Queries executed via Mongo Compass:

1. Find the total number of flights for each airline, showing the airline code, name, and the count of flights.

-db.flights.aggregate([

{ $group: { \_id: { Airline\_Code: "$Airline\_Code", Airline\_Name: "$Airline\_Name" }, totalFlights: { $sum: 1 } } },

{ $project: { \_id: 0, Airline\_Code: "$\_id.Airline\_Code", Airline\_Name: "$\_id.Airline\_Name", Total\_Flights: "$totalFlights" } }

]);  
A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a phone

Description automatically generated

1. Calculate the average departure delay for each airline, showing the airline code, name, and the average delay.

db.flights.aggregate([

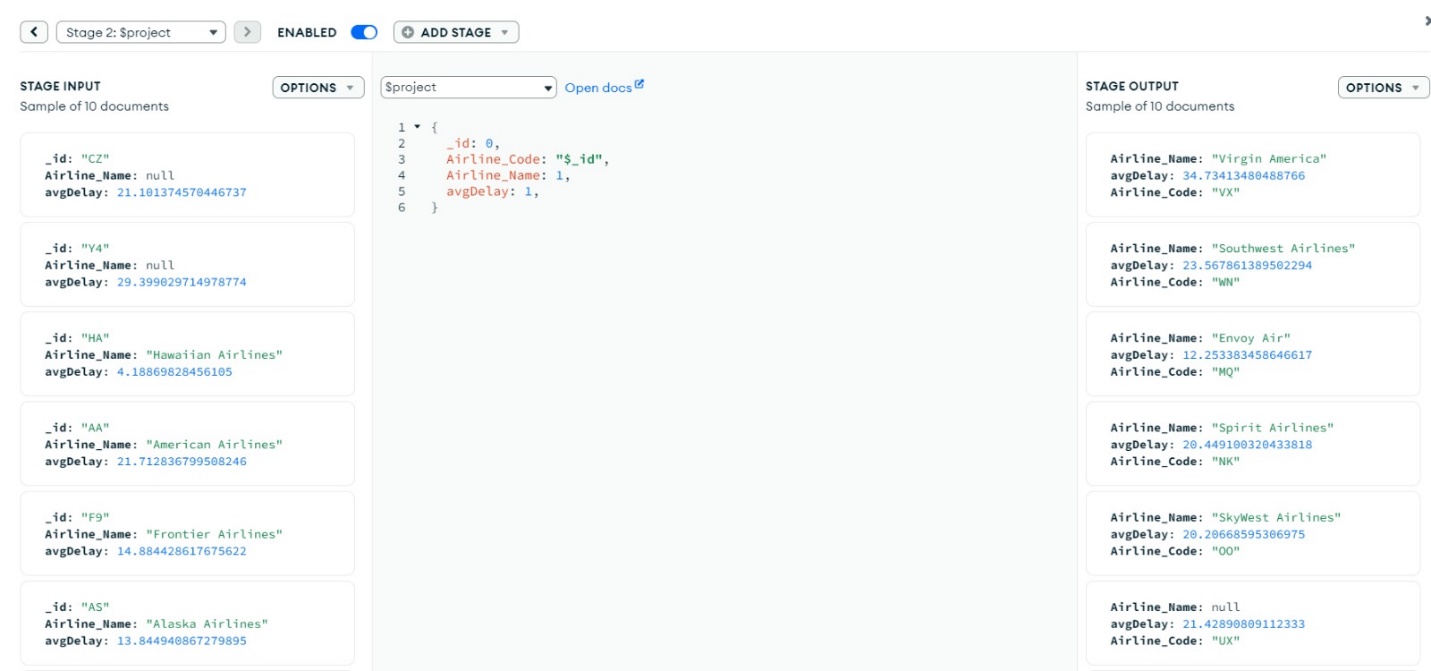
{ $group: { \_id: { Airline\_Code: "$Airline\_Code", Airline\_Name: "$Airline\_Name" }, avgDelay: { $avg: "$DEP\_DELAY\_NEW" } } },

{ $project: { \_id: 0, Airline\_Code: "$\_id.Airline\_Code", Airline\_Name: "$\_id.Airline\_Name", Average\_Departure\_Delay: "$avgDelay" } }

]);

A screenshot of a computer

Description automatically generated



1. Show the count of canceled flights and the cancellation reasons.

db.flights.aggregate([

{ $match: { CANCELLED: 1 } },

{ $group: { \_id: "$CANCELLATION\_CODE", count: { $sum: 1 } } },

{ $project: { \_id: 0, Cancellation\_Code: "$\_id", Count: "$count" } }

]);

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

1. List the top 5 busiest airports based on the total number of departures and arrivals.

db.flights.aggregate([

{ $group: { \_id: "$ORIGIN\_AIRPORT\_ID", totalFlights: { $sum: 1 } } },

{ $sort: { totalFlights: -1 } },

{ $limit: 5 },

{ $lookup: { from: "airports", localField: "\_id", foreignField: "AIRPORT\_ID", as: "AirportInfo" } },

{ $project: { \_id: 0, Airport\_ID: "$\_id", Airport\_Name: "$AirportInfo.ORIGIN", Total\_Flights: "$totalFlights" } }

]);

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

1. Show the distribution of departure delays with categories (e.g., early, on time, minor delay, moderate delay, severe delay).

db.flights.aggregate([

{

$project: {

\_id: 0,

Flight\_ID: 1,

Dep\_Delay\_Category: {

$cond: [

{ $eq: ["$DEP\_DELAY\_NEW", 0] }, "On Time",

{ $lt: ["$DEP\_DELAY\_NEW", 15] }, "Minor Delay",

{ $lt: ["$DEP\_DELAY\_NEW", 60] }, "Moderate Delay",

"Severe Delay"

]

}

}

},

{ $group: { \_id: "$Dep\_Delay\_Category", count: { $sum: 1 } } }

]);

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

1. Calculate the average departure delay for each day of the week.

db.flights.aggregate([

{ $group: { \_id: "$DAY\_OF\_WEEK", avgDelay: { $avg: "$DEP\_DELAY\_NEW" } } },

{ $project: { \_id: 0, Day\_Of\_Week: "$\_id", Average\_Departure\_Delay: "$avgDelay" } }

]);

A screenshot of a computer

Description automatically generated

1. Find the airline with the maximum weather delay, showing the airline code and name.

db.flights.aggregate([

{ $match: { WEATHER\_DELAY: { $gt: 0 } } },

{ $sort: { WEATHER\_DELAY: -1 } },

{ $limit: 1 },

{ $project: { \_id: 0, Airline\_Code: "$Airline\_Code", Airline\_Name: "$Airline\_Name", Max\_Weather\_Delay: "$WEATHER\_DELAY" } }

]);

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

1. Show the total number of flights for each month.

db.flights.aggregate([

{ $group: { \_id: { $month: "$FL\_DATE" }, totalFlights: { $sum: 1 } } },

{ $project: { \_id: 0, Month: "$\_id", Total\_Flights: "$totalFlights" } }

]);

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

1. Determine the busiest day of the week based on the total number of flights.

db.flights.aggregate([

{ $group: { \_id: "$DAY\_OF\_WEEK", totalFlights: { $sum: 1 } } },

{ $sort: { totalFlights: -1 } },

{ $limit: 1 },

{ $project: { \_id: 0, Busiest\_Day: "$\_id", Total\_Flights: "$totalFlights" } }

]);

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Show the count of on-time, delayed, and canceled flights for each route (origin-destination pair).

db.flights.aggregate([

{

$project: {

\_id: 0,

Route: { $concat: ["$ORIGIN", "-", "$DEST"] },

Flight\_Status: {

$switch: {

branches: [

{ case: { $eq: ["$CANCELLED", 1] }, then: "Canceled" },

{ case: { $gte: ["$DEP\_DELAY\_NEW", 15] }, then: "Delayed" },

{ case: { $and: [ { $eq: ["$CANCELLED", 0] }, { $eq: ["$DEP\_DELAY\_NEW", 0] } ] }, then: "On Time" }

],

default: "Unknown"

}

}

}

},

{ $group: { \_id: { Route: "$Route", Flight\_Status: "$Flight\_Status" }, count: { $sum: 1 } } }

]);

A screenshot of a computer

Description automatically generated

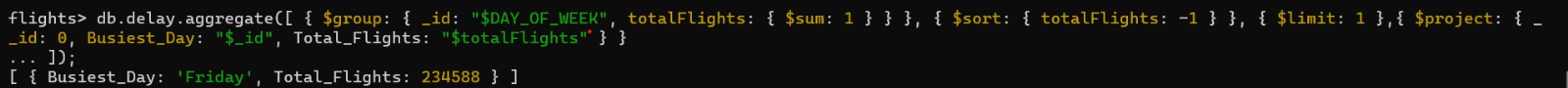
Queries executed via Mongo Shell:

Calculate the average departure delay for each airport (origin and destination combined).

A screenshot of a computer

Description automatically generated

Determine the busiest day of the week based on the total number of flights.



Identify the peak departure times during the day.

A computer screen with text on it

Description automatically generated