user search

{

StartDate:

ReturnDate:

FromLocation:

ToLocation:

SeatAdult:

SeatChildren:

IP:

DeviceID:

RegisteredEmail:

Searchtimestamp:

}

MasterUser

{

RegisterdEmail:

ContactNo:

UserName:

FirstName:

LastName:

BirthDate:

}

MasterAirPlane

{

FlightID:

Crew1:

Crew2:

DepartureTime:

ArrivalTime:

SeatCapacity:

FromLocation:

ToLocation:

Altitude:

Route:

}

FlightBook

{

RegisteredEmail:

FlightID:

DateOfTravel:

SeatsBookedAdult:

SeatsBookedChild:

Class:

PurposeOfTravel:

AmountPaid:

TransactionFormat:

CurrentTimestaamp:

IsCancelled:

}

//solutions

1. How many times user has searched for the flight from X location to Y location in particular date range.

2. if person is searching more often on alaskaair.com for any specific location, we can give recommendation or offers to them using their email and contact number. (If user has not logged in, we can flash personalized offer on the website)

3. Frequent user, offer him the alaskaair points-based credit card or loyalty credits.

4. change the rates of the flight based on-demand and supply based on search history and availability of the seats.

5. FlightBook gives user perspective like if he/she is a frequent traveler, then offer them with deals and free services or upgrades to retain the customer.

5. based on purpose of visit, offers can be recommended to the guests.

//Recommendations/Suggestions:

1.Gather information when user searches for destination where there are no flights, based on the demand/season/occasion add flights to accommodate travel. this can increase the guests using Alaskaair.

2.Suggest the next nearest airport where the prices are cheaper/flights are available to the guests, this can attract more guests. example (if there are no flights from Tampa to Denver, suggest Orlando to Denver)

3.feedback forms can help improve the services.

//Technologies to use:

For the transactional data: Cosmos DB (Mongo DB API)

Loading Data to warehouse: ADF

Transformation and ML algorithms: Databricks

Datawarehouse: Synapse/Snowflake