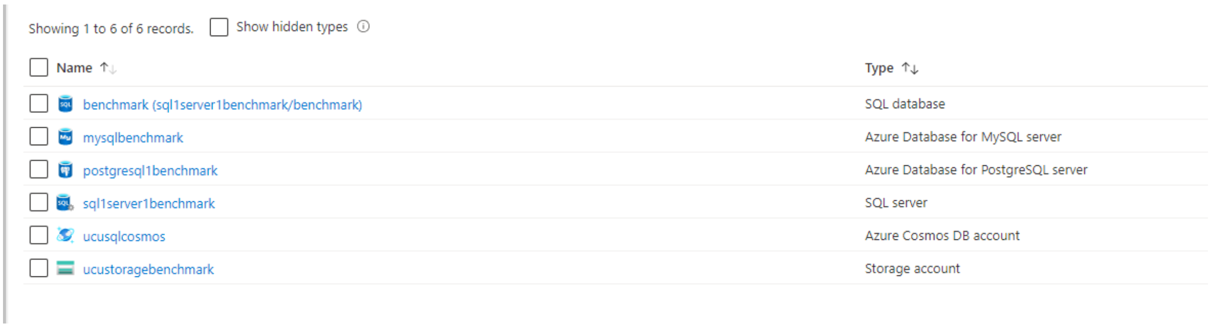
Azure, as a cloud platform, has a lot of storage services, including various SQL databases. This document contains a comparison table with performance results for multi-model database service CosmosDB, using its SQL API, and few most popular SQL databases.



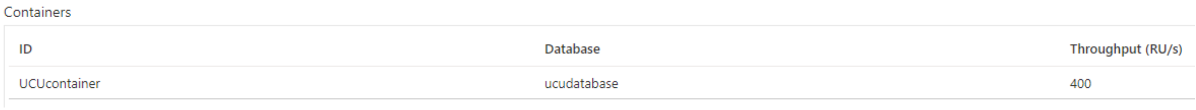
Showing 1 to 6 of 6 records. □ Show hidden types ⓘ

□ Name ↑↓	Type ↑↓
□ benchmark (sql1server1benchmark/benchmark)	SQL database
□ mysqlbenchmark	Azure Database for MySQL server
□ postgresql1benchmark	Azure Database for PostgreSQL server
□ sql1server1benchmark	SQL server
□ ucusqlcosmos	Azure Cosmos DB account
□ ucustoragebenchmark	Storage account

Code in the repository contains Jupiter notebooks to populate databases with test data.

Test Data	CosmosDB (SQL API)	PosgreSQL	SQL Server	MySQL
Dataset: file with 48439 rows, 21 columns, 4.4 MB https://globaldatalab.org/assets/2019/09/SHDI%20Complete%203.0.csv	Insert from Python app - near 15 minutes Read data - 7.4 sec	Insert into table - 45.1 s Read data - 59.3 s	Insert into table - 52 s Read data - 45 s	Insert into table - 55 s Read data - 70 s

CosmosDB (SQL API) being the most expensive has no reliable Python client. Because of that inserting consumes a lot of time using the loop approach. Complexity is O(n^). This DB service shows better results on high-concurrent workloads, rather than huge bulk data insertion.
Despite the loop approach for inserting data, reading was fast, comparing to other database services.
The lowest possible capacity was used for CosmosDB (Throughput (RU/s) - 400).



Containers

ID	Database	Throughput (RU/s)
UCUcontainer	ucudatabase	400

PostgreSQL, SQL Server, MySQL showed near the same result, however, PostgreSQL was slightly faster. Those databases have comparably similar capacities at the lowest level.