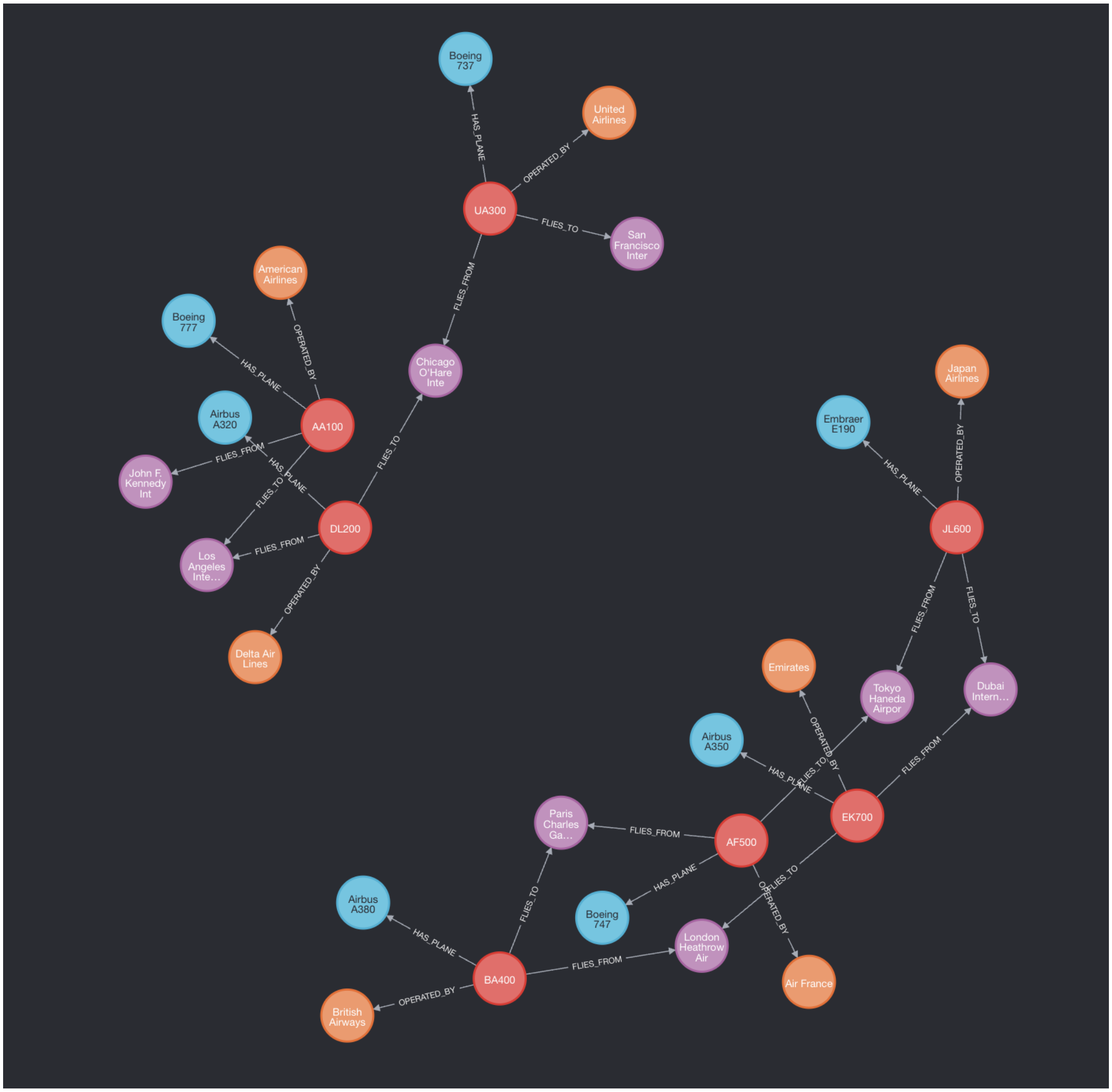
Neo4j

Data Model

```
// Airports
CREATE (a1:Airport {name: "John F. Kennedy International Airport", city: "New York", country: "USA", IATA_code: "JFK", latitude: 40.639751, longitude: -73.778925, elevation: 13, runways: 4})
CREATE (a2:Airport {name: "Los Angeles International Airport", city: "Los Angeles", country: "USA", IATA_code: "LAX", latitude: 33.942536, longitude: -118.407389, elevation: 126, runways: 4})
CREATE (a3:Airport {name: "Chicago O'Hare International Airport", city: "Chicago", country: "USA", IATA_code: "ORD", latitude: 41.978603, longitude: -87.904842, elevation: 672, runways: 7})
CREATE (a4:Airport {name: "San Francisco International Airport", city: "San Francisco", country: "USA", IATA_code: "SFO", latitude: 37.618972, longitude: -122.374889, elevation: 14, runways: 4})
CREATE (a5:Airport {name: "London Heathrow Airport", city: "London", country: "UK", IATA_code: "LHR", latitude: 51.470025, longitude: -0.454972, elevation: 83, runways: 2})
CREATE (a6:Airport {name: "Paris Charles de Gaulle Airport", city: "Paris", country: "France", IATA_code: "CDG", latitude: 49.012779, longitude: 2.55, elevation: 392, runways: 4})
CREATE (a7:Airport {name: "Tokyo Haneda Airport", city: "Tokyo", country: "Japan", IATA_code: "HND", latitude: 35.552258, longitude: 139.779694, elevation: 21, runways: 4})
CREATE (a8:Airport {name: "Dubai International Airport", city: "Dubai", country: "United Arab Emirates", IATA_code: "DXB", latitude: 25.252778, longitude: 55.364444, elevation: 62, runways: 2})
```

```
// Airlines
CREATE (airline1:Airline {name: "American Airlines", headquarters: "Fort Worth, Texas", founded_year: 1926, fleet_size: 1500, passengers_carried: 200000000, destinations: 350})
CREATE (airline2:Airline {name: "Delta Air Lines", headquarters: "Atlanta, Georgia", founded_year: 1924, fleet_size: 1300, passengers_carried: 180000000, destinations: 325})
CREATE (airline3:Airline {name: "United Airlines", headquarters: "Chicago, Illinois", founded_year: 1926, fleet_size: 1200, passengers_carried: 162000000, destinations: 300})
CREATE (airline4:Airline {name: "British Airways", headquarters: "London, UK", founded_year: 1974, fleet_size: 280, passengers_carried: 45000000, destinations: 200})
CREATE (airline5:Airline {name: "Air France", headquarters: "Paris, France", founded_year: 1933, fleet_size: 219, passengers_carried: 51000000, destinations: 195})
CREATE (airline6:Airline {name: "Japan Airlines", headquarters: "Tokyo, Japan", founded_year: 1951, fleet_size: 170, passengers_carried: 35000000, destinations: 220})
CREATE (airline7:Airline {name: "Emirates", headquarters: "Dubai, UAE", founded_year: 1985, fleet_size: 270, passengers_carried: 58000000, destinations: 150})

// Airplanes
CREATE (plane1:Airplane {model: "Boeing 777", manufacturer: "Boeing", seating_capacity: 396, registration_number: "N123AA", range: 7725, cruising_speed: 590})
CREATE (plane2:Airplane {model: "Airbus A320", manufacturer: "Airbus", seating_capacity: 180, registration_number: "N456DA", range: 3300, cruising_speed: 515})
CREATE (plane3:Airplane {model: "Boeing 737", manufacturer: "Boeing", seating_capacity: 160, registration_number: "N789UA", range: 3550, cruising_speed: 485})
CREATE (plane4:Airplane {model: "Airbus A380", manufacturer: "Airbus", seating_capacity: 853, registration_number: "F-HPJ
E", range: 8200, cruising_speed: 560})
CREATE (plane5:Airplane {model: "Boeing 747", manufacturer: "Boeing", seating_capacity: 605, registration_number: "N123BA", range: 6100, cruising_speed: 570})
CREATE (plane6:Airplane {model: "Embraer E190", manufacturer: "Embraer", seating_capacity: 114, registration_number: "N456JA", range: 2450, cruising_speed: 450})
CREATE (plane7:Airplane {model: "Airbus A350", manufacturer: "Airbus", seating_capacity: 325, registration_number: "F-WZFG", range: 7600, cruising_speed: 560})

// Flights
CREATE (flight1:Flight {flight_number: "AA100", date: "2024-04-13", departure_time: "15:00", arrival_time: "18:00", duration: 180, distance: 2148})
CREATE (flight2:Flight {flight_number: "DL200", date: "2024-04-14", departure_time: "09:00", arrival_time: "12:00", duration: 180, distance: 1740})
CREATE (flight3:Flight {flight_number: "UA300", date: "2024-04-15", departure_time: "07:00", arrival_time: "10:00", duration: 180, distance: 1857})
CREATE (flight4:Flight {flight_number: "BA400", date: "2024-04-16", departure_time: "11:00", arrival_time: "13:30", duration: 150, distance: 340})
CREATE (flight5:Flight {flight_number: "AF500", date: "2024-04-17", departure_time: "06:00", arrival_time: "09:00", duration: 180, distance: 540})
CREATE (flight6:Flight {flight_number: "JL600", date: "2024-04-18", departure_time: "14:00", arrival_time: "17:30", duration: 210, distance: 5932})
CREATE (flight7:Flight {flight_number: "EK700", date: "2024-04-19", departure_time: "22:00", arrival_time: "07:00", duration: 540, distance: 5675})

// Relationships
CREATE (flight1)-[:FLIES_FROM]->(a1)
CREATE (flight1)-[:FLIES_TO]->(a2)
CREATE (flight1)-[:OPERATED_BY]->(airline1)
CREATE (flight1)-[:HAS_PLANE]->(plane1)

CREATE (flight2)-[:FLIES_FROM]->(a2)
CREATE (flight2)-[:FLIES_TO]->(a3)
CREATE (flight2)-[:OPERATED_BY]->(airline2)
CREATE (flight2)-[:HAS_PLANE]->(plane2)

CREATE (flight3)-[:FLIES_FROM]->(a3)
CREATE (flight3)-[:FLIES_TO]->(a4)
CREATE (flight3)-[:OPERATED_BY]->(airline3)
CREATE (flight3)-[:HAS_PLANE]->(plane3)

CREATE (flight4)-[:FLIES_FROM]->(a5)
CREATE (flight4)-[:FLIES_TO]->(a6)
CREATE (flight4)-[:OPERATED_BY]->(airline4)
CREATE (flight4)-[:HAS_PLANE]->(plane4)

CREATE (flight5)-[:FLIES_FROM]->(a6)
CREATE (flight5)-[:FLIES_TO]->(a7)
CREATE (flight5)-[:OPERATED_BY]->(airline5)
CREATE (flight5)-[:HAS_PLANE]->(plane5)

CREATE (flight6)-[:FLIES_FROM]->(a7)
CREATE (flight6)-[:FLIES_TO]->(a8)
CREATE (flight6)-[:OPERATED_BY]->(airline6)
CREATE (flight6)-[:HAS_PLANE]->(plane6)

CREATE (flight7)-[:FLIES_FROM]->(a8)
CREATE (flight7)-[:FLIES_TO]->(a5)
CREATE (flight7)-[:OPERATED_BY]->(airline7)
CREATE (flight7)-[:HAS_PLANE]->(plane7)
```



Explanation of the Data Model

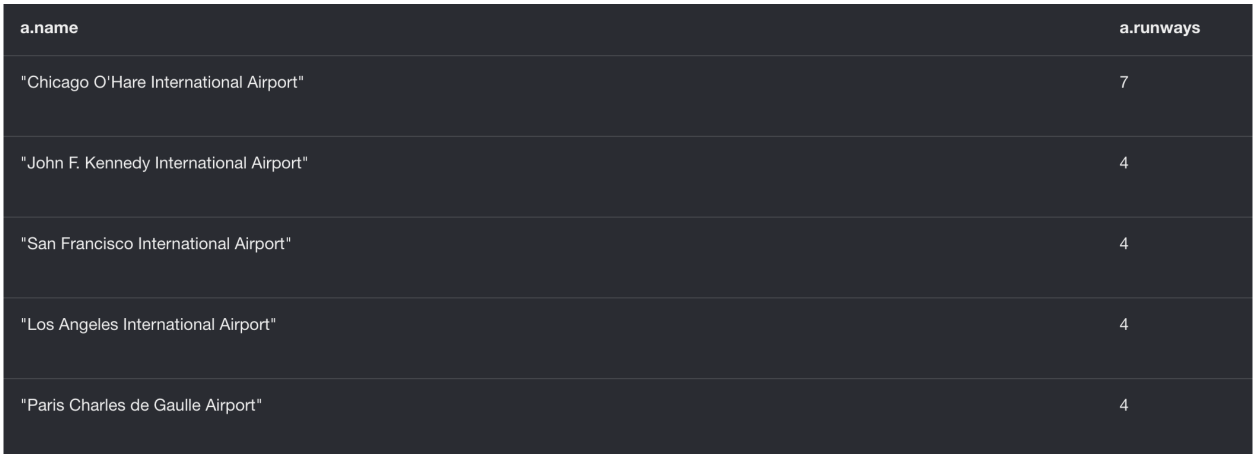
- Chosen Domain: International Air Travel Schedule for the Week of April 13th
- Collected Information:
    - Four Types of Nodes:
        - Airports
            - Name, Location (City/Country), Airport Code, Geographical Pinpoint (Latitude/Longitude, Elevation), Number of Runways

        - Airlines
            - Name, HQ City, Founding Year, Fleet Size, Volume of Passengers Reached, Number of Destinations
        - Airplanes
            - Model, Manufacturer, Seating Capacity, Registration Number, Range, Cruising Speed
        - Flights
            - Number, Date, Departure/Arrival Times, Duration of Flight, Distance Traveled
- Relationships
    - Origin <> Destination- Connecting flights to their departure and arrival airports, respectively.
    - Flight <> Airline- Linking flights to the airlines that operate them.
    - Flight <> Airplane- Associating flights with specific airplanes.

Queries

> ❓ *The most important thing when scheduling flights is to understand if a plane can even land/take off in the first place. Hence, knowing the number of runways can be crucial.*
>
> **Which airports have the most runways? Return the top five airports.**

```
MATCH (a:Airport)
RETURN a.name, a.runways
ORDER BY a.runways DESC
LIMIT 5
```

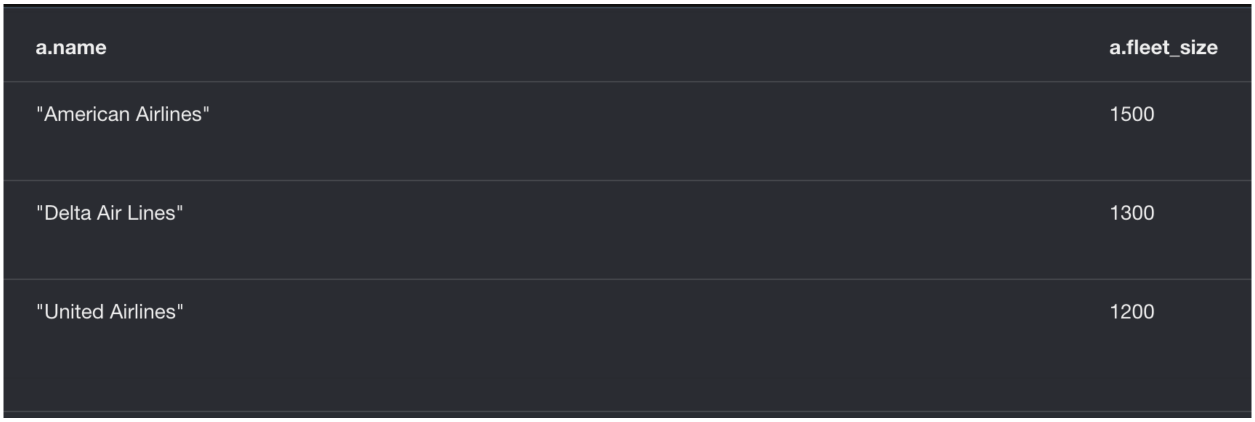


a.name	a.runways
"Chicago O'Hare International Airport"	7
"John F. Kennedy International Airport"	4
"San Francisco International Airport"	4
"Los Angeles International Airport"	4
"Paris Charles de Gaulle Airport"	4

❓ *For an external, third-party partner/service provider (eg. Amex), there may be strategic value in knowing which airlines have the largest fleets, hence are most desirable to partner with (eg. rewards points).*

Which airlines have the largest fleets? Return the top three airlines.

```
MATCH (a:Airline)
RETURN a.name, a.fleet_size
ORDER BY a.fleet_size DESC
LIMIT 3
```

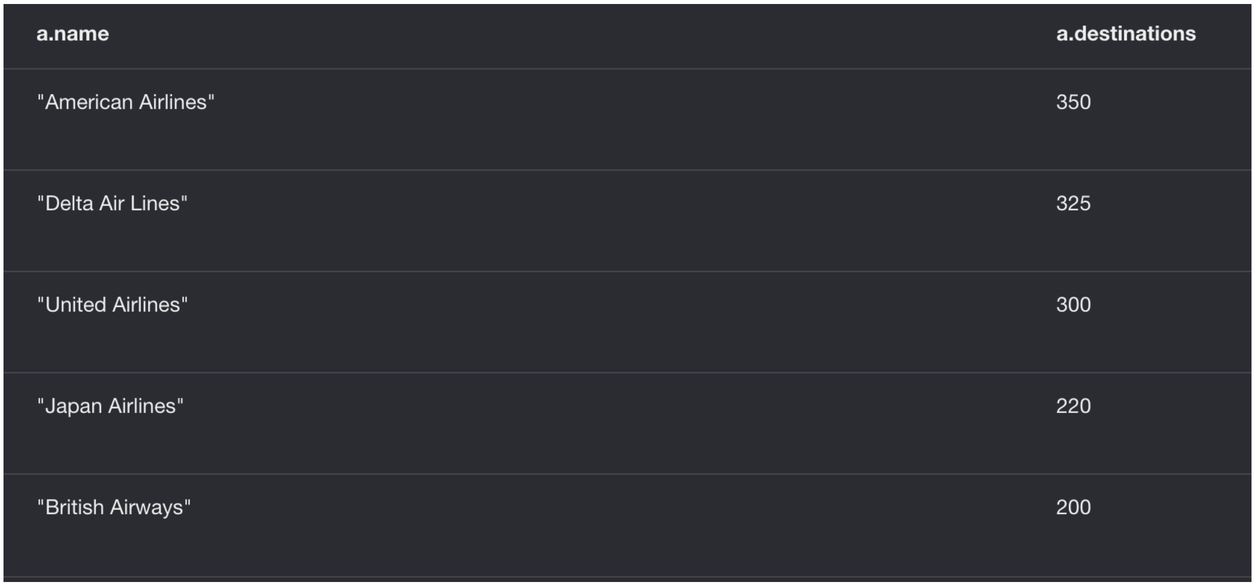


a.name	a.fleet_size
"American Airlines"	1500
"Delta Air Lines"	1300
"United Airlines"	1200

? *If we are a large company such as JPM that needs to fly employees out on travel trips all over the world, we want to have our airline of choice operate across all the hubs we are interested in. Hence, we want to know which airlines fly the most destinations.*

Return the number of destinations that each airline flies to. Only display the airlines with the most destinations (top 5).

```
MATCH (a:Airline)
RETURN a.name, a.destinations
ORDER BY a.destinations DESC
LIMIT 5
```

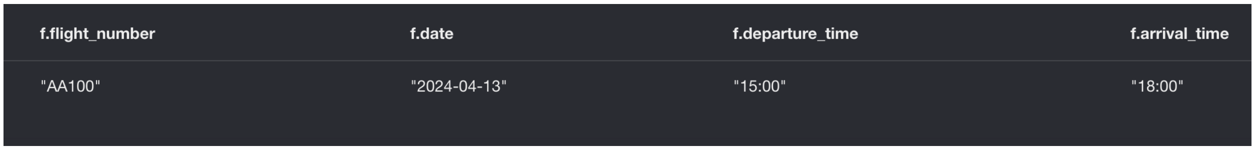


a.name	a.destinations
"American Airlines"	350
"Delta Air Lines"	325
"United Airlines"	300
"Japan Airlines"	220
"British Airways"	200

❓ *And once we find the airline of choice, what if we want to find all the flights that are scheduled?*

Return all flights operated by a given airline. In this case, we look at “American Airlines”.

```
MATCH (f:Flight)-[:OPERATED_BY]->(a:Airline {name: "American Airlines"})
RETURN f.flight_number, f.date, f.departure_time, f.arrival_time
```

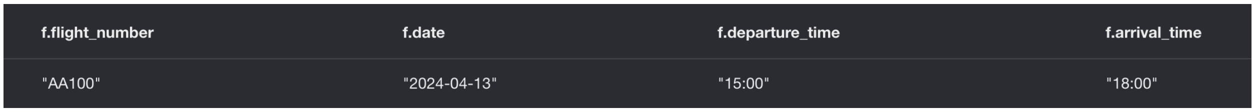


f.flight_number	f.date	f.departure_time	f.arrival_time
"AA100"	"2024-04-13"	"15:00"	"18:00"

❓ *And then, what if we want to find the available flights between chosen origin and destination? (eg. JFK and LAX)*

Return the flights that fly from JFK (New York) to LAX (Los Angeles).

```
MATCH (a1:Airport {IATA_code: "JFK"})<-[:FLIES_FROM]-(f:Flight)-[:FLIES_TO]->(a2:Airport {IATA_code: "LAX"})
RETURN f.flight_number, f.date, f.departure_time, f.arrival_time
```

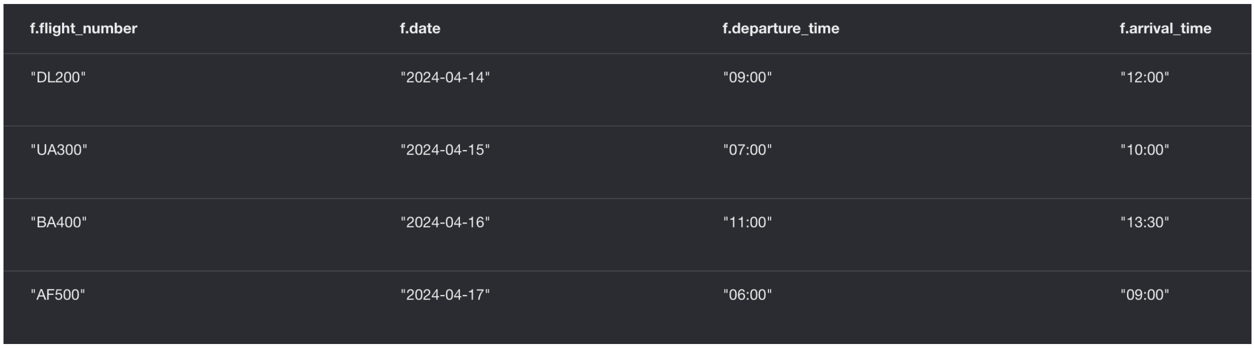


f.flight_number	f.date	f.departure_time	f.arrival_time
"AA100"	"2024-04-13"	"15:00"	"18:00"

? *Departure time can often be important for scheduling purposes. For instance, what if we only want to see the flights that depart in the morning hours?*

Return all flights that depart in the morning (before 12PM).

```
MATCH (f:Flight)
WHERE f.departure_time < '12:00'
RETURN f.flight_number, f.date, f.departure_time, f.arrival_t
ime
```

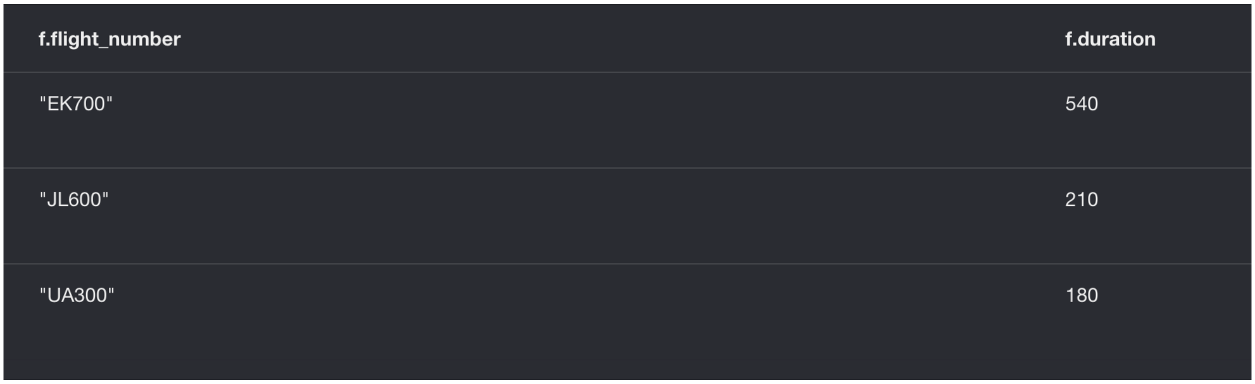


f.flight_number	f.date	f.departure_time	f.arrival_time
"DL200"	"2024-04-14"	"09:00"	"12:00"
"UA300"	"2024-04-15"	"07:00"	"10:00"
"BA400"	"2024-04-16"	"11:00"	"13:30"
"AF500"	"2024-04-17"	"06:00"	"09:00"

? *The duration of flights can be important for airline planning purposes. For instance, if the duration of a flight is longer than 200 minutes, we know that there will need to be two meals served, as opposed to only one.*

What are the longest flights in our data model? Return the top three flights, displaying the flight number and their corresponding duration.

```
MATCH (f:Flight)
RETURN f.flight_number, f.duration
ORDER BY f.duration DESC
LIMIT 3
```

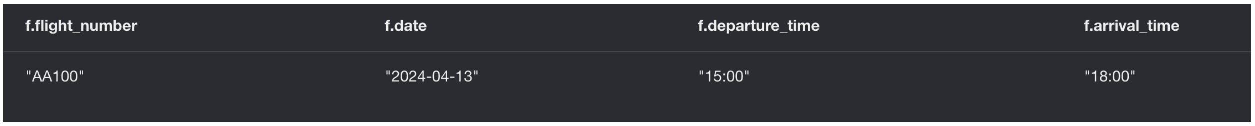


f.flight_number	f.duration
"EK700"	540
"JL600"	210
"UA300"	180

❓ *If we have a highly skeptical client that refuses to board on specific fleets, how do we ensure that they are not put onto a flight that is that specific plane model? (eg. Boeing 777)*

Return all flights operated by a Boeing 777.

```
MATCH (p:Airplane {model: "Boeing 777"})<-[:HAS_PLANE]-(f:Flight)
RETURN f.flight_number, f.date, f.departure_time, f.arrival_time
```

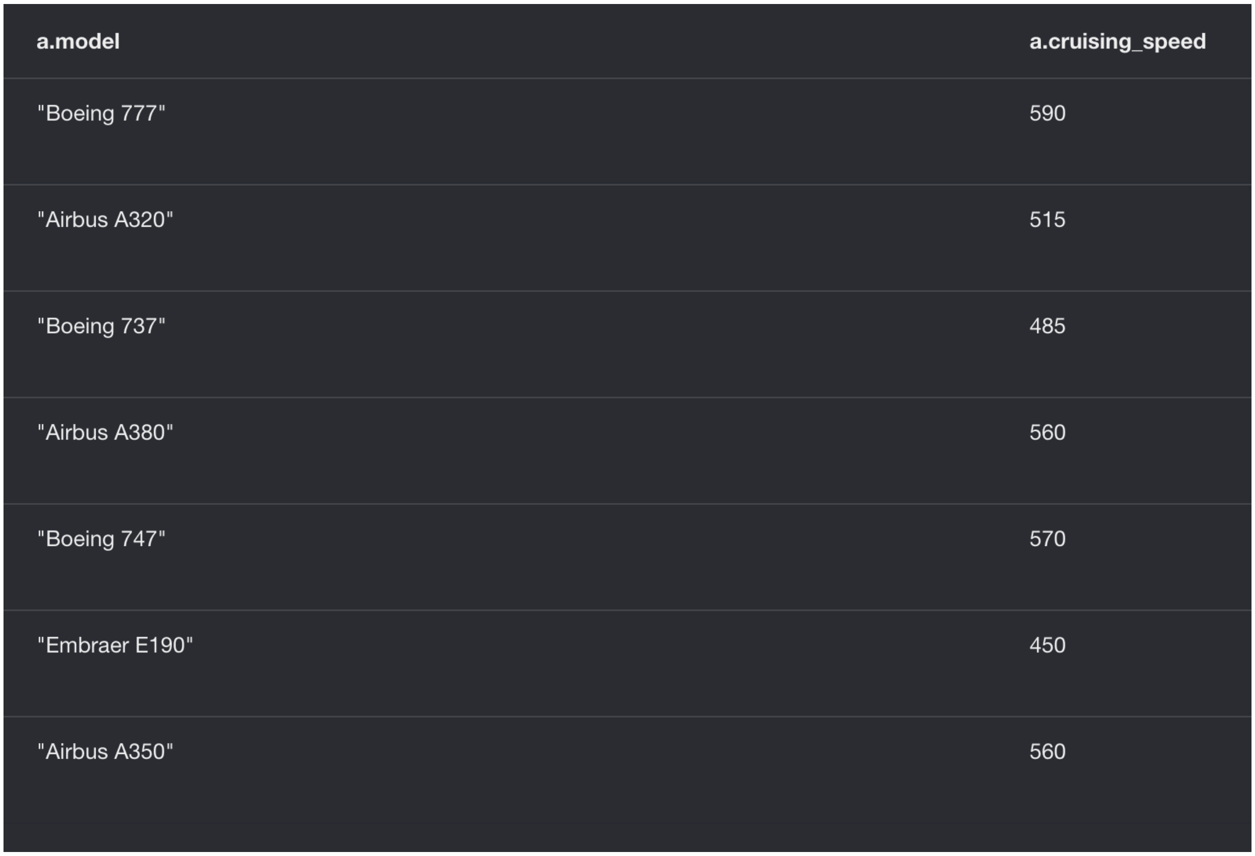


f.flight_number	f.date	f.departure_time	f.arrival_time
"AA100"	"2024-04-13"	"15:00"	"18:00"

Personnel may want to know what the top performing plane models are, and take the cruising speeds as a proxy for such. This can be returned for each of the planes in our data model.

Return the cruising speed for each of the airplanes.

```
MATCH (a:Airplane)
RETURN a.model, a.cruising_speed
```

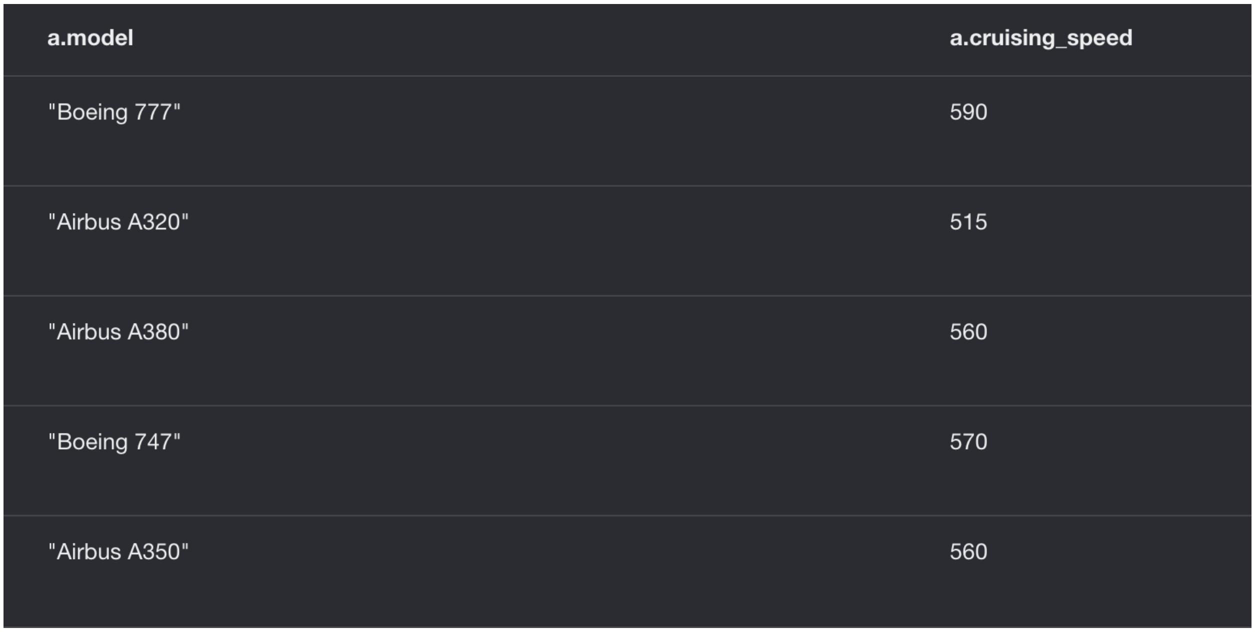


a.model	a.cruising_speed
"Boeing 777"	590
"Airbus A320"	515
"Boeing 737"	485
"Airbus A380"	560
"Boeing 747"	570
"Embraer E190"	450
"Airbus A350"	560

❓ *We can filter this even further so that we only return the planes that meet a given threshold.*

Return all the airplanes that have cruising speeds greater than 500.

```
MATCH (a:Airplane)
WHERE a.cruising_speed > 500
RETURN a.model, a.cruising_speed
```



a.model	a.cruising_speed
"Boeing 777"	590
"Airbus A320"	515
"Airbus A380"	560
"Boeing 747"	570
"Airbus A350"	560