



УНИВЕРЗИТЕТ У НОВОМ САДУ  
ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА У  
НОВОМ САДУ

---



CLOUD COMPUTING U ELEKTROENERGETSKIM SISTEMIMA

***Osnove Microsoft WindowsAzure servisa***

**-TEORIJA-**

Novi Sad, 2020

### ***Vežba 3 – Windows Azure Storage servisi (Table)***

Windows Azure storage servis je servis koji nam nudi Microsoft za čuvanje podataka. Garantovano je da će podaci biti 100% sačuvani, međutim ne i da će biti dostupni. Ovo se postiže replikacijom. Pri upisu podataka u storage servis on se sinhrono replicira na drugu mašinu u istom datacentru pa zatim i asinhrono u drugi datacentar. Storage servis se vezuje za nalog koji moramo da imamo kako bismo ga koristili (na vežbama koristimo storage emulator).

Postoje *Blob*, *Table* i *Queue* storage servisi:

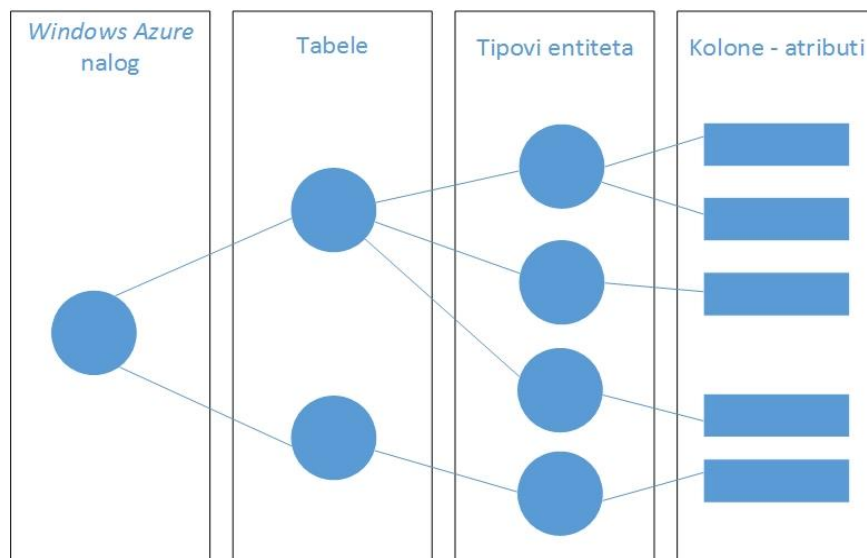
1. **Blob** – koristi se za čuvanje velikih binarnih objekata (*Binary Large Object*). To mogu biti slike, video zapisi, neki dokumenti i slično. Ovi podaci su organizovani u kontejnerima. Za interakciju sa blob storage servisom koristimo klasu *CloudBlobClient*.
2. **Table** – koristi se za čuvanje struktuiranih podataka. Ovo nije relaciona baza već noSQL, što znači da se u okviru jedne tabele mogu čuvati entiteti različitih struktura. Entitet je objekat koji se čuva u tabeli i predstavlja skup atributa klase. Jedan entitet je zapisan u jednom redu tabele.
3. **Queue** – predstavlja asinhroni komunikacioni kanal za povezivanje servisa. Komunikacija se vrši postavljanjem poruka u red i njihovim preuzimanjem. Dizajniran tako da podrži praktično neograničen broj poruka.

Jedna od bitnih osobina storage servisa je njihova mogućnost skaliranja. Ovo je moguće jer se oslanjaju na logiku particionisanja. Svaki podatak prilikom smeštanja u storage servis pripadne nekoj particiji u sistemu. Te particije se koriste posle za skaliranje. Ako na primer imamo na raspolaganju 3 virtuelne mašine za storage servis, u zavisnosti od veličine i load-a, particije se mogu premeštati sa jedne na drugu virtuelnu mašinu. Ovim se postiže dobar protok i garancija da će podatak biti sačuvan. Za spomenuta tri servisa, particije se određuju na sledeći način:

1. **Blob** – svaki blob je posebna particija
2. **Table** – korisnik odlučuje kako će podaci biti raspoređeni po particijama
3. **Queue** – svaki queue je posebna particija

## Table storage

Na slici 1 je prikazan primer organizacije *Table storage*-a. Sve kreće od naloga, za njega se vezuju tabele kojima pripadaju isti ili različiti tipovi entiteta koji imaju određene kolone.



Slika 1. Table storage

Svaki entitet koji je upisan u tabelu sadrži tri podrazumevana polja:

- PartitionKey – koristi se za distribuiranje entiteta na različite virtuelne mašine u *cloud* okruženju.
- RowKey – identifikator entiteta u okviru particije kojoj pripada
- Timestamp – automatski se očuvava od strane table servisa

PartitionKey i RowKey zajedno određuju ključ entiteta i jedinstveno određuju entitet u okviru jedne tabele. Kao što je opisano gore, kod table servisa korisnici sami određuju particije i raspoređuju podatke u njima. Prilikom definisanja polja PartitionKey treba imati u vidu da isti PartitionKey imaju samo entiteti za koje se očekuje da će im se pristupati zajedno. Takođe treba imati u vidu da se gubi mogućnost skaliranja što može dovesti do problema ukoliko se napravi samo jedna particija jer smo u tom slučaju limitirani veličinom mašine koja se koristi za storage. Takođe ukoliko svi entiteti pripadaju različitoj particiji može doći do usporenja prilikom pretrage i pristupanju storage servisu.