عنوان: **NOSQL قسمت اول**
نویسنده: حمید سامانی
تاریخ: ۱۳۹۱/۱۱/۲۴ ۱۷:۴۰
آدرس: www.dotnettips.info
گروه‌ها: NoSQL, Database, پایگاه‌داده, نوسی‌کوال, نواس‌کیوال

در این پست نگاهی کلی به ویژگی‌های پایگاه‌های داده NOSql خواهیم داشت و با بررسی تاریخچه و دلیل پیدایش این سیستم‌ها آشنا خواهیم شد.
با فراگیر شدن اینترنت در سال‌های اخیر و افزایش کاربران ، سیستم‌های RDBMS جوابگوی نیازهای برنامه‌نویسان در حوزه‌ی وب نبودند زیرا نیاز به نگهداری داده‌ها با حجم بالا و سرعت خواندن و نوشتن بالا از جمله نقط ضعف سیستم‌های RDBMS میباشد ، چرا که با افزایش شدید کاربران داده‌ها اصولا به صورت منطقی ساختار یکدست خود را جهت نگه‌داری از دست می‌دهند و به این ترتیب عملیات نرمال سازی منجر به ساخت جداول زیادی می‌شود که نتیجه آن برای هر کوئری عملیات Joinهای متعدد می‌باشد که سرعت خواندن و نوشتن را به خصوص برای برنامه‌های با گستره‌ی وب پایین می‌آورد و مشکلات دیگری در سیستم‌های RDBMS که ویژگی‌های سیستم‌های NoSql مشخص کننده آن مشکلات است که در ادامه به آن می‌پردازیم.
طبق تعریف کلی پایگاه داده NOSql عبارت است از:
نسل بعدی پایگاه داده (نسل از بعد RDBMS) که اصولا دارای چند ویژگی زیر باشد:
-۱ داده‌ها در این سیستم به صورت رابطه‌ای (جدولی) نمی‌باشند
-۲داده‌ها به صورت توزیع شده نگهداری می‌شوند.
-۳سیستم نرم‌افزاری متن باز می‌باشد.
-۴پایگاه داده مقیاس پذیر به صورت افقی می‌باشد(در مطالب بعدی توضیح داده خواهد شد.)

همان‌گونه که گفته شد این نوع پایگاه داده به منظور رفع نیازهای برنامه‌های با حجم ورود و خروج داده بسیار بالا (برنامه‌های مدرن وب فعلی) ایجاد شدند.
شروع کار پیاده‌سازی این سیستم‌ها در اوایل سال ۲۰۰۹ شکل گرفت و با سرعت زیادی رشد کرد و همچنین ویژگی‌های کلی دیگری نیز به این نوع سیستم اضافه شد.
که این ویژگی‌ها عبارتند از:
Schema-free : بدون شَما ! ، با توجه به برنامه‌های وبی فعلی ممکن است شمای نگه‌داری داده‌ها (ساختار کلی) مرتبا و یا گهگاهی تغییر کند. لذا در این سیستم‌ها اصولا داده‌ها بدون شمای اولیه طراحی و ذخیره می‌شوند. (به عنوان مثال می‌توان در یک سیستم که مشخصات کاربران وارد سیستم می‌شود برای یک کاربر یک سری اطلاعات اضافی و برای کاربری دیگر از ورود اطلاعات اضافی صرف‌نظر کرد ، و در مقایسه با RDBMS به این ترتیب از ورود مقادیر Null و یا پیوندهای بیمورد جلوگیری کرد.
کنترل اطلاعات الزامی توسط لایه سرویس برنامه انجام می‌شود. (در زبان جاوا توسط jsr-303 و یاBean Validation ها)
easy replication support : در این سیستم ، نحوه‌ی گرفتن نسخه‌های پشتیبان و sync بودن نسخه‌های مختلف بسیار ساده و سر راست می‌باشد و سرور پایگاه داده به محض عدم توانایی خواندن و یا نوشتن از روی دیسک سراغ نسخه‌ی پشتیبان می‌رود و آن نسخه را به عنوان نسخه‌ی اصلی در نظر می‌گیرد.
Simple API : به دلیل متن‌باز بودن و فعال بودن Community این سیستم‌ها APIهای ساده و بهینه‌ای برای اکثر زبان‌های برنامه‌نویس محبوب ایجاد شده است که در پست‌های بعدی با ارائه مثال آنها را بررسی خواهیم کرد.
eventually consistent : در سیستم‌های RDBMS که داده‌ها خاصیت ACID را (در قالب Transaction) پیاده می‌کنند ، در این سیستم‌های داده‌ها در وضعیت BASE قرار دارند که سرنام کلمات Eventual Consistency ، Soft State ، Basically Available می‌باشد.
huge amount of data : این سیستم‌ها به منظور کار با داده‌های با حجم بالا ایجاد شده‌اند ، یک تعریف کلی می‌گوید اگر مقدار داده‌های نگهداری شده در پایگاه‌های داده برنامه شما ظرفیتی کمتر از یک ترابایت داده دارد از پایگاه داده RDBMS استفاده کنید واگر ظرفیت آن از واحد ترابایت فراتر می‌رود از سیستم‌های NOSql استفاده کنید.
به طور کلی پایگاه داده‌ای که در چارچوب موارد ذکر شده قرار گیرد را می‌توان از نوع NoSql که سرنام کلمه (Not Only SQL) می‌باشد قرار داد. تاکنون پیاده‌سازی‌های زیادی از این سیستم‌ها ایجاد شده است که رفتار و نحوه‌ی نگه‌داری داده‌ها (پرس‌وجو ها) در این سیستم‌ها با یکدیگر متفاوت می‌باشد.

جهت پیاده سازی پایگاه داده با این سیستم‌ها تا حدودی نگرش کلی به داده‌ها و نحوه‌ی چیدمان آنها تغییر می‌کند ، به صورت کلی

مباحث مربوط به normalization و de-normalization و تصور داده‌ها به صورت جدولی کنار می‌رود.
سیستم NoSql به جهت دسته‌بندی نحوه‌ی ذخیره‌سازی داده‌ها و ارتباط بین آنها به ۴ دسته کلی تقسیم می‌شود که معرفی کلی آن دسته‌بندی‌ها موضوع مطلب بعدی می‌باشد.