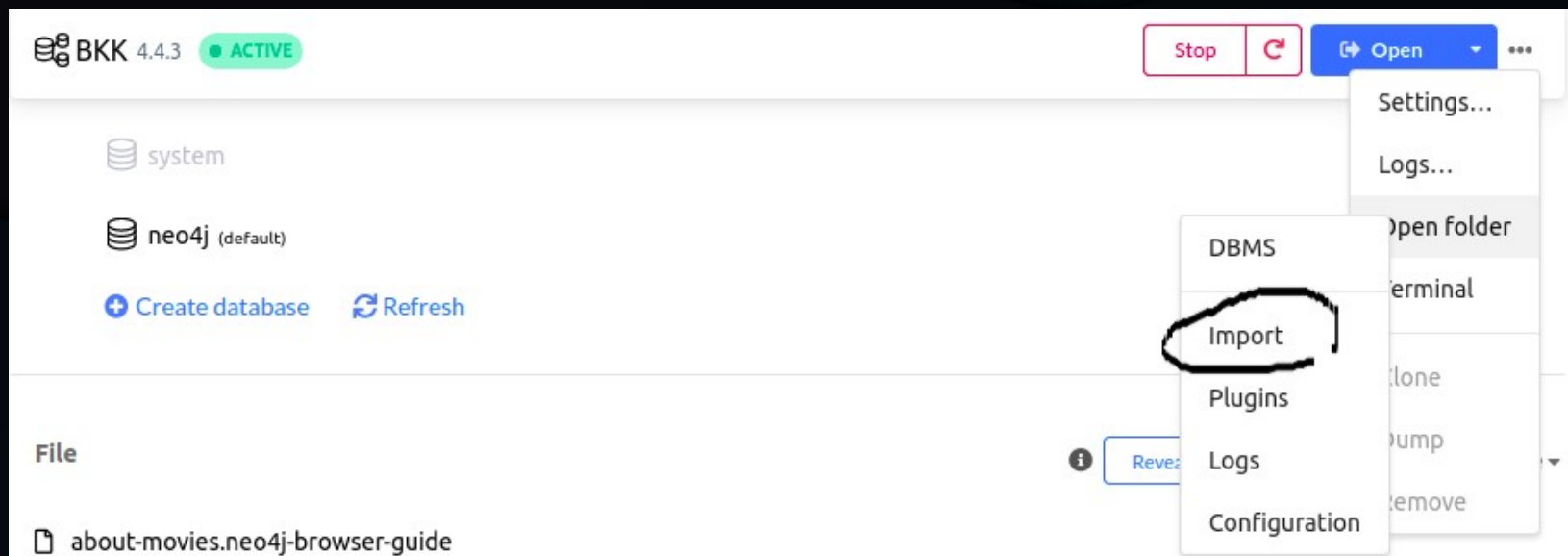


Budapesti tömegközlekedés ábrázolása Neo4j segítségével

Első lépések

- Az adatok bemásolása az adott adatbázis mappájába



Első lépések

- Adatok megtekintése, hibaellenőrzés

```
trips = pd.read_csv("trips.txt")  
trips[["route_id", "trip_id", "shape_id"]].isna().sum(axis=0)
```

```
route_id    0  
trip_id     0  
shape_id    0  
dtype: int64
```

Első lépések

- Az idő probléma

```
stop_times.departure_time.max()
```

```
'28:39:00'
```

```
def idoalakitas(time):  
    t = time.split(":")  
    t = [ *map(int, t)]  
  
    t[0] = t[0] % 24  
    return "{:02d}:{:02d}:{:02d}".format(*t)
```

```
stop_times.departure_time = stop_times.departure_time.map(idoalakitas)  
stop_times.arrival_time = stop_times.arrival_time.map(idoalakitas)
```

```
stop_times.departure_time.max()
```

```
'23:59:35'
```

Első lépések

- Kapcsolatok a táblák között

	shape_dist_traveled				
	route_id		stop_id	trip_id	route_id
pathway_id	route_short_name		stop_name	stop_id	trip_id
pathway_mode	route_long_name	shape_id	stop_lat	arrival_time	service_id
is_bidirectional	route_type	shape_pt_sequence	stop_lon	departure_time	trip_headsign
from_stop_id	route_desc	shape_pt_lat	stop_code	stop_sequence	direction_id
to_stop_id	route_color	shape_pt_lon	location_type	stop_headsign	block_id
traversal_time	route_text_color		location_sub_type	pickup_type	shape_id
	route_sort_order		parent_station	drop_off_type	wheelchair_accessible
	route_icon_display_text		wheelchair_boarding	shape_dist_traveled	bikes_allowed
			stop_direction		boarding_door

Első lépések

- Elkezdődhet a beolvasás

```
CREATE CONSTRAINT trip_id ON (t: Trip) ASSERT t.trip_id IS UNIQUE
LOAD CSV WITH HEADERS FROM 'file:///trips.csv' AS row
CREATE (t: Trip {trip_id: row.trip_id, route_id: row.route_id, shape_id: row.shape_id})
```

Első lépések

- Hiba a beolvasásnál

ERROR Neo.ClientError.Schema.ConstraintValidationFailed

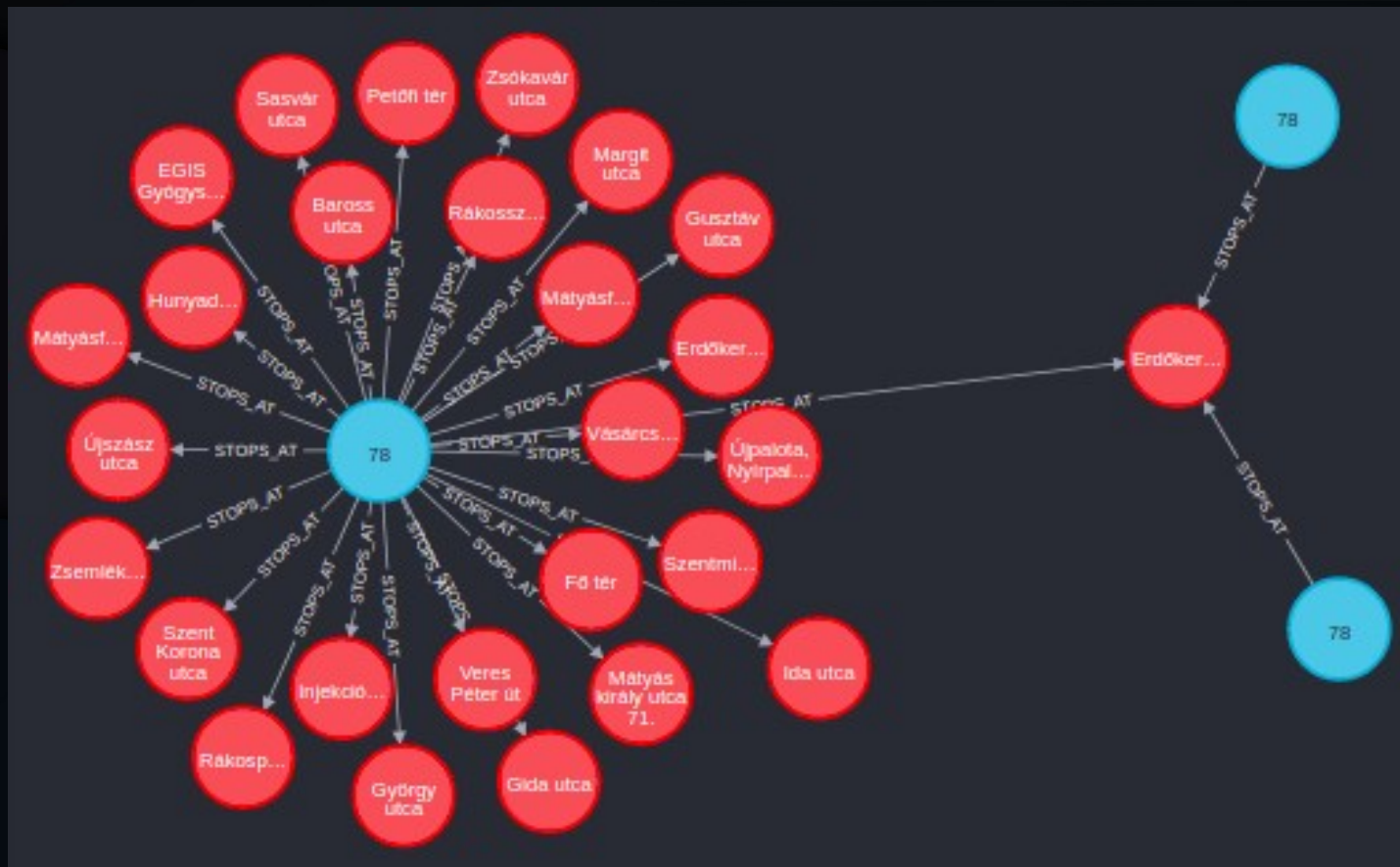
```
Node(821) already exists with label `Stop` and property `stop_id` = '9001'
```

Kapcsolatok

- Kapcsolatok megteremtése, Trip-Stop
- :auto.....- minden 100 000-ik után ment, hogy el ne vesszenek az adatok.
- MERGE – létrehozunk vagy frissítünk.

```
:auto USING PERIODIC COMMIT 100000
LOAD CSV WITH HEADERS FROM 'file:///stop_times.csv' AS row
MATCH (t: Trip {trip_id: row.trip_id})
MATCH (s: Stop {stop_id: row.stop_id})
MERGE (t)-[:STOPS_AT { arrival_time: time(row.arrival_time), departure_time:
time(row.departure_time), stop_sequence: toInteger(row.stop_sequence) }]-
(s)
```


Kapcsolatok

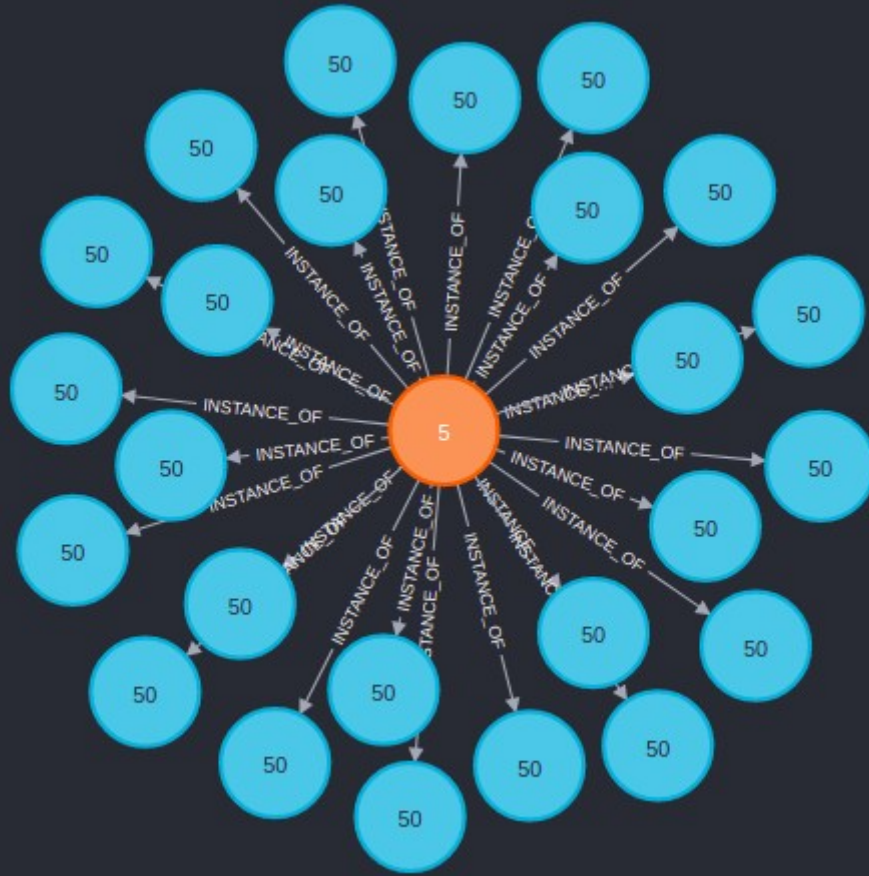


Kapcsolatok

- Kapcsolat megteremtése, Routes-Trip
- Az r tábla és t tábla route_id egyezésnél

```
MATCH (r: Route)
MATCH (t: Trip)
WHERE r.route_id = t.route_id
MERGE (r)-[:INSTANCE_OF]-(t)
```

Kapcsolatok



Kapcsolatok

- A kapcsolatok bárhogyan kialakíthatók
- A gráfokat mi alkotjuk

Lekérdezés

```
MATCH (s1: Stop)
WHERE s1.stop_name CONTAINS "Városház tér"
MATCH (s1)-[:STOPS_AT]-(t: Trip)-[:STOPS_AT]-(s2: Stop)
RETURN DISTINCT s2.stop_name
LIMIT 10
```

	s2.stop_name
1	"Tenkes utca"
3	"Bányalég utca"
4	"Jókai Mór utca"
5	"Angeli utca / Nagytétényi út"
6	"Nagytétény, Erdélyi utca"

Köszönöm a figyelmet!