

4 رده و گروه عمده بانک‌های اطلاعاتی NoSQL وجود دارند؛ شامل:  
 الف) Key-Value stores که پایه بانک‌های اطلاعاتی NoSQL را تشکیل داده و اهدافی عمومی را دنبال می‌کنند.  
 ب) Wide column stores که در شرکت‌های بزرگ اینترنتی بیشتر مورد استفاده قرار گرفته‌اند.  
 ج) Document stores یا بانک‌های اطلاعاتی NoSQL سندگرا.  
 د) Graph databases که بیشتر برای ردیابی ارتباطات بین موجودیت‌ها بکار می‌روند.

و در تمام این گروه‌ها، مکانیزم‌های Key-Value به شدت مورد استفاده‌اند.

### الف) Key-Value stores

[Key-Value stores](#) یکی از عمومی‌ترین و پایه‌ای‌ترین گروه‌های بانک‌های اطلاعاتی NoSQL را تشکیل می‌دهند. البته این مورد بدین معنا نیست که این رده، جزو محبوب‌ترین‌ها نیز به‌شمار می‌روند.

dynomite



این نوع بانک‌های اطلاعاتی شامل جداولی از اطلاعات هستند. هر جدول نیز شامل تعدادی ردیف است؛ چیزی همانند بانک‌های اطلاعاتی رابطه‌ای. اما در هر ردیف، یک Dictionary یا آرایه‌ای از اطلاعات key-value شکل را شاهد خواهید بود. در اینجا ساختار و اسکیمای ردیف‌ها می‌توانند نسبت به یکدیگر کاملاً متفاوت باشند (دید لیبرال نسبت به اسکیمای، که [در قسمت قبل](#) به آن پرداخته شد). در این بین، تنها تضمین خواهد شد که هر ردیف، Id منحصر بفردی دارد.

از این نوع بانک‌های اطلاعاتی، در سکوهاى کارى ابرى زیاد استفاده می‌شود. دو مثال مهم در اینباره شامل [Amazon SimpleDB](#) و [Azure Table Storage](#) هستند.

سایر نمونه‌های مهم دیگری از بانک‌های اطلاعاتی NoSQL که بر مبنای مفهوم Key-Value stores کار می‌کنند، عبارتند از [MemcacheDB](#) و [Voldemort](#). به علاوه در Amazon web services بانک اطلاعاتی دیگری به نام [DynamoDB](#) به عنوان یک سرویس عمومی در دسترس است. همچنین [Dynomite](#) نیز به عنوان نمونه سورس باز Dynamo مطرح است. [Redis](#) و [Riak](#) نیز جزو بانک‌های اطلاعاتی Key-Value store بسیار معروف به‌شمار می‌روند.

## Key-Value Stores

Database	
Table: Customers	Table: Orders
Row ID: 101 First_Name: Vahid Last_Name: Nasiri Address: Iran Last_Order: 1501	Row ID: 1501 Price: 400 Item1: 42134 Item2: 23455
Row ID: 201 First_Name: Ali Last_Name: Moshfegh Address: Iran Last_Order: 1502	Row ID: 1502 Price: 300 Item1: 52134 Item2: 26455

همانطور که در تصویر فوق ملاحظه می‌کنید، Key-Value stores دارای بانک‌های اطلاعاتی شامل جداول مختلف هستند. در اینجا همچنین ساختار ردیف‌هایی از اطلاعات این جداول نیز مشخص شده‌اند. هر ردیف، یک کلید دارد به همراه تعدادی جفت کلید-مقدار. در این جداول، اسکیما ثابت نگه داشته شده است و از ردیفی به ردیف دیگر متفاوت نیست؛ اما این مساله اختیاری است. برای مثال می‌توان در ردیف اطلاعات یک مشتری خاص، کلید-مقدارهایی خاص او را نیز درج کرد که لزوماً در سایر ردیف‌ها، نیازی به وجود آن‌ها نیست. به علاوه باید به خاطر داشت که هرچند به ظاهر last\_orderها به شماره Id سفارشات مرتبط هستند، اما مفاهیمی مانند کلیدهای خارجی بانک‌های اطلاعاتی رابطه‌ای، در اینجا وجود خارجی ندارند. بیشتر در اینجا هدف سهولت جستجوی اطلاعات است.

### Wide column stores (ب)

Wide column stores دارای جداولی است که درون آن‌ها ستون‌هایی قابل تعریف است. درون این ستون‌ها که یادآور بانک‌های اطلاعاتی رابطه‌ای هستند، اطلاعات به شکل key-value با ساختاری متفاوت، قابل ذخیره سازی هستند. در اینجا هر ستون، می‌تواند شامل گروهی از ستون‌ها که بر اساس مفاهیم جفت‌های key-value کار می‌کنند، باشد. این نوع بانک‌های اطلاعاتی عموماً در سایت‌های اینترنتی بسیار بزرگ و برنامه‌های «Big data» استفاده می‌شوند. برای مثال:



- [BigTable](#) گوگل که یک محصول اختصاصی و غیرعمومی است؛ اما جزئیات آن را به عنوان مقالات علمی منتشر کرده است.  
 - دنیای سورس باز به رهبری Yahoo، نمونه سورس باز BigTable را به نام [Hbase](#) ارائه داده است.  
 - در فیس بوک، از بانک اطلاعاتی دیگری به نام [Cassandra](#) استفاده می‌کنند. در اینجا به گروهی از ستون‌ها super columns و جداول super column families گفته می‌شود.

### Wide column stores

Table: Customers	Table: Orders
Row ID: 101 Super column: Name Column: First_Name: Vahid Column: Last_Name: Nasiri Super column: Address Column: Number: 10 Column: Street: Somewhere Super column: Orders Column: Last_Order: 1501	Row ID: 1501 Super column: Pricing Column: Price: 400 Super column: Items Column: Item1: 12345 Column: Item2: 14345
Row ID: 201 Super column: Name Column: First_Name: Ali Column: Last_Name: Moshfegh Super column: Address Column: Number: 101 Column: Street: Somewhere Super column: Orders Column: Last_Order: 1502	Row ID: 1502 Super column: Pricing Column: Price: 500 Super column: Items Column: Item1: 17345 Column: Item2: 14945

در اینجا نیز جداول و ردیف‌ها وجود دارند و هر ستون باید عضوی از خانواده یک super column باشد. ساختار ردیف‌ها در این تصویر یکسان در نظر گرفته شده‌اند، اما اگر نیاز بود، برای مثال می‌توان در ردیفی خاص، ساختار را تغییر داد و مثلا middle name را نیز بر اساس نیاز، به ردیفی اضافه کرد.

### Document stores (ج)

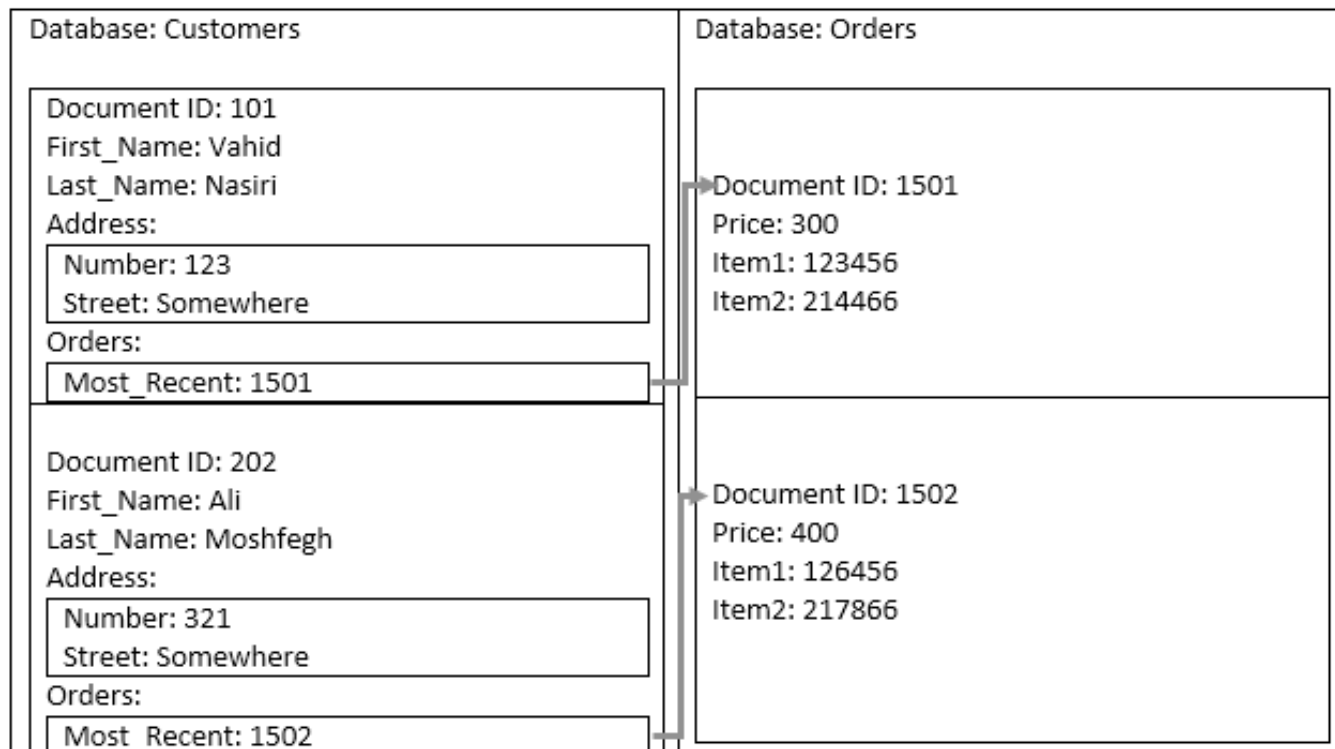
Document stores بجای جداول، دارای بانک‌های اطلاعاتی مختلفی هستند و در اینجا بجای ردیف‌ها، سند یا document دارند. ساختار سندها نیز عموماً بر مبنای اشیاء [JSON](#) تعریف می‌گردد (که البته این مورد الزامی نبوده و از هر محصول، به محصول دیگری ممکن است متفاوت باشد؛ اما عمومیت دارد). بنابراین هر سند دارای تعدادی خاصیت است (چون اشیاء JSON به این نحو تعریف می‌گردند) که دارای مقدار هستند. در نگاه اول، شاید این نوع اسناد، بسیار شبیه به key-value stores به نظر برسند. اما در حین تعریف اشیاء JSON، یک مقدار می‌تواند خود یک شیء کامل دیگر باشد و نه صرفاً یک مقدار ساده. به همین جهت عده‌ای به این نوع بانک‌های اطلاعاتی، بانک‌های اطلاعاتی Key-value store سفارشی و خاص نیز می‌گویند. این نوع ساختار منعطف، برای ذخیره سازی اطلاعات اشیاء تو در تو و درختی بسیار مناسب است. همچنین این اسناد می‌توانند حاوی پیوست‌هایی نیز باشد؛ مانند پیوست یک فایل به یک سند. در Document stores، نگارش‌های قدیمی اسناد نیز نگهداری می‌گردند. به همین جهت این نوع بانک‌های اطلاعاتی برای ایجاد برنامه‌های مدیریت محتوا نیز بسیار مطلوب می‌باشند. با توجه به مزایایی که برای این رده از بانک‌های اطلاعاتی NoSQL ذکر گردید، Document stores در بین برنامه نویسی‌ها بسیار محبوب و پرکاربرد هستند.

از این دست بانک‌های اطلاعاتی NoSQL، می‌توان به [MongoDB](#)، [CouchDB](#) و [RavenDB](#) اشاره کرد. سایر مزایای Document stores که به پرکاربرد شدن آن‌ها کمک کرده‌اند به شرح زیر هستند:

- هر سند را می‌توان با یک URI آدرس دهی کرد.
- برای نمونه CouchDB از یک full REST interface برای دسترسی و کار با اسناد پشتیبانی می‌کند (چیزی شبیه به ASP.NET WEB API در دات نت). در اینجا با استفاده از یک وب سرور توکار و بکارگیری HTTP Verbs مانند Put, Delete, Get و غیره، امکان کار با اسناد وجود دارد.

- اغلب بانک‌های اطلاعاتی Document stores از JavaScript به عنوان native language خود بهره می‌برند (جهت سهولت کار با اشیاء JSON).

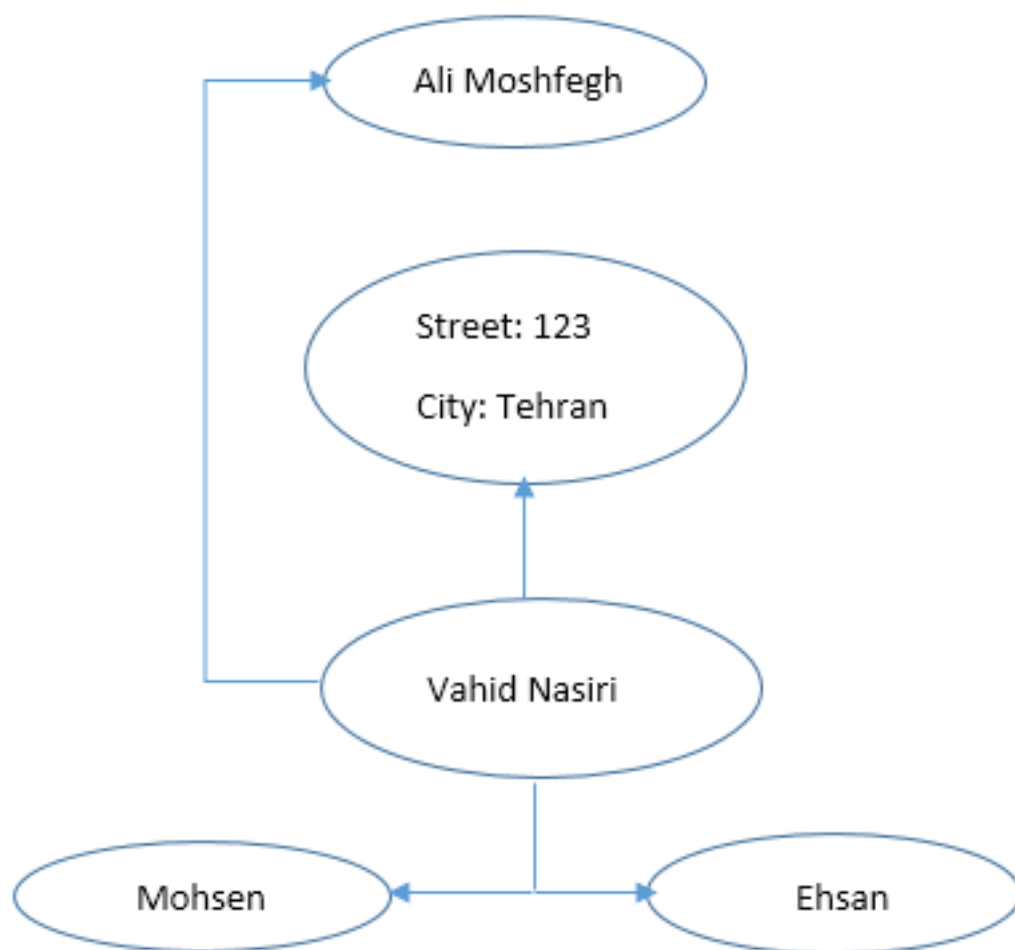
### Document stores



در اینجا دو دیتابیس، بجای دو جدول وجود دارند. همچنین در مقایسه با بانک‌های اطلاعاتی key-value، برای نمونه، مقدار خاصیت آدرس، خود یک شیء است که از دو خاصیت تشکیل شده است. به علاوه هر خاصیت Most\_Recent یک Order، به سند دیگری در بانک اطلاعاتی Orders لینک شده است.

### Graph databases (د)

Graph databases نوع خاصی از بانک‌های اطلاعاتی NoSQL هستند که جهت ردیابی ارتباطات بین اطلاعات طراحی شده‌اند و برای برنامه‌های شبکه‌های اجتماعی بسیار مفید هستند. در واژه نامه این بانک‌های اطلاعاتی Nodes و Edges (اتصال دهنده‌های نودها) تعریف شده‌اند. در اینجا نودها می‌توانند دارای خاصیت‌ها و مقادیر متناظر با آن‌ها باشند. یکی از معروفترین Graph databases مورد استفاده، [Neo4j](#) نام دارد.



در اینجا یک شخص را که دارای رابطه آدرس با شیء آدرس ذکر شده است را مشاهده می‌کنید. همچنین این شخص دارای رابطه دوستی با سه شخص دیگر است.