

## **A Táblák betöltése**

```
CREATE CONSTRAINT route_id ON (r: Route) ASSERT r.route_id IS UNIQUE
LOAD CSV WITH HEADERS FROM 'file:///routes.csv' AS row
CREATE (r: Route {route_id: row.route_id, route_short_name: row.route_short_name,
route_type: row.route_type, route_desc: row.route_desc})
```

```
CREATE CONSTRAINT stop_id ON (s: Stop) ASSERT s.stop_id IS UNIQUE
LOAD CSV WITH HEADERS FROM 'file:///stops.csv' AS row
CREATE (s: Stop {stop_id: row.stop_id, stop_name: row.stop_name, stop_lat:
row.stop_lat, stop_lon: row.stop_lon })
```

```
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, service_id: row.service_id, route_id: row.route_id,
shape_id: row.shape_id})
```

```
CREATE CONSTRAINT pathway_id ON (p: Pathway) ASSERT p.pathway_id IS UNIQUE
CREATE (p: Pathway {pathway_id: row.pathway_id, pathway_mode: row.pathway_mode,
is_bidirectional: row.is_bidirectional, from_stop_id: row.from_stop_id, to_stop_id:
row.to_stop_id, traversal_time: row.traversal_time })
```

## **Lekérdezések:**

### **Átszállás nélküli utak a megadott megállóhoz:**

```
MATCH (s1: Stop)
WHERE s1.stop_name CONTAINS "Márta utca"
MATCH (s1)-[:STOPS_AT]-(t: Trip)-[:STOPS_AT]-(s2: Stop)
RETURN DISTINCT s2.stop_name
```

Így a neveket kapjuk vissza, ha .stop\_name nélkül írom, akkor minden adatot megkapok.

### **Bizonyos távolságon belüli megállóhelyek keresése:**

```
MATCH (s1: Stop), (s2: Stop)
WITH point.distance(point( {x: s1.stop_lon, y: s1.stop_lat, crs: "WGS-84"}),
point( {x: s2.stop_lon, y: s2.stop_lat, crs: "WGS-84"})) as dist, s1 as s1, s2 as s2
WHERE dist < 600
MERGE (s1)-[:CLOSE_TO {d: dist}]- (s2)
```

## Két megálló közötti lehetséges utvonalak egy időintervallumon belül

```
MATCH (sd: SDate {service_date: date("2022-01-21")})-[*1]-(services)
MATCH (t1: Trip{route_id : "1530"})-[:OPERATES]-(services)
MATCH (t2: Trip)-[:OPERATES]-(services)
MATCH path = (:Stop{stop_name: "Puskás Ferenc Stadion"})-[st1:STOPS_AT]-(t1)-
[:STOPS_AT*0..1]-(t2)- [st2:STOPS_AT]-(Stop{stop_name: "Bécsi út / Vörösvári út
"})
WHERE time("07:00:00") <=st1.arrival_time <= time("10:00:00")
AND time("07:00:00") <=st2.arrival_time <= time("10:00:00")
AND st1.stop_sequence < st2.stop_sequence
RETURN path
```