QUERIES:

Q1: Show quarantined people that went to a place while being infected.

```
MATCH
 (a:Place { name:"Casual meeting on the street at datetime('2021-12-23T19:3')" })-
[c:HOSTED]->(p:Person)-[:GOT_AN]->(i:Infection)
 WHERE c.entry_moment >= i.date_of_infection
RETURN p,a
```

Q2: Display the top K places with most infected people, and the number of infected for each of these places.

```
MATCH (p:Place)-[:HOSTED]->(:Person)-[r:GOT_AN]->(:Infection)
WITH p, COUNT(r) AS cnt ORDER BY cnt desc
RETURN collect(p.name) as names
LIMIT 10
```

Q3: Show a table containing each vaccination type and the number of people that have done it.

```
MATCH (gp: GreenPass)
WITH  gp.type as Vaccination_type, COUNT(gp) as number
RETURN COLLECT(Vaccination_type) as type_of_vaccine, number  ORDER BY number
DESC
```

Q4: Show the daily infected/healthy ratio.

```
MATCH (pInfected:Person)-[:GOT_AN]->(:Infection)
WITH COUNT(pInfected) as infected
MATCH (pHealthy:Person)
WHERE EXISTS ((pHealthy)-[:GOT_AN]->(:Infection) == FALSE
WITH infected, COUNT(pHealthy) AS healthy
RETURN (infected / toFloat(healthy)) AS dailyRatio
```

Q5: Show the most visited day of the most visited place.

```
match (:Person)-[r:WENT_TO]->(p:Place)
with count(r) as num, p
order by num desc limit 1

match (a:Person)-[r1:WENT_TO]->(p)<-[r2:WENT_TO]-(b:Person)
where r1.exit_moment.epochSeconds > r2.entry_moment.epochSeconds AND
r1.entry_moment < r2.exit_moment
with count(a)+1 as number, p.name as place, r1, date(r1.entry_moment) as date, a
return date, number, place, collect(a) order by number desc limit 1
```

COMMANDS:

C1: Set positive a list of already registered people.

```
MATCH (p1: Person {name:"Nikita",taxCode:"544",Surname:"Bro1”}), (p2: Person
{name:"Nikita",taxCode:"545",Surname:"Bro2”}), …
CREATE
(p1)-[:GOT_AN]->
      (:Infection{date_of_infection:"2021-12-12T13:16:54.414000000Z"}),
 (p2)-[:GOT_AN]->
      (:Infection{date_of_infection:"2021-12-12T13:16:54.415000000Z”}),
…
```

C2: Attach a Green Pass with activation date D1, expiration date D2 and type T to a Person with taxCode TC and, eventually, delete its Infection.

```
MATCH (a:Person)
WHERE a.taxCode= "1"
CREATE (gp:GreenPass), (a)-[:HAS_A]->(gp)-[:BELONGS_TO]->(a)
WITH 1 as dummy
MATCH (a)-[r:GOT_AN]->(i:Infection)
WHERE a.taxCode= "1"
DETACH DELETE i
```

C3: Data cleaning command: delete expired Green Passes and visits (after 14 days).

```
MATCH (a)-[r:HAS_A|BELONGS_TO|WENT_TO|HOSTED]->(p)
WHERE (datetime().epochSeconds-(r.exit_moment).epochSeconds>=0
      AND datetime().epochSeconds-(r.exit_moment).epochSeconds >= 86400*14)
   OR
      (datetime().epochSeconds > p.date2.epochSeconds OR datetime().epochSeconds >
a.date2.epochSeconds)
DELETE r
WITH 1 AS dummy
MATCH (gp:GreenPass) WHERE NOT EXISTS( (gp)<-[:HAS_A]-(:Person) )
DELETE gp
```