابع مختلف من أجزاء من المستندات بشرط أن تكون well–formed، والهدف الحقيقي من XQuery هو إتاحة إمكانية إنشاء استعلامات مرنة تستطيع استخراج ومعالجة المعلومات من مستندات XML.

وهي لغة مدعومة بمجموعة من الإجرائيات التي تتيح القيام بعمليات مختلفة، وفي حال عدم توفر إجرائية ما، فإمكان المطور كتابتها بنفسه.

يوجد الآن العديد من أنظمة إدارة قواعد المعطيات XML الأصيلة المتوفرة في سوق الأعمال، بعضها مفتوح المصدر وبعضها الآخر تجاري، ولكن ليس من الضروري أن تكون البرامج التجارية المعروضة منها(مثل NaxBase,Tamino,Ipedo,Total XML) هي الأفضل والأسهل استخداماً فبعض البرامج المفتوحة المصدر منها(مثل Ozone,Exist,Xindic) تمتلك إمكانيات تكاد تكون شبه كاملة وهو ما سنحاول الاستفادة منه في معرض دراستنا.

إن التقنيات المرتبطة بأنظمة قواعد المعطيات XML الأصيلة تُعتَبر حقلاً مهماً للنشاط والعمل ولكنها كما غيرها من الأدوات تمتلك العديد من نقاط القوة والضعف وهي:

- جيدة لأجل تخزين واسترجاع المعطيات إذا هي بالأصل مُخزنة سابقاً بصيغة XML.
- تُمكن من استعادة الوثائق في شكل ونموذج قابل للقراءة والفهم.
- معظمها تأتي مع محرك بحث عالي الجودة والإمكانات.
- يمكن أن تخدع المستخدم عند البدء بفهرسة الوثائق ومن ثم إجراء عملية البحث.
- لا توفر أدوات وإجراءات مناسبة لأجل القيام بعمليات تجميع جيدة.

المقارنة على أساس النموذج والتقنيات المعتمدة:

نُلخص في الجدول التالي مقارنة أو نوعاً من التقابلات بين نظام إدارة قواعد المعطيات XML الأصيلة ونظام إدارة قواعد المعطيات العلائقية، وذلك من ناحية نماذج التخزين المعتمدة وطرائق العرض والمعالجة:

Relational Database	Native XML DB
SQL	XQuery, Xpath
Table /row/	XML document
JDBC/ODBC	DOM API/ XAPI
XPath Model	Relational Model
Relational model is unordered set of values	XML document is a tree of nodes
Retrieve multiple result (rows..)	Retrieve documents, subdocument

يُبين الجدول أعلاه نوعاً من المقارنة الأولية بين النظامين، حيث تَعتمد أنظمة إدارة قواعد المعطيات العلائقية لغة الاستعلام SQL وهي لُغة إجرائية مشهورة وسهلة الفهم، في حين أن اللغة المعتمدة للاستعلام في قواعد المعطيات XML الأصيلة هي نموذج XPath ولغة الاستعلام XQuery وفي حين أن SQL تدعم القيام بعمليات أساسية: كالإضافة والبحث والتعديل والحذف، فإن XQuery تمكننا أن نستعيد المعلومات من وثيقة XML أي القيام بعمليات استعلام (Selecting) فقط، فهي بدايةً لا تمكننا من الإضافة و الحذف من قاعدة البيانات أو التعديل عليها، مما شكل فجوة كبيرة في الإمكانيات، ولكن في الوقت الحاضر معظم أنظمة NXDs قد ملأت تلك الفجوة باستخدام ما سمي XUpdate الذي هو ما يزال حتى اليوم قيد التطوير وقد تم تضمينه في كل النسخ الحديثة للغات الاستعلام على وثائق

XML، وبالتالي تطورت إمكانات ومقدرات لغة الاستعلام Xquery عن طريق إضافة تسهيلات مختلفة لعمليات التعديل.

معظم قواعد المعطيات الحالية NXDs تدعم لإنجاز استعلاماتها لغة XQuery التي تتضمن نموذج XPath مضافاً إليه بعض قدرات التحويل (Transformation Capabilities) وأنظمة NXDs ينبغي عليها أن تدعم على الأقل نموذج واحد لأجل الاستعلام هو على الأقل XPath وذلك لأجل الاستعلام على مستند أو تجمع من المستندات.

للنفاذ إلى مخزن وثائق XML فإن أنظمة NXDs تدعم واجهة تسمى XAPI وهي مستقلة عن بيئة التنجيز الفعلية للنظام وهي تقابل التقنيات المعروفة لنفاذ النظام إلى مخزن البيانات في أنظمة قواعد المعطيات العلائقية وهي ODBC و JDBC، حيث أن XAPI، ODBC وJDBC هي الواجهات البرمجية المستخدمة في النفاذ للمعطيات ضمن أنظمة قواعد المعطيات المعتمدة.

إن النموذج المعتمد في قواعد المعطيات العلائقية هو النموذج العلائقي الذي يُمثل البيانات ضمن جداول ترتبط ببعضها البعض بعلاقات مختلفة، في حين يتم تمثيل وثيقة XML في نموذج XPath كشجرة من العقد المرتبطة ببعضها البعض بوساطة علاقات الأب والابن وكل شجرة جزئية تضم عنصراً أو أكثر في وثيقة XML، ويمكن الاستفادة من تقنيات التجوال على الشجرة للبحث وإيجاد المعلومة المناسبة، بينما تنظم المعلومات في قواعد المعطيات العلائقية ضمن حقول وسجلات تتبع لجداول، وتَستخدم آليات المطابقة التقليدية للقيم الموجودة مع القيم المحددة في الاستعلامات وذلك لأجل تحديد المطلوب بعمليات البحث المختلفة.

بينما نستعيد حقولاً وصفوف وسجلات محددة في الاستعلام بقواعد المعطيات العلائقية، نستعيد وثيقة XML أو جزءاً منها أو مجموعة من الوثائق المرتبطة ببعضها البعض في قواعد المعطيات XML الأصيلة وذلك وفق صيغة شجرية سهلة القراءة والفهم.

فنموذج التخزين المعتمد في قواعد المعطيات العلائقية هو الجدول يقابله نموذج XML في أنظمة قواعد معطيات XML الأصيلة.

المقارنة حسب المعطيات ومدى مناسبتها لكلا النظامين:

أضحت قواعد المعطيات العلائقية كاسم بديل مرادف لمصطلح قاعدة البيانات، وأشهر طريقة مُعتمدة فيها في عالم الأعمال هي المعالجة بالمناقلات المباشرة باستخدام OLTP و المعالجة التحليلية للمعطيات باستخدام OLAP، وأحد أهم الأنظمة التي يكثر فيها استخدام OLTP هي: أنظمة إدارة المستخدمين customer management وأنظمة الصيرفة accounting systems . وأحد أهم الأمثلة على أنظمة تستخدم OLAP هي أنظمة إدارة المعطيات المؤرشفة تاريخياً historical archival systems وفي العادة أنظمة OLTP تحتوي معطيات قياسية (normalized data)، في حين أن أنظمة OLAP تحتوي معطيات زائدة غير قياسية تسمى denormalized data، وبما أن وحدة التخزين الأساسية في النموذج العلائقي هو الجدول فهو مناسب جيداً للعمل مع OLTP، أما المعطيات التي تنتج عن عملية دمج جداول (في OLAP) غالباً ينتج عنها معطيات زائدة ربما لا يمكن تقييدها ضمن بنية ثابتة وصلبة كالجدول الأمر الذي يتطلب البحث عن بُنى أخرى أكثر مرونة واستيعاب وهذا ما أعطى قواعد المعطيات NXDs أهميتها وجعلها مثالية لأجل التعامل مع المعطيات التي هي denormalized وذلك بتوفير مختلف إمكانات التخزين والاسترجاع والاستعلام(Nicola,2010).

وحيث أن denormalizing تُعرف كإجرائية تهدف إلى تحسين الأداء لقاعدة المعطيات عن طريق إضافة بيانات ومعطيات زائدة (redundant)، حيث أن متطلبات العمل في كثير من الأحيان تقتضي الحاجة إلى صياغة استعلامات لتوليد تقارير تستوجب وجود بيانات يتم جمعها من أكثر من مصدر واحد، وفي هذه الحالة يكون استخدام XML مثالياً للمعالجة والاسترجاع اللاحق والإجابة على مثل هذه الاستعلامات.

ولابد أن نشير أن المعطيات من نمط denormalized تأتي بتنسيقيات مختلفة وتخدم الكثير من الاحتياجات للمستخدمين، وتمكنهم من التخلص من القيود (constraints) والشواذ (anomiles) وأشياء أخرى مرتبطة بموضع normalization.

من النماذج المعروفة والمشهورة في تصنيف البيانات هي:

- نموذج المعطيات المرتبط بالبيانات (data–centric–model): وهو يتضمن مجموعة من البيانات المهيكلة بانتظام regular structure ولا تتضمن محتوى ناتج عن دمج مصادر بيانات متعددة وهي مناسبة للتمثيل والحفظ و المعالجة في نموذج قواعد المعطيات العلائقية.
- نموذج المعطيات المرتبط بالوثائق (document–centric–model): يتضمن مجموعة من البيانات التي تخضع لبنية أقل هيكلية وانتظام less regular structure وهي يمكن أن تحتوي على محتوى مدمج ناتج من مصادر متعددة، والمرونة التي تبديها أنظمة NXDs تجعلها مثالية لأجل استضافة وتخزين مثل هكذا بيانات. ونعرض الجدول التالي تلخيصاً للمقارنات:

	RDBs	NXDs
المعطيات المناسبة	معطيات قياسية (normalized)	معطيات غير قياسية (denormalized)
نموذج المعطيات	نموذج البيانات	نموذج الوثائق
بنية المعطيات المخزنة	بنية ثابتة ومهيكلة تفرض شروط مسبقة للتخزين	بنية مرنة و أقل هيكلية ولا تفرض شروطاً أو هيكليات مسبقة
مخطط قاعدة المعطيات الممثل	كيانات منفصلة وارتباطات متنوعة	كيانات متداخلة وبنية هرمية وطغيان لعلاقات من نمط واحد–كثير

من المتعارف عليه أن البيانات التي تخزن في أنظمة قواعد المعطيات العلائقية هي بيانات ثابتة ذات محتوى محدد وصلب (rigid content) مثل integer,text,string… وغيرها وبالتالي هي من طبيعة ليست مرنة، ولكن في كثير من الأحيان المستخدمون بحاجة إلى أن ينفذوا إلى المعلومات المقروءة أكثر وبالتالي إضافة بيانات تعريفية أكثر (denormalized form)للبيانات المؤرشفة يجعلها لاحقاً أكثر قابلية للقراءة والفهم، كما أن المستخدمين في كثير من الحالات يبحثون عن المرونة لكي يخزنوا بيانات يصعب تخزينها في قواعد المعطيات العلائقية وبما أن مستند XML

يسمح للمستخدم بأن يخزن ما يريد دون شروط و هيكليات و بُنى مفروضة مسبقاً، بالتالي أصبحت قواعد معطيات NXDs حلاً مثالياً تتجه له الكثير من الخيارات في مجال الأعمال.

تستخدم قواعد المعطيات XML الأصيلة في تخزين المعطيات من نوع recursive والتي يمكن أن نمثلها بوساطة بيان مباشر (directed Graph)، وهي بُنى ديناميكية مثل القوائم والأشجار وهي أمثلة هامة في علم الحاسوب لكونها بنية مرنة قابلة للتوسع(w3c xml,2005).

أمن قاعدة المعطيات في كلا النظامين:

إن الحقل الأساسي في المنافسة بين الشركات المصنعة لأنظمة إدارة قواعد المعطيات ليس فقط هو تأمين نظام قوي يضمن معالجة البيانات وعرض المطلوب منها، بل القوة تكمن في المحافظة على أمن قاعدة المعطيات ومنع غير المرخص لهم من العَبث بها، والمحافظة على صحة البيانات وخلق أدوار عمل للمستخدمين للنظام ومنحهم صلاحيات عمل كل منها يتناسب مع المهمة المناطة به.

بالنسبة لأنظمة إدارة قواعد المعطيات العلائقية فهي اليوم الأكثر موثوقية من ناحية الأمن كونها أثبتت جدارة بشكل عملي، في حين أن أنظمة قواعد معطيات NXDs أقل انتشاراً وحديثة العهد وقيد التطوير والتحديث ولم تتم تجربتها للعمل في بيئات تعتمد قواعد معطيات ضخمة الحجم و ذات حساسية كبيرة و مايزال العمل على تطويرها جارياً لكسب ثقة العملاء، سيما وأن البعض يأخذون على أنظمة قواعد المعطيات XML الأصيلة بعض التحفظات من ناحية أمن المعلومة وحتى نتمكن من عرضها يتطلب ذلك تحميلاً كاملاً لمحتوى قاعدة البيانات، فلذلك نحن لا نستطيع أن نتفحص جزء من قاعدة المعطيات ما لم نتفحصها ونحملها كاملةً ويتطلب هذا كلف زائدة من ناحية الوقت.

إن أنظمة NXDs لا تقدم أمن مناسب باستخدام الأدوار (role–base security) وتحديد من يستطيع القيام بالإضافة، أو الحذف أو التعديل في مجموعة محددة من البيانات، فهي تحدد صلاحيات أمن على أماكن تواجد البيانات(containers) ولكن عندما المستخدم يمتلك صلاحية النفاذ إلى حاوية معينة يستطيع عرض كل المعلومات في الحاوية، وعليه الحل يكون بضبط طرائق النفاذ للمستخدمين عن طريق إنشاء حاويات جزئية وتحديد سماحيات وصلاحيات وصول للمستخدمين كل منهم إلى الحاوية المناسبة ولكن هذا الحل يزيد الوقت المطلوب لإنجاز عمليات التحكم بالنفاذ(access control).

الفهرسة في كلا النظامين:

تُستخدم الفهارس في قواعد البيانات، لتسريع عملية الوصول إلى البيانات وتفادي المرور وسبر كامل البيانات المخزنة في القاعدة، وبالتالي تقليل عمليات القراءة من القرص الصلب والنفاذ له، وبالتالي توفير زمن وتحقيق الوصول إلى المعلومة بالسرعة الممكنة.

جميع قواعد البيانات الموجودة حالياً تستخدم الفهارس كبنية ملاصقة لا يمكن الاستغناء عنها إذ تسرع في عملية الحصول على المعلومة وهذا متطلب هام اليوم في عالم الأعمال.

الفهارس في قواعد المعطيات العلائقية تتنوع أحجامها وأشكالها، فإذا بنيت على واصفة واحدة فتسمى فهارس بسيطة، وإلا إذا بنيت على أكثر من واصفة فتسمى فهارس مركبة وإذا كانت الواصفة التي بني عليها الفهرس لا تسمح بتكرار القيم في الحقول التابعة لها عندها يسمى الفهرس unique وهو واجب التعديل بعد كل عملية إدخال، أما إذا كانت الواصفة التي بني عليها الفهرس تسمح بتكرار القيم فالفهرس عندها يعتبر ununique.

إن أنظمة إدارة قواعد المعطيات العلائقية تُنشىء بشكل أوتوماتيكي فهرساً خاصاً بالحقول التي تتمتع بنمط مفتاح رئيسي.

أما بالنسبة لأنظمة NXDs فاليوم الكثير من الشركات تعالج بيانات XML التي تأتي من مصادر عديدة وتُدار من قبل تطبيقات مستقلة، وتخزن في قاعدة بيانات منفصلة، ولسهولة الإدارة يتم الاعتماد على مجموعة من الفهارس مع التركيز على جودة الاستعلام وليس على جودة الإدخال(IBM,2011).

على أية حال التطبيقات المعتمدة على تخزين XML تتطلب إجراء عمليات إدخال وإخراج وتعديلات كثيرة وهي تؤثر على كامل بنية النظام إذا كانت بنية الفهرس المعتمد معقدة وتتطلب صيانة مكلفة.

البحث ضمن مستندات XML وباستخدام البنية الشجرية يقلل كثيراً من فضاء البحث بدلاً من خوارزميات الدمج المعتمدة في أنظمة قواعد المعطيات العلائقية المكلفة زمنياً، والمكلفة من حيث حجم الذاكرة المستخدمة.

في حين أن الإدخال أو الحذف في قاعدة المعطيات العلائقية يتطلب بأن كل فهرس لأجل الجدول يجب أن يعدل وإذا كان لدينا عدد كبير من الفهارس فهذا يؤثر على الإنجاز.

في قواعد المعطيات العلائقية الصف لديه حجم محدد وعدد معروف من الحقول وعندها المفتاح يمكن ببساطة أن يستخرج من الصف وبالتالي يتم بناء الفهرس بسرعة.

في حين أن XML مختلفة في طبيعتها، فغياب المعرفة المُسبقة ببنية وهيكلية المستند الذي يتم مسحه، حيث وثيقة XML يمكن أن تكون طويلة وتتضمن العديد من صفحات المعطيات.

مفاتيح الفهرس يجب أن تولد بالتزامن مع معالجة البيانات وبدقة أكثر هي يجب أن تولد بينما مستند XML يتم مسحه وتجزئته إلى صفحات بيانات(data pages)، وهذه الإجرائية تعمل بشكل تدفقي حيث مداخل الفهارس وصفحات البيانات تولد عند بداية مستند كبير بينما نحن لم نصل إلى نهايته وبالتالي البنية والهيكلية الحقيقية والحجم للمستند مازال غير معروف أي فقط جزئياً.

وعند تصميم الفهارس يجب أن نأخذ بعين الاعتبار جملة من المتطلبات الهامة:

- التصميم والكود يجب أن يكون بسيطاً.
- استهلاك الذاكرة يحتاج إلى الأخذ بعين الاعتبار الإدخالات المتزامنة الضخمة.
- الفهرس يجب أن ينجز بشكل جيد مراعاةً لطبيعة الإدخالات.

الإدخال أو الحذف من جدول ما ضمن قواعد المعطيات العلائقية يتطلب تعديلاً للفهرس (أو لجميع الفهارس) المرتبطة بالجدول المعني، وإذا كان لدينا عدد كبير من الفهارس فهذا يؤثر على الإنجاز، أما في قواعد المعطيات XML الأصيلة فإذا كان المسار المعرف بموجب معرف وثيقة XML (XML pattern)غير موجود ضمن المستند عندها لا يوجد مفتاح سوف يولد للفهرس، وبالتالي أي إدخال أو حذف من هذا المستند لن يؤثر على الفهرس المنشأ.

نستعرض فيما يلي بعض أنظمة قواعد معطيات XML الأصيلة المفتوحة المصدر:

نظام إدارة قواعد المعطيات exist:

هو نظام مفتوح المصدر لإدارة قواعد معطيات XML الأصيلة، وهو يتميز بزمن استجابة جيد وهو سهل الاستخدام (user friendly)، ومن السهولة تنصيبه والعمل عليه ومكتوب بلغة الجافا وهذا هو سبب استقلاليته عن بيئة التنجيز، وليس من الضروري أن تكون مبرمجاً بلغة جافا حتى تتمكن من استخدامه،فيمكن للمستخدم أن يستخدم واجهة exist البيانية لإنجاز أي مهمة يريدها، ويمكننا أيضاً إنجاز ذلك بوساطة التعليمات السطرية.

إن نظام exist يدعم الكثير من المعايير المعتمدة في web 2.0 وهو بيئة مناسبة لتطوير تطبيقات وخدمات الويب.

نظام إدارة قواعد المعطيات Sedna:

هو نظام يستخدم لإدارة المعطيات XML الأصيلة، صمم من قبل الأكاديمية الروسية للعلوم

لكي يحقق هدفين، أولاً: إنه نظام كامل الميزات وهذا يتطلب الدعم لكافة الخدمات المعروفة المقدمة من قبل قواعد المعطيات العلائقية، كإدارة الذواكر وتسهيلات تتعلق بالاستعلام والتحديث وغيرها من الخدمات.

ثانياً: توفير بيئة عمل للتعامل مع معطيات XML في زمن التنفيذ.

ويستخدم لغة XQuery وهو نظام كامل الميزات يدعم توسيع الذاكرة والتزامن والتحديث للبيانات ويستخدم آلية توفر في زمن المعالجة.

يدعم نظام sedna النفاذ لعدد كبير من المستخدمين، وهو يدعم بروتوكول الأقفال لكي يحل مشاكل التزامن ويضمن صحة المعطيات ويجعل التحديث والاستعلام فعالاً، وهنا لا نعمل إقفالاً على كامل وثيقة XML فالخوارزميات المعتمدة تحدد أصغر شجرة جزئية تتضمن الإقفال المطلوب وهذا لا يشمل العقد السابقة أي الأسلاف والآباء.

لقد ركز نظام sedna على توزيع البيانات بطريقة تدعم التحديث والاستعلام حيث أنجز التالي: مؤشرات مباشرة تستخدم لتمثيل العلاقات بين العقد الممثلة لوثيقة XML كعقد الآباء والأبناء وهو يعتمد على عنقدة العقد بناءً على موقعها في مخطط توصيفها حيث ما وجدت.

التطبيق المقترح لدراسة المقارنة:

بعد المقارنة النظرية بين قواعد المعطيات الأصيلة وقواعد المعطيات التقليدية العلائقية، لابد من أن نبين بشكل عملي ماهية الفروقات في الأداء والاستخدام عند تصميم تطبيق عمل حقيقي معتمد على كل من التقنيتين كل منهما على حِدى.وبالتالي المقترح كان بناء تطبيق ويب تفاعلي يقدم مجموعة من الخدمات الالكترونية للمغتربين السوريين وربطه بقاعدتي معطيات إحداهما مبنية باستخدام النموذج العلائقي والأخرى مبنية باستخدام نموذج قواعد المعطيات XML الأصيلة، وبالتالي ظهور الصعوبات في بناء كل من التطبيقين ومدى تحقيقهما للوظائف المطلوبة من النظام واستنتاج الفروقات في الأداء الفعلي لكل من التطبيقين المعتمدين، والسبب في اختيار تطبيق ويب هو أنه من الصعب الحديث عن XML دون تطبيقات وخدمات الويب الالكتروني حيث أن البيئة الناجعة التي تطورت بها تقنية XML هي الويب، حيث أنها تستخدم كوسيط مراسلة بين التطبيقات المختلفة المنجزة على منصات متباينة.

توصيف النظام المقترح:

بناء تطبيق ويب تفاعلي للتواصل مع المغتربين وتقديم الخدمات الإلكترونية المتاحة، حيث يتيح الموقع للمغترب التسجيل فيه بعد الحصول على البعض من بياناته و بريده الإلكتروني الخاص ومنحه اسم مستخدم وكلمة مرور خاصة به وبالتالي حصوله على حساب يتيح له النفاذ لطلب مجموعة من الخدمات المتنوعة التي يقدمها الموقع :

بالنسبة للمغتربين المتخلفين عن خدمة العلم يتيح الموقع مجموعة من الخدمات وهي:

1 – تقديم طلب زيارة قطر من أجل الحصول على الموافقة المطلوبة للزيارة (لثلاثة أشهر أو شهر) حيث ينفذ المغترب المتخلف عن أداء الخدمة والمشترك بالموقع لواجهة طلب الزيارة ويقوم بتعبئة مجموعة من البيانات وهي: الاسم والكنية واسم الأب والأم ومكان وتاريخ الولادة وشعبة التجنيد وبلد

الاغتراب، ومن ثم إرسال طلب الزيارة وانتظار الموافقة التي يُبلغ عنها لاحقاً حال ورودها من قبل إدارة الموقع على بريده الإلكتروني الخاص.

2 – النفاذ لواجهة تقديم طلب تمديد إقامة في القطر ولمدة شهر واحد في حال عدم كفاية مدة الإقامة الممنوحة له بموجب موافقة الدخول السابقة وحاجته للتمديد الإضافي لظروف خاصة وبالتالي تعبئة نفس البيانات المذكورة أعلاه بالإضافة إلى ذكره لموجبات التمديد.

3 – النفاذ لواجهة تقديم طلب تسهيل مغادرة القطر في حال انتهاء مدة الإقامة الممنوحة له بموجب الموافقة السابقة للزيارة أو مدة موافقة التمديد أو في حال دخوله الفجائي للقطر دون الحصول على موافقة مسبقة، وبالتالي تعبئة نفس البيانات المذكورة أعلاه، وإرسال طلب تسهيل لإدارة الموقع بغية الحصول على الموافقة المطلوبة، وعند حصوله على موافقة التمديد تُرسل له على بريده الالكتروني الخاص.

يستطيع المغترب في حال رغبته بالمشاركة في دعم مشروع قاعدة بيانات المغتربين، النفاذ لطلب واجهة إدخال بياناته الخاصة وملئ الاستمارات الالكترونية التي يوفرها الموقع ومن ثم إرسالها لتتم إضافتها لقاعدة البيانات المخزنة على المخدم المرتبط بالموقع وتتضمن تلك القاعدة معلومات عن البيانات الشخصية للمغترب بالإضافة لعمره وبلد اغترابه والنشاطات والهوايات التي يمارسها والمنظمات التي ينتمي إليها وغيرها.....

يستطيع المغترب تصفح قائمة النشاطات والفعاليات والملتقيات والمؤتمرات الاغترابية والندوات المقترحة المتعلقة بالمغتربين وتواريخ تلك الفعاليات وبرنامج عملها وأماكن انعقادها والأهداف المبتغاة منها.

يُمكن للمغترب النفاذ لواجهة إرسال المقترحات والتي يكتب فيها مقترحه ويقوم بإرسالها لإدارة الموقع ليتم دراسته والإفادة منه إن أمكن ذلك.

يستطيع المغترب طرح أي شكوى أو قضية لديه وإرفاقها بالوثائق الداعمة والمؤيدة وإرسالها لإدارة الموقع لتتم دراستها وإجابته عنها لاحقاً بعد المعالجة.

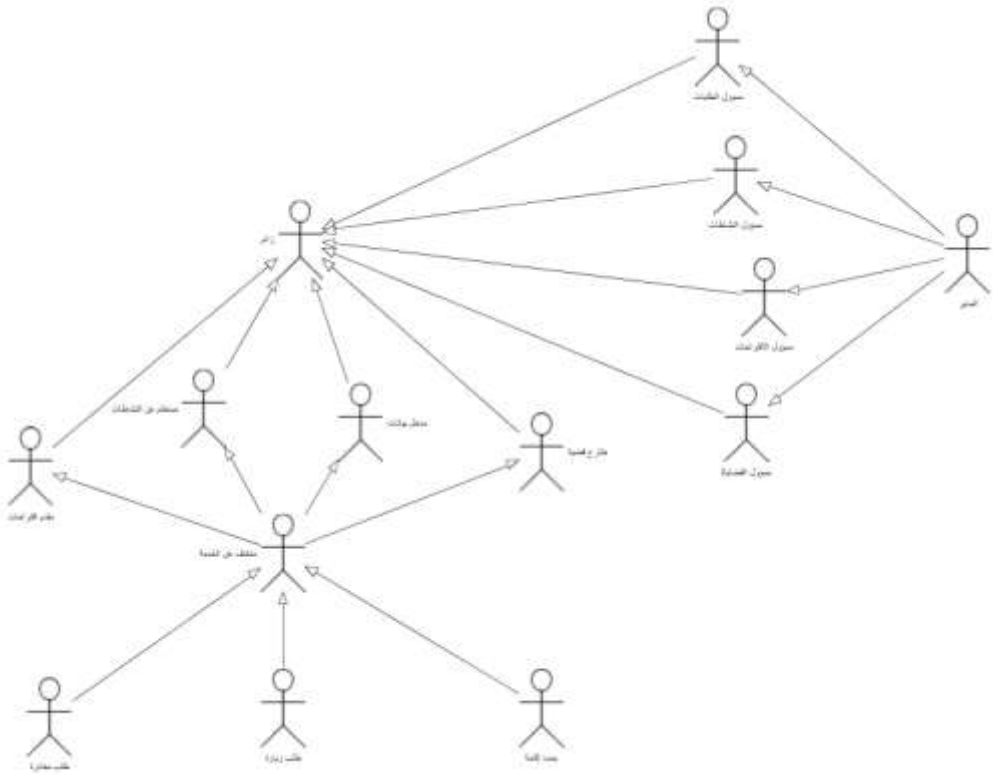
يوفر الموقع مجموعة من المستخدمين لإدارة الموقع وخدماته التي يقدمها وللتفاعل مع المغتربين وتخديمهم إلكترونياً والاستجابة لطلباتهم وهم كالتالي:

- مدير الموقع الذي يستطيع التحكم بكامل الموقع وإدارته والقيام بأي مهمة من مهام العاملين في إدارة الموقع.
- مدير النشاطات والذي يقوم بالإضافة والحذف والتعديل المطلوب للنشاطات والفعاليات والإعلانات المتعلقة بالمغتربين والتي ينشرها الموقع ويتيحها للمستخدمين.
- مدير المقترحات وهو يستقبل مقترحات المغتربين ويقوم بطباعتها ودراستها مع الإدارة ثم التعليق عليها وإعلام المغترب إلكترونياً في حال تطلب الأمر ذلك.
- مسؤول الطلبات وهو يقوم باستقبال طلبات الزيارة والتمديد وتسهيل المغادرة ويقوم بطباعتها وإرسالها للجهات المعنية ومن ثم إعلام المغتربين على بريدهم الإلكتروني في حال ورود الموافقة وذلك بإرسال رقمها وتاريخها ومدة صلاحيتها.
- مسؤول القضايا والذي يستقبل كافة الشكاوي والقضايا الواردة من المغتربين ويقوم بطباعتها ودراستها ومراسلة الجهات المعنية وإرسال الرد المناسب للمغترب على بريده الإلكتروني في حال انتهاء المعالجة.

لكل من الموظفين المسؤولين عن إدارة الموقع حساباتهم الخاصة عليه والتي ينفذون من خلالها كل بحسب صلاحياته للقيام بعمله الموكل إليه في الموقع.

فيما يلي نورد المخططات التفصيلية للمستخدمين وحالات الاستخدام

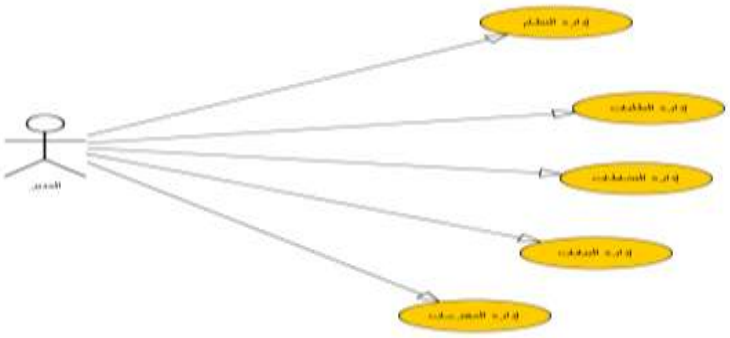
يمثل الشكل (1) مخطط المستخدمين للنظام:



الشكل (1)

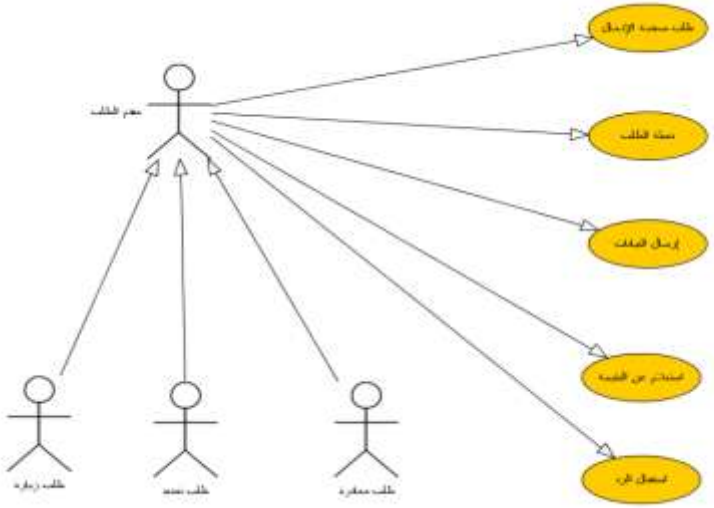
مخططات حالات الاستخدام في النظام:

مخطط حالات الاستخدام لمدير النظام مبينة في الشكل (2):



الشكل (2)

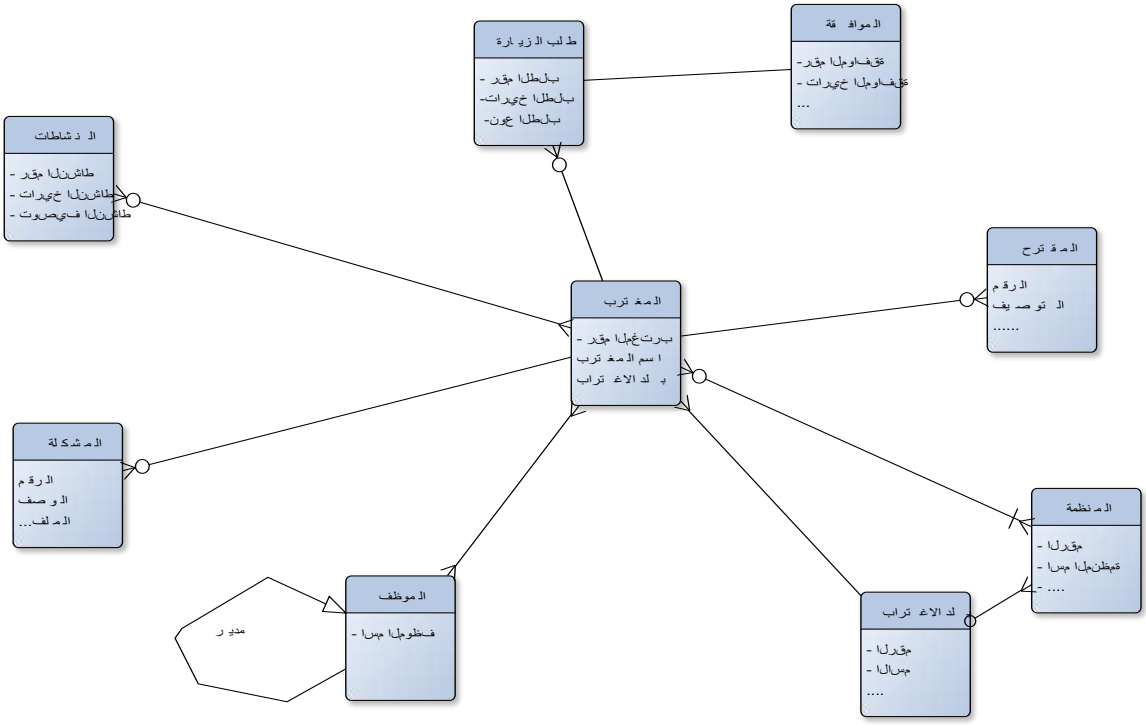
نبين بالشكل (3) مخطط حالات الاستخدام للمغترب المتخلف عن أداء الخدمة مقدم الطلبات:



الشكل (3)

المخططات التصميمية للنظام المقترح:

مخطط قاعدة المعطيات العلائقية مبين في الشكل (4):



الشكل

بيئة العمل المستخدمة في تنجيز التطبيق:

تم استخدام لغة php في بناء تطبيق الويب المقترح وتم اختيار قاعدة المعطيات العلائقية المعروفة mysql في بناء التطبيق الممثل لقواعد المعطيات التقليدية، في حين أنه اخترنا نظام sedna كممثل عن أنظمة قواعد معطيات XML الأصيلة.

النتائج والمناقشة

الاختبارات والنتائج:

بعد بناء تطبيق الويب المقترح باستخدام كل من قاعدتي البيانات:

1 – قاعدة معطيات XML الأصيلة.

2 – قاعدة معطيات mysql العلائقية.

هنا لابد لنا من إجراء الاختبارات المناسبة لتقييم أداء الموقع باستخدام كل من النظامين السابقين، وسوف نركز على اختبار تقييم الأداء للنظام في كلا الحالتين، وبالتالي إيجاد بنية معطيات صلبة تسمح لنا باختيار الأفضل، وهنا لابد لنا من الحديث عن مفهوم benchmarking بشيء من التفصيل.

مفهوم Benchmarking:

هي إجرائية تُستخدم لتقييم أداء النظام وفقاً لمعايير معينة، حيث يتم من خلالها التقاط مواصفات أساسية للتطبيق يتم البناء عليها قي عملية التقييس الجارية باستخدامها كمؤشرات أساسية لتحسين أداء النظام، وإن أي إجرائية لقياس الأداء يجب على الأقل أن تستجيب لعدة شروط أهمها:

- أن تكون قادرة على قياس أداء النظام في وقت الذروة.
- أن تكون قابلة للتوسعة.
- أن تكون قابلة للفهم.
- أن تكون قدر الإمكان مستقلة عن بيئة التنجيز، وبالتالي تقييمها لأداء أنظمة تقوم بعمل متشابه ولكنها تعمل على منصات عمل مختلفة.

يُوجد الكثير من الأنظمة التي تُستخدم لتقييم أداء أنظمة قواعد المعطيات ومنها:

- Apache JMeter: يُستخدم لتقييس أداء مخدم قاعدة البيانات العلائقي المعروف mysql فقط.
- TPoX: هو إجرائية تُستخدم لتقييس أداء نظام قاعدة معطيات XML مبني على سيناريو تطبيق مالي.

ولكن كل من الأداتين السابقتين غير مناسبتين لإجراء الاختبارات المطلوبة على التطبيق المقترح في البحث،كونهما تعملان على بيئة عمل محددة وملتزمتان كل منهما بتطبيق محدد وفق سيناريو مرسوم.

وعليه تم التركيز في البحث على اختيار أدوات تُستخدم لقياس أداء تطبيق الويب المنجز في كلتا الحالتين سواءً كان مخدم قاعدة البيانات هو علائقي أم منجز بوساطة XML، وبالتالي إعطاء مؤشرات واضحة يُعتمد عليها في المقارنة، وتحديد فيما إذا كان استخدام قواعد البيانات XML الأصيلة مناسب في مثل هكذا تطبيقات، خاصةً وأننا أثبتنا ومن خلال التطبيق العملي أنه يمكن لنا في حال بناء أنظمة معينة الاستعاضة عن أنظمة قواعد المعطيات التقليدية

المستخدمة فيها بنظام قاعدة معطيات XML أصيلة وتحقيق كافة الوظائف المطلوبة من النظام، ولكن السؤال الهام الذي ينبغي أن نجيب عليه:

هل استخدام قواعد المعطيات الأصيلة في تنجيز مثل هكذا تطبيقات ينعكس إيجاباً على أدائها، ويوفر مرونة وسرعة أكثر واستجابة أكبر ويُحسن من التفاعلية القائمة بين المستخدم والتطبيق مقارنةً مع أنظمة قواعد المعطيات العلائقية في حال استخدامها ضمن التطبيق؟.

نُلاحظ أن التطبيق مبني باستخدام هيكلية مخدم– زبون، ولدينا هنا جهة الزبون واحدة والاختلاف يَكمُن من طرف المخدم المستخدم، وهو مخدم ويب وبالأصل هو مخدم قاعدة بيانات أي أنه نظام قاعدة بيانات متكامل يأخذ شكله بحسب الإعدادات المختارة،وبالتالي إن اختبار أداء الموقع في كلتا الحالتين،هو اختبار فعلي لأداء مخدمات قواعد البيانات المعتمدة والاختلافات التي ستظهر مردها إلى نظام قاعدة البيانات المستخدم، وهو ما يمكن البناء عليه في إظهار الفروقات باستخدام كل من مخدمي قاعدتي البيانات، وبالتالي تقديم النصيحة المناسبة عن الأفضل منهما بالنسبة لمثل هكذا نوع من تطبيقات الويب.

وبعد بحث شامل عن أدوات اختبار مناسبة تحقق المطلوب أعلاه، توصلنا إلى إيجاد أداة مستقلة عن بيئة التنجيز المستخدمة في التطبيق وتعطينا مؤشرات بيانية بتعداد ستة مخططات تفصيلية بالإضافة إلى جداول واضحة بما يكفي لإجراء المقارنة واستنتاج المطلوب، وهي الأداة الذائعة الصيت webserver stress tool. وقد قمنا بدعم النتائج والخرج المبين باستخدام الأداة أعلاه التي لا تعتمد الإظهار البياني، حيث أنها أداة تعتمد الأوامر السطرية باستخدام واجهة الأوامر السطرية في نظام DOS، والخرج مرتب على نفس النافذة باستخدام تعليمة ab ومحدداتها، وقد حصلنا على نتائج متطابقة.

- ✓ الأداة webserver stress tool:

هي أداة تُستخدم لتقييم أداء مخدمات الويب(هُنا في تطبيقنا مخدم قاعدة البيانات) وتعطينا فكرة صحيحة عن أداء تطبيق الويب خلال فترات زمنية قد يتعرض فيها مخدم الويب لحمل زائد ناتج عن محاولة آلاف المستخدمين النفاذ بشكل متزامن مستخدمين نفس الصفحات، بالإضافة لقيامهم بعمليات إدخال أو استعلام كثيرة، وهذه الأداة تختبر أداء الموقع بغض النظر عن بيئة التنجيز المعتمدة في التطبيق أو بنية مخدم الويب أو قاعدة البيانات المعتمد مما جعلها أداة مثالية لإجراء الاختبارات المطلوبة على التطبيق المعتمد، كون نفس الأداة تقيِّم كلاً من المخدمين المستخدمين باستخدام نفس الخوارزميات والطرائق، مما يعطي المقارنة وثوقية أكثر مما لو كان التقييم لكل تطبيق بأداة مختلفة.

كما أن الأداة أعلاه توفر محاكاة رسومية خلال مدة الاختبار، حيث تعرض إظهاراً بيانياً في أسفل نافذة العرض، يُمثل فيه كل مستخدم بوساطة مستطيل مع لون محدد يوضح حالته، وتتغير الألوان مع تقدم الاختبار ويكون العرض سريعاً متناسباً مع سيناريو معين.

الاختبار المعتمد هو اختبار يتم فيه زيادة عدد المستخدمين مع تقدم الزمن وتنفيذهم للطلبات بشكل متزامن حيث يزداد عددهم من الواحد حتى العدد الأعظمي المحدد للمستخدمين.

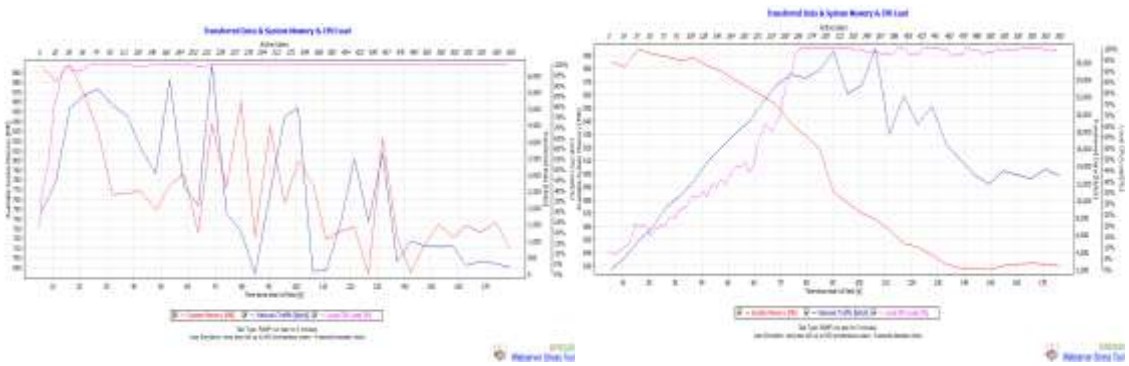
معظم المخططات تستخدم الزمن في بداية الاختبار كمحور أفقي، في حين يُمثل المحور الشاقولي المستخدمين الفعليين.

الصفحات التي سنقوم باختبارها ضمن الموقع هي عشر صفحات، تمثل الصفحات الرئيسية التي تُستخدم في التفاعل مع مخدم البيانات من ناحية إدخال البيانات، تعديلها والاستعلام عنها وهي التالي:

- http://localhost/XML db (صفحة الدخول الرئيسية)
- http://localhost/XML db/extendvisit.php
  (واجهة تسهيل زيارة)
- http://localhost/XML db/extendleave.php
  (واجهة تسهيل مغادرة)
- http://localhost/XML db/addinfo.php
  (واجهة الإضافة لقاعدة البيانات)
- http://localhost/XML db/viewinfo.php
  (واجهة الاستعلام عن البيانات المخزنة)
- http://localhost/XML db/requestsanswer.php
  (واجهة ردود الطلبات)
- http://localhost/XML db/checkanswer.php
  (واجهة الاستعلام عن الطلب)
- http://localhost/XML db/sendproblem.php
  (واجهة إرسال مقترح أو مشكلة)
- http://localhost/XML db/problemsanswer.php
  (واجهة ردود المشاكل)
- http://localhost/XML db/showactivities.php
  (واجهة استعراض النشاطات)

سوف نجري ثلاث أنواع من الاختبارات، حيث يتم تحديد العدد الأعظمي للمستخدمين بخمسمائة مستخدم ينفذون على التوازي إلى النظام خلال ثلاثة دقائق، ونركز في كل اختبار على ثلاثة أنواع من المخططات البيانية تعكس إلينا اختبارات العتاد، الانتظار، الاستجابة حيث يعكس الاختبار الأول ما أُستهلِك من العتاد، ويعكس الاختبار الثاني معدل انتظار المستخدمين للحصول على نتائج طلباتهم، في حين يُظهر الاختبار الثالث معدل الاستجابة للطلبات والأخطاء الحاصلة. نَبين فيما يلي نتائج الاختبارات الثلاثة على التتالي ونعلق على كل منها في نهاية الاختبار ونعرض للخلاصة، مع الإشارة إلى أن المخطط المعروض أولاً يمثل مخدم البيانات العلائقي mysql موضح في الشكل(5) المخطط اليميني ، في حين أن المخطط الثاني يمثل قاعدة المعطيات xml الأصيلة ممثلة بالمخطط اليساري في الشكل(5):

1 - اختبار العتاد: عدد المستخدمين 500 مستخدماً، مدة الاختبار 3 دقائق



الشكل(5)

-يزداد جهد المعالج في الموقع المبني باستخدام مخدم sedna بشكل أسرع اكبر من الموقع المبني باستخدام مخدم mysql والمقصود بجهد المعالج الزمن الذي يحتاجه المعالج لمعالجة الطلب بدءاً من لحظة استلامه ومواصفات المخدم مرتبطة بالأنظمة المشغلة سواء Sedna أو Mysql .

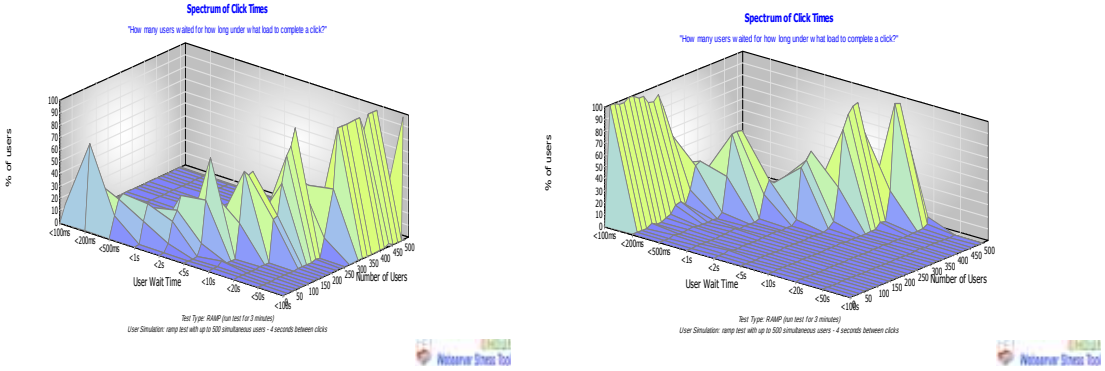
-يبلغ جهد المعالج وبسرعة في الموقع المبني باستخدام sedna نسبة 100% عندما يصل عدد المستخدمين 38 مستخدماً، ويستمر الجهد كذلك خلال الفترة المتبقية من الاختبار مع تزايد عدد المستخدمين حتى الوصول إلى 500 مستخدماً، في حين أن يبلغ نسبة 100% في الموقع المبني باستخدام مخدم mysql بعد وصول عدد المستخدمين إلى 256 مستخدماً ويستمر كذلك حتى وصول عدد المستخدمين إلى 500 مستخدم حسب ما هو موضح في المنحني الزهري في مخططات الشكل(5).

-تزداد مشغولية الشبكة في الموقع المبني باستخدام مخدم mysql بشكل أكبر من انشغالها في الموقع المبني باستخدام sedna وذلك موضح بالمنحني الأزرق في الشكل(5).

-يزداد استهلاك الذاكرة بشكل مضطرد مع الزمن ومع تزايد عدد المستخدمين وذلك في كل من الموقعين حسب المنحني الأحمر الموضح في الشكل(5).

-تبين من خلال الاختبارات السابقة أن بناء الموقع الإلكتروني المقترح باستخدام مخدم mysql، يمنحنا وفراً من ناحية الذاكرة المستخدمة ولا يُجهد المعالج كما هو الحال في استخدامنا لمخدم sedna في التنجيز، وبالتالي لتحسين أداء الموقع باستخدام sedna هذا يتطلب الحاجة لعتاديات أكبر قدرة من تلك المعتمدة في حال استخدامنا لمخدم mysql، من ناحية سرعة المعالج وحجم الذاكرة المستخدمة وبالتالي هذا يفرض علينا كلفة مادية أكبر.

2 - اختبار الانتظار: عدد المستخدمين 500 مستخدماً مدة الاختبار 3 دقائق موضح بالشكل(6):

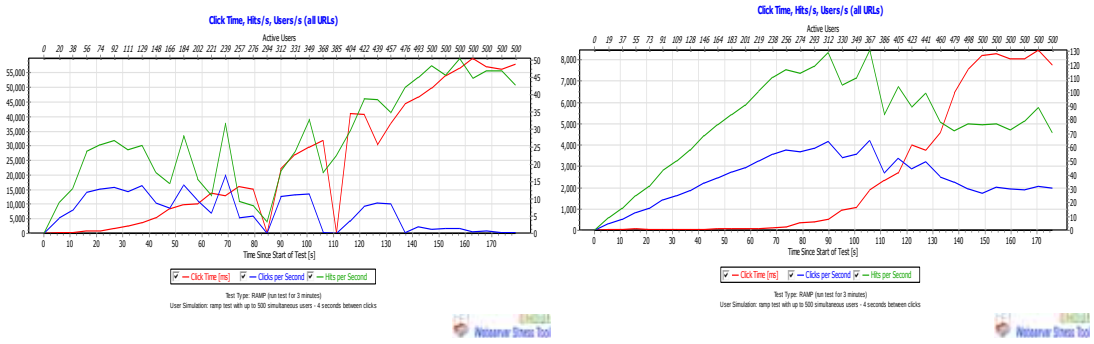


الشكل(6)

– بالنسبة للمخطط اليساري في الشكل (6) المتعلق بالموقع المبني باستخدام مخدم sedna: نلاحظ أن فترة الانتظار للمستخدمين من أجل الاستجابة لطلباتهم تتراوح بين 100 ms وأقل من 100 s كحد أقصى للانتظار، حيث أن 95% من المستخدمين ينتظرون مدة لا تتجاوز 100 s للاستجابة لطلباتهم.(يُلاحظ انزياح المخططات نحو اليمين نتيجة زيادة زمن الاستجابة)

– بالنسبة للمخطط الأيمن في الشكل(6) المرتبط بالموقع المبني باستخدام مخدم mysql: نلاحظ أن فترة الانتظار للمستخدمين من أجل الاستجابة لطلباتهم تتراوح بين 100 ms وأقل من 50 s كحد أقصى للانتظار، حيث أن 90% من المستخدمين ينتظرون مدة لا تتجاوز 50 s للاستجابة لطلباتهم.

تبين من خلال الاختبارات السابقة أن بناء الموقع الإلكتروني باستخدام مخدم sedna يزيد وبشكل ملحوظ الزمن الوسطي لانتظار المستخدمين لتلقي نتائج طلباتهم، وذلك مقارنة بالوفر من ناحية الزمن الذي يؤمنه استخدام مخدم mysql في بناء الموقع.



الشكل(7)

يمثل منحني click عدد الطلبات التي تم الاستجابة لها بنجاح في حين يمثل منحني hits الحالة المعاكسة.

– يُلاحظ من خلال تتبع المخططين زيادة نسبة الطلبات في كل من منحنيي (باللون الأخضر) hits و(اللون الأحمر) clicks في المخطط الأيمن في الشكل(7) المرتبط بالموقع المبني باستخدام مخدم mysql، وذلك مقارنةً مع مثيليهما في المخطط الأيسر في الشكل (7) المتعلق بالموقع المبني باستخدام مخدم sedan.

بالتالي نسبة الاستجابة للطلبات في الموقع المبني باستخدام mysql أعلى من نسبة الاستجابة لها في الموقع المبني باستخدام sedna وذلك بفارق محدود جزئياً، حيث يزداد بشكل مضطرد مع زيادة عدد المستخدمين.

✓ **الأداة ab:** هي تعليمة سطرية تُستخدم لتقييس أداء مخدمات الويب، بدايةً هي صُممت لكي تختبر أداء مخدم Apache، ولكنها عامة بشكل كافٍ لتختبر أي مخدم بيانات آخر، وهي مجانية وتأتي ضمن رخصة مخدم Apache.

ونتيجة لذلك تم استخدامها لتقييس أداء مخدمات قواعد البيانات المستخدمة في الموقع، وحصلت على نتائج مشابهة وداعمة للاستنتاجات أعلاه.

وكمثال عليها نعرض التعليمة التالية:

```
ab  –n 100 –c 20   http://www.mysite.sy/
```

وهنا ستتم عملية تقويم أداء النظام في حال تنفيذ مئة طلب وفق طريقة معينة من قبل عشرين مستخدماً على الرابط المعروض، حيث يتم عرض النتائج على نافذة الأوامر السطرية، لتتم عملية تقييمها.

سوف نقوم باستخدام الأداة التي توفرها تعليمة ab في تقييم أداء الموقع باستعمال كل من مخدمي قاعدتي البيانات، وسنعرض في التقرير لبعض الخرج الذي يوضح الأداء:

باستخدام مخدم قواعد البيانات XML الأصيلة ننفذ التعليمة التالية على واجهة الأوامر السطرية:

```
ab  –n 80  –c 50      http://localhost/XML db/viewinfo.php/
```

ونختار من الخرج التالي:

```
Server Hostname:        localhost
Document Path:          /XML db/viewinfo.php/
Concurrency Level:      50
Time taken for tests:   11.481620 seconds
Complete requests:      80
Time per request:       7176.012 [ms] (mean)
Connection Times (ms)
              min  mean[+/-sd] median   max
Waiting:      624 2992 1030.2   2964    8330
Percentage of the requests served within a certain time (ms)
  50%   2964
  66%   3104
  75%   3229
  80%   3369
  90%   3478
  95%   3541
  98%   8283
  99%   8330
 100%   8330 (longest request)
```

وباستخدام مخدم قاعدة البيانات العلائقية ننفذ نفس التعليمة السابقة على واجهة الأوامر السطرية:

```
ab  –n 80  –c 50      http://localhost/XML db/viewinfo.php/
```

ونعرض من الخرج التالي:

```
Server Hostname:        localhost
Document Path:          /XML db/viewinfo.php/
Concurrency Level:      50
Time taken for tests:   0.561601 seconds
Complete requests:      80
Time per request:       351.001 [ms] (mean)
Connection Times (ms)
              min  mean[+/-sd] median   max
Waiting:      109  304 134.0      343      468
Percentage of the requests served within a certain time (ms)
  50%    358
  66%    405
  75%    436
  80%    452
  90%    452
  95%    468
  98%    468
  99%    468
 100%    468 (longest request)
```

وبالمقارنة يتبين أن السرعة في الاستجابة للطلبات وزمن انتظار المستخدمين هو أبطأ باستخدام قواعد المعطيات XML الأصيلة.

و بمقارنة سريعة نجد أن زمن انتظار المستخدمين وسرعة الاستجابة للطلبات هي أفضل باستخدام قواعد المعطيات العلائقية في بناء تطبيق الويب، وهو ما يتماشى تماماً مع نتائج الخرج الظاهر في الاختبارات المبينة في المخططات البيانية أعلاه.

الاستنتاجات والتوصيات:

باستعراض نتائج الاختبارات السابقة، نجد أن استخدام مخدم قواعد المعطيات XML الأصيلة في بناء تطبيق الويب المقترح أو أية تطبيقات مشابهة يؤدي إلى بطئ في النظام بشكل عام وانتظار المستخدمين لفترات أطول للاستجابة إلى طلباتهم، كما أنه يتطلب استخدام ذاكرة ذات حجم أكبر، ومعالج أسرع مما هو مطلوب في حال استخدمنا قواعد البيانات التقليدية العلائقية في بناء تطبيق الويب، ومن هذه الدراسة أُثبتت قواعد المعطيات العلائقية فعاليتها ومناسبتها للاستخدام في تطبيقات الويب المشابهة للتطبيق المعروض، بالرغم من أن استخدام قواعد معطيات XML الأصيلة قد حقق كافة الوظائف المطلوبة من النظام(holstage,2008).

والأسباب تعود إلى أن طبيعة المعطيات المعتمدة مناسبة للتخزين ضمن بيئة المعطيات العلائقية أكثر من مناسبتها للحفظ ضمن قاعدة بيانات XML الأصيلة، كونها معطيات قياسية، كما أن آلية التخزين وإدارة الذاكرة الرئيسية

في أنظمة NXDs تَستوجب تحميلاً كاملاً لمحتوى قاعدة البيانات حتى نتمكن من عرضها والتعامل معها، الأمر الذي يستدعي في غالب الأحيان الاستعانة بالذاكرة الثانوية وفي هذا كلفة من ناحية الزمن إضافة إلى الزمن المُستغرق لمسح وثائق XML للوصول إلى المعلومة، في حين أن إدارة الذاكرة وتنظيم طلبات النفاذ هي أكثر فعالية في أنظمة RDBMs مما يوفر في حجم الذاكرة المستخدمة ويزيد في سرعة الاستجابة للطلبات، ولذا ينبغي البحث عن وسائل وآليات جديدة لمسألة إدارة الذاكرة وعمليات النفاذ في أنظمة NXDs تكون فعالية لتحقيق أداء أفضل. ونؤكد أن أنظمة قواعد المعطيات XML الأصيلة لم تُوجد لتكون بديلاً لأنظمة قواعد المعطيات التقليدية وهي ليست استبدالاً للنموذج الهرمي منها، ولم توجد لتكون حلاً لكل مشاكل قواعد المعطيات، وعندما يُستخدَمان في المواقع المناسبة ينجزان المطلوب منهما بالشكل الأمثل، فقواعد المعطيات العلائقية مناسبة للبيانات ذات البنية المهيكلة المعروفة والمحددة مسبقاً ولكن أنظمة NXDs مناسبة للمعطيات الأقل هيكلية والتي تتطلب مرونة في التمثيل واستيعاب معطيات زائدة مستقبلاً(Osman,2012). فلكل منهما بيئته وطبيعة المعطيات المناسبة لكي تخزن به، فالمقارنة بينهما على أن أي منهما هو إلغاء أو استبدال للآخر هي خاطئة، فكلاهما مُكمل للآخر، فالطائرة والقطار كل منهما يمكن أن يستخدم في النقل للركاب والشحن للبضائع وغيرها ولكن أياً منهما لم يوجد ليكون بديلاً للآخر، لأن لكل منهما بيئته ومستخدموه.

إن انتشار الاعتماد على ملفات XML، و وجود مُجمَّعات منها، فرضَ التوجه نحو بناء أنظمة قادرة على إدارتها وتنظيمها، فظهرت أنظمة NXDs، ولكن أنظمة RDBMs موجودة منذ أكثر من أربعة عقود تطورت خلالها وتحسن أدائها لتناسب بيئات عمل ضخمة، كما أن نموذج XML وتقنياته ما زالت قيد التطوير، لذا فأنظمة RDBMs أكثر سرعة و موثوقية وأفضل أداء.

التوجه حالياً نحو بناء أنظمة هجينة تجمع التقنيتين، لذا نرى أن معظم الإصدارات الحديثة من أنظمة RDBMs تدعم تخزين وثائق XML وتدعم لغة XQuery، والتحدي الحقيقي هو في تحديد أي المعطيات أكثر مناسبة للتخزين بصيغة XML، وأيُّها مُناسب ليُخزن ضمن جداول بشكل علائقي، وهل يُمكن إجراء الفرز للبيانات بشكل آلي؟ وهذا هو محل تساؤل يَدفع للبحث مستقبلاً.

المراجع

1. Nicola, M., "Lessons Learned from DB2 pureXML Applications A Practitioner's Perspective", 7th International Database Symposium, XSYM 2010.
2. Schmidt, A.; Waas, F.; Kersten, M. L.; Carey, M. J.; Manolescu, I., Busse, R.:XMark: A Benchmark for XML Data Management. In Proceedings of the 28thVLDB Conference, Hong Kong, 2002.
3. BöhmeBenchmarkingXMLDataManagementSystems.http://dbs.unileipzig.de/en/projekte/ XML /XML Benchmarking.html, June 2002.
4. W3C XML Query (XQuery), http://www.w3.org/XML /xQuery/ (2005).
5. R. Bourret, Going Native: Making the Case for XML Databases,
a. http://www.XML.com/pub/a/2005/03/30/native.html (2005)
6. Osman, R., &Knottenbelt, W. (2012)."Database system performance evaluation models: A survey".Performance Evaluation.69(10), pp. 471 –493.
7. IBM Corporation.(2011)Best Practices Managing XML Data. United States:IBM.
8. Kay, M. (2011).Saxon the XSLT and XQuery Processor. [online] Available at: http://saxon.sourceforge.net/[Accessed 19thSept 2014].