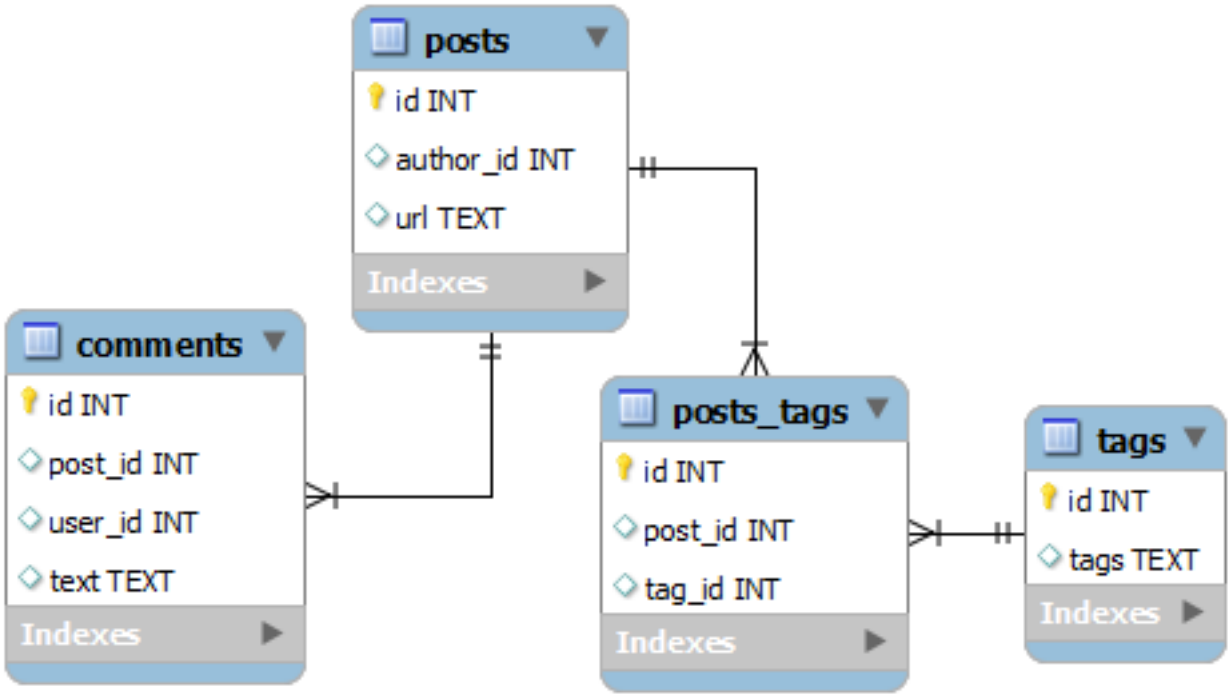
عنوان: **NOSQL قسمت سوم**
نویسنده: حمید سامانی
تاریخ: ۱۳۹۱/۱۱/۲۹ ۱۹:۱۰
آدرس: www.dotnettips.info
گروه‌ها: NoSQL, Database, نوسی‌کوال, نواس‌کیوال, mongodb, پایگاه

در مطلب قبلی با نوع اول پایگاه‌های‌داده NoSQL یعنی Key/Value Store آشنا شدیم و در این مطلب به معرفی دسته دوم یعنی Document Database خواهیم پرداخت.

در این نوع پایگاه داده ، داده‌ها مانند نوع اول در قالب کلید/مقدار ذخیره می‌شوند و بازگردانی مقادیر نیز دقیقا مشابه نوع اول یعنی Key/Value Store بر اساس کلید م ی‌باشد. اما تفاوت این سیستم با نوع اول در دسته‌بندی داده‌های مرتبط با یکدیگر در قالب یک Document می‌باشد. سعی کردم در این مطلب با ذکر مثال مطالب را شفاف‌تر بیان کنم:

به عنوان مثال اگر بخواهیم جداول مربوط به پست‌های یک سیستم CMS را بصورت رابطه‌ای پیاده کنیم ، یکی از ساده‌ترین حالات پایه برای پست‌های این سیستم در حالت نرمال به صورت زیر می‌باشد.



جداول واضح بوده و نیازی به توضیح ندارد ، حال نحوه‌ی ذخیره‌سازی داده‌ها در سیستم Document Database برای چنین مثالی را بررسی می‌کنیم:

```
{
_id: ObjectID(‘4bf9e8e17cef4644108761bb’),
Title: ‘NoSQL Part3’,
url: ‘http://dotnettips.info/yyy/xxxx’,
author: ‘hamid samani’,
tags: [‘databases’,’mongoDB’],
comments:[
{user: ‘unknown user’,
 text:’unknown test’
},
{user:unknown user2’,
 text:’unknown text2
```

```
}
]
}
```

همانگونه که مشاهد می‌کنید نحوه‌ی ذخیره‌سازی داده‌ها بسیار با سیستم رابطه‌ای متفاوت می‌باشد ، با جمع‌بندی تفاوت نحوه‌ی نگه‌داری داده‌ها در این سیستم و RDBMS و بررسی این سیستم نکات اصلی به شرح زیر می‌باشند:

-۱فرمت ذخیره سازی داده‌ها مشابه فرمت JSON می‌باشد.

-۲به مجموعه داده‌های مرتبط به یکدیگر Document گفته می‌شود.

-۳در این سیستم JOIN ها وجود ندارند و داده‌های مرتبط کنار یکدیگر قرار می‌گیرند ، و یا به تعریف دقیق‌تر داده‌ها در یک داکیومنت اصلی Embed می‌شوند .

به عنوان مثال در اینجا مقدار comment ها برابر با آرایه‌ای از Document ها می‌باشد.

-۴مقادیر می‌توانند بصورت آرایه نیز در نظر گرفته شوند.

-۵در سیستم‌های RDBMS در صورتی که بخواهیم از وجود JOIN ها صرفنظر کنیم. به عدم توانایی در نرمال‌سازی برخواهیم خورد که یکی از معایب عدم نرمال‌سازی وجود مقادیر Null در جداول می‌باشد؛ اما در این سیستم به دلیل Schema free بودن می‌توان ساختارهای متفاوت برای Document ها در نظر گرفت.

به عنوان مثال برای یک پست می‌توان مقدار n کامنت تعریف کرد و برای پست دیگر هیچ کامنتی تعریف نکرد.

-۶در این سیستم اصولا نیازی به تعریف ساختار از قبل موجود نمی‌باشد و به محض اعلان دستور قرار دادن داده‌ها در پایگاه‌داده ساختار متناسب ایجاد می‌شود.

با مقایسه دستورات CRUD در هر دو نوع پایگاه داده با نحوه‌ی کوئری گرفتن از Document Database آشنا می‌شویم:

در SQL برای ایجاد جدول خواهیم داشت:

```
CREATE TABLE posts (
    id INT NOT NULL
        AUTO_INCREMENT,
    author_id INT NOT NULL,
    url VARCHAR(50),
    PRIMARY KEY (id)
)
```

دستور فوق در Document Database معادل است با:

```
db.posts.insert({id: “256” , author_id:”546”,url:"http://example.com/xxx"}) // با قرار دادن مقدار نوع ساختار مشخص می‌شود
```

در SQL جهت خواندن خواهیم داشت:

```
SELECT * from posts
WHERE author_id > 100
```

و معادل آن برابر است با:

```
db.posts.find({author_id:{$gt:”1000”}})
```

در SQL جهت بروزرسانی داریم:

```
UPDATE posts
SET author_id= "123"
```

که معادل است با:

```
db.posts.update({ $set: { author_id: "123" }})
```

در SQL جهت حذف خواهیم داشت:

```
DELETE FROM posts
WHERE author_id= "654"
```

که معادل است با:

```
db.posts.remove( { author_id: "654" } )
```

همانگونه که مشاهده می‌فرمایید نوشتن کوئری برای این پایگاه داده ساده بوده و زبان آن نیز بر پایه جاوا اسکریپت می‌باشد که برای اکثر برنامه‌نویسان قابل درک است.

تاکنون توسط شرکت‌های مختلف پیاده‌سازی‌های مختلفی از این سیستم انجام شده است که از مهم‌ترین و پر استفاده‌ترین آنها می‌توان به موارد زیر اشاره کرد:

MongoDB

CouchDB

RavenDB

نظرات خوانندگان

نویسنده: سعید یزدانی
تاریخ: ۱۳۹۱/۱۱/۲۹ ۱۹:۵۷

با تشکر از مطلب زیباتون
یک سوال داشتم ایا این روش اونقدر به بلوغ رسیده که بشه در پروژه‌ها روش حساب کرد . یا اینکه فعلا از همون روش قبلی استفاده کنیم
سوال دیگر من هم این هست که به نظر شما در nosql اینده ایی دیده میشه ؟
با تشکر

نویسنده: سعید یزدانی
تاریخ: ۱۳۹۱/۱۱/۲۹ ۱۹:۵۹

اگر هم امکان داره refrence ی در این زمینه هست link بدید

نویسنده: حمید سامانی
تاریخ: ۱۳۹۱/۱۱/۲۹ ۲۱:۳

در رابطه با سوال اولتون عارضم که در حال حاضر همه‌ی شرکت‌های بزرگ و فعال در این صنعت مثل گوگل ، فیس ب و ک توئیتر و از این شیوه استفاده می‌کنند ، در حالت کلی این مبحث یک تکنولوژی خاص نیست که مصرفی باشه و بعد از مدتی تاریخش بگذره ، یک Movement و یا یک نگرش کلی در تعریف عامه از مجموعه‌ای از راه حل‌ها به منظور رفع مشکلاتRDBMS در پردازش داده‌های بزرگ (BigData) ، داده‌ها در حوزه‌ی وب هم که رشدی نمایی دارند.
در رابطه با سوال دوم هم بستگی به خود فرد و یا شرکت مربوطه داره ، در حوزه‌ی نرم‌افزارهای داخلی به دلیل پایین‌تر بودن حجم داده‌ها الزامی در استفاده از این روش‌ها نیست. (استفاده و یا عدم استفاده مستقیما به نوع نرم‌افزار و ساختار آن بستگی دارد)

نویسنده: حمید سامانی
تاریخ: ۱۳۹۱/۱۱/۲۹ ۲۱:۵

از اینجا که شما شروع کنید به همه جا لینک می‌شوید :)

نویسنده: سعید یزدانی
تاریخ: ۱۳۹۱/۱۱/۲۹ ۲۱:۲۴

ممنون بابت جواب کاملتون

نویسنده: توحید عزیزی
تاریخ: ۱۳۹۱/۱۱/۳۰ ۳:۵۲

سلام

سپاسگزارم از موضوع جالبی که انتخاب کرده اید و مطالب خوبی که می‌نویسید.
آیا امکان دارد که در مورد هر کدام از انواع دیتابیس نوسیکوئل، مثالهای بیشتری بزنید.
از یک سرویس رایگان برای نوشتن مثال‌ها می‌تواند استفاده کرد که برای همه در دسترس باشد، مثل: cloudant.com
با تشکر

نویسنده: Meysam Navaei
تاریخ: ۱۳۹۱/۱۱/۳۰ ۹:۱۶

سلام

توحید این سایت که معرفی کردی خیلی جالب بود.می خاستم بینم محدودیت حجمی در استفاده ازش وجود داره یا نه نامحدود. ریسک محسوب نمیشه‌ی پروژه بزرگ داشته باشی و بخای دیتابیس رو از سرویس این سایت استفاده کنی؟منظورم اینکه از این سایت‌ها نباشه که یهو محدودیت ایجاد بکنه و یا پولی بشه و...

نویسنده: حمید سامانی
تاریخ: ۱۳۹۱/۱۱/۳۰ ۱۶:۱۵

سلام
سعی می‌کنم مثال‌های بیشتری را در مطالب آتی بگنجانم.
با سپاس :)

نویسنده: توحید عزیزی
تاریخ: ۱۳۹۱/۱۲/۰۳ ۹:۳

سلام.
نسخه رایگانش محدودیت داره: فکر کنم 2000 کوئری در روز.
اگر می‌خواهید پروژه‌ی بزرگ روش ببرید، باید از نسخه‌های تجاریش استفاده کنید. البته من خودم تستش نکرده ام هنوز.
https://cloudant.com/#home-pricing

نویسنده: masi
تاریخ: ۱۳۹۲/۰۲/۱۹ ۱:۵۶

سلام ، واقعا مطالب خوبی بود هر جا رو گشتم کاملتر و جامع‌تر از همه بودید ، موضوع پروژه‌ی من روی این موضوعه ، ای کاش میشد در مورد موضوع زیر صحبت کنید. key value store

نویسنده: جواد زبیدی
تاریخ: ۱۳۹۲/۰۵/۱۶ ۳:۳۳

سلام تشکر از مطلب بسیار مفیدتون .
می خواستم بدونم که کدام یک از روش‌ها بیشتر امتحان خودش رو توی داده‌های زیاد پس داده .و بشه راحتر باهاش کار کرد .
روش
Document store
Key value
روش هایی دیگری رو هم توی سایت دیدم اگر امکان داره مزایا و معایب هر کدوم رو توضیح دهید ممنون.

نویسنده: دادخواه
تاریخ: ۱۳۹۲/۰۶/۰۷ ۱۲:۱۰

سلام
تشکر از مطالب خوبتون
اما چند تا سوال دارم.
1- از این سه تا پایگاه داده که در اخر نوشتید فکر کنم فقط MongoDB مجانی باشه. درسته؟
2- آیا دستورات در همه این پایگاه داده‌ها به همین صورت است؟
3- آیا همه سرورها و هاست‌ها از این پایگاه داده‌ها مانند MS SQL پشتیبانی می‌کنند و یا سرورهای خاص را باید پیدا کرد؟
تشکر

نویسنده: محسن خان

تاریخ: ۱۳۹۲/۰۶/۰۷ ۱۲:۲۵

اگر مطالب مقدماتی‌تر رو مطالعه می‌کردید، می‌دید که اصلا هدف از بانک اطلاعاتی NoSQL این نیست که باهاش سایت معمولی درست کنند اون هم روی سرور اجاره‌ای با 100 مگ فضا. هدفش توزیع شده بودن در سرورهایی متعدد و یا با پراکندگی جغرافیایی بالا است.

نتیجه گیری؟ ابزار زده نباشید. اول مفاهیم رو مطالعه کنید. اول تئوری کار مهمه.

نویسنده: saremi
تاریخ: ۱۳۹۲/۱۱/۲۴ ۱۶:۴۴

با سلام؛ میخواستم در مورد ،HQL، CQL ،UNQL، map reduce و... بپرسم. توی همون سایتی که لینکش رو دادین این‌ها جزو انواع query methodها هستند. من دقیق نمی‌فهمم الان ما دستورات مشابه sql رو معادلش رو با java script نوشتیم. در مورد تفاوت اینها و استفاده شون اگر میشه کمی توضیح بدین لطفا. دقیقا توی انواع مختلف پایگاه داده با چه زبانی کوئری نویسی می‌شه؟ با تشکر

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۱۱/۲۴ ۱۶:۵۲

در مورد تفاوت این‌ها در مطلب مروری بر مفاهیم مقدماتی NoSQL بیشتر توضیح داده شده. برچسب NoSQL را بهتر است دنبال کنید.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۱۱/۲۴ ۱۷:۴۹

در مورد MongoDB یک کتابچه‌ی فارسی 90 صفحه‌ای موجود است .

نویسنده: salam
تاریخ: ۱۳۹۳/۰۵/۰۹ ۱۱:۱۶

سلام
در قسمت دوم این مطلب اومده که "در حالت کلی پایگاهای داده NoSQL به ۴ دسته تقسیم می‌شوند " دسته 3 و 4 را توضیح نمی‌دین؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۵/۰۹ ۱۱:۲۶

برای دنبال کردن مطالب هم خانواده در این سایت، در ذیل هر مطلب یک سری گروه یا برچسب تعریف شده‌اند. برای مثال اگر برچسب NoSQL را دنبال کنید، در مطالب دیگری پاسخ خود را خواهید یافت.