Indexes

First Index: CREATE INDEX TakeInd ON Take(Date, PassengerID)

a) **Attributes the query is defined on:** Date, PassengerID
b) **Properties of the index:** Secondary, Unclustered, B+ Tree Index
c) **List of Queries the index will help:** Q1, Q6, Q7, Q8, Q9, Q10, Q12
d) **Why?**

I believe this index will strengthen the performance of my queries as they have SELECT statements in which involve PassengerID. In addition, in almost all of them, Take is joined to other tables, therefore this will decrease response time. Within the queries, PassengerID is being referenced and compared in order to gather information about a passenger. The reason for including date is because in addition to an individual being compared by their PassengerID, there are queries in which use the BETWEEN or LIKE operator in order to select flights of certain dates. Also, I would make it a B+ Tree index as they are efficient for ranges.

Second Index: CREATE INDEX RouteServeInd ON RouteServe(FlightNumber)

a) **Attributes the query is defined on:** FlightNumber
b) **Properties of the index:** Secondary, Unclustered, B+ Tree Index
c) **List of Queries the index will help:** Q5, Q6, Q7, Q8, Q9, Q10, Q11, Q12
d) **Why?**

I have selected this index as it will be easier for the database to find the data relating to a FlightNumber. Specifically, in most of these queries RouteServe is being joined as what we are seeking to find is dealing with the origin and destination of the routes. RouteServe is the relationship in helping us retrieve this data and so if we have an index for the FlightFumbers then the joins and information we want to retrieve based on a FlightNumber will be quicker.

Third Index: CREATE INDEX OperateInd ON Operate(Carrier)

a) **Attributes the query is defined on:** Carrier
b) **Properties of the index:** Secondary, Unclustered, B+ Tree Index
c) **List of Queries the index will help:** Q2, Q3, Q5, Q6, Q10, Q11
d) **Why?**

Using this index, the table Operate will have an ordering on the Carriers. Therefore, when we run queries in which relate to the Carrier, the rest of the information in the Operate table will be retrieved smoother as it follows the carrier attribute. For example, in some of the queries we reference Carrier from Operate to Carrier from Airline to get the Name of the Carrier. Including this index would speed up this process in those queries. In the queries we tend to join Operate for the fact that is contains information about the databases carriers, therefore, if we were to sort the data by the carrier, the database will retrieve this data at in increased pace.

Forth Index: CREATE INDEX OperateInd ON Person(ID,FirstName,LastName)

a) **Attributes the query is defined on:** ID, FirstName, LastName
b) **Properties of the index:** Secondary, Unclustered, Hashing Index
c) **List of Queries the index will help:** Q1, Q7, Q8, Q12

d) Why?

The motivation for this index is the fact that in a couple queries we are required to select certain information to identify a person in the database. Specifically, this is returning a person's ID, first name and last name. This index will decrease response times and accomplish better query performance. This index makes the system run quicker as the information wanting to be returned from these queries is organized in the manner we want to retrieve it. Therefore, the queries become equality statements and for this reason, I went with a hashing index as these hashes are good for equality selections.