



بررسی گزینه‌های معماری اطلاعات سازمان‌ها با ظهور داده‌های بزرگ

حسین عباسی مهر، یوسف پورسلیمان صومعه‌دل
استادیار گروه مهندسی فناوری اطلاعات و کامپیوتر، دانشگاه شهید مدنی آذربایجان

^۲کارشناسی مهندسی فناوری اطلاعات، دانشگاه شهید مدنی آذربایجان

چکیده

سازمان‌ها از انبار داده‌ها برای مدیریت و یکپارچه‌سازی داده‌های ساخت‌یافته استفاده می‌کنند. با این وجود، حجم و ساختار داده‌های تولید شده در دنیای امروزی، اصلی‌ترین مسکلی است که توان پردازشی انبار داده‌های سنتی را مورد چالش قرار می‌دهد. چرا که مدل‌های داده‌ای ساخت‌یافته برای پردازش داده‌های غیرساخت‌یافته مناسب نیستند. در این مقاله به معرفی گزینه‌های معماری انبار داده برای پردازش داده‌های ساخت‌یافته و غیرساخت‌یافته پرداختیم. گزینه‌های معماری عبارتند از: انبار داده سنتی، انبار داده سنتی با تغییر در فرایند ETL، انبار داده سنتی (مدل ستاره‌ای مبتنی بر NoSQL)، انبار داده مبتنی بر هادوپ (مبتنی بر ستاره یا کاملاً غیرنرمال شده). همچنین بررسی رویکردهای استفاده شده نشان می‌دهد که دلیل اینکه متدولوژی خاصی برای طراحی انبار داده‌های بزرگ وجود ندارد، طراحان انبار داده، اغلب دانشی که در زمینه طراحی انبار داده سنتی دارند را به کار می‌گیرند. به عبارت دیگر اغلب از مدل داده‌ای ستاره‌ای استفاده می‌کنند که با توجه به ویژگی‌های داده‌های بزرگ ممکن است عملکرد مطلوبی مورد تحقق واقع نشود.

کلمات کلیدی: انبار داده، داده‌های بزرگ، پایگاه داده NoSQL، هادوپ، Hive

ادامه این مقاله به این صورت است که در بخش ۲ به معرفی مختصر مفاهیمی نظیر داده‌های بزرگ، انبار داده، اینترنت اشیا، هادوپ و NoSQL می‌پردازیم. سپس در بخش ۳ گزینه‌های معماری انبار داده با در نظر گرفتن داده‌های غیرساخت یافته معرفی می‌شود. در بخش ۴ هم به نتیجه‌گیری در مورد مقاله می‌پردازیم.

۲- مفاهیم مورد استفاده در تحقیق

در این بخش به معرفی مختصر مفاهیم ذکر شده در طول این مقاله می‌پردازیم.

۲-۱- داده‌های بزرگ

داده‌های بزرگ به داده‌هایی اطلاق می‌شود که از نظر اندازه به حدی بزرگ هستند که با نرم افزارهای متداول در زمان معقولی قابل پردازش و مدیریت نیستند. مشکلات اصلی در کار با این نوع داده‌ها مربوط به برداشت و جمع‌آوری، ذخیره‌سازی، جست‌وجو، اشتراک‌گذاری، تحلیل و نمایش آن‌ها است. بیشتر تحلیل‌های مورد نیاز در پردازش داده‌های عظیم، توسط دانشمندان در علوممانند، به‌اشناسی، ژنتیک، شبیه‌سازی‌های پیچیده فیزیک، جست‌وجوی اینترنت و تحلیل‌های اقتصادی مورد استفاده قرار می‌گیرد [۳].

۲-۲- انبار داده:

بر طبق اینمون و همکاران [۵] به عنوان مخزن داده‌ای یک سازمان، انبار داده دارای ویژگیهای زیر است:

- موضوع‌گرایی
- یکپارچه بودن
- غیر فرار بودن
- متغیر با زمان

که برای پشتیبانی از تصمیم‌گیری در یک سازمان پیاده‌سازی می‌شود.

بر اساس کیمبال [۶] و همکاران [۶]، انبار داده جمعیتی از تمامی دیتامارت‌ها در داخل یک سازمان است. مدل داده‌ای انبار داده‌ها، مدل چندبعدي است. در مدل‌سازی بعدی دو نوع جدول وجود دارد که عبارتند از جداول حقیقت و جداول بعد.

۱- مقدمه:

در طول چند دهه گذشته، فناوری اطلاعات در بخش‌های مختلف سازمان‌ها مورد استفاده قرار گرفته است. استفاده از فناوری اطلاعات باعث شده است تا در سازمان‌ها داده‌های حجیمی در حوزه‌های مختلفی از قبیل تولید، تدارکات، بازاریابی، فروش و غیره تولید شود. با این وجود، در اختیار داشتن داده‌ها اساساً نمی‌تواند برای یک سازمان مزیت محسوب شود. در حقیقت داده‌ها باید با استفاده از تکنیک‌های تحلیل به اطلاعات تبدیل شوند و در نهایت مورد استفاده تصمیم‌گیرندگان سازمانی قرار گیرند.

در سازمان‌ها به دلیل ناهمگن بودن سیستم‌های فناوری اطلاعات، جریان داده وجود ندارد و در حقیقت داده‌های تولید شده در یک سازمان مجزا از هم نگهداری می‌شوند. با توجه به اینکه برای انجام یک تحلیل جامع نیاز به داشتن داده‌های کامل است، برای این منظور انبار داده‌ها بوجود آمدند. نکته مهمی که در این زمینه وجود دارد این است که انبار داده‌ها برای مدیریت و یکپارچه‌سازی داده‌های ساخت‌یافته در یک سازمان مورد استفاده قرار می‌گیرد. [۱،۲] با این وجود، حجم و ساختار داده‌های تولید شده در دنیای امروزی، اصلی‌ترین مشکلی است که توان پردازشی انبار داده‌های سنتی را مورد چالش قرار می‌دهد. چرا که مدل‌های داده‌ای ساخت‌یافته برای پردازش داده‌های غیرساخت‌یافته (داده‌های متنی، تصویر، ویدئو و داده‌های حسگرها) بهینه نیستند. [۳] برای مثال داده‌ها دیگر متمرکز نیستند و منابع داده‌ای صرفاً سیستم‌های OLTP^۱ نیست. بلکه داده‌ها توزیع شده هستند و دارای ساختارهای متفاوت هستند و به صورت نمایی رشد می‌کنند. بنابراین انبار داده مورد نیاز برای داده‌های بزرگ کاملاً متفاوت است از انبار داده سنتی.

در این مقاله به بررسی معماری فناوری اطلاعات سازمان با ورود داده‌های بزرگ و اینترنت اشیا می‌پردازیم. در واقع سوال اصلی این تحقیق این است که با ظهور داده‌های غیرساخت‌یافته از چه ابزارها و رویکردهایی برای طراحی انبار داده‌ها می‌توان استفاده کرد.

موارد و تغییراتی که در انبار داده‌های بزرگ مطرح است:

- قابلیت پردازش داده‌های توزیع شده
- قابلیت مقیاس پذیری با هزینه کمتر
- توانایی تحلیل حجم عظیمی از اطلاعات
- پردازش مصورسازی داده‌ها به صورت به هنگام برای بهبود فرآیند تصمیم سازی
- یکپارچه سازی داده‌هایی با ساختار متفاوت از منابع داخلی و بیرونی
- پشتیبانی از بار پردازشی بسیار سنگین [۷، ۸]

۲-۴- اینترنت اشیا

پیش از اختراع این عبارت، بسیاری از سازندگان دستگاه قبلا قابلیت جمع‌آوری داده‌ها را از طریق دستگاه‌های خود با استفاده از حسگرها و کنترل کننده‌های هوشمند اضافه کرده بودند. برخی از آنها شامل نسخه‌های اولیه نرم‌افزار جمع‌آوری داده‌ها بودند. با این حال، قیمت، اندازه و قابلیت محدود حسگرهای اولیه و کنترل کننده‌ها گاهی اوقات مفید بودن داده‌هایی که می‌توان جمع‌آوری کرد را محدود می‌کرد.

تقاضای رو به رشد برای حسگرها منجر به تحقیق بیشتری برای مینیاتوری کردن و کاهش قابل توجه قیمت حسگرها شد. مینیاتوری سازی تا حدی با کاهش شدید انرژی مورد نیاز برای حسگر فراهم شد.

خوشبختانه، پیشرفت دوم با تکثیر حسگرها و کنترل های هوشمند رخ داد. این پیشرفت به توانایی انواع جدیدتر نرم افزار مدیریت داده‌ها منجر شد تا حجم زیادی از داده‌های در جریان به دست آید و تجزیه و تحلیل شود.

۲-۳- پایگاه داده NoSQL

عبارت NoSQL به مجموعه‌ای از پایگاه‌های داده گفته می‌شود که از زبان پرس و جوی ساخت یافته استفاده نمی‌کنند و در پاسخ به افزایش بسیار سریع حجم داده‌های ذخیره شده و نیاز به پردازش با سرعت بالا توسعه داده شد. اند [۳، ۴]. در جدول ۱ انواع مختلف پایگاه داده NoSQL توصیف شده است.

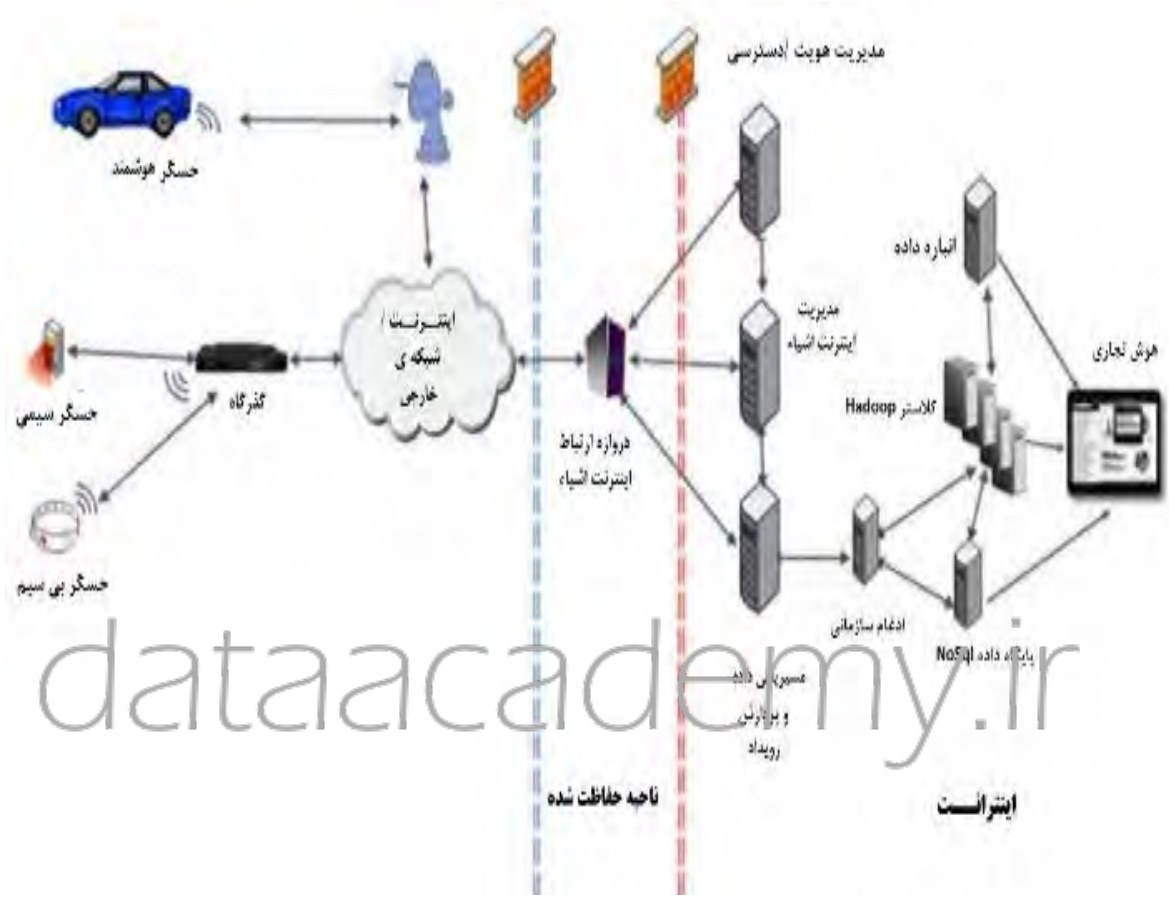
جدول ۱: توصیف انواع مختلف پایگاه داده NoSQL [۳، ۴]

انواع	ساختار	کاربرد
جفت کلید مقدار	شامل کلیدها و مقدار یا مجموعه ای از مقادیر	تراکنش‌های بسیار سبک
مبتنی بر ستون	مجموعه‌ای از یک یا چند جفت کلید مقدار به عنوان آرایه‌های دو بعدی	نمایش سوابق
مبتنی بر سند	شبهه پایگاه داده‌های مبتنی بر ستون	ذخیره سازی سند و ویژگی‌های عمیق تو در تو
مبتنی بر گراف	از ساختار درخت با گره ها و لبه‌های ارتباطی	شبکه‌های اجتماعی مانند فیس بوک و توییتر که در آن هر فرد یک گره در نظر گرفته می‌شود و رابطه دوستی بین افراد، یال‌های این گراف را تشکیل می‌دهند.

۳-۲- گزینه‌های معماری انبار داده‌ها برای تحلیل

داده‌های ساخت یافته و غیر ساخت یافته

در اینجا می‌خواهیم بررسی کنیم که اساسا چرا نیاز به اضافه کردن هادوپ و پایگاه داده‌های NoSQL در معماری اطلاعات یک سازمان وجود دارد. نیاز به بسط و گسترش معماری اطلاعات در حقیقت برخاسته از نیاز به پردازش منابع داده جریانی و نیمه‌ساخت یافته است. در واقع انبار داده‌های مبتنی بر پایگاه داده رابطه‌ای برای پردازش داده‌های غیرساخت یافته بهینه نمی‌باشند. به طور کلی می‌توان گفت که منابع داده‌های غیر ساخت یافته عبارتند از :



شکل ۱: یک معماری ساده از مولفه‌های اینترنت اشیا [۳]

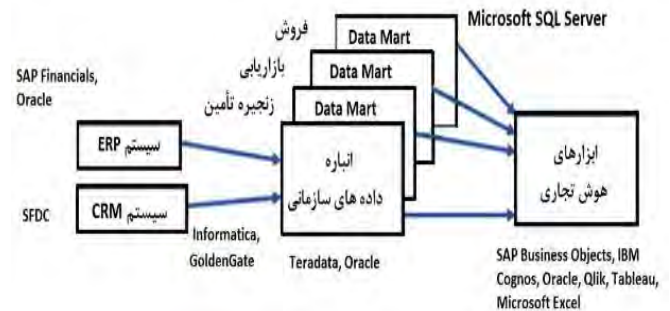
- داده‌های تولید شده در رسانه‌های اجتماعی

- داده‌های تولید شده توسط حسگرهای بی سیم

گزینه‌هایی که برای طراحی انبارداده‌ها در وضعیت فعلی سازمان‌ها وجود دارد را توصیف می‌کنیم.

۳-۱- انبار داده سنتی

همانطور که در شکل ۲ نشان داده شده است، انبار داده سنتی برای

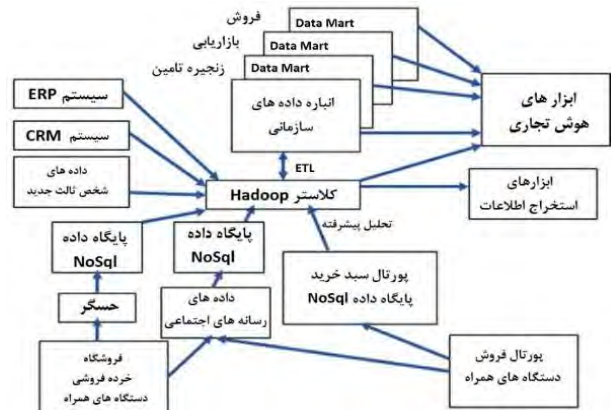


شکل ۲: شمایی از معماری انبارداده سنتی [۳]

نگهداری داده‌های ساخت‌یافته یک سازمان که توسط سیستم‌های عملیاتی سازمان تولید می‌شوند بکار می‌رود. مدل داده‌ای مورد استفاده برای طراحی این نوع از انبارداده‌ها مدل چند بُعدی (ستاره‌ای یا دانه برفی) می‌باشد انبارداده‌ها به عنوان منبعی برای انجام گزارشات تحلیلی مانند OLAP و انجام عملیات داده‌کاوی در نظر گرفته می‌شوند.

۳-۲- انبارداده سنتی با تغییر در فرایند ETL^۴

در شکل ۳ نشان داده شده است که چگونه هادوپ و NoSQL به عنوان بخشی از معماری انبارداده‌ها خواهند بود. به دلیل اینکه می‌توان از هادوپ در فرایند ETL نیز استفاده کرد، در شکل زیر معماری طوری



شکل ۳: انبارداده سنتی که در فرایند ETL آن از ابزارهای

NoSQL و هادوپ استفاده شده است [۳]

دستکاری شده است که داده‌ها ابتدا در هادوپ بارگذاری می‌شوند. شکل زیر یک شمای ساده از نحوه اضافه کردن هادوپ و NoSQL است. سولاتی که در این زمینه وجود دارد این است که هنگام انجام عملیات OLAP داده‌های روی هادوپ به انبارداده منتقل می‌شوند یا اینکه باید امکانی فراهم شود تا عملیات OLAP مستقیماً از طریق هادوپ صورت گیرد. لازم به ذکر است که در این گزینه معماری از ابزارهای هادوپ و NoSQL صرفاً برای مقاصد ETL استفاده می‌شوند و برای پیاده‌سازی مدل انبار داده مورد استفاده قرار نمی‌گیرند.

۳-۳- انبار داده سنتی (مدل ستاره‌ای مبتنی بر

NoSQL)

بعضی از تحقیقات برای حل مشکل انبارداده‌های بزرگ، مدل پایگاه داده چند بعدی (ستاره‌ای یا دانه برفی) را با استفاده از پایگاه داده‌های NoSQL پیاده‌سازی می‌کنند. در حقیقت رویکرد سنتی و معروف پیشنهاد شده توسط کیمبال یعنی مدل‌سازی بعدی را اتخاذ کرده ولی پیاده‌سازی را با استفاده از پایگاه داده‌های NoSQL انجام می‌دهند. این در حالی است که پایگاه داده‌های NoSQL به منظور بهبود مقیاس‌پذیری و انعطاف‌پذیری سیستم‌های OLTP طراحی شده‌اند [۹]. برای مثال در مقاله [۱۰] پیاده‌سازی انبار داده مبتنی بر پایگاه داده NoSQL سندگرا انجام شده است. همچنین در [۱۱]، به دلیل نبود یک رویکرد روشن برای پیاده‌سازی انبارداده‌ها بر مبنای مدل NoSQL، قوانینی در مورد چگونگی تبدیل انبارداده بعدی به مدل‌های داده‌های مبتنی بر NoSQL مانند مبتنی بر ستون و مبتنی بر سند ارائه شده است. این قوانین با استفاده از سیستم کاساندار^۵ به عنوان یک سیستم NoSQL مبتنی بر ستون و MongoDB به عنوان یک سیستم NoSQL مبتنی بر سند مقایسه شده است. نتیجه تحقیق آنها نشان داد که MongoDB برای پرس و جوهای OLAP مناسب تر است [۱۱].

۳-۴- انبارداده مبتنی بر هادوپ (مبتنی بر ستاره یا

کاملاً غیر نرمال شده)

به دلیل اینکه انبارداده‌ها به عنوان اساسی ترین دارایی سازمان‌ها در تصمیم‌گیری محسوب می‌شود [۷]. سازمان‌ها به دنبال این هستند تا انبارداده کنونی خود را برای پردازش داده‌های غیر ساخت‌یافته توسعه دهند [۱۲، ۱۳]. در این میان هادوپ به عنوان اصلی‌ترین کاندید به شمار می‌رود. با این وجود به دلیل اینکه متدولوژی خاصی برای طراحی انبارداده‌ها مبتنی بر هادوپ وجود ندارد، طراحان انبارداده، اغلب دانشی که

سوالاتی که اینجا مطرح است این است که آیا مدل‌سازی بُعدی انبار داده برای طراحی انبار داده روی Hive مناسب است؟ آیا مدل بُعدی بهینه‌تر از مدل جداول کاملاً غیرنرمال است؟ به عبارت دیگر بررسی اینکه چه مدل داده‌ای برای انبار داده مبتنی بر Hive مناسب است [۲۰].

در مقاله [۲۰] به بررسی این موضوع پرداخته شده است. نتیجه آن مقاله به این شرح است که انبار داده‌های مبتنی بر هادوپ دارای عملکرد بهتری با اتخاذ رویکرد مدل‌سازی داده کاملاً غیر نرمال شده در مقایسه با شما ستاره‌ای هستند. علی‌رغم امکان پیاده‌سازی شمای ستاره‌ای روی Hive، این رویکرد ممکن است الگوی طراحی کارایی نباشد. در نتیجه علی‌رغم تلاش‌های متعدد برای مدل‌سازی بُعدی مبتنی بر هادوپ، می‌توان گفت که انبار داده‌های مبتنی بر هادوپ با استفاده از ساختارهای داده‌های کاملاً غیرنرمال شده بهینه‌ترین گزینه هستند [۲۰].

۴- نتیجه‌گیری

انبار داده به عنوان مخزن داده‌های یک سازمان، منبع داده‌ای مناسب و بهینه جهت پشتیبانی از تصمیم‌مورد استفاده قرار می‌گیرد. ابزارها و مدل‌های داده‌ای سنتی انبار داده‌ها، برای کار با داده‌های ساختاریافته مورد طراحی واقع شده‌اند. با این وجود امروزه سازمان‌ها با داده‌های بزرگ و غیرساخت‌یافته مواجه هستند که مدل‌های سنتی انبار داده برای مدیریت و پردازش آنها بهینه نیستند. در این مقاله با توجه به تولید داده‌های غیرساخت‌یافته، گزینه‌های مختلف معماری انبار داده بررسی شد و ویژگی‌های هر کدام از آنها تشریح گردید. نتایج بررسی نشان می‌دهد که دو موضوع برای ساخت انبار داده‌های بزرگ مطرح است:

(۱) اینکه چه مدلی برای طراحی انبار داده استفاده شود (۲) و از چه ابزاری برای پیاده‌سازی مدل طراحی شده استفاده شود. در تحقیقات پیشین غالباً از مدل داده‌ای ستاره‌ای برای این منظور استفاده شده است همچنین رویکردهای مختلفی برای پیاده‌سازی انبار داده‌ها مورد استفاده قرار گرفته است. بر اساس تحقیقات پیشین سیستم‌های SQL-on-Hadoop به عنوان یک راه‌حل استاندارد برای طراحی انبار داده بزرگ محسوب می‌شوند.

در زمینه طراحی انبار داده سنتی را دارند به کار می‌گیرند. برای مثال، پیاده‌سازی مدل ستاره‌ای یا دانه برفی روی Hive (SQL-on-Hadoop) [۱۴].

همچنین سیستم‌های SQL-on-Hadoop به عنوان یک راه‌حل استاندارد برای طراحی انبار داده بزرگ محسوب می‌شوند. لیست سیستم‌های SQL-on-Hadoop گسترده است. برای مثال Hive [۱۴]، Presto [۱۵]، Impala [۱۶]، Sparksql [۱۷] و Apache drill [۱۸].

در ادامه این مقاله روی Hive تمرکز می‌کنیم. به عنوان یک ابزار انبار داده، Hive عملکرد پرس و جو را در پایگاه داده‌های خیلی بزرگ توزیع شده بهبود می‌بخشد. Hive به عنوان انبار داده، بیشتر مورد استفاده توسط بسیاری از سازمان‌ها به منظور مدیریت و پردازش حجم عظیمی از داده‌ها است. Hive توسط فیسبوک برای بهبود قابلیت پرس جوی هادوپ که ناکارآمد بود توسعه داده شده است [۱۴].

Hive دارای ویژگی‌های مهمی است که در زیر فهرست شده است:

- ابزارهایی را فراهم می‌کند تا ETL را تسهیل کند.
- دارای مکانیزم‌هایی برای سازگار نمودن داده‌های با فرمت متفاوت است.
- دارای امکاناتی برای انجام پرس و جو از داده‌های HDFS و سایر منابع داده‌ای است.
- وظایف آن با استفاده از چارچوب MapReduce اجرا می‌شوند.

Hive دارای زبانی شبیه SQL است که HiveQL نامیده می‌شود که کاربران را قادر می‌سازد تا پرس و جو‌هایی روی داده‌ها با استفاده از دستورات شبیه SQL انجام دهند. همچنین پلاگین‌هایی را فراهم می‌کند برای برنامه نویسان تا دستورات نگاشت و کاهش خودشان را برای انجام تحلیل‌های پیچیده آماده کنند [۱۹]. همچنین داده‌ها را در قالب جداول (هر جدولی متناظر با یک دایرکتوری HDFS است)، پارتیشن‌ها (زیر دایرکتوری) و باکت‌ها (سگمنت‌هایی از فایل در HDFS) سازماندهی می‌کند. همچنین Hive دارای یک زبان پرس و جوی مبتنی بر SQL که HiveQL نامیده می‌شود، است [۱۹].

در بکارگیری Hive برای پیاده‌سازی انبار داده دو گزینه وجود دارد:

- پیاده‌سازی بر اساس مدل بُعدی
- پیاده‌سازی بر اساس مدل جدول کاملاً غیرنرمال شده^۷



- NoSQL Data Base: Comparative Study. *Procedia Computer Science*, 2016. **96**: p. 255-264.
- [12]. Russom, P., *Evolving data warehouse architectures in the age of big data*. The Data Warehouse Institute, 2014.
- [13]. Russom, P., *Data warehouse modernization in the age of big data analytics*. The Data Warehouse Institute, 2016.
- [14]. Thusoo, A., et al. *Hive-a petabyte scale data warehouse using hadoop*. in *Data Engineering (ICDE), 2010 IEEE 26th International Conference on*. 2010. IEEE.
- [15]. *Presto: Presto | Distributed SQL Query Engine for Big Data*. <https://prestodb.io/>.
- [16]. Bittorf, M., et al. *Impala: A modern, open-source SQL engine for Hadoop*. in *Proceedings of the 7th Biennial Conference on Innovative Data Systems Research*. 2015.
- [17]. Armbrust, M., et al. *Spark sql: Relational data processing in spark*. in *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*. 2015. ACM.
- [18]. Hausenblas, M. and J. Nadeau, *Apache drill: interactive ad-hoc analysis at scale*. *Big Data*, 2013. **1**(2): p. 100-104.
- [19]. Yang, Q. and M. Helfert. *Revisiting arguments for a three layered data warehousing architecture in the context of the Hadoop platform*. in *The 6th International Conference on Cloud Computing and Services Science (CLOSER)*. 2016. SCITEPRESS - Science and Technology Publications.
- [20]. Costa, E., C. Costa, and M.Y. Santos. *Efficient Big Data Modelling and Organization for Hadoop Hive-Based Data Warehouses*. in *European, Mediterranean, and Middle Eastern Conference on Information Systems*. 2017. Springer.
- [1]. Stackowiak, R., et al., *Evaluating the Art of the Possible*, in *Big Data and the Internet of Things: Enterprise Information Architecture for a New Age*, R. Stackowiak, et al., Editors. 2015, Apress: Berkeley, CA. p. 29-48.
- [2]. Krishnan, K., *Data warehousing in the age of big data*. 2013, Boston: Morgan Kaufmann.
- [3]. Matouk, Kamal, and Mieczysław L. Owoc. "A survey of data warehouse architectures—Preliminary results." In *Computer Science and Information Systems (FedCSIS), 2012 Federated Conference on*, pp. 1121-1126. IEEE, 2012.
- [4]. Ariyachandra, Thilini, and Hugh J. Watson. "Which data warehouse architecture is most successful?." *Business Intelligence Journal* **11**, no. 1 (2006): 4.
- [5]. Inmon, W.H., D. Strauss, and G. Neushloss, *DW 2.0: The architecture for the next generation of data warehousing*. 2010: Elsevier.
- [6]. Kimball, R. and M. Ross, *The data warehouse toolkit: The definitive guide to dimensional modeling*. 2013: John Wiley & Sons.
- [7]. Goss, R. G. and K. Veeramathu. *Heading towards big data building a better data warehouse for more data, more speed, and more users*. in *Advanced Semiconductor Manufacturing Conference (ASMC), 2013 24th Annual SEMI*. 2013. IEEE.
- [8]. Mohanty, S., M. Jagadeesh, and H. Srivatsa, *Big data imperatives: Enterprise 'Big Data' warehouse, 'BI' implementations and analytics*. 2013: Apress.
- [9]. Cattell, R., *Scalable SQL and NoSQL data stores*. *SIGMOD Rec.*, 2011. **39**(4): p. 12-27.
- [10]. Chevalier, M., et al., *Document-oriented Models for Data Warehouses*, in *Proceedings of the 18th International Conference on Enterprise Information Systems*. 2016, SCITEPRESS - Science and Technology Publications, Lda: Rome, Italy. p. 142-149.
- [11]. Yangui, R., A. Nabli, and F. Gargouri, *Automatic Transformation of Data Warehouse Schema to*

زیر نویس‌ها

⁵ Apache Cassandra
⁶ Bucket
⁷ Fully denormalized

¹ Online Transaction Processing System
² Inmon et al.
³ Kimball
⁴ Extract-Transform-Load