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

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

PostgreSQL, SQL Server showed near the same result, however, MySQL was much faster.

Test Data	CosmosDB (SQL API)	PosgreSQL	SQL Server	MySQL (InnoDB)
Dataset: file with 48439 rows, 21 columns, 4.4 MB For benchmark purposes, dataframe with 1000 rows was created basing on the original CSV. https://globaldatalab.org/assets/2019/09/SHDI%20Complete%203.0.csv	**Iterative insert of 1000** rows - 335.3 s **Read data by key** - 0.3 s -fastest read by key	**Iterative insert of 1000 rows** - 525.8 s **Read data by key** - 0.5 s	**Iterative insert of 1000 rows** - 477.8 s **Read data by key** - 0.3s -fastest read by key	**Iterative insert of 1000 rows** - 182.1 s **Read data by key** -0.8 s -fastest insert

Example of database structures	{	CREATE TABLE	CREATE TABLE	CREATE TABLE

```
{
    "id": "0",
    "iso_code": "AFG",
    "country":
"Afghanistan",
    "year": "2002",
    "GDLCODE": "Ar101",
    "level": "Subnat",
    "region": "Central",
    "shi": ".454",
    "healthindex": "58",
    "incindex": ".394",
    "edindex": ".409",
    "lifexp": "57.7",
    "lgnic": "7.213",
    "esch": "9.1",
    "msch": "4.7",
    "pop": "3443.952",
    "_rid": "",
    "_selfAA==/",
    "_etag":
"\"b800871e-0000-070
0-0000-6128ef410000\
"",
    "_attachments":
"attachments/",
    "_ts":
1630072641
}
```

PostgreSQL
```
CREATE TABLE
public.benchmark
(
    id text,
    iso_code text,
    country text,
    year text,
    gdlcode text,
    level text,
    region text,
    shdi text,
    healthindex text,
    incindex text,
    edindex text,
    lifexp text,
    lgnic text,
    esch text,
    msch text
)
```

SQL Server
```
CREATE TABLE
[dbo].[Benchmark](
[id] [varchar](max) NULL,
[iso_code] [varchar](max)
NULL,
[country] [varchar](max) NULL,
[year] [varchar](max) NULL,
[GDLCODE] [varchar](max)
NULL,
[level] [varchar](max) NULL,
[region] [varchar](max) NULL,
[shdi] [varchar](max) NULL,
[healthindex] [varchar](max)
NULL,
[incindex] [varchar](max)
NULL,
[edindex] [varchar](max) NULL,
[lifexp] [varchar](max) NULL,
[lgnic] [varchar](max) NULL,
[esch] [varchar](max) NULL,
[msch] [varchar](max) NULL
) ON [PRIMARY]
TEXTIMAGE_ON [PRIMARY]
GO
```

MySQL
```
CREATE TABLE
Benchmark(
`id` Longtext NULL,
`iso_code` Longtext NULL,
`country` Longtext NULL,
`year` Longtext NULL,
`GDLCODE` Longtext NULL,
`level` Longtext NULL,
`region` Longtext NULL,
`shdi` Longtext NULL,
`healthindex` Longtext NULL,
`incindex` Longtext NULL,
`edindex` Longtext NULL,
`lifexp` Longtext NULL,
`lgnic` Longtext NULL,
`esch` Longtext NULL,
`msch` Longtext NULL
);
```

SQL Dbs settings:

To reach similar condition for all SQL Dbs, I used tables with max string column types for each database:

text	for PosgreSQL,
[varchar](max)	for SQL Server,
Longtext	for MySQL (InnoDB)

Each result was calculated basing on 3 executions.
Each database was created using the lowest possible performance capabilities in the cloud.
Data was inserted without using bufferisation, however buffesition is possible to use for MySQL and PostgreSQL.

CosmosDB (SQL API) settings:

1) **Cosmos has default consistency level named as session** -

Check Microsoft animation for more details
https://docs.microsoft.com/en-us/azure/cosmos-db/consistency-levels

2) The lowest possible capacity was used for CosmosDB (Throughput (RU/s) - 400).

Containers

ID	Database	Throughput (RU/s)
UCUcontainer	ucudatabase	400

3) No replication was enabled for benchmark