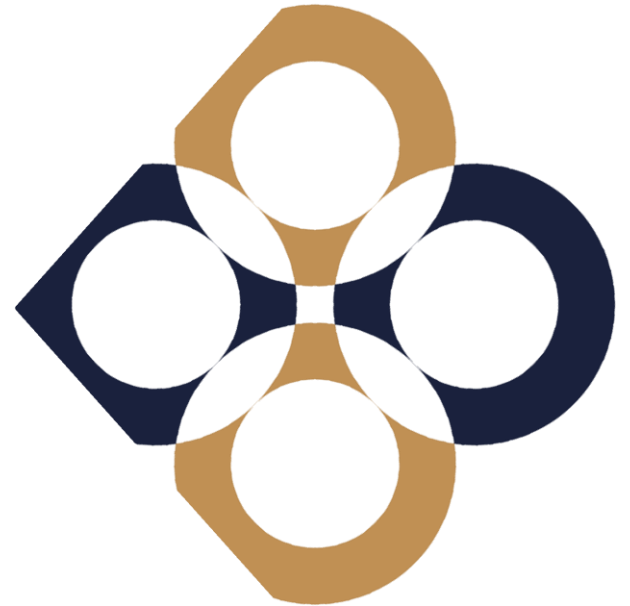


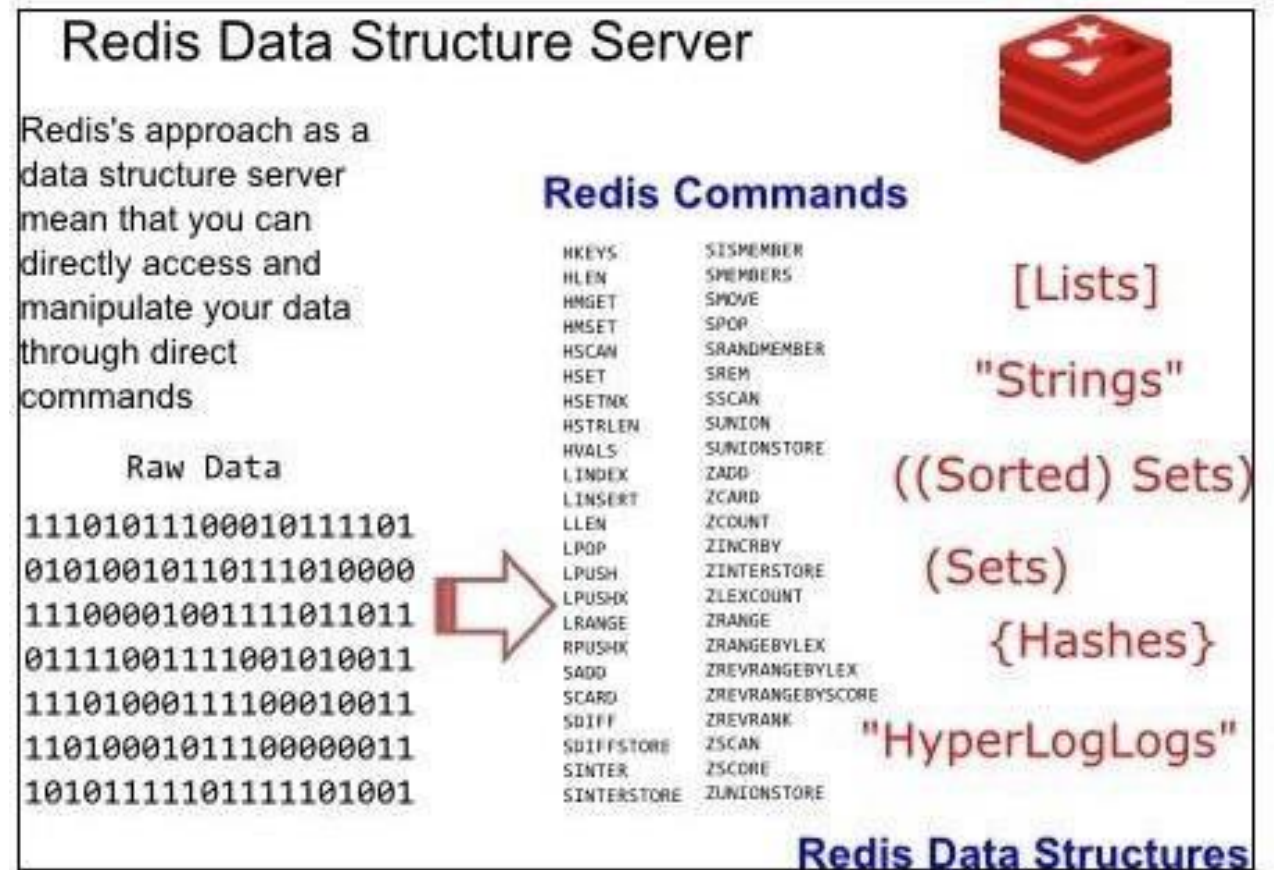
# Adatbázisok gyakorlat 10.

Kulcs-érték adatbázisok



# Redis adattípusok

- ☐ Key – tulajdonság neve\*
- ☐ String – szöveg
- ☐ List – szöveges adatok rendezett listája
- ☐ Hashes – objektumhoz hasonló adattípus, amely objektum kulcsból és kulcs-érték párokból áll
- ☐ Set – egyedi szöveges adatok rendezetlen halmaza
- ☐ Sorted set – egyedi szöveges adatok rendezett halmaza



# Utasítások\* I. – Key, string

## SET kulcs érték [opciók]

Új kulcs-érték pár létrehozása, vagy a meglévő felülírása  
Fontosabb opciók: EX seconds – lejárati idő, NX – csak akkor jön létre a kulcs, ha még nem létezik, GET – a régi érték megjelenítése, amennyiben létezik

## GET kulcs

A kulcs értékének lekérdezése – ha nem létezik, akkor a visszaadott érték üres (nil)

## EXISTS kulcs(ok)

Ellenőrzi, hogy az adott kulcs létezik-e. Egymás után szóközzel elválasztva több kulcs is megadható

## DEL kulcs(ok)

Törli a megadott kulcsot vagy kulcsokat

\* A Redis verziótól függően a felsorolt utasításoknak csak egy része támogatott

# Utasítások I. – Key, string (folytatás)

COPY kulcs1 kulcs2 [logikai adatbázis]  
[REPLACE]

A kulcs1 értékével létrehozza a még nem létező kulcs2-t. A kulcs2 lehet egy másik logikai adatbázisban is. A REPLACE segítségével törölhető a meglévő kulcs2.

MOVE kulcs logikai\_adatbázis

A kulcsot átmozgatja egy másik logikai adatbázisba, amennyiben ott az nem létezik

RENAME kulcsnév újkulcsnév

Új nevet ad a kulcsnak

EXPIRE kulcs seconds

Beállítja a kulcs élettartamát (másodpercben)

TYPE kulcs

Megadja a kulcshoz tartozó érték típusát

# Utasítások I. - Példák

```
C:\Windows\System32\cmd.exe - redis-cli
127.0.0.1:6379>
127.0.0.1:6379>
127.0.0.1:6379>
127.0.0.1:6379> set mainap hetfo
OK
127.0.0.1:6379> get mainap
"hetfo"
127.0.0.1:6379> set mainap hetfo ex 5
OK
127.0.0.1:6379> get mainap
(nil)
127.0.0.1:6379> exists mainap
(integer) 0
127.0.0.1:6379> set mainap kedd
OK
127.0.0.1:6379> exists mainap
(integer) 1
127.0.0.1:6379> del mainap
(integer) 1
127.0.0.1:6379> exists mainap
(integer) 0
```

- ☐ Létrehozunk egy új kulcsot mainap néven, amelynek értéke hetfo
- ☐ Lekérdezzük az értékét, majd beállítjuk a lejáratí idejét 5 másodperce
- ☐ Utána a kulcs törlődik (nil), azaz már nem létezik
- ☐ Ismét létrehozzuk a mainap kulcsot, most értéke kedd
- ☐ Ellenőrizzük, hogy létezik-e, majd töröljük. Utána ismét megnézzük, hogy létezik-e

# Utasítások II. – Hashes

HSET | HMSET kulcs mező érték [mező érték]

Beállítja a mező(k) értékét

HGET kulcs mező

Lekérdezi a kulcs adott mezőjének értékét

HGETALL kulcs

Lekérdezi a kulcs összes mezőjének értékét

HEXISTS kulcs mező

Megadja, hogy létezik-e a kulcs adott mezője

HDEL kulcs mező [kulcs mező]

Törli az adott mező(ke)t.

HKEYS kulcs

Megadja, hogy milyen mezői vannak a kulcsnak

# Utasítások II. - Példák

```
C:\Windows\System32\cmd.exe - redis-cli
127.0.0.1:6379> hmset tanulo nev "Kiss Bela"
szulido 2000.01.01 neptun abcdef
OK
127.0.0.1:6379> hgetall tanulo
1) "nev"
2) "Kiss Bela"
3) "szulido"
4) "2000.01.01"
5) "neptun"
6) "abcdef"
127.0.0.1:6379> type tanulo
hash
127.0.0.1:6379> hkeys tanulo
1) "nev"
2) "szulido"
3) "neptun"
127.0.0.1:6379> hdel tanulo neptun
(integer) 1
127.0.0.1:6379>
```

- ☐ Létrehozzuk a tanulo kulcsot több mezővel (nev, szulido, neptun)
- ☐ Lekérdezzük a tanulo kulcsot
- ☐ Lekérdezzük a kulcs típusát
- ☐ Lekérdezzük a kulcs mezőit
- ☐ Töröljük a tanulo kulcs neptun mezőjét

# Utasítások III. – List

LSET kulcs index érték

Beállítja a lista adott indexű elemének értékét. Az index 0-ról indul. Speciálisan a -1 az utolsó elemre, -2 az utolsó előtti-re utal stb.

LPUSH kulcs elem [elem]

Elem(ek)et szúr be a lista elejére

RPUSH kulcs elem [elem]

Elem(ek)et szúr be a lista végére

LPOP kulcs [n]

Törli és visszaadja a lista első (n) elemét

RPOP kulcs [n]

Törli és visszaadja a lista utolsó (n) elemét

LINSERT kulcs BEFORE | AFTER elem újelem

Új elemet szúr be a lista adott eleme elé vagy után

LRANGE kulcs start stop

Visszaadja a lista elemeit start indextől a stop-ig



# Utasítások III. - Példák

```
C:\Windows\System32\cmd.exe - redis-cli
127.0.0.1:6379>
127.0.0.1:6379> rpush napok hetfo kedd
(integer) 2
127.0.0.1:6379> linsert napok after kedd szerda
(integer) 3
127.0.0.1:6379> lrange napok 0 -1
1) "hetfo"
2) "kedd"
3) "szerda"
127.0.0.1:6379> lpop napok
"hetfo"
127.0.0.1:6379> lrange napok 0 -1
1) "kedd"
2) "szerda"
127.0.0.1:6379>
```

- ☐ Létrehozunk egy napok listát, tartalma: hetfo, kedd
- ☐ Beszúrjuk a szerdát a végére
- ☐ Megjelenítjük a lista elemeit
- ☐ Töröljük az első napot
- ☐ Megjelenítjük az így kapott lista minden elemét

# Utasítások IV. – Set

SADD kulcs elem [elem]

Elem(ek)et ad hozzá a halmazhoz. Ha a halmaz még nem létezik, akkor létrehozza. Ha az elem már létezik, akkor nem kerül be még egyszer a halmazba

SCARD kulcs

Visszaadja a halmaz elemszámát

SPOP kulcs [n]

Töröl és visszaad (n) elemet a halmazból.

SISMEMBER kulcs érték

Visszaadja, hogy az adott érték benne van-e a halmazban

SMEMBERS kulcs

Listázza a halmaz összes elemét

SDIFF | SUNION | SINTER kulcs1 kulcs2

Listázza a halmazok különbségét, unióját, metszetét

SORT kulcs [BY minta] [LIMIT interval]  
[ASC | DESC] [ALPHA]

Rendezi a list, set és ordered set kulcs elemeit

# Utasítások IV. - Példák

```
C:\Windows\System32\cmd.exe - redis-cli
127.0.0.1:6379> sadd evszakok tavasz nyar osz tel
(integer) 4
127.0.0.1:6379> type evszakok
set
127.0.0.1:6379> smembers evszakok
1) "tel"
2) "osz"
3) "tavasz"
4) "nyar"
127.0.0.1:6379> scard evszakok
(integer) 4
127.0.0.1:6379> sadd kedvencevszak nyar
(integer) 1
127.0.0.1:6379> sdiff evszakok kedvencevszak
1) "tavasz"
2) "osz"
3) "tel"
127.0.0.1:6379> sort evszakok alpha
1) "nyar"
2) "osz"
3) "tavasz"
4) "tel"
```

- ☐ Létrehozunk egy halmazt evszakok neven, majd lekérdezzük a típusát
- ☐ Listázzuk a halmaz tartalmát
- ☐ Megjelenítjük a halmaz elemszámát
- ☐ Létrehozunk egy új halmazt kedvencevszak neven
- ☐ Képezzük a két halmaz különbségét
- ☐ Rendezve jelenítjük meg az evszakok halmaz elemeit

# Utasítások V. – Sorted set

ZADD kulcs [opciók] pontszám elem  
[pontszám elem]

Új elem(ek)et ad a sorted set-hez. Ha a set nem létezik, akkor létrehozza. A létező elemek felülíródnak. Opciók: XX – nem ad hozzá új elemet, csak update, LT/GT – csak akkor módosítja a meglévőt, ha az új elem kisebb/nagyobb. A rendezés az adott pontszámok (score) alapján történik

ZCARD kulcs

Visszaadja a sorted set elemszámát

ZCOUNT kulcs min max

Visszaadja a min és max közötti elemek számát

ZINCRBY kulcs n elem

A sorted set adott elemének értékét n-nel növeli

ZINTER | ZUNION | ZDIFF n kulcsok

Visszadja az adott számú halmaz különbségét, unióját, metszetét

ZRANGE kulcs min max [opciók]

Visszadja a sorted set adott indextartományba eső elemeit

# Utasítások V. - Példák

```
C:\Windows\System32\cmd.exe - redis-cli
127.0.0.1:6379>
127.0.0.1:6379>
127.0.0.1:6379> zadd nevsor 1 laci 2 nora 3 zita
(integer) 3
127.0.0.1:6379> type nevsor
zset
127.0.0.1:6379> zcard nevsor
(integer) 3
127.0.0.1:6379> zcount nevsor 2 3
(integer) 2
127.0.0.1:6379> zrange nevsor 0 -1 withscores
1) "laci"
2) "1"
3) "nora"
4) "2"
5) "zita"
6) "3"
127.0.0.1:6379> zincrby nevsor 5 nora
"7"
127.0.0.1:6379> zrange nevsor 0 -1
1) "laci"
2) "zita"
3) "nora"
127.0.0.1:6379>
```

- ☐ Létrehozunk egy sorted set-et nevsor néven (laci, nora, zita)
- ☐ Lekérdezzük a nevsor típusát, elemszámát
- ☐ Lekérdezzük, hogy hány elem van, amelynek pontszáma 2 és 3 közötti
- ☐ Lekérdezzük a nevsor minden pontszámát és elemét
- ☐ Növeljük zita pontszámát 5-tel, utána ismét listázzuk a nevsor elemeit  
→ a sorrend megváltozik

# Lekérdezések

SELECT \* FROM tábla – minden táblára

SCAN 0 – Az adatbázisban lévő összes kulcsot megadja

SELECT \* FROM tábla WHERE ... LIKE ...

SCAN | SSCAN | HSCAN | ZSCAN iterátor  
[MATCH minta] [COUNT n] [TYPE típus]

SELECT \* FROM tábla WHERE id=érték

HGETALL kulcs

SELECT SUM(), AVG(), MIN(), MAX()  
FROM tábla

Nincs beépített lehetőség – csak program segítségével (ciklusokkal) oldható meg

SELECT FROM ... GROUP BY

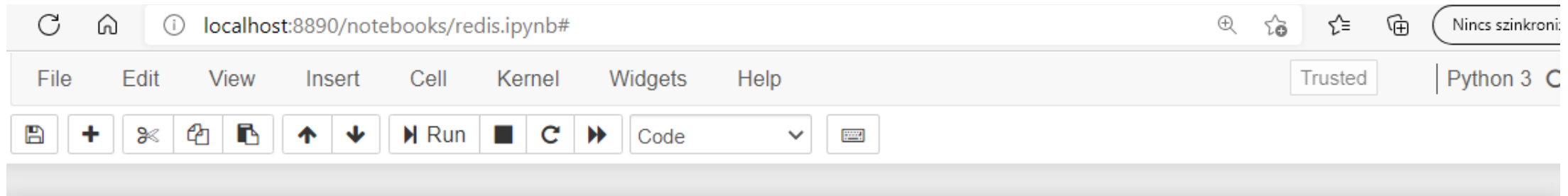
Nincs beépített lehetőség – csak program segítségével (ciklusokkal) oldható meg

# Lekérdezések- Példák

```
C:\Windows\System32\cmd.exe - redis-cli
127.0.0.1:6379>
127.0.0.1:6379> scan 0
1) "7"
2) 1) "mainap"
   2) "napok"
   3) "evszakok"
   4) "mykey"
   5) "a"
   6) "hello"
   7) "tegnap"
   8) "kedvenevszak"
   9) "nevsor"
  10) "nev"
127.0.0.1:6379> zscan nevsor 0 match *a*
1) "0"
2) 1) "laci"
   2) "1"
   3) "zita"
   4) "3"
   5) "nora"
   6) "7"
127.0.0.1:6379>
```

- ☐ Lekérdezzük az aktuális adatbázisban tárolt összes kulcsot
- ☐ Listázzuk a nevsor sorted set azon elemeit és pontszámait, ahol az elem nevében van a betű

# Redis elérés Python-ból



```
In [ ]: !pip install redis
```

```
In [ ]: import redis
```

```
In [ ]: #a host felhős rendszerénél az endpoint értéke, on-premise esetben localhost  
#a port értéke on-premise rendszerénél 6379  
#a jelszó on-premise rendszerénél - ha nem állítottuk be - nem kell  
  
r = redis.Redis(host='', port= , password='')  
r.set('hello', 'world')  
print(r.get('hello'))
```





**Köszönöm  
a figyelmet!**