

# به نام خدا

امیر مهدی شاه حیدری

ازمایشگاه پایگاه داده

استاد میثاق یاریان

Sql & nosql

## SQL: بانک اطلاعاتی رابطه‌ای

اولین موضوعی که تفاوت کلیدی دو حالت را بیان می‌کند، شیوه‌ای است که این بانک‌های اطلاعات، داده‌ها را ساختاردهی و ساختمان بندی می‌کنند. بانک‌های اطلاعاتی SQL بسیار در این رابطه سختگیرانه عمل می‌کنند و همه چیز را بسیار مرتب قرار می‌دهند، درست مانند دفترچه‌های تلفن. این مدل در سال ۱۹۷۰ توسط IBM طراحی شد. بانک‌های اطلاعاتی رابطه‌ای می‌توانند شامل چندین جدول مختلف باشند، موجودیت‌ها در ردیف‌ها قرار می‌گیرد و برای نمایش داده‌های مربوط به هر آیت‌ما از ستون‌ها استفاده می‌کنیم، مانند ستون نام، ستون شماره تلفن و... در علم بانک‌های اطلاعاتی به رابطه میان جدول و نوع فیلدها عمدتاً schema گفته می‌شود. در یک بانک اطلاعاتی رابطه‌ای قبل از اینکه هرگونه داده‌ای وارد شود، ابتدا باید به صورت کامل و درست schema را تعریف کنیم.

برای اینکه یک بانک اطلاعاتی رابطه‌ای به درستی کار کند، داده‌هایی که شما وارد می‌کنید، لازم است که به صورت بسیار صحیح و ساختارمندی قرار بگیرند. وقتی یک شمای کلی یا schema درست را طراحی کنید، در بانک اطلاعاتی‌تان کمترین میزان افزونگی و یا ناهمخوانی داده‌ها را خواهید داشت. این موضوع برای تجارت‌هایی با میزان ریسک‌پذیری بالا، بسیار حیاتی و لازم است. وقتی شمای کلی را به صورتی ضعیف طراحی کنید، بعدها به صورت کامل دچار سرگیجه و سردرد خواهید شد. وقتی شما در این مدل بخواهید که یک آیت‌م جدید را به شمای بانک اطلاعاتی‌تان وارد نمایید نیاز دارید که تمام بانک اطلاعاتی را ویرایش کنید و همه موارد را با همدیگر همگام سازی کنید.

## SQL: زبان ساختارمند پرسش‌ها

SQL زبان برنامه‌نویسی است که معماران بانک‌های اطلاعاتی برای ساخت بانک‌های اطلاعاتی رابطه‌ای از آن استفاده می‌کنند. در یک بانک اطلاعاتی مانند MySQL با استفاده از این حالت می‌توانید کوئری‌هایی را اجرا کنید، داده‌هایی را برگشت دهید، ویرایش کنید، حذف نمایید و یا رکورد جدیدی را اضافه کنید. این زبان بسیار سبک است و از دسته زبان‌های اعلامی است، با استفاده از این زبان، توسعه‌دهنده می‌تواند بسیاری از کارهای سختی که انجام می‌دهد را به سادگی در این حالت انجام دهد.

یکی دیگر از دلایلی که بانک‌های اطلاعاتی SQL تا به این اندازه محبوب است، این است که در اکثر برنامه‌ها و پکیج‌های توسعه مانند LAMP و یا پشته‌های مبتنی بر Ruby وجود دارند. این نوع از بانک‌ها به خوبی درک شده‌اند و پشتیبانی بسیاری از آن می‌شود. به همین دلیل استفاده و اجرای آن کار ساده‌تر به نسبت حالت بعدی‌مان است.

## NoSQL: بانک اطلاعاتی غیر رابطه‌ای و توزیع شده

اگر داده‌های شما به صورت واضح تعیین نشده‌اند و یا اینکه ساختاردهی درستی ندارند، پس ساختن یک schema برای‌تان بسیار مشکل است. بجای اینکه می‌توانید از بانک‌های اطلاعاتی غیر رابطه‌ای استفاده کنید، این مدل از بانک اطلاعاتی بسیار منعطف‌تر نسبت به همتهای خود است. ایده بانک‌های اطلاعاتی غیر رابطه‌ای مانند فایل‌ها و پوشه‌ها است که در آن تمام اطلاعات مرتبط به هم (فایل‌ها) در یک جا (پوشه‌ها) قرار می‌گیرند. اگر یک وبلاگ وردپرسی از بانک اطلاعاتی NoSQL استفاده کند، هر کدام از فایل‌ها یک پست وبلاگی را در خود نگه می‌دارند، لینک‌های شبکه‌های اجتماعی، تلفن‌ها، متن‌ها و تمام موارد دیگر.

تلاش برای ذخیره، پردازش و آنالیز داده‌هایی که ساختارمند نیستند، به شما می‌گویند که بهتر است به دنبال یک جایگزین برای SQL باشید، در حقیقت بانک‌های اطلاعاتی NoSQL سطح دسترسی و کار بسیار منعطف‌تری را با داده به شما می‌دهد.

چگونگی کارکرد بانک‌های اطلاعاتی NoSQL در مقابل مدل رابطه‌ای این است که در بانک‌های اطلاعاتی NoSQL بجای اینکه داده‌ها در جدول‌ها ذخیره شود، در سندها ذخیره می‌شود. از این طریق می‌شود داده‌هایی ساختاردهی نشده را مانند (تصاویر، مقالات، ویدیوها و موارد مختلف دیگر) در یک سند ذخیره کرد و به سادگی آن‌ها را پیدا نمود، در این حالت، دیگر برای ذخیره موارد مختلف از جداول استفاده نمی‌شود. این موضوع درک‌پذیری بالاتری دارد، اما بخاطر داشته باشید که استفاده از این حالت نیازمند پردازش بیشتر و حافظه بیشتر نسبت به SQL است.

بانک‌های اطلاعاتی NoSQL همچنین مزیت دیگری را نیز ارائه می‌دهند و آن آسان بودن دسترسی است. بانک‌های اطلاعاتی SQL بیشتر به عنوان گزینه مناسبی برای اپلیکیشن‌هایی است که با زبان‌های برنامه‌نویسی شی‌گرا مانند جاوا، پایتون، پی‌اچ‌پی و... نوشته می‌شود. اما NoSQL برای توسعه‌دهندگانی مناسب است که از طریق API‌ها به داده‌ها دسترسی پیدا می‌کنند و بدون اینکه نیاز به یادگیری SQL یا ساختار و معماری بانک اطلاعاتی آشنایی داشته باشند، می‌توانند با آن کار کنند.

در رابطه با بانک‌های اطلاعاتی NoSQL ما انواع متفاوتی را داریم که در اینجا قصد داریم در رابطه با ۴ نوع مرسوم از آن‌ها صحبت کنیم:

- **مدل Key-Value :** در این مدل اطلاعات در یک راه بدون شما یا (شما کم) ذخیره می‌شوند و داده‌ها شامل کلیدها و مقادیر Index هستند. Cassandra، Azure، LevelDB و Riak نمونه‌هایی از این حالت هستند.
- **ذخیره ستونی :** به این مدل ذخیره Wide-Column نیز گفته می‌شود، در این حالت بجای ردیف‌ها، جداول و ستون‌های مربوط به داده ذخیره می‌شوند. با استفاده از این مدل شما بیشترین میزان از مقیاس پذیری و کارایی را در اختیار دارید. نمونه‌هایی از این حالت Hbase، BigTable و HyperTable است.
- **بانک اطلاعاتی سندی :** مدل Key-Value را در نظر بگیرید و حال پیچیدگی بیشتری را به آن اضافه کنید، هر کدام از سندهای موجود در این بانک اطلاعاتی داده‌های مربوط به خود و کلید منحصر به فرد خود را دارد، که از آن برای

بازیابی داده استفاده می شود. این مورد گزینه بسیار عالی برای ذخیره سازی، بازیابی و داده های مبتنی بر سند است، اما باید گفت که این مورد نیز به صورت منحصر به فردی ساختارمند است. نمونه هایی از این حالت MongoDB و CouchDB است.

- **بانک اطلاعاتی گراف** : آیا داده هایی دارید که با گراف ها ارتباط دارد؟ این متد یکی از بهترین راه حل ها برای شما خواهد بود. نمونه هایی از آن Polyglot و Neo4J است.

## برخی از تفاوت های کلی SQL و NoSQL

- بانک های اطلاعاتی SQL به صورت اصلی با عنوان بانک اطلاعاتی رابطه ای یا RDBMS شناخته می شود؛ بانک های اطلاعاتی NoSQL به عنوان بانک اطلاعاتی غیر رابطه ای و توزیع شده شناخته می شود.
- بانک های اطلاعاتی SQL داده ها را در جداول و ستون ها ذخیره می کند، این درحالیست که ذخیره سازی در بانک های اطلاعاتی NoSQL مبتنی بر سندات، key-value ها، گراف ها و wide-column است.
- بانک های اطلاعاتی SQL، شمای کلی را از بانک پیشنهاد می دهند، اما NoSQL برای این مورد منعطف است و می تواند بدون دردرس تغییر کند.
- بانک های اطلاعاتی SQL تمرکز و تاکید بسیاری روی موضوع ACID که مخفف (Atomicity, Consistency, Isolation and Durability) است تمرکز دارد، درحالیکه NoSQL روند CAP را اجرا می کند که مخفف (Consistency, Availability and Partition tolerance) است.
- برای محیط هایی با تراکنش بسیار زیاد، بانک های اطلاعاتی SQL توانایی بهتری را از خودشان نشان داده اند، این مورد برای اپلیکیشن هایی با حجم داده بالا بسیار مناسب هستند.
- مقیاس پذیری در بانک های اطلاعاتی مدل رابطه ای بسیار سخت است و تقریباً نیاز دارد که کلیت بانک اطلاعاتی را تغییر دهید، این در حالیست که تغییر دادن و توسعه یک بانک اطلاعاتی در سیستم بانک های اطلاعاتی NoSQL بسیار ساده تر و روند آسان تری دارد.
- در زمینه پشتیبانی کردن، بانک های اطلاعاتی مدل رابطه ای معمولاً از پشتیبانی خوبی برخوردار هستند و حتی برخی از شرکت ها به صورت رسمی و منحصرراً سرویس های پشتیبانی را برای یک شرکت بزرگ در نظر می گیرد. اما تنها برخی از بانک های اطلاعاتی NoSQL به این صورت عمل می کنند، بیشتر آن ها پشتیبانی رسمی ندارند و تنها راه حل برای یافتن پاسخ ها استفاده از انجمن ها و مواردی مانند این است.

## دلایلی برای استفاده از بانک اطلاعاتی SQL

وقتی موضوع به تکنولوژی های بانک اطلاعاتی مربوط می شود، نمی شود گفت که تنها یک راه حل کلی وجود دارد و تمام. درست به همین دلیل است که بسیاری از شرکت ها همزمان از هر دو سیستم بانک اطلاعاتی استفاده می کنند. با وجود آنکه ممکن است سرعت و مقیاس پذیری بالای NoSQL شما را شگفت زده کند اما موارد دیگری نیز وجود دارد که در آن نیاز دارید که از یک سیستم بانک اطلاعاتی SQL استفاده کنید. دو مورد برای اینکه از بانک اطلاعاتی SQL استفاده کنید در زیر ذکر شده است:

- **شما باید از سازگاری ACID مطمئن شوید.** وجود سازگاری ACID در بانک های اطلاعاتی در کاهش آنومالی و ناسازگاری داده کمک بسیاری می کند، با این کار همه چیز مطابق با میل شما انجام می شود. ممکن است این قابلیت در برخی از بانک های اطلاعاتی NoSQL وجود نداشته باشد و یا اینکه چندان به آن اهمیت داده نشده باشد. البته این موضوع نیز به دلیل بالا بودن کارایی و مقیاس پذیری آنهاست.
- **داده های شما ساختارمند است و تغییر نمی کند.** اگر از رشد نکردن و تغییر نکردن مداوم بانک اطلاعاتی تان مطمئن هستید، پس بدانید که SQL انتخاب بسیار مناسبی برای شما خواهد بود.

## دلایلی برای استفاده از بانک‌های اطلاعاتی NoSQL

وقتی قصد دارید تا اپلیکیشن‌هایی با سرعت بالا ایجاد کنید و بتوانید با داده‌های بزرگ و مسئله‌ای مانند BigData کار کنید، بانک اطلاعاتی NoSQL می‌تواند انتخاب بسیار خوبی برای شما باشد. انجام این کار با بانک‌های اطلاعاتی سنتی کاری سخت و دشوار است، به همین دلیل برای انجام چنین کارهایی بانک اطلاعاتی مانند MongoDB، CouchDB و ... انتخاب بسیار خوبی هستند.

- **ذخیره‌سازی داده‌هایی با حجم بسیار و ساختاردهی نشده.** در بانک‌های اطلاعاتی NoSQL شما هیچگونه محدودیتی روی نوع داده‌هایی که در کنار هم قرار می‌دهید ندارد. همچنین به شما اجازه می‌دهد آنگونه که می‌خواهید داده‌ها را تغییر دهید. با استفاده از روش مبتنی بر سند، شما می‌توانید داده‌های مختلفی را بدون اینکه نوع آن‌ها مسئله مهمی برای ذخیره‌سازی باشد، در یک مکان قرار دهید.
- **ذخیره‌سازی و انجام محاسبات ابری.** ذخیره‌سازی مبتنی بر ابر یکی از راحل‌های بسیار مناسب همراه با میزان هزینه خوب است، اما برای بالاتر رفتن میزان مقیاس‌پذیری نیاز است که داده‌ها را در سرورهای مختلف منتشر کنید.
- **توسعه سریع.** وقتی قصد دارید یک پروژه جدید را راه اندازی کنید، در حالتی که از بانک‌های اطلاعاتی رابطه‌ای استفاده کنید، باید ابتدای کار ساختار برنامه را مشخص کنید و شمای کلی برای آن طراحی نمایید که این کارها در ابتدا بسیار زمان‌بر است. اما در حالت بانک اطلاعاتی NoSQL دیگر این مشکلات را نخواهید داشت.