Azure Table storage overview

Tip

The content in this article applies to the original Azure Table storage. However, there is now a premium offering for table storage, the Azure Cosmos DB Table API that offers throughput-optimized tables, global distribution, and automatic secondary indexes. To learn more and try out the premium experience, please check out [Azure Cosmos DB Table API](https://aka.ms/premiumtables).

Azure Table storage is a service that stores structured NoSQL data in the cloud, providing a key/attribute store with a schemaless design. Because Table storage is schemaless, it's easy to adapt your data as the needs of your application evolve. Access to Table storage data is fast and cost-effective for many types of applications, and is typically lower in cost than traditional SQL for similar volumes of data.

You can use Table storage to store flexible datasets like user data for web applications, address books, device information, or other types of metadata your service requires. You can store any number of entities in a table, and a storage account may contain any number of tables, up to the capacity limit of the storage account.

What is Table storage

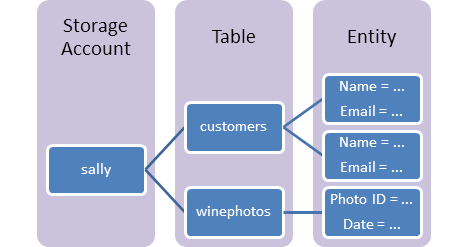
Azure Table storage stores large amounts of structured data. The service is a NoSQL datastore which accepts authenticated calls from inside and outside the Azure cloud. Azure tables are ideal for storing structured, non-relational data. Common uses of Table storage include:

* Storing TBs of structured data capable of serving web scale applications
* Storing datasets that don't require complex joins, foreign keys, or stored procedures and can be denormalized for fast access
* Quickly querying data using a clustered index
* Accessing data using the OData protocol and LINQ queries with WCF Data Service .NET Libraries

You can use Table storage to store and query huge sets of structured, non-relational data, and your tables will scale as demand increases.

Table storage concepts

Table storage contains the following components:



* **URL format:** Code addresses tables in an account using this address format:  
  http://<storage account>.table.core.windows.net/<table>

You can address Azure tables directly using this address with the OData protocol. For more information, see [OData.org](http://www.odata.org/).

* **Storage Account:** All access to Azure Storage is done through a storage account. See [Azure Storage Scalability and Performance Targets](https://docs.microsoft.com/en-in/azure/storage/common/storage-scalability-targets) for details about storage account capacity.
* **Table**: A table is a collection of entities. Tables don't enforce a schema on entities, which means a single table can contain entities that have different sets of properties. The number of tables that a storage account can contain is limited only by the storage account capacity limit.
* **Entity**: An entity is a set of properties, similar to a database row. An entity can be up to 1MB in size.
* **Properties**: A property is a name-value pair. Each entity can include up to 252 properties to store data. Each entity also has three system properties that specify a partition key, a row key, and a timestamp. Entities with the same partition key can be queried more quickly, and inserted/updated in atomic operations. An entity's row key is its unique identifier within a partition.

+

For details about naming tables and properties, see [Understanding the Table Service Data Model](https://docs.microsoft.com/en-us/rest/api/storageservices/Understanding-the-Table-Service-Data-Model)