**پیوستن به جنبش NoSQL**

این فصل به دو بخش تقسیم می شود: شروع نظری و پایان عملی.

■ در قسمت اول این فصل به طور کلی به پایگاه داده های NoSQL می پردازیم

و به این سوالات پاسخ دهید: چرا آنها وجود دارند؟ چرا تا همین اواخر نه؟ چی

انواع وجود دارد و چرا باید اهمیت دهید؟

■ در بخش دوم، با استفاده از داده‌های رایگان در دسترس، پایتون و پایگاه داده NoSQL، به یک مشکل واقعی – تشخیص بیماری و نمایه‌سازی – می‌پردازیم.

بدون شک در مورد پایگاه های داده NoSQL و نحوه استفاده مذهبی از آنها شنیده اید

بسیاری از شرکت های با تکنولوژی بالا اما پایگاه های داده NoSQL چیست و چه چیزی آنها را چنین می کند

متفاوت از پایگاه داده های رابطه ای یا SQL که به آن ها عادت دارید؟ NoSQL مخفف Not است

فقط زبان پرس و جوی ساختاریافته، اما اگر چه درست است که پایگاه های داده NoSQL می توانند اجازه دهند

شما برای پرس و جو از آنها با SQL، لازم نیست روی نام واقعی تمرکز کنید. بحث زیاد

در حال حاضر بر سر نام و اینکه آیا این گروه از پایگاه های داده جدید باید حتی

اصلا اسم جمعی داشته باشی در عوض، بیایید ببینیم که آنها در مقابل چه چیزی را نشان می دهند

سیستم های مدیریت پایگاه داده رابطه ای (RDBMS). پایگاه های داده سنتی در یک واحد قرار دارند

کامپیوتر یا سرور تا زمانی که داده‌های شما از شما بیشتر نشده باشد، این کار خوب بود

سرور، اما برای بسیاری از شرکت ها برای مدت طولانی اینطور نبوده است. با

رشد اینترنت، شرکت هایی مانند گوگل و آمازون احساس کردند که نگه داشته شده اند

به این پایگاه‌های داده تک‌گره برگشته و به دنبال جایگزین‌هایی گشته است.

شرکت های متعددی از پایگاه داده های NoSQL تک نود مانند MongoDB استفاده می کنند زیرا

آنها طرحواره انعطاف پذیر یا توانایی تجمیع داده ها را به صورت سلسله مراتبی می خواهند. اینجا هستند

چندین مثال اولیه:

■ اولین راه حل NoSQL گوگل Google BigTable بود که شروع آن را نشان داد

پایگاه داده های ستونی.1

■ آمازون با Dynamo، یک فروشگاه با ارزش کلیدی 0.2 آمد

■ دو نوع پایگاه داده دیگر در تلاش برای پارتیشن بندی ظاهر شد: سند

ذخیره و پایگاه داده گراف

در ادامه این فصل به جزئیات هر یک از این چهار نوع خواهیم پرداخت.

لطفاً توجه داشته باشید که اگرچه اندازه فاکتور مهمی بود، اما این پایگاه‌های داده این کار را نکردند

صرفاً از نیاز به مدیریت حجم بیشتری از داده ها سرچشمه می گیرد. هر V از داده های بزرگ

تأثیر دارد (حجم، تنوع، سرعت و گاهی صحت). پایگاه داده های گراف، برای

به عنوان مثال، می تواند داده های شبکه را مدیریت کند. علاقه مندان به پایگاه داده گراف حتی ادعا می کنند که هر چیز را می توان به عنوان یک شبکه دید. مثلا شام را چگونه تهیه می کنید؟ با مواد اولیه این مواد در کنار هم قرار می گیرند تا ظرف را تشکیل دهند و می توان از آنها استفاده کرد

همراه با مواد دیگر برای تشکیل ظروف دیگر. از این منظر،

مواد و دستور العمل ها بخشی از یک شبکه هستند. اما دستور العمل ها و مواد تشکیل دهنده نیز می توانند

در پایگاه داده رابطه ای یا یک فروشگاه اسناد ذخیره شود. همه چیز این است که شما به آن نگاه می کنید

مسئله. نقطه قوت NoSQL در اینجا نهفته است: توانایی نگاه کردن به یک مشکل از a

زاویه های مختلف، شکل دادن به ساختار داده ها به مورد استفاده. به عنوان یک دانشمند داده، شغل شماست

یافتن بهترین پاسخ برای هر مشکلی است. اگرچه گاهی اوقات این هنوز آسان تر است

با استفاده از RDBMS، اغلب یک پایگاه داده NoSQL خاص رویکرد بهتری را ارائه می دهد.

آیا پایگاه داده های رابطه ای محکوم به ناپدید شدن در شرکت هایی با داده های بزرگ هستند زیرا

از نیاز به پارتیشن بندی؟ خیر، پلتفرم های NewSQL (با NoSQL اشتباه نشود)

پاسخ RDBMS به نیاز به راه اندازی خوشه هستند. پایگاه داده های NewSQL از مدل رابطه ای پیروی می کنند اما می توانند به یک خوشه توزیع شده مانند NoSQL تقسیم شوند.

پایگاه های داده این پایان پایگاه‌های داده رابطه‌ای نیست و مطمئناً پایان SQL نیست

پلتفرم هایی مانند Hive SQL را به MapReduce job برای Hadoop ترجمه می کنند. علاوه بر این، نه هر

شرکت به داده های بزرگ نیاز دارد. بسیاری از آنها با پایگاه داده های کوچک به خوبی عمل می کنند و پایگاه های داده سنتی ارتباطی برای آن عالی هستند.

اگر به نقشه ذهنی کلان داده نشان داده شده در شکل 6.1 نگاه کنید، چهار نوع از آن را خواهید دید

این چهار نوع عبارتند از: ذخیره اسناد، ذخیره کلید-مقدار، پایگاه داده گراف و پایگاه داده ستونی. نقشه ذهنی همچنین شامل پایگاه داده های رابطه ای پارتیشن بندی شده NewSQL است. در

در آینده این شکاف بزرگ بین NoSQL و NewSQL منسوخ خواهد شد زیرا هر

نوع پایگاه داده تمرکز خاص خود را خواهد داشت، در حالی که عناصر هر دو NoSQL و

پایگاه های داده NewSQL خطوط به آرامی محو می شوند زیرا انواع RDBMS ویژگی های NoSQL را دریافت می کنند

مانند نمایه سازی ستون گرا که در پایگاه داده های ستونی دیده می شود. اما در حال حاضر این یک است

راه خوبی برای نشان دادن اینکه پایگاه داده های رابطه ای قدیمی از تک گره خود عبور کرده اند

راه اندازی، در حالی که سایر انواع پایگاه داده تحت مخرج NoSQL در حال ظهور هستند

**مقدمه ای بر NoSQL**

همانطور که خواندید، هدف پایگاه های داده NoSQL تنها ارائه راهی برای پارتیشن بندی موفقیت آمیز پایگاه های داده بر روی چندین گره نیست، بلکه ارائه اساساً متفاوت است.

روش‌هایی برای مدل‌سازی داده‌های موجود به‌منظور تطبیق ساختار آن با مورد استفاده آن و نه اینکه چگونه یک پایگاه داده رابطه‌ای نیاز به مدل‌سازی دارد.

برای کمک به درک NoSQL، با نگاهی به هسته ACID شروع می کنیم

اصول پایگاه داده های رابطه ای تک سرور و نشان می دهد که چگونه پایگاه های داده NoSQL بازنویسی می شوند

آنها را در اصول BASE قرار می دهند تا به شکلی توزیع شده بسیار بهتر کار کنند. ما نیز خواهیم کرد

به قضیه CAP نگاه کنید، که مشکل اصلی توزیع پایگاه های داده در چندین گره و نحوه برخورد پایگاه های داده ACID و BASE را توضیح می دهد.

**ACID: اصل اصلی پایگاه های داده رابطه ای**

جنبه های اصلی یک پایگاه داده سنتی رابطه ای را می توان با مفهوم ACID خلاصه کرد:

■ اتمی - اصل "همه یا هیچ". اگر یک رکورد در پایگاه داده قرار داده شود، این است

به طور کامل وارد کنید یا اصلا به عنوان مثال، اگر قطع برق در برق رخ دهد

در وسط یک اقدام نوشتن پایگاه داده، در نهایت نصف رکورد نخواهید داشت. آی تی

اصلا آنجا نخواهد بود

■ سازگاری - این اصل مهم یکپارچگی داده ها را حفظ می کند. خیر

ورودی که آن را در پایگاه داده ایجاد می کند همیشه با از پیش تعریف شده در تضاد خواهد بود

قوانینی مانند نداشتن فیلد الزامی یا عددی بودن فیلد به جای متن.

■ جداسازی — وقتی چیزی در پایگاه داده تغییر می کند، هیچ اتفاقی نمی تواند روی آن بیفتد

این داده دقیقاً در همان لحظه. در عوض، اعمال اتفاق می افتد

در سریال با تغییرات دیگر انزوا مقیاسی است که از انزوای کم به بالا می رود

انزوا در این مقیاس، پایگاه های داده سنتی در انتهای "ایزوله بالا" قرار دارند.

نمونه ای از انزوا کم می تواند Google Docs باشد: چند نفر می توانند بنویسند

به یک سند دقیقاً در همان زمان و مشاهده تغییرات یکدیگر

فورا. یک سند Word سنتی، در انتهای دیگر طیف، دارای

انزوا بالا؛ برای ویرایش توسط اولین کاربری که آن را باز می کند قفل شده است. دومین نفری که سند را باز می کند می تواند آخرین نسخه ذخیره شده آن را ببیند اما نمی تواند ببیند

تغییرات ذخیره نشده یا ویرایش سند بدون اینکه ابتدا آن را به عنوان کپی ذخیره کنید. پس یکبار

کسی آن را باز کرده است، به روزترین نسخه کاملاً از آن جدا شده است

هر کسی جز ویرایشگری که سند را قفل کرده است.

■ دوام - اگر داده ها وارد پایگاه داده شده باشند، باید به طور دائم زنده بمانند.

آسیب فیزیکی به هارد دیسک ها رکوردها را از بین می برد، اما قطعی برق و

خرابی نرم افزار نباید

ACID برای همه پایگاه‌های اطلاعاتی رابطه‌ای و پایگاه‌های داده NoSQL خاص، مانند

پایگاه داده گراف Neo4j. ما در ادامه این فصل در مورد پایگاه داده های نمودار بیشتر بحث خواهیم کرد

در فصل 7. برای اکثر پایگاه های داده NoSQL دیگر اصل دیگری اعمال می شود: BASE. به

درک BASE و اینکه چرا در اکثر پایگاه های داده NoSQL اعمال می شود، باید به آن نگاه کنیم

قضیه CAP

**قضیه CAP: مشکل DB ها در بسیاری از گره ها**

هنگامی که یک پایگاه داده روی سرورهای مختلف پخش می شود، پیروی از ACID دشوار است

اصل به دلیل قوام ACID وعده می دهد. قضیه CAP دلیل آن را نشان می دهد

این مشکل ساز می شود قضیه CAP بیان می کند که یک پایگاه داده می تواند هر دو باشد

موارد زیر اما هرگز هر سه:

■ تحمل پارتیشن - پایگاه داده می تواند پارتیشن شبکه یا خرابی شبکه را مدیریت کند.

■ در دسترس - تا زمانی که گره‌ای که به آن متصل می‌شوید فعال است و شما

می تواند به آن متصل شود، گره پاسخ خواهد داد، حتی اگر ارتباط بین

گره های مختلف پایگاه داده از بین می روند.

■ سازگار - مهم نیست به کدام گره متصل می شوید، همیشه دقیقاً آن را خواهید دید

همان داده ها

برای یک پایگاه داده تک گره به راحتی می توان مشاهده کرد که چگونه همیشه در دسترس و سازگار است:

■ در دسترس - تا زمانی که گره بالا است، در دسترس است. این تمام وعده های در دسترس بودن CAP است.

■ سازگار - گره دومی وجود ندارد، بنابراین هیچ چیز نمی تواند ناسازگار باشد.

وقتی پایگاه داده تقسیم می شود، همه چیز جالب می شود. سپس باید از بین آنها یکی را انتخاب کنید

در دسترس بودن و سازگاری، همانطور که در شکل 6.2 نشان داده شده است.

بیایید مثالی از یک فروشگاه آنلاین با a

سرور در اروپا و سرور در ایالات متحده،

با یک مرکز توزیع واحد یک آلمانی به نام

فریتز و یک آمریکایی به نام فردی در حال خرید هستند

در همان زمان در همان فروشگاه آنلاین. آنها

یک مورد را ببینید و فقط یک مورد هنوز در انبار موجود است: یک برنز،

میز قهوه اختاپوس شکل. فاجعه رخ می دهد، و

ارتباط بین دو سرور محلی است

به طور موقت خاموش است اگر شما صاحب آن بودید

خرید کنید، دو گزینه دارید:

■ در دسترس بودن — به سرورها اجازه می‌دهید که فعال باشند

خدمات رسانی به مشتریان، و شما همه چیز را مرتب می کنید

پس از آن

■ ثبات—شما تمام فروش ها را تا زمانی که به حالت تعلیق در می آیید

ارتباط دوباره برقرار می شود

در مورد اول، فریتز و فردی هر دو میز قهوه اختاپوس را می خرند، زیرا

آخرین شماره سهام شناخته شده برای هر دو گره "یک" است و هر دو گره مجاز به فروش آن هستند،

همانطور که در شکل 6.3 نشان داده شده است.

اگر دسترسی به میز قهوه سخت است، باید به فریتز یا فردی اطلاع دهید

که او میز خود را در تاریخ تحویل وعده داده شده دریافت نخواهد کرد یا حتی بدتر از آن، خواهد گرفت

هرگز آن را دریافت نکنید به عنوان یک تاجر خوب، ممکن است یکی از آنها را با یک جبران کنید

کوپن تخفیف برای خرید بعدی، و ممکن است همه چیز درست باشد.

گزینه دوم (شکل 6.4) شامل قرار دادن درخواست های دریافتی در انتظار است

به طور موقت

این ممکن است برای فریتز و فردی منصفانه باشد اگر بعد از پنج دقیقه فروشگاه اینترنتی راه اندازی شود

دوباره برای تجارت باز شوید، اما ممکن است هر دو فروش و احتمالاً بسیاری دیگر را از دست بدهید.

فروشگاه های وب تمایل دارند در دسترس بودن را به یکپارچگی انتخاب کنند، اما این انتخاب بهینه نیست

در تمام موارد از یک جشنواره محبوب مانند Tomorrowland استفاده کنید. جشنواره ها تمایل به داشتن یک

حداکثر ظرفیت مجاز به دلایل ایمنی اگر بیشتر از آنچه هستید بلیط بفروشید

مجاز است زیرا سرورهای شما در طول یک شکست ارتباطی گره به فروش خود ادامه دادند،

شما می توانید دو برابر تعداد مجاز تا زمان برقراری مجدد ارتباطات بفروشید. در چنین حالتی ممکن است عاقلانه تر باشد که به دنبال سازگاری و خاموش کردن گره ها باشید

به طور موقت جشنواره ای مانند Tomorrowland در چند ساعت اول فروخته می شود

به هر حال، پس اندکی از کار افتادگی به اندازه نیاز به برداشت هزاران صدمه نخواهد بود

**اصول پایه پایگاه های داده NoSQL**

RDBMS از اصول ACID پیروی می کند. پایگاه های داده NoSQL که از ACID پیروی نمی کنند، مانند

ذخیره‌سازی اسناد و ذخیره‌های ارزش کلیدی، BASE را دنبال کنید. BASE مجموعه ای از بسیار نرم تر است

پایگاه داده وعده می دهد:

■ اساساً در دسترس - در دسترس بودن به مفهوم CAP تضمین شده است. گرفتن وب

به عنوان مثال، اگر یک گره راه‌اندازی و در حال اجرا باشد، می‌توانید به خرید ادامه دهید. بسته به نحوه تنظیم چیزها، گره ها می توانند از گره های دیگر استفاده کنند. برای مثال جستجوی الاستیک یک موتور جستجوی نوع سند NoSQL است که تقسیم می‌کند

و داده های خود را به گونه ای تکرار می کند که لزوماً به معنای خرابی گره نیست

خرابی سرویس، از طریق فرآیند به اشتراک گذاری. هر قطعه را می توان به عنوان یک نمونه سرور پایگاه داده فردی مشاهده کرد، اما همچنین قادر به برقراری ارتباط با

سایر قطعات برای تقسیم حجم کار تا حد امکان موثر (شکل 6.5). چند خرده می تواند در یک گره وجود داشته باشد. اگر هر خرده یک ماکت داشته باشد

گره دیگر، شکست گره به راحتی با تقسیم مجدد کار بین آنها برطرف می شود

گره های باقی مانده

■ حالت نرم — وضعیت یک سیستم ممکن است در طول زمان تغییر کند. این مربوط به

اصل سازگاری نهایی: سیستم ممکن است برای ایجاد داده ها تغییر کند

دوباره سازگار در یک گره داده ها ممکن است "A" و در گره دیگر ممکن است

بگویید "B" زیرا اقتباس شده بود. بعدها، در حل تعارض زمانی که شبکه است

به صورت آنلاین، ممکن است "A" در اولین گره با "B" جایگزین شود. زوج

اگرچه هیچ کس کاری برای تغییر صریح "A" به "B" انجام نداد، اما این کار را انجام می دهد

مقدار آن زمانی که با گره دیگر سازگار می شود.

■ سازگاری نهایی - پایگاه داده در طول زمان ثابت می شود. در وب

به عنوان مثال، جدول دو بار فروخته می شود که منجر به ناهماهنگی داده ها می شود. یک بار

ارتباط بین گره های جداگانه برقرار می شود، آنها ارتباط برقرار می کنند و تصمیم می گیرند که چگونه آن را حل کنند. این تعارض می تواند حل شود، برای مثال، در

بر اساس اولویت، اولین خدمت یا با ترجیح دادن مشتری که متحمل می شود

کمترین هزینه حمل و نقل پایگاه داده ها دارای رفتار پیش فرض هستند، اما با توجه به آن

در اینجا باید یک تصمیم واقعی تجاری گرفت، این رفتار را می توان 10 بازنویسی کرد. حتی اگر اتصال برقرار باشد و در حال اجرا باشد، تأخیر ممکن است باعث ایجاد گره ها شود

ناسازگار شود اغلب، محصولات در یک سبد خرید آنلاین نگهداری می شوند،

اما قرار دادن یک کالا در یک سبد آن را برای سایر کاربران قفل نمی کند. اگر فریتز بزند

فردی به دکمه تسویه حساب، وقتی فردی به آن مراجعه کند مشکلی پیش خواهد آمد

وارسی. این را می توان به راحتی برای مشتری توضیح داد: او خیلی دیر کرده بود. ولی

چه می شود اگر هر دو دکمه پرداخت را دقیقاً در یک میلی ثانیه فشار دهند و هر دو

فروش اتفاق می افتد؟

### مرحله 3 اصلاح شده: آماده سازی داده ها برای پروفایل بیماری

نباید تعجب آور باشد که به آماده سازی داده ها بازگشته اید، همانطور که در شکل نشان داده شده است. پس از همه، فرآیند علم داده یک فرآیند تکراری است

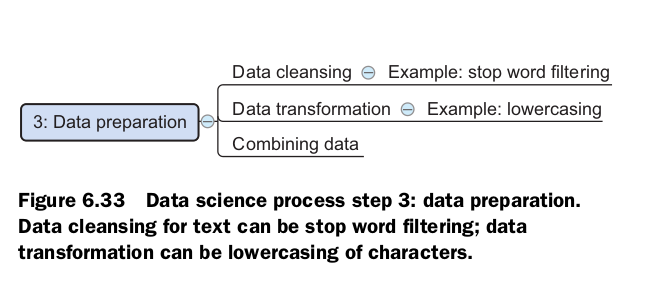
داده های خود را، شما عملاً هیچ پاکسازی یا تغییر داده ای انجام ندادید. شما می توانید اضافه کنید

برای مثال، پاکسازی داده‌ها اکنون با توقف فیلتر کردن کلمات. کلمات توقف کلماتی هستند که هستند

آنقدر رایج است که اغلب دور انداخته می شوند زیرا می توانند نتایج را آلوده کنند. ما

در اینجا به توقف فیلتر کلمه (یا سایر پاکسازی داده ها) نمی پردازیم، اما خودتان آن را امتحان کنید.

برای نمایه سازی بیگرام ها باید فیلتر رمز و تحلیلگر متن خود را ایجاد کنید.



فیلتر توکن قادر است تغییراتی را روی توکن ها ایجاد کند. فیلتر توکن خاص شما نیاز به ترکیب توکن‌ها برای ایجاد n گرم دارد که به آن زونا نیز می‌گویند. الاستیک پیش فرض

توکنایزر جستجو، توکنایزر استاندارد نامیده می شود و به دنبال مرزهای کلمه می گردد.

مانند فاصله بین کلمات، برای برش متن به نشانه ها یا اصطلاحات مختلف. گرفتن a

همانطور که در فهرست زیر نشان داده شده است، به تنظیمات جدید برای شاخص بیماری خود نگاه کنید.



شما دو عنصر جدید ایجاد می کنید: فیلتر رمز به نام "فیلتر درخشش من" و یک فیلتر جدید

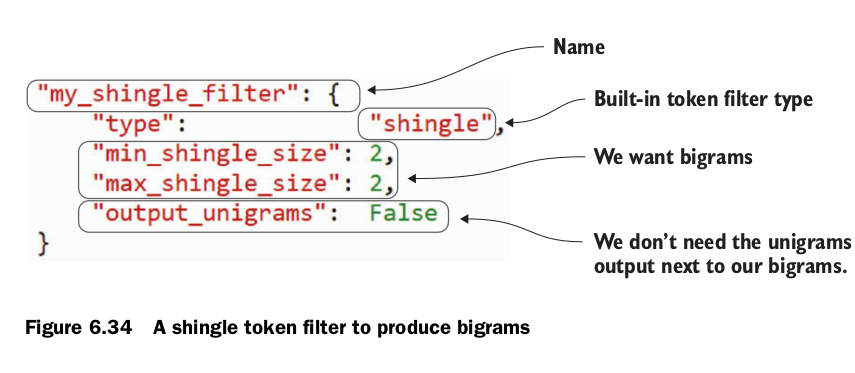
آنالایزر به نام my\_shingle\_analyzer. از آنجا که n-گرم بسیار رایج است، Elastic-

جستجو با یک نوع فیلتر توکن درخشش داخلی ارائه می شود. تنها چیزی که باید به آن بگویید این است که

شما می خواهید بیگرام های "min\_shingle\_size" : 2، "max\_shingle\_size" : 2، همانطور که نشان داده شده است

در شکل 6.34. شما می توانید برای سه گرام و بالاتر بروید، اما برای اهداف نمایشی

این خدمت خواهد کرد.



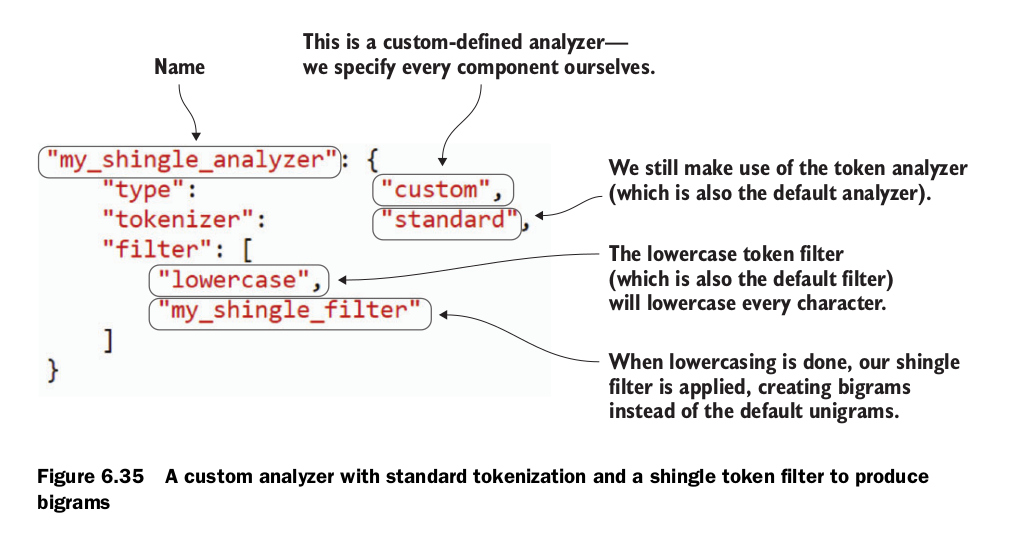
آنالایزر نشان داده شده در شکل 6.35 ترکیبی از تمام عملیات مورد نیاز است

از متن ورودی به نمایه بروید. این فیلتر براق را در خود جای داده است، اما بسیار بیشتر از آن است

این. توکنایزر متن را به نشانه ها یا اصطلاحات تقسیم می کند. سپس می توانید از حروف کوچک استفاده کنید

فیلتر کنید تا هنگام جستجوی "دیابت" در مقابل "دیابت" تفاوتی وجود نداشته باشد. سرانجام،

شما فیلتر درخشندگی خود را اعمال می کنید و بیگرام های خود را ایجاد می کنید.



توجه داشته باشید که قبل از به‌روزرسانی تنظیمات باید فهرست را ببندید. آن وقت می توانید

با اطلاع از اینکه تنظیمات شما به روز شده است، با خیال راحت فهرست را دوباره باز کنید. نه همه تنظیمات

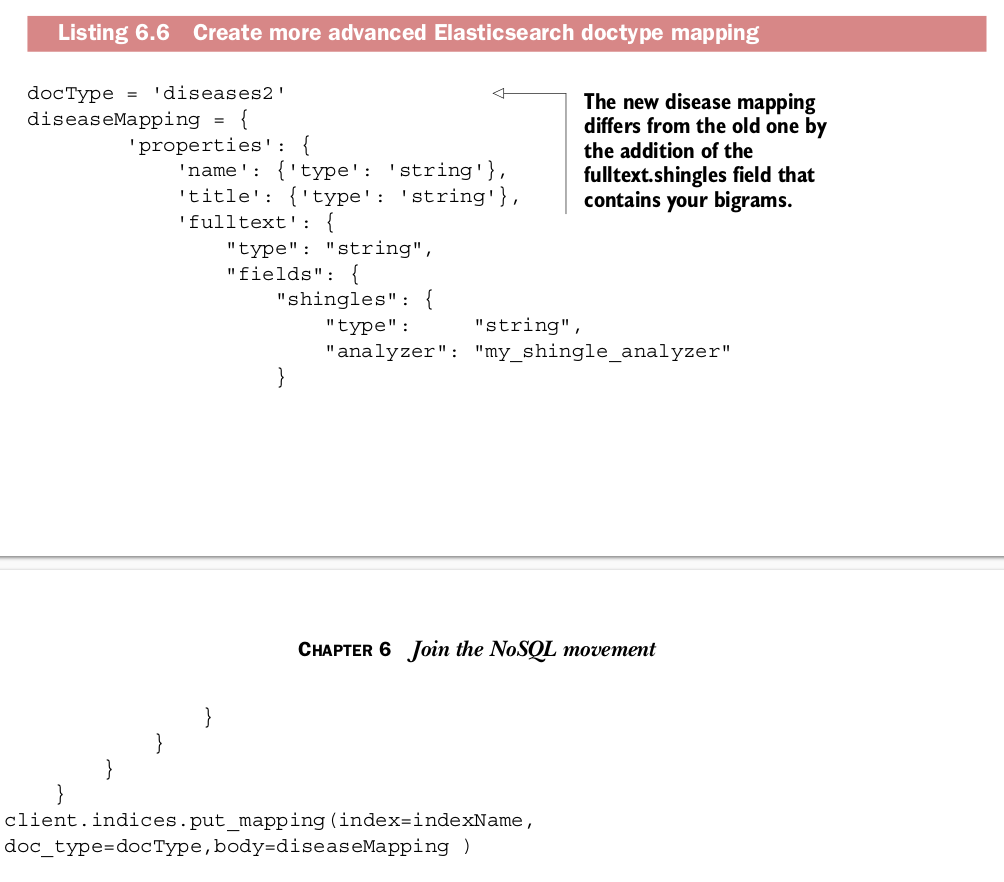
تغییرات نیاز به بسته شدن ایندکس دارند، اما این یکی این کار را می کند. شما می توانید یک نمای کلی از

چه تنظیماتی نیاز به بسته شدن ایندکس در http://www.elastic.co/guide/en/elastic- دارد

search/reference/current/indices-update-settings.html.

ایندکس اکنون برای استفاده از تحلیلگر جدید شما آماده است. برای این شما یک سند جدید ایجاد خواهید کرد

نوع بیماری، بیماری2، با یک نقشه برداری جدید، همانطور که در لیست زیر نشان داده شده است.



در متن کامل، اکنون یک پارامتر اضافی، فیلدها دارید. در اینجا می توانید همه را مشخص کنید

ایزوتوپ های مختلف متن کامل شما فقط یکی دارید؛ به نام شینگل می رود

و متن کامل را با my\_shingle\_analyzer جدید شما تجزیه و تحلیل خواهد کرد. تو هنوز داری

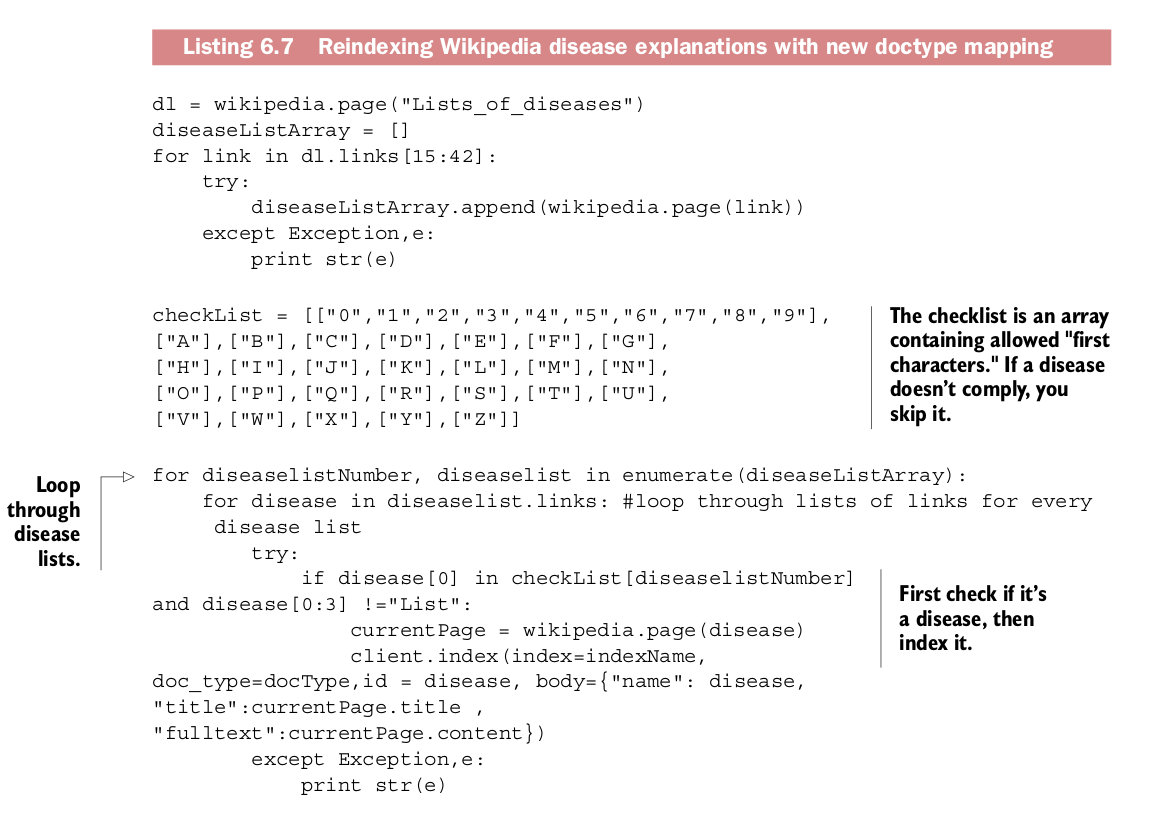
به متن کامل خود دسترسی داشته باشید، و شما یک تحلیلگر برای این کار مشخص نکرده اید، بنابراین

استاندارد مانند قبل استفاده خواهد شد. با دادن ملک می توانید به جدید دسترسی پیدا کنید

نام و به دنبال آن نام فیلد آن: fulltext.shingles. تنها کاری که اکنون باید انجام دهید این است که بروید

مراحل قبلی را طی کنید و داده ها را با استفاده از API ویکی پدیا فهرست کنید، همانطور که در نشان داده شده است

لیست زیر



هیچ چیز جدیدی در اینجا وجود ندارد، فقط این بار به جای آن، بیماری های doc\_type2 را ایندکس می کنید

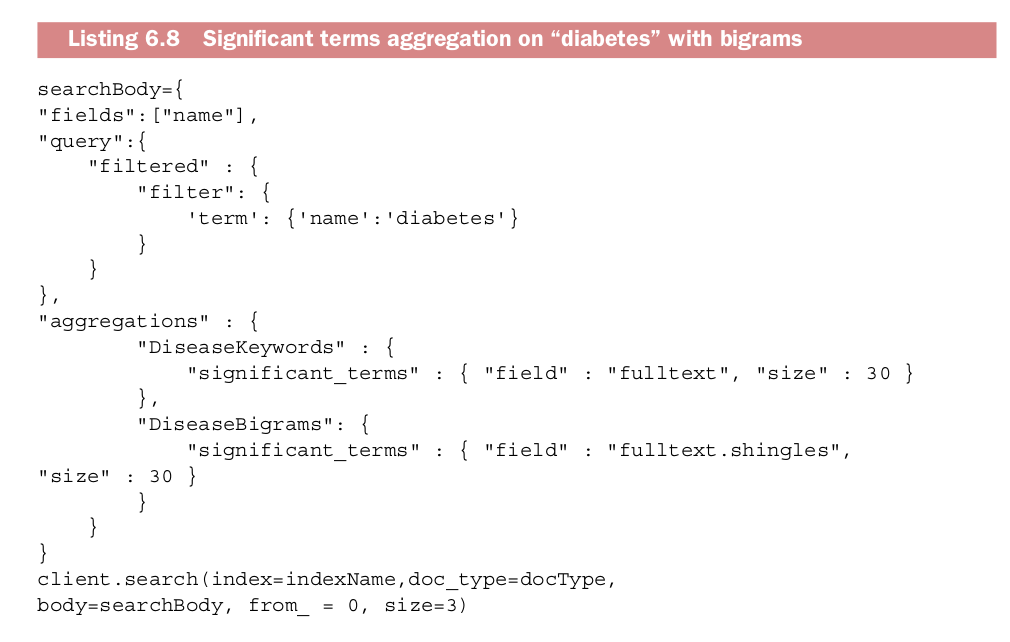
بیماری ها وقتی این کار کامل شد، می توانید دوباره به مرحله 4 بروید، داده ها را کاوش کنید

و نتایج را بررسی کنید.

### مرحله 4 بازبینی شده: کاوش داده برای پروفایل بیماری

شما یک بار دیگر به کاوش داده ها رسیدید. می توانید پرس و جوی تجمعات را تطبیق دهید

و از رشته جدید خود برای ارائه مفاهیم کلیدی مرتبط با دیابت به شما استفاده کنید:



مجموعه جدید شما، به نام DiseaseBigrams، از فیلد fulltext.shingles برای حمایت از

چند بینش جدید در مورد دیابت ارائه دهید. این عبارات کلیدی جدید نشان داده می شوند:

■

■

■

■

ترشح بیش از حد - یک بیمار دیابتی نیاز به ادرار مکرر دارد.

باعث پلی اوری می شود - این نشان دهنده همان چیزی است: دیابت باعث می شود که بیمار این مشکل را ایجاد کند

مکرر ادرار کنید

تست محرومیت - این در واقع یک تریگرام، "تست محرومیت از آب" است، اما تشخیص می دهد:

تست محرومیت nized زیرا شما فقط بیگرام دارید. این یک آزمایش برای تعیین است

آیا بیمار دیابت دارد یا خیر.

تشنگی بیش از حد - شما قبلاً با جستجوی کلمه کلیدی unigram خود "Thirst" را پیدا کرده اید، اما

در آن مرحله از نظر فنی می‌توانست به معنای «بدون تشنگی» باشد.

بیگرام ها، یونیگرام ها و احتمالا تریگرام های جالب دیگری نیز وجود دارند. به عنوان گرفته شده است

به طور کلی، می توان از آنها برای تجزیه و تحلیل یک متن یا مجموعه ای از متون قبل از خواندن استفاده کرد

آنها توجه داشته باشید که بدون رسیدن به مدل سازی به نتایج دلخواه رسیدید

صحنه. گاهی اوقات حداقل مقدار مساوی از اطلاعات ارزشمند وجود دارد

در کاوش داده ها و مدل سازی داده ها یافت می شود. اکنون که به طور کامل به خواسته خود رسیده اید

هدف ثانویه، می توانید به مرحله 6 فرآیند علم داده بروید: ارائه-

عملیات و اتوماسیون

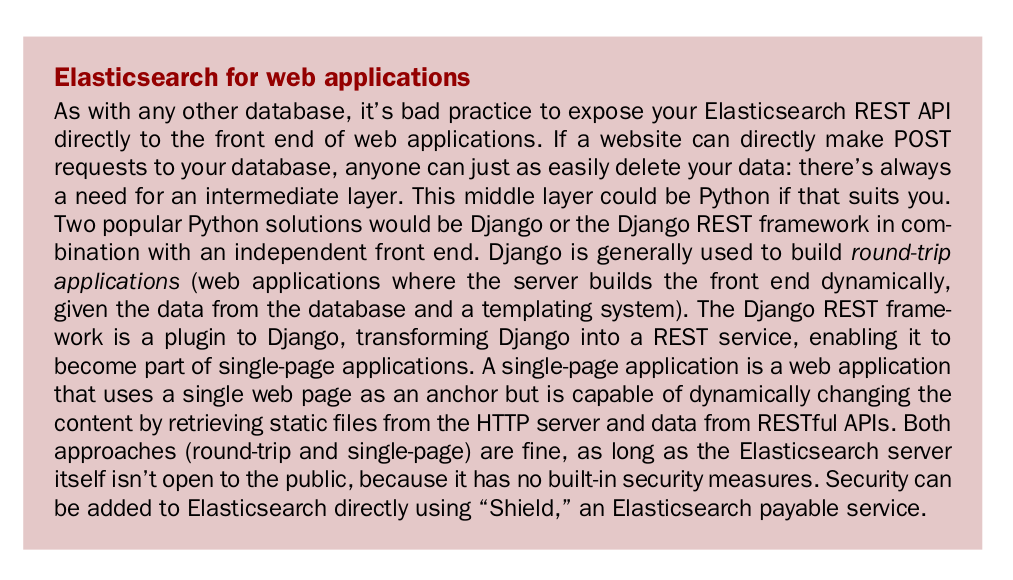
### مرحله 6: ارائه و اتوماسیون

هدف اصلی شما، تشخیص بیماری، به یک ابزار تشخیص سلف سرویس تبدیل شده است

با اجازه دادن به پزشک برای پرس و جو از طریق، به عنوان مثال، یک برنامه وب. شما نمی سازید

یک وب سایت در این مورد، اما اگر قصد انجام این کار را دارید، لطفاً نوار کناری «Elastic-

برنامه های تحت وب را جستجو کنید."



هدف ثانویه، پروفایل بیماری، را نیز می توان به سطح یک کاربر برد

رابط؛ این امکان وجود دارد که به نتایج جستجو اجازه دهید یک ابر کلمه تولید کنند که به صورت بصری

نتایج جستجو را خلاصه می کند. ما در این کتاب آنقدر دور نمی‌شویم، اما اگر اینطور هستید

علاقه مند به راه اندازی چیزی شبیه به این در پایتون، از کتابخانه word\_cloud استفاده کنید

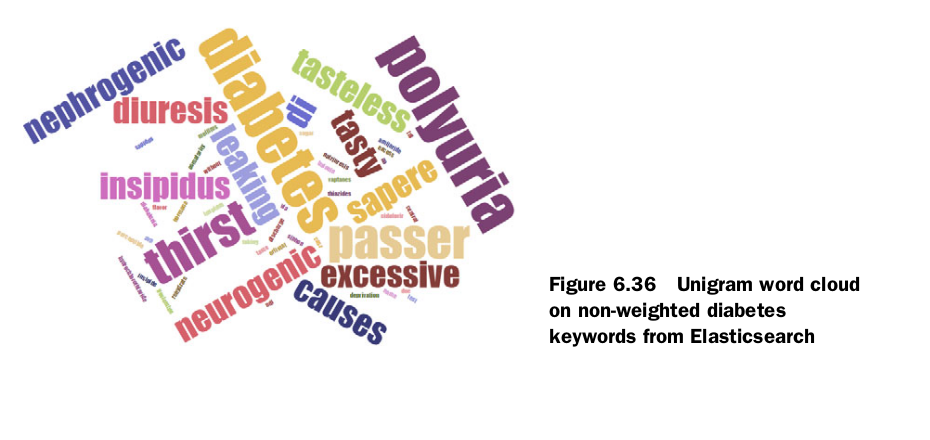
(پیپ نصب word\_cloud). یا اگر جاوا اسکریپت را ترجیح می دهید، D3.js راه خوبی است. شما

می توانید یک نمونه پیاده سازی را در http://www.jasondavies.com/wordcloud/ پیدا کنید

#%2F%2Fwww.jasondavies.com%2Fwordcloud%2Fabout%2F.

افزودن کلمات کلیدی خود در این وب سایت مبتنی بر D3.js یک کلمه یونیگرام تولید می کند

ابر مانند شکل 6.36 که می تواند در ارائه نتایج پروژه شما گنجانده شود. اصطلاحات در این مورد بر اساس امتیاز آنها وزن نمی شوند، اما این است در حال حاضر نمایش خوبی از یافته ها ارائه می دهد.



پیشرفت های زیادی برای برنامه شما امکان پذیر است، به خصوص در زمینه داده

تهیه شده. اما غواصی در تمام احتمالات در اینجا ما را بیش از حد دور می کند. بنابراین ما داریم

به پایان این فصل بیایید. در قسمت بعدی نگاهی به جریان داده خواهیم انداخت.