**Dados não relacionais e NoSQL**

Um *banco de dados não relacional* é um banco de dados que não usa o esquema de tabela de linhas e colunas encontrado na maioria dos sistemas de banco de dados tradicionais. Em vez disso, os bancos de dados não relacionais usam um modelo de armazenamento otimizado para os requisitos específicos do tipo de dados que está sendo armazenado. Por exemplo, os dados podem ser armazenados como pares chave/valor simples, como documentos JSON ou como um gráfico que consiste em bordas e vértices.

O que esses armazenamentos de dados têm em comum é que eles não usam um [modelo relacional](https://docs.microsoft.com/pt-br/azure/architecture/data-guide/). Além disso, eles tendem a ser mais específicos no tipo de dados ao qual dão suporte e no modo como os dados podem ser consultados. Por exemplo, os armazenamentos de dados de série temporal são otimizados para consultas em sequências de dados baseadas em tempo. No entanto, os armazenamentos de dados de grafo são otimizados para explorar as relações ponderadas entre entidades. Nenhum dos dois formatos será bem generalizado para a tarefa de gerenciamento de dados transacionais.

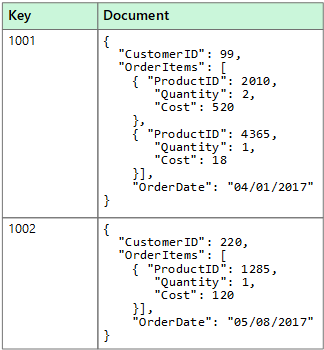
O termo *NoSQL* refere-se aos armazenamentos de dados que não usam o SQL para consultas. Em vez disso, os armazenamentos de dados usam outras linguagens de programação e constructos para consultar os dados. Na prática, "NoSQL" significa "banco de dados não relacionais", mesmo que muitos desses bancos de dados deem suporte a consultas compatíveis com SQL. No entanto, a estratégia de execução de consulta subjacente é geralmente muito diferente da maneira como um RDBMS tradicional executa a mesma consulta SQL.

As seções a seguir descrevem as principais categorias de banco de dados não relacional ou NoSQL.

**Armazenamentos de dados de documentos**

Um armazenamento de dados de documento gerencia um conjunto de campos de cadeia de caracteres nomeados e valores de dados de objeto em uma entidade conhecida como *documento*. Normalmente, esses repositórios de dados armazenam dados na forma de documentos JSON. Cada valor de campo pode ser um item escalar, como um número ou um elemento composto, como uma lista ou uma coleção de pai-filho. Os dados nos campos de um documento podem ser codificados de várias maneiras, incluindo XML, YAML, JSON, BSON ou, até mesmo, armazenados como texto sem formatação. Os campos nos documentos são expostos ao sistema de gerenciamento de armazenamento, permitindo que um aplicativo consulte e filtre dados utilizando os valores nesses campos.

Normalmente, um documento contém todos os dados de uma entidade. Quais itens constituem uma entidade são específicos do aplicativo. Por exemplo, uma entidade pode conter os detalhes de um cliente, um pedido ou uma combinação de ambos. Um único documento pode conter informações que serão distribuídas em várias tabelas relacionais em um RDBMS. Um repositório de documentos não requer que todos os documentos tenham a mesma estrutura. Essa abordagem em forma livre oferece grande flexibilidade. Por exemplo, os aplicativos podem armazenar dados diferentes em documentos, em resposta a uma alteração nos requisitos de negócios.



O aplicativo pode recuperar documentos utilizando a chave de documento. A chave é um identificador exclusivo do documento, que geralmente tem um hash, para ajudar a distribuir os dados de maneira uniforme. Alguns bancos de dados de documentos criam a chave do documento automaticamente. Outros permitem que você especifique um atributo do documento a ser usado como chave. O aplicativo também pode consultar documentos com base no valor de um ou mais campos. Alguns bancos de dados de documentos dão suporte à indexação para facilitar a pesquisa rápida de documentos com base em um ou mais campos indexados.

Muitos bancos de dados de documentos fornecem suporte a atualizações in loco, permitindo que um aplicativo modifique os valores de campos específicos em um documento sem regravar o documento inteiro. As operações de leitura e gravação em vários campos em um só documento normalmente são atômicas.

Serviço do Azure relevante:

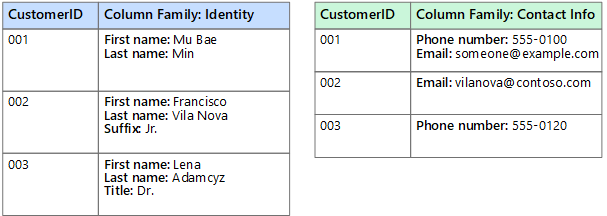
* [Azure Cosmos DB](https://azure.microsoft.com/services/cosmos-db/)

**Armazenamentos de dados de colunas**

Um armazenamento de dados de colunas ou de família de colunas organiza os dados em colunas e linhas. Em sua forma mais simples, um armazenamento de dados de família de colunas pode parecer muito semelhante a um banco de dados relacional, pelo menos conceitualmente. O poder real de um banco de dados de família de colunas está em sua abordagem desnormalizada para a estruturação de dados esparsos, que deriva da abordagem orientada por coluna para armazenar dados.

É possível pensar em um armazenamento de dados de família de colunas como contendo dados de tabela com linhas e colunas, mas as colunas são divididas em grupos conhecidos como famílias de colunas. Cada família de colunas contém um conjunto de colunas que estão logicamente relacionadas e geralmente são recuperadas ou manipuladas como uma unidade. Outros dados acessados separadamente podem ser armazenados em famílias de colunas separadas. Dentro de uma família de colunas, novas colunas podem ser adicionadas dinamicamente e as linhas podem ser esparsas (ou seja, uma linha não precisa ter um valor para cada coluna).

O diagrama a seguir mostra um exemplo com duas famílias de colunas, Identity e Contact Info. Os dados de uma única entidade têm a mesma chave de linha em cada família de colunas. Essa estrutura, em que as linhas de determinado objeto em uma família de colunas podem variar dinamicamente, é um benefício importante da abordagem de família de colunas, que torna essa forma de armazenamento de dados altamente adequada para armazenar dados com esquemas variados.



Ao contrário de um repositório de valor/chave ou um banco de dados de documentos, a maioria dos bancos de dados de família de colunas armazenam dados fisicamente em ordem de chave, em vez de computar um hash. A chave de linha é considerada o índice primário e permite o acesso baseado em chave por meio de uma chave específica ou um intervalo de chaves. Algumas implementações permitem que você crie índices secundários em colunas específicas de uma família de colunas. Os índices secundários permitem recuperar dados por valor de colunas, em vez da chave de linha.

No disco, todas as colunas de uma família de colunas são armazenadas juntas no mesmo arquivo, com um número específico de linhas em cada arquivo. Com conjuntos de dados grandes, esse método cria um benefício de desempenho, reduzindo a quantidade de dados que precisam ser lidos do disco quando apenas algumas colunas são consultadas juntas por vez.

As operações de leitura e gravação para uma linha normalmente são atômicas em uma só família de colunas, embora algumas implementações forneçam atomicidade em toda a linha, abrangendo várias famílias de colunas.

Serviço do Azure relevante:

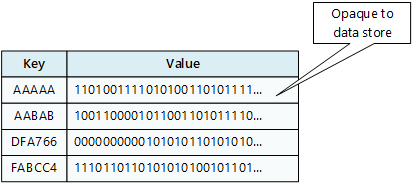
* [API do Cassandra do Cosmos DB](https://docs.microsoft.com/pt-br/azure/cosmos-db/cassandra-introduction)
* [HBase no HDInsight](https://docs.microsoft.com/pt-br/azure/hdinsight/hdinsight-hbase-overview)

**Armazenamentos de dados de chave/valor**

Um armazenamento de valor/chave é essencialmente uma tabela de hash grande. Você associa cada valor de dados a uma chave exclusiva e o armazenamento de valor/chave usa essa chave para armazenar os dados, utilizando uma função de hash apropriada. A função de hash é selecionada para fornecer uma distribuição uniforme de chaves de hash em todo o armazenamento de dados.

A maioria dos armazenamentos de valor/chave fornecem suporte apenas às operações de exclusão, inserção e consulta simples. Para modificar um valor (parcial ou completamente), um aplicativo deverá substituir os dados existentes para todo o valor. Na maioria das implementações, a leitura ou gravação de um único valor é uma operação atômica. Se o valor for grande, a gravação poderá demorar algum tempo.

Um aplicativo pode armazenar dados arbitrários como um conjunto de valores, embora alguns armazenamentos de valor/chave imponham limites ao tamanho máximo de valores. Os valores armazenados são opacos para o software do sistema de armazenamento. Quaisquer informações de esquema deverão ser fornecidas e interpretadas pelo aplicativo. Essencialmente, os valores são blobs e o armazenamento de valor/chave simplesmente recupera ou armazena o valor por chave.



Armazenamentos de chave/valor são altamente otimizados para aplicativos que realizam pesquisas simples usando o valor da chave ou por um intervalo de chaves, mas são menos adequados para sistemas que precisam consultar dados em tabelas diferentes de chaves/valores, como a união de dados em várias tabelas.

Os armazenamentos de chave/valor também não são otimizados para cenários em que a consulta ou a filtragem por valores não chave é importante, em vez de realizar pesquisas baseadas somente em chaves. Por exemplo, com um banco de dados relacional, você pode encontrar um registro usando uma cláusula WHERE para filtrar as colunas não chave, mas armazenamentos de chave/valores geralmente não têm esse tipo de funcionalidade de pesquisa de valores ou se têm, é preciso fazer um exame lento de todos os valores.

Um armazenamento de valor/chave pode ser extremamente escalável, pois o armazenamento de dados pode facilmente distribuir dados em vários nós em computadores separadas.

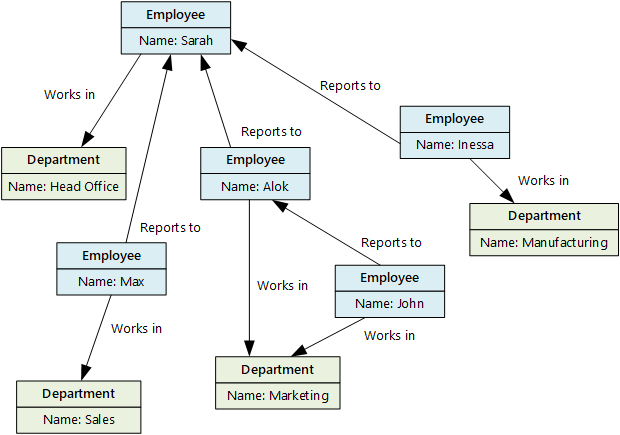
Serviços do Azure relevantes:

* [API de Tabela do Azure Cosmos DB](https://docs.microsoft.com/pt-br/azure/cosmos-db/table-introduction)
* [Cache Redis do Azure](https://azure.microsoft.com/services/cache/)
* [Armazenamento de Tabelas do Azure](https://azure.microsoft.com/services/storage/tables/)

**Armazenamentos de dados de gráficos**

Um armazenamento de dados de gráficos armazena dois tipos de informações: nós e bordas. Nós representam entidades e bordas especificam as relações entre essas entidades. Ambos os nós e as bordas podem ter propriedades que fornecem informações sobre esse nó ou borda, semelhante às colunas em uma tabela. As bordas também podem ter uma direção indicando a natureza do relacionamento.

O objetivo de um armazenamento de dados de gráficos é permitir que um aplicativo execute com eficácia consultas que atravessam a rede de nós e bordas e analise as relações entre entidades. O diagrama a seguir mostra os dados de pessoal de uma organização estruturados como um gráfico. As entidades são funcionários e departamentos, e as bordas indicam os relacionamentos de relatórios e o departamento em que os funcionários trabalham. Nesse grafo, as setas nas bordas mostram a direção das relações.



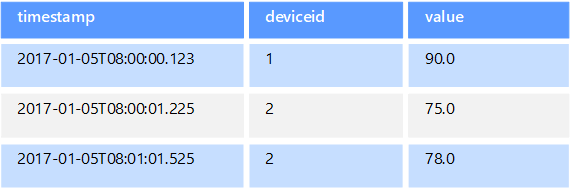
Essa estrutura simplifica a execução de consultas como "Encontrar todos os funcionários que são subordinados diretos ou indiretos de Marina" ou "Quem trabalha no mesmo departamento que Júlio?". Para grafos grandes com muitas entidades e relações, você pode executar análises complexas rapidamente. Muitos bancos de dados de gráficos fornecem uma linguagem de consulta que você pode utilizar para percorrer uma rede de relacionamentos de forma eficiente.

Serviço do Azure relevante:

* [API do Graph do Azure Cosmos DB](https://docs.microsoft.com/pt-br/azure/cosmos-db/graph-introduction)

**Armazenamentos de dados de série temporal**

Os dados de série temporal são um conjunto de valores organizados por tempo e um armazenamento de dados de série temporal é otimizado para esse tipo de dados. Os armazenamentos de dados de série temporal precisam dar suporte a um número muito alto de gravações, pois geralmente coletam grandes quantidades de dados em tempo real de uma grande variedade de fontes. Os armazenamentos de dados de série temporal são otimizados para armazenar dados telemétricos. Os cenários incluem sensores de IoT ou contadores de sistemas/aplicativos. As atualizações são raras e as exclusões geralmente são feitas como operações em massa.



Embora os registros gravados em um banco de dados de série temporal sejam geralmente pequenos, muitas vezes, há um grande número de registros e o tamanho total dos dados pode aumentar rapidamente. Os armazenamentos de dados de série temporal também manipulam dados fora de ordem e de chegada tardia, a indexação automática de pontos de dados e otimizações de consultas descritas em termos de janelas de tempo. Esse último recurso permite a execução de consultas em milhões de pontos de dados e em vários fluxos de dados com rapidez, para dar suporte a visualizações de série temporal, que é uma maneira comum pela qual os dados de séries temporal são consumidos.

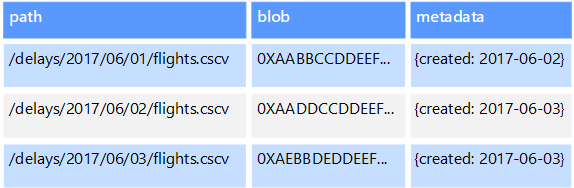
Para obter mais informações, consulte [Soluções de série temporal](https://docs.microsoft.com/pt-br/azure/architecture/data-guide/scenarios/time-series)

Serviços do Azure relevantes:

* [Azure Time Series Insights](https://azure.microsoft.com/services/time-series-insights/)
* [OpenTSDB com HBase no HDInsight](https://docs.microsoft.com/pt-br/azure/hdinsight/hdinsight-hbase-overview)

**Armazenamentos de dados de objetos**

Os armazenamentos de dados de objetos são otimizados para armazenar e recuperar grandes objetos binários ou blobs como imagens, arquivos de texto, fluxos de áudio e vídeo, grandes documentos e objetos de dados de aplicativos e imagens de disco de máquina virtual. Um objeto consiste nos dados armazenados, em alguns metadados e uma ID exclusiva para acessar o objeto. Os armazenamentos de objetos foram projetados para dar suporte a arquivos que são individualmente muito grandes, além de fornecer grandes quantidades de armazenamento total para gerenciar todos os arquivos.



Alguns armazenamentos de dados de objetos replicam determinado blob em vários nós de servidor, o que permite rápidas leituras paralelas. Esse processo, por sua vez, habilita a consulta de expansão dos dados contidos em arquivos grandes, pois vários processos, normalmente em execução em diferentes servidores, podem consultar o arquivo de dados grande simultaneamente.

Um caso especial de armazenamentos de dados de objetos é o compartilhamento de arquivos de rede. O uso de compartilhamentos de arquivos permite que os arquivos sejam acessados em uma rede usando protocolos de rede padrão como o protocolo SMB. Considerando os mecanismos apropriados de segurança e controle de acesso simultâneo, compartilhar dados dessa forma pode permitir que os serviços distribuídos forneçam acesso a dados altamente escalonáveis para operações básicas de baixo nível, como solicitações de leitura e gravação simples.

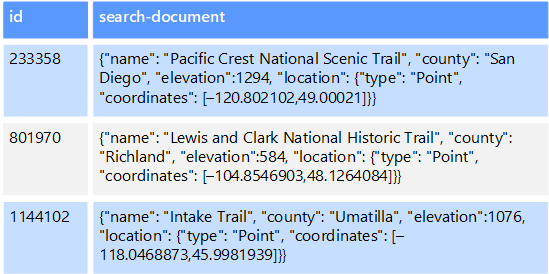
Serviços do Azure relevantes:

* [Armazenamento de Blobs do Azure](https://azure.microsoft.com/services/storage/blobs/)
* [Repositório Azure Data Lake](https://azure.microsoft.com/services/data-lake-store/)
* [Armazenamento de Arquivos do Azure](https://azure.microsoft.com/services/storage/files/)

**Armazenamentos de dados de índice externo**

Os armazenamentos de dados de índice externo fornecem a capacidade de pesquisar informações mantidas em outros armazenamentos de dados e serviços. Um índice externo atua como um índice secundário para qualquer armazenamento de dados e pode ser usado para indexar grandes volumes de dados e fornecer acesso quase em tempo real a esses índices.

Por exemplo, talvez você tenha arquivos de texto armazenados em um sistema de arquivos. Encontrar um arquivo por seu arquivo de caminho é rápido, mas pesquisar com base no conteúdo do arquivo exige um exame de todos os arquivos, o que é lento. Um índice externo permite criar índices de pesquisa secundária e, em seguida, localizar rapidamente o caminho para os arquivos que correspondem aos critérios. Outro aplicativo de exemplo de um índice externo são armazenamentos de chave/valor que somente indexam pela chave. Você pode criar um índice secundário com base nos valores dos dados e pesquisar rapidamente a chave que identifica exclusivamente cada item correspondente.



Os índices são criados pela execução de um processo de indexação. Isso pode ser feito usando um modelo de pull, disparado pelo armazenamento de dados ou usando um modelo de push, iniciado pelo código do aplicativo. Os índices podem ser multidimensionais e dar suporte a pesquisas de texto livre em grandes volumes de dados de texto.

Os armazenamentos de dados de índice externo costumam ser usados para dar suporte às pesquisas de texto completo e baseadas na Web. Nesses casos, a pesquisa pode ser exata ou difusa. Uma pesquisa difusa localiza documentos que correspondem um conjunto de termos e calcula a forma como eles correspondem. Alguns índices externos também dão suporte à análise linguística que pode retornar correspondências com base em sinônimos, expansões de gênero (por exemplo, correspondência de "cachorros" com "animais de estimação") e lematização (por exemplo, a pesquisa de "correr" também corresponde a "correu" e "correndo").

Serviço do Azure relevante:

* [Azure Search](https://azure.microsoft.com/services/search/)

**Requisitos típicos**

Armazenamentos de dados não relacionais costumam usar uma arquitetura de armazenamento diferente da usada pelos bancos de dados relacionais. Especificamente, eles tendem a não ter nenhum esquema fixo. Além disso, eles tendem a não dar suporte a transações ou restringir o escopo das transações e geralmente não incluem índices secundários por motivos de escalabilidade.

Veja a seguir uma comparação dos requisitos de cada um dos armazenamentos de dados não relacionais:

| **Requisito** | **Dados de documentos** | | **Dados de família de colunas** | | **Dados de chave/valor** | | **Dados de gráficos** | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Normalização | Desnormalizado | | Desnormalizado | | Desnormalizado | | Normalizado | |
| Esquema | Esquema na leitura | | Famílias de colunas definidas na gravação, esquema de coluna na leitura | | Esquema na leitura | | Esquema na leitura | |
| Consistência (entre transações simultâneas) | Consistência ajustável, garantias no nível do documento | | Garantias de nível da família de colunas | | Garantias no nível da chave | | Garantias no nível do gráfico | |
| Atomicidade (escopo de transação) | Coleção | | Tabela | | Tabela | | Grafo | |
| Estratégia de Bloqueio | Otimista (livre de bloqueio) | | Pessimista (bloqueios de linha) | | Otimista (Etag) | |  | |
| Padrão de acesso | Acesso aleatório | | Agregações em dados altos/largos | | Acesso aleatório | | Acesso aleatório | |
| Indexação | Índices primários e secundários | | Índices primários e secundários | | Somente índice primário | | Índices primários e secundários | |
| Forma dos dados | Documento | | Tabela com famílias de colunas que contém colunas | | Chave e valor | | Gráfico que contém bordas e vértices | |
| Esparsos | Sim | | Sim | | Sim | | Não | |
| Largo (grande quantidade de colunas/atributos) | Sim | | Sim | | Não | | Não | |
| Tamanho do dado | Pequeno (KBs) a médio (alguns MBs) | | Médio (MBs) a grande (alguns GBs) | | Pequeno (KBs) | | Pequeno (KBs) | |
| Escala máxima geral | Muito grande (PBs) | | Muito grande (PBs) | | Muito grande (PBs) | | Grande (TB) | |
| **Requisito** | | **Dados de série temporal** | | **Dados de objetos** | | **Dados de índice externo** | |
| Normalização | | Normalizado | | Desnormalizado | | Desnormalizado | |
| Esquema | | Esquema na leitura | | Esquema na leitura | | Esquema na gravação | |
| Consistência (entre transações simultâneas) | | N/D | | N/D | | N/D | |
| Atomicidade (escopo de transação) | | N/D | | Objeto | | N/D | |
| Estratégia de Bloqueio | | N/D | | Pessimista (bloqueios de blobs) | | N/D | |
| Padrão de acesso | | Acesso aleatório e agregação | | Acesso sequencial | | Acesso aleatório | |
| Indexação | | Índices primários e secundários | | Somente índice primário | | N/D | |
| Forma dos dados | | Tabular | | Blob e metadados | | Documento | |
| Esparsos | | Não | | N/D | | Não | |
| Largo (grande quantidade de colunas/atributos) | | Não | | Sim | | Sim | |
| Tamanho do dado | | Pequeno (KBs) | | Grande (GBs) a muito grandes (TBs) | | Pequeno (KBs) | |
| Escala máxima geral | | Grande (alguns TBs) | | Muito grande (PBs) | | Grande (alguns TBs) | |

**Dados não relacionais e NoSQL**

Cosmos DB

Armazenamento de Blobs

Data Lake

Um *banco de dados não relacional* é um banco de dados que não usa o esquema de tabela de linhas e colunas encontrado na maioria dos sistemas de banco de dados tradicionais. Em vez disso, os bancos de dados não relacionais usam um modelo de armazenamento otimizado para os requisitos específicos do tipo de dados que está sendo armazenado. Por exemplo, os dados podem ser armazenados como pares chave/valor simples, como documentos JSON ou como um gráfico que consiste em bordas e vértices.

O que esses armazenamentos de dados têm em comum é que eles não usam um [modelo relacional](https://docs.microsoft.com/pt-br/azure/architecture/data-guide/). Além disso, eles tendem a ser mais específicos no tipo de dados ao qual dão suporte e no modo como os dados podem ser consultados. Por exemplo, os armazenamentos de dados de série temporal são otimizados para consultas em sequências de dados baseadas em tempo. No entanto, os armazenamentos de dados de grafo são otimizados para explorar as relações ponderadas entre entidades. Nenhum dos dois formatos será bem generalizado para a tarefa de gerenciamento de dados transacionais.

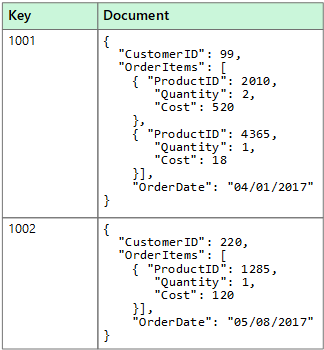
O termo *NoSQL* refere-se aos armazenamentos de dados que não usam o SQL para consultas. Em vez disso, os armazenamentos de dados usam outras linguagens de programação e constructos para consultar os dados. Na prática, "NoSQL" significa "banco de dados não relacionais", mesmo que muitos desses bancos de dados deem suporte a consultas compatíveis com SQL. No entanto, a estratégia de execução de consulta subjacente é geralmente muito diferente da maneira como um RDBMS tradicional executa a mesma consulta SQL.

As seções a seguir descrevem as principais categorias de banco de dados não relacional ou NoSQL.

**Armazenamentos de dados de documentos**

Um armazenamento de dados de documento gerencia um conjunto de campos de cadeia de caracteres nomeados e valores de dados de objeto em uma entidade conhecida como *documento*. Normalmente, esses repositórios de dados armazenam dados na forma de documentos JSON. Cada valor de campo pode ser um item escalar, como um número ou um elemento composto, como uma lista ou uma coleção de pai-filho. Os dados nos campos de um documento podem ser codificados de várias maneiras, incluindo XML, YAML, JSON, BSON ou, até mesmo, armazenados como texto sem formatação. Os campos nos documentos são expostos ao sistema de gerenciamento de armazenamento, permitindo que um aplicativo consulte e filtre dados utilizando os valores nesses campos.

Normalmente, um documento contém todos os dados de uma entidade. Quais itens constituem uma entidade são específicos do aplicativo. Por exemplo, uma entidade pode conter os detalhes de um cliente, um pedido ou uma combinação de ambos. Um único documento pode conter informações que serão distribuídas em várias tabelas relacionais em um RDBMS. Um repositório de documentos não requer que todos os documentos tenham a mesma estrutura. Essa abordagem em forma livre oferece grande flexibilidade. Por exemplo, os aplicativos podem armazenar dados diferentes em documentos, em resposta a uma alteração nos requisitos de negócios.



O aplicativo pode recuperar documentos utilizando a chave de documento. A chave é um identificador exclusivo do documento, que geralmente tem um hash, para ajudar a distribuir os dados de maneira uniforme. Alguns bancos de dados de documentos criam a chave do documento automaticamente. Outros permitem que você especifique um atributo do documento a ser usado como chave. O aplicativo também pode consultar documentos com base no valor de um ou mais campos. Alguns bancos de dados de documentos dão suporte à indexação para facilitar a pesquisa rápida de documentos com base em um ou mais campos indexados.

Muitos bancos de dados de documentos fornecem suporte a atualizações in loco, permitindo que um aplicativo modifique os valores de campos específicos em um documento sem regravar o documento inteiro. As operações de leitura e gravação em vários campos em um só documento normalmente são atômicas.

Serviço do Azure relevante:

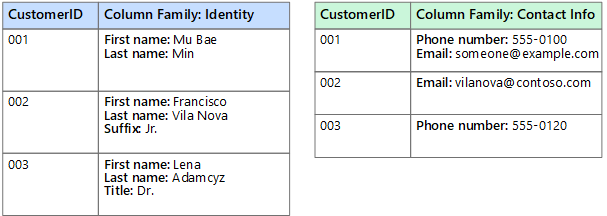
* [Azure Cosmos DB](https://azure.microsoft.com/services/cosmos-db/)

**Armazenamentos de dados de colunas**

Um armazenamento de dados de colunas ou de família de colunas organiza os dados em colunas e linhas. Em sua forma mais simples, um armazenamento de dados de família de colunas pode parecer muito semelhante a um banco de dados relacional, pelo menos conceitualmente. O poder real de um banco de dados de família de colunas está em sua abordagem desnormalizada para a estruturação de dados esparsos, que deriva da abordagem orientada por coluna para armazenar dados.

É possível pensar em um armazenamento de dados de família de colunas como contendo dados de tabela com linhas e colunas, mas as colunas são divididas em grupos conhecidos como famílias de colunas. Cada família de colunas contém um conjunto de colunas que estão logicamente relacionadas e geralmente são recuperadas ou manipuladas como uma unidade. Outros dados acessados separadamente podem ser armazenados em famílias de colunas separadas. Dentro de uma família de colunas, novas colunas podem ser adicionadas dinamicamente e as linhas podem ser esparsas (ou seja, uma linha não precisa ter um valor para cada coluna).

O diagrama a seguir mostra um exemplo com duas famílias de colunas, Identity e Contact Info. Os dados de uma única entidade têm a mesma chave de linha em cada família de colunas. Essa estrutura, em que as linhas de determinado objeto em uma família de colunas podem variar dinamicamente, é um benefício importante da abordagem de família de colunas, que torna essa forma de armazenamento de dados altamente adequada para armazenar dados com esquemas variados.



Ao contrário de um repositório de valor/chave ou um banco de dados de documentos, a maioria dos bancos de dados de família de colunas armazenam dados fisicamente em ordem de chave, em vez de computar um hash. A chave de linha é considerada o índice primário e permite o acesso baseado em chave por meio de uma chave específica ou um intervalo de chaves. Algumas implementações permitem que você crie índices secundários em colunas específicas de uma família de colunas. Os índices secundários permitem recuperar dados por valor de colunas, em vez da chave de linha.

No disco, todas as colunas de uma família de colunas são armazenadas juntas no mesmo arquivo, com um número específico de linhas em cada arquivo. Com conjuntos de dados grandes, esse método cria um benefício de desempenho, reduzindo a quantidade de dados que precisam ser lidos do disco quando apenas algumas colunas são consultadas juntas por vez.

As operações de leitura e gravação para uma linha normalmente são atômicas em uma só família de colunas, embora algumas implementações forneçam atomicidade em toda a linha, abrangendo várias famílias de colunas.

Serviço do Azure relevante:

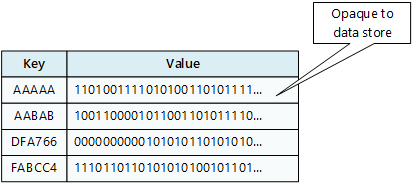
* [API do Cassandra do Cosmos DB](https://docs.microsoft.com/pt-br/azure/cosmos-db/cassandra-introduction)
* [HBase no HDInsight](https://docs.microsoft.com/pt-br/azure/hdinsight/hdinsight-hbase-overview)

**Armazenamentos de dados de chave/valor**

Um armazenamento de valor/chave é essencialmente uma tabela de hash grande. Você associa cada valor de dados a uma chave exclusiva e o armazenamento de valor/chave usa essa chave para armazenar os dados, utilizando uma função de hash apropriada. A função de hash é selecionada para fornecer uma distribuição uniforme de chaves de hash em todo o armazenamento de dados.

A maioria dos armazenamentos de valor/chave fornecem suporte apenas às operações de exclusão, inserção e consulta simples. Para modificar um valor (parcial ou completamente), um aplicativo deverá substituir os dados existentes para todo o valor. Na maioria das implementações, a leitura ou gravação de um único valor é uma operação atômica. Se o valor for grande, a gravação poderá demorar algum tempo.

Um aplicativo pode armazenar dados arbitrários como um conjunto de valores, embora alguns armazenamentos de valor/chave imponham limites ao tamanho máximo de valores. Os valores armazenados são opacos para o software do sistema de armazenamento. Quaisquer informações de esquema deverão ser fornecidas e interpretadas pelo aplicativo. Essencialmente, os valores são blobs e o armazenamento de valor/chave simplesmente recupera ou armazena o valor por chave.



Armazenamentos de chave/valor são altamente otimizados para aplicativos que realizam pesquisas simples usando o valor da chave ou por um intervalo de chaves, mas são menos adequados para sistemas que precisam consultar dados em tabelas diferentes de chaves/valores, como a união de dados em várias tabelas.

Os armazenamentos de chave/valor também não são otimizados para cenários em que a consulta ou a filtragem por valores não chave é importante, em vez de realizar pesquisas baseadas somente em chaves. Por exemplo, com um banco de dados relacional, você pode encontrar um registro usando uma cláusula WHERE para filtrar as colunas não chave, mas armazenamentos de chave/valores geralmente não têm esse tipo de funcionalidade de pesquisa de valores ou se têm, é preciso fazer um exame lento de todos os valores.

Um armazenamento de valor/chave pode ser extremamente escalável, pois o armazenamento de dados pode facilmente distribuir dados em vários nós em computadores separadas.

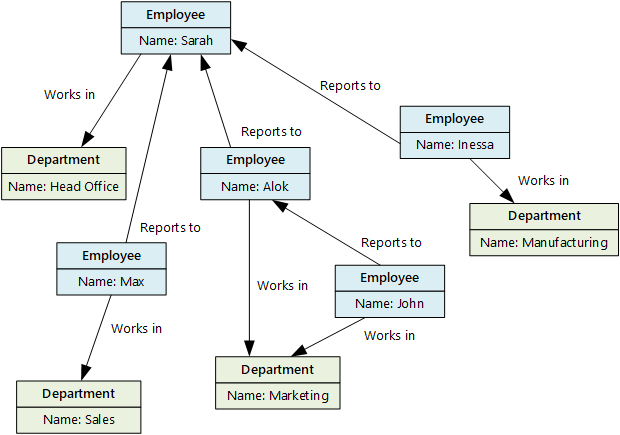
Serviços do Azure relevantes:

* [API de Tabela do Azure Cosmos DB](https://docs.microsoft.com/pt-br/azure/cosmos-db/table-introduction)
* [Cache Redis do Azure](https://azure.microsoft.com/services/cache/)
* [Armazenamento de Tabelas do Azure](https://azure.microsoft.com/services/storage/tables/)

**Armazenamentos de dados de gráficos**

Um armazenamento de dados de gráficos armazena dois tipos de informações: nós e bordas. Nós representam entidades e bordas especificam as relações entre essas entidades. Ambos os nós e as bordas podem ter propriedades que fornecem informações sobre esse nó ou borda, semelhante às colunas em uma tabela. As bordas também podem ter uma direção indicando a natureza do relacionamento.

O objetivo de um armazenamento de dados de gráficos é permitir que um aplicativo execute com eficácia consultas que atravessam a rede de nós e bordas e analise as relações entre entidades. O diagrama a seguir mostra os dados de pessoal de uma organização estruturados como um gráfico. As entidades são funcionários e departamentos, e as bordas indicam os relacionamentos de relatórios e o departamento em que os funcionários trabalham. Nesse grafo, as setas nas bordas mostram a direção das relações.



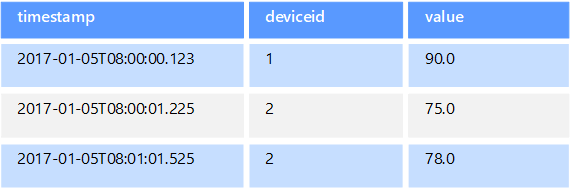
Essa estrutura simplifica a execução de consultas como "Encontrar todos os funcionários que são subordinados diretos ou indiretos de Marina" ou "Quem trabalha no mesmo departamento que Júlio?". Para grafos grandes com muitas entidades e relações, você pode executar análises complexas rapidamente. Muitos bancos de dados de gráficos fornecem uma linguagem de consulta que você pode utilizar para percorrer uma rede de relacionamentos de forma eficiente.

Serviço do Azure relevante:

* [API do Graph do Azure Cosmos DB](https://docs.microsoft.com/pt-br/azure/cosmos-db/graph-introduction)

**Armazenamentos de dados de série temporal**

Os dados de série temporal são um conjunto de valores organizados por tempo e um armazenamento de dados de série temporal é otimizado para esse tipo de dados. Os armazenamentos de dados de série temporal precisam dar suporte a um número muito alto de gravações, pois geralmente coletam grandes quantidades de dados em tempo real de uma grande variedade de fontes. Os armazenamentos de dados de série temporal são otimizados para armazenar dados telemétricos. Os cenários incluem sensores de IoT ou contadores de sistemas/aplicativos. As atualizações são raras e as exclusões geralmente são feitas como operações em massa.



Embora os registros gravados em um banco de dados de série temporal sejam geralmente pequenos, muitas vezes, há um grande número de registros e o tamanho total dos dados pode aumentar rapidamente. Os armazenamentos de dados de série temporal também manipulam dados fora de ordem e de chegada tardia, a indexação automática de pontos de dados e otimizações de consultas descritas em termos de janelas de tempo. Esse último recurso permite a execução de consultas em milhões de pontos de dados e em vários fluxos de dados com rapidez, para dar suporte a visualizações de série temporal, que é uma maneira comum pela qual os dados de séries temporal são consumidos.

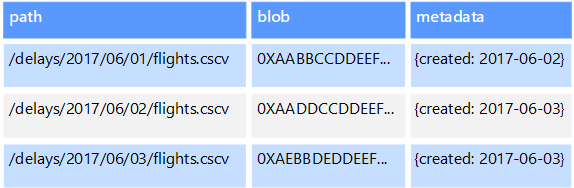
Para obter mais informações, consulte [Soluções de série temporal](https://docs.microsoft.com/pt-br/azure/architecture/data-guide/scenarios/time-series)

Serviços do Azure relevantes:

* [Azure Time Series Insights](https://azure.microsoft.com/services/time-series-insights/)
* [OpenTSDB com HBase no HDInsight](https://docs.microsoft.com/pt-br/azure/hdinsight/hdinsight-hbase-overview)

**Armazenamentos de dados de objetos**

Os armazenamentos de dados de objetos são otimizados para armazenar e recuperar grandes objetos binários ou blobs como imagens, arquivos de texto, fluxos de áudio e vídeo, grandes documentos e objetos de dados de aplicativos e imagens de disco de máquina virtual. Um objeto consiste nos dados armazenados, em alguns metadados e uma ID exclusiva para acessar o objeto. Os armazenamentos de objetos foram projetados para dar suporte a arquivos que são individualmente muito grandes, além de fornecer grandes quantidades de armazenamento total para gerenciar todos os arquivos.



Alguns armazenamentos de dados de objetos replicam determinado blob em vários nós de servidor, o que permite rápidas leituras paralelas. Esse processo, por sua vez, habilita a consulta de expansão dos dados contidos em arquivos grandes, pois vários processos, normalmente em execução em diferentes servidores, podem consultar o arquivo de dados grande simultaneamente.

Um caso especial de armazenamentos de dados de objetos é o compartilhamento de arquivos de rede. O uso de compartilhamentos de arquivos permite que os arquivos sejam acessados em uma rede usando protocolos de rede padrão como o protocolo SMB. Considerando os mecanismos apropriados de segurança e controle de acesso simultâneo, compartilhar dados dessa forma pode permitir que os serviços distribuídos forneçam acesso a dados altamente escalonáveis para operações básicas de baixo nível, como solicitações de leitura e gravação simples.

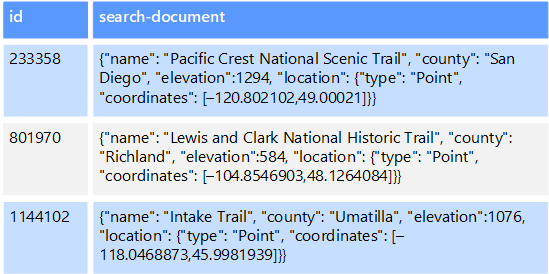
Serviços do Azure relevantes:

* [Armazenamento de Blobs do Azure](https://azure.microsoft.com/services/storage/blobs/)
* [Repositório Azure Data Lake](https://azure.microsoft.com/services/data-lake-store/)
* [Armazenamento de Arquivos do Azure](https://azure.microsoft.com/services/storage/files/)

**Armazenamentos de dados de índice externo**

Os armazenamentos de dados de índice externo fornecem a capacidade de pesquisar informações mantidas em outros armazenamentos de dados e serviços. Um índice externo atua como um índice secundário para qualquer armazenamento de dados e pode ser usado para indexar grandes volumes de dados e fornecer acesso quase em tempo real a esses índices.

Por exemplo, talvez você tenha arquivos de texto armazenados em um sistema de arquivos. Encontrar um arquivo por seu arquivo de caminho é rápido, mas pesquisar com base no conteúdo do arquivo exige um exame de todos os arquivos, o que é lento. Um índice externo permite criar índices de pesquisa secundária e, em seguida, localizar rapidamente o caminho para os arquivos que correspondem aos critérios. Outro aplicativo de exemplo de um índice externo são armazenamentos de chave/valor que somente indexam pela chave. Você pode criar um índice secundário com base nos valores dos dados e pesquisar rapidamente a chave que identifica exclusivamente cada item correspondente.



Os índices são criados pela execução de um processo de indexação. Isso pode ser feito usando um modelo de pull, disparado pelo armazenamento de dados ou usando um modelo de push, iniciado pelo código do aplicativo. Os índices podem ser multidimensionais e dar suporte a pesquisas de texto livre em grandes volumes de dados de texto.

Os armazenamentos de dados de índice externo costumam ser usados para dar suporte às pesquisas de texto completo e baseadas na Web. Nesses casos, a pesquisa pode ser exata ou difusa. Uma pesquisa difusa localiza documentos que correspondem um conjunto de termos e calcula a forma como eles correspondem. Alguns índices externos também dão suporte à análise linguística que pode retornar correspondências com base em sinônimos, expansões de gênero (por exemplo, correspondência de "cachorros" com "animais de estimação") e lematização (por exemplo, a pesquisa de "correr" também corresponde a "correu" e "correndo").

Serviço do Azure relevante:

* [Azure Search](https://azure.microsoft.com/services/search/)

**Requisitos típicos**

Armazenamentos de dados não relacionais costumam usar uma arquitetura de armazenamento diferente da usada pelos bancos de dados relacionais. Especificamente, eles tendem a não ter nenhum esquema fixo. Além disso, eles tendem a não dar suporte a transações ou restringir o escopo das transações e geralmente não incluem índices secundários por motivos de escalabilidade.

Veja a seguir uma comparação dos requisitos de cada um dos armazenamentos de dados não relacionais:

| **Requisito** | **Dados de documentos** | | **Dados de família de colunas** | | **Dados de chave/valor** | | **Dados de gráficos** | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Normalização | Desnormalizado | | Desnormalizado | | Desnormalizado | | Normalizado | |
| Esquema | Esquema na leitura | | Famílias de colunas definidas na gravação, esquema de coluna na leitura | | Esquema na leitura | | Esquema na leitura | |
| Consistência (entre transações simultâneas) | Consistência ajustável, garantias no nível do documento | | Garantias de nível da família de colunas | | Garantias no nível da chave | | Garantias no nível do gráfico | |
| Atomicidade (escopo de transação) | Coleção | | Tabela | | Tabela | | Grafo | |
| Estratégia de Bloqueio | Otimista (livre de bloqueio) | | Pessimista (bloqueios de linha) | | Otimista (Etag) | |  | |
| Padrão de acesso | Acesso aleatório | | Agregações em dados altos/largos | | Acesso aleatório | | Acesso aleatório | |
| Indexação | Índices primários e secundários | | Índices primários e secundários | | Somente índice primário | | Índices primários e secundários | |
| Forma dos dados | Documento | | Tabela com famílias de colunas que contém colunas | | Chave e valor | | Gráfico que contém bordas e vértices | |
| Esparsos | Sim | | Sim | | Sim | | Não | |
| Largo (grande quantidade de colunas/atributos) | Sim | | Sim | | Não | | Não | |
| Tamanho do dado | Pequeno (KBs) a médio (alguns MBs) | | Médio (MBs) a grande (alguns GBs) | | Pequeno (KBs) | | Pequeno (KBs) | |
| Escala máxima geral | Muito grande (PBs) | | Muito grande (PBs) | | Muito grande (PBs) | | Grande (TB) | |
| **Requisito** | | **Dados de série temporal** | | **Dados de objetos** | | **Dados de índice externo** | |
| Normalização | | Normalizado | | Desnormalizado | | Desnormalizado | |
| Esquema | | Esquema na leitura | | Esquema na leitura | | Esquema na gravação | |
| Consistência (entre transações simultâneas) | | N/D | | N/D | | N/D | |
| Atomicidade (escopo de transação) | | N/D | | Objeto | | N/D | |
| Estratégia de Bloqueio | | N/D | | Pessimista (bloqueios de blobs) | | N/D | |
| Padrão de acesso | | Acesso aleatório e agregação | | Acesso sequencial | | Acesso aleatório | |
| Indexação | | Índices primários e secundários | | Somente índice primário | | N/D | |
| Forma dos dados | | Tabular | | Blob e metadados | | Documento | |
| Esparsos | | Não | | N/D | | Não | |
| Largo (grande quantidade de colunas/atributos) | | Não | | Sim | | Sim | |
| Tamanho do dado | | Pequeno (KBs) | | Grande (GBs) a muito grandes (TBs) | | Pequeno (KBs) | |
| Escala máxima geral | | Grande (alguns TBs) | | Muito grande (PBs) | | Grande (alguns TBs) | |

Fonte: https://docs.microsoft.com/pt-br/azure/architecture/data-guide/big-data/non-relational-datahttps://docs.microsoft.com/pt-br/azure/architecture/data-guide/big-data/non-relational-data