***NoSQL Beadandó Feladat***

**Relációs adatbázis leírása:**

Táblák:

1. Users (Felhasználók)

* user\_id (PK): A felhasználó azonosítója
* username: A felhasználónév
* email: A felhasználó e-mail címe

1. Posts (Bejegyzések)

* post\_id (PK): A bejegyzés azonosítója
* user\_id (FK): A bejegyzést író felhasználó azonosítója
* content: A bejegyzés tartalma
* timestamp: A bejegyzés időbélyege

1. Comments (Hozzászólások)

* comment\_id (PK): A hozzászólás azonosítója
* post\_id (FK): Bejegyzés idegenkulcsa
* user\_id (FK): A hozzászólást író felhasználó azonosítója
* content: A hozzászólás tartalma
* timestamp: A hozzászólás időbélyege

Kapcsolatok:

A Posts tábla user\_id mezője kapcsolatban van a Users tábla user\_id mezőjével.

A Comments tábla post\_id mezője kapcsolatban van a Posts tábla post\_id mezőjével.

A Comments tábla user\_id mezője kapcsolatban van a Users tábla user\_id mezőjével.

**NoSQL adatbázis leírása (DataStax Astra használatával):**

CREATE TABLE users ( // users tábla létrehozása

user\_id UUID PRIMARY KEY,

username TEXT,

email TEXT

);

CREATE TABLE posts ( // posts tábla létrehozása

post\_id UUID PRIMARY KEY,

user\_id UUID,

content TEXT,

timestamp TIMESTAMP,

FOREIGN KEY (user\_id) REFERENCES users (user\_id)

);

CREATE TABLE comments ( // comments tábla létrehozása

comment\_id UUID PRIMARY KEY,

post\_id UUID,

user\_id UUID,

content TEXT,

timestamp TIMESTAMP,

FOREIGN KEY (post\_id) REFERENCES posts (post\_id),

FOREIGN KEY (user\_id) REFERENCES users (user\_id)

);

// Felhasználók beszúrása

INSERT INTO users (user\_id, username, email) VALUES (uuid(), 'KissJoska123', 'Kiss.J123@gmail.com');

INSERT INTO users (user\_id, username, email) VALUES (uuid(), 'Varga\_Szilvia', 'vszilvia@citromail.hu');

INSERT INTO users (user\_id, username, email) VALUES (uuid(), 'Marton1995', 'marton1995@freemail.hu');

// Bejegyzések beszúrása

INSERT INTO posts (post\_id, user\_id, content, timestamp) VALUES (uuid(), user\_id1, '1. Bejegyzés', toTimestamp(now()));

INSERT INTO posts (post\_id, user\_id, content, timestamp) VALUES (uuid(), user\_id2, '2. Bejegyzés', toTimestamp(now()));

INSERT INTO posts (post\_id, user\_id, content, timestamp) VALUES (uuid(), user\_id3, '3. Bejegyzés', toTimestamp(now()));

// Hozzászólások beszúrása

INSERT INTO comments (comment\_id, post\_id, user\_id, content, timestamp) VALUES (uuid(), post\_id1, user\_id1, ' Szuper!', toTimestamp(now()));

INSERT INTO comments (comment\_id, post\_id, user\_id, content, timestamp) VALUES (uuid(), post\_id1, user\_id2, 'Köszönöm szépen!', toTimestamp(now()));

INSERT INTO comments (comment\_id, post\_id, user\_id, content, timestamp) VALUES (uuid(), post\_id2, user\_id3, 'Lenne egy kérdésem', toTimestamp(now()));

Lekérdezések:

//1. Keresés a legújabb bejegyzés és hozzá tartozó hozzászólások között egy adott felhasználó által:

SELECT \*

FROM posts

WHERE user\_id = 'user\_id'

ORDER BY timestamp

DESC LIMIT 1;

//2. Keresés a legtöbb bejegyzéssel rendelkező felhasználóra:

SELECT user\_id, COUNT(\*) AS post\_count

FROM posts

GROUP BY user\_id

ORDER BY post\_count

DESC LIMIT 10;

//3. Szűrés bejegyzéseken van a legtöbb hozzászólás:

SELECT post\_id, COUNT(\*) AS comment\_count

FROM comments

GROUP BY post\_id

ORDER BY comment\_count

DESC LIMIT 5;

//4. Az összes bejegyzéshez tartozó hozzászólások számának összesítése:

SELECT post\_id, COUNT(\*) AS comment\_count

FROM comments

GROUP BY post\_id;

//5. Keresés azon felhasználók között, akik a legtöbb különböző témában osztottak meg bejegyzéseket:

SELECT user\_id, COUNT(DISTINCT post\_id) AS topic\_count

FROM posts GROUP BY user\_id

ORDER BY topic\_count

DESC LIMIT 5;