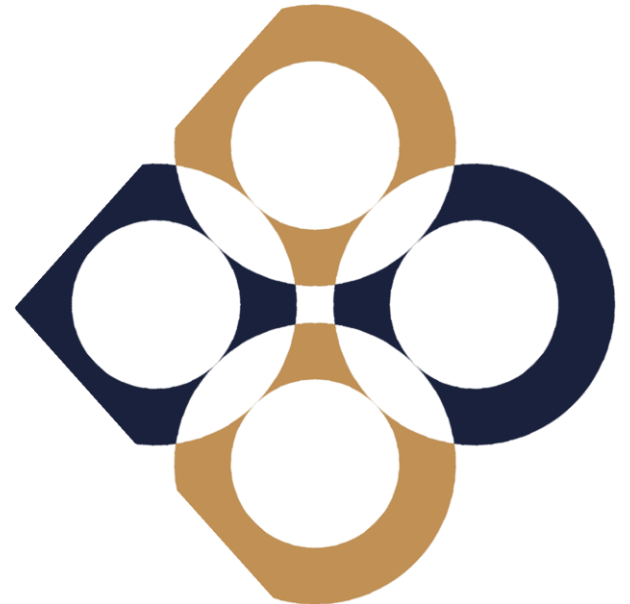


Adatbázisok gyakorlat 11.

Oszlopalapú adatbázisok



Cassandra – CQL nyelv

- ☐ Deklaratív nyelv
- ☐ SQL nyelvhez hasonló szintaktika, de más elvekre épül (az adatmodell nem relációs!)
- ☐ A SQL-nél limitáltabb lehetőségek
 - ☐ Nincsenek tábla összekapcsolások, beágyazott lekérdezések
 - ☐ Nincs tranzakciókezelés, tárolt eljárás, trigger
 - ☐ WHERE feltétel csak elsődleges kulcs oszlopra vagy indexelt oszlopra alkalmazható
 - ☐ Adatot módosítani (frissíteni) csak elsődleges kulcs alapján lehet

Verziótól függően további korlátozások is lehetnek, pl. 3.1 verzió előtt nem volt GROUP BY sem

Cassandra adattípusok

☐ Egyszerű adattípusok

- ☐ Szöveges: ascii, varchar, inet, text, date, time
- ☐ Numerikus: int, varint, bigint, counter, float , decimal, double
- ☐ Logikai: boolean
- ☐ Egyéb: timestamp, uuid, blob

☐ Összetett adattípusok

- ☐ list - elemek rendezett gyűjteménye
 - ☐ map – kulcs-érték párok gyűjteménye
 - ☐ set – elemek halmaza
- ## ☐ Felhasználó által definiált adattípusok

Cassandra – Tábla létrehozása

CREATE TABLE | COLUMNFAMILY táblanév (oszlop definíciók)
[WITH opciók AND opció];

- ❑ Elsődleges kulcs megadása a PRIMARY KEY kulcsszóval történik az oszlopdefiníciónál vagy a tábladefiníció végén
- ❑ Minden táblánál kötelező elsődleges kulcs létrehozása
- ❑ A WITH után tábla opciók adhatók meg pl. tömörítés vagy klaszter oszlopok (CLUSTERED ORDER BY (oszloplista))
- ❑ Az AND után egyéb opciók adhatók meg, pl. oszlop szerinti rendezettség
- ❑ Példa

```
CREATE TABLE Termek (ID INT PRIMARY KEY, Name VARCHAR, Price INT);
```

Létrehozza a Termek táblát három mezővel, ahol az ID az elsődleges kulcs

Cassandra – Kulcsok

- ❑ PRIMARY KEY – Egyedi azonosító
 - ❑ Egyszerű, ha csak egy oszlopból áll
 - ❑ Összetett (composite), ha több oszlopból áll
- ❑ PARTITION KEY – Ez alapján osztódnak el az adatok a node-ok között
- ❑ CLUSTERING KEY – A partíción belül az adatok sorrendjét határozza meg
- ❑ Egyszerű kulcs esetén PRIMARY KEY = PARTITION KEY
- ❑ Összetett kulcs esetén PRIMARY KEY első mezője a PARTITION KEY, a többi a CLUSTERING KEY, pl:
 - ❑ PRIMARY KEY (x) – a PARTITION KEY x
 - ❑ PRIMARY KEY (x, y, z) – a PARTITION KEY x, CLUSTERING KEY y, z
 - ❑ PRIMARY KEY ((x, y), v, w) – a PARTITION KEY (x, y), CLUSTERING KEY (v, w)

Cassandra – Tábla kezelő parancsok

ALTER TABLE | COLUMNFAMILY táblanév
művelet; - Módosítja az adott táblát. A művelet
lehet ADD és DROP

PI: ALTER TABLE
Termek ADD
description TEXT;

DROP TABLE táblanév; - Törli az adott táblát

PI: DROP TABLE
Termek;

TRUNCATE táblanév; - Törli a tábla tartalmát,
a szerkezet megmarad

PI: TRUNCATE Termek;

Cassandra – CRUD utasítások - INSERT

INSERT INTO táblanév (oszlopnevek) VALUES (értékek) [USING opció]

- ❑ Új értékeket szúr be a tábla adott oszlopaiba
- ❑ Az oszlopnevek megadás kötelező, vagy helyette a JSON formátum használható
- ❑ Példák
 - ❑ INSERT INTO Termek (id, name, price) VALUES (1, 'tej', 250);
 - ❑ INSERT INTO Termek JSON '{"id": 2, "name": "kakao", "price": 300}'; *

* Cassandra 2.2 verziótól alkalmazható

Cassandra – CRUD utasítások - UPDATE

UPDATE táblanév SET értékadás(ok) WHERE feltétel(ek);

- ❑ Módosítja a tábla feltételnek megfelelő adatait
- ❑ A WHERE feltételben hivatkozni kell az elsődleges kulcs oszlopra
- ❑ Példák
 - ❑ UPDATE Termek SET price = 200 WHERE id = 1;
 - ❑ UPDATE Termek SET price = 200 WHERE id IN (1, 2);
 - ❑ UPDATE Termek SET price = 200 WHERE name = 'tej';
-- Ez hibás, mert nincs kulcs hivatkozás

DELETE FROM táblanév WHERE feltétel(ek);

- ❑ Törli a tábla feltételnek megfelelő adatait
- ❑ A WHERE feltételben itt is kell kulcsra hivatkozni
- ❑ Példák
 - ❑ DELETE FROM Termek WHERE id = 2;
 - ❑ DELETE FROM Termek WHERE price = 200;
-- Ez hibás, mert nincs kulcs hivatkozás

Cassandra – Map, List, Set példák

```
CREATE TABLE Raktar(id INT, nev VARCHAR, aruk SET<TEXT>, készlet  
MAP<TEXT, INT>, dolgozok LIST<TEXT>, PRIMARY KEY(id));
```

- ❑ INSERT INTO raktar (id, aruk, dolgozok, készlet, nev)
VALUES (1, {'kifli', 'tej', 'kenyer'}, ['Kiss Bela', 'Nagy Ivett'], {'kifli':
200, 'tej': 50, 'kenyer': 130}, 'Raktar01');
- ❑ UPDATE Raktar
SET aruk = aruk + {'zsemle'},
dolgozok = dolgozok + ['Kozepes Jeno'],
készlet['kenyer'] = 500
WHERE id = 1;

Cassandra - Lekérdezések












- ☐ A DISTINCT kulcsszó használható
- ☐ Az oszlopokat elnevezhetjük az AS után
- ☐ A WHERE utáni oszlopok lehetnek elsődleges kulcs oszlopai, indexelt oszlopok vagy clustered oszlopok
- ☐ Az ORDER BY után csak clustered oszlopok adhatók meg, a WHERE után ilyenkor szükséges a kulcsra hivatkozás
- ☐ A GROUP BY oszlopai csak az elsődleges kulcs oszlopai lehetnek megfelelő sorrendben*
- ☐ A LIMIT után megadható, hogy az eredménysorok közül hány jelenjen meg

```
SELECT ...  
FROM ...  
WHERE ...  
GROUP BY ...  
ORDER BY ...  
LIMIT ...
```

*A partition key mindegyik oszlopának szerepelnie kell

Cassandra elérés Python-ból

FileEditViewInsertCellKernelWidgetsHelpTrustedPython 3 C



Code

```
In [ ]: !pip install cassandra-driver

In [2]: from cassandra.cluster import Cluster

In [3]: cluster = Cluster(['127.0.0.1'])
        session = cluster.connect()

In [23]: session.execute('use system;')
         r= session.execute('select * from schema_keyspaces;')
         print(r.current_rows)

[Row(keyspace_name='kproba', durable_writes=True, strategy_class='org.apache.cassandra.locator.Simple
Strategy', strategy_options='{"replication_factor":"3"}'), Row(keyspace_name='system', durable_writes
=True, strategy_class='org.apache.cassandra.locator.LocalStrategy', strategy_options='{}'), Row(keysp
ace_name='system_traces', durable_writes=True, strategy_class='org.apache.cassandra.locator.SimpleStr
ategy', strategy_options='{"replication_factor":"2"}')]

In [24]: cluster.shutdown()
```

Aktiválja a Windowst



**Köszönöm
a figyelmet!**