

توسعه بانک‌های اطلاعاتی غیررابطه‌ای به واسطه نیاز شرکت‌هایی همچون فیسبوک، گوگل، آمازون و دیگر بزرگان عرصه فناوری که با حجم بسیار زیادی از اطلاعات روبه‌رو بودند شکل گرفت. بانک‌های اطلاعاتی NoSQL به طور فزاینده در ارتباط با بزرگ‌داده‌ها و برنامه‌های وبی که باید اطلاعات را به صورت بی‌درنگ دریافت و پردازش می‌کردند به کار گرفته شد. با توجه به اهمیت بانک‌های اطلاعاتی غیررابطه‌ای در این مقاله تصمیم گرفتیم انواع مختلف این بانک‌های اطلاعاتی را به شما معرفی کنیم و در ادامه قابلیت‌های هر یک از این بانک‌ها را به شما نشان دهیم.

بانک‌های اطلاعاتی غیررابطه‌ای که بعضی منابع از عبارت Not Only SQL برای آن‌ها استفاده می‌کنند، راهکاری ویژه برای مدیریت داده‌ها و همچنین طراحی بانک‌های اطلاعاتی به شمار می‌روند که قرار است حجم بسیار گسترده‌ای از داده‌های غیرمتمرکز را سازمان‌دهی کنند. NoSQL مشتمل بر طیف گسترده‌ای از فناوری‌ها و معماری‌هایی است که به دنبال حل مسائل مرتبط با گسترش‌پذیری و همچنین افزایش کارایی بزرگ داده‌ها هستند. دو نیازمندی مهم عصر کنونی که بانک‌های اطلاعاتی رابطه‌ای همچون MySQL قادر نیستند پاسخ‌گوی آن‌ها باشند. فناوری NoSQL به‌ویژه برای سازمان‌هایی مناسب است که در نظر دارند به حجم بسیار زیادی از داده‌های فاقد ساختار دسترسی پیدا کرده و آن‌ها را تجزیه و تحلیل کنند. داده‌هایی که در یک سرور راه دور ذخیره‌سازی نشده‌اند، بلکه روی سرورهای مجازی چندگانه‌ای در بستر کلاود میزبانی می‌شوند. فناوری NoSQL در واقع از سوی رهبران بزرگ اینترنت همچون فیسبوک، آمازون، گوگل و دیگران ساخته شد و در ادامه مورد استفاده قرار گرفت. شرکت‌هایی که به سامانه‌های مدیریت بانک‌های اطلاعاتی ویژه‌ای نیاز داشتند که بتوانند در هر نقطه از جهان آن‌ها را فراخوانی کرده، اطلاعات مورد نظر را درون آن‌ها نوشته یا از آن‌ها خوانده و در عین حال با مشکل گسترش‌پذیری و حجم رو به رشد داده‌ها روبه‌رو نشوند. حجم بسیار عظیمی از داده‌ها که در قالب میلیون‌ها مجموعه داده‌ای از سوی میلیون‌ها کاربر دریافت می‌شود.

مزایای بانک‌های اطلاعاتی NoSQL

بانک‌های اطلاعاتی NoSQL در مقایسه با بانک‌های اطلاعاتی رابطه‌ای سنتی مزایای متعددی را به همراه دارند. شکل 1 ویژگی‌های مربوط به بانک‌های اطلاعاتی غیررابطه‌ای را نشان می‌دهد. اما از مهم‌ترین مزایای NoSQL در مقایسه با بانک‌های اطلاعاتی رابطه‌ای مرسوم به موارد زیر می‌توان اشاره کرد.

عدم ساخت یافتگی

در بانک‌های اطلاعاتی NoSQL نیازی ندارید از همان ابتدا و به شکل کاملاً دقیق اسکیمای بانک اطلاعاتی خود را طراحی کنید تا در ادامه بتوانید داده‌های خود را در بانک اطلاعاتی مورد نظرتان وارد کنید. شما این توانایی را دارید تا کدنویسی را آغاز کرده و بدون اطلاع از این موضوع که بانک اطلاعاتی چگونه داده‌ها را ذخیره‌سازی کرده یا

مکانیسم داخلی آن به چه شکلی کار می‌کند فرآیند ذخیره‌سازی و دریافت داده‌ها را مدیریت کنید. همچنین اگر به قابلیت‌های پیشرفته‌ای نیاز دارید، به راحتی می‌توانید اسکیمای بانک اطلاعاتی را به شکل دستی و پیش از آنکه فرآیند نمایه‌سازی داده‌ها را آغاز کنید ویرایش و سفارشی‌سازی کنید. قابلیت ساخت یافته نبودن یکی از مهم‌ترین تفاوت‌های میان بانک‌های اطلاعاتی غیررابطه‌ای و بانک‌های اطلاعاتی رابطه‌ای است.

بانک‌های اطلاعاتی غیررابطه‌ای که بعضی منابع از عبارت **Not Only SQL** برای آن‌ها استفاده می‌کنند، راهکاری ویژه برای مدیریت داده‌ها و همچنین طراحی بانک‌های اطلاعاتی به شمار می‌روند که قرار است حجم بسیار گسترده‌ای از داده‌های غیرمتمرکز را سازمان‌دهی کنند

گسترش‌پذیری

بانک‌های اطلاعاتی NoSQL از رویکرد گسترش‌پذیری افقی پشتیبانی می‌کنند. این تکنیک باعث می‌شود تا بدون نیاز به ارتقای سخت‌افزار در سریع‌ترین زمان ممکن ظرفیت بانک اطلاعاتی را کم یا زیاد کنید. رویکرد فوق به میزان قابل توجهی از پیچیدگی‌ها می‌کاهد و هزینه‌های مربوط به تغییر اسکیمای بانک اطلاعاتی را کاهش می‌دهد.

کارایی

تعدادی از بانک‌های اطلاعاتی به‌گونه‌ای طراحی شده‌اند که بهترین عملکرد و کارایی را (تنها) زمانی از خود نشان می‌دهند که از تجهیزات سخت‌افزاری خاصی در ارتباط با فرآیند ذخیره‌سازی و پردازش اطلاعات استفاده شده باشد. در ارتباط با بانک‌های NoSQL برای آنکه بتوانید عملکرد را افزایش دهید تنها به اضافه کردن چند سرور ارزان‌قیمت که در اصطلاح به آن‌ها commodity Servers گفته می‌شود نیاز دارید. این تکنیک به سازمان‌ها کمک می‌کند به بهترین تجربه کاربری، قابلیت اطمینان بالا و سرعت بالا در وارد کردن و دریافت داده‌ها دست پیدا کنند. رویکردی که در نهایت به سازمان‌ها اجازه می‌دهد بدون آنکه به صرف هزینه‌های مضاعف نیازی داشته باشند به شیوه دستی فرآیندهای مربوط به طراحی بانک‌های اطلاعاتی را مدیریت کنند.

دسترس‌پذیری بالا

بانک‌های اطلاعاتی NoSQL با هدف در دسترس بودن (دسترس‌پذیری بالا) و همچنین دور بودن از پیچیدگی‌هایی که همراه با معماری رایج RDBMS به وجود می‌آید طراحی می‌شوند. بانک‌های اطلاعاتی با تکیه برندهای (Nodes) اولیه و ثانویه کار می‌کنند که همین موضوع بر پیچیدگی آن‌ها افزوده است. تعدادی از بانک‌های اطلاعاتی غیررابطه‌ای غیرمتمرکز از سبک خاصی از معماری که Masterless نام دارد استفاده می‌کنند. معماری ویژه‌ای که قادر است به شکل خودکار داده‌ها را میان منابع مختلف تقسیم کرده تا فرآیند ثبت و فراخوانی حتی زمانی که یکی از منابع در دسترس نیست امکان‌پذیر باشد. این تقسیم بار روی منابع مختلف به یک برنامه اجازه می‌دهد حتی زمانی که ندها در دسترس نیستند باز هم فرآیند خواندن و نوشتن را انجام دهند.

دسترس‌پذیری در مقیاس جهانی

بانک‌های اطلاعاتی غیرمتمرکز NoSQL به‌طور خودکار داده‌ها را میان سروهای متعدد، مراکز داده‌ها یا منابع کلاود تکثیر کرده تا به این شکل بر مشکل تأخیر زمانی غلبه کرده و آن را به حداقل برسانند. این تکنیک باعث می‌شود تا صرف نظر از موقعیت جغرافیایی، همه کاربران در زمان دسترسی به داده‌ها تجربه کاربری یکسانی داشته باشند. مزیت بزرگ دیگری که بانک‌های اطلاعاتی غیررابطه‌ای به همراه می‌آورند به کم کردن فرآیند مدیریت دستی بانک‌های اطلاعاتی باز می‌گردد. به طوری که به میزان قابل توجهی از حجم کار مربوط به پیکربندی بانک اطلاعاتی و کار با نرم‌افزار RDBMS کم می‌کند. رویکردی که به تیم‌های عملیاتی اجازه می‌دهد روی اولویت‌های جدی کسب و کار متمرکز شوند.



چالش بزرگ بعدی اکوسیستم داده‌محور چیست؟
انتقال سریع و هوشمندانه داده‌ها، معدن طلای عصر جدید

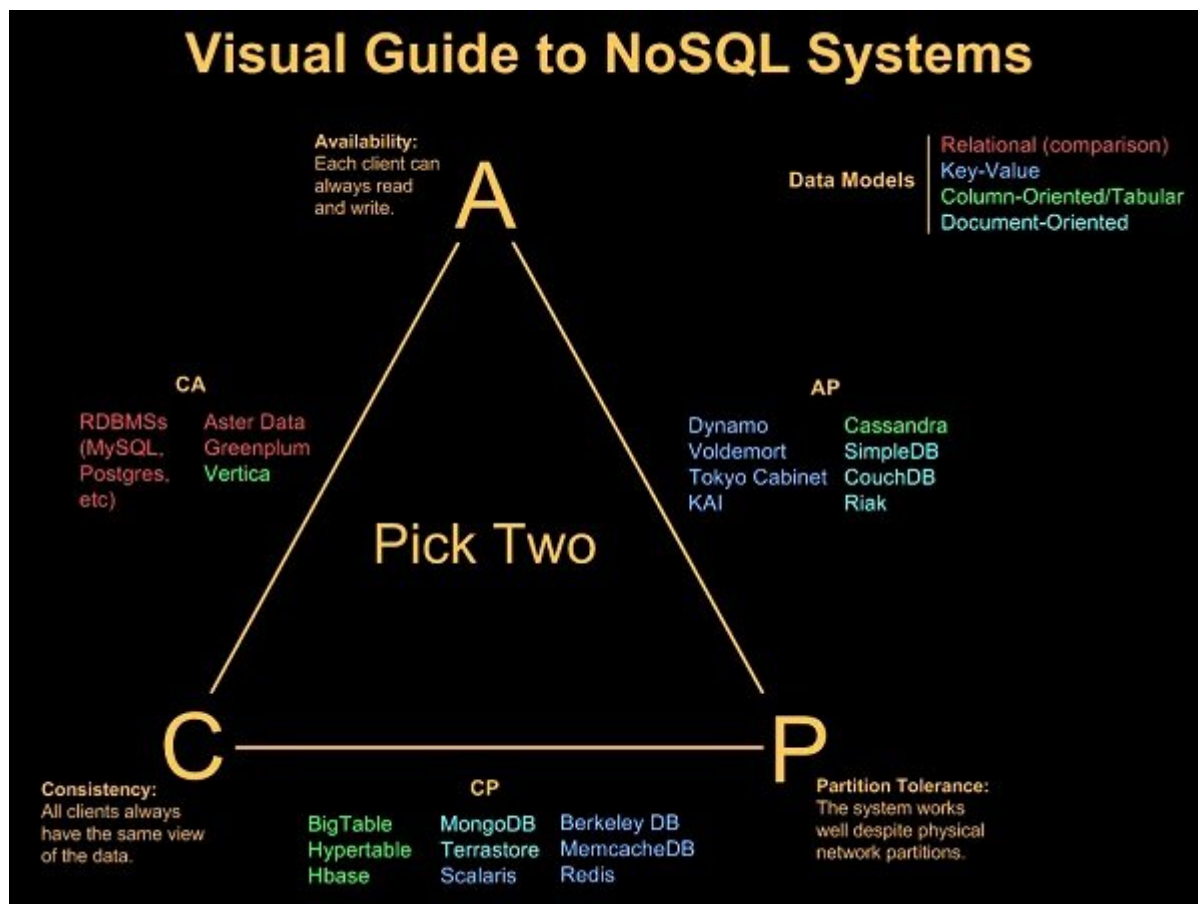
اشکال مختلف بانک‌های اطلاعاتی غیررابطه‌ای

تا به امروز انواع مختلفی از بانک‌های اطلاعاتی غیررابطه‌ای برای پاسخ‌گویی به نیازهای خاص سازمان‌ها طراحی و به کار گرفته شده‌اند. این بانک‌های اطلاعاتی را می‌توان در چهار گروه زیر طبقه‌بندی کرد.

1- بانک‌های اطلاعاتی غیررابطه‌ای مبتنی بر کلید - مقدار

در میان انواع مختلف بانک‌های اطلاعاتی غیررابطه‌ای مدل مبتنی بر کلید - مقدار (Key-Value) یکی از ساده‌ترین‌ها به شمار می‌رود، به طوری که توسعه‌دهندگان به شکل ساده‌ای قادر هستند از این بانک اطلاعاتی استفاده کنند. کاربر این مدل از بانک‌های اطلاعاتی قادر است یک مقدار را بر مبنای کلید خاصی دریافت کرده، یک مقدار را به کلیدی اختصاص داده یا کلیدی را از یک مجموعه داده‌ای حذف کند. مقادیر درون این بانک‌های اطلاعاتی همگی به شکل Blob (آبجکت‌های باینری) ذخیره‌سازی می‌شوند. در نتیجه بانک اطلاعاتی بدون آنکه درباره نوع مقادیر و مفهوم آن‌ها اطلاعی داشته باشد مجموعه‌های داده‌ای را ذخیره‌سازی می‌کند. در واقع این وظیفه برنامه کاربردی است که درک کند چه عنصری با چه نوع داده‌ای در بانک اطلاعاتی ذخیره‌سازی شده است. از آنجا که مدل‌های ذخیره‌سازی کلید - مقدار همواره از تکنیک دسترسی کلید اصلی primary-key استفاده می‌کنند، در نتیجه عملکرد بالایی را ارائه کرده و به راحتی قابلیت گسترش‌پذیری را در خود جای داده‌اند. بانک‌های اطلاعاتی کلید - مقدار از یک جدول هش به منظور ذخیره‌سازی کلیدهای منحصر به فرد (unique key) و اشاره‌گرها (که در بعضی بانک‌های اطلاعاتی به نام شاخص معکوس از آن‌ها نام برده می‌شود که در ارتباط با مقادیر داده‌ای هستند) استفاده می‌کنند. در این مدل بانک‌های اطلاعاتی، هیچ رابطه‌ای از نوع ستون وجود ندارد و از این رو، پیاده‌سازی آن‌ها در یک پروژه به ساده‌ترین شکل امکان‌پذیر خواهد بود. بانک‌های اطلاعاتی مبتنی بر کلید - مقدار کارایی بسیار بالایی داشته و به آسانی بر مبنای نیازهای کسب و کار قابل گسترش هستند. از مهم‌ترین بانک‌های اطلاعاتی این گروه به Riak، MemcacheDB و Redis می‌توان اشاره کرد.

شکل 1-
نمایی
بصری از
سامانه‌ها
ی
NoSQL



موارد استفاده بانک‌های اطلاعاتی غیررابطه‌ای مبتنی بر کلید - مقدار:

این بانک‌های اطلاعاتی عمدتاً در ارتباط با ذخیره‌سازی داده‌های مرتبط با نشست کاربران در زمان ورود به یک سیستم، مدیریت پروفایل‌های فاقد اسکیمای کاربران، ذخیره‌سازی تنظیمات حساب کاربری و ذخیره‌سازی داده‌های مربوط به ثبت خرید کاربران استفاده می‌شود. همچنین، به این نکته توجه داشته باشید که بانک‌های اطلاعاتی کلید - مقدار را نمی‌توان در ارتباط با سناریوهای مختلف مورد استفاده قرار داد. به عبارت دقیق‌تر در ارتباط با پروژه‌های زیر بهتر است از این مدل بانک‌های اطلاعاتی استفاده نشود.

زمانی که در نظر داریم یک محاوره را در بانک اطلاعاتی بر اساس مقدار داده‌ای مشخصی وارد کنیم. زمانی که به دنبال رابطه‌ای میان مقادیر داده‌ای هستیم. نیاز داریم عملیات را روی کلیدهای چندگانه منحصر به فرد پیاده‌سازی کنیم. کسب و کار ما نیاز دارد تا به‌طور مستمر بخش‌هایی از داده‌ها را به‌روزرسانی کند.

2- بانک‌های اطلاعاتی غیررابطه‌ای مبتنی بر سند

این مدل بانک‌های اطلاعاتی از الگویی مشابه با بانک‌های اطلاعاتی کلید - مقدار استفاده می‌کنند. به عبارت دیگر این بانک‌های اطلاعاتی نیز از جفت کلید - مقدار استفاده می‌کنند. داده‌ها نیز در قالب مقادیر در این بانک‌های اطلاعاتی ذخیره‌سازی می‌شوند. کلید به کار گرفته شده در این بانک اطلاعاتی یک شناسه منحصر به فرد است. اما تفاوت این بانک اطلاعاتی با مدل کلید - مقدار در ارتباط با مدل ذخیره‌سازی اطلاعات است. به طوری که اطلاعات مبتنی بر سند ذخیره‌سازی می‌شوند. به عبارت دقیق‌تر، مقادیر ذخیره شده در این بانک اطلاعاتی شامل اطلاعات ساخت یافته و نیمه ساخت یافته هستند. این داده‌های ساخت یافته و نیمه ساخت یافته در قالب یک سند شناخته شده و قادر هستند در فرمت‌های JSON، BSON یا XML ذخیره‌سازی شوند. از جمله بانک‌های اطلاعاتی غیررابطه‌ای که مبتنی بر سند کار می‌کنند به MongoDB، Apache CouchDB و Elasticsearch می‌توان اشاره کرد.



داده‌های بزرگ؛ فردای بزرگتر
دنیایی که در آن داده‌ها حرف اول و آخر را می‌زنند

موارد استفاده بانک‌های اطلاعاتی غیررابطه‌ای مبتنی بر سند:

سکوه‌های مبتنی بر تجارت الکترونیک، سامانه‌های مدیریت محتوا، سکوه‌های تجزیه و تحلیل، سکوه‌های وبلاگ‌نویسی از اصلی‌ترین مشتریان این مدل بانک‌های اطلاعاتی غیررابطه‌ای هستند. همچنین، به این نکته توجه داشته باشید که اگر به دنبال اجرای محاوره‌های جست‌وجوی پیچیده روی یک بانک اطلاعاتی هستید یا اگر برنامه کاربردی شما به تراکنش‌های چندگانه پیچیده نیاز دارد، بهتر است به سراغ بانک‌های اطلاعاتی غیررابطه‌ای مبتنی بر سند نروید.



3- بانک‌های اطلاعاتی غیررابطه‌ای مبتنی بر ستون

در بانک‌های اطلاعاتی غیررابطه‌ای مبتنی بر ستون که به نام بانک‌های اطلاعاتی ستون‌محور نیز معروف هستند، داده‌ها به جای آنکه در قالب ردیف‌ها و سلول‌هایی ذخیره‌سازی شوند، در قالب چند ستون با یکدیگر گروه‌بندی و ذخیره‌سازی شده‌اند. ستون‌های مرتبط با یکدیگر می‌توانند شامل تعداد تقریباً نامحدودی از ستون‌ها شوند که ممکن است در زمان اجرا یا در زمان تعریف اسکیمای ساخته شوند. در این بانک‌های اطلاعاتی عملیات مربوط به خواندن و نوشتن داده‌ها به جای آنکه بر مبنای سطرها انجام شود، بر مبنای ستون‌ها انجام می‌شود. در این بانک‌های اطلاعاتی ستون‌هایی که داده‌های مرتبط با یکدیگر دارند معمولاً همراه با هم مورد دسترسی قرار می‌گیرند. به طور مثال، ما اغلب ستون‌های مربوط به نام و مشخصات پروفایل مشتریان را در آن واحد و یکجا با هم فراخوانی می‌کنیم، اما در بعضی مواقع همچون زمان سفارش کالا نیازی نداریم تا همه این اطلاعات به یک باره فراخوانی شوند.

فناوری NoSQL به‌ویژه برای سازمان‌هایی مناسب است که در نظر دارند به حجم بسیار زیادی از داده‌های فاقد ساختار دسترسی پیدا کرده و آن‌ها را تجزیه و تحلیل کنند

مزیت اصلی ذخیره‌سازی داده‌ها در ستون در مقایسه با بانک‌های اطلاعاتی DBMS در این است که عملیاتی همچون جست‌وجو و دستیابی سریع به داده‌های تجمعی با سرعت بالایی انجام می‌شود. بانک‌های اطلاعاتی رابطه‌ای یک ردیف از داده‌ها را به‌عنوان یک ورودی به هم پیوسته ذخیره‌سازی می‌کنند. در این بانک‌ها ردیف‌های مختلف در مکان‌های مختلفی از دیسک ذخیره‌سازی می‌شوند، در حالی که در بانک‌های اطلاعاتی مبتنی بر ستون همه سلول‌های مربوط به یک ستون در قالب یک ورودی پیوسته ذخیره‌سازی می‌شوند.

رویکردی که سرعت دسترسی به داده‌ها و همچنین جست‌وجوها را افزایش می‌دهد. در این بانک‌های اطلاعاتی هر گروه از ستون‌ها را می‌توان با مجموعه ردیف‌های موجود در جداول بانک‌های اطلاعاتی رابطه‌ای مقایسه کرد. جایی که یک کلید مشخص‌کننده یک سطر و یک سطر شامل چندین ستون است. با این تفاوت که ردیف‌های متفاوت لزومی ندارند که حتماً ستون‌های یکسانی را داشته باشند و ستون‌ها را می‌توان زمانی که نیاز است به هر ردیفی اضافه کرد. بدون آنکه ضرورتی داشته باشد که آن‌ها را به ردیف‌های دیگری اضافه کنیم.

مطلب پیشنهادی



بزرگ داده‌ها بازیگر اصلی فناوری‌های فردا
بزرگ داده‌ها چه هستند و چرا باید به آن‌ها اهمیت دهیم؟

موارد استفاده بانک‌های اطلاعاتی غیررابطه‌ای مبتنی بر ستون:

سامانه‌های مدیریت محتوا، سکوه‌های وبلاگ‌نویسی، سرویس‌هایی که تاریخ مصرف مشخصی دارند، سامانه‌هایی که به نوشتن محاوره‌های بسیار سنگین نیاز دارند، همانند سامانه‌های مربوط به ثبت و جمع‌آوری اطلاعات از مشتریان اصلی بانک‌های اطلاعاتی مبتنی بر ستون هستند. همچنین، به این موضوع توجه داشته باشید که اگر از محاوره‌های پیچیده یا الگوهای محاوره‌ای که دائم در حال تغییر هستند استفاده می‌کنید بهتر است به سراغ بانک‌های اطلاعاتی مبتنی بر ستون نروید. اگر درباره ملزومات بانک اطلاعاتی خود اطلاع درستی در اختیار ندارید، بهتر است از این مدل بانک‌های اطلاعاتی استفاده نکنید.

از جمله بانک‌های اطلاعاتی غیررابطه‌ای در این زمینه می‌توان به Apache Hadoop Hbase و Cassandra اشاره کرد.

4- بانک‌های اطلاعاتی غیررابطه‌ای مبتنی بر گراف

بانک‌های اطلاعاتی مبتنی بر گراف اساساً بر مبنای مدل موجودیت - ویژگی - مقدار (Entity-Attribute-Value) ساخته می‌شوند. موجودیت‌ها همچنین به نام ند/گره یا نقطه (Node) نیز شناخته می‌شوند که هر یک خصوصیات خاص خود را دارند. الگویی که همراه با این مدل بانک‌های اطلاعاتی ارائه شد، یکی از انعطاف‌پذیرترین راهکارها برای توصیف اینکه داده‌ها چگونه با داده‌های دیگر در ارتباط هستند را معرفی کرد. در شرایطی که در بانک‌های اطلاعاتی سنتی توضیحات متعلق به هر رابطه در قالب یک فیلد حاوی کلید خارجی یا جداول میانی ذخیره‌سازی می‌شوند، در مقابل بانک‌های اطلاعاتی مبتنی بر گراف به توسعه‌دهندگان این توانایی را می‌دهند تا به شکل مجازی هر رابطه را در هر لحظه تعریف کنند. در بانک‌های اطلاعاتی مبتنی بر گراف، ندها داده‌های مربوط به هر موجودیتی که در بانک اطلاعاتی قرار دارد را ذخیره‌سازی می‌کنند. از جمله بانک‌های اطلاعاتی مبتنی بر گراف‌ها به Neo4j، ArangoDB و OrientDB می‌توان اشاره کرد.

موارد استفاده بانک‌های اطلاعاتی غیررابطه‌ای مبتنی بر گراف:

سامانه‌های تشخیص کلاه‌برداری، جست‌وجوی مبتنی بر گراف، عملیات مربوط به شبکه، شبکه‌های اجتماعی و... از مشتریان اصلی این بانک‌های اطلاعاتی هستند.

نشانی منبع:

<https://www.shabakeh-mag.com/workshop/11095/%D8%A8%D8%A7-%D8%A7%D9%86%D9%88%D8%A7%D8%B9-%D9%85%D8%AE%D8%AA%D9%84%D9%81-%D8%A8%D8%A7%D9%86%DA%A9%E2%80%8C%D9%87%D8%A7%DB%8C-%D8%A7%D8%B7%D9%84%D8%A7%D8%B9%D8%A7%D8%AA%DB%8C-nosql-%D8%A2%D8%B4%D9%86%D8%A7-%D8%B4%D9%88%DB%8C%D8%AF>