## **INDEX**

* [**Part 1 – Designing the Airlines Database, storing information and setting up queries**](#_Part_1_–)
  + **Assumptions**
  + **Data Structure**
    - **Setting up the collections in our database**
    - **Collections in the database**
  + **Queries**
    - **Find() Queries**
    - **Aggregate() Queries**
* [**Part 2 – Performance Tuning**](#_Part_2_–)
  + **Use of Indexes and comparison of the “Explain” output**
  + **Use of the MongoDB profiler for examining performance**
* [**Appendix**](#_Appendix)

## **Part 1 – Designing the Airlines Database, storing information and setting up queries**

**Assumptions:**

During setting up the database, our team considered the below assumptions:

* All pilots can perform as pilots or co pilots
* Each airport has its own "cost of use" price, which is fixed for all plane models
  + Cost of use is zero for UK airports
* Cost of fuel is for a full tank
* Salaries of employees and pilots are per month
* All flights only have 2 seating options – Business and Economy
* For every flight, bookings are split into two groups – Business and Economy
  + Each booking document (stored in the collection journey\_details) contains all the passengers taking that flight
  + All passengers in one document have the same booking date
  + Each booking document has only one booking agent
  + Assume if passenger booked business or economy, it will be for the whole journey
* The price of tickets for each flight is consistent and does not increase or decrease by date – only the price for business and economy class differs

Diagram

Description automatically generated**Data Structure:**

**Setting up the collections in our database**:

All our collections have a validation schema to ensure data integrity. We have used JSON Schema validation to achieve the same.

**Journey Details Schema Validation:**

{

$jsonSchema: {

bsonType: 'object',

required: [

'journey\_id',

'flight\_id',

'booking\_agent\_id',

'booking\_type',

'passenger\_ids',

'booking\_date'

],

properties: {

journey\_id: {

bsonType: 'string',

description: 'PNR cannot be empty, must be a string, and unique'

},

flight\_id: {

bsonType: 'array',

items: {

bsonType: 'string'

},

description: 'There must be atleast one flight'

},

booking\_agent\_id: {

bsonType: 'string',

description: 'Booking Agent ID cannot be empty, must be a string'

},

booking\_type: {

'enum': [

'business',

'economy'

],

description: 'Booking type can be either Business or Economy'

},

passenger\_ids: {

bsonType: 'array',

items: {

bsonType: 'string'

},

description: 'There must be atleast one passenger'

},

booking\_date: {

bsonType: 'date',

description: 'Booking date cannot be empty, must be a date, in ISO format'

}

}

}

}

**For instance**, on the right is the schema validation used for journey\_details. For a complete list of all our schema validation code, please refer to the appendix ([here](#_Appendix)).

The schema for journey details contains the fields *journey\_id*, *flight\_id*, *booking\_agent\_id*, *booking\_type, passenger\_ids* and *booking\_date*.

*flight\_id* is set as an array to accommodate trips that may include switching flights, round trips etc. *booking\_type* is set to only accept either “*business*” or “*economy*” as per the bookings. *passenger\_ids* have also been set as arrays to accommodate the full list of passenger ids (~3.9k)

All schema validations are set to the default value, strict, and throw an error on wrong inputs.

The next step in setting up our collections was to assign a unique index, for each collection, using *createIndex.*

Below is the code used for setting up *journey\_id* as the index for the journey\_details collection:

*db.journey\_details.createIndex( { "journey\_id": 1 }, { unique: true } )*

Finally, we populated our collections with the data, abiding by the schema validation.

We used *insertMany* to insert multiple documents in one go. An excerpt from the code used for journey\_details collection’s documents is below:

*db.journey\_details.insertMany ([*

*{*

*journey\_id: 'JAIR45B1',*

*flight\_id: ['air45\_a2'],*

*booking\_agent\_id: 'rt889',*

*booking\_type: 'business',*

*passenger\_ids:['A45\_1310','A45\_18E1','A45\_813B'],*

*booking\_date:ISODate("2021-10-25")*

*},…*

For the full set of codes for all collections, please visit the appendix ([here](#_Appendix)).

**Collections in the database**:

Diagram

Description automatically generatedThe collections in the database have different relationships to better fit the need of an airline database. We have used fields like *flight\_id, pilot\_id* etc. to create a link between all the collections. Embedded documents have been used for grouping together relative information, so that the whole set can be retrieved together, if needed. Arrays have also been utilised to allow us to have a flexible number of entries yet maintain the datatype and enforce the schema validation. Below is the details for all our collections:

1. **Airports**: The airports collection contains the airport id, name, location (nested document) that contains city and country, the cost to refuel a plane (in GBP), and the cost of use per hour (also in GBP). *airport\_id* is set as the index for this collection and is used for linking with other collections.
2. Diagram

   Description automatically generated**Employees**: The employees collection contains the documents for all our staff apart from our pilots. These include stewards and air hostesses, as well as booking agents. The data for each employee consists of their *employee\_id* (which is the index for this collection), names (nested document with firstname and lastname), addresses (also a nested document) broken down to street, region and pin code, their contact details (also a nested document) which includes email, phone number and an emergency number. At last, we also have the contract details (also a nested document), which includes start and end dates along with the employee’s position and salary.
3. Diagram

   Description automatically generated**Flights**: The flights collection is the most important collections in our database with many to one relation. Every document in the flights collection signifies a complete trip to a certain destination. The documents contain the details for the *flight\_id* (the index), the plane id, linked to the *planes* collection, pilot ids (nested document), linked to the *pilots* collection, that lists the pilot as well co-pilot’s ids, route (nested document), linked to *airports* collection, that has the starting airport id and destination airport id, cabin crew ids (array), which are linked to the *employees* collection and have the employee ids, timings (nested document) with taking off and landing time along with date, duration of the flight and finally the price of tickets (nested document), split into business and economy, in GBP.
4. Diagram

   Description automatically generated**Journey\_details**: The journey\_details is another important collection as it contains all the details about the bookings of flights. *journey\_id* is used as the index along with flight\_id, linked to the *flights* collection, the booking agent id, linked to *employees* collection using the employee ids, an array of all passenger ids and finally the booking date
5. Diagram

   Description automatically generated**Pilots**: Pilots is very similar to the *employees* collection, and has the same details, the only difference being the designation.
6. Diagram

   Description automatically generated**Planes**: The planes collection has the documents that contain the details for each flight owned by the airline. This includes *plane\_id* (the index), the brand and the model of the plane, the range, seating capacity (nested document) split into business and economy, length of service and remaining length of service in months and the status of the flight (repair, working etc.)

**Queries**

Below listed are the queries that we have created to retrieve the relevant information from our database:

**Queries based on find():**

* + - 1. A list of pilot names sorted by the earliest “fit to fly” test date
      2. A list of all stewards working for the airline
      3. A list of planes under “repair” status

**Queries based on aggregate():**

* + - 1. Next available pilot in London Heathrow Airport
      2. Latest location from which a flight landed in London
      3. Itinerary for all flights
      4. Pilot with the highest number of flying hours in the period
      5. Percentage of total flying capacity that was utilised in the period
      6. Total Revenue for the period
      7. Opportunity Cost in the period
      8. Booking agent with the highest revenue
      9. Route with the highest revenue
      10. Net Profit for the period

**Find() Queries:**

Some of the simple queries that we wrote using find().

1. **A list of pilot names sorted by the earliest “fit to fly” test date:**

*db.pilots.find ({}, {fullName:1, \_id:0}).sort({lastFitToFlyTest:1})*

Text, calendar

Description automatically generatedOutput:

1. **A list of all stewards working for the airline:**

*db.employees.find({"contract.position" : "steward"},{"name" : true, "contact\_details.number" : true, "\_id" : false})*

Output:

Chart

Description automatically generated

1. **A list of planes under “repair” status:**

*db.planes.find({"status" : "repair"},{"plane\_id" : true, "brand" : true,"model" : true, "\_id" : false})*

Output:

**

**Aggregate() Queries:**

1. **Next available pilot in London Heathrow Airport:** A query that fetches the next pilot that is available at the London Heathrow Airport. The query uses lookup to join the collections *flights* and *pilots* as we need to pull the name of the pilot based on *pilot\_id* from the flights table.

*db.flights.aggregate(*

*{$match: {"route.destination\_airport\_id": "LHR"}},*

*{$sort: {"timings.landing\_date\_time":1}},*

*{$limit: 1},*

*{$lookup:*

*{from: 'pilots',*

*localField: "pilot\_id.pilot\_id",*

*foreignField: "pilot\_id",*

*as: "p\_id"*

*}*

*},*

*{$group: {\_id: "1", pilotName : {$push: "$p\_id.fullName"}}},*

*)*

Output:



1. **Latest flight that landed in London:** Retrieves the last airport from which a flight landed in London. We can also use a similar approach to find flights leaving London, be it the latest one or for a particular date.

*db.flights.aggregate(*

*{$match:*

*{"route.destination\_airport\_id": "LHR"}*

*},*

*{$sort:*

*{"timings.start\_date\_time": -1}*

*},*

*{$limit: 1},*

*{$lookup:*

*{from: 'airports',*

*localField: "route.start\_airport\_id",*

*foreignField: "airport\_id",*

*as: "airportInfo"*

*}*

*},*

*{$unwind: "$airportInfo"},*

*{$project:*

*{\_id: 0, AirportName:"$airportInfo.name"}*

*} )*

Output:

In order to obtain **neat and formatted outputs**, we’ve adopted the use of the $concat aggregation operator **throughout our following queries**, the output is converted into a string in the case that it’s not, and then the output unit is added (hours, km, GBP, etc…).

1. **Itinerary for all flights**: This query retrieves all the relevant information for the flights from the airline in the period. The details include the route, timings, distance, flight duration and the price of seats.

*db.flights.aggregate([*

*{$project: {*

*\_id: 0,*

*DepartingFrom: "$route.start\_airport\_id",*

*LandingIn: "$route.destination\_airport\_id",*

*DepartureTime: { $dateToString: { format: "%Y-%m-%d %H:%M", date: "$timings.start\_date\_time"} },*

*LandingTime: { $dateToString: { format: "%Y-%m-%d %H:%M", date: "$timings.landing\_date\_time"} },*

*DistanceInKm: {$concat: [{$toString: "$distance\_km"}, " km"]},*

*TripDuration: {$concat: [{$toString: "$duration\_hrs"}, " hours"]},*

*EconomyPrice: {$concat: [{$toString: "$price\_gbp.economy"}, " GBP"]},*

*BusinessPrice: {$concat: [{$toString: "$price\_gbp.business"}, " GBP"]},*

*}*

*}*

*])*

Output:

Text

Description automatically generated

Text

Description automatically generatedText

Description automatically generated

1. **Pilot with the highest number of flying hours in the period:** The query usesthe flights and pilots collection again to find the pilot with highest number of flying hours using the duration of the flights.

*db.flights.aggregate([*

*{$project:*

*{\_id:0, pilot: "$pilot\_id.pilot\_id", fullName:1, flight\_hrs:*

*{$sum:"$duration\_hrs"}*

*}*

*},*

*{ "$group": {*

*"\_id": "$pilot",*

*"totalFlightHrs": {*

*"$sum": "$flight\_hrs"*

*}*

*}*

*},*

*{$sort: {"totalFlightHrs": -1}},*

*{$limit: 1},*

*{$lookup:*

*{from: 'pilots',*

*localField: "\_id",*

*foreignField: "pilot\_id",*

*as: "plt\_id"*

*}*

*},*

*{$unwind: "$plt\_id"},*

*{$project:*

*{\_id:0, pilotName: "$plt\_id.fullName", FlyingHours:*

*{$concat: [{$toString: "$totalFlightHrs"}, " hours"]},*

*}*

*},*

*])*

Output:

****

1. **Percentage of total flying capacity that was utilised in the period:** The query calculates the percentage of seats that were occupied in all the flights during the period. The seats include business as well as economy. The query uses multiple lookups to connect all the required collections. We also use $cond and $regexMatch to build a conditional statement that selects the price of business or economy seats depending on the booking.

*db.journey\_details.aggregate([*

*{*

*$lookup:{*

*from:"flights",*

*localField:"flight\_id",*

*foreignField:"flight\_id",*

*as:"f\_id"*

*}*

*},*

*{$unwind:"$f\_id"},*

*{*

*$lookup:{*

*from:"planes",*

*localField:"f\_id.plane\_id",*

*foreignField:"plane\_id",*

*as:"p\_id"*

*}*

*},*

*{$unwind:"$p\_id"},*

*{$project:*

*{\_id:0, flight\_id:1, booking\_type:1, plane\_id:1, cap:*

*{$multiply:[100,{$divide:[*

*{$size:"$passenger\_ids"},{$cond: [{$regexMatch: {input:"economy", regex: "$booking\_type"}}, "$p\_id.capacity.economy", "$p\_id.capacity.business"]}]}]}*

*}*

*},*

*{ "$group": {*

*"\_id": null,*

*"OverallCap": {*

*"$avg": "$cap"*

*}*

*} },*

*{ $project:*

*{\_id:0, OverallCapactiyUsed: {*

*$concat: [{$toString: "$OverallCap"}, "%"]},*

*}*

*}*

*])*

Output:



1. **Total Revenue for the period:** The query calculates the total revenue generated by the flights in the given period. Similar to the previous query we use multiple lookups to get the required data. Similar to the previous query, we use $cond and $regexMatch to get the right calculation based on business/economy bookings.

Please note, this is the revenue, we calculate the Net Profit, i.e., total money from bookings – airport costs – total salary bill, further ahead.

db.journey\_details.aggregate([

{$lookup:

{from: 'flights',

localField: "flight\_id",

foreignField: "flight\_id",

as: "f\_id"

}

},

{$project:

{\_id:0, flight\_id:1, booking\_type:1, rev:

{$multiply: [{$sum: {$cond: [{$regexMatch: {input:"economy", regex: "$booking\_type"}}, "$f\_id.price\_gbp.economy", "$f\_id.price\_gbp.business"]}},{$size:"$passenger\_ids"}]}

}

},

{ "$group": {

"\_id": null,

"totalRevenue": {

"$sum": "$rev"

}

}

},

{ $project:

{\_id:0, TotalRevenue: {$concat: [{$toString: "$totalRevenue"}, " GBP"]}

} }

])

Output:



1. **Opportunity Cost in the period:** The query retrieves the opportunity cost (the potential amount the airline could have earned had the remaining seats been filled).

*db.journey\_details.aggregate([*

*{ $lookup:{*

*from:"flights",*

*localField:"flight\_id",*

*foreignField:"flight\_id",*

*as:"f\_id"*

*}*

*},*

*{$unwind:"$f\_id"},*

*{$lookup:{*

*from:"planes",*

*localField:"f\_id.plane\_id",*

*foreignField:"plane\_id",*

*as:"p\_id"*

*}*

*},*

*{$unwind:"$p\_id"},*

*{$project:*

*{\_id:0, flight\_id:1, booking\_type:1, plane\_id:1, oppcost:*

*{$multiply:[{$subtract:[{$cond: [{$regexMatch: {input:"economy", regex: "$booking\_type"}}, "$p\_id.capacity.economy", "$p\_id.capacity.business"]},{$size:"$passenger\_ids"}]},{$cond: [{$regexMatch: {input:"economy", regex: "$booking\_type"}}, "$f\_id.price\_gbp.economy", "$f\_id.price\_gbp.business"]}]}*

*}*

*},*

*{ "$group": {*

*"\_id": null,*

*"totalOppCost": {*

*"$sum": "$oppcost"*

*}*

*} },*

*{ $project: {*

*\_id:0, TotalOpportunityCost:*

*{$concat: [{$toString: "$totalOppCost"}, " GBP"]}*

*} }*

*])*

Output:



1. **Booking agent with the highest revenue:** The query retrieves the top 3 booking agents based on the revenue generated as per the bookings in the period

*db.journey\_details.aggregate([*

*{$lookup:*

*{from: 'flights',*

*localField: "flight\_id",*

*foreignField: "flight\_id",*

*as: "f\_id"*

*}*

*},*

*{$unwind: "$f\_id"},*

*{$project:*

*{\_id:0, flight\_id:1, booking\_type:1, booking\_agent\_id: 1, rev:*

*{$multiply: [{$sum: {$cond: [{$regexMatch: {input:"economy", regex: "$booking\_type"}}, "$f\_id.price\_gbp.economy", "$f\_id.price\_gbp.business"]}},{$size:"$passenger\_ids"}]}*

*}*

*},*

*{ $group: {*

*"\_id": "$booking\_agent\_id",*

*"totalRevenue": {*

*"$sum": "$rev"*

*}*

*}*

*},*

*{$sort: {"totalRevenue":-1}},*

*{$limit: 3},*

*{$lookup:*

*{from: 'employees',*

*localField: "\_id",*

*foreignField: "employee\_id",*

*as: "emp"*

*}*

*},*

*{$unwind: "$emp"},*

*{$project:*

*{\_id:0, employeeName: {$concat: [{$toString: "$emp.name.firstname"}," " ,{$toString: "$emp.name.lastname"}]},*

*totalRevenue: {$concat: [{$toString: "$totalRevenue"}, " GBP"]},*

*}*

*},*

*])*

Output:

**A screenshot of a computer

Description automatically generated with low confidence**

1. **Route with the highest revenue:** One of the trickier queries, this one outputs the route that has the highest revenue. It takes into consideration the total revenue earned by business and economy tickets, for both to and from the destinations. The price of the tickets for to and from flights are different.

*db.flights.aggregate([*

*{$lookup:*

*{from: 'journey\_details',*

*localField: "flight\_id",*

*foreignField: "flight\_id",*

*as: "f\_id"*

*}*

*},*

*{$unwind:"$f\_id"},*

*{$project:*

*{\_id:0, flight\_id:1, plane\_id: 1, name: "$route.destination\_airport\_id", rev:*

*{$multiply: [{$sum: {$cond: [{$regexMatch: {input:"economy", regex: "$f\_id.booking\_type"}}, "$price\_gbp.economy", "$price\_gbp.business"]}},{$size:"$f\_id.passenger\_ids"}]}*

*}*

*},*

*{ $group:*

*{*

*"\_id": "$plane\_id",*

*"totalRevenue":*

*{*

*"$sum": "$rev"*

*}*

*}*

*},*

*{$sort: {"totalRevenue": -1}},*

*{$limit: 1},*

*{$lookup:*

*{from: 'flights',*

*localField: "\_id",*

*foreignField: "plane\_id",*

*as: "route"*

*}*

*},*

*{$unwind: "$route"},*

*{$sort: {"route.timings.start\_date\_time": 1}},*

*{$limit: 1},*

*{*

*$project:*

*{\_id:0,*

*Route: {$concat: [{$toString: "$route.route.start\_airport\_id"}, " - ",{$toString: "$route.route.destination\_airport\_id"}," - ", {$toString: "$route.route.start\_airport\_id"}]},*

*RouteRevenue: {$concat: [{$toString: "$totalRevenue"}, " GBP"]},*

*}*

*}*

*])*

Output:



1. **Net Profit for the period:** Finally, obtaining the net profit was performed using a sequence of four queries, the reason for that is in order to calculate it, access is needed for multiple collections, this could not be obtained using lookups. This consists of four aggregation queries, 3 of which write their final results to a new collection “netprofit” using the $merge stage, and a 4th one which calculates the net profit from those 3 values.

1- FIRST QUERY

*db.journey\_details.aggregate([*

*{$match: {"booking\_type": "economy"}},*

*{$lookup:*

*{from: 'flights',*

*localField: "flight\_id",*

*foreignField: "flight\_id",*

*as: "f\_id"*

*}*

*},*

*{$lookup:*

*{from: 'planes',*

*localField: "f\_id.plane\_id",*

*foreignField: "plane\_id",*

*as: "p\_id"*

*}*

*},*

*{$lookup:*

*{from: 'airports',*

*localField: "f\_id.route.start\_airport\_id",*

*foreignField: "airport\_id",*

*as: "start\_airport\_id"*

*}*

*},*

*{$lookup:*

*{from: 'airports',*

*localField: "f\_id.route.destination\_airport\_id",*

*foreignField: "airport\_id",*

*as: "destination\_airport\_id"*

*}*

*},*

*{$project: {*

*profit:{ $subtract: [*

*{$multiply: [*

*{$sum: {$cond: [*

*{$regexMatch: {input:"economy", regex: "$booking\_type"}},*

*"$f\_id.price\_gbp.economy", "$f\_id.price\_gbp.business"*

*]*

*}*

*},*

*{$size:"$passenger\_ids"}*

*]},*

*{$sum: [{$multiply: [2,{$sum: "$destination\_airport\_id.costOfUsePerHour\_GBP"}]}, {$sum: "$start\_airport\_id.costOfUsePerHour\_GBP"}, {$sum: "$start\_airport\_id.refuelCost\_GBP"}, {$sum: "$destination\_airport\_id.refuelCost\_GBP"}]*

*}]*

*}*

*}},*

*{ "$group": {*

*"\_id": "journeyRev",*

*"amount": {*

*"$sum": "$profit"*

*}*

*} },*

*{ $merge:{into : "netprofit"} }*

*])*

2- SECOND QUERY

*db.employees.aggregate( [*

*{ $group : { \_id : "employeesSalary", amount: { $sum: "$contract.salary"}}},*

*{ $merge:{into : "netprofit"} }*

*] )*

3- THIRD QUERY

*db.pilots.aggregate( [*

*{ $group : { \_id : "pilotssalary", amount: { $sum: "$contract.salary"}}},*

*{ $merge:{into : "netprofit"} }*

*] )*

4- FOUTH QUERY

*db.netprofit.aggregate([*

*{ "$group": {*

*"\_id": "journeyRev",*

*"NetProfit": {*

*"$sum": "$amount"*

*}*

*} },*

*])*

Final Output:



## **Part 2 – Performance Tuning**

**Use of Indexes and comparison of the “Explain” output**

To understand the impact of use of Indexes in MongoDB on the efficiency, we ran two of our queries with and without indexes: 1) Calculation of opportunity cost 2) The first query from the calculation of Net Profit. Below are our observations:

* **Calculation of Opportunity Cost:**

|  |  |
| --- | --- |
| With Indexes | Without Indexes |
| { explainVersion: '1', | { explainVersion: '1', |
| stages: | stages: |
| [ { '$cursor': | [ { '$cursor': |
| { queryPlanner: | { queryPlanner: |
| { namespace: 'air45.journey\_details', | { namespace: 'air45.journey\_details', |
| indexFilterSet: false, | indexFilterSet: false, |
| parsedQuery: {}, | parsedQuery: {}, |
| queryHash: '348DB3D8', | queryHash: '348DB3D8', |
| planCacheKey: '3ECB9FD5', | planCacheKey: '3ECB9FD5', |
| maxIndexedOrSolutionsReached: false, | maxIndexedOrSolutionsReached: false, |
| maxIndexedAndSolutionsReached: false, | maxIndexedAndSolutionsReached: false, |
| maxScansToExplodeReached: false, | maxScansToExplodeReached: false, |
| winningPlan: | winningPlan: |
| { stage: 'PROJECTION\_DEFAULT', | { stage: 'PROJECTION\_DEFAULT', |
| transformBy: | transformBy: |
| { booking\_type: 1, | { booking\_type: 1, |
| 'f\_id.plane\_id': 1, | 'f\_id.plane\_id': 1, |
| 'f\_id.price\_gbp.business': 1, | 'f\_id.price\_gbp.business': 1, |
| 'f\_id.price\_gbp.economy': 1, | 'f\_id.price\_gbp.economy': 1, |
| flight\_id: 1, | flight\_id: 1, |
| 'p\_id.capacity.business': 1, | 'p\_id.capacity.business': 1, |
| 'p\_id.capacity.economy': 1, | 'p\_id.capacity.economy': 1, |
| passenger\_ids: 1, | passenger\_ids: 1, |
| plane\_id: 1, | plane\_id: 1, |
| \_id: 0 }, | \_id: 0 }, |
| inputStage: { stage: 'COLLSCAN', direction: 'forward' } }, | inputStage: { stage: 'COLLSCAN', direction: 'forward' } }, |
| rejectedPlans: [] }, | rejectedPlans: [] }, |
| executionStats: | executionStats: |
| { executionSuccess: true, | { executionSuccess: true, |
| nReturned: 31, | nReturned: 31, |
| executionTimeMillis: 8, | executionTimeMillis: 5, |
| totalKeysExamined: 0, | totalKeysExamined: 0, |
| totalDocsExamined: 31, | totalDocsExamined: 31, |
| executionStages: | executionStages: |
| { stage: 'PROJECTION\_DEFAULT', | { stage: 'PROJECTION\_DEFAULT', |
| nReturned: 31, | nReturned: 31, |
| executionTimeMillisEstimate: 0, | executionTimeMillisEstimate: 0, |
| works: 33, | works: 33, |
| advanced: 31, | advanced: 31, |
| needTime: 1, | needTime: 1, |
| needYield: 0, | needYield: 0, |
| saveState: 1, | saveState: 1, |
| restoreState: 1, | restoreState: 1, |
| isEOF: 1, | isEOF: 1, |
| transformBy: | transformBy: |
| { booking\_type: 1, | { booking\_type: 1, |
| 'f\_id.plane\_id': 1, | 'f\_id.plane\_id': 1, |
| 'f\_id.price\_gbp.business': 1, | 'f\_id.price\_gbp.business': 1, |
| 'f\_id.price\_gbp.economy': 1, | 'f\_id.price\_gbp.economy': 1, |
| flight\_id: 1, | flight\_id: 1, |
| 'p\_id.capacity.business': 1, | 'p\_id.capacity.business': 1, |
| 'p\_id.capacity.economy': 1, | 'p\_id.capacity.economy': 1, |
| passenger\_ids: 1, | passenger\_ids: 1, |
| plane\_id: 1, | plane\_id: 1, |
| \_id: 0 }, | \_id: 0 }, |
| inputStage: | inputStage: |
| { stage: 'COLLSCAN', | { stage: 'COLLSCAN', |
| nReturned: 31, | nReturned: 31, |
| executionTimeMillisEstimate: 0, | executionTimeMillisEstimate: 0, |
| works: 33, | works: 33, |
| advanced: 31, | advanced: 31, |
| needTime: 1, | needTime: 1, |
| needYield: 0, | needYield: 0, |
| saveState: 1, | saveState: 1, |
| restoreState: 1, | restoreState: 1, |
| isEOF: 1, | isEOF: 1, |
| direction: 'forward', | direction: 'forward', |
| docsExamined: 31 } }, | docsExamined: 31 } }, |
| allPlansExecution: [] } }, | allPlansExecution: [] } }, |
| nReturned: 31, | nReturned: 31, |
| executionTimeMillisEstimate: 0 }, | executionTimeMillisEstimate: 0 }, |
| { '$lookup': | { '$lookup': |
| { from: 'flights', | { from: 'flights', |
| as: 'f\_id', | as: 'f\_id', |
| localField: 'flight\_id', | localField: 'flight\_id', |
| foreignField: 'flight\_id', | foreignField: 'flight\_id', |
| unwinding: { preserveNullAndEmptyArrays: false } }, | unwinding: { preserveNullAndEmptyArrays: false } }, |
| totalDocsExamined: 34, | totalDocsExamined: 434, |
| totalKeysExamined: 37, | totalKeysExamined: 0, |
| collectionScans: 0, | collectionScans: 62, |
| indexesUsed: [ 'flight\_id\_1' ], | indexesUsed: [], |
| nReturned: 34, | nReturned: 34, |
| executionTimeMillisEstimate: 6 }, | executionTimeMillisEstimate: 3 }, |
| { '$lookup': | { '$lookup': |
| { from: 'planes', | { from: 'planes', |
| as: 'p\_id', | as: 'p\_id', |
| localField: 'f\_id.plane\_id', | localField: 'f\_id.plane\_id', |
| foreignField: 'plane\_id', | foreignField: 'plane\_id', |
| unwinding: { preserveNullAndEmptyArrays: false } }, | unwinding: { preserveNullAndEmptyArrays: false } }, |
| totalDocsExamined: 34, | totalDocsExamined: 306, |
| totalKeysExamined: 34, | totalKeysExamined: 0, |
| collectionScans: 0, | collectionScans: 68, |
| indexesUsed: [ 'plane\_id\_1' ], | indexesUsed: [], |
| nReturned: 34, | nReturned: 34, |
| executionTimeMillisEstimate: 7 }, | executionTimeMillisEstimate: 5 }, |
| { '$project': | { '$project': |
| { flight\_id: true, | { flight\_id: true, |
| plane\_id: true, | booking\_type: true, |
| booking\_type: true, | plane\_id: true, |
| oppcost: | oppcost: |
| { '$multiply': | { '$multiply': |
| [ { '$subtract': | [ { '$subtract': |
| [ { '$cond': | [ { '$cond': |
| [ { '$regexMatch': { input: { '$const': 'economy' }, regex: '$booking\_type' } }, | [ { '$regexMatch': { input: { '$const': 'economy' }, regex: '$booking\_type' } }, |
| '$p\_id.capacity.economy', | '$p\_id.capacity.economy', |
| '$p\_id.capacity.business' ] }, | '$p\_id.capacity.business' ] }, |
| { '$size': [ '$passenger\_ids' ] } ] }, | { '$size': [ '$passenger\_ids' ] } ] }, |
| { '$cond': | { '$cond': |
| [ { '$regexMatch': { input: { '$const': 'economy' }, regex: '$booking\_type' } }, | [ { '$regexMatch': { input: { '$const': 'economy' }, regex: '$booking\_type' } }, |
| '$f\_id.price\_gbp.economy', | '$f\_id.price\_gbp.economy', |
| '$f\_id.price\_gbp.business' ] } ] }, | '$f\_id.price\_gbp.business' ] } ] }, |
| \_id: false }, | \_id: false }, |
| nReturned: 34, | nReturned: 34, |
| executionTimeMillisEstimate: 7 }, | executionTimeMillisEstimate: 5 }, |
| { '$group': | { '$group': |
| { \_id: { '$const': null }, | { \_id: { '$const': null }, |
| totalOppCost: { '$sum': '$oppcost' } }, | totalOppCost: { '$sum': '$oppcost' } }, |
| maxAccumulatorMemoryUsageBytes: { totalOppCost: 80 }, | maxAccumulatorMemoryUsageBytes: { totalOppCost: 80 }, |
| totalOutputDataSizeBytes: 229, | totalOutputDataSizeBytes: 229, |
| usedDisk: false, | usedDisk: false, |
| nReturned: 1, | nReturned: 1, |
| executionTimeMillisEstimate: 7 } ], | executionTimeMillisEstimate: 5 } ], |
| serverInfo: | serverInfo: |
| { host: 'DESKTOP-3D0QR9V', | { host: 'DESKTOP-3D0QR9V', |
| port: 27017, | port: 27017, |
| version: '5.0.5', | version: '5.0.5', |
| gitVersion: 'd65fd89df3fc039b5c55933c0f71d647a54510ae' }, | gitVersion: 'd65fd89df3fc039b5c55933c0f71d647a54510ae' }, |
| serverParameters: | serverParameters: |
| { internalQueryFacetBufferSizeBytes: 104857600, | { internalQueryFacetBufferSizeBytes: 104857600, |
| internalQueryFacetMaxOutputDocSizeBytes: 104857600, | internalQueryFacetMaxOutputDocSizeBytes: 104857600, |
| internalLookupStageIntermediateDocumentMaxSizeBytes: 104857600, | internalLookupStageIntermediateDocumentMaxSizeBytes: 104857600, |
| internalDocumentSourceGroupMaxMemoryBytes: 104857600, | internalDocumentSourceGroupMaxMemoryBytes: 104857600, |
| internalQueryMaxBlockingSortMemoryUsageBytes: 104857600, | internalQueryMaxBlockingSortMemoryUsageBytes: 104857600, |
| internalQueryProhibitBlockingMergeOnMongoS: 0, | internalQueryProhibitBlockingMergeOnMongoS: 0, |
| internalQueryMaxAddToSetBytes: 104857600, | internalQueryMaxAddToSetBytes: 104857600, |
| internalDocumentSourceSetWindowFieldsMaxMemoryBytes: 104857600 }, | internalDocumentSourceSetWindowFieldsMaxMemoryBytes: 104857600 }, |
| command: | command: |
| { aggregate: 'journey\_details', | { aggregate: 'journey\_details', |
| pipeline: | pipeline: |
| [ { '$lookup': | [ { '$lookup': |
| { from: 'flights', | { from: 'flights', |
| localField: 'flight\_id', | localField: 'flight\_id', |
| foreignField: 'flight\_id', | foreignField: 'flight\_id', |
| as: 'f\_id' } }, | as: 'f\_id' } }, |
| { '$unwind': '$f\_id' }, | { '$unwind': '$f\_id' }, |
| { '$lookup': | { '$lookup': |
| { from: 'planes', | { from: 'planes', |
| localField: 'f\_id.plane\_id', | localField: 'f\_id.plane\_id', |
| foreignField: 'plane\_id', | foreignField: 'plane\_id', |
| as: 'p\_id' } }, | as: 'p\_id' } }, |
| { '$unwind': '$p\_id' }, | { '$unwind': '$p\_id' }, |
| { '$project': | { '$project': |
| { \_id: 0, | { \_id: 0, |
| flight\_id: 1, | flight\_id: 1, |
| booking\_type: 1, | booking\_type: 1, |
| plane\_id: 1, | plane\_id: 1, |
| oppcost: | oppcost: |
| { '$multiply': | { '$multiply': |
| [ { '$subtract': | [ { '$subtract': |
| [ { '$cond': | [ { '$cond': |
| [ { '$regexMatch': { input: 'economy', regex: '$booking\_type' } }, | [ { '$regexMatch': { input: 'economy', regex: '$booking\_type' } }, |
| '$p\_id.capacity.economy', | '$p\_id.capacity.economy', |
| '$p\_id.capacity.business' ] }, | '$p\_id.capacity.business' ] }, |
| { '$size': '$passenger\_ids' } ] }, | { '$size': '$passenger\_ids' } ] }, |
| { '$cond': | { '$cond': |
| [ { '$regexMatch': { input: 'economy', regex: '$booking\_type' } }, | [ { '$regexMatch': { input: 'economy', regex: '$booking\_type' } }, |
| '$f\_id.price\_gbp.economy', | '$f\_id.price\_gbp.economy', |
| '$f\_id.price\_gbp.business' ] } ] } } }, | '$f\_id.price\_gbp.business' ] } ] } } }, |
| { '$group': { \_id: null, totalOppCost: { '$sum': '$oppcost' } } } ], | { '$group': { \_id: null, totalOppCost: { '$sum': '$oppcost' } } } ], |
| cursor: {}, | cursor: {}, |
| '$db': 'air45' }, | '$db': 'air45' }, |
| ok: 1 | ok: 1 |

**Observation**: The key differences were found in *totalDocsExamined, collectionScans* and *executionTimeMillis* in mainly the lookup stages. The reason we see this in the lookup stages is because without indexes, the code has to scan more document in comparison to when we have indexes in the collections. Hence, in the results without indexes, the number of docs and collections examined are at least 10X more than when we have indexes. However, when we talk about the execution times, the code when not using indexes seemingly runs ~30% faster at all stages. However, this is seen because our database is relatively simpler and smaller in comparison to what we would see for any actual organisation. The sheer increase in the number of documents and collections that need to be scanned is bound to increase the execution times as the query and the database become more complex.

* **The first query from the calculation of Net Profit:**

|  |  |
| --- | --- |
| With Indexes | Without Indexes |
| { explainVersion: '1', | { explainVersion: '1', |
| stages: | stages: |
| [ { '$cursor': | [ { '$cursor': |
| { queryPlanner: | { queryPlanner: |
| { namespace: 'air45.journey\_details', | { namespace: 'air45.journey\_details', |
| indexFilterSet: false, | indexFilterSet: false, |
| parsedQuery: { booking\_type: { '$eq': 'economy' } }, | parsedQuery: { booking\_type: { '$eq': 'economy' } }, |
| queryHash: 'C23B2348', | queryHash: 'C23B2348', |
| planCacheKey: '7F52274E', | planCacheKey: '7F52274E', |
| maxIndexedOrSolutionsReached: false, | maxIndexedOrSolutionsReached: false, |
| maxIndexedAndSolutionsReached: false, | maxIndexedAndSolutionsReached: false, |
| maxScansToExplodeReached: false, | maxScansToExplodeReached: false, |
| winningPlan: | winningPlan: |
| { stage: 'PROJECTION\_DEFAULT', | { stage: 'PROJECTION\_DEFAULT', |
| transformBy: | transformBy: |
| { \_id: 1, | { \_id: 1, |
| booking\_type: 1, | booking\_type: 1, |
| 'destination\_airport\_id.costOfUsePerHour\_GBP': 1, | 'destination\_airport\_id.costOfUsePerHour\_GBP': 1, |
| 'destination\_airport\_id.refuelCost\_GBP': 1, | 'destination\_airport\_id.refuelCost\_GBP': 1, |
| 'f\_id.plane\_id': 1, | 'f\_id.plane\_id': 1, |
| 'f\_id.price\_gbp.business': 1, | 'f\_id.price\_gbp.business': 1, |
| 'f\_id.price\_gbp.economy': 1, | 'f\_id.price\_gbp.economy': 1, |
| 'f\_id.route.destination\_airport\_id': 1, | 'f\_id.route.destination\_airport\_id': 1, |
| 'f\_id.route.start\_airport\_id': 1, | 'f\_id.route.start\_airport\_id': 1, |
| flight\_id: 1, | flight\_id: 1, |
| passenger\_ids: 1, | passenger\_ids: 1, |
| 'start\_airport\_id.costOfUsePerHour\_GBP': 1, | 'start\_airport\_id.costOfUsePerHour\_GBP': 1, |
| 'start\_airport\_id.refuelCost\_GBP': 1 }, | 'start\_airport\_id.refuelCost\_GBP': 1 }, |
| inputStage: | inputStage: |
| { stage: 'COLLSCAN', | { stage: 'COLLSCAN', |
| filter: { booking\_type: { '$eq': 'economy' } }, | filter: { booking\_type: { '$eq': 'economy' } }, |
| direction: 'forward' } }, | direction: 'forward' } }, |
| rejectedPlans: [] }, | rejectedPlans: [] }, |
| executionStats: | executionStats: |
| { executionSuccess: true, | { executionSuccess: true, |
| nReturned: 15, | nReturned: 15, |
| executionTimeMillis: 8, | executionTimeMillis: 5, |
| totalKeysExamined: 0, | totalKeysExamined: 0, |
| totalDocsExamined: 31, | totalDocsExamined: 31, |
| executionStages: | executionStages: |
| { stage: 'PROJECTION\_DEFAULT', | { stage: 'PROJECTION\_DEFAULT', |
| nReturned: 15, | nReturned: 15, |
| executionTimeMillisEstimate: 0, | executionTimeMillisEstimate: 0, |
| works: 33, | works: 33, |
| advanced: 15, | advanced: 15, |
| needTime: 17, | needTime: 17, |
| needYield: 0, | needYield: 0, |
| saveState: 1, | saveState: 1, |
| restoreState: 1, | restoreState: 1, |
| isEOF: 1, | isEOF: 1, |
| transformBy: | transformBy: |
| { \_id: 1, | { \_id: 1, |
| booking\_type: 1, | booking\_type: 1, |
| 'destination\_airport\_id.costOfUsePerHour\_GBP': 1, | 'destination\_airport\_id.costOfUsePerHour\_GBP': 1, |
| 'destination\_airport\_id.refuelCost\_GBP': 1, | 'destination\_airport\_id.refuelCost\_GBP': 1, |
| 'f\_id.plane\_id': 1, | 'f\_id.plane\_id': 1, |
| 'f\_id.price\_gbp.business': 1, | 'f\_id.price\_gbp.business': 1, |
| 'f\_id.price\_gbp.economy': 1, | 'f\_id.price\_gbp.economy': 1, |
| 'f\_id.route.destination\_airport\_id': 1, | 'f\_id.route.destination\_airport\_id': 1, |
| 'f\_id.route.start\_airport\_id': 1, | 'f\_id.route.start\_airport\_id': 1, |
| flight\_id: 1, | flight\_id: 1, |
| passenger\_ids: 1, | passenger\_ids: 1, |
| 'start\_airport\_id.costOfUsePerHour\_GBP': 1, | 'start\_airport\_id.costOfUsePerHour\_GBP': 1, |
| 'start\_airport\_id.refuelCost\_GBP': 1 }, | 'start\_airport\_id.refuelCost\_GBP': 1 }, |
| inputStage: | inputStage: |
| { stage: 'COLLSCAN', | { stage: 'COLLSCAN', |
| filter: { booking\_type: { '$eq': 'economy' } }, | filter: { booking\_type: { '$eq': 'economy' } }, |
| nReturned: 15, | nReturned: 15, |
| executionTimeMillisEstimate: 0, | executionTimeMillisEstimate: 0, |
| works: 33, | works: 33, |
| advanced: 15, | advanced: 15, |
| needTime: 17, | needTime: 17, |
| needYield: 0, | needYield: 0, |
| saveState: 1, | saveState: 1, |
| restoreState: 1, | restoreState: 1, |
| isEOF: 1, | isEOF: 1, |
| direction: 'forward', | direction: 'forward', |
| docsExamined: 31 } }, | docsExamined: 31 } }, |
| allPlansExecution: [] } }, | allPlansExecution: [] } }, |
| nReturned: 15, | nReturned: 15, |
| executionTimeMillisEstimate: 0 }, | executionTimeMillisEstimate: 0 }, |
| { '$lookup': | { '$lookup': |
| { from: 'flights', | { from: 'flights', |
| as: 'f\_id', | as: 'f\_id', |
| localField: 'flight\_id', | localField: 'flight\_id', |
| foreignField: 'flight\_id' }, | foreignField: 'flight\_id' }, |
| totalDocsExamined: 16, | totalDocsExamined: 210, |
| totalKeysExamined: 17, | totalKeysExamined: 0, |
| collectionScans: 0, | collectionScans: 30, |
| indexesUsed: [ 'flight\_id\_1' ], | indexesUsed: [], |
| nReturned: 15, | nReturned: 15, |
| executionTimeMillisEstimate: 7 }, | executionTimeMillisEstimate: 0 }, |
| { '$lookup': | { '$lookup': |
| { from: 'planes', | { from: 'planes', |
| as: 'p\_id', | as: 'p\_id', |
| localField: 'f\_id.plane\_id', | localField: 'f\_id.plane\_id', |
| foreignField: 'plane\_id' }, | foreignField: 'plane\_id' }, |
| totalDocsExamined: 15, | totalDocsExamined: 135, |
| totalKeysExamined: 15, | totalKeysExamined: 0, |
| collectionScans: 0, | collectionScans: 30, |
| indexesUsed: [ 'plane\_id\_1' ], | indexesUsed: [], |
| nReturned: 15, | nReturned: 15, |
| executionTimeMillisEstimate: 7 }, | executionTimeMillisEstimate: 4 }, |
| { '$lookup': | { '$lookup': |
| { from: 'airports', | { from: 'airports', |
| as: 'start\_airport\_id', | as: 'start\_airport\_id', |
| localField: 'f\_id.route.start\_airport\_id', | localField: 'f\_id.route.start\_airport\_id', |
| foreignField: 'airport\_id' }, | foreignField: 'airport\_id' }, |
| totalDocsExamined: 16, | totalDocsExamined: 195, |
| totalKeysExamined: 18, | totalKeysExamined: 0, |
| collectionScans: 0, | collectionScans: 30, |
| indexesUsed: [ 'airport\_id\_1' ], | indexesUsed: [], |
| nReturned: 15, | nReturned: 15, |
| executionTimeMillisEstimate: 7 }, | executionTimeMillisEstimate: 5 }, |
| { '$lookup': | { '$lookup': |
| { from: 'airports', | { from: 'airports', |
| as: 'destination\_airport\_id', | as: 'destination\_airport\_id', |
| localField: 'f\_id.route.destination\_airport\_id', | localField: 'f\_id.route.destination\_airport\_id', |
| foreignField: 'airport\_id' }, | foreignField: 'airport\_id' }, |
| totalDocsExamined: 16, | totalDocsExamined: 195, |
| totalKeysExamined: 18, | totalKeysExamined: 0, |
| collectionScans: 0, | collectionScans: 30, |
| indexesUsed: [ 'airport\_id\_1' ], | indexesUsed: [], |
| nReturned: 15, | nReturned: 15, |
| executionTimeMillisEstimate: 8 }, | executionTimeMillisEstimate: 5 }, |
| { '$project': | { '$project': |
| { \_id: true, | { \_id: true, |
| profit: | profit: |
| { '$subtract': | { '$subtract': |
| [ { '$multiply': | [ { '$multiply': |
| [ { '$sum': | [ { '$sum': |
| [ { '$cond': | [ { '$cond': |
| [ { '$regexMatch': { input: { '$const': 'economy' }, regex: '$booking\_type' } }, | [ { '$regexMatch': { input: { '$const': 'economy' }, regex: '$booking\_type' } }, |
| '$f\_id.price\_gbp.economy', | '$f\_id.price\_gbp.economy', |
| '$f\_id.price\_gbp.business' ] } ] }, | '$f\_id.price\_gbp.business' ] } ] }, |
| { '$size': [ '$passenger\_ids' ] } ] }, | { '$size': [ '$passenger\_ids' ] } ] }, |
| { '$sum': | { '$sum': |
| [ { '$sum': [ '$destination\_airport\_id.costOfUsePerHour\_GBP' ] }, | [ { '$sum': [ '$destination\_airport\_id.costOfUsePerHour\_GBP' ] }, |
| { '$sum': [ '$start\_airport\_id.costOfUsePerHour\_GBP' ] }, | { '$sum': [ '$start\_airport\_id.costOfUsePerHour\_GBP' ] }, |
| { '$sum': [ '$start\_airport\_id.refuelCost\_GBP' ] }, | { '$sum': [ '$start\_airport\_id.refuelCost\_GBP' ] }, |
| { '$sum': [ '$destination\_airport\_id.refuelCost\_GBP' ] } ] } ] } }, | { '$sum': [ '$destination\_airport\_id.refuelCost\_GBP' ] } ] } ] } }, |
| nReturned: 15, | nReturned: 15, |
| executionTimeMillisEstimate: 8 }, | executionTimeMillisEstimate: 5 }, |
| { '$group': | { '$group': |
| { \_id: { '$const': 'journeyRev' }, | { \_id: { '$const': 'journeyRev' }, |
| amount: { '$sum': '$profit' } }, | amount: { '$sum': '$profit' } }, |
| maxAccumulatorMemoryUsageBytes: { amount: 80 }, | maxAccumulatorMemoryUsageBytes: { amount: 80 }, |
| totalOutputDataSizeBytes: 229, | totalOutputDataSizeBytes: 229, |
| usedDisk: false, | usedDisk: false, |
| nReturned: 1, | nReturned: 1, |
| executionTimeMillisEstimate: 8 }, | executionTimeMillisEstimate: 5 }, |
| { '$merge': | { '$merge': |
| { into: { db: 'air45', coll: 'netprofit' }, | { into: { db: 'air45', coll: 'netprofit' }, |
| on: '\_id', | on: '\_id', |
| whenMatched: 'merge', | whenMatched: 'merge', |
| whenNotMatched: 'insert' }, | whenNotMatched: 'insert' }, |
| nReturned: 0, | nReturned: 0, |
| executionTimeMillisEstimate: 8 } ], | executionTimeMillisEstimate: 5 } ], |
| serverInfo: | serverInfo: |
| { host: 'DESKTOP-3D0QR9V', | { host: 'DESKTOP-3D0QR9V', |
| port: 27017, | port: 27017, |
| version: '5.0.5', | version: '5.0.5', |
| gitVersion: 'd65fd89df3fc039b5c55933c0f71d647a54510ae' }, | gitVersion: 'd65fd89df3fc039b5c55933c0f71d647a54510ae' }, |
| serverParameters: | serverParameters: |
| { internalQueryFacetBufferSizeBytes: 104857600, | { internalQueryFacetBufferSizeBytes: 104857600, |
| internalQueryFacetMaxOutputDocSizeBytes: 104857600, | internalQueryFacetMaxOutputDocSizeBytes: 104857600, |
| internalLookupStageIntermediateDocumentMaxSizeBytes: 104857600, | internalLookupStageIntermediateDocumentMaxSizeBytes: 104857600, |
| internalDocumentSourceGroupMaxMemoryBytes: 104857600, | internalDocumentSourceGroupMaxMemoryBytes: 104857600, |
| internalQueryMaxBlockingSortMemoryUsageBytes: 104857600, | internalQueryMaxBlockingSortMemoryUsageBytes: 104857600, |
| internalQueryProhibitBlockingMergeOnMongoS: 0, | internalQueryProhibitBlockingMergeOnMongoS: 0, |
| internalQueryMaxAddToSetBytes: 104857600, | internalQueryMaxAddToSetBytes: 104857600, |
| internalDocumentSourceSetWindowFieldsMaxMemoryBytes: 104857600 }, | internalDocumentSourceSetWindowFieldsMaxMemoryBytes: 104857600 }, |
| command: | command: |
| { aggregate: 'journey\_details', | { aggregate: 'journey\_details', |
| pipeline: | pipeline: |
| [ { '$match': { booking\_type: 'economy' } }, | [ { '$match': { booking\_type: 'economy' } }, |
| { '$lookup': | { '$lookup': |
| { from: 'flights', | { from: 'flights', |
| localField: 'flight\_id', | localField: 'flight\_id', |
| foreignField: 'flight\_id', | foreignField: 'flight\_id', |
| as: 'f\_id' } }, | as: 'f\_id' } }, |
| { '$lookup': | { '$lookup': |
| { from: 'planes', | { from: 'planes', |
| localField: 'f\_id.plane\_id', | localField: 'f\_id.plane\_id', |
| foreignField: 'plane\_id', | foreignField: 'plane\_id', |
| as: 'p\_id' } }, | as: 'p\_id' } }, |
| { '$lookup': | { '$lookup': |
| { from: 'airports', | { from: 'airports', |
| localField: 'f\_id.route.start\_airport\_id', | localField: 'f\_id.route.start\_airport\_id', |
| foreignField: 'airport\_id', | foreignField: 'airport\_id', |
| as: 'start\_airport\_id' } }, | as: 'start\_airport\_id' } }, |
| { '$lookup': | { '$lookup': |
| { from: 'airports', | { from: 'airports', |
| localField: 'f\_id.route.destination\_airport\_id', | localField: 'f\_id.route.destination\_airport\_id', |
| foreignField: 'airport\_id', | foreignField: 'airport\_id', |
| as: 'destination\_airport\_id' } }, | as: 'destination\_airport\_id' } }, |
| { '$project': | { '$project': |
| { profit: | { profit: |
| { '$subtract': | { '$subtract': |
| [ { '$multiply': | [ { '$multiply': |
| [ { '$sum': | [ { '$sum': |
| { '$cond': | { '$cond': |
| [ { '$regexMatch': { input: 'economy', regex: '$booking\_type' } }, | [ { '$regexMatch': { input: 'economy', regex: '$booking\_type' } }, |
| '$f\_id.price\_gbp.economy', | '$f\_id.price\_gbp.economy', |
| '$f\_id.price\_gbp.business' ] } }, | '$f\_id.price\_gbp.business' ] } }, |
| { '$size': '$passenger\_ids' } ] }, | { '$size': '$passenger\_ids' } ] }, |
| { '$sum': | { '$sum': |
| [ { '$sum': '$destination\_airport\_id.costOfUsePerHour\_GBP' }, | [ { '$sum': '$destination\_airport\_id.costOfUsePerHour\_GBP' }, |
| { '$sum': '$start\_airport\_id.costOfUsePerHour\_GBP' }, | { '$sum': '$start\_airport\_id.costOfUsePerHour\_GBP' }, |
| { '$sum': '$start\_airport\_id.refuelCost\_GBP' }, | { '$sum': '$start\_airport\_id.refuelCost\_GBP' }, |
| { '$sum': '$destination\_airport\_id.refuelCost\_GBP' } ] } ] } } }, | { '$sum': '$destination\_airport\_id.refuelCost\_GBP' } ] } ] } } }, |
| { '$group': { \_id: 'journeyRev', amount: { '$sum': '$profit' } } }, | { '$group': { \_id: 'journeyRev', amount: { '$sum': '$profit' } } }, |
| { '$merge': { into: 'netprofit' } } ], | { '$merge': { into: 'netprofit' } } ], |
| cursor: {}, | cursor: {}, |
| '$db': 'air45' }, | '$db': 'air45' }, |
| ok: 1 } | ok: 1 } |

**Observation**: Upon seeing the results for this query as well, the pattern clearly remains the same, with ~10X increased document and collection scans when not using indexes.

**Use of the MongoDB profiler for examining performance**

We will be using the MongDB’s built-in profiler tool to obtain query performances, this will be done by first changing the default profiler level 0 to level 1, and then performing queries on the system.profile collection. We will first analyze profiler outputs for 3 queries individually, then we will perform an overall comparison of 6 queries after running them consecutively. The output of which will be found in the profiler.js file

**Query 3: List of planes under “repair” status profiler output (Find):**

{ op: 'query',

ns: 'airlines.planes',

keysExamined: 0,

docsExamined: 9,

cursorExhausted: true,

numYield: 0,

nreturned: 1,

queryHash: 'A288E2FC',

planCacheKey: 'BE89A40E',

locks:

{ ReplicationStateTransition: { acquireCount: { w: 1 } },

Global: { acquireCount: { r: 2 } },

Mutex: { acquireCount: { r: 1 } } },

flowControl: {},

storage: {},

responseLength: 166,

protocol: 'op\_msg',

millis: 0,

planSummary: 'COLLSCAN',

execStats:

{ stage: 'PROJECTION\_SIMPLE',

nReturned: 1,

executionTimeMillisEstimate: 0,

works: 11,

advanced: 1,

needTime: 9,

needYield: 0,

saveState: 0,

restoreState: 0,

isEOF: 1,

transformBy: {},

inputStage:

{ stage: 'COLLSCAN',

filter: { status: { '$eq': 'repair' } },

nReturned: 1,

executionTimeMillisEstimate: 0,

works: 11,

advanced: 1,

needTime: 9,

needYield: 0,

saveState: 0,

restoreState: 0,

isEOF: 1,

direction: 'forward',

docsExamined: 9 } },

ts: 2021-12-22T22:54:22.658Z,

client: '127.0.0.1',

appName: 'MongoDB Compass',

allUsers: [],

user: '' }

This query’s profiler shows that it is of type “query” since it’s a find query. The output shows that it scans through all the 9 documents in the planes collection, within a 0-millisecond timeframe. Even though it took 0 milliseconds to execute, the number of scanned documents could be problematic given a larger number of documents. It also shows no use of index scanning because keysExamined is zero, this is due to the query’s simplicity. n\_Returned shows that this query returns only one document.

**Query 6: Total ticket revenue profiler output (Aggregation):**

{ op: 'command',

ns: 'airlines.flights',

keysExamined: 1,

docsExamined: 15,

hasSortStage: true,

cursorExhausted: true,

numYield: 0,

nreturned: 1,

queryHash: 'E54049AB',

planCacheKey: '5C157ED7',

locks:

{ ReplicationStateTransition: { acquireCount: { w: 1 } },

Global: { acquireCount: { r: 5 } },

Mutex: { acquireCount: { r: 4 } } },

flowControl: {},

storage: {},

responseLength: 171,

protocol: 'op\_msg',

millis: 1,

planSummary: 'COLLSCAN',

ts: 2021-12-22T23:26:20.451Z,

client: '127.0.0.1',

appName: 'MongoDB Compass',

allUsers: [],

user: '' }

Since this query is an aggregate query, it is shown on the “op” field as type “command”. Its profiling shows that it took 1 millisecond to execute and examined 15 documents which include the use of 1 index (keyExamined: 1). Although the use of an index could have been avoided by using embedded documents instead, we preferred linking by index due to the potentially large, linked data (getting seating capacity from the planes collection to calculate revenue from tickets). Therefore, due to the indexing of several documents taking no more than 1 millisecond, we consider this query to be high-performing.

**Query 8: Pilot with most hours of flying profiler output (Aggregation):**

{ op: 'command',

ns: 'airlines.flights',

keysExamined: 0,

docsExamined: 14,

hasSortStage: true,

cursorExhausted: true,

numYield: 0,

nreturned: 1,

queryHash: 'E54049AB',

planCacheKey: '5C157ED7',

locks:

{ ReplicationStateTransition: { acquireCount: { w: 1 } },

Global: { acquireCount: { r: 5 } },

Mutex: { acquireCount: { r: 4 } } },

flowControl: {},

storage: {},

responseLength: 169,

protocol: 'op\_msg',

millis: 1,

planSummary: 'COLLSCAN',

ts: 2021-12-23T01:22:35.503Z,

client: '127.0.0.1',

appName: 'MongoDB Compass',

allUsers: [],

user: '' }

This is another aggregation query, even though it includes a lookup stage, its execution time is 1 millisecond, and no keys are examined; therefore it is an efficient performance.

Now let us have a bird eye view on the overall performance of all our queries by reviewing the profiler output after running all the queries in a sequential manner. The complete profiler output is available in profiler\_output.js. We will be reviewing the profiler output by performing a set of queries on the system.profile collection as follows:

1. **Reviewing all the queries which took more than 10 milliseconds to execute**:

*db.system.profile.find({millis:{$gt:10}}).pretty()*

We ran the above query which returns the details of the queries which took more than 10 milliseconds to execute. The profiler output is given by:

Text

Description automatically generated

Text

Description automatically generated

On reviewing the above output, we can see that the query took 12 milliseconds to complete its execution, while scanning through 71 keys and 99 documents, which returned a single output. Since the number of *keysExamined* is much greater than *nreturned*, the database is scanning many index keys to find the result documents. We can improve the performance of our model by adjusting the indexing or creating a new index to improve the query performance. On reviewing the *planSummary*, we can see that COLLSCAN is the operation performed as a part of query execution. It is the abbreviation for collection scan (In case if we had an index set on the selected elements that supports the query, the operation performed will be ITSCAN, for scanning index keys).

1. **Find the maximum and average time taken by each type of operations during aggregation:**

*db.system.profile.aggregate(*

*{ $group : { \_id :"$op", count:{$sum:1},*

*"max\_time":{$max:"$millis"},*

*"avg\_time":{$avg:"$millis"}}}).pretty()*

Output :

A screenshot of a computer

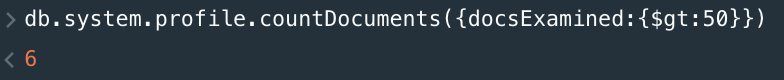
Description automatically generated with medium confidence

The above output shows that the query and update operations had much faster execution and the command operations performed by our queries took a maximum time of 12 milliseconds, with an average on 3.0625 milliseconds per operation.

1. **Number of queries performing larger scans on the database documents**

*db.system.profile.countDocuments({docsExamined:{$gt:50}})*

Output:



On reviewing the output, we can infer that 6 of our queries are performing larger scans (scanning more than 50 documents) on the database when executed.

1. **Find the maximum and average time taken by queries in each database using aggregation:**

*db.system.profile.aggregate(*

*{ $group : { \_id :"$ns", count:{$sum:1},*

*"max\_time":{$max:"$millis"},*

*"avg\_time":{$avg:"$millis"} }}).pretty()*

Output:

Text

Description automatically generated

The maximum and average time taken each query in *netprofit, planes, employees* and *pilots* collection is negligible. On reviewing the output, the maximum time was spent on the journey\_details collection (12 milliseconds), making it an average on 8.4 milliseconds per scan.

1. **Number of queries targeted on each collection**

db.system.profile.aggregate([

*{*

*$group: {\_id: "$ns", targetcount: {$sum: 1}}*

*},*

*{*

*$sort: {targetcount: -1}*

*} ]);*

Output:

Text

Description automatically generated

On reviewing the above output, we can summarize that the planes collection is affected by 1 query, *employees* collection by 2 queries, *pilots* collection queried by 3 queries, *journey\_details* and *flights* collection being queried by 5 queries. As an inference we can say that the most queried collections are the flights and *journey\_details* collections.

## **Appendix**

The following is the list of all the codes that we used to create our collections. We have also added the export for all our collections in the main folder. Kindly use the exports or the below codes to re-create our database.

**Collection “airports”**

Validation**:**

*db.createCollection( "airports", {*

*validator: { $jsonSchema: {*

*bsonType: "object",*

*required: [ "airport\_id", "name", "location", "refuelCost\_GBP", "costOfUsePerHour\_GBP"],*

*properties: {*

*airport\_id: {*

*bsonType: "string",*

*description: "A unique airport code, must be string"*

*},*

*name: {*

*bsonType: "string",*

*description: "Airport's name, must be string"*

*},*

*location: {*

*bsonType:"object",*

*required: ["country", "city"],*

*properties:*

*{*

*country: {bsonType: "string", description: "Country must be a string and is required"},*

*city: {bsonType: "string", description: "City must be a string and is required"},*

*}},*

*refuelCost\_GBP: {bsonType: "double", description: "RefuelCost must be a string and is required"}*

*}*

*} }*

*} )*

Index Creation:

*db.airports.createIndex( { "airport\_id": 1 }, { unique: true } )*

Inserting documents:

*db.airports.insertMany ([*

*{*

*airport\_id: "MAN",*

*name: "Manchester Airport",*

*location: {*

*country: "United Kingdom",*

*city: "Manchester"*

*},*

*refuelCost\_GBP: 26564.01,*

*costOfUsePerHour\_GBP: 0.0*

*},*

*{*

*airport\_id: "LGW",*

*name: "Gatwick Airport",*

*location: {*

*country: "United Kingdom",*

*city: "Gatwick"*

*},*

*refuelCost\_GBP: 25864.35,*

*costOfUsePerHour\_GBP: 0.0*

*},*

*{*

*airport\_id: "LHR",*

*name: "Heathrow Airport",*

*location: {*

*country: "United Kingdom",*

*city: "London"*

*},*

*refuelCost\_GBP: 24514.07,*

*costOfUsePerHour\_GBP: 0.0*

*},{*

*airport\_id: "CDG",*

*name: "Paris Charles de Gaulle Airport",*

*location: {*

*country: "France",*

*city: "Paris"*

*},*

*refuelCost\_GBP: 26851.07,*

*costOfUsePerHour\_GBP: 80.6*

*},*

*{*

*airport\_id: "JFK",*

*name: "John F. Kennedy International Airport",*

*location: {*

*country: "United States",*

*city: "New York"*

*},*

*refuelCost\_GBP: 25854.21,*

*costOfUsePerHour\_GBP: 80.6*

*},*

*{*

*airport\_id: "KUL",*

*name: "Kuala Lumpur International Airport",*

*location: {*

*country: "Malaysia",*

*city: "Kuala Lumpur"*

*},*

*refuelCost\_GBP: 24594.5,*

*costOfUsePerHour\_GBP: 99.3*

*},*

*{*

*airport\_id: "SCQ",*

*name: "Santiago–Rosalía de Castro Airport",*

*location: {*

*country: "Spain",*

*city: "Santiago de Compostela"*

*},*

*refuelCost\_GBP: 26154.41,*

*costOfUsePerHour\_GBP: 68.9*

*},*

*{*

*airport\_id: "EZE",*

*name: "Ministro Pistarini International Airport",*

*location: {*

*country: "Argentina",*

*city: "Ezeiza"*

*},*

*refuelCost\_GBP: 24591.81,*

*costOfUsePerHour\_GBP: 76.5*

*},*

*{*

*airport\_id: "BLR",*

*name: "Bangalore International Airport",*

*location: {*

*country: "India",*

*city: "Bangalore"*

*},*

*refuelCost\_GBP: 25491.55,*

*costOfUsePerHour\_GBP: 69.7*

*},*

*{*

*airport\_id: "CAI",*

*name: "Cairo International Airport",*

*location: {*

*country: "Egypt",*

*city: "Cairo"*

*},*

*refuelCost\_GBP: 25613.4,*

*costOfUsePerHour\_GBP: 60.1*

*},*

*{*

*airport\_id: "CPT",*

*name: "Cape Town International Airport",*

*location: {*

*country: "South Africa",*

*city: "Cape Town"*

*},*

*refuelCost\_GBP: 26851.4,*

*costOfUsePerHour\_GBP: 90.6*

*},*

*{*

*airport\_id: "JNB",*

*name: "O.R Tambo International Airport",*

*location: {*

*country: "South Africa",*

*city: "Johannesburg"*

*},*

*refuelCost\_GBP: 24931.4,*

*costOfUsePerHour\_GBP: 46.7*

*},*

*{*

*airport\_id: "LAX",*

*name: "Los Angeles International Airport",*

*location: {*

*country: "United States",*

*city: "Los Angeles"*

*},*

*refuelCost\_GBP: 24692.4,*

*costOfUsePerHour\_GBP: 80.6*

*},*

*])*

**Collection “employees”**

Validation**:**

*db.createCollection("employees", {*

*validator: {*

*$jsonSchema: {*

*bsonType: "object",*

*required: [ "employee\_id","name","address","contact\_details","contract"],*

*properties: {*

*employee\_id: {*

*bsonType:"string",*

*description: "Employee id cannot be empty"*

*},*

*name: {*

*bsonType:"object",*

*required: ["firstname", "lastname"],*

*properties: {*

*firstname: {*

*bsonType:"string",*

*description: "Firstname cannot be empty and must be a string"*

*},*

*lastname: {*

*bsonType:"string",*

*description: "Lastname cannot be empty and must be a string"*

*},*

*},*

*},*

*address: {*

*bsonType:"object",*

*required: ["street", "region", "pincode"],*

*properties: {*

*address: {*

*bsonType:"string",*

*description: "Address cannot be empty and must be a string"*

*},*

*region: {*

*bsonType:"string",*

*description: "region cannot be empty and must be a string"*

*},*

*pincode: {*

*bsonType:"string",*

*description: "Pincode cannot be empty and must be a string"*

*},*

*},*

*},*

*contact\_details: {*

*bsonType:"object",*

*required: ["email", "number", "emergency\_number"],*

*properties: {*

*email: {*

*bsonType:"string",*

*description: "Email cannot be empty and must be a string"*

*},*

*number: {*

*bsonType:"string",*

*description: "Contact Number cannot be empty and must be integers"*

*},*

*emergency\_number: {*

*bsonType:"string",*

*description: "Emergency Number cannot be empty and must be integers"*

*},*

*},*

*},*

*contract: {*

*bsonType:"object",*

*required: ["start\_date", "end\_date", "position", "salary"],*

*properties: {*

*start\_date: {*

*bsonType:"date",*

*description: "Contract start date cannot be empty and must be a valid date"*

*},*

*end\_date: {*

*bsonType:"date",*

*description: "Contract end date cannot be empty and must be a valid date"*

*},*

*position: {*

*bsonType:"string",*

*description: "Position cannot be empty and must be a string"*

*},*

*salary: {*

*bsonType:"int",*

*description: "Salary cannot be empty and must be integers"*

*},*

*},*

*},*

*}*

*}*

*}*

*})*

Index Creation:

*db.employees.createIndex( { "employee\_id": 1 }, { unique: true } )*

Inserting documents:

*db.employees.insertMany ([*

*{*

*employee\_id : 'em203',*

*name : {firstname: 'Rohan', lastname: 'Nair'},*

*address : {street: 'St. George Gardens, Athens avenue, Liverpool', region: 'Liverpool', pincode:'EM22RY' },*

*contact\_details : {email: 'rohann221@gmail.com', number: '+44 8276565654', emergency\_number: '+42 8276565654'},*

*contract : {start\_date: ISODate("2015-07-14"), end\_date: ISODate("2025-07-15"), position: 'steward', salary: 13000},*

*},*

*{*

*employee\_id : 'em242',*

*name : {firstname: 'Robert', lastname: 'Brown'},*

*address : {street: 'Bawdsmay building, Athens avenue, Oxford Street', region: 'Oxford Street', pincode:'GL72FY' },*

*contact\_details : {email: 'robertbr@gmail.com', number: '+91 6576556633', emergency\_number: '+91 6576556633'},*

*contract : {start\_date: ISODate("2013-05-14"), end\_date: ISODate("2023-05-15"), position: 'steward', salary: 13000},*

*},*

*{*

*employee\_id : 'em273',*

*name : {firstname: 'Alex', lastname: 'Parker'},*

*address : {street: 'Clark towers, Ramsey avenue, Citadel', region: 'Citadel', pincode:'PM64RT' },*

*contact\_details : {email: 'alexpa21@gmail.com', number: '+86 6756557733', emergency\_number: '+86 6756557733'},*

*contract : {start\_date: ISODate("2013-02-14"), end\_date: ISODate("2023-02-15"), position: 'steward', salary: 13000},*

*},*

*{*

*employee\_id : 'em256',*

*name : {firstname: 'Lauren', lastname: 'Colins'},*

*address : {street: 'Clark towers, Ramsey avenue, Citadel', region: 'Citadel', pincode:'PM64RT' },*

*contact\_details : {email: 'laurencoli@gmail.com', number: '+21 6767554411', emergency\_number: '+21 6767554411'},*

*contract : {start\_date: ISODate("2019-07-14"), end\_date: ISODate("2029-07-15"), position: 'Air-hostess', salary: 11000},*

*},*

*{*

*employee\_id : 're356',*

*name : {firstname: 'Lucas', lastname: 'Gordon'},*

*address : {street: 'St. James Gardens, Chadwell Heath, Romford', region: 'Romford', pincode:'RM65RY' },*

*contact\_details : {email: 'lucasg@gmail.com', number: '+44 8787990054', emergency\_number: '+44 8787990054'},*

*contract : {start\_date: ISODate("2013-02-14"), end\_date: ISODate("2023-02-15"), position: 'steward', salary: 13000},*

*},*

*{*

*employee\_id : 'ls45t',*

*name : {firstname: 'Eva', lastname: 'Tyler'},*

*address : {street: '12-a, Eagle Apartments, Ramsey avenue, Bond Street', region: 'Bond Street', pincode:'RG56RY' },*

*contact\_details : {email: 'evatyler12@gmail.com', number: '+25 7643998844', emergency\_number: '+25 7643998844'},*

*contract : {start\_date: ISODate("2012-11-14"), end\_date: ISODate("2022-11-15"), position: 'Air-hostess', salary: 15000},*

*},*

*{*

*employee\_id : 'me678',*

*name : {firstname: 'Calvin', lastname: 'Reed'},*

*address : {street: 'St. Mary Gardens, Bawdsway avenue, Manchester', region: 'Manchester', pincode:'YT44RT' },*

*contact\_details : {email: 'calvinrd@gmail.com', number: '+91 7766909887', emergency\_number: '+91 7766909887'},*

*contract : {start\_date: ISODate("2019-05-14"), end\_date: ISODate("2029-05-15"), position: 'steward', salary: 11000},*

*},*

*{*

*employee\_id : 'ty975',*

*name : {firstname: 'Adam', lastname: 'Grey'},*

*address : {street: '7-a, Charles Apartments, Road-man road, Bond Street', region: 'Bond Street', pincode:'RG67RY' },*

*contact\_details : {email: 'adamgr14@gmail.com', number: '+22 7966224336', emergency\_number: '+22 7966224336'},*

*contract : {start\_date: ISODate("2015-07-02"), end\_date: ISODate("2025-07-03"), position: 'steward', salary: 13000},*

*},*

*{*

*employee\_id : 'st987',*

*name : {firstname: 'Emily', lastname: 'Brown'},*

*address : {street: 'St. George Gardens, Athens avenue, Liverpool', region: 'Liverpool', pincode:'EM22RY' },*

*contact\_details : {email: 'emilybrown001@gmail.com', number: '+61 6565773293', emergency\_number: '+61 6565773293'},*

*contract : {start\_date: ISODate("2020-01-14"), end\_date: ISODate("2030-01-15"), position: 'Air-hostess', salary: 10000},*

*},*

*{*

*employee\_id : 'em564',*

*name : {firstname: 'Rachel', lastname: 'Clay'},*

*address : {street: 'St. Peters Avenue, Rotherham Junction, Liverpool', region: 'Liverpool', pincode:'EM65IY' },*

*contact\_details : {email: 'rachelcl1@gmail.com', number: '+42 6767554409', emergency\_number: '+42 6767554409'},*

*contract : {start\_date: ISODate("2020-01-14"), end\_date: ISODate("2030-01-15"), position: 'Air-hostess', salary: 10000},*

*},*

*{*

*employee\_id : 'bg564',*

*name : {firstname: 'Jeff', lastname: 'Parker'},*

*address : {street: '15, Dorsten Square, Crawley', region: ' West Sussex', pincode:'RH11 8WB' },*

*contact\_details : {email: 'jeffp45@gmail.com', number: '+44 1737773099', emergency\_number: '+44 1737773099'},*

*contract : {start\_date: ISODate("2020-07-20"), end\_date: ISODate("2030-07-21"), position: 'Steward', salary: 10000},*

*},*

*{*

*employee\_id : 'lk815',*

*name : {firstname: 'Natasha', lastname: 'Brown'},*

*address : {street: '20, Oxenholme, Kendal', region: 'Cumbria', pincode:'LA9 7RF' },*

*contact\_details : {email: 'natb87@gmail.com', number: '+44 1539724094', emergency\_number: '+44 1539724094'},*

*contract : {start\_date: ISODate("2017-01-14"), end\_date: ISODate("2027-01-15"), position: 'Air-hostess', salary: 13000},*

*},*

*{*

*employee\_id : 'iu871',*

*name : {firstname: 'Roman', lastname: 'Bradwell'},*

*address : {street: '15, Oystermouth Rd, Swansea', region: 'West Glamorgan', pincode:'SA1 3ST' },*

*contact\_details : {email: 'romanbrad76@yahoo.com', number: '+44 1792466500', emergency\_number: '+44 1792466500'},*

*contract : {start\_date: ISODate("2020-09-14"), end\_date: ISODate("2030-09-15"), position: 'Steward', salary: 10500},*

*},*

*{*

*employee\_id : 'sc354',*

*name : {firstname: 'Olivier', lastname: 'Duhamel'},*

*address : {street: '273 Avenue Daumesnil, Paris', region: 'Paris', pincode:'75012' },*

*contact\_details : {email: 'olivier87@gmail.com', number: '+43 43075697', emergency\_number: '+43 43075697'},*

*contract : {start\_date: ISODate("2018-05-24"), end\_date: ISODate("2028-05-25"), position: 'Steward', salary: 11000},*

*},*

*{*

*employee\_id : 'ab486',*

*name : {firstname: 'Ariana Almeida', lastname: 'de Mendes'},*

*address : {street: '12, Rua da Praia, Lagos', region: 'Faro', pincode:'8600-136' },*

*contact\_details : {email: 'ariana12@aol.com', number: '+351 282771200', emergency\_number: '+351 282771200'},*

*contract : {start\_date: ISODate("2020-11-14"), end\_date: ISODate("2030-11-15"), position: 'Air-hostess', salary: 10000},*

*},*

*{*

*employee\_id : 'al127',*

*name : {firstname: 'Harbhajan', lastname: 'Baral'},*

*address : {street: 'Vivekanada Circle, Girinagar, Bangalore', region: 'Bangalore', pincode:'560085' },*

*contact\_details : {email: 'harbhajan76@gmail.com', number: '+91 5673998853', emergency\_number: '+91 5673998853'},*

*contract : {start\_date: ISODate("2019-11-20"), end\_date: ISODate("2029-11-21"), position: 'Steward', salary: 10500},*

*},*

*{*

*employee\_id : 'jd645',*

*name : {firstname: 'Claire', lastname: 'Raymond'},*

*address : {street: '14 Avenue, Edmond Michelet', region: 'Cholet', pincode:'49300' },*

*contact\_details : {email: 'claire12ray@gmail.com', number: '+43 241654277', emergency\_number: '+43 241654277'},*

*contract : {start\_date: ISODate("2015-06-14"), end\_date: ISODate("2025-06-15"), position: 'Air-hostess', salary: 13000},*

*},*

*{*

*employee\_id : 'op874',*

*name : {firstname: 'Padama', lastname: 'Roy'},*

*address : {street: 'Rajnikant Jani Marg, Lalbaug, Manjalpur, Vadodara', region: 'Gujarat', pincode:'390009' },*

*contact\_details : {email: 'padmaroy001@gmail.com', number: '+91 6588913256', emergency\_number: '+91 6588913256'},*

*contract : {start\_date: ISODate("2018-08-14"), end\_date: ISODate("2028-08-15"), position: 'Air-hostess', salary: 11500},*

*},*

*{*

*employee\_id : 'dv465',*

*name : {firstname: 'Jefferey', lastname: 'Dooley'},*

*address : {street: '1118 Lakeview Crescent, Williams Lake', region: 'British Columbia', pincode:'V2G 1A3' },*

*contact\_details : {email: 'jeffreyd45@yahoo.com', number: '+1 2503923321', emergency\_number: '+1 2503923321'},*

*contract : {start\_date: ISODate("2020-02-19"), end\_date: ISODate("2030-02-20"), position: 'Steward', salary: 10500},*

*},*

*{*

*employee\_id : 'mn453',*

*name : {firstname: 'Jaswant', lastname: 'Palla'},*

*address : {street: 'Dhanpura, Haridwar, Uttarakhand,Padartha Urf Dhanpura', region: 'Uttarakhand', pincode:'247663' },*

*contact\_details : {email: 'jaswant87@gmail.com', number: '+91 8765098789', emergency\_number: '+91 8765098789'},*

*contract : {start\_date: ISODate("2018-11-24"), end\_date: ISODate("2028-11-25"), position: 'Steward', salary: 12000},*

*},*

*{*

*employee\_id : 'rt889',*

*name : {firstname: 'Stephanie', lastname: 'Jackson'},*

*address : {street: '7, Gallowgate, Newcastle upon Tyne', region: 'Tyne and Wear', pincode:'NE1 4SG' },*

*contact\_details : {email: 'stephanie56@gmail.com', number: '+44 1912329551', emergency\_number: '+44 1912329551'},*

*contract : {start\_date: ISODate("2020-01-14"), end\_date: ISODate("2025-01-15"), position: 'Booking Clerk', salary: 7000},*

*},*

*{*

*employee\_id : 'gf658',*

*name : {firstname: 'Graham', lastname: 'Taylor'},*

*address : {street: '223A Turners Hill, Cheshunt', region: 'Herts', pincode:'EN8 9DG' },*

*contact\_details : {email: 'grahamt53@gmail.com', number: '+44 9926226744', emergency\_number: '+44 9926226744'},*

*contract : {start\_date: ISODate("2018-05-14"), end\_date: ISODate("2023-05-15"), position: 'Booking Clerk', salary: 9000},*

*},*

*{*

*employee\_id : 'fg430',*

*name : {firstname: 'Helen', lastname: 'Evans'},*

*address : {street: 'Clark towers, Ramsey avenue, Citadel', region: 'Citadel', pincode:'SA61 2NB' },*

*contact\_details : {email: 'heleneva542@yahoo.com', number: '+44 5477824365', emergency\_number: '+44 5477824365'},*

*contract : {start\_date: ISODate("2019-10-14"), end\_date: ISODate("2024-10-15"), position: 'Booking Clerk', salary: 9000},*

*},*

*{*

*employee\_id : 'lm349',*

*name : {firstname: 'David', lastname: 'Harper'},*

*address : {street: '5, Avenue Square,Remington', region: 'Romford', pincode:'RM6 5RY' },*

*contact\_details : {email: 'davidhar01@gmail.com', number: '+44 6500986713', emergency\_number: '+44 6500986713'},*

*contract : {start\_date: ISODate("2017-02-14"), end\_date: ISODate("2022-02-15"), position: 'Booking Clerk', salary: 8000},*

*},*

*{*

*employee\_id : 'fg108',*

*name : {firstname: 'Randy', lastname: 'Hall'},*

*address : {street: 'Milton towers, Ramsey park, Little Heath', region: 'Romford', pincode:'RM6 5RY' },*

*contact\_details : {email: 'randyh12@gmail.com', number: '+44 3009263439', emergency\_number: '+44 3009263439'},*

*contract : {start\_date: ISODate("2018-05-14"), end\_date: ISODate("2023-05-15"), position: 'Booking Clerk', salary: 8500},*

*},*

*{*

*employee\_id : 'ar562',*

*name : {firstname: 'Joe', lastname: 'Miller'},*

*address : {street: '15G, Gordon tower , Broadway avenue, Chadwell Heath', region: 'Chadwell', pincode:'CH42RY' },*

*contact\_details : {email: 'joemiller56@aol.com', number: '+44 5488978987', emergency\_number: '+44 5488978987'},*

*contract : {start\_date: ISODate("2019-02-24"), end\_date: ISODate("2024-02-25"), position: 'Booking Clerk', salary: 7500},*

*},*

*{*

*employee\_id : 'lm127',*

*name : {firstname: 'Jasper', lastname: 'Clyne'},*

*address : {street: '15, Ripple towers, Ramsey park, Epping', region: 'Epping', pincode:'GH5 6RY' },*

*contact\_details : {email: 'jasperc98@gmail.com', number: '+44 7565018264', emergency\_number: '+44 7565018264'},*

*contract : {start\_date: ISODate("2016-12-14"), end\_date: ISODate("2021-12-15"), position: 'Booking Clerk', salary: 8000},*

*},*

*{*

*employee\_id : 'gh491',*

*name : {firstname: 'Connor', lastname: 'Wilson'},*

*address : {street: '210-212 George St, Aberdeen', region: 'Aberdeen City', pincode:'B25 1BS' },*

*contact\_details : {email: 'Connor47@gmail.com', number: '+44 5409814085', emergency\_number: '+44 5409814085'},*

*contract : {start\_date: ISODate("2018-07-24"), end\_date: ISODate("2023-07-25"), position: 'Booking Clerk', salary: 8500},*

*},*

*{*

*employee\_id : 'nc453',*

*name : {firstname: 'Jodie', lastname: 'Hunt'},*

*address : {street: '1 Ardwick St, Ramsey avenue, Merseyside', region: 'Merseyside', pincode:'WA9 1LZ' },*

*contact\_details : {email: 'jodie25@aol.com', number: '+44 9822735409', emergency\_number: '+44 9822735409'},*

*contract : {start\_date: ISODate("2019-06-16"), end\_date: ISODate("2024-06-17"), position: 'Booking Clerk', salary: 8000},*

*},*

*{*

*employee\_id : 'lk885',*

*name : {firstname: 'Mary', lastname: 'Jane'},*

*address : {street: 'Stark towers, Gordon avenue, Brook', region: 'Brook City', pincode:'TY8 7PO' },*

*contact\_details : {email: 'maryj65@yahoo.com', number: '+44 6599769870', emergency\_number: '+44 6599769870'},*

*contract : {start\_date: ISODate("2019-09-11"), end\_date: ISODate("2024-09-12"), position: 'Booking Clerk', salary: 8000},*

*}*

*])*

**Collection “flights”**

Validation**:**

*db.createCollection("flights", {*

*validator: {*

*$jsonSchema: {*

*bsonType: "object",*

*required: [ "flight\_id","plane\_id","pilot\_id","route","cabin\_crew","timings","distance\_km","duration\_hrs","price\_gbp"],*

*properties: {*

*flight\_id: {*

*bsonType:"string",*

*description: "Flight id cannot be empty, must be a string, and unique"*

*},*

*plane\_id: {*

*bsonType:"string",*

*description: "Plane id cannot be empty, must be a string"*

*},*

*pilot\_id: {*

*bsonType:"object",*

*required: [ "pilot\_id","copilot\_id"],*

*properties: {*

*pilot\_id: {*

*bsonType:"string",*

*description: "Pilot ID cannot be empty, must be a string"*

*},*

*copilot\_id: {*

*bsonType:"string",*

*description: "Co-Pilot ID cannot be empty, must be a string"*

*},*

*},*

*},*

*route: {*

*bsonType:"object",*

*required: [ "start\_airport\_id","destination\_airport\_id"],*

*properties: {*

*start\_airport\_id: {*

*bsonType:"string",*

*description: "Starting Airport ID cannot be empty, must be a string"*

*},*

*destination\_airport\_id: {*

*bsonType:"string",*

*description: "Destination Airport ID cannot be empty, must be a string"*

*},*

*},*

*},*

*cabin\_crew: {*

*bsonType:"array", items: {bsonType: "string"},*

*description: "There must be atleast one cabin crew"*

*},*

*timings: {*

*bsonType:"object",*

*required: [ "start\_date\_time","landing\_date\_time"],*

*properties: {*

*start\_date\_time: {*

*bsonType:"date",*

*description: "Starting time cannot be empty and must be ISO format"*

*},*

*landing\_date\_time: {*

*bsonType:"date",*

*description: "Landing time cannot be empty and must be ISO format"*

*},*

*},*

*},*

*distance\_km: {*

*bsonType:"int",*

*description: "Distance cannot be empty, must be an integer, in kms"*

*},*

*duration\_hrs: {*

*bsonType:"double",*

*description: "Duration cannot be empty, must be an double, in hrs"*

*},*

*price\_gbp: {*

*bsonType:"object",*

*required: ["economy"],*

*properties: {*

*business: {*

*bsonType:"int",*

*minimum: 1,*

*description: "Value must be greater than 0"*

*},*

*economy: {*

*bsonType:"int",*

*minimum: 1,*

*description: "Economy cannot be empty and value must be greater than 0"*

*}*

*}*

*}*

*}*

*}*

*}*

*})*

Index Creation:

*db.flights.createIndex( { "flight\_id": 1 }, { unique: true } )*

Inserting documents:

*db.flights.insertMany ([*

*{*

*flight\_id: 'air45\_a2',*

*plane\_id : 'a3802',*

*pilot\_id: {pilot\_id: 'p4921', copilot\_id: 'p8854'},*

*route: {start\_airport\_id: 'LHR', destination\_airport\_id:'CDG'},*

*cabin\_crew: ['em203', 'em242', 'em273', 'em256', 're356'],*

*timings: {start\_date\_time: ISODate("2021-11-25T07:15"), landing\_date\_time: ISODate("2021-11-25T08:30")},*

*distance\_km : 348,*

*duration\_hrs : 1.15,*

*price\_gbp: {business:242, economy:71},*

*},*

*{*

*flight\_id: 'air45\_b1',*

*plane\_id : 'b737M71',*

*pilot\_id: {pilot\_id: 'p7951', copilot\_id: 'p6973'},*

*route: {start\_airport\_id: 'LHR', destination\_airport\_id:'MAN'},*

*cabin\_crew: ['ls45t', 'me678'],*

*timings: {start\_date\_time: ISODate("2021-11-29T07:45"), landing\_date\_time: ISODate("2021-11-29T08:45")},*

*distance\_km : 243,*

*duration\_hrs : 1.05,*

*price\_gbp: {business:437, economy:181},*

*},*

*{*

*flight\_id: 'air45\_b2',*

*plane\_id : 'b737M72',*

*pilot\_id: {pilot\_id: 'p7951', copilot\_id: 'p6973'},*

*route: {start\_airport\_id: 'LHR', destination\_airport\_id:'SCQ'},*

*cabin\_crew: ['ls45t', 'me678'],*

*timings: {start\_date\_time: ISODate("2021-12-02T11:25"), landing\_date\_time: ISODate("2021-12-02T17:00")},*

*distance\_km : 1739,*

*duration\_hrs : 5.35,*

*price\_gbp: {business:554, economy:144},*

*},*

*{*

*flight\_id: 'air45\_a31',*

*plane\_id : 'a3301',*

*pilot\_id: {pilot\_id: 'p6938', copilot\_id: 'p5981'},*

*route: {start\_airport\_id: 'MAN', destination\_airport\_id:'EZE'},*

*cabin\_crew: ['st987', 'em564', 'bg564', 'ty975'],*

*timings: {start\_date\_time: ISODate("2021-12-01T11:25"), landing\_date\_time: ISODate("2021-12-02T05:15")},*

*distance\_km : 11173,*

*duration\_hrs : 17.50,*

*price\_gbp: {business:2987, economy:702},*

*},*

*{*

*flight\_id: 'air45\_ab71',*

*plane\_id : 'b787d1',*

*pilot\_id: {pilot\_id: 'p1984', copilot\_id: 'p4859'},*

*route: {start\_airport\_id: 'LHR', destination\_airport\_id:'JFK'},*

*cabin\_crew: ['lk815', 'iu871', 'sc354'],*

*timings: {start\_date\_time: ISODate("2021-12-02T09:15"), landing\_date\_time: ISODate("2021-12-02T18:05")},*

*distance\_km : 5534,*

*duration\_hrs : 8.50,*

*price\_gbp: {business:2210, economy:426},*

*},*

*{*

*flight\_id: 'air45\_ab72',*

*plane\_id : 'b787d2',*

*pilot\_id: {pilot\_id: 'p1984', copilot\_id: 'p4859'},*

*route: {start\_airport\_id: 'LHR', destination\_airport\_id:'KUL'},*

*cabin\_crew: ['lk815', 'iu871', 'sc354'],*

*timings: {start\_date\_time: ISODate("2021-12-05T07:15"), landing\_date\_time: ISODate("2021-12-05T22:20")},*

*distance\_km : 10611,*

*duration\_hrs : 15.50,*

*price\_gbp: {business:1967, economy:443},*

*},*

*{*

*flight\_id: 'air45\_ab73',*

*plane\_id : 'b787d3',*

*pilot\_id: {pilot\_id: 'p8719', copilot\_id: 'p5514'},*

*route: {start\_airport\_id: 'LGW', destination\_airport\_id:'BLR'},*

*cabin\_crew: ['ab486', 'al127', 'jd645'],*

*timings: {start\_date\_time: ISODate("2021-12-01T11:25"), landing\_date\_time: ISODate("2021-12-02T01:10")},*

*distance\_km : 8033,*

*duration\_hrs : 13.45,*

*price\_gbp: {business:2004, economy:697},*

*},*

*{*

*flight\_id: 'air45\_a2d',*

*plane\_id : 'a3802',*

*pilot\_id: {pilot\_id: 'p4921', copilot\_id: 'p8854'},*

*route: {start\_airport\_id: 'CDG', destination\_airport\_id:'LHR'},*

*cabin\_crew: ['em203', 'em242', 'em273', 'em256', 're356'],*

*timings: {start\_date\_time: ISODate("2021-11-26T14:15"), landing\_date\_time: ISODate("2021-11-26T15:30")},*

*distance\_km : 348,*

*duration\_hrs : 1.15,*

*price\_gbp: {business:246, economy:87},*

*},*

*{*

*flight\_id: 'air45\_b1d',*

*plane\_id : 'b737M71',*

*pilot\_id: {pilot\_id: 'p7951', copilot\_id: 'p6973'},*

*route: {start\_airport\_id: 'MAN', destination\_airport\_id:'LHR'},*

*cabin\_crew: ['ls45t', 'me678'],*

*timings: {start\_date\_time: ISODate("2021-11-29T17:30"), landing\_date\_time: ISODate("2021-11-29T18:30")},*

*distance\_km : 243,*

*duration\_hrs : 1.05,*

*price\_gbp: {business:421, economy:165},*

*},*

*{*

*flight\_id: 'air45\_b2d',*

*plane\_id : 'b737M72',*

*pilot\_id: {pilot\_id: 'p7951', copilot\_id: 'p6973'},*

*route: {start\_airport\_id: 'SCQ', destination\_airport\_id:'LHR'},*

*cabin\_crew: ['ls45t', 'me678'],*

*timings: {start\_date\_time: ISODate("2021-12-03T06:25"), landing\_date\_time: ISODate("2021-12-03T12:00")},*

*distance\_km : 1739,*

*duration\_hrs : 5.35,*

*price\_gbp: {business:560, economy:134},*

*},*

*{*

*flight\_id: 'air45\_a31d',*

*plane\_id : 'a3301',*

*pilot\_id: {pilot\_id: 'p6938', copilot\_id: 'p5981'},*

*route: {start\_airport\_id: 'EZE', destination\_airport\_id:'MAN'},*

*cabin\_crew: ['st987', 'em564', 'bg564', 'ty975'],*

*timings: {start\_date\_time: ISODate("2021-12-03T13:25"), landing\_date\_time: ISODate("2021-12-04T07:15")},*

*distance\_km : 11173,*

*duration\_hrs : 17.50,*

*price\_gbp: {business:3023, economy:781},*

*},*

*{*

*flight\_id: 'air45\_ab71d',*

*plane\_id : 'b787d1',*

*pilot\_id: {pilot\_id: 'p1984', copilot\_id: 'p4859'},*

*route: {start\_airport\_id: 'JFK', destination\_airport\_id:'LHR'},*

*cabin\_crew: ['lk815', 'iu871', 'sc354'],*

*timings: {start\_date\_time: ISODate("2021-12-03T22:30"), landing\_date\_time: ISODate("2021-12-04T07:20")},*

*distance\_km : 5534,*

*duration\_hrs : 8.50,*

*price\_gbp: {business:2145, economy:398},*

*},*

*{*

*flight\_id: 'air45\_ab72d',*

*plane\_id : 'b787d2',*

*pilot\_id: {pilot\_id: 'p1984', copilot\_id: 'p4859'},*

*route: {start\_airport\_id: 'KUL', destination\_airport\_id:'LHR'},*

*cabin\_crew: ['lk815', 'iu871', 'sc354'],*

*timings: {start\_date\_time: ISODate("2021-12-07T06:15"), landing\_date\_time: ISODate("2021-12-07T21:20")},*

*distance\_km : 10611,*

*duration\_hrs : 15.50,*

*price\_gbp: {business:2011, economy:478},*

*},*

*{*

*flight\_id: 'air45\_ab73d',*

*plane\_id : 'b787d3',*

*pilot\_id: {pilot\_id: 'p8719', copilot\_id: 'p5514'},*

*route: {start\_airport\_id: 'BLR', destination\_airport\_id:'LGW'},*

*cabin\_crew: ['ab486', 'al127', 'jd645'],*

*timings: {start\_date\_time: ISODate("2021-12-03T17:25"), landing\_date\_time: ISODate("2021-12-04T07:10")},*

*distance\_km : 8033,*

*duration\_hrs : 13.45,*

*price\_gbp: {business:2112, economy:786},*

*}*

*])*

**Collection “journey\_details”**

Validation**:**

*db.createCollection("journey\_details", {*

*validator: {*

*$jsonSchema: {*

*bsonType: "object",*

*required: [ "journey\_id","flight\_id","booking\_agent\_id","booking\_type","passenger\_ids","booking\_date"],*

*properties: {*

*journey\_id: {*

*bsonType:"string",*

*description: "PNR cannot be empty, must be a string, and unique"*

*},*

*flight\_id: {*

*bsonType:"array", items: {bsonType: "string"},*

*description: "There must be atleast one flight"*

*},*

*booking\_agent\_id: {*

*bsonType:"string",*

*description: "Booking Agent ID cannot be empty, must be a string"*

*},*

*booking\_type: {*

*enum: ["business", "economy"],*

*description: "Booking type can be either Business or Economy"*

*},*

*passenger\_ids: {*

*bsonType:"array", items: {bsonType: "string"},*

*description: "There must be atleast one passenger"*

*},*

*booking\_date: {*

*bsonType:"date",*

*description: "Booking date cannot be empty, must be an date, in ISO format"*

*}*

*}*

*}*

*}*

*})*

Index Creation:

*db.journey\_details.createIndex( { "journey\_id": 1 }, { unique: true } )*

Inserting documents:

*db.journey\_details.insertMany ([*

*{*

*journey\_id: 'JAIR45B1',*

*flight\_id: ['air45\_a2'],*

*booking\_agent\_id: 'rt889',*

*booking\_type: 'business',*

*passenger\_ids:['A45\_1310','A45\_18E1','A45\_813B','A45\_6141','A45\_D361','A45\_178A','A45\_8194','A45\_5207','A45\_4517','A45\_1220','A45\_8DC1','A45\_1399','A45\_212D','A45\_7142','A45\_A411','A45\_F181','A45\_1211','A45\_B411','A45\_1016','A45\_4201','A45\_27C2','A45\_2017','A45\_7E8D','A45\_B131','A45\_6146','A45\_F141','A45\_1311','A45\_11B2','A45\_1517','A45\_159E','A45\_19E2','A45\_A117','A45\_915B','A45\_75FC','A45\_6E10','A45\_9181','A45\_F13A','A45\_D204','A45\_8F7F','A45\_1122','A45\_3221','A45\_142E'],*

*booking\_date:ISODate("2021-10-25")*

*},*

*{*

*journey\_id: 'JAIR45E1',*

*flight\_id: ['air45\_a2'],*

*booking\_agent\_id: 'rt889',*

*booking\_type: 'economy',*

*passenger\_ids:['A45\_F197','A45\_19D1','A45\_2561','A45\_F19A','A45\_8A10','A45\_1416','A45\_125A','A45\_1352','A45\_B719','A45\_117A','A45\_1617','A45\_614A','A45\_D944','A45\_9187','A45\_1915','A45\_1114','A45\_2016','A45\_2020','A45\_205A','A45\_8A59','A45\_3101','A45\_6E21','A45\_E272','A45\_A112','A45\_A817','A45\_1294','A45\_9151','A45\_1372','A45\_6A21','A45\_FFE1','A45\_F6D1','A45\_5461','A45\_1613','A45\_2121','A45\_1241','A45\_20E1','A45\_1010','A45\_1111','A45\_73E8','A45\_1415','A45\_142A','A45\_19ED','A45\_7151','A45\_8917','A45\_A515','A45\_6C11','A45\_1911','A45\_20D1','A45\_9F17','A45\_A25B','A45\_1941','A45\_8511','A45\_FCDF','A45\_1212','A45\_2CD1','A45\_18B1','A45\_8820','A45\_5111','A45\_1252','A45\_C187','A45\_D148','A45\_3B18','A45\_B18B','A45\_F7C8','A45\_1314','A45\_1426','A45\_15CC','A45\_1710','A45\_2111','A45\_1124','A45\_181F','A45\_C15B','A45\_515D','A45\_17C1','A45\_3F7E','A45\_1971','A45\_12C1','A45\_19AD','A45\_4916','A45\_1818','A45\_F217','A45\_10E5','A45\_16F4','A45\_14F2','A45\_8EB1','A45\_1751','A45\_EAF7','A45\_7153','A45\_126C','A45\_13CC','A45\_AEF1','A45\_6327','A45\_149C','A45\_4AA6','A45\_AD15','A45\_4171','A45\_510D','A45\_1210','A45\_1020','A45\_1919','A45\_1720','A45\_A161','A45\_5EC5','A45\_A141','A45\_15B1','A45\_1017','A45\_C20F','A45\_1011','A45\_416F','A45\_10E1','A45\_E18A','A45\_851B','A45\_615B','A45\_19B2','A45\_913D','A45\_A201','A45\_1862','A45\_139D','A45\_1737','A45\_105B','A45\_F1C1','A45\_1991','A45\_4F14','A45\_2012','A45\_1920','A45\_134F','A45\_16E1','A45\_7128','A45\_7C71','A45\_1082','A45\_1320','A45\_B121','A45\_13E1','A45\_4113','A45\_1718','A45\_5E16','A45\_1531','A45\_19F1','A45\_1181','A45\_2212','A45\_DE13','A45\_1075','A45\_9117','A45\_9111','A45\_C182','A45\_BB18','A45\_12A8','A45\_1119','A45\_F101','A45\_1916','A45\_1511','A45\_8103','A45\_E38D','A45\_1552','A45\_C514','A45\_898F','A45\_FD72','A45\_8113','A45\_1924','A45\_1091','A45\_F10C','A45\_1912','A45\_F16E','A45\_173F','A45\_1D19','A45\_10FB','A45\_6942','A45\_3514','A45\_6125','A45\_209A','A45\_11A6','A45\_1715','A45\_1461','A45\_14C7','A45\_25DA','A45\_22AF','A45\_209D','A45\_F201','A45\_2071','A45\_1631','A45\_7618','A45\_211C','A45\_411B','A45\_F11F','A45\_116B','A45\_1381','A45\_116F','A45\_5639','A45\_EEC1','A45\_1713','A45\_1611','A45\_73C8','A45\_A151','A45\_2811','A45\_1391','A45\_8E14','A45\_520A','A45\_1918','A45\_199B','A45\_4619','A45\_106B','A45\_1420','A45\_15B6','A45\_14AB','A45\_4C16','A45\_A4AD','A45\_7113','A45\_D731','A45\_1714','A45\_3181','A45\_14F9','A45\_7D13','A45\_F121','A45\_B201','A45\_9201','A45\_553F','A45\_1321','A45\_1315','A45\_C122','A45\_4162','A45\_E4A1','A45\_2118','A45\_FEA1','A45\_1515','A45\_6921','A45\_5201','A45\_1620','A45\_179F','A45\_1917','A45\_1019','A45\_56D7','A45\_45D1','A45\_3131','A45\_6118','A45\_3417','A45\_1815','A45\_1455','A45\_BE13','A45\_316F','A45\_10A1','A45\_7610','A45\_E101','A45\_7B11','A45\_8119','A45\_1411','A45\_1513','A45\_19D7','A45\_1644','A45\_19CB','A45\_2011','A45\_FF14','A45\_17C6','A45\_1141','A45\_2109','A45\_A12E','A45\_16CE','A45\_CD18','A45\_2081','A45\_B633','A45\_C5C1','A45\_6101','A45\_4181','A45\_9417','A45\_1116','A45\_1115','A45\_7191','A45\_B142','A45\_C131','A45\_10B1','A45\_1541','A45\_7148','A45\_A10F','A45\_47C4','A45\_15AF','A45\_7581','A45\_6EDB','A45\_1413','A45\_D315','A45\_1553','A45\_1422','A45\_E71C','A45\_8111','A45\_1351','A45\_D913','A45\_81E1','A45\_C88F','A45\_1127','A45\_F510','A45\_E20A','A45\_12A1','A45\_ADA5','A45\_B10C','A45\_3510','A45\_9127','A45\_D516','A45\_F17F','A45\_1716','A45\_18F2','A45\_6514','A45\_715F','A45\_219E','A45\_1572','A45\_12B5','A45\_C63C','A45\_6D1A','A45\_5182','A45\_D169','A45\_919A','A45\_18D5','A45\_E189','A45\_F125','A45\_4128','A45\_9141','A45\_1961','A45\_EB4D','A45\_F4F1','A45\_9BDC','A45\_10FD','A45\_13EA','A45\_1113','A45\_7A5D','A45\_9131','A45\_1174','A45\_1207','A45\_F115','A45\_9142','A45\_9B14','A45\_9DD6','A45\_C111','A45\_1712','A45\_1412','A45\_710E','A45\_B12C','A45\_F219','A45\_E201','A45\_E161','A45\_1161','A45\_1910','A45\_F203','A45\_13BA','A45\_413E','A45\_18A1','A45\_14A1','A45\_977E','A45\_F14D','A45\_518E','A45\_1341','A45\_5181','A45\_7182','A45\_1871','A45\_ACEA','A45\_1218','A45\_2167','A45\_E715','A45\_64CB','A45\_CB20','A45\_35D1','A45\_FE20','A45\_A111','A45\_2131','A45\_23D1','A45\_185D','A45\_D16F','A45\_1797','A45\_7199','A45\_2DE1','A45\_1656','A45\_1214','A45\_202B','A45\_73F1','A45\_F310','A45\_8AB1','A45\_D11B','A45\_9178','A45\_5121','A45\_6C5B','A45\_20EB','A45\_1551','A45\_123C','A45\_12E4','A45\_10F1','A45\_2175','A45\_217B','A45\_12C4','A45\_1171','A45\_612F','A45\_E912','A45\_15F1','A45\_1012','A45\_1719','A45\_1441','A45\_1789','A45\_2166','A45\_1616','A45\_1014','A45\_8161','A45\_17E3','A45\_1741','A45\_D131','A45\_1711','A45\_EE87','A45\_FF61','A45\_2014','A45\_1510','A45\_11EE','A45\_6CF1','A45\_DBA1','A45\_DD81','A45\_1182','A45\_A215','A45\_1817','A45\_17E1','A45\_208D','A45\_A17E','A45\_19C2','A45\_C201','A45\_155D','A45\_153E','A45\_18E8','A45\_F92E','A45\_737A','A45\_149E','A45\_107F','A45\_1BEF','A45\_1493','A45\_EE14','A45\_1814','A45\_1973','A45\_CAFF','A45\_413F','A45\_387D','A45\_1821','A45\_2BD1','A45\_7171','A45\_97C4','A45\_79B1','A45\_11BA','A45\_AE38','A45\_20B6','A45\_C132','A45\_AC14','A45\_229B','A45\_6F13','A45\_1191','A45\_2DFC','A45\_9E8E','A45\_1084','A45\_1810','A45\_8D33','A45\_11D7','A45\_35CE','A45\_1232','A45\_31EE','A45\_5CA3','A45\_CD41','A45\_C9C1','A45\_6911','A45\_B196','A45\_FFCF','A45\_169E','A45\_143D','A45\_1129','A45\_9520','A45\_1612','A45\_1B18','A45\_1512','A45\_5139','A45\_16F5','A45\_5215','A45\_5155','A45\_110D','A45\_1120','A45\_1811','A45\_B173','A45\_A121','A45\_17D1','A45\_9931','A45\_7141','A45\_2061','A45\_18AB','A45\_1410','A45\_7C18','A45\_182A','A45\_4813','A45\_4363','A45\_8171','A45\_319A','A45\_B161','A45\_1587','A45\_2412','A45\_3714','A45\_12B9'],*

*booking\_date:ISODate("2021-10-19")*

*},*

*{*

*journey\_id: 'JAIR45B2',*

*flight\_id: ['air45\_b1'],*

*booking\_agent\_id: 'gf658',*

*booking\_type: 'business',*

*passenger\_ids:['A45\_1061','A45\_3D21','A45\_7139','A45\_EA11','A45\_7F55','A45\_1839','A45\_1387','A45\_EA16','A45\_C981','A45\_1373','A45\_D817','A45\_FE51','A45\_812C','A45\_6D19','A45\_8141'],*

*booking\_date:ISODate("2021-09-29")*

*},*

*{*

*journey\_id: 'JAIR45E2',*

*flight\_id: ['air45\_b1'],*

*booking\_agent\_id: 'gf658',*

*booking\_type: 'economy',*

*passenger\_ids:['A45\_1641','A45\_1419','A45\_1914','A45\_1757','A45\_FD71','A45\_9491','A45\_134C','A45\_C13C','A45\_CC21','A45\_A615','A45\_1319','A45\_38D1','A45\_213D','A45\_3441','A45\_1118','A45\_D18C','A45\_B861','A45\_3199','A45\_EF10','A45\_D283','A45\_D153','A45\_6A20','A45\_1514','A45\_2947','A45\_4219','A45\_D16E','A45\_2141','A45\_3815','A45\_3C19','A45\_2F19','A45\_3319','A45\_10D1','A45\_1BAA','A45\_4851','A45\_32FC','A45\_45C1','A45\_9162','A45\_110F','A45\_17B1','A45\_1652','A45\_7C10','A45\_1268','A45\_4114','A45\_2018','A45\_16D1','A45\_A13A','A45\_A171','A45\_2D87','A45\_2D19','A45\_5316','A45\_C142','A45\_217F','A45\_10F2','A45\_EE17','A45\_8FBD','A45\_17C3','A45\_15E8','A45\_113B','A45\_5191','A45\_9132','A45\_11D1','A45\_16A1','A45\_3151','A45\_4616','A45\_2024','A45\_743C','A45\_D15D','A45\_4AE1','A45\_1820','A45\_20B1','A45\_1518','A45\_17E5','A45\_10F9','A45\_16F3','A45\_A122','A45\_197B','A45\_162E','A45\_5E11','A45\_139C','A45\_FB11','A45\_13A1','A45\_1561','A45\_A755','A45\_1213','A45\_7898','A45\_1823','A45\_1313','A45\_118F','A45\_56B9','A45\_D419','A45\_1418','A45\_D191','A45\_A811','A45\_2095','A45\_2010','A45\_6112','A45\_EF12','A45\_9212','A45\_1519','A45\_2F20','A45\_C20E','A45\_146F','A45\_5881','A45\_D115','A45\_A812','A45\_19A1','A45\_FF28','A45\_410A','A45\_1417','A45\_4121','A45\_5141','A45\_8519','A45\_C171','A45\_B172','A45\_8714','A45\_1725','A45\_F567','A45\_8EE1','A45\_F16D','A45\_9471','A45\_C361','A45\_87A7','A45\_E181','A45\_6121','A45\_2E3C','A45\_1571','A45\_209B','A45\_A13F','A45\_316C','A45\_CE21','A45\_E17F','A45\_A131','A45\_1721','A45\_C717','A45\_E2F3','A45\_1662','A45\_1015'],*

*booking\_date:ISODate("2021-09-27")*

*},*

*{*

*journey\_id: 'JAIR45B3',*

*flight\_id: ['air45\_b2'],*

*booking\_agent\_id: 'fg430',*

*booking\_type: 'business',*

*passenger\_ids:['A45\_D152','A45\_4131','A45\_8123','A45\_C5B1','A45\_D181','A45\_2F17','A45\_17A3','A45\_C181','A45\_BD3B','A45\_1879','A45\_EF11','A45\_93B1'],*

*booking\_date:ISODate("2021-10-01")*

*},*

*{*

*journey\_id: 'JAIR45E3',*

*flight\_id: ['air45\_b2'],*

*booking\_agent\_id: 'fg430',*

*booking\_type: 'economy',*

*passenger\_ids:['A45\_7121','A45\_211B','A45\_613A','A45\_1434','A45\_1431','A45\_FD12','A45\_17EF','A45\_D144','A45\_8117','A45\_210E','A45\_BF10','A45\_5611','A45\_135C','A45\_E13C','A45\_10B2','A45\_B189','A45\_1661','A45\_19C1','A45\_C758','A45\_2015','A45\_9B12','A45\_3185','A45\_13C5','A45\_C13D','A45\_1216','A45\_C10F','A45\_4136','A45\_E17E','A45\_54C3','A45\_1149','A45\_1717','A45\_BD18','A45\_11C7','A45\_9177','A45\_4151','A45\_DE15','A45\_FF92','A45\_BD15','A45\_6A11','A45\_B135','A45\_E151','A45\_5163','A45\_E171','A45\_8327','A45\_E674','A45\_D311','A45\_3141','A45\_1591','A45\_1117','A45\_10AC','A45\_CF34','A45\_4196','A45\_17D5','A45\_1AD3','A45\_7201','A45\_2019','A45\_7101','A45\_19D6','A45\_2091','A45\_5206','A45\_1819','A45\_C141','A45\_FD16','A45\_8642','A45\_5112','A45\_9271','A45\_1615','A45\_7111','A45\_E112','A45\_3910','A45\_7512','A45\_1451','A45\_B7A1','A45\_FF41','A45\_3C17','A45\_14BE','A45\_13F2','A45\_B202','A45\_1672','A45\_82F1','A45\_AF10','A45\_218E','A45\_D151','A45\_BE20','A45\_E131','A45\_C18C','A45\_1761','A45\_8515','A45\_88C1','A45\_6E61','A45\_412C','A45\_1813','A45\_5D51','A45\_149A','A45\_B102','A45\_1981','A45\_3320','A45\_162B','A45\_B33F','A45\_61F1','A45\_D2D1','A45\_E202','A45\_19E4','A45\_2413','A45\_12E1','A45\_D561','A45\_A164','A45\_58AD','A45\_9EAB','A45\_205F','A45\_FAC1','A45\_12A7','A45\_10C1','A45\_1986','A45\_9420','A45\_719C','A45\_114C','A45\_F12B','A45\_2037','A45\_1281','A45\_A207','A45\_6361','A45\_14FC','A45\_BFEB','A45\_EAB6','A45\_DB57','A45\_F543','A45\_58CA','A45\_EA41','A45\_9101','A45\_1975','A45\_114E','A45\_C101','A45\_B613','A45\_BA13','A45\_518F','A45\_2144','A45\_F157'],*

*booking\_date:ISODate("2021-10-05")*

*},*

*{*

*journey\_id: 'JAIR45B4',*

*flight\_id: ['air45\_a31'],*

*booking\_agent\_id: 'lm349',*

*booking\_type: 'business',*

*passenger\_ids:['A45\_8712','A45\_1110','A45\_2101','A45\_C126','A45\_E315','A45\_17DD','A45\_18FD','A45\_A15D','A45\_6111','A45\_1041','A45\_DA86','A45\_3201','A45\_14C5','A45\_D171','A45\_2072','A45\_7F71','A45\_1175','A45\_D102','A45\_53CF','A45\_3166','A45\_E133','A45\_E351','A45\_1948','A45\_711A','A45\_E1B1','A45\_6192','A45\_F415','A45\_5B5B','A45\_113F','A45\_5AB1','A45\_F511','A45\_B520','A45\_224C','A45\_4142','A45\_820F','A45\_15C1'],*

*booking\_date:ISODate("2021-10-08")*

*},*

*{*

*journey\_id: 'JAIR45E4',*

*flight\_id: ['air45\_a31'],*

*booking\_agent\_id: 'lm349',*

*booking\_type: 'economy',*

*passenger\_ids:['A45\_1785','A45\_B151','A45\_6812','A45\_7A21','A45\_1691','A45\_E511','A45\_C311','A45\_1414','A45\_9E62','A45\_1101','A45\_8F13','A45\_3177','A45\_D18B','A45\_18F1','A45\_EE71','A45\_C117','A45\_4FB1','A45\_E197','A45\_4BC1','A45\_C613','A45\_D551','A45\_A37E','A45\_17A1','A45\_176F','A45\_122A','A45\_11F5','A45\_8102','A45\_E143','A45\_6815','A45\_18BE','A45\_2021','A45\_720C','A45\_C6E1','A45\_5721','A45\_7F15','A45\_11A2','A45\_1031','A45\_1932','A45\_163F','A45\_315F','A45\_CC12','A45\_813C','A45\_A191','A45\_42D1','A45\_511E','A45\_FB54','A45\_C16C','A45\_718F','A45\_44F1','A45\_4F41','A45\_7E16','A45\_1151','A45\_4141','A45\_519F','A45\_4313','A45\_119C','A45\_1614','A45\_1812','A45\_E6ED','A45\_BC81','A45\_17DB','A45\_7161','A45\_D135','A45\_14ED','A45\_5138','A45\_15A5','A45\_C191','A45\_16F1','A45\_B162','A45\_A995','A45\_617B','A45\_B7E1','A45\_A16E','A45\_75AD','A45\_4B20','A45\_5515','A45\_FA12','A45\_14D2','A45\_EF13','A45\_3874','A45\_4DCD','A45\_E17A','A45\_8415','A45\_1485','A45\_FB3F','A45\_5812','A45\_5152','A45\_2031','A45\_1619','A45\_16AC','A45\_AD68','A45\_76FA','A45\_D111','A45\_14A2','A45\_4167','A45\_1162','A45\_E9C8','A45\_1121','A45\_1622','A45\_9123','A45\_13AD','A45\_4DE4','A45\_D518','A45\_6A15','A45\_74E1','A45\_A126','A45\_18C5','A45\_8157','A45\_F191','A45\_F139','A45\_4512','A45\_199E','A45\_9158','A45\_D18E','A45\_C17E','A45\_5921','A45\_9618','A45\_F171','A45\_1778','A45\_3D11','A45\_C178','A45\_D141','A45\_15E2','A45\_4D16','A45\_B11A','A45\_207F','A45\_EF17','A45\_167B','A45\_7718','A45\_1378','A45\_1131','A45\_1651','A45\_67E1','A45\_2038','A45\_6151','A45\_C4FB','A45\_2055','A45\_188B','A45\_14BF','A45\_A7CC','A45\_7116','A45\_BCE1','A45\_1132','A45\_CABA','A45\_AA31','A45\_7531','A45\_CBC3','A45\_8EB4','A45\_2213','A45\_64A1','A45\_2519','A45\_515C','A45\_5474','A45\_8192','A45\_1219','A45\_210C','A45\_A101','A45\_2BD6','A45\_E141','A45\_F151','A45\_1051','A45\_D519','A45\_D114','A45\_F213','A45\_C20B','A45\_DB10','A45\_16BF','A45\_121D','A45\_DE17','A45\_CF57','A45\_4154','A45\_4102','A45\_7F8B','A45\_1FC1','A45\_14E5','A45\_1316','A45\_E14E','A45\_4917','A45\_15E3','A45\_D6E1','A45\_C143','A45\_127C','A45\_204D','A45\_1516','A45\_121C','A45\_C20A','A45\_83C1','A45\_D188','A45\_2116','A45\_8658','A45\_EBC5','A45\_218D','A45\_1081','A45\_14D4','A45\_F7EA','A45\_717A','A45\_EC41','A45\_7162','A45\_11F1','A45\_3A14','A45\_410B','A45\_E152','A45\_1202','A45\_3144','A45\_6518','A45\_BA16','A45\_135B','A45\_19C8','A45\_9165','A45\_AE15','A45\_1681','A45\_8ECA','A45\_14C8','A45\_4978','A45\_7815','A45\_8C5C','A45\_E851','A45\_1771','A45\_6182','A45\_9661','A45\_916A','A45\_181D','A45\_115C','A45\_1135','A45\_419C','A45\_2820','A45\_4413','A45\_5E19','A45\_717C','A45\_7132','A45\_3FE2','A45\_13C1','A45\_1112','A45\_B7D1','A45\_4410','A45\_A135','A45\_9193','A45\_A15B','A45\_2201','A45\_D201','A45\_1312','A45\_8617','A45\_1557','A45\_2664','A45\_1931','A45\_F119','A45\_11C1','A45\_3111','A45\_63AA','A45\_6F47','A45\_3C14','A45\_33B1','A45\_28B9','A45\_E971','A45\_83AC','A45\_4A56','A45\_1521','A45\_20F1','A45\_1C94','A45\_1442','A45\_1215','A45\_187C','A45\_4147','A45\_C105','A45\_49BE','A45\_1389','A45\_C17D','A45\_213A','A45\_19CE','A45\_2518','A45\_A154','A45\_B2A1','A45\_8E4C','A45\_D197','A45\_8142','A45\_E54A','A45\_4819','A45\_1491','A45\_FA10','A45\_9A71','A45\_718D','A45\_1C3A','A45\_15A1','A45\_14A3','A45\_15DC','A45\_518A','A45\_E168','A45\_192A','A45\_F161','A45\_C12A','A45\_13D2','A45\_C3E1','A45\_6816','A45\_9A12','A45\_1A2A','A45\_CE7D','A45\_6156','A45\_2112','A45\_9755','A45\_C14F','A45\_1861','A45\_1342','A45\_25B7','A45\_12D9','A45\_9AE1','A45\_20B7','A45\_4792','A45\_E810','A45\_1962','A45\_3741','A45\_E957','A45\_27A1','A45\_12EE','A45\_6819','A45\_DC51','A45\_F12A','A45\_A514','A45\_1B19','A45\_C315','A45\_F5D3','A45\_9918','A45\_AB21','A45\_112A','A45\_1D51','A45\_1221','A45\_2234','A45\_B516','A45\_9A87','A45\_17B9','A45\_B316','A45\_9511','A45\_E198','A45\_11AE','A45\_DDA1','A45\_1982','A45\_6C16','A45\_E8DD','A45\_8391','A45\_612B','A45\_5820','A45\_BA31','A45\_17AF','A45\_5D3E','A45\_FFE4','A45\_E186','A45\_818D','A45\_13F1','A45\_4158','A45\_5142','A45\_1913','A45\_4B61','A45\_A9C3','A45\_2A8C','A45\_D162','A45\_124D','A45\_13C8','A45\_164B','A45\_714D','A45\_3133','A45\_4161','A45\_710C','A45\_1393','A45\_C18F','A45\_11E7','A45\_1176','A45\_C7FA','A45\_6632','A45\_E16A','A45\_1520','A45\_BA32','A45\_1172','A45\_1C68','A45\_5616','A45\_715A','A45\_115B','A45\_1102','A45\_C818','A45\_20FB','A45\_1318','A45\_17E2','A45\_86F8','A45\_EEE6','A45\_720B','A45\_D831','A45\_A181','A45\_BCAD','A45\_EEF2','A45\_B145','A45\_FB14','A45\_11A1','A45\_D811','A45\_AF71'],*

*booking\_date:ISODate("2021-10-12")*

*},*

*{*

*journey\_id: 'JAIR45B5',*

*flight\_id: ['air45\_ab71'],*

*booking\_agent\_id: 'fg108',*

*booking\_type: 'business',*

*passenger\_ids:['A45\_3918','A45\_6215','A45\_57E1','A45\_5451','A45\_8919','A45\_1742','A45\_8116','A45\_C3FC','A45\_DE11','A45\_F8F1','A45\_14F1','A45\_3110','A45\_D19E','A45\_915A','A45\_4BE1','A45\_9616','A45\_4317','A45\_9DC2','A45\_1481','A45\_C129','A45\_2191','A45\_E16B','A45\_9116','A45\_618A','A45\_FB15'],*

*booking\_date:ISODate("2021-10-08")*

*},*

*{*

*journey\_id: 'JAIR45E5',*

*flight\_id: ['air45\_ab71'],*

*booking\_agent\_id: 'fg108',*

*booking\_type: 'economy',*

*passenger\_ids:['A45\_18FA','A45\_1F20','A45\_3931','A45\_E512','A45\_FF10','A45\_1471','A45\_47B2','A45\_F103','A45\_B5FD','A45\_4891','A45\_1193','A45\_7811','A45\_11D4','A45\_6DD1','A45\_201C','A45\_B191','A45\_4DF1','A45\_7BCA','A45\_A6C1','A45\_1831','A45\_10D4','A45\_7513','A45\_7661','A45\_17C2','A45\_6A31','A45\_E953','A45\_D113','A45\_92C3','A45\_BCD1','A45\_5720','A45\_2075','A45\_2107','A45\_2187','A45\_3124','A45\_D312','A45\_E208','A45\_ACBB','A45\_2913','A45\_1B20','A45\_1331','A45\_8861','A45\_17F3','A45\_1021','A45\_A198','A45\_1023','A45\_6161','A45\_14E1','A45\_1371','A45\_17FE','A45\_9191','A45\_18E7','A45\_19A8','A45\_1965','A45\_B203','A45\_20F6','A45\_14C1','A45\_F112','A45\_7B13','A45\_191D','A45\_6F15','A45\_C426','A45\_11FA','A45\_5ED1','A45\_1348','A45\_122E','A45\_C169','A45\_619D','A45\_43A2','A45\_B15A','A45\_1FEA','A45\_6A32','A45\_5A51','A45\_3172','A45\_51B1','A45\_20F9','A45\_2033','A45\_B714','A45\_9196','A45\_F519','A45\_EA12','A45\_4661','A45\_182F','A45\_D219','A45\_DA1A','A45\_4A81','A45\_123F','A45\_CF21','A45\_20BB','A45\_15E1','A45\_617A','A45\_A128','A45\_C3D1','A45\_815E','A45\_B14A','A45\_9112','A45\_C104','A45\_4378','A45\_1954','A45\_1274','A45\_3715','A45\_48A5','A45\_2863','A45\_EDE1','A45\_A11A','A45\_4CA1','A45\_19B1','A45\_D610','A45\_AA1E','A45\_1071','A45\_A921','A45\_3216','A45\_3F14','A45\_353B','A45\_C715','A45\_5D10','A45\_9813','A45\_20E3','A45\_867F','A45\_419E','A45\_1548','A45\_7FB5','A45\_1654','A45\_E12A','A45\_A94E','A45\_5171','A45\_318B','A45\_8D12','A45\_B632','A45\_6F7A','A45\_2EC1','A45\_159D','A45\_18A8','A45\_DD12','A45\_CE18','A45\_2D16','A45\_A142','A45\_1C11','A45\_4191','A45\_1257','A45\_B218','A45\_13D4','A45\_BAC7','A45\_2084','A45\_BFB8','A45\_1058','A45\_2F16','A45\_DC15','A45\_D5B2','A45\_5620','A45\_8614','A45\_6208','A45\_1251','A45\_51A1','A45\_C16E','A45\_510F','A45\_6315','A45\_146B','A45\_1039','A45\_9121','A45\_E97F','A45\_8810','A45\_3C67','A45\_9DC8','A45\_171E','A45\_10F6','A45\_23C1','A45\_3520','A45\_4F19','A45\_11AB','A45\_13DE','A45\_B54F','A45\_2053','A45\_156B','A45\_A10E','A45\_E166','A45\_C121','A45\_1317','A45\_6861','A45\_8847','A45\_3814','A45\_2172','A45\_313B','A45\_C66C','A45\_8C11','A45\_C622','A45\_511C','A45\_BC14','A45\_1618','A45\_6A16','A45\_165B','A45\_1337','A45\_1C20','A45\_E102','A45\_3311','A45\_2F45','A45\_20AF','A45\_AC22','A45\_13DB','A45\_95AD','A45\_E19D','A45\_A4E1','A45\_16D9','A45\_2169','A45\_16FE','A45\_1217','A45\_8D73','A45\_8BEF','A45\_A4E4','A45\_AF15','A45\_8F88','A45\_2045','A45\_CD25'],*

*booking\_date:ISODate("2021-11-14")*

*},*

*{*

*journey\_id: 'JAIR45B6',*

*flight\_id: ['air45\_ab72'],*

*booking\_agent\_id: 'ar562',*

*booking\_type: 'business',*

*passenger\_ids:['A45\_147E','A45\_3171','A45\_AB15','A45\_17EB','A45\_D18F','A45\_FB33','A45\_2B13','A45\_A813','A45\_CC10','A45\_BC8F','A45\_1791','A45\_4520','A45\_C145','A45\_1D3B','A45\_219A','A45\_3B51','A45\_FD85','A45\_BC41','A45\_A11C','A45\_F111','A45\_E215','A45\_711F','A45\_BDA1','A45\_EFC4','A45\_FF11'],*

*booking\_date:ISODate("2021-11-02")*

*},*

*{*

*journey\_id: 'JAIR45E6',*

*flight\_id: ['air45\_ab72'],*

*booking\_agent\_id: 'ar562',*

*booking\_type: 'economy',*

*passenger\_ids:['A45\_F712','A45\_15CA','A45\_FBD2','A45\_15B5','A45\_1851','A45\_D112','A45\_B141','A45\_2079','A45\_18C7','A45\_C336','A45\_2410','A45\_15F2','A45\_BF18','A45\_3A15','A45\_F551','A45\_B114','A45\_9A82','A45\_511F','A45\_EED2','A45\_D142','A45\_2082','A45\_3322','A45\_6A71','A45\_C161','A45\_20C1','A45\_E913','A45\_A11E','A45\_1629','A45\_8834','A45\_812D','A45\_EC52','A45\_2912','A45\_418A','A45\_C38A','A45\_183C','A45\_7320','A45\_E2D1','A45\_187F','A45\_93FA','A45\_1231','A45\_3DB8','A45\_B415','A45\_9219','A45\_BC18','A45\_C220','A45\_1816','A45\_1267','A45\_6187','A45\_8713','A45\_8F64','A45\_63F8','A45\_12B1','A45\_B67A','A45\_CE91','A45\_12AD','A45\_A558','A45\_16CA','A45\_5617','A45\_3FFA','A45\_6142','A45\_1841','A45\_16BD','A45\_E193','A45\_B126','A45\_B781','A45\_1123','A45\_E196','A45\_16DF','A45\_844E','A45\_7157','A45\_820B','A45\_17B6','A45\_2202','A45\_B156','A45\_9215','A45\_262B','A45\_C19C','A45\_1066','A45\_EA19','A45\_BB15','A45\_5104','A45\_118D','A45\_8169','A45\_4426','A45\_B113','A45\_DA7E','A45\_BD77','A45\_1271','A45\_2821','A45\_19EC','A45\_E142','A45\_1E20','A45\_911A','A45\_C164','A45\_ED12','A45\_DCD2','A45\_5910','A45\_2137','A45\_4277','A45\_F132','A45\_A99F','A45\_719A','A45\_B9FD','A45\_6B11','A45\_AC12','A45\_AA18','A45\_8182','A45\_F187','A45\_15BA','A45\_DF21','A45\_A106','A45\_1139','A45\_D161','A45\_C412','A45\_B17A','A45\_1013','A45\_E310','A45\_47E1','A45\_17B2','A45\_A1E1','A45\_B8A1','A45\_17BB','A45\_4A14','A45\_312B','A45\_D156','A45\_18D3','A45\_E220','A45\_5D14','A45\_2611','A45\_FB16','A45\_3162','A45\_E187','A45\_B391','A45\_3121','A45\_5A18','A45\_CE12','A45\_A614','A45\_1610','A45\_CD27','A45\_207A','A45\_B242','A45\_6317','A45\_CF10','A45\_14EE','A45\_7AE9','A45\_1967','A45\_125E','A45\_2E8F','A45\_141F','A45\_2B18','A45\_5153','A45\_EB19','A45\_1272','A45\_6188','A45\_7831','A45\_318A','A45\_4292','A45\_DB20','A45\_59A1','A45\_9145','A45\_3C61','A45\_9815','A45\_13E2','A45\_13A6','A45\_B13B','A45\_5EF2','A45\_67B1','A45\_7131','A45\_1D31','A45\_8914','A45\_C562','A45\_7761','A45\_F131','A45\_1C81','A45\_818A','A45\_AC10','A45\_E13A','A45\_115D','A45\_7178','A45\_11A8','A45\_8120','A45\_5AF4','A45\_4C12','A45\_B217','A45\_2013','A45\_515E','A45\_8F46','A45\_B163','A45\_1969','A45\_5218','A45\_2A19','A45\_12A4','A45\_8210','A45\_197A','A45\_B25B','A45\_13F7','A45\_89B6','A45\_2117','A45\_17B8','A45\_8143','A45\_616B','A45\_54A1','A45\_6C15','A45\_4C17','A45\_AB13','A45\_6162','A45\_414A','A45\_1142','A45\_11AD','A45\_1369','A45\_19AF','A45\_4146','A45\_E144','A45\_FBE1','A45\_5718','A45\_111E','A45\_8518'],*

*booking\_date:ISODate("2021-11-03")*

*},*

*{*

*journey\_id: 'JAIR45B7',*

*flight\_id: ['air45\_ab73'],*

*booking\_agent\_id: 'lm127',*

*booking\_type: 'business',*

*passenger\_ids:['A45\_195A','A45\_1BA1','A45\_9185','A45\_B171','A45\_3182','A45\_1B13','A45\_168E','A45\_4561','A45\_CB14','A45\_1881','A45\_4F15','A45\_42FE','A45\_104C','A45\_3DF9','A45\_2E61','A45\_C4E7','A45\_B101','A45\_3163','A45\_9126','A45\_8AC1','A45\_B517','A45\_1756','A45\_7681','A45\_28F1','A45\_3A20','A45\_18E5'],*

*booking\_date:ISODate("2021-11-07")*

*},*

*{*

*journey\_id: 'JAIR45E7',*

*flight\_id: ['air45\_ab73'],*

*booking\_agent\_id: 'lm127',*

*booking\_type: 'economy',*

*passenger\_ids:['A45\_C14D','A45\_2815','A45\_8D41','A45\_20B8','A45\_1828','A45\_D1E1','A45\_1025','A45\_6D61','A45\_C6B1','A45\_3143','A45\_F417','A45\_1984','A45\_19C5','A45\_A147','A45\_3AE1','A45\_1949','A45\_19B3','A45\_7472','A45\_E89C','A45\_2136','A45\_BD5A','A45\_94BE','A45\_C821','A45\_19A4','A45\_1697','A45\_A510','A45\_D16B','A45\_11C2','A45\_6D7F','A45\_4719','A45\_8121','A45\_CE39','A45\_BE5F','A45\_4418','A45\_14E3','A45\_16FA','A45\_9D13','A45\_13A2','A45\_1104','A45\_653B','A45\_31EC','A45\_153F','A45\_75D1','A45\_8451','A45\_2152','A45\_311C','A45\_1864','A45\_7D58','A45\_B931','A45\_5619','A45\_8181','A45\_9C15','A45\_A11D','A45\_11F9','A45\_A7D1','A45\_5C4C','A45\_1874','A45\_1555','A45\_11AF','A45\_5D1D','A45\_B179','A45\_20D6','A45\_174C','A45\_3179','A45\_D68F','A45\_7915','A45\_A331','A45\_117F','A45\_10CC','A45\_A816','A45\_8B14','A45\_1323','A45\_D149','A45\_4117','A45\_F613','A45\_714C','A45\_1108','A45\_7691','A45\_B8D6','A45\_3718','A45\_11DC','A45\_148C','A45\_816B','A45\_3E4A','A45\_D138','A45\_BF88','A45\_FCD4','A45\_D445','A45\_1752','A45\_6665','A45\_F189','A45\_2115','A45\_A791','A45\_19F6','A45\_1D20','A45\_CA15','A45\_3B11','A45\_D571','A45\_83ED','A45\_1083','A45\_7316','A45\_11BE','A45\_8C41','A45\_12CC','A45\_1664','A45\_3916','A45\_E520','A45\_17A9','A45\_FB19','A45\_33D1','A45\_C176','A45\_5C19','A45\_15AE','A45\_EC16','A45\_8183','A45\_13DA','A45\_771B','A45\_142C','A45\_AFC1','A45\_13D1','A45\_8101','A45\_3447','A45\_E104','A45\_8203','A45\_D192','A45\_D7D5','A45\_720D','A45\_D958','A45\_18CF','A45\_19C3','A45\_189E','A45\_1494','A45\_67A9','A45\_5C16','A45\_1EFA','A45\_2034','A45\_C20D','A45\_8621','A45\_7E9F','A45\_4820','A45\_A16C','A45\_A127','A45\_612E','A45\_D581','A45\_CF47','A45\_76F4','A45\_207B','A45\_6131','A45\_12F2','A45\_A882','A45\_F3B3','A45\_C9D5','A45\_6ECB','A45\_911B','A45\_4341','A45\_88F1','A45\_D3E2','A45\_1921','A45\_2041','A45\_719B','A45\_3EFC','A45\_105F','A45\_99DB','A45\_8195','A45\_EE51','A45\_813E','A45\_4FD6','A45\_99C8','A45\_19A6','A45\_817C','A45\_A714','A45\_2216','A45\_6772','A45\_5113','A45\_BE24','A45\_CD21','A45\_6D81','A45\_610F','A45\_1854','A45\_119D','A45\_19E1','A45\_583D','A45\_E20E','A45\_E414','A45\_E191','A45\_199D','A45\_610A','A45\_383A','A45\_19EF','A45\_FF46','A45\_E111','A45\_6171','A45\_C193','A45\_6691','A45\_A204','A45\_113C','A45\_C561','A45\_8201','A45\_319C','A45\_B124','A45\_920F','A45\_8481','A45\_A213','A45\_E147','A45\_17FD','A45\_4A13','A45\_128F','A45\_D6FA','A45\_20A1','A45\_18C1','A45\_116C'],*

*booking\_date:ISODate("2021-11-07")*

*},*

*{*

*journey\_id: 'JAIR45B1D',*

*flight\_id: ['air45\_a2d'],*

*booking\_agent\_id: 'rt889',*

*booking\_type: 'business',*

*passenger\_ids:['A45\_7DE7','A45\_8BE1','A45\_5132','A45\_1152','A45\_6991','A45\_3824','A45\_B16E','A45\_53F1','A45\_9D9C','A45\_314C','A45\_14A4','A45\_62D5','A45\_C114','A45\_BD31','A45\_9518','A45\_1583','A45\_D172','A45\_169A','A45\_15FF','A45\_2AEE','A45\_1B11','A45\_1278','A45\_12F9','A45\_F954','A45\_D165','A45\_1284','A45\_5CC1','A45\_D14F','A45\_5209','A45\_B16D','A45\_F183','A45\_1178','A45\_3112','A45\_1367','A45\_15C3','A45\_15D1','A45\_C138','A45\_2321','A45\_417D'],*

*booking\_date:ISODate("2021-10-29")*

*},*

*{*

*journey\_id: 'JAIR45E1D',*

*flight\_id: ['air45\_a2d'],*

*booking\_agent\_id: 'rt889',*

*booking\_type: 'economy',*

*passenger\_ids:['A45\_B20F','A45\_5DE1','A45\_2310','A45\_4168','A45\_EB13','A45\_1735','A45\_2510','A45\_FC21','A45\_5413','A45\_7143','A45\_7C15','A45\_A842','A45\_11C6','A45\_6941','A45\_20A2','A45\_19EB','A45\_139E','A45\_20CD','A45\_7712','A45\_611C','A45\_3210','A45\_15AB','A45\_77D7','A45\_C4E1','A45\_4518','A45\_1671','A45\_16C4','A45\_319F','A45\_5C88','A45\_515F','A45\_59C5','A45\_9761','A45\_315A','A45\_7FEC','A45\_E121','A45\_9171','A45\_17D2','A45\_B319','A45\_BAA1','A45\_7A20','A45\_B8FA','A45\_37E1','A45\_9C22','A45\_C124','A45\_115A','A45\_8A3A','A45\_C113','A45\_C549','A45\_EC82','A45\_16B1','A45\_1275','A45\_5D20','A45\_19C9','A45\_9314','A45\_A18A','A45\_1951','A45\_87E2','A45\_1B10','A45\_1B5F','A45\_3132','A45\_1361','A45\_10A7','A45\_9820','A45\_3D53','A45\_9320','A45\_8131','A45\_1018','A45\_8175','A45\_1492','A45\_3611','A45\_8733','A45\_AA12','A45\_914B','A45\_A361','A45\_B614','A45\_720A','A45\_A15A','A45\_7F16','A45\_B9B6','A45\_F618','A45\_1192','A45\_A14D','A45\_2199','A45\_1573','A45\_610E','A45\_1987','A45\_1BA9','A45\_7147','A45\_312C','A45\_C861','A45\_4176','A45\_1497','A45\_A611','A45\_A717','A45\_C616','A45\_D281','A45\_8616','A45\_E813','A45\_BBA1','A45\_16DB','A45\_E10A','A45\_5148','A45\_7237','A45\_5179','A45\_F113','A45\_17AB','A45\_D471','A45\_101B','A45\_12F1','A45\_E203','A45\_2171','A45\_3935','A45\_168A','A45\_D4D1','A45\_2161','A45\_94F1','A45\_BAFF','A45\_16D2','A45\_D411','A45\_14A8','A45\_CAA4','A45\_A517','A45\_DFA1','A45\_210A','A45\_AE19','A45\_5101','A45\_C198','A45\_3917','A45\_4511','A45\_A961','A45\_EF68','A45\_61B9','A45\_171B','A45\_7F12','A45\_7186','A45\_1488','A45\_22D1','A45\_E271','A45\_1201','A45\_BA14','A45\_7102','A45\_4A11','A45\_17F1','A45\_1621','A45\_1197','A45\_ABB1','A45\_8C4D','A45\_159F','A45\_2710','A45\_2173','A45\_144A','A45\_8C10','A45\_A241','A45\_BA17','A45\_5B15','A45\_3122','A45\_8B31','A45\_C916','A45\_2311','A45\_14D8','A45\_825F','A45\_984C','A45\_1157','A45\_DB18','A45\_919F','A45\_191F','A45\_F271','A45\_1624','A45\_363C','A45\_E1F1','A45\_916B','A45\_A918','A45\_12BE','A45\_C214','A45\_B620','A45\_C512','A45\_517F','A45\_A18F','A45\_8152','A45\_4914','A45\_D14E','A45\_1DB1','A45\_15BE','A45\_AC15','A45\_915D','A45\_186F','A45\_1475','A45\_1359','A45\_12E9','A45\_16BC','A45\_48DB','A45\_13C7','A45\_ACEB','A45\_159B','A45\_11BD','A45\_2314','A45\_E13F','A45\_6E17','A45\_A10B','A45\_A115','A45\_E12E','A45\_A661','A45\_7619','A45\_67D1','A45\_7112','A45\_4163','A45\_5815','A45\_18A2','A45\_A136','A45\_19D5','A45\_5D15','A45\_6BB6','A45\_193E','A45\_F165','A45\_B936','A45\_6168','A45\_20EC','A45\_DF17','A45\_915E','A45\_4815','A45\_8115','A45\_717E','A45\_412D','A45\_59B1','A45\_1462','A45\_CE13','A45\_1477','A45\_126D','A45\_3A2D','A45\_5216','A45\_8202','A45\_E108','A45\_9E13','A45\_C620','A45\_6951','A45\_216C','A45\_1BB2','A45\_2BA1','A45\_244B','A45\_9441','A45\_BDB1','A45\_1325','A45\_5185','A45\_98C8','A45\_6139','A45\_7816','A45\_23DC','A45\_D143','A45\_9104','A45\_6FE1','A45\_CCFC','A45\_121B','A45\_444A','A45\_8631','A45\_B14F','A45\_165F','A45\_9157','A45\_7414','A45\_AA8D','A45\_9B20','A45\_536C','A45\_3F11','A45\_5C41','A45\_C520','A45\_BEBF','A45\_1165','A45\_166A','A45\_786C','A45\_EC14','A45\_3420','A45\_10C2','A45\_BBD8','A45\_FCE1','A45\_A217','A45\_AB17','A45\_F15F','A45\_ABD1','A45\_A320','A45\_D9A2','A45\_3751','A45\_F13F','A45\_1779','A45\_FC77','A45\_7944','A45\_F58E','A45\_FED1','A45\_520E','A45\_2281','A45\_10B6','A45\_A12C','A45\_1957','A45\_766A','A45\_317A','A45\_A5C1','A45\_D117','A45\_F208','A45\_C989','A45\_9719','A45\_6B12','A45\_213E','A45\_1891','A45\_2058','A45\_7716','A45\_C49A','A45\_9316','A45\_DB93','A45\_E127','A45\_1A16','A45\_14B1','A45\_203F','A45\_C96C','A45\_F159','A45\_9148','A45\_715E','A45\_17CE','A45\_D179','A45\_427B','A45\_1581','A45\_111C','A45\_DCB1','A45\_6413','A45\_438C','A45\_20DE','A45\_12D1','A45\_910B','A45\_9B17','A45\_188F','A45\_DC20','A45\_7181','A45\_5131','A45\_2816','A45\_ADBC','A45\_14F4','A45\_E209','A45\_9C16','A45\_158A','A45\_3613','A45\_9E20','A45\_338D','A45\_CE44','A45\_314E','A45\_1392','A45\_814D','A45\_576E','A45\_37A1','A45\_1099','A45\_6D10','A45\_1579','A45\_9317','A45\_9415','A45\_4361','A45\_7E17','A45\_19CF','A45\_89F5','A45\_89FD','A45\_D186','A45\_13E4','A45\_1ED9','A45\_3196','A45\_2C7A','A45\_AF61','A45\_725A','A45\_9155','A45\_15FA','A45\_15C6','A45\_C411','A45\_4A10','A45\_1D12','A45\_DBC1','A45\_182D','A45\_2751','A45\_2215','A45\_413C','A45\_10EC','A45\_17A6','A45\_8EBD','A45\_3D56','A45\_B15B','A45\_4611','A45\_C813','A45\_1777','A45\_ABDC','A45\_8158','A45\_59D3','A45\_D881','A45\_14EB','A45\_1473','A45\_D207','A45\_A518','A45\_14E7','A45\_B312','A45\_311A','A45\_1A14','A45\_20ED','A45\_1027','A45\_119F','A45\_16B2','A45\_8651','A45\_91F9','A45\_3737','A45\_E12D','A45\_231F','A45\_5134','A45\_D7E1','A45\_BC17','A45\_14C2','A45\_11FF','A45\_FE61','A45\_C6AC','A45\_1262','A45\_F433','A45\_5123','A45\_BB10','A45\_DF11','A45\_9515','A45\_C2E1','A45\_F174','A45\_7E32','A45\_1645','A45\_A511','A45\_737D','A45\_8B39','A45\_B111','A45\_1049','A45\_3518','A45\_5192','A45\_194B','A45\_7DC2','A45\_C7F8','A45\_16B4','A45\_1228','A45\_C1DE','A45\_1582','A45\_A612','A45\_D558','A45\_17B3','A45\_1154','A45\_14FF','A45\_E381','A45\_77CB','A45\_55A1','A45\_F4E1','A45\_57B1','A45\_2E73','A45\_3B12','A45\_9331','A45\_7515','A45\_18B2','A45\_7E35','A45\_122C','A45\_7867','A45\_B4E9','A45\_1421','A45\_E17C','A45\_16FB','A45\_8E17','A45\_6185','A45\_D121','A45\_DB17','A45\_E619','A45\_7123','A45\_19CD','A45\_C185','A45\_D136','A45\_314A','A45\_B204','A45\_104B','A45\_245F','A45\_D472','A45\_B119','A45\_6F22','A45\_9371','A45\_95A1','A45\_102A','A45\_516E','A45\_16C1','A45\_1125','A45\_A199','A45\_3342','A45\_DD61','A45\_20E7','A45\_4125','A45\_7854','A45\_4153'],*

*booking\_date:ISODate("2021-11-01")*

*},*

*{*

*journey\_id: 'JAIR45B2D',*

*flight\_id: ['air45\_b1d'],*

*booking\_agent\_id: 'gf658',*

*booking\_type: 'business',*

*passenger\_ids:['A45\_5814','A45\_FB25','A45\_6181','A45\_11EB','A45\_1437','A45\_A13B','A45\_1222','A45\_F7A1','A45\_1BE8','A45\_13B1','A45\_4CA7'],*

*booking\_date:ISODate("2021-11-05")*

*},*

*{*

*journey\_id: 'JAIR45E2D',*

*flight\_id: ['air45\_b1d'],*

*booking\_agent\_id: 'gf658',*

*booking\_type: 'economy',*

*passenger\_ids:['A45\_C418','A45\_82D1','A45\_A7BD','A45\_F26D','A45\_9C58','A45\_D4FB','A45\_1443','A45\_51F1','A45\_9197','A45\_F168','A45\_F215','A45\_DE99','A45\_15D2','A45\_209C','A45\_DAC1','A45\_98C6','A45\_C4A1','A45\_6614','A45\_1335','A45\_F715','A45\_6D18','A45\_AE11','A45\_2168','A45\_C165','A45\_A120','A45\_1459','A45\_DC13','A45\_FA13','A45\_2B16','A45\_3C33','A45\_71E1','A45\_CEEE','A45\_1227','A45\_C912','A45\_5F17','A45\_1032','A45\_15E9','A45\_5418','A45\_3491','A45\_199F','A45\_6871','A45\_B18A','A45\_1753','A45\_ECBE','A45\_A412','A45\_4188','A45\_188C','A45\_B311','A45\_5C11','A45\_4206','A45\_4581','A45\_123D','A45\_B910','A45\_2032','A45\_12CD','A45\_10D2','A45\_19E9','A45\_11CA','A45\_6218','A45\_D15B','A45\_BA19','A45\_B215','A45\_F220','A45\_FDD4','A45\_4C4A','A45\_133B','A45\_1695','A45\_36F1','A45\_C863','A45\_CD14','A45\_14A9','A45\_2096','A45\_E514','A45\_3D15','A45\_41D1','A45\_C317','A45\_3D19','A45\_FD11','A45\_313A','A45\_A718','A45\_A3D1','A45\_BC1B','A45\_13FB','A45\_1596','A45\_112E','A45\_BA18','A45\_20E2','A45\_22A1','A45\_5149','A45\_5919','A45\_7615','A45\_7A5E','A45\_C139','A45\_597D','A45\_38A1','A45\_E36F','A45\_DE19','A45\_12AE','A45\_7193','A45\_10AE','A45\_2051','A45\_1347','A45\_AF21','A45\_88D1','A45\_E122','A45\_1291','A45\_D649','A45\_BC4C','A45\_2C17','A45\_14D1','A45\_813A','A45\_35A1','A45\_D912','A45\_B146','A45\_84AD','A45\_2E15','A45\_5D11','A45\_1029','A45\_7851','A45\_9C11','A45\_1382','A45\_6216','A45\_A137','A45\_3511','A45\_3313','A45\_183F','A45\_1898','A45\_512D','A45\_AA52','A45\_5514','A45\_F53A','A45\_6E14'],*

*booking\_date:ISODate("2021-10-14")*

*},*

*{*

*journey\_id: 'JAIR45B3D',*

*flight\_id: ['air45\_b2d'],*

*booking\_agent\_id: 'fg430',*

*booking\_type: 'business',*

*passenger\_ids:['A45\_6CBF','A45\_C611','A45\_D88C','A45\_29B1','A45\_2A13','A45\_219C','A45\_ACC3','A45\_613E','A45\_1635','A45\_3B71','A45\_5918','A45\_613C','A45\_152D'],*

*booking\_date:ISODate("2021-10-28")*

*},*

*{*

*journey\_id: 'JAIR45E3D',*

*flight\_id: ['air45\_b2d'],*

*booking\_agent\_id: 'fg430',*

*booking\_type: 'economy',*

*passenger\_ids:['A45\_BADC','A45\_3BD6','A45\_CC11','A45\_4193','A45\_8151','A45\_B94A','A45\_A451','A45\_E10B','A45\_11B1','A45\_2119','A45\_AB5E','A45\_59A6','A45\_14F6','A45\_517B','A45\_3175','A45\_75E8','A45\_2267','A45\_E123','A45\_B18E','A45\_2218','A45\_10D6','A45\_2157','A45\_D18D','A45\_185F','A45\_3113','A45\_E691','A45\_8215','A45\_15A2','A45\_37CD','A45\_16C3','A45\_9161','A45\_1882','A45\_BAE1','A45\_9C14','A45\_BD61','A45\_4612','A45\_A312','A45\_165D','A45\_2981','A45\_2143','A45\_D812','A45\_5719','A45\_124B','A45\_AB37','A45\_F14E','A45\_8DCE','A45\_E20B','A45\_138B','A45\_C815','A45\_3377','A45\_5120','A45\_CB15','A45\_5F15','A45\_1146','A45\_47EA','A45\_1103','A45\_6562','A45\_B16A','A45\_CFC7','A45\_2720','A45\_320D','A45\_15BF','A45\_C10D','A45\_1038','A45\_32A1','A45\_32D1','A45\_D10D','A45\_203A','A45\_1731','A45\_1E51','A45\_105E','A45\_BE14','A45\_2EA6','A45\_9C56','A45\_313D','A45\_D616','A45\_1136','A45\_F3D3','A45\_D420','A45\_F148','A45\_D175','A45\_7D89','A45\_7819','A45\_48AE','A45\_3C13','A45\_C116','A45\_14D3','A45\_A291','A45\_2610','A45\_9E51','A45\_D8D1','A45\_1781','A45\_8F35','A45\_114F','A45\_415D','A45\_173B','A45\_12E3','A45\_B129','A45\_8122','A45\_B149','A45\_C12E','A45\_C118','A45\_164F','A45\_9414','A45\_273D','A45\_B751','A45\_B416','A45\_7F2F','A45\_5186','A45\_F4B3','A45\_317D','A45\_19F3','A45\_4192','A45\_18F5','A45\_B477','A45\_122D','A45\_136E','A45\_11B9','A45\_3C91','A45\_11FD','A45\_4B12','A45\_8611','A45\_2153','A45\_9D19','A45\_5518','A45\_1FE1','A45\_D116'],*

*booking\_date:ISODate("2021-11-15")*

*},*

*{*

*journey\_id: 'JAIR45B4D',*

*flight\_id: ['air45\_a31d'],*

*booking\_agent\_id: 'lm349',*

*booking\_type: 'business',*

*passenger\_ids:['A45\_2315','A45\_A2C1','A45\_8162','A45\_B8AF','A45\_B720','A45\_39A8','A45\_2122','A45\_175E','A45\_81A5','A45\_F182','A45\_16E3','A45\_4513','A45\_9620','A45\_712B','A45\_B441','A45\_2E2D','A45\_5714','A45\_4933','A45\_CBAE','A45\_AB1E','A45\_5116','A45\_D2C8','A45\_7156','A45\_B19D','A45\_9204','A45\_1676','A45\_EE91','A45\_FA51','A45\_F842','A45\_3187','A45\_3142','A45\_6163','A45\_D820','A45\_15AC','A45\_C119','A45\_2181','A45\_FC18'],*

*booking\_date:ISODate("2021-11-21")*

*},*

*{*

*journey\_id: 'JAIR45E4D',*

*flight\_id: ['air45\_a31d'],*

*booking\_agent\_id: 'lm349',*

*booking\_type: 'economy',*

*passenger\_ids:['A45\_1226','A45\_C920','A45\_A218','A45\_D85D','A45\_FA20','A45\_6671','A45\_1C14','A45\_9298','A45\_F317','A45\_7B41','A45\_31D4','A45\_EDD9','A45\_4111','A45\_182B','A45\_2918','A45\_CE19','A45\_10CB','A45\_F2F1','A45\_AF1D','A45\_EA14','A45\_BF19','A45\_FC14','A45\_EE11','A45\_8174','A45\_135D','A45\_2102','A45\_1CF1','A45\_1792','A45\_1889','A45\_4A4F','A45\_FA2E','A45\_C516','A45\_1698','A45\_F172','A45\_5161','A45\_156E','A45\_F615','A45\_C58C','A45\_F6E6','A45\_8E16','A45\_BB17','A45\_F147','A45\_518C','A45\_2616','A45\_D182','A45\_7419','A45\_EF23','A45\_B711','A45\_3F10','A45\_AC21','A45\_8512','A45\_126F','A45\_ED19','A45\_3841','A45\_15FD','A45\_C19B','A45\_B17E','A45\_44B1','A45\_AA93','A45\_8EA1','A45\_2EA8','A45\_E5A6','A45\_A8C8','A45\_5213','A45\_411F','A45\_1988','A45\_4ABC','A45\_4E15','A45\_C992','A45\_6133','A45\_5751','A45\_D13B','A45\_135E','A45\_D718','A45\_4101','A45\_CE5A','A45\_1575','A45\_C11D','A45\_BBC4','A45\_C521','A45\_52B1','A45\_517D','A45\_C84D','A45\_127B','A45\_DB14','A45\_15F6','A45\_8191','A45\_3E85','A45\_9F3A','A45\_36F9','A45\_8711','A45\_729F','A45\_9EF1','A45\_CC51','A45\_816A','A45\_6B61','A45\_815C','A45\_2074','A45\_8C22','A45\_7981','A45\_151B','A45\_DD20','A45\_6D71','A45\_EB11','A45\_2A15','A45\_1199','A45\_4166','A45\_2E17','A45\_6412','A45\_16B8','A45\_7227','A45\_1B12','A45\_18DF','A45\_7A17','A45\_2176','A45\_C971','A45\_1783','A45\_7CB9','A45\_12E2','A45\_6A7D','A45\_4E12','A45\_8166','A45\_15C8','A45\_819E','A45\_45A1','A45\_DB71','A45\_E8ED','A45\_95F6','A45\_4F13','A45\_5151','A45\_713E','A45\_614B','A45\_559F','A45\_7D37','A45\_41E1','A45\_4261','A45\_4314','A45\_10A4','A45\_3919','A45\_1233','A45\_16AB','A45\_E9EA','A45\_A152','A45\_1856','A45\_7E13','A45\_9166','A45\_536D','A45\_1126','A45\_6C8E','A45\_65E9','A45\_1923','A45\_5109','A45\_8B8B','A45\_F912','A45\_D118','A45\_ED16','A45\_7D20','A45\_15FC','A45\_1436','A45\_34A8','A45\_789A','A45\_D119','A45\_14CF','A45\_C211','A45\_203B','A45\_19F2','A45\_2C1B','A45\_E132','A45\_B183','A45\_6CA5','A45\_B181','A45\_4831','A45\_F185','A45\_191C','A45\_DD16','A45\_5524','A45\_3B17','A45\_6124','A45\_D919','A45\_6148','A45\_6167','A45\_F210','A45\_CA12','A45\_3A78','A45\_CC38','A45\_15EC','A45\_8B91','A45\_4316','A45\_A415','A45\_89C3','A45\_BB91','A45\_7FEE','A45\_139A','A45\_9176','A45\_B109','A45\_20A4','A45\_1133','A45\_1683','A45\_3137','A45\_C208','A45\_5F20','A45\_D154','A45\_513D','A45\_DE51','A45\_134B','A45\_2C14','A45\_118C','A45\_A713','A45\_5D17','A45\_710B','A45\_9617','A45\_1496','A45\_8164','A45\_2138','A45\_20C9','A45\_BB6F','A45\_F823','A45\_4ED1','A45\_15DA','A45\_6F9E','A45\_C151','A45\_9413','A45\_2105','A45\_FE11','A45\_A716','A45\_DC10','A45\_D911','A45\_5195','A45\_13C6','A45\_EE12','A45\_112C','A45\_A681','A45\_A118','A45\_6EF1','A45\_1533','A45\_3193','A45\_16F7','A45\_6447','A45\_68F1','A45\_912C','A45\_1264','A45\_E617','A45\_F81E','A45\_5220','A45\_18AF','A45\_220A','A45\_6158','A45\_6136','A45\_2089','A45\_12F6','A45\_A832','A45\_6D11','A45\_8920','A45\_8136','A45\_53C2','A45\_11D2','A45\_7B15','A45\_5416','A45\_618B','A45\_8179','A45\_3189','A45\_53A6','A45\_A149','A45\_113A','A45\_CC4E','A45\_2DCF','A45\_15EE','A45\_7F51','A45\_C10E','A45\_918F','A45\_D108','A45\_1265','A45\_1D4E','A45\_D520','A45\_FC20','A45\_55FE','A45\_167E','A45\_218B','A45\_5B16','A45\_8E87','A45\_EC17','A45\_EB56','A45\_4C20','A45\_17CB','A45\_11F3','A45\_E817','A45\_810A','A45\_D95F','A45\_9281','A45\_7167','A45\_8F12','A45\_7159','A45\_4FF1','A45\_DF13','A45\_7A15','A45\_2B14','A45\_56E1','A45\_1584','A45\_BE51','A45\_C610','A45\_51C6','A45\_29C6','A45\_1C18','A45\_1328','A45\_AD87','A45\_8933','A45\_EAA9','A45\_895E','A45\_1239','A45\_65A1','A45\_B117','A45\_F5E9','A45\_4124','A45\_35C1','A45\_6149','A45\_206F','A45\_3BC1','A45\_20A8','A45\_8110','A45\_204A','A45\_67C1','A45\_F116','A45\_1776','A45\_1772','A45\_24BA','A45\_11CB','A45\_2151','A45\_111D','A45\_182E','A45\_418F','A45\_C915','A45\_F108','A45\_19FF','A45\_163D','A45\_E182','A45\_C571','A45\_AFC6','A45\_1832','A45\_9812','A45\_7133','A45\_1853','A45\_12F4','A45\_6201','A45\_7AD8','A45\_7203','A45\_2717','A45\_E820','A45\_98C1','A45\_C172','A45\_84EA','A45\_4922','A45\_9F82','A45\_8DB1','A45\_7B81','A45\_C631','A45\_11E8','A45\_117B','A45\_1357','A45\_8212','A45\_36C1','A45\_CD16','A45\_1964','A45\_65F1','A45\_EC65','A45\_1899','A45\_A211','A45\_8C61','A45\_8B11','A45\_2813','A45\_F15D','A45\_134D','A45\_125F','A45\_413B','A45\_7B10','A45\_CDD1','A45\_5A11','A45\_5351','A45\_13CD','A45\_D681','A45\_38D3','A45\_3EB9','A45\_B913','A45\_7FF1','A45\_F123','A45\_6F12','A45\_ECF5','A45\_F761','A45\_F152','A45\_2312','A45\_161C','A45\_AD2D','A45\_AE18','A45\_2A1E','A45\_4185'],*

*booking\_date:ISODate("2021-10-13")*

*},*

*{*

*journey\_id: 'JAIR45B5D',*

*flight\_id: ['air45\_ab71d'],*

*booking\_agent\_id: 'fg108',*

*booking\_type: 'business',*

*passenger\_ids:['A45\_1977','A45\_14EC','A45\_3214','A45\_3716','A45\_2083','A45\_6E51','A45\_9551','A45\_BE67','A45\_BFA1','A45\_7B69','A45\_F199','A45\_15A6','A45\_B771','A45\_B136','A45\_54DF','A45\_A162','A45\_6214','A45\_136D','A45\_6711','A45\_72A1'],*

*booking\_date:ISODate("2021-11-08")*

*},*

*{*

*journey\_id: 'JAIR45E5D',*

*flight\_id: ['air45\_ab71d'],*

*booking\_agent\_id: 'fg108',*

*booking\_type: 'economy',*

*passenger\_ids:['A45\_625C','A45\_DE55','A45\_8921','A45\_5312','A45\_10A2','A45\_3191','A45\_4215','A45\_3347','A45\_A2D1','A45\_6132','A45\_BC20','A45\_5663','A45\_D915','A45\_F351','A45\_107B','A45\_D103','A45\_ACA1','A45\_515B','A45\_162F','A45\_65D1','A45\_A3A1','A45\_C48E','A45\_A581','A45\_FB37','A45\_A119','A45\_A1D1','A45\_E6B6','A45\_B182','A45\_B828','A45\_4B14','A45\_13E5','A45\_132B','A45\_12C7','A45\_A158','A45\_2211','A45\_E163','A45\_B152','A45\_AAF6','A45\_B3B6','A45\_7BF8','A45\_618F','A45\_1595','A45\_B810','A45\_15D8','A45\_FACF','A45\_1332','A45\_19D2','A45\_A114','A45\_7FC1','A45\_4AC1','A45\_E620','A45\_C81E','A45\_7173','A45\_376F','A45\_E451','A45\_F169','A45\_18CE','A45\_11D5','A45\_9651','A45\_7210','A45\_539E','A45\_115F','A45\_6199','A45\_7107','A45\_6C7E','A45\_5571','A45\_1052','A45\_E14A','A45\_19E7','A45\_1254','A45\_7571','A45\_D53A','A45\_6E48','A45\_1128','A45\_22B4','A45\_2186','A45\_202E','A45\_7198','A45\_E204','A45\_9712','A45\_145B','A45\_1995','A45\_7C11','A45\_D6C7','A45\_CBE9','A45\_2515','A45\_2162','A45\_2915','A45\_8B9B','A45\_D2C6','A45\_B210','A45\_132A','A45\_E510','A45\_3412','A45\_A76D','A45\_917A','A45\_137E','A45\_488E','A45\_37A2','A45\_F1EF','A45\_3E10','A45\_FEC3','A45\_8172','A45\_FB89','A45\_143A','A45\_A168','A45\_5D18','A45\_19BE','A45\_10AB','A45\_3D29','A45\_6719','A45\_E719','A45\_B13E','A45\_D109','A45\_6A85','A45\_ED51','A45\_D11F','A45\_6DC1','A45\_B186','A45\_4610','A45\_B79D','A45\_AC17','A45\_5581','A45\_6236','A45\_195E','A45\_CC59','A45\_2214','A45\_1385','A45\_E167','A45\_37F1','A45\_C128','A45\_DD2D','A45\_9619','A45\_B4F2','A45\_CE3E','A45\_D622','A45\_B710','A45\_10C6','A45\_128B','A45\_5C91','A45\_C332','A45\_106E','A45\_C492','A45\_2517','A45\_BBAD','A45\_417F','A45\_19FE','A45\_DF16','A45\_CF11','A45\_4DD1','A45\_DE21','A45\_4E9F','A45\_16B6','A45\_2618','A45\_D20F','A45\_9F13','A45\_641C','A45\_E153','A45\_DDC2','A45\_7177','A45\_4D61','A45\_83B5','A45\_A697','A45\_4E18','A45\_A209','A45\_16C5','A45\_3C6C','A45\_A761','A45\_7134','A45\_20B2','A45\_20A7','A45\_1261','A45\_BB20','A45\_13CA','A45\_4211','A45\_1562','A45\_20F4','A45\_A5B1','A45\_410D','A45\_2615','A45\_10DD','A45\_C12D','A45\_12B2','A45\_112B','A45\_155C','A45\_5199','A45\_D39E','A45\_C351','A45\_19BF','A45\_F122','A45\_1168','A45\_D415','A45\_9B11','A45\_85B2','A45\_4A18'],*

*booking\_date:ISODate("2021-10-04")*

*},*

*{*

*journey\_id: 'JAIR45B6D',*

*flight\_id: ['air45\_ab72d'],*

*booking\_agent\_id: 'ar562',*

*booking\_type: 'business',*

*passenger\_ids:['A45\_117E','A45\_C417','A45\_B918','A45\_1098','A45\_B412','A45\_A988','A45\_C314','A45\_109C','A45\_18D1','A45\_3C1F','A45\_B9CA','A45\_157E','A45\_178E','A45\_328F','A45\_12DE','A45\_DD10','A45\_15B8','A45\_514D','A45\_8311','A45\_712D','A45\_FE16','A45\_16A7','A45\_C615','A45\_5D13','A45\_6689'],*

*booking\_date:ISODate("2021-11-22")*

*},*

*{*

*journey\_id: 'JAIR45E6D',*

*flight\_id: ['air45\_ab72d'],*

*booking\_agent\_id: 'ar562',*

*booking\_type: 'economy',*

*passenger\_ids:['A45\_1093','A45\_3414','A45\_18C4','A45\_F206','A45\_75EA','A45\_F109','A45\_2B64','A45\_5C13','A45\_F472','A45\_1686','A45\_25E4','A45\_9512','A45\_1897','A45\_1A3F','A45\_19DC','A45\_7CC1','A45\_21E1','A45\_8513','A45\_918D','A45\_1137','A45\_F981','A45\_61AF','A45\_102F','A45\_CF8B','A45\_567E','A45\_1554','A45\_39A1','A45\_3161','A45\_63B1','A45\_11E1','A45\_2E12','A45\_2791','A45\_8E15','A45\_519C','A45\_DFC8','A45\_25C1','A45\_1287','A45\_8FFF','A45\_2A6A','A45\_1677','A45\_8D14','A45\_A19A','A45\_D13D','A45\_3610','A45\_3136','A45\_2919','A45\_5CE7','A45\_6110','A45\_2E10','A45\_E231','A45\_A516','A45\_8632','A45\_2C13','A45\_9199','A45\_C2B9','A45\_2A3D','A45\_9156','A45\_5A12','A45\_15A8','A45\_ECB1','A45\_39D1','A45\_94B9','A45\_2520','A45\_6211','A45\_3204','A45\_C320','A45\_1722','A45\_1C92','A45\_118E','A45\_14BC','A45\_315E','A45\_7179','A45\_20DF','A45\_1042','A45\_14B5','A45\_189B','A45\_E891','A45\_2BFD','A45\_20BD','A45\_F19D','A45\_2911','A45\_1844','A45\_8104','A45\_C16F','A45\_1632','A45\_1445','A45\_E211','A45\_977B','A45\_3C7E','A45\_5B18','A45\_685F','A45\_1453','A45\_2147','A45\_17AC','A45\_3212','A45\_6C52','A45\_BC12','A45\_10C5','A45\_DF12','A45\_C4AE','A45\_DC14','A45\_E513','A45\_F16B','A45\_995D','A45\_967F','A45\_2AF6','A45\_7CEB','A45\_D122','A45\_8517','A45\_E261','A45\_56DB','A45\_138D','A45\_208C','A45\_C2A8','A45\_9614','A45\_68C9','A45\_175C','A45\_4375','A45\_7714','A45\_897F','A45\_6B18','A45\_2183','A45\_13B3','A45\_DB11','A45\_8934','A45\_BC11','A45\_2521','A45\_17D6','A45\_19E6','A45\_6172','A45\_179C','A45\_C13A','A45\_CA84','A45\_75DF','A45\_B8F1','A45\_1F15','A45\_13D5','A45\_14F3','A45\_18A4','A45\_B120','A45\_B12F','A45\_7FA6','A45\_14AC','A45\_19D4','A45\_CBD3','A45\_B166','A45\_168F','A45\_15D5','A45\_B2D1','A45\_6207','A45\_C110','A45\_FFB1','A45\_10C3','A45\_1866','A45\_B571','A45\_99A5','A45\_10E2','A45\_6122','A45\_B9AF','A45\_B4A2','A45\_6A3F','A45\_4D8A','A45\_CDA4','A45\_EA38','A45\_B133','A45\_F15C','A45\_A6A5','A45\_572E','A45\_CB19','A45\_6427','A45\_8F9E','A45\_4991','A45\_314F','A45\_8E20','A45\_1EB2','A45\_D1AC','A45\_5C21','A45\_1846','A45\_23F2','A45\_3512','A45\_12FE','A45\_5811','A45\_FBD1','A45\_A820','A45\_58A1','A45\_F154','A45\_5219','A45\_1578','A45\_AC78','A45\_1E21','A45\_CC16','A45\_C19E','A45\_11AC','A45\_FA11','A45\_C16A','A45\_B428','A45\_7196','A45\_E896','A45\_617E','A45\_6114','A45\_6220','A45\_18D8','A45\_9F18','A45\_1397','A45\_1594','A45\_14E2','A45\_1028','A45\_7BA1','A45\_8BF4','A45\_10B4','A45\_CC17','A45\_5119','A45\_2B8A','A45\_4119','A45\_F188','A45\_4E10'],*

*booking\_date:ISODate("2021-11-13")*

*},*

*{*

*journey\_id: 'JAIR45B7D',*

*flight\_id: ['air45\_ab73d'],*

*booking\_agent\_id: 'lm127',*

*booking\_type: 'business',*

*passenger\_ids:['A45\_179A','A45\_CE5C','A45\_514B','A45\_F1A9','A45\_6226','A45\_312F','A45\_DE10','A45\_55F2','A45\_C103','A45\_4F9C','A45\_D757','A45\_6175','A45\_D63E','A45\_D9D1','A45\_3717','A45\_D173','A45\_E7B9','A45\_83E5','A45\_B636','A45\_97A4','A45\_812F','A45\_DAF1','A45\_E146'],*

*booking\_date:ISODate("2021-10-07")*

*},*

*{*

*journey\_id: 'JAIR45E7D',*

*flight\_id: ['air45\_ab73d'],*

*booking\_agent\_id: 'lm127',*

*booking\_type: 'economy',*

*passenger\_ids:['A45\_3119','A45\_C12F','A45\_4149','A45\_318E','A45\_B811','A45\_129B','A45\_D711','A45\_DEA4','A45\_D720','A45\_EF16','A45\_181E','A45\_7FF5','A45\_528D','A45\_1928','A45\_14AF','A45\_89C1','A45\_76BF','A45\_AC16','A45\_5F25','A45\_4BEB','A45\_6D37','A45\_D7B8','A45\_F7CE','A45\_103E','A45\_84CC','A45\_4715','A45\_EB12','A45\_17FB','A45\_5189','A45\_220E','A45\_6611','A45\_77E4','A45\_416B','A45\_19FB','A45\_6FA3','A45\_3D10','A45\_6119','A45\_213C','A45\_6D21','A45\_3116','A45\_99E2','A45\_F316','A45\_F964','A45\_6108','A45\_D101','A45\_7EA7','A45\_1952','A45\_C3C1','A45\_FD97','A45\_6641','A45\_9CA1','A45\_4717','A45\_4B31','A45\_516B','A45\_6191','A45\_11E2','A45\_12A2','A45\_261D','A45\_148F','A45\_9E9C','A45\_2FDF','A45\_9CF1','A45\_315C','A45\_11A7','A45\_9A19','A45\_415F','A45\_1289','A45\_DB16','A45\_15F8','A45\_913B','A45\_D510','A45\_1F51','A45\_167F','A45\_5B75','A45\_D613','A45\_4E71','A45\_28E3','A45\_E81A','A45\_A19B','A45\_E109','A45\_EE2B','A45\_C209','A45\_1782','A45\_1567','A45\_B511','A45\_17DC','A45\_C112','A45\_411D','A45\_C215','A45\_F681','A45\_AC11','A45\_6261','A45\_F612','A45\_3331','A45\_3410','A45\_16E5','A45\_F117','A45\_B631','A45\_D531','A45\_C3FE','A45\_3318','A45\_F5D6','A45\_2B21','A45\_A157','A45\_FDF1','A45\_192C','A45\_16E9','A45\_B16B','A45\_7614','A45\_20CF','A45\_C419','A45\_AEF5','A45\_9164','A45\_716D','A45\_1687','A45\_1893','A45\_711B','A45\_A156','A45\_A6E6','A45\_2F15','A45\_D574','A45\_AFD7','A45\_D981','A45\_161D','A45\_5164','A45\_7195','A45\_E4FC','A45\_20AD','A45\_5D28','A45\_18ED','A45\_2782','A45\_B1C8','A45\_C17C','A45\_6541','A45\_6DB9','A45\_139B','A45\_C196','A45\_2819','A45\_184D','A45\_BF1E','A45\_4178','A45\_6712','A45\_220D','A45\_135F','A45\_17C5','A45\_14FB','A45\_4189','A45\_7115','A45\_6741','A45\_611B','A45\_A51D','A45\_6DA4','A45\_3DA1','A45\_192F','A45\_C711','A45\_98B7','A45\_FF91','A45\_B116','A45\_167A','A45\_151C','A45\_8761','A45\_2414','A45\_7152','A45\_E117','A45\_7146','A45\_20F2','A45\_2120','A45\_15B4','A45\_2DAF','A45\_9154','A45\_DC12','A45\_2189','A45\_6134','A45\_1482','A45\_13D8','A45\_7611','A45\_98B8','A45\_5162','A45\_7C89','A45\_7A12','A45\_11FB','A45\_BA15','A45\_7328','A45\_C910','A45\_819C','A45\_108C','A45\_1297','A45\_BB16','A45\_1565','A45\_5912','A45\_2066','A45\_5962','A45\_5E5C','A45\_1848','A45\_D97D','A45\_9917','A45\_B371','A45\_BD7C','A45\_8918','A45\_513B','A45\_D124','A45\_BDE1','A45\_A3DD','A45\_F5A9','A45\_10ED','A45\_9D17','A45\_C8F7','A45\_A814','A45\_118B','A45\_8A4A','A45\_1433','A45\_103A','A45\_CA9B','A45\_10C8','A45\_E312','A45\_F617','A45\_A107','A45\_16E2','A45\_6176','A45\_187B'],*

*booking\_date:ISODate("2021-11-09")*

*}*

*])*

**Collection “pilots”**

Validation**:**

*db.createCollection("pilots", {*

*validator: { $jsonSchema: {*

*bsonType: "object",*

*required: [ "pilot\_id", "fullName", "address", "contactDetails", "contract", "lastFitToFlyTest"],*

*properties: {*

*pilot\_id: {*

*bsonType: "string",*

*description: "A unique pilot ID, must be string"*

*},*

*fullName: {*

*bsonType: "string",*

*description: "Pilot's full name, must be string"*

*},*

*address: {*

*bsonType:"object",*

*required: ["street", "zip", "city", "region"],*

*properties:*

*{*

*street: {bsonType: "string", description: "Street must be a string and is required"},*

*zip: {bsonType: "string", description: "Zip must be a string and is required"},*

*city: {bsonType: "string", description: "City must be a string and is required"},*

*region: {bsonType: "string", description: "State must be a string and is required"}*

*}},*

*contactDetails: {*

*bsonType:"object",*

*required: ["email", "phoneNumber", "emergencyNumber"],*

*properties:*

*{*

*email: {bsonType: "string", description: "Email must be a string and is required"},*

*phoneNumber: {bsonType: "string", description: "PhoneNumber must be a string and is required"},*

*emergencyNumber: {bsonType: "string", description: "EmergencyNumber must be a string and is required"}*

*}},*

*contract: {*

*bsonType:"object",*

*required: ["startDate", "endDate", "salary"],*

*properties:*

*{*

*startDate: {bsonType: "date", description: "StartDate must be a date and is required"},*

*endDate: {bsonType: "date", description: "EndDate must be an date and is required"},*

*salary: {bsonType: "double", description: "Salary must be a double and is required"}*

*}},*

*}*

*} },*

*} )*

Index Creation:

*db.pilots.createIndex( { "pilot\_id": 1 }, { unique: true } )*

Inserting documents:

*db.pilots.insertMany ([*

*{*

*pilot\_id: "p4921",*

*fullName: "Alexander Avery",*

*address: {street: "Main Rd", zip: "SY5 7JD", city: "Shrewsbury", region: "Shropshire"*

*},*

*contactDetails: {*

*email: "a\_avery@gmail.com",*

*phoneNumber: "+44 5789 649816", emergencyNumber: "+44 4392 467819"*

*},*

*contract: {*

*startDate: ISODate("2018-04-17"), endDate: ISODate("2023-06-18"), salary: 7564.5*

*},*

*employmentRecord: [*

*{startDate: ISODate("2014-02-25"), endDate: ISODate("2018-02-26"), salary: 6492.2},*

*{startDate: ISODate("2012-02-23"), endDate: ISODate("2014-02-24"), salary: 5270.8}*

*],*

*lastFitToFlyTest: ISODate("2021-07-14")*

*},*

*{*

*pilot\_id: "p7951",*

*fullName: "Edward Campbell",*

*address: {street: "Wash Farm", zip: "TQ11 0JU", city: "Buckfastleigh", region: "Devon"*

*},*

*contactDetails: {*

*email: "e\_campbell758@gmail.com",*

*phoneNumber: "+44 4619 784915", emergencyNumber: "+44 4659 157943"*

*},*

*contract: {*

*startDate: ISODate("2017-08-01"), endDate: ISODate("2022-07-01"), salary: 6496.4*

*},*

*employmentRecord: [*

*{startDate: ISODate("2015-09-01"), endDate: ISODate("2017-07-31"), salary: 4596.3}*

*],*

*lastFitToFlyTest: ISODate("2021-12-15")*

*},*

*{*

*pilot\_id: "p8854",*

*fullName: "Nicholas Lyman",*

*address: {street: "51 Upper Berkeley St", zip: "W1H 7QW", city: " Marylebone", region: "London"*

*},*

*contactDetails: {*

*email: "nicholas\_ly76@gmail.com",*

*phoneNumber: "+44 6794 124896", emergencyNumber: "+44 5921 689134"*

*},*

*contract: {*

*startDate: ISODate("2016-01-13"), endDate: ISODate("2022-04-13"), salary: 7184.8*

*},*

*employmentRecord: [*

*{startDate: ISODate("2012-06-12"), endDate: ISODate("2016-01-12"), salary: 5492.2},*

*{startDate: ISODate("2011-04-11"), endDate: ISODate("2012-06-11"), salary: 4589.7},*

*{startDate: ISODate("2010-10-10"), endDate: ISODate("2011-04-10"), salary: 4995.6}*

*],*

*lastFitToFlyTest: ISODate("2021-06-30")*

*},*

*{*

*pilot\_id: "p6973",*

*fullName: "Thomas Johnston",*

*address: {street: "Tabernacle St", zip: " TR1 2EJ", city: "Truro", region: "Cornwall"*

*},*

*contactDetails: {*

*email: "th\_Joh791@gmail.com",*

*phoneNumber: "+44 9781 269187", emergencyNumber: "+44 3581 367815"*

*},*

*contract: {*

*startDate: ISODate("2019-08-14"), endDate: ISODate("2025-11-14"), salary: 7329.4*

*},*

*employmentRecord: [],*

*lastFitToFlyTest: ISODate("2021-07-07")*

*},*

*{*

*pilot\_id: "p8719",*

*fullName: "Victor Knox",*

*address: {street: "75 Belfast Rd", zip: "BT38 8BX", city: "Carrickfergus", region: "Antrim"*

*},*

*contactDetails: {*

*email: "victor276\_ox@gmail.com",*

*phoneNumber: "+44 2849 365871", emergencyNumber: "+44 6584 128459"*

*},*

*contract: {*

*startDate: ISODate("2020-04-17"), endDate: ISODate("2023-12-17"), salary: 5489.8*

*},*

*employmentRecord: [*

*{startDate: ISODate("2017-02-16"), endDate: ISODate("2020-04-16"), salary: 5018.5},*

*{startDate: ISODate("2015-01-15"), endDate: ISODate("2017-02-15"), salary: 4682.2},*

*{startDate: ISODate("2013-12-14"), endDate: ISODate("2015-01-14"), salary: 4156.8}*

*],*

*lastFitToFlyTest: ISODate("2021-12-05")*

*},*

*{*

*pilot\_id: "p5514",*

*fullName: "Owen Ferguson",*

*address: {street: "179 High St", zip: "SS1 1LL", city: "SOUTHEND ON SEA", region: "Essex"*

*},*

*contactDetails: {*

*email: "ferg\_ow87@gmail.com",*

*phoneNumber: "+44 3598 155266", emergencyNumber: "+44 1584 111458"*

*},*

*contract: {*

*startDate: ISODate("2019-07-17"), endDate: ISODate("2022-12-17"), salary: 4158.6*

*},*

*employmentRecord: [],*

*lastFitToFlyTest: ISODate("2021-11-10")*

*},*

*{*

*pilot\_id: "p6938",*

*fullName: "Joseph Smith",*

*address: {street: "42 Heatherburn Ct", zip: "DL5 4UZ", city: "Newton Aycliffe", region: "County Durham"*

*},*

*contactDetails: {*

*email: "jo76\_smith@gmail.com",*

*phoneNumber: "+44 8966 485571", emergencyNumber: "+44 6662 445932"*

*},*

*contract: {*

*startDate: ISODate("2017-06-31"), endDate: ISODate("2021-12-31"), salary: 6894.6*

*},*

*employmentRecord: [*

*{startDate: ISODate("2015-02-30"), endDate: ISODate("2017-06-30"), salary: 6289.4},*

*{startDate: ISODate("2013-10-29"), endDate: ISODate("2015-02-29"), salary: 5471.6}*

*],*

*lastFitToFlyTest: ISODate("2021-10-28")*

*},*

*{*

*pilot\_id: "p5981",*

*fullName: "David Simpson",*

*address: {street: "Woodhouse Rd", zip: "N12 9EY", city: "Finchley", region: "London"*

*},*

*contactDetails: {*

*email: "d\_simpson43@gmail.com",*

*phoneNumber: "+44 9544 165448", emergencyNumber: "+44 6611 449862"*

*},*

*contract: {*

*startDate: ISODate("2017-09-13"), endDate: ISODate("2023-02-19"), salary: 7482.5*

*},*

*employmentRecord: [*

*{startDate: ISODate("2014-04-13"), endDate: ISODate("2017-09-13"), salary: 6492.2}*

*],*

*lastFitToFlyTest: ISODate("2021-09-01")*

*},*

*{*

*pilot\_id: "p1984",*

*fullName: "Michael Turner",*

*address: {street: "105 London Rd", zip: "CM21 9JJ", city: "Sawbridgeworth", region: "Hertfordshire"*

*},*

*contactDetails: {*

*email: "michael\_t763@gmail.com",*

*phoneNumber: "+44 1644 458485", emergencyNumber: "+44 6225 455114"*

*},*

*contract: {*

*startDate: ISODate("2020-06-17"), endDate: ISODate("2021-06-17"), salary: 4128.9*

*},*

*employmentRecord: [],*

*lastFitToFlyTest: ISODate("2021-06-10")*

*},*

*{*

*pilot\_id: "p4859",*

*fullName: "Leonard Ward",*

*address: {street: "404 Kenton Ln", zip: "HA3 8RQ", city: "Harrow", region: "Greater London"*

*},*

*contactDetails: {*

*email: "leo\_ward48@gmail.com",*

*phoneNumber: "+44 8996 265114", emergencyNumber: "+44 4855 144855"*

*},*

*contract: {*

*startDate: ISODate("2019-06-27"), endDate: ISODate("2022-07-27"), salary: 5254.6*

*},*

*employmentRecord: [*

*{startDate: ISODate("2017-02-26"), endDate: ISODate("2019-06-26"), salary: 4956.8},*

*{startDate: ISODate("2016-01-25"), endDate: ISODate("2017-02-25"), salary: 4581.9}*

*],*

*lastFitToFlyTest: ISODate("2021-09-23")*

*}*

*])*

**Collection “planes”**

Validation**:**

*db.createCollection("planes", {*

*validator: {*

*$jsonSchema: {*

*bsonType: "object",*

*required: [ "plane\_id","brand","model","range","capacity","lengthOfService\_months","remaininglengthOfService\_months"],*

*properties: {*

*plane\_id: {*

*bsonType:"string",*

*description: "Plane id cannot be empty, must be a string, and unique"*

*},*

*brand: {*

*bsonType:"string",*

*description: "Brand name cannot be empty and must be a string"*

*},*

*model: {*

*bsonType:"string",*

*description: "Model cannot be empty and must be a string"*

*},*

*range: {*

*bsonType:"int",*

*minimum: 1,*

*description: "Range cannot be empty and must be an integer"*

*},*

*capacity: {*

*bsonType:"object",*

*required: ["business", "economy"],*

*properties: {*

*business: {*

*bsonType:"int",*

*minimum: 0,*

*description: "Business cannot be empty and must be an integer"*

*},*

*economy: {*

*bsonType:"int",*

*minimum: 0,*

*description: "Economy cannot be empty and must be an integer"*

*},*

*},*

*},*

*lengthOfService\_months: {*

*bsonType:"int",*

*description: "Length of service cannot be empty and must be an integer"*

*},*

*remaininglengthOfService\_months: {*

*bsonType:"int",*

*description: "Remaining length of service cannot be empty and must be an integer"*

*},*

*status: {*

*bsonType:"string",*

*description: "Status cannot be empty and must be a string"*

*}*

*}*

*}*

*}*

*})*

Index Creation:

*db.planes.createIndex( { "plane\_id": 1 }, { unique: true } )*

Inserting documents:

*db.planes.insertMany ([*

*{*

*plane\_id : 'a3801',*

*brand : 'airbus',*

*model : 'A380',*

*range : 10000,*

*capacity : {business: 45, economy: 500},*

*lengthOfService\_months : 258,*

*remaininglengthOfService\_months : 102,*

*status : 'repair'*

*},*

*{*

*plane\_id : 'a3802',*

*brand : 'airbus',*

*model : 'A380',*

*range : 10000,*

*capacity : {business: 45, economy: 500},*

*lengthOfService\_months : 100,*

*remaininglengthOfService\_months : 260,*

*status : 'working'*

*},*

*{*

*plane\_id : 'b737M71',*

*brand : 'Boeing',*

*model : '737 MAX 7',*

*range : 7130,*

*capacity : {business: 15, economy: 138},*

*lengthOfService\_months : 110,*

*remaininglengthOfService\_months : 250,*

*status : 'working'*

*},*

*{*

*plane\_id : 'b737M72',*

*brand : 'Boeing',*

*model : '737 MAX 7',*

*range : 7130,*

*capacity : {business: 15, economy: 138},*

*lengthOfService\_months : 290,*

*remaininglengthOfService\_months : 70,*

*status : 'working'*

*},*

*{*

*plane\_id : 'b737M73',*

*brand : 'Boeing',*

*model : '737 MAX 7',*

*range : 7130,*

*capacity : {business: 15, economy: 138},*

*lengthOfService\_months : 300,*

*remaininglengthOfService\_months : 60,*

*status : 'upgrading'*

*},*

*{*

*plane\_id : 'a3301',*

*brand : 'Airbus',*

*model : 'A330',*

*range : 11750,*

*capacity : {business: 40, economy: 400},*

*lengthOfService\_months : 120,*

*remaininglengthOfService\_months : 240,*

*status : 'working'*

*},*

*{*

*plane\_id : 'b787d1',*

*brand : 'Boeing',*

*model : '787 Dreamliner',*

*range : 13530,*

*capacity : {business: 28, economy: 220},*

*lengthOfService\_months : 90,*

*remaininglengthOfService\_months : 270,*

*status : 'working'*

*},*

*{*

*plane\_id : 'b787d2',*

*brand : 'Boeing',*

*model : '787 Dreamliner',*

*range : 13530,*

*capacity : {business: 28, economy: 220},*

*lengthOfService\_months : 42,*

*remaininglengthOfService\_months : 318,*

*status : 'working'*

*},*

*{*

*plane\_id : 'b787d3',*

*brand : 'Boeing',*

*model : '787 Dreamliner',*

*range : 13530,*

*capacity : {business: 28, economy: 220},*

*lengthOfService\_months : 10,*

*remaininglengthOfService\_months : 350,*

*status : 'working'*

*}*

*])*