Relational database VS Mongodb

|  |  |
| --- | --- |
| Relational database | Mongodb |
| Relational DBMS, מייצג נתונים בטבלאות ושורות. | Document-oriented, מייצג נתונים כאוסף של קבצי json. |
| שאילתות נכתבות באמצעות שפת SQL | שאילתות נכתבות באמצעות שפת json |
| אחד הדברים הטובים ביותר במסדי נתונים יחסיים זה האפשרות להוסיף פעולות/פונקציות. זה מאפשר לבצע שאילתות על פני מספר רב של טבלאות. | mongoDB אינו תומך בjoins , אבל הוא עושה סוגי נתונים רב ממדיים כגון מערכים ומסמכים אחרים. |
| עוד דבר טוב במסדי נתונים יחסי הוא התמיכה ב טרנזקציות אטומיות. היכולת להכיל פונקציות מרובות בטרנזקציה אחת ולהחזיר את הכול כאילו זו היתה פעולה אחת | אינו תומך בטרנזקציות, אבל כל פונקציה יחידה היא אטומית. |
| דורש להגדיר טבלאות ועמודות לפני שאפשר לאחסן נתונים, לכל שורה בטבלה חייבות להיות עמודות זהות. | הדבר הטוב בmongoDb שאין צורך להגדיר סכימה, זה דינאמי, פשוט מורידים את הקבצים לתוך אוסף ואפילו לא צריך לדאוג שהשדות היו זהים. |
| Scalability - היכולת להוסיף משאבים נוספים היא מסוג Scale vertically, סילומיות אנכית להוספה של משאבים לצומת יחיד במערכת, בד"כ ע"י הוספת מעבדים או זיכרון לאותו מחשב. | Scalability - היכולת להוסיף משאבים נוספים היא מסוג Scale horizontally, סילומיות אופקית, הכוונה להוספת צמתים נוספים למערכת כגון הוספה של מחשב חדש עבור ישום תוכנה מבוזר. כלומר אין הגבלה לכמות המידע. |
| הביצועים לעיתים ירודים מאוד. אבל אם משתמשים במסד נתונים פשוט וקובעים אינדקסים בצורה נכונה אפשר לקבל ביצועים טובים. | בכך שאינו תומך בjoins ולא מביא כלים יעילים לניתוח הביצועים, ביצועי mongoDB הרבה יותר טובים ביחס לבסיס נתונים יחסי. |
| מסדי נתונים יחסיים לא נועדו לתמיכה בפיתוח תוכנות agile, ולקלות הרחבה. | תומך בפיתוח תוכנות agile . כיום משתמשים הרבה בשיטות agile לפיתוח מוצרים, לריצות קצרות עם איטרציה מהירה. האפשרות להגיר סכימה דינאמית מאפשרת זאת. |
| מנהלי התקנים עבור שפות תכנות:  .NET, Java, PHP, Python, Ruby, Visual Basic | מנהלי התקנים עבור שפות תכנות:  Actionscript, C, C#, C++, Clojure, ColdFusion, D, Dart, Delphi, Erlang, Go, Groovy, Haskell, Java, JavaScript, Lisp, Lua, MatLab, Perl, PHP, PowerShell, Prolog, Python, R, Ruby, Scala, Smalltalk |
| תמיכה במערת הפעלה של Windows | תמיכה במערכות ההפעלה הבאות: Windows, Linux, OS X, Solaris |

איך לדעת איזה בסיס נתונים לבחור?

MySQL וMongoDB שניהם מאוד שימושיים, ויש הרבה הבדלים חשובים בינהים בפעולות הבסיסיות שלהם מאשר בביצועים. צריך באמת לדעת לבחור לפי צרכי היישום הספציפי.

אם המבנה נתונים משתלב יפה בטבלאות ועמודות,ואין דאגה לגבי הביצועים אז מסד נתונים יחסי יציע אינטראקציה חזקה וקלה עם הנתונים.

אם הנתונים נראים מורכבים למסד נתונים יחסי, ויש קידוד סביב בעיות ביצועים כדאי לשקול את שימוש MongoDB. אם אתה מוצא את עצמך מנסה לאחסן מערכים באופן סדרתי או אובייקטי JSON, שזה סימן שאתה צריך לבחור ב-MongoDB. אם אתה לא יכול להגדיר מראש הסכמה שלך או שאתה רוצה לאחסן רשומות באותו האוסף שיש לו שדות שונים, זה עוד סיבה טובה.

Databases שונים:

|  |  |  |  |
| --- | --- | --- | --- |
| שמות databases: | סוג | יתרונות | חסרונות |
| Neo4j, Infinite Graph, DEX, TITAN, InfoGrid, HyperGraphDB, GraphBase, Trinity, AllegroGraph, BrightstarDB, Bigdata, Meronymy, WhiteDB, OpenLink Virtuoso, VertexDB, FlockDB, BrightstarDB, Execom IOG,  Fallen 8, | graph database | * כאשר רוצים לנתח יחסים וקשרים מסדי נתונים גרפים הם נוחים מאוד בגלל מבנה הנתונים. * מסדי נתונים גרף הם מהירים מאוד עבור מערכות נתונים אסוציאטיביות כמו רשתות חברתיות. * טוב לאפליקציות של object .oriented סיווג אובייקט ויחסים של הורה וילד. | * אם הנתונים הם רק טבלאיים עם לא הרבה קשר בין הנתונים, אז מסדי נתונים גרף לא פועלים היטב. * תמיכת OLAP עבור מסדי נתונים גרף אינה מפותחת היטב. |
| Versant, db4o, Objectivity, GemStone/s, Perst, VelocityDB, HSS Database, ZODB, Magma, NEO, siaqodb, Sterling, Morantex, EyeDB, FramerD, Ninja Database Pro, NDatabase, PicoLisp, acid-state, ObjectDB, CoreObject | Object Databases | * אובייקטים אינם דורשים הרכבה ופירוק וכך חוסכים זמן קידוד וזמן ביצוע בלהרכיב או לפרק אובייקטים. * פחות קידוד. * ניווט קל יותר. * בקרת מקביליות טובה- היררכיה של אובייקטים יכולה להיות נעולה. * עובד היטב עם distributed architectures. * דורש פחות קוד כאשר יישומים הם object oriented. | * יעילות נמוכה כאשר יש נתונים וקשרים פשוטים. * טבלאות קשרים פשוטים. * לפעמים עשויים להאט מהירות גישה * סטנדרטים של בסיס נתונים רציונלי הם יותר יציבים. |
| CouchDB | document databases | * CouchDB הוא קל מאוד, ואין שום סיבה שאי אפשר להריץ מסדי נתונים מרובים. * תומך ב fast peer-to-peer replication מה יכול להתמודד הרבה עבודה ברמה הנמוכה בשביל המתכנת. | * יש צורך בכתיבת שאילתות Map/Reduce. * מהירות נמוכה ביחס לmongoDB CouchDB שומר נתונים כjson ואילו mongoDB שומר כ בינארי json כך זה יותר יעיל. |
|  |  |  |  |
|  |  |  |  |