Journal of Education and Science
Vol. 29, No.2, 2020, pp.82-100
ISSN 1812-125X
http://edusj.mosuljournals.com

Reasons of the transformed toward NOSQL Databases and its data models

Shaymaa Ahmed Razoqi
University of Mosul
College of Education for Pure Science
Computer Science Dep.
shymaa.raazoqi@uomosul.edu.iq

DOI: 10.33899/edusj.2020.165303

Received **Accepted**
12/ 06 / 2019 **30 / 10 / 2019**

Abstract

The relational database model is the main focus of the database for the previous time. It provides robust data storage and structure that supports transaction characteristics and data retrieval with structured query language- SQL. The emergence of web technologies and clustering technology in large servers called for the need to store unstructured data and move away from the pattern of tables and static fields and not topic to transaction conditions, especially in distributed systems. Instead, it used CAP theory. New data models are emergence including: Document Data model, Key-Value model, Column-Family model, and Graph model, And new programming languages that deal with these models.

An asset inventory system is a model of systems that are not topic to a fixed structure. Each location in the organization has fairly different assets and assets in the same location have different specifications that are difficult to organize in the form of tables and columns. For this reason using of NOSQL CouchDB database system was adopted and the use of MAP-REDUCE technology to organize data display and Cloudant Query in Mango-view method to retrieve data from the inventory database.

Keywords : Relational Database, NOSQL Databases, NOSQL Data Model, CouchDB DBMS

أسباب التحول نحو قواعد بيانات NOSQL ونماذج البيانات التابعة لها

شيماء احمد رزوقي
جامعة الموصل
كلية التربية للعلوم الصرفة
قسم علوم الحاسوب
shymaa.raazoqi@uomosul.edu.iq

DOI: 10.33899/edusj.2020.165303

الاستلام 2019/06/12 القبول 2019/10/30

الخلاصة

يعتبر نموذج قواعد البيانات العلائقية هو المحور الأساس لقواعد البيانات طوال الفترة السابقة لكونه يوفر خزن وهيكلة بيانات قوية تدعم خصائص المعاملة وسهولة استرجاع البيانات بوجود لغة الاستعلام المهيكلة SQL. ظهور تقنيات الويب وتقنية العناقيد في الخادمات الكبيرة دعت الحاجة الى خزن البيانات غير المهيكلة NOSQL والابتعاد عن نمط الجداول والحقول الثابتة ولا تخضع لشروط المعاملة، خصوصا في النظم الموزعة، وبدلا من ذلك تستخدم نظرية الCAP وتبعا لذلك ظهر نماذج بيانات جديدة منها نموذج وثيقة البيانات ونموذج القيمة الرئيسة ونموذج عائلة الاعمدة ونموذج المخطط، وظهرت لغات برمجية جديدة تتعامل مع هذه النماذج.

نظام جرد الموجودات هو من نماذج الانظمة التي لا تخضع لهيكلة ثابتة فكل موقع في المؤسسة يضم موجودات مختلفة نوعا ما وكذلك الموجودات في نفس الموقع لها مواصفات مختلفة يصعب تنظيمها على شكل جداول من صفوف واعمدة ثابتة، لهذه الاسباب تم استخدام نظام قواعد بيانات NOSQL CouchDB واستخدام تقنية المقابلة والتخفيض لتنظيم عرض البيانات ولغة Cloudant Query بطريقة Mango-view لاسترجاع البيانات من قاعدة البيانات جرد الموجودات.

الكلمات المفتاحية : قواعد البيانات العلائقية، قواعد بينات NOSQL، نماذج بيانات NOSQL، نظام ادارة قواعد بيانات CouchDB

1. مقدمة

المتابع للتطورات التي حدثت في علوم وتقنيات الحاسوب يرى أن هناك تغيرات كبيرة ومتنوعة طرأت على مجالات اللغات البرمجية والهندسة المعمارية، ولكن الشيء الوحيد الذي لم تطرأ عليه تغيرات كبيرة هي قواعد البيانات العلائقية Relational Databases(RDB)، وفيها البيانات المنظمة تدوم وتكون ذات خزن مستقر، والذي يسهل الوصول إليها ومعالجتها بشكل جيد جدا من قبل العديد من التطبيقات البرمجية المتنوعة.

دفع تطور البرمجيات الحديثة وظهور كميات واسعة من البيانات الكبيرة لتغيرات أساسية في بناء القطع المادية، من ذلك مثلا بناء خادمات كبيرة تستخدم تقنية العناقيد(Clusters) التي تتضمن بيانات غير منظمة وبكميات كبيرة جداً ومتنوعة، وتخضع لمعالجات وتحليلات مستمرة. الأمر الذي اوجب بالابتعاد عن قواعد البيانات العلائقية وشروط المعاملات(Transactions) ولغة الاستعلام المهيكلة (Structure Query language -SQL)، ولكن هذا لا يعني أن قواعد البيانات العلائقية سوف تتوقف. فمن المتوقع أن تبقى مستعملة لعقود قادمة، وستبقى هي المسيطر الرئيس لقواعد البيانات المنظمة على الرغم من ظهور تقنيات قواعد بيانات NOSQL واستخدام نظرية الCAP وتنوع نماذج البيانات غير المهيكلة، وتبعا لهذا ظهرت لغات برمجة جديدة وبرمجيات لإدارة قواعد البيانات مثل CouchDB و MongoDB و Cassandra.

2. الأعمال السابقة

- اقترح Vera [1] وآخرون طريقة لنمذجة بيانات NOSQL والغرض منها هو إعطاء رؤيا واضحة للبيانات بالإضافة إلى التأكيد على أهمية أسلوب نمذجة البيانات وحالة الدراسة التي تم اعتمادها كانت استخدام بيانات جينية.
- هدفت الدراسة التي قام بها Salehnia [2] لإيجاد طريقة لتحويل قواعد البيانات العلائقية إلى بيئة البيانات الكبيرة Big-Data . ودراسة أسباب عدم قدرة بعض أنواع أنظمة قواعد البيانات القوية مثل أوراكل في الوصول إلى تحقيق متطلبات البيانات الكبيرة بالإضافة إلى عمل مقارنة بين البيانات المهيكلة وشبه المهيكلة وغير المهيكلة .
- قدم Chandra [3] دراسة تحليلية للخصائص الأساسية لقواعد بيانات NOSQL مع التركيز على برمجياتها الواسعة النطاق واستخدام تقنيات مختلفة لتحقيق اتساق البيانات، وأيضا تقنيات توفر البيانات مقارنة مع خصائص ACID لقواعد البيانات العلائقية.
- قدم Meijer [4] واخرون مقارنة بين ثلاثة أنواع من قواعد البيانات وطريقة أدائهم النسبي مع الأخذ بنظر الاعتبار لطريقة خزن البيانات، وكذلك مقارنة تنفيذ عمل هذه القواعد على خادم فيزيائي وعملهم على آلة افتراضية بالنسبة لعمليات القراءة المفردة والكتابة المفردة وكذلك عمليات القراءة والكتابة المتعددة.
- كشف Varga [5] وآخرون في الورقة التي قدموها طريقة عرض تصوري تقليدي لقواعد بيانات الوثيقة باستخدام تحليل المفاهيم الرسمي ومدى قدرته على إعطاء صورة مرئية كاملة لتحليل البيانات واسعة النطاق.

3. قواعد البيانات العلائقية

قواعد البيانات العلائقية أصبحت جزءاً ضمنياً في أغلب التطبيقات البرمجية ولا يمكن الاستغناء عنه حسب تصور أغلب المبرمجين، وذلك لسهولة استعمالها وتعاملها مع البيانات بأسلوب منظم وخزن ثابت فهي توفر استرجاع بيانات ثابت، وهي في نفس الوقت تسمح للعديد من المستخدمين بالوصول إلى نفس البيانات بشكل متزامن مع إمكانية تعديل تلك البيانات[2].

إن أغلب التطبيقات الحديثة لقواعد البيانات العلائقية تكتب من قبل فرق مختلفة من المبرمجين، وتستخدم هذه التطبيقات نفس البيانات،لذا فالتغير الذي يحدثه أي تطبيق يجب أن يكون مرئياً لبقية التطبيقات.والطريقة

المستخدمة لذلك هي تكامل قاعدة البيانات المشتركة وهي أكثر من تطبيق يخزن البيانات نفسها، والتي تعامل بها تعدد المستخدمين في تطبيق واحد[6].

نجحت قواعد البيانات العلائقية بشكل كبير كونها تستخدم نماذج قياسية رئيسة مع اختلافات بسيطة جداً، ولكن الميكانيكية الأساسية هي نفسها وكذلك طرق التعامل مع المناقلات. ومنذ بداية قواعد البيانات العلائقية كانت هناك مشكلة الاختلاف مع تركيب الذاكرة، ففي النموذج العلائقي كل شيء ينظم في جداول (القيم فيها بسيطة غير مركبة) وعلاقات، وتعتمد على الجبر العلائقي، وهذا يعطي مرونة وبساطة في المعالجة ولكنه ينتج محددات وتعقيدات بشكل خاص عند التعامل مع التركيب الهيكلي للذاكرة حيث من الممكن يأخذ هيكل أغنى[7].

العامل الرئيس لبقاء قواعد البيانات العلائقية عبر الزمن هو قوانين لغة الاستعلام المهيكلة SQL والآلية تكامل قاعدة البيانات مع التطبيقات المختلفة. وقاعدة البيانات التي يتم بناؤها لأداء مجموعة أغراض تخزن كميات كبيرة من البيانات في مكان واحد مشترك تصله مجموعة من التطبيقات البرمجية التي يتم تصميمها للتعامل مع قاعدة البيانات هذه. طريقة قاعدة البيانات المشتركة مفيدة من حيث تحسين الاتصال وتوفير موقع الخزن، لكنها تأتي مع سلبيات منها عدم قدرة احد التطبيقات لتغيير تنسيق البيانات –إذا احتاج لذلك– إلا بما يتناسب مع بقية التطبيقات التي تستعمل قاعدة البيانات ولا يتعارض معها.

مع بداية الألفية الثانية بدأ استخدام تطبيقات الويب وبروتوكولات HTTP لنقل البيانات وبدأت هذه التطبيقات الجديدة تمثل تحدياً جديداً لاستعمال SQL مع قواعد البيانات المشتركة، ونتج عن هذا الانتقال مرونة أكثر لهياكل البيانات التي بدأت تبتعد عن قواعد SQL وتتعامل مع السجلات المتداخلة أو القوائم أو وثائق XML.[8][2]

4. ظهور تقنية العناقيد

مع ظهور الألفية الجديدة بدأت الشبكات الحاسوبية بالانتشار وتوسعت خدمات الويب وتنوعت وظهرت برمجيات التواصل الاجتماعي، التي أدت إلى ظهور مجموعات كبيرة من البيانات، ومع النمو في البيانات جاء أيضا تزايد ملحوظ في عدد المستخدمين والزوار لمواقع الويب، ونتيجة لزيادة حجم البيانات والطلبات زادت الحاجة إلى استعمال مصادر الحاسبات بشكل أكبر، ولمعالجة هذا النوع من الزيادة ظهر خياران:

- الاختيار الأول هو استخدام آلات حاسبة كبيرة ومعالجات أكثر قوة وأقراص خزن وذاكرة أوسع، ولكن هذا الحل يحتاج كلفة أكبر.
- والاختيار الثاني هو استخدام عناقيد من آلات حاسبة صغيرة قليلة الكلفة، وتكون أكثر مرونة، وتكون حالات الفشل فردية، إذ ان بقية الأجهزة في العنقود يمكن أن تصمم لكي تستمر بالعمل حتى مع فشل أحد الأجهزة، وهذا يعطي مستوى عالياً من الثقة، حيث بدأ الاتجاه العناقيد وهذا أدى لظهور مشكلة جديدة وهي ان قواعد البيانات العلائقية ليست متهيئة للعمل في بيئة العناقيد [9].

4–1 ظهور قواعد بياناتNOSQL

ظهرت هذه التسمية أول مرة في التسعينات كاسم لقواعد بيانات مفتوحة المصدر تحت قيادة Carlo-Strozzi [9]. قواعد البيانات هذه تخزن جداولها كملفات اسكيASCII، وكل سجل يمثل سطراً من البيانات بشكل

حقول تفصلها فراغات. وجاء الاسم من حقيقة ان قاعدة البيانات لا تستخدم لغة الاستعلام المهيكلةSQL. وبدلاً من ذلك عولجت قاعدة البيانات من خلال المستندات المغلقة والمدمجة مع خطوط يونكس العادية[2].

مع بداية الألفية الجديدة الاسم NOSQL بدأ يعبر عن شيء مختلف نوعاً ما عن السابق، إذ ان هذه التقنية الجديدة ظهرت على يد Johan Oskarsson . والتعبير انتشر بشكل كبير لكن لم يكن هناك تعريف واضح، فهو ناتج عن البحث عن قواعد بيانات غير علائقية موزعة مفتوحة المصدر، لذا فكل ما هو معروف عنها هو مجموعة خصائص مشتركة لقواعد بيانات التي تميل إلى اسمNOSQL [3]، واهم تلك الخصائص:

- ان قواعد بيانات NOSQL لا تستخدم لغة الاستعلام المهيكلةSQL. وهذه تعتبر النقطة الأكثر وضوحاً على الرغم من أن البعض منها تستخدم لغة استعلام خاصة بها مشابهة للغة الاستعلام المهيكلة لكنها أسهل في التعلم مثل لغة استعلام كساندرا CQLولغة Cloudant Query
- والخاصية الثانية لقواعد البيانات هذه أنها قواعد مفتوحة المصدر.
- خاصية أخرى تشترك بها معظم قواعد NOSQL تقود إلى الحاجة للتنفيذ على العناقيد، وهذا له تأثير على نماذج البيانات بالإضافة إلى تناسق البيانات.

في حين تستخدم قواعد البيانات العلائقية خصائص المعاملة ACID (الوحدة Atomicity، والاتساق Consistency، والعزل Isolation، والمتانة Durability) لمعالجة تناسق البيانات عبر تكامل قواعد البيانات، وهذا يتعارض مع بيئة العناقيد، فإن قواعد بيانات NOSQL لا تعطي أهمية لخصائص المعاملة وفي بعض الحالات تتجاهلها تماماً خصوصاً في النظم الموزعة، والتي يكون فيها غالباً من غير الممكن ضمان الالتزام بهذه الخصائص[10]. قواعد بياناتNOSQL تقدم مدى من الاختيارات للتزامن والتناسق والتوزيع للبيانات. وتستخدم نظرية الCAP(الاتساقConsistency، والتوفرAvailability، وتحمل التقسيم Partition tolerance)، والتي قدمت سنة 2000 من قبل Eric Brewer [12]، وأُثبتت فيما بعد من قبل Seth Gilbert و Nancy Lynch سنة 2002. وتذكر النظرية أن اثنين فقط من الخصائص الثلاث تكون متوفرة في نقطة ما في زمن معين ولكن ليس الثلاثة معاً[11]. وعملية الاختيار يمكن أن تتم كالتالي:

- التوفر يمكن أن يتم تجاهله ويفضل عليه كل من الاتساق وتحمل التقسيم.
- الاتساق مع التوفر يكون مفضلاً إذا كان النظام فيه تقسيماً قليلاً أو غير مقسم.
- الاتساق يمكن تجاهله ، لكن النظام يجب أن يكون متوفراً دائماً، ويستمر بالعمل عندما تكون اجزاء منه مقسمة.

من الأفضل التفكير في قواعد بيانات NOSQL كتقنية جديدة لأنها لن تلغي وجود قواعد البيانات العلائقية ولكنها تبقى المساهم الأكبر لأنواع قواعد البيانات المستخدمة في الوقت الحالي فهي توفر دعماً وأمناً واستقراراً لأغلب المشاريع. ومع وجود البيانات الكبيرة خلقت لدى المستخدمين الحاجة لخزن البيانات بطريقة أفضل، واستخدام قواعد NOSQLسهل الوصول إلى البيانات، وساعد في زيادة معدل إنتاج المعرفة. قواعد بيانات NOSQL تعمل بدون أي هيكلية، وتسمح بإضافة حقول جديدة إلى الجداول بدون تغير في الهيكل الأصلي أو التطبيقات البرمجية وهذا، مفيد عند التعامل مع بيانات ليس لها شكل منظم رسمي وحقول حسب الطلب[9].

4-2 نماذج البيانات في قواعد بياناتNOSQL

تسمى طريقة تصميم وتعريف ومعالجة البيانات بصورة ما بنموذج البيانات وكيفية وصف البيانات، وكيف سيتم التفاعل مع البيانات في قاعدة البيانات، وهذا يتم بمعزل عن نماذج الخزن التي تصف كيف يتم خزن قاعدة البيانات، وكيف تعالج البيانات داخلياً. نموذج البيانات الأكثر انتشارا لقواعد البيانات هو النموذج العلائقي، والذي تنتظم فيه البيانات بشكل جداول وعلاقات بينها، وكل جدول يتكون من صفوف وأعمدة تتقاطع مكونة خلايا تحمل كل منها قيمة وحيدة .

النموذج العلائقي يأخذ المعلومات المراد تخزينها ويقسمها إلى جداول مكونة من صفوف والصفوف هي هياكل بيانات محددة تأخذ بيانات بسيطة، ولا يمكن مراكمة صف فوق آخر للحصول على سجلات متراكمة أو متداخلة ولا يتم وضع مجموعة قيم ضمن أخرى. وكل العمليات تخضع لشروط العمل على الجداول[9].

المجموعات الموجهة كنموذج بيانات جديد تأخذ طريقة مختلفة، فهي تعمل على بيانات تكون بشكل هياكل أكثر تعقيداً من الجداول تسمى وحدة المجموعة، والتي تعرف على أنها وحدة بيانات تستخدم للمعالجة والتناسق والتكامل، وتتكون من عناصر مترابطة تعامل كوحدة مستقلة واستخدام هذا النوع من وحدات البيانات يتناسب مع أنواع قواعد NOSQL.المجموعات الموجهة هي نماذج بشكل مجموعات موجهة قوية وكل منها له رمز تعريفي يستخدم لاسترجاع البيانات، وهنا يكون من الممكن خزن أي أنواع البيانات في المجموعة الواحدة، قاعدة البيانات قد تعرض حجماً عاماً محدداً للمجموعة الواحدة[13].

في قواعد بيانات NOSQL يُرى بشكل واضح الابتعاد عن النموذج العلائقي والانتقال إلى نماذج مختلفة أخرى يمكن وصفها في أربع نماذج واسعة للاستخدام في هذه البيئة، والذي يربط عناصر البيانات المختلفة عند التخطيط للنموذج العلائقي هي المفاتيح في حين لا يوجد شيء مماثل لتمييز العلاقات التي تربط المجموعات مع بعضها في قواعد بيانات المجموعات الموجهة. إن السبب الرئيس لاستخدام وحدة المجموعة الموجهة كهياكل بيانات أنها تساعد بشكل جيد التعامل مع العناقيد ولكن يتم السعي دائما إلى تقليل عدد العقد للاستعلام الواحد عندما تجمع البيانات من العناقيد. لذا هناك حاجة دائما إلى دراسة أنواع الاستعلامات ووضع البيانات التي سوف تطلب مع بعضها في نفس العقدة أو المجموعة[10].

استخدام وحدة المجموعة لها نتائج مهمة للمعاملات. وكما هو معروف قواعد البيانات العلائقية تسمح بمعالجة مجموعة من السجلات (صفوف) من أي جدول في معاملة واحدة، وهذا يخضع لشروط المناقلة ولكن قواعد بيانات NOSQLلا تدعم هذه الشروط، وهذا معناه التضحية بالتناسق كشرط لتنظيم البيانات وبذلك فهي تكون أسرع في معالجة المجموعات المتعددة، وبدلاً من ذلك فهي تدعم المعالجة الذرية للمجموعة الواحدة.[1]

4-2-1 نموذج وثيقة البيانات Document Data Model

يتم تخزين البيانات في هذا النوع على شكل ملفات لا تعتمد على تصميم ثابت، وكل ملف يمكن ان يحتوي على تصميم مختلف وغير منظم. قاعدة بيانات الوثيقة الموجهة كما يدل الاسم نموذج خزن البيانات المطلوبة على

شكل وثيقة، وهي تفرض قيوداً على كل مايمكن خزنه بالمقابل هناك مرونة أكثر في الوصول للبيانات. ضمن نموذج بيانات الوثيقة الموجهة في قواعد بيانات NOSQL تكون البيانات غير منظمة ولا تتبع صيغة معيارية ثابتة أي تكون شبه مهيكلة[14]. مثلا إذا كان لدينا مكتبة كتب تخزن معلومات الكتب بشكل وثائق تسمى Book وضمن هذه الوثيقة يتم خزن اسم الكتاب وأسماء المؤلفين وتاريخ النشر واسم دار النشر والطبعة وغيرها من معلومات الكتاب، كذلك يمكن خزن مؤشرات لوثائق أخرى، وهذه المؤشرات تكون بمكانة المفاتيح الأجنبية لقواعد البيانات العلائقية، يوضح (الشكل1) نموذج بيانات الوثيقة الموجهة.كل من برنامج MongoDBو برنامج CouchDB هي أمثلة لإنتاج قواعد بيانات الوثيقة الموجهة[15][16]. مع قواعد بيانات الوثيقة يتوقع في الغالب تقديم بعض النماذج للاستفسار بالاعتماد على الهيكل الداخلي للوثيقة[17].

Book Title: Business Intelligence and Analytics: Systems for Decision Support

By Ramesh Sharda, Dursun Delen, Efraim Turban

Publication Date: 2015

Edition: 10th

Publisher: Pearson

Publisher Location: Upper Saddle River, NJ:.

ISBN-13: 978-0-13-305090-5

شكل(1) نموذج بيانات الوثيقة

4-2-2 نموذج القيمة الرئيسةKey-Value Model

تخزن البيانات في نموذج القيمة الرئيسية بشكل أزواج متناظرة من الأعمدة (عمودين متناظرين) ، يضم العمود الأول قيمة المفتاح، ويضم العمود الثاني القيمة الفعلية للبيانات. القيمة الفعلية للبيانات يمكن أن تكون نصاً بسيط، أو أنواع بيانات مركبة معقدة. في طريقة القيمة الرئيسة للخزن يمكن فقط الوصول إلى المجموعة بالبحث عن مفتاحها.مع قواعد بيانات القيمة الرئيسة يتوقع في الغالب البحث في التجمعات باستخدام المفتاح. وتكمن قوة قواعد بيانات القيمة الرئيسة في سرعة القراءة، إذ يتم استخدامها في عملية التخزين المؤقت للبيانات[16][10].

من الأمثلة لقواعد بيانات NOSQLالتي تستخدم نموذج بيانات القيمة الرئيسة هي Project Voldemort, Cache and Dynamo. في المثال السابق لتمثيل بيانات الكتاب باستخدام نموذج بيانات القيمة الرئيسة يمكن توضيحها بـ(الشكل2) [17].

Key	Value
Book Title	Business Intelligence and Analytics: Systems for De-cision Support
Author (set)	Ramesh Sharda
	Dursun Delen
	Efraim Turban
Publication Date	2015
Edition	10^{th}
Publisher	Pearson
…	…

شكل(2) نموذج بيانات القيمة الرئيسية

4-2-3 مخازن عائلة العمودColumn-Family Stores

تستخدم جوجل قواعد بيانات NOSQLنوع يسمى جداول جوجل الكبيرة واسمه استخلص من الهياكل المجدولة، والتي تكون بشكل أعمدة متناثرة وليس لها اسكيما، ولا يمكن التفكير في هذا التركيب كجدول وبالأحرى هو خريطة بمستويين. أي جداول الأعمدة فيها من نوع جدول، ومن اشهر الامثلة على هذه النظم قواعد البيانات التالية مثل كاسندرا[18][16].معظم قواعد البيانات العلائقية يكون فيها الخزن على شكل سطر كوحدة خزن، والذي يساعد في تحسين الأداء. في معظم الاستعلامات عن البيانات يتم جلب البيانات من أكثر من جدول (أي عدة اعمدة من عدة جداول في المرة الواحدة)، لهذا الغرض من الأفضل خزن مجموعة أعمدة لكل السطور في وحدة خزن واحدة. لذا تسمى هذه الطريقة مخازن الأعمدة وطريقة الجداول الكبيرة جاءت من هذه الفكرة، وهي خزن مجموعة أعمدة مع بعضها كعائلة واحدة.

نموذج بيانات عائلة العمود له صيغة خزن بيانات مشابهة إلى حد ما لقواعد البيانات العلائقية ، بالرغم من ان قواعد بيانات العلائقية تميل إلى التعامل مع أنواع بيانات بسيطة وذات هيكل منظم ومعروف مسبقا، قواعد بيانات NOSQL عائلة العمود توفر مرونة أكثر بكثير، فهي تدعم أنواع بيانات مركبة ومعقدة، وغير مهيكلة[15]. يبين (الشكل3) نموذج عائلة العمود، وكل من (المؤلف وتاريخ النشر والطبعة والناشر) قد تم تضمينها في نوع بيانات مركب يدعى تفاصيل الكتاب(book details).يعتبر كاسندرا ولغة (Cassandra Query Language)مثالاً لقواعد بيانات NOSQLالتي تستخدم نموذج بيانات عائلة عمود[17][19].

Business Intelligence and Analytics: Systems for Decision Support	
Book Details (includes authors, year, edition, publisher, etc.)	
Ramesh Sharda	
Dursun Delen	
Efraim Turban	
2015	
10^{th}	
Pearson	

الشكل(3) نموذج عائلة العمود

كما في مخازن القيمة الرئيسة المفتاح الأول يصف معرف السطر. والفرق مع عائلة الأعمدة في ان مجموعة الأسطر نفسها تصف الخريطة لتفاصيل القيم، أما قيم المستوى الثاني فهي تشير إلى الأعمدة . طريقة قواعد بيانات عائلة الأعمدة تنظم الأعمدة في عوائل كل عمود هو جزء من عائلة معينة. ويكون وحدة وصول، لذلك فالقيم المخزونة في الأعمدة المتشابهة في نفس العائلة سوف يتم الوصول لها مع بعضها.

4-2-4 نموذج المخطط البيانيGraphModel

يتم خزن البيانات في هذا النموذج على شكل عقد وعلاقات تربط العقد مع بعضها البعض. نموذج بيانات المخطط البياني يدعم خزن البيانات التي لها عدد غير معروف من الارتباطات في شبكة الانترنت. مثل بيانات الخرائط و خطوط النقل، وعلاقات الأوساط الاجتماعية[15][16] على سبيل المثال عمل رسم بياني لإيجاد المسافة الأقصر بين المدن يكون صعبا جدا باستخدام نموذج قاعدة البيانات العلائقية التقليدية، في حين قواعد بيانات NOSQLالمخطط البياني يمكن أن تسهل هذا النوع من المعالجة. من أمثلة قواعد البيانات التي تستخدم نموذج المخطط البياني هي Neo4j ونظام Virtuoso [10].

5.قواعد بيانات بدون مخطط Schema less Databases

في قواعد البيانات العلائقية يستخدم مخطط Schema وتستخدم لعدد من التطبيقات باستخدام نفس قاعدة البيانات بدون تغير في التركيب وبسهولة، ولان هناك معرفة مسبقة بتنظيم الجداول والأعمدة والعلاقات، أما في قواعد بيانات NOSQLلا يوجد مخطط، وهنا سوف يتم الاعتماد على التطبيقات في الوصول السليم للبيانات، أي يتم عمل زحفاً للمخطط من قاعدة بيانات نحو التطبيقات البرمجية وطريقة الوصول ، أي الاعتماد على التطبيقات في الحصول على خزن جيد واسترجاع جيد، فهنا لا توجد طريقة محددة مسبقة تعتمد على التطبيق، وتظهر المشكلة في حالة وجود أكثر من تطبيق يريد الوصول إلى نفس البيانات[1].

إحدى طرق معالجة المشكلة هذه استخدام طريقة كبسلة وتغليف قاعدة البيانات في احد التطبيقات واستخدامها من قبل تطبيقات أخرى عبر هذا التطبيق باستخدام خدمات الويب[14].

6.استخدام المشاهدViewsفي قواعد البيانات

تسمح قواعد البيانات العلائقية بالوصول إلى البيانات بطرق مختلفة وتوفر تقنيات تسمح بالبحث بطرق مختلفة مثل استخدام المشاهد. والمشهد يشبه الجدول أو العلاقة، ولكن يتم تعريفه برمجياً والمشاهد تخفي المكان الحقيقي لخزن البيانات، وكذلك طريقة هيكلة قاعدة البيانات، ولكن بعض المشاهد مكلف برمجيا[15].وبالرغم من أن NOSQLليس فيها مشاهد كما في SQL ولكن فيها إمكانية تكوين استفسارات محسوبة مسبقاً ومخزونة ويمكن استخدام نفس التعبير ظاهرة المشاهد. عادة قواعد بياناتNOSQLتكون ظاهرة المشاهد باستخدام حسابات -Map Reduce.[14]

هناك نظريتان لبناء ظاهرة المشهد في قواعد بياناتNOSQL، الأولى النظرة المتلهفةEagerحيث يتجدد المشهد في نفس اللحظة لتجدد البيانات الأصلية له. وهذه الطريقة جيدة عندما يكون معدل القراءات أكثر من معدل الكتابة للبيانات في المجموعة أو المشهد، والمطلوب دائما أن يكون المشهد متجدداً قدر الإمكان[16].النظرة الثانية قاعدة بيانات التطبيقApplication database هي نظرة الشيء الثمين، هنا الأمر أكثر سهولة لضمان أي تغيير أو تعديل للبيانات الأصلية سوف يعدل المشهد مجدداً. ولتقليل زيادة الحمل عند كل تحديث يمكن عمل أو تنفيذ دفعة أعمال لتحديث عدة مشاهد عند مرور فترة زمنية معينة، وهذا يتطلب معرفة متطلبات العمل بشكل دقيق وكيفية وعدد مرات الوصول للبيانات، وكيف يتم تكوين المشاهد. ويمكن تكوين المشهد خارج حدود قاعدة البيانات باستخدام تطبيق برمجي وخزن المشهد بعدها في قواعد البيانات.ومعظم قواعد البيانات تدعم تكوين المشاهد بنفسها وتجهز عمليات حسابية لذلك عند الحاجة وحسب الطلب والمتغيرات المحددة.

6-1 تقنية المقابلة والتخفيضMap-Reduce

هي معالجة تقنية أو برنامج يستعمل كنموذج في الحسابات الموزعة، تتضمن خوارزمية المقابلة والتخفيض Map-Reduce كما هو واضح من الاسم عمليتين هما المقابلةMap وعملية التخفيض Reduce. تقوم عملية المقابلة باستقبال مجموعة البيانات وتحويلها إلى مجموعة أخرى، إذ أن عناصر البيانات الفردية تحول إلى أزواج من البيانات Tuples مكونة من (مفتاح Key/ قيمةValue) . البيانات الداخلة تكون بشكل ملف أو دليل وترسل إلى دالة المقابلة سطر بعد أخر. وتعالج هذه الخطوة البيانات وتنتج قطع صغيرة من البيانات المسماة Chuncks.

الخطوة الثانية هي عملية التخفيض، وهي تأخذ ناتج عملية المقابلة كإدخال وتتضمن مرحلة الخلط والتخفيض للبيانات. تقوم هذه العملية بدمج أزواج البيانات القادمة من عملية المقابلة إلى مجموعة اصغر من الأزواج الجديدة. كما يدل الاسم فان مهمة التخفيض لا تبدأ إلا بعد انتهاء مهمة المقابلة، وان النظام هو المسؤول عن إرسال مهمات المقابلة والتخفيض إلى الخادم المناسب في العنقود. لذلك تكون هذه التقنية مناسبة جداً للعمل في بيئة العناقيد إذ يمكن بسهولة حساب أي معالجة موزعة للبيانات عبر عقد متعددة تتألف من مئات أو آلاف المعالجات في العنقود الواحد وهذا أدى إلى انجذاب المبرمجين لاستعمال نموذج المقابلة والتخفيض[20].

7.تناسق البياناتData Consistency

واحد من التغييرات الكبيرة من قواعد البيانات العلائقية إلى قواعد بيانات NOSQL هي كيفية التفكير تجاه التناسق أو الاتساق. قواعد البيانات العلائقية تحاول بقوة الحفاظ على التناسق وتجنب التضارب، وهناك أنواع مختلفة للتناسق منها ما يتعلق بعمليات التحديث أو القراءة[21].

7-1 تناسق التحديثUpdate Consistency

يفقد التناسق عادة عندما يحصل تضارب في الكتابة Write-Write Conflictويتم ذلك عندما يأتي شخص بالكتابة على مقطع بيانات، ويأتي شخص ثاني بالكتابة على نفس مقطع البيانات، عندها سوف يلاحظ ان التحديث الثاني سيتم إلغاؤه على الرغم من أنه آخر تحديث. هناك طريقتان معتمدتان لصيانة الاتساق وهي طريقة التشاؤمPessimistic أو طريقة التفاؤلOptimistic تجاه التقاء الكتابة.طريقة النظرة المتشائمة تعمل بمنع النزاع من الحصول نهائيا، أما طريقة النظرة المتفائلة فهي تسمح بحدوث النزاعات وبعد اكتشافها تعمل على طرق لفك النزاع.[3]

في نزاعات تحديث النظرة المتشائمة الأكثر شيوعاً هي قفل الكتابة، ولأجل تحديث أي بيانات يجب امتلاك قفل الكتابة، وعندها النظام يضمن أن نقطة طرفية واحدة فقط هي التي تملك حق الكتابة في وقت معين. بينما النظرة المتفائلة هي الشائعة، وتتم عن طريق التحديث الشرطي، فأي نقطة طرفية تريد عمل تحديث يجب ان تفحص أولا القيمة قبل التحديث لرؤية التغيير على البيانات منذ آخر قراءة .كلتا النظريتين السابقتين تعتمد تسلسلات ثابتة للتحديث وهذا مع خادم يمكن معالجته الطرف الأول فالثاني وهكذا. لكن مع وجود أكثر من خادم في النظام والبيانات تستخدم طريقة الند للند لتكرار البيانات بين عدة نقاط عندها لو كان هناك طرفيتان تعملان على نفس البيانات كل منها على خادم مختلف تكون النتيجة تغير القيم بطريقة مختلفة في كل خادم.[4]

7-2 تضارب القراءة Read Consistency

عندما يكون هناك شخص يقوم بعملية كتابة وأخر يقوم بعملية قراءة لنفس البيانات في ان واحد فتكون هناك قراءة متناقضةInconsistent readأو ما يسمى تضارب قراءة كتابةRead-Write Conflictويشار إلى هذا النوع من تناسق البيانات المنطقيLogical Consistencyوهو ضمان جعل مختلف عناصر البيانات يتم تحديثها سوية. لتجنب التضارب المنطقي توفر قواعد البيانات العلائقية مبدأ المعاملات، وهو ضمان ان يتم تحديث مختلف عناصر البيانات سويةً.الادعاء الشائع ان NOSQLلا تدعم المعاملة وهذا لا يمكن ان يكون متوافق. إن هذا الادعاء خاطئ لأنه لا يبرر الكثير من التفاصيل[5].وقواعد بيانات المخطط البياني تميل إلى دعم المعاملة في حين تدعم قواعد بيانات المجموعة الموجهة التحديث الذري لكن فقط مع المجموعة الواحدة. أي هناك تناسق منطقي في المجموعة الواحدة وليس بين المجموعات. وبالطبع لا يمكن وضع جميع البيانات في مجموعة واحدة، لذا فان أي تحديث يؤثر على عدة مجموعات يترك المجال مفتوحاً لحدوث قراءات متضاربة عندما يقوم أكثر من زبون بالوصول إلى نفس البيانات[7] . يدعى طول الوقت (المدة الزمنية) لحدوث التضارب الحالي يدعى نافذة التضارب Inconsistency Window. ويجب على نظام قواعد بيانات NOSQL أن يكون له نافذة تضارب قصيرة.[3]

3-7 تناسق التكرار

وهو ضمان أن كل النسخ يصلها نفس التحديث لنفس البيانات في وقت واحد. مثلا اذا كان هناك ثلاث نقاط كل منها في منطقة جغرافية مختلفة، فعندما تريد اثنان من النقاط اجراء قراءة والنقطة الثالثة تنفذ تحديث في نفس الوقت سيتم ملاحظة أن النقطة الاقرب جغرافيا سوف ترى تحديثاً اسرع من النقطة الأبعد، لذلك يحصل تضارب تكرار. وفي النهاية تلقائيا كل التحديثات ستنتشر بالكامل على كافة النقاط. لذا تدعى هذه الحالة التوافق النهائي، أي مهما كان هناك من تضارب تكرار فإنه وبعد انتهاء عمليات التجديد سوف تكون كل النسخ قد تم تحديثها ولها نفس البيانات في النهاية.[3][4]

8.نظام إدارة قاعدة البيانات (NOSQL(CouchDB

هو نظام إدارة قاعدة بيانات NOSQL تختص بشكل كامل بخزن البيانات باستخدام نموذج وثائق JSON، وهو لا يتطلب هيكلاً خاصاً لجميع الوثائق[22]. نظام CouchDB يمكن أن يضم أكثر من قاعدة بيانات واحدة في نفس الخادم، وهو يدعم التوزيع لقاعدة البيانات باستخدام تقنية العنقدة وعمل التكرار Replication ويوفر نظام إدارة التضاربات(اكتشاف ومعالجة النزاعات في قاعدة البيانات بصورة تلقائية)[5]. تعد الوثيقة الوحدة الرئيسة لقاعدة البيانات، وكل وثيقة لها رمز مميز (ID) وحيد غير قابل للتكرار في قاعدة البيانات. وتتكون الوثيقة الواحدة من عدد غير محدد من الحقول والمؤشرات. الحقول في الوثيقة ليس لها شروط ثابتة أو أسماء محددة وتضم أنواعاً مختلفة من البيانات، ولا يوجد حد معين لحجم النص أو عدد عناصر المصفوفة، ويمكن تكرار اسم الحقل مع نوع بيانات مختلف[23].

يوفر نظام CouchDB طريقة جديدة لعرض بيانات قاعدة البيانات بشكل تفاعلي غير مشابهة لطريقة البرمجة المهيكلةSQL، إذ إن طبيعة بيانات الوثائق شبه مهيكلة وتختلف عن طبيعة بيانات الجداول المهيكلة[1][5]. يمكن الاستعلام عن البيانات المخزونة في قاعدة البيانات CouchDB باستخدام جمل بناء استعلامات يتم تنفيذها عبر لغة جافا سكربت وضمن نظام عمل طريقة Map-Reduce إذ تكون الوثيقة ادخالاً وتجري عليها المقارنات المطلوبة على حقول الوثيقة، وتحدد فيما إذا كانت هذه الوثيقة جزء من النتيجة أم لا. وتبنى الاستعلامات (المشاهد view) بشكل ديناميكي ولا تؤثر على الوثائق الأصلية، وكذلك فإن المشاهد التي يتم بناؤها في إحدى نسخ قاعدة البيانات تكون خاصة بها ولا يتم مزامنتها مع باقي النسخ.يوفر نظام CouchDB العديد من البرمجيات التي تسهل التعامل مع إدارة قواعد البيانات منها Faoxton وهو واجهة تطبيق سهلة الاستخدام ويوفر وصولاً كاملاً لكل عناصر ووظائف CouchDB

9.تكوين قاعدة البيانات جرد الموجودات Inventory DBS

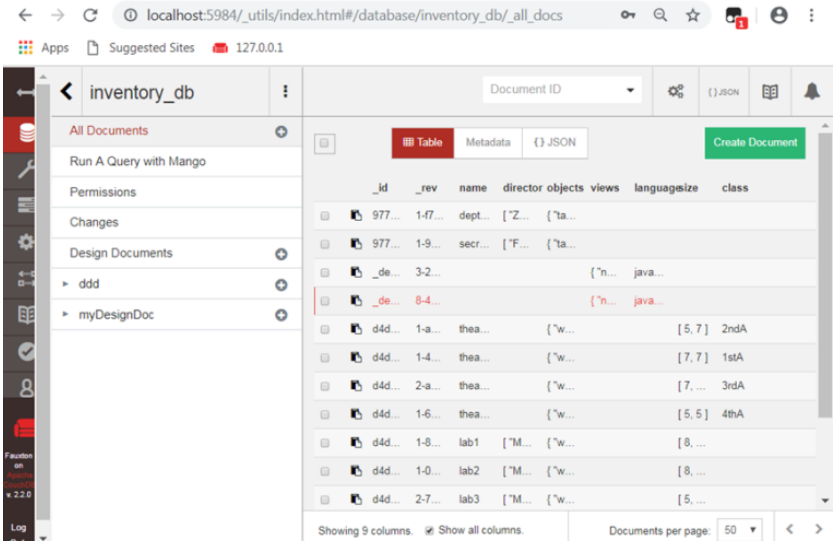
نظام جرد الموجودات هو نظام يعمل على إحصاء الموجودات الثابتة والمتحركة في أقسام المؤسسة حسب موقع تواجدها. وتتم عملية الجرد بتسجيل كافة الموجودات مع تفاصيلها المتواجدة في أحد أجزاء بناياتها ، ويقوم بهذه المهمة مجموعة من الأشخاص يشكلون أعضاء لجنة الجرد. تتم عملية الجرد عادة عند نهاية كل سنة أو عند ورود تغيرات أو تحديثات على موجودات المؤسسة. تتم عملية الجرد بتسجيل كافة المواد العينية على استمارات خاصة معدة مسبقا من قبل لجنة الجرد.

9-1 تنفيذ قاعدة البيانات باستخدام النموذج العلائقي

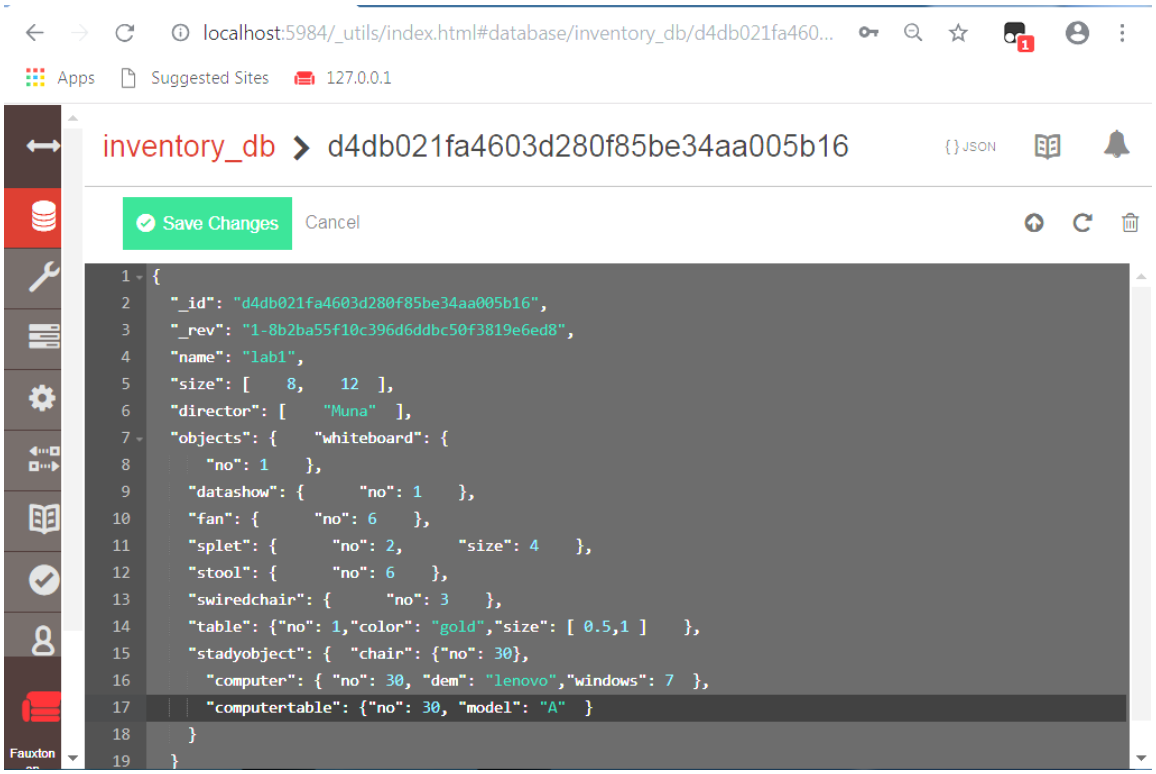
من أجل تحويل قاعدة بيانات الجرد إلى نظام حاسوبي باستخدام النموذج العلائقي يتم جمع وتحليل البيانات وتنظيمها بشكل مهيكل (جداول وعلاقات) ولأجل ذلك يتم عمل جدول بالمواد(الموجودات) وأماكن تواجدها ومواصفات كل منها. ومن الملاحظ أن الموجودات تختلف عن بعضها البعض في عدد ونوع المواصفات، وبذلك فمن الصعوبة تكوين جدول بالموجودات ومواصفاتها مثلا جهاز الطابعة يختلف بعدد مواصفاته عن الكرسي ويختلف عن جهاز التكييف. ولأجل حل هذه المشكلة يمكن تحديد اكبر عدد مسموح للصفات ليكن مثلا 5، هذه الطريقة سوف تسبب ضياعاً في مساحة الخزن لكون أن هناك عدداً كبيراً من سجلات الموجودات في الجدول لا تحمل هذا العدد من المواصفات، وإنما تكتفي باثنين أو ثلاث صفات وتترك باقي حقول الصفات خالية. كذلك عند إضافة موجودات جديدة وتحتاج هذه الموجودات إلى ذكر مواصفات اكثر عندها يجب تعديل بنية الجدول.

9-2 تنفيذ قاعدة البيانات باستخدام نموذج NOSQL CouchDB

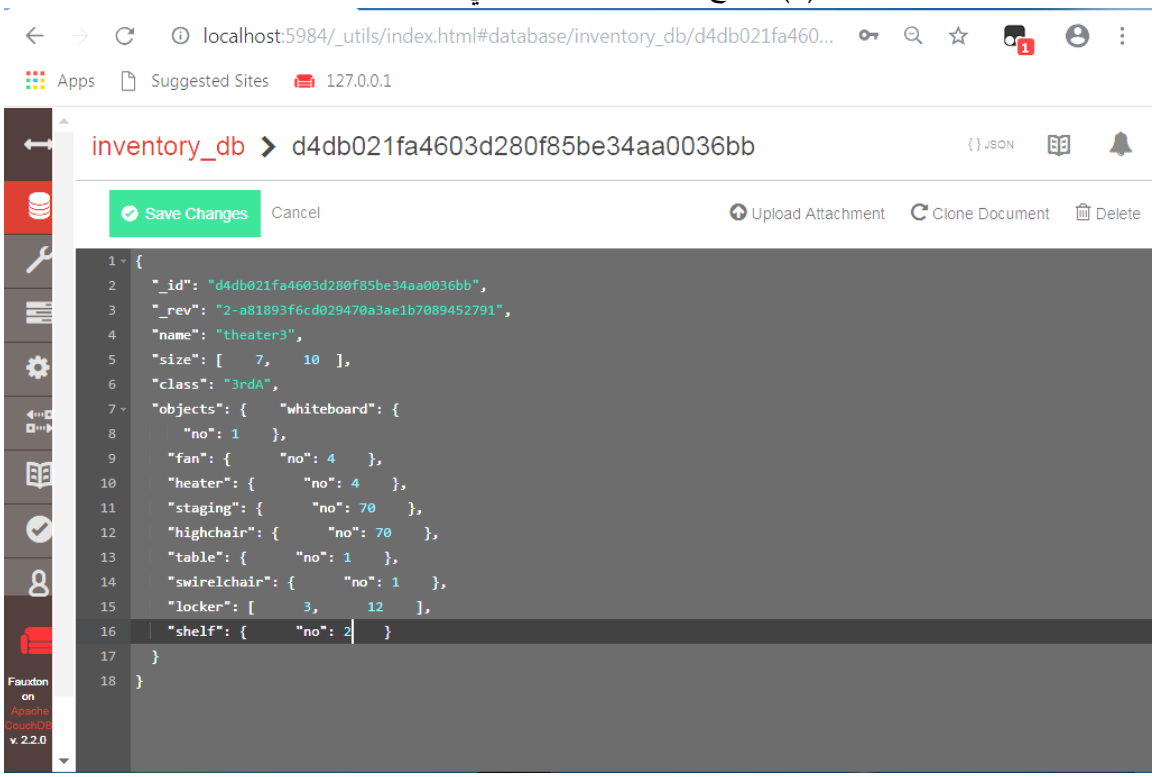
يعد استخدام نموذج وثيقة البيانات هو الأقرب لطبيعة البيانات الموجودة في قاعدة بيانات جرد الموجودات الورقي، إذ يتم خزن كل قائمة ورقية بوثيقة بيانات JSON . يمكن الاستعلام بسهولة عن الموجودات وصفاتها وأماكن تواجدها باستخدام الأدوات واللغات البرمجية التي يوفرها نظام CouchDB وباستخدام أي جزء من الوثيقة في الاستعلام. تم استخدام تطبيق Fauxton لأجل تكوين قاعدة بيانات الوثيقة وهي مكونة من ثلاثين وثيقة تمثل كل منها أحد أجزاء المبنى وتضم داخلها معلومات عن الموجودات وتفاصيلها كما هو موضح بـ(الشكل4) و(الشكل5) و(الشكل6)، بالإضافة الى ذلك تم تكوين مجموعة من المشاهد حسب الحاجة (مثل تنظيم الوثائق بالاعتماد على وجود نوع محدد من الأغراض أو حسب مواصفات الموجودات أو أسماء المسؤولين) باعتماد طريقة تقنية المقابلة والتخفيضMap-Reduce كما هو موضح بـ(الشكل7).



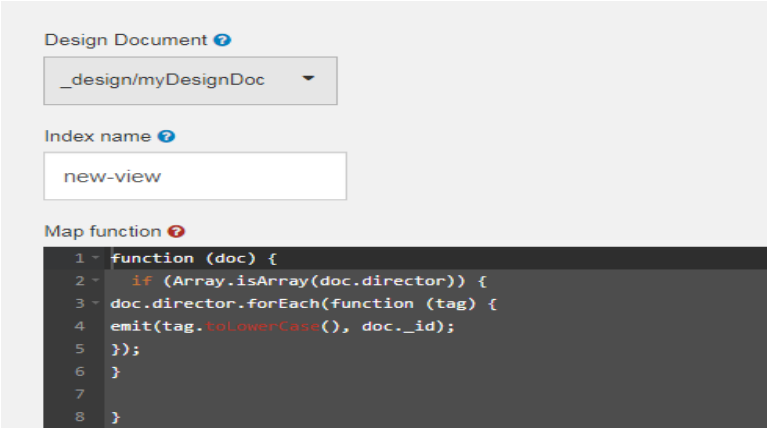
الشكل (4)مكونات قاعدة بيانات الجرد



الشكل (5) نموذج وثيقة بيانات لمختبر في قاعدة بيانات الجرد



الشكل (6) نموذج وثيقة بيانات لقاعة محاضرات في قاعدة بيانات الجرد



الشكل (7)نموذج تكوين مشهد Map-Reduce في قاعدة بيانات الجرد

3-9 لغة الاستعلام Cloudant-Query

تم العمل بطريقة Mango Query التي تستخدم لغة Cloudant Query للاستعلام عن بيانات وثائق JSON وتتكون جمل الاستعلام هذه من الاجزاء التالية التي تقابل الأجزاء الموجودة في لغة الاستعلام المهيكلة SQL:

- selector: وهي مجموعة البيانات التي سوف تم استعادتها.
- fields: الحقول التي سوف يتم استعادتها.
- sort: تبين كيف سيتم ترتيب عرض النتائج .
- limit: تحدد عدد النتائج التي ممكن عرضها
- مجموعة الدوال النصية والرقمية ودوال المقارنة

{ $text, $gt,$lt, $in[,],…}

- مجموعة العمليات المنطقية

{$and, $or, $not}

يوضح (الشكل8) استعلام1 الذي يمثل الاستعلام عن المواقع التي يوجد فيها خزانة locker وهو يقابل تنفيذ الأمر التالي بلغة SQL:

Select name, locker
from inventory
where locker <>null

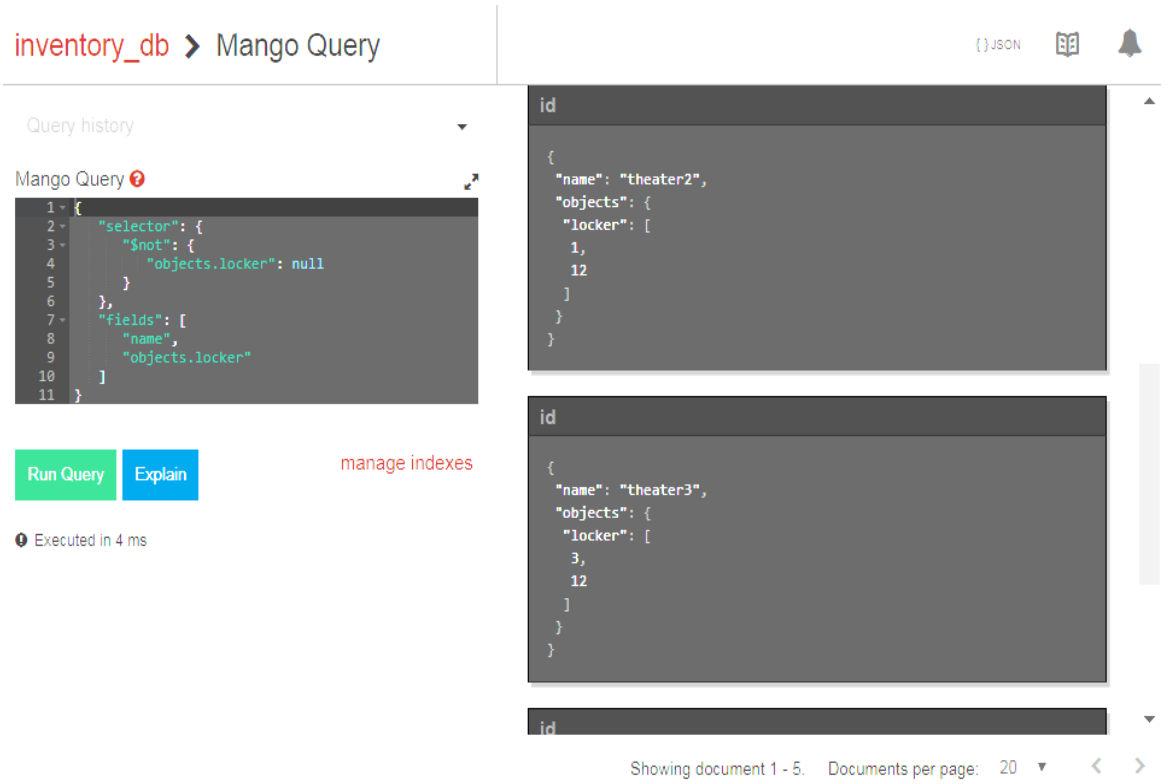
ويوضح (الشكل9) استعلام2 الذي يمثل الاستعلام عن المواقع التي يديرها أي من [Muna ,Ahmed] وهو يقابل تنفيذ الامر التالي بلغة SQL

Select name, director
from inventory
where director in ["Muna","Ahmed"]

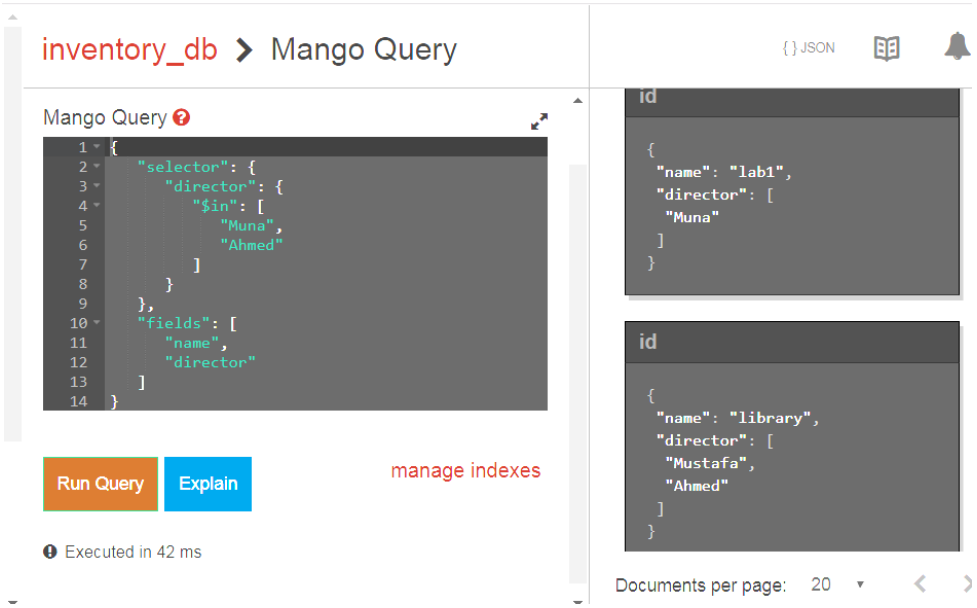
ويوضح (الشكل10) استعلام3 الذي يمثل الاستعلام عن المختبرات التي يكون فيها نقص اجهزة حاسوب

Select name, director , stadyobject
from inventory

where name like ("tab%") and computer_no< computertable_no
order by name



الشكل(8) استعلام1



الشكل(9) استعلام2

شكل(10) استعلام3

10. الاستنتاجات

إن كل من قواعد البيانات العلائقية وقواعد بيانات NOSQL لديه مميزات وعيوب وخصائص تميزها ولا توجد قاعدة بيانات أفضل من الأخرى، بل لكل منها استخدامات مختلفة حسب طبيعة النظام المراد تصميمه ويوضح (الجدول1) اهم الفروقات بين قواعد البيانات العلائقية وقواعد بيانات NOSQL:

جدول(1) الفروقات بين قواعد البيانات العلائقية وقواعد بيانات NOSQL

قواعد البيانات العلائقية	قواعد بيانات NOSQL
تعتمد مخطط بيانات ثابت مكون من جداول وعلاقات تربطها معاً	لا تعتمد مخطط ثابت للبيانات، وتميل إلى استخدام بروتوكولات مفتوحة المصدر للتواصل مع العميل وقابلة للتوسع
تدعم معالجة المناقلات وتستخدم شروط المناقلة ACID وتعالج تناسق البيانات بشكل سلس	تدعم تطبيقات الويب والتي لا تهتم بشكل كبير بشروط المناقلة ACID وبدلا من ذلك تستخدم نظرية CAP
تستخدم لغة الاستعلام المهيكلة SQL وتدعم الربط Join بين الجداول للاستعلام عن البيانات	لا يوجد لغة موحدة تستخدم لغات مختلفة حسب نوع البيانات الخاصة بكل نظام، مثال ذلك لغة Cassandra Query Language ولغة Cloudant Query
تعتمد مبدأ تطبيع البيانات Normalization للتخلص من تكرار البيانات	لا تهتم بقواعد تطبيع البيانات وتنظر لتكرار البيانات على أنه مشكلة يمكن تخطيها
تدعم مبدأ المعالجة المركزية لهياكل البيانات التي لا يمكن تقسيمها	تدعم المعالجة المتوازية على اكثر من خادم والبيانات تتبع عدة هياكل منفصلة مختلفة
من الصعوبة عمل تحجيم لهياكل البيانات لأنها تعامل كوحدة متكاملة	يمكن بسهولة تحجيم هياكل البيانات كونها منفصلة عن بعضها البعض

استخدام قواعد بيانات NOSQL يوفر طريقة جديدة للتعامل مع البيانات شبه المهيكلة وغير المهيكلة إذ تكون البيانات كثيرة التفاصيل وموزعة على مواقع متعددة. تعتبر البيانات الناتجة من جمع المعلومات عن عمليات جرد الموجودات في المؤسسات من مختلف مواقعها مثال جيد عن البيانات شبه المهيكلة والتي من الصعب جداً تنظيمها باستخدام النموذج العلائقي (بشكل جداول مكونة من صفوف واعمدة) وذلك لاختلاف طبيعة الموجودات في كل موقع ومواصفاتها عن المواقع الأخرى، بالإضافة إلى اختلاف طبيعة الموجودات في الموقع الواحد عن بعضها البعض. لذلك فإن تنظيم هذا النوع من البيانات في قواعد NOSQL هو الحل الأمثل وبشكل وثائق بيانات JSON التي تكون ذات طبيعة مرنة.

باستخدام نظام قواعد بيانات CouchDB لبناء قاعدة بيانات جرد الموجودات والاعتماد على نموذج وثيقة البيانات JSON كإدخالات سهل عمليات الاضافة والحذف والتعديل للموجودات في كل موقع مع إمكانية إضافة مواقع جديدة إلى قاعدة بيانات المؤسسة، حتى وإن كانت طبيعة الموجودات المضافة جديدة كلياً لأنها تضاف بشكل وثائق جديدة. الحقول في وثيقة JSON ليس لها شروط ثابتة أو أسماء محددة وتضم انواعاً مختلفة من البيانات، ولا يوجد حد معين لعدد الحقول في الوثيقة ولا حد معين لحجم النص أو عناصر المصفوفة، وكذلك يمكن تكرار اسم الحقل أكثر من مرة مع أنواع بيانات مختلفة ويتم التعامل معها عن طريق معرف الوثيقة.

استخدام طريقة العرض بشكل مشاهد ديناميكية تعتمد تقنية المقابلة والتخفيض ولا تؤثر على طبيعة البيانات الاصلية وتكون متوفرة عند الحاجة، ولا تحتاج الى إعادة تكوين في كل استخدام لها إذ إنها مشاهد مخزنة وتنظم البيانات فيها حسب طريقة العرض المطلوبة.عند التعامل مع قاعدة بيانات الوثيقة ممكن معالجة وطرح استفسارات واستعلامات بالاعتماد على حقول في المجموعة وكذلك يمكن استرجاع جزء من المجموعة بدل المجموعة بالكامل. وأيضا يمكن تكوين فهارس على أجزاء المجموعة. كما يدعم نظام قواعد بيانات CouchDB لغة الاستعلام Cloudant Query وهي لغة قوية تتعامل مع مكونات وثائق JSON لتكون طريقة استرجاع بيانات مشابهة لما تملكه لغة الاستعلام المهيكلة SQL.

المصادر

1. Vera, Harley and Valeria Guimaraes, Wagner Boaventura, CEUR Workshop Proceeding Vol.1478, September 2015, p.129-135.
2. Salehnia, Ali, "Comparisons of Relational Databases with Big Data: a Teaching Approach", South Dakota State University, Brookings, SD 57007.
3. Chandra, Deka G., Future Generation Computer Systems Vol.52, November 2015, p.13-21.
4. Meijer, Robert J. and Jan S. der Veen, IEEE Fifth International Conference on Cloud Computing 2012, June 2012, p.431-438.
5. Varga, Viorica and Katalin Tunde, Acta Polytech. Hungarica Vol.13.2, January 2016, p.229-248.
6. Woolf, Hohpe and Bobby Woolf, "Enterprise Integration Patterns". Addison Wesley. 2003. ISBN 0321200683.
7. Fowler, PoEAA., "Patterns of Enterprise Application Architecture". Addison Wesley. 2003. ISBN 0321127420

8. Robert, Daigneau, "Service Design Patterns". Addison-Wesley.2012.ISBN 032154420X
9. Hadjigeorgiou, Christoforos, MSc in High Performance Computing, The University of Edinburgh, 23 August 2013.
10. Mohamed, A. Mohamed and Mohammed O. Ismail, International Journal of Computer and Information Technology Vol.3.3, May 2014, p.598-601.
11. Lynch, Nancy and Seth. Gilbert, Acm Sigact News Vol.33.2, June 2002, p.51-59.
12. Brewer, Eric A., PODC Vol.7, July 2000.
13. Eric., Evans, "Domain-Driven Design". Addison-Wesley. 2004. ISBN 0321125215
14. Sadalage, Pramod J. and Martin Fowler, "NoSQL Distilled, A Brief Guide to the Emerging World of Polyglot Persistence", 2013 Pearson Education.
15. Gosain, Dishant and Ishita Kathuria, International Journal of Engineering Research & Technology Vol 1.6, August 2012, p.1-5.
16. Arnicans, Guntis and Girts Karnitis, 7th International Conference on Computational Intelligence, Communication Systems and Networks (CICSyN), 2015.
17. Mason, R. T., "NoSQL databases and data modeling techniques for a document-oriented NoSQL database". Informing Science & IT Education Conference (InSITE) 2015.
18. Chang, Fay, Jeffrey Dean and S. Ghemawat, ACM Transaction on computer system(TOCS) Vol.26.2, June 2008.
19. Lakshman, Avinash and P. Malik, SIGOPS Oper. Syst. Rev. 44.2 , 2010.
20. Yang, Hung-chih and Ali Dasdan, R. Hsiao, The 2007 ACM SIGMOD international conference on management of data, June 2007, p.1029-1040.
21. Gilbert, Seth and Nancy Lynch, IEEE computer Vol.45.2 , January 2012, p.30-36.
22. Drew, Adams, "Oracle Database JSON Developer's Guide", Copyright 2018.
23. Droettboom, Michael, "Understanding JSON Schema", Release 6.0,Space Telescope Science Institute, 2018.