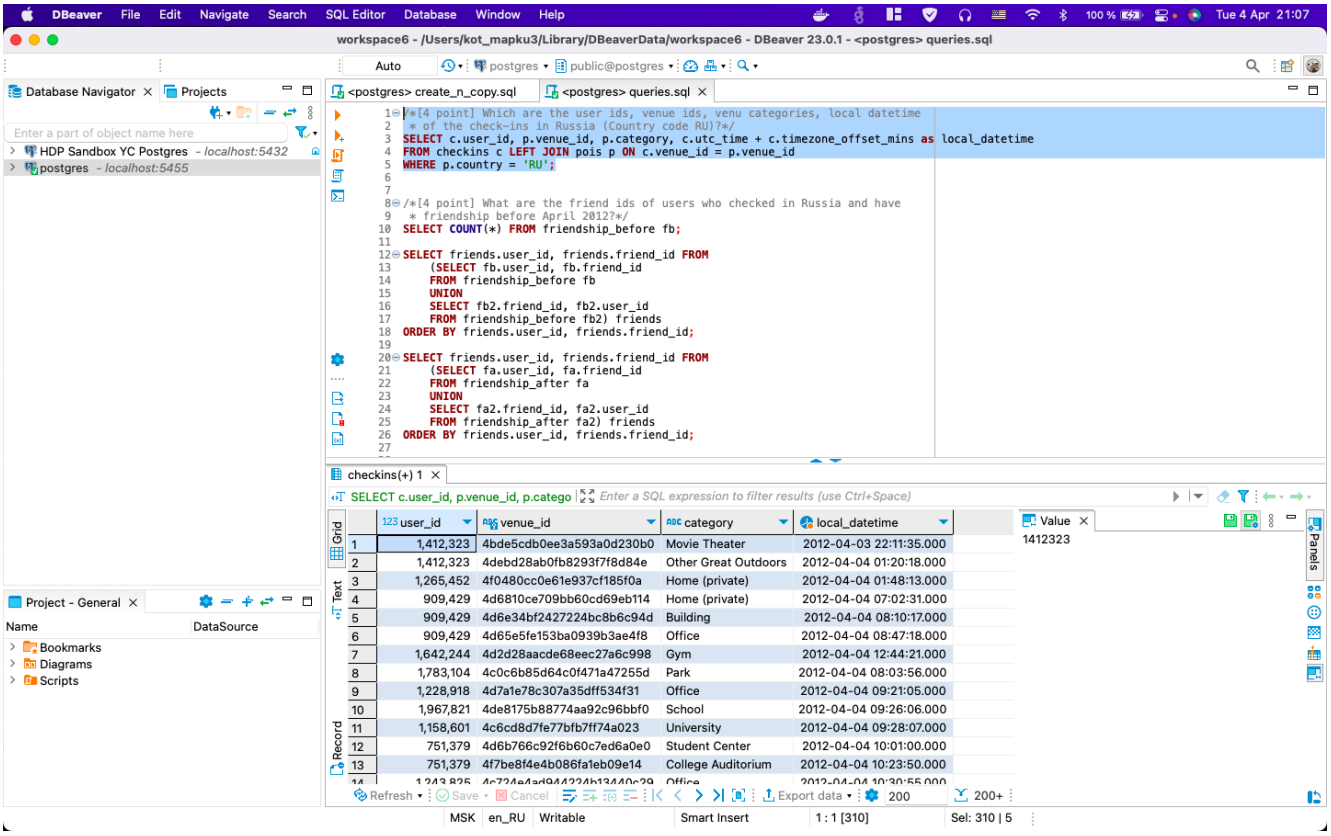
Track: B20-AI-01	**Week 2**
Name: Artem Chernitsa	**Introduction to Big Data**
Email: a.chernitsa@innopolis.university	**28.03.2023**

Report

<PostgreSQL Part>

Task 1

Your query as a text
SELECT c.user_id, p.venue_id, p.category, c.utc_time + c.timezone_offset_mins **as** local_datetime **FROM** checkins c **LEFT JOIN** pois p **ON** c.venue_id = p.venue_id **WHERE** p.country = 'RU';

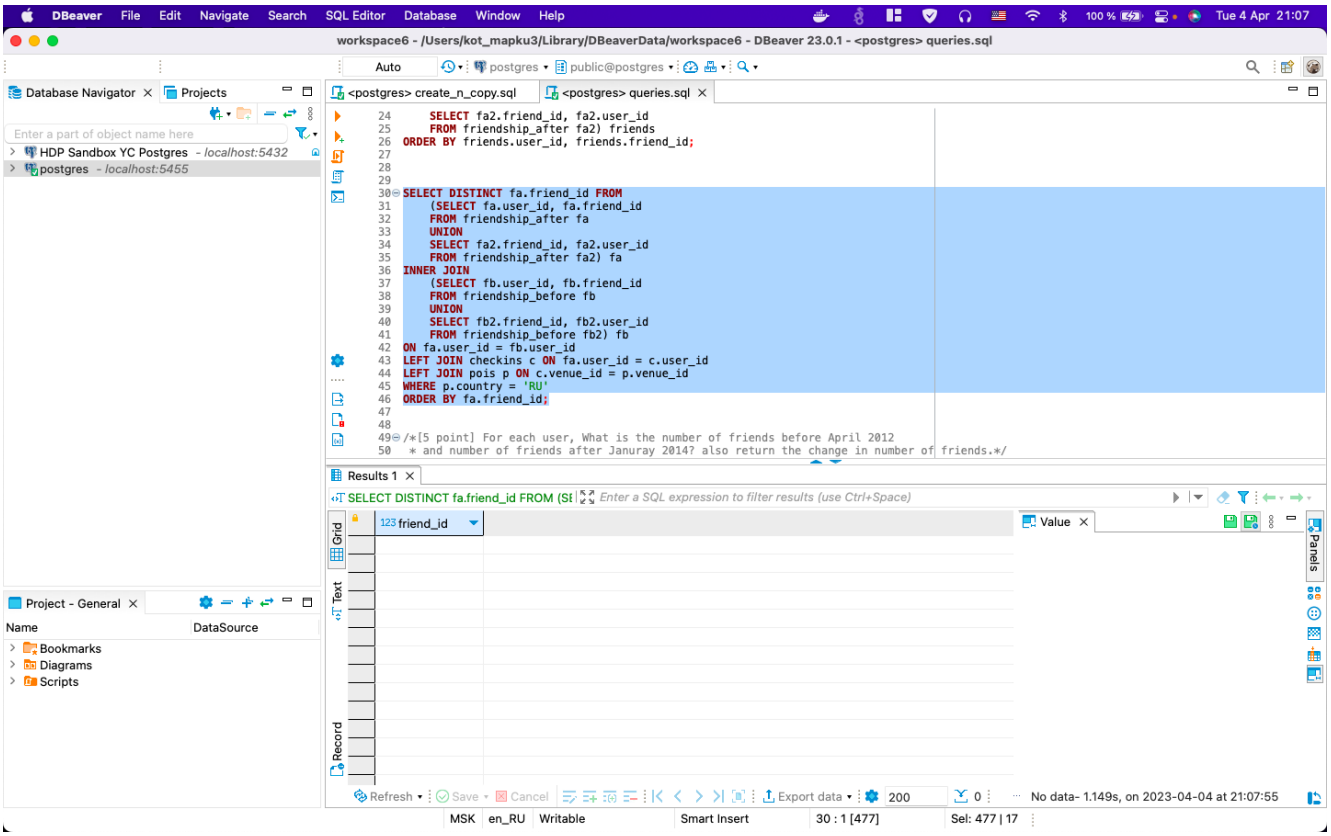


Your comments (optional)
I did it on the HDP cluster, but I didn't save any screenshots. Since I dropped my server cause of mongoDB, I simply up docker container with postgres and make absolutely the same just to save screenshots.

Task 2

Your query as a text

```
SELECT DISTINCT fa.friend_id FROM
        (SELECT fa.user_id, fa.friend_id
        FROM friendship_after fa
        UNION
        SELECT fa2.friend_id, fa2.user_id
        FROM friendship_after fa2) fa
INNER JOIN
        (SELECT fb.user_id, fb.friend_id
        FROM friendship_before fb
        UNION
        SELECT fb2.friend_id, fb2.user_id
        FROM friendship_before fb2) fb
ON fa.user_id = fb.user_id
LEFT JOIN checkins c ON fa.user_id = c.user_id
LEFT JOIN pois p ON c.venue_id = p.venue_id
WHERE p.country = 'RU'
ORDER BY fa.friend_id;
```



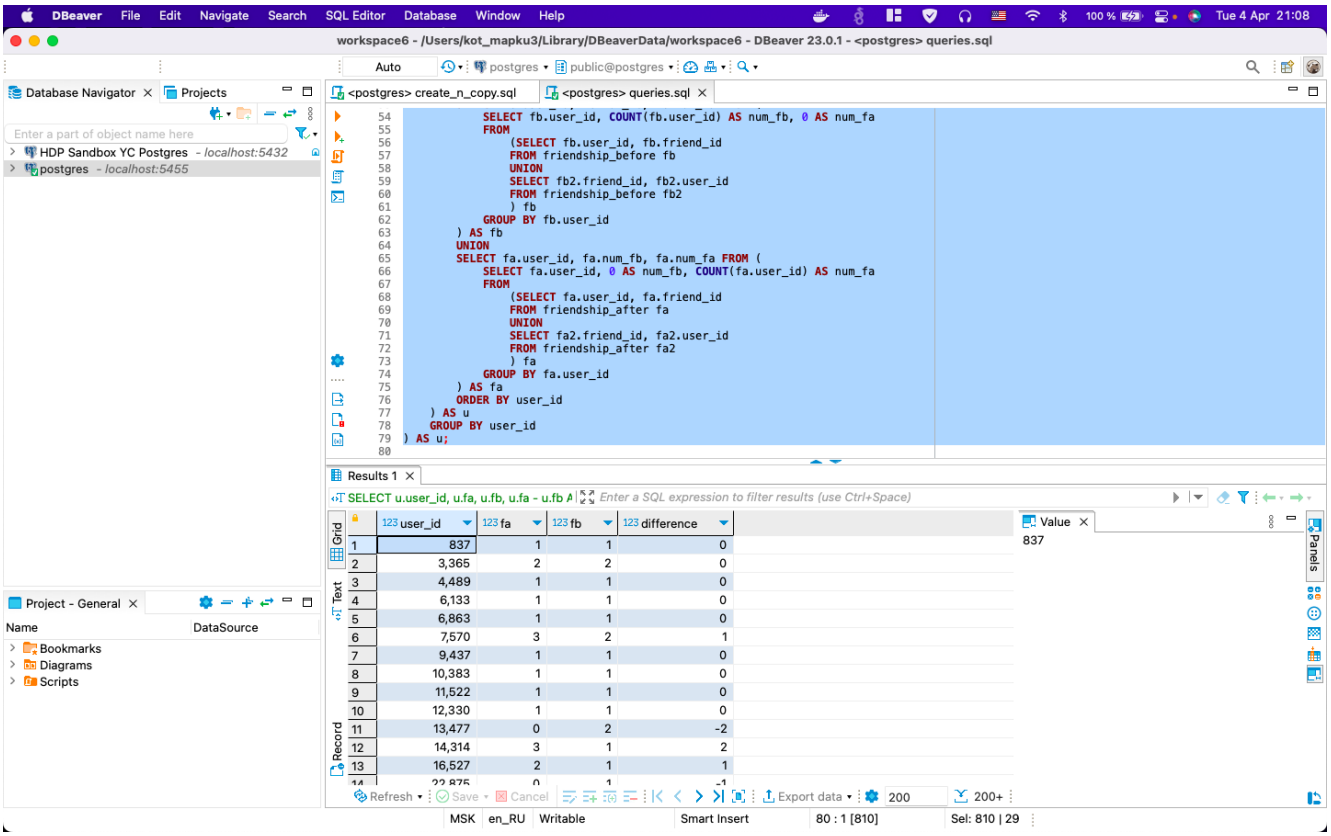
Your comments (optional)
For RU code is empty, but e.g. for the US is gives a lot of rows as a result.

Task 3

Your query as a text

```
SELECT u.user_id, u.fa, u.fb, u.fa - u.fb AS difference FROM (
    SELECT u.user_id, SUM(u.num_fb) AS fb, SUM(u.num_fa) AS fa FROM (
        SELECT fb.user_id, fb.num_fb, fb.num_fa FROM (
            SELECT fb.user_id, COUNT(fb.user_id) AS num_fb, 0 AS num_fa
            FROM
                (SELECT fb.user_id, fb.friend_id
                FROM friendship_before fb
                UNION
                SELECT fb2.friend_id, fb2.user_id
```

```
                        FROM friendship_before fb2
                        ) fb
                GROUP BY fb.user_id
            ) AS fb
            UNION
            SELECT fa.user_id, fa.num_fb, fa.num_fa FROM (
                SELECT fa.user_id, 0 AS num_fb, COUNT(fa.user_id) AS num_fa
                FROM
                    (SELECT fa.user_id, fa.friend_id
                    FROM friendship_after fa
                    UNION
                    SELECT fa2.friend_id, fa2.user_id
                    FROM friendship_after fa2
                    ) fa
                GROUP BY fa.user_id
            ) AS fa
            ORDER BY user_id
        ) AS u
        GROUP BY user_id
) AS u;
```



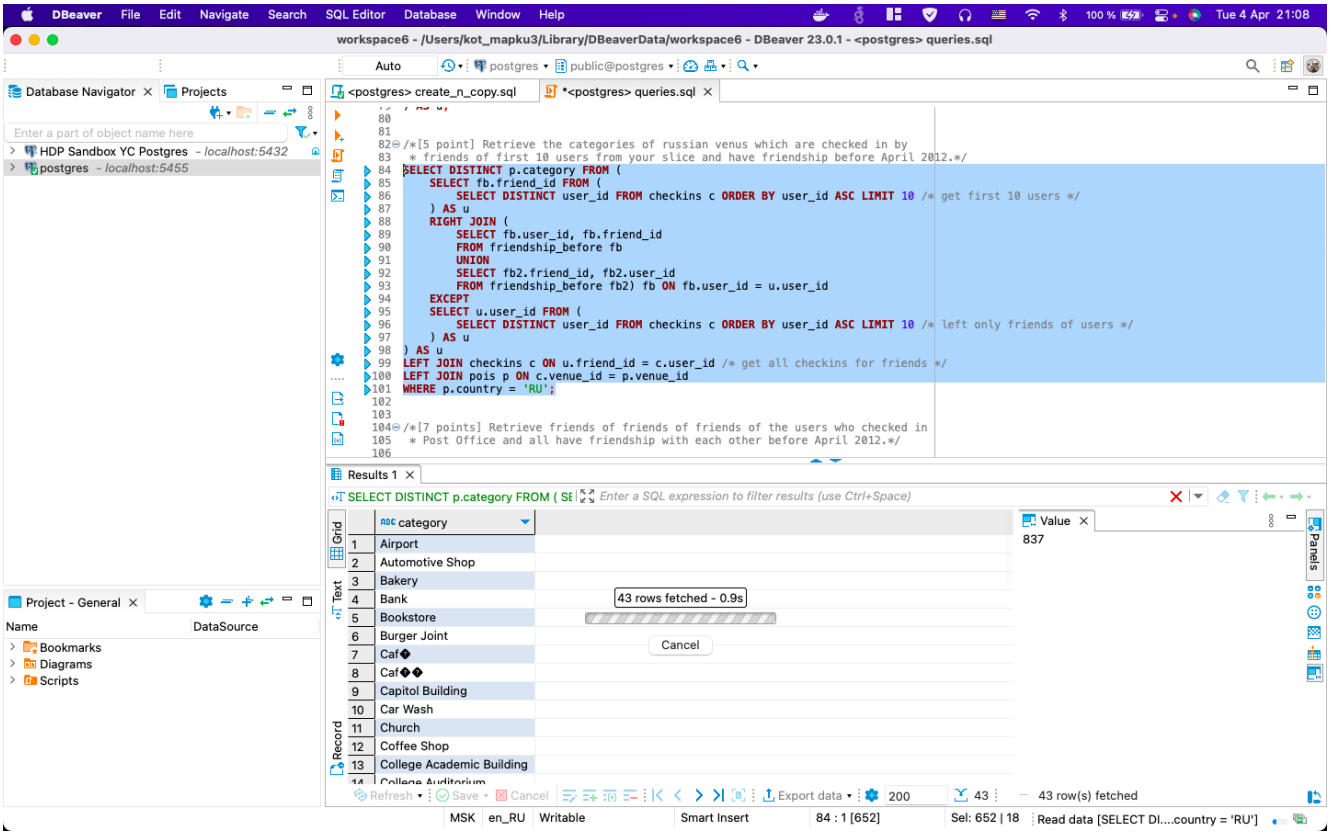
Your comments (optional)

Task 4

Your query as a text

```
SELECT DISTINCT p.category FROM (
    SELECT fb.friend_id FROM (
        SELECT DISTINCT user_id FROM checkins c ORDER BY user_id ASC
LIMIT 10 /* get first 10 users */
    ) AS u
    RIGHT JOIN (
        SELECT fb.user_id, fb.friend_id
        FROM friendship_before fb
        UNION
```

```
            SELECT fb2.friend_id, fb2.user_id

            FROM friendship_before fb2) fb ON fb.user_id = u.user_id

        EXCEPT

        SELECT u.user_id FROM (

            SELECT DISTINCT user_id FROM checkins c ORDER BY user_id ASC
LIMIT 10 /* left only friends of users */

        ) AS u

) AS u

LEFT JOIN checkins c ON u.friend_id = c.user_id /* get all checkins for friends */

LEFT JOIN pois p ON c.venue_id = p.venue_id

WHERE p.country = 'RU';
```



Your comments (optional)

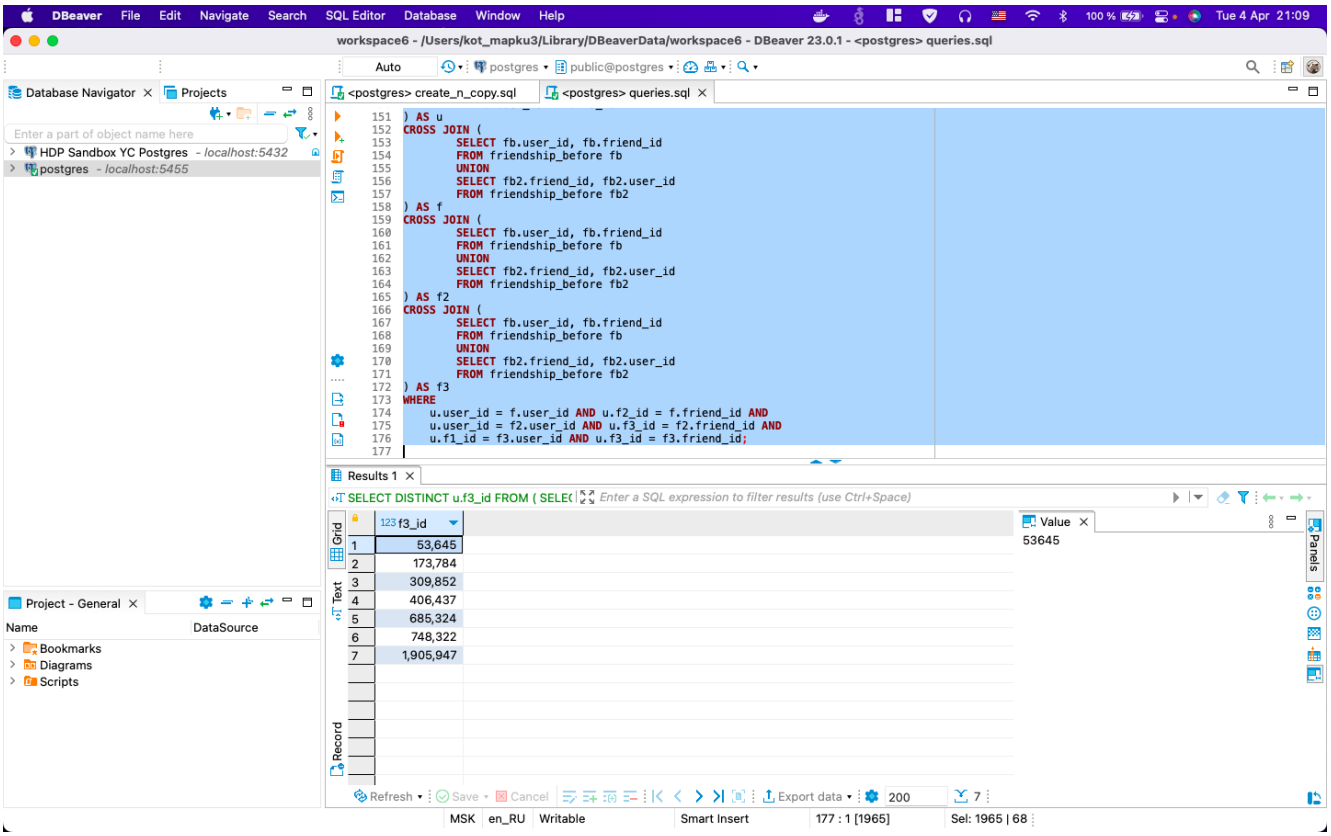
Task 5

Your query as a text

```
SELECT /*u.user_id, u.f1_id, u.f2_id,*/ DISTINCT u.f3_id FROM (
        SELECT u.user_id, u.f1_id, u.f2_id, u.f3_id FROM (
                SELECT u.user_id, f1.friend_id AS f1_id, f2.friend_id AS f2_id, f3.friend_id
AS f3_id FROM (
                        SELECT DISTINCT users.user_id FROM
                                (SELECT fb.user_id, fb.friend_id
                                FROM friendship_before fb
                                UNION
                                SELECT fb2.friend_id, fb2.user_id
                                FROM friendship_before fb2) users
                        LEFT JOIN checkins c ON c.user_id = users.user_id
                        LEFT JOIN pois p ON c.venue_id = p.venue_id
                        WHERE p.category = 'Post Office'
                ) AS u
                LEFT JOIN (
                        SELECT fb.user_id, fb.friend_id
                        FROM friendship_before fb
                        UNION
                        SELECT fb2.friend_id, fb2.user_id
                        FROM friendship_before fb2
                ) AS f1 ON u.user_id = f1.user_id
                LEFT JOIN (
                        SELECT fb.user_id, fb.friend_id
                        FROM friendship_before fb
                        UNION
```

```
                    SELECT fb2.friend_id, fb2.user_id
                    FROM friendship_before fb2
                ) AS f2 ON f1.friend_id = f2.user_id
                LEFT JOIN (
                    SELECT fb.user_id, fb.friend_id
                    FROM friendship_before fb
                    UNION
                    SELECT fb2.friend_id, fb2.user_id
                    FROM friendship_before fb2
                ) AS f3 ON f2.friend_id = f3.user_id
                ORDER BY u.user_id ASC
        ) AS u
        WHERE u.user_id != u.f1_id AND
                    u.f1_id != u.f2_id AND
                    u.f2_id != u.f3_id AND
                    u.f3_id != u.f1_id AND
                    u.user_id != u.f2_id AND
                    u.user_id != u.f3_id
) AS u
CROSS JOIN (
                SELECT fb.user_id, fb.friend_id
                FROM friendship_before fb
                UNION
                SELECT fb2.friend_id, fb2.user_id
                FROM friendship_before fb2
) AS f
CROSS JOIN (
```

```
            SELECT fb.user_id, fb.friend_id
            FROM friendship_before fb
            UNION
            SELECT fb2.friend_id, fb2.user_id
            FROM friendship_before fb2
) AS f2
CROSS JOIN (
            SELECT fb.user_id, fb.friend_id
            FROM friendship_before fb
            UNION
            SELECT fb2.friend_id, fb2.user_id
            FROM friendship_before fb2
) AS f3
WHERE
        u.user_id = f.user_id AND u.f2_id = f.friend_id AND
        u.user_id = f2.user_id AND u.f3_id = f2.friend_id AND
        u.f1_id = f3.user_id AND u.f3_id = f3.friend_id;
```

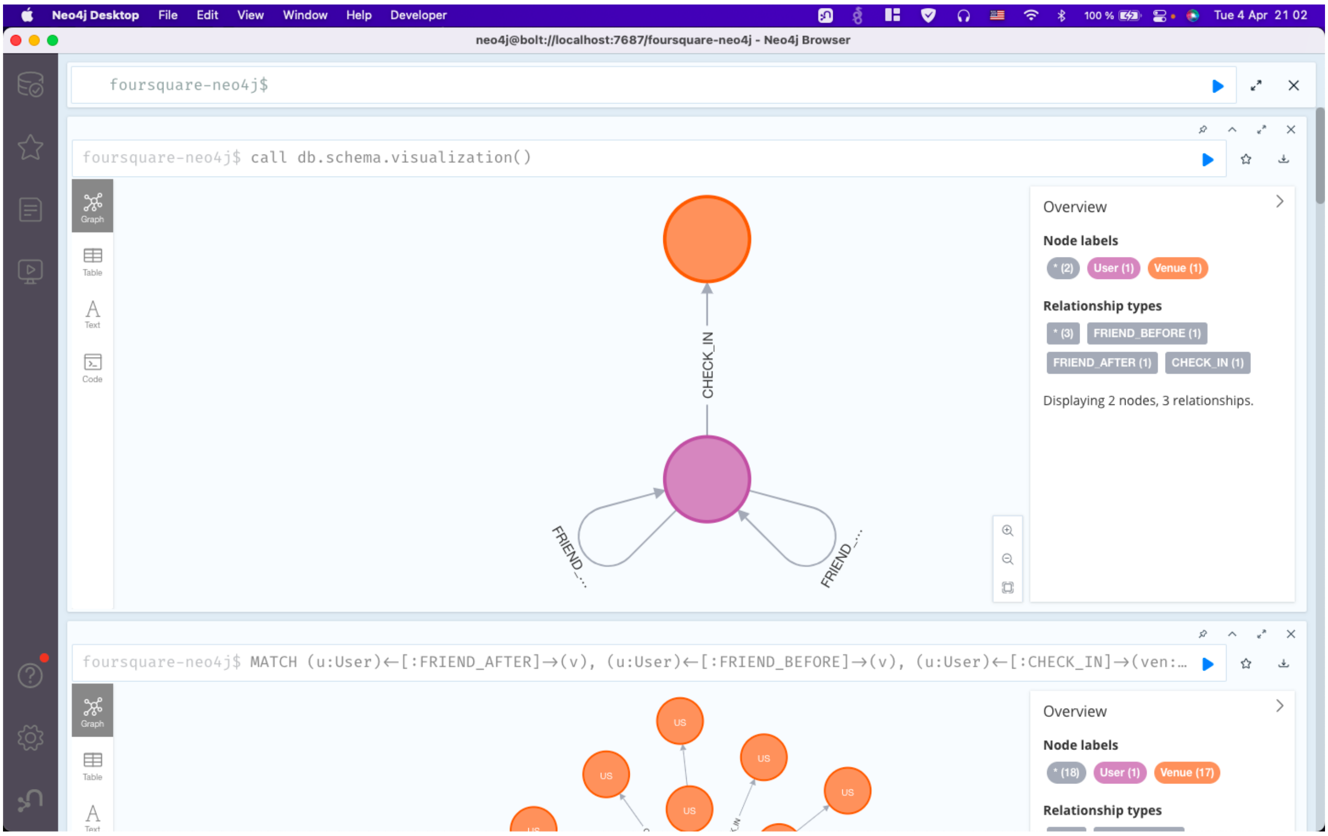


Your comments (optional)

<Neo4j Part>

Task 1

Your schema output as a text
╒═══╤═════════════════════════════════╕ │nodes │relationships │ ╞═══╪═════════════════════════════════╡ │[(:User {name: "User",indexes: ["user_id"],constraints: []}), (:Venue │[[:FRIEND_BEFORE], [:FRIEND_AFTER], [:CHECK_IN]]│ │{name: "Venue",indexes: ["venue_id"],constraints: []})] │ │ └───┴─────────────────────────────────┘



Your comments (optional)

Task 2

Your query as a text

< put here the screenshot of the output>

Your comments (optional)

Task 3

Your query as a text

< put here the screenshot of the output>

Your comments (optional)

Task 4

Your query as a text

< put here the screenshot of the output>

Your comments (optional)

Task 5

Your query as a text

< put here the screenshot of the output>

Your comments (optional)

Task 6

Your query as a text

< put here the screenshot of the output>

Your comments (optional)