Yen Pham

Project: Database Design for Airline

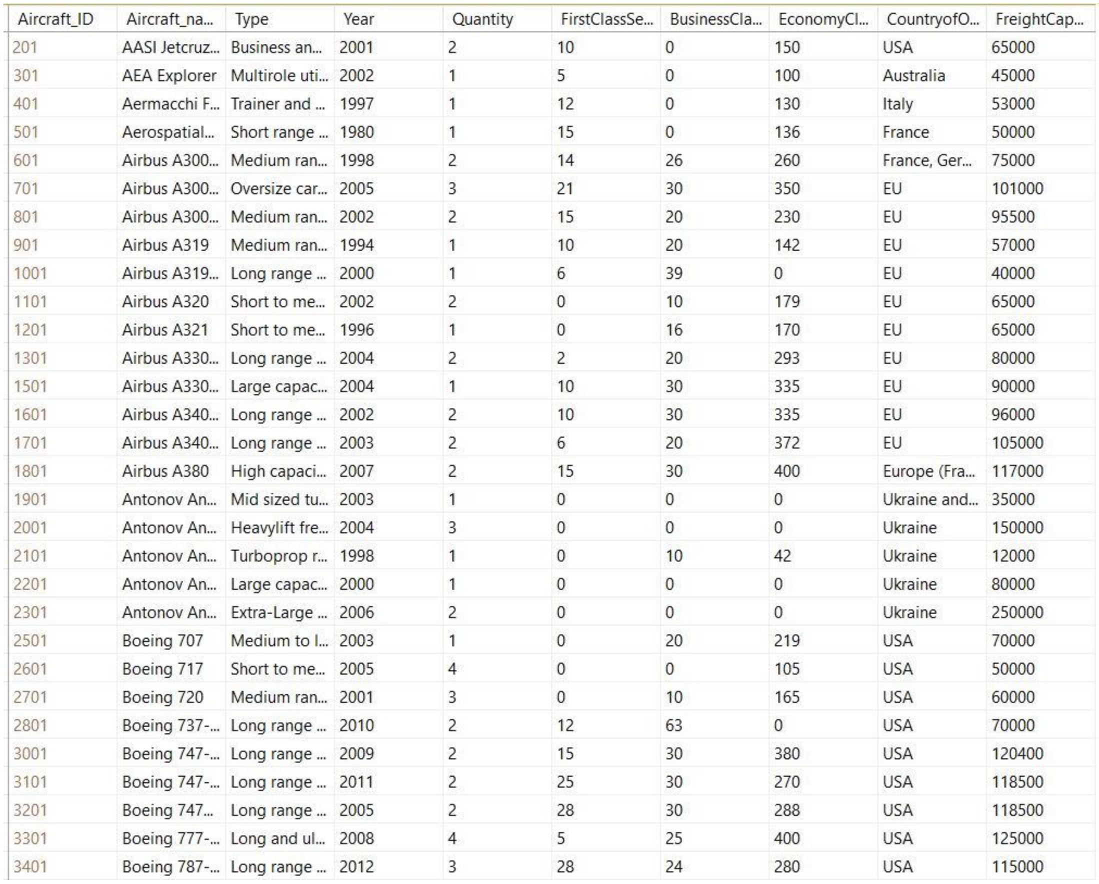
CS 5350

Date: Nov 22, 2020

I. *Data Tables Description*

- Aircraft (Aircraft_ID, Aircraft_Name, Type, Year, Quantity, FirstClassSeats, BusinessClassSeats, EconomyClassSeats, CountryOfOrigins, FreightCapacity)

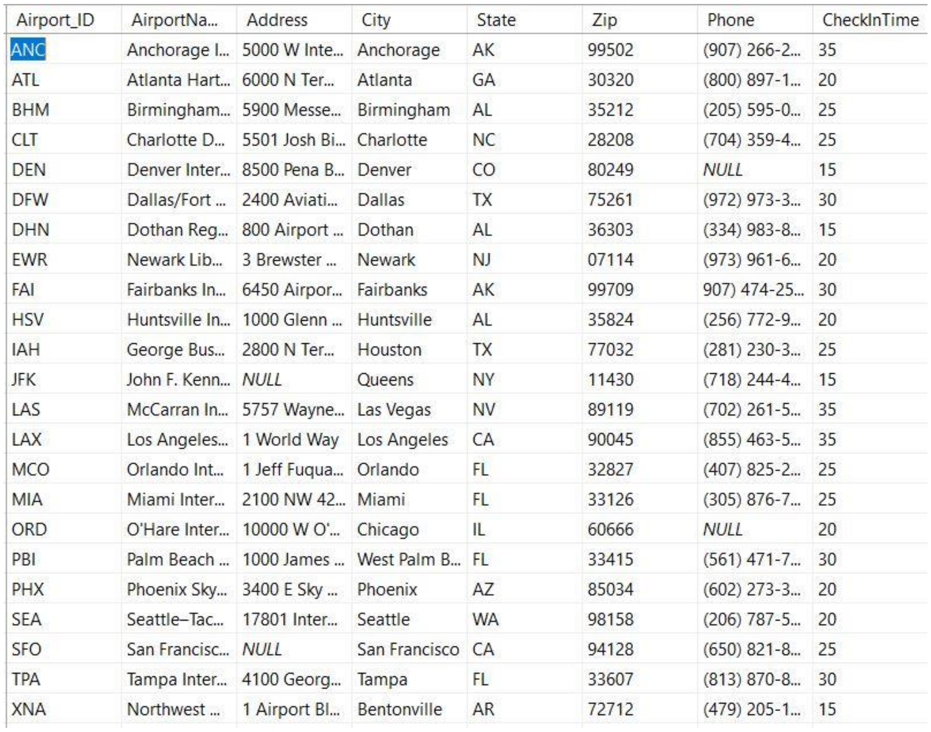
An airline will have a list of bought aircrafts with the maximum number of seats per class (first, business, economy) information. The maximum number of seats will be used to assign seat to each customer whenever they get their boarding pass during check-in and to make sure that the flight is not overbooked.



Aircraft_ID	Aircraft_na...	Type	Year	Quantity	FirstClassSe...	BusinessCla...	EconomyCl...	CountryofO...	FreightCap...
201	AASI Jetcruz...	Business an...	2001	2	10	0	150	USA	65000
301	AEA Explorer	Multirole uti...	2002	1	5	0	100	Australia	45000
401	Aermacchi F...	Trainer and ...	1997	1	12	0	130	Italy	53000
501	Aerospatial...	Short range ...	1980	1	15	0	136	France	50000
601	Airbus A300...	Medium ran...	1998	2	14	26	260	France, Ger...	75000
701	Airbus A300...	Oversize car...	2005	3	21	30	350	EU	101000
801	Airbus A300...	Medium ran...	2002	2	15	20	230	EU	95500
901	Airbus A319	Medium ran...	1994	1	10	20	142	EU	57000
1001	Airbus A319...	Long range ...	2000	1	6	39	0	EU	40000
1101	Airbus A320	Short to me...	2002	2	0	10	179	EU	65000
1201	Airbus A321	Short to me...	1996	1	0	16	170	EU	65000
1301	Airbus A330...	Long range ...	2004	2	2	20	293	EU	80000
1501	Airbus A330...	Large capac...	2004	1	10	30	335	EU	90000
1601	Airbus A340...	Long range ...	2002	2	10	30	335	EU	96000
1701	Airbus A340...	Long range ...	2003	2	6	20	372	EU	105000
1801	Airbus A380	High capaci...	2007	2	15	30	400	Europe (Fra...	117000
1901	Antonov An...	Mid sized tu...	2003	1	0	0	0	Ukraine and...	35000
2001	Antonov An...	Heavylift fre...	2004	3	0	0	0	Ukraine	150000
2101	Antonov An...	Turboprop r...	1998	1	0	10	42	Ukraine	12000
2201	Antonov An...	Large capac...	2000	1	0	0	0	Ukraine	80000
2301	Antonov An...	Extra-Large ...	2006	2	0	0	0	Ukraine	250000
2501	Boeing 707	Medium to l...	2003	1	0	20	219	USA	70000
2601	Boeing 717	Short to me...	2005	4	0	0	105	USA	50000
2701	Boeing 720	Medium ran...	2001	3	0	10	165	USA	60000
2801	Boeing 737-...	Long range ...	2010	2	12	63	0	USA	70000
3001	Boeing 747-...	Long range ...	2009	2	15	30	380	USA	120400
3101	Boeing 747-...	Long range ...	2011	2	25	30	270	USA	118500
3201	Boeing 747...	Long range ...	2005	2	28	30	288	USA	118500
3301	Boeing 777-...	Long and ul...	2008	4	5	25	400	USA	125000
3401	Boeing 787-...	Long range ...	2012	3	28	24	280	USA	115000

- Airport (Airport_ID, AirportName, Address, City, State, Zip, Phone, CheckIn Time)

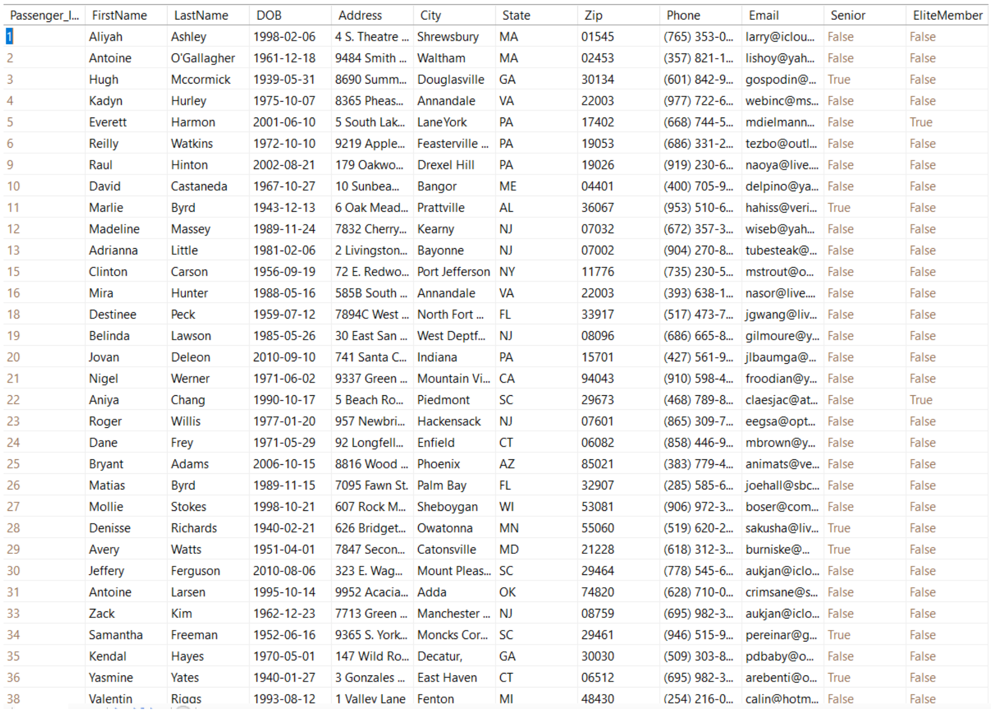
Each airport will have specific minimum boarding time. This time will be used to decide the boarding time of each flight based on the departure time



Airport_ID	AirportNa...	Address	City	State	Zip	Phone	CheckInTime
ANC	Anchorage I...	5000 W Inte...	Anchorage	AK	99502	(907) 266-2...	35
ATL	Atlanta Hart...	6000 N Ter...	Atlanta	GA	30320	(800) 897-1...	20
BHM	Birmingham...	5900 Messe...	Birmingham	AL	35212	(205) 595-0...	25
CLT	Charlotte D...	5501 Josh Bi...	Charlotte	NC	28208	(704) 359-4...	25
DEN	Denver Inter...	8500 Pena B...	Denver	CO	80249	*NULL*	15
DFW	Dallas/Fort ...	2400 Aviati...	Dallas	TX	75261	(972) 973-3...	30
DHN	Dothan Reg...	800 Airport ...	Dothan	AL	36303	(334) 983-8...	15
EWR	Newark Lib...	3 Brewster ...	Newark	NJ	07114	(973) 961-6...	20
FAI	Fairbanks In...	6450 Airpor...	Fairbanks	AK	99709	907) 474-25...	30
HSV	Huntsville In...	1000 Glenn ...	Huntsville	AL	35824	(256) 772-9...	20
IAH	George Bus...	2800 N Ter...	Houston	TX	77032	(281) 230-3...	25
JFK	John F. Kenn...	*NULL*	Queens	NY	11430	(718) 244-4...	15
LAS	McCarran In...	5757 Wayne...	Las Vegas	NV	89119	(702) 261-5...	35
LAX	Los Angeles...	1 World Way	Los Angeles	CA	90045	(855) 463-5...	35
MCO	Orlando Int...	1 Jeff Fuqua...	Orlando	FL	32827	(407) 825-2...	25
MIA	Miami Inter...	2100 NW 42...	Miami	FL	33126	(305) 876-7...	25
ORD	O'Hare Inter...	10000 W O'...	Chicago	IL	60666	*NULL*	20
PBI	Palm Beach ...	1000 James ...	West Palm B...	FL	33415	(561) 471-7...	30
PHX	Phoenix Sky...	3400 E Sky ...	Phoenix	AZ	85034	(602) 273-3...	20
SEA	Seattle–Tac...	17801 Inter...	Seattle	WA	98158	(206) 787-5...	20
SFO	San Francisc...	*NULL*	San Francisco	CA	94128	(650) 821-8...	25
TPA	Tampa Inter...	4100 Georg...	Tampa	FL	33607	(813) 870-8...	30
XNA	Northwest ...	1 Airport Bl...	Bentonville	AR	72712	(479) 205-1...	15

- Customer (Passenger_ID, FirstName, LastName, DOB, Address, City, State, Zip, Phone, Email, Senior, EliteMember)

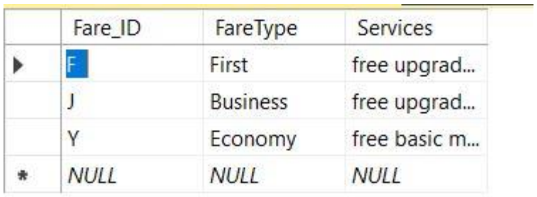
Customer's information is saved only once when they made a first transaction with the airline. People over 65 will get 10% off. If a customer spent >= $1000 in the past 6 months, they will become Elite member who can earn some mileages + benefits for the future flights.



Passenger_I...	FirstName	LastName	DOB	Address	City	State	Zip	Phone	Email	Senior	EliteMember
1	Aliyah	Ashley	1998-02-06	4 S. Theatre ...	Shrewsbury	MA	01545	(765) 353-0...	larry@iclou...	False	False
2	Antoine	O'Gallagher	1961-12-18	9484 Smith ...	Waltham	MA	02453	(357) 821-1...	lishoy@yah...	False	False
3	Hugh	Mccormick	1939-05-31	8690 Summ...	Douglasville	GA	30134	(601) 842-9...	gospodin@...	True	False
4	Kadyn	Hurley	1975-10-07	8365 Pheas...	Annandale	VA	22003	(977) 722-6...	webinc@ms...	False	False
5	Everett	Harmon	2001-06-10	5 South Lak...	LaneYork	PA	17402	(668) 744-5...	mdielmann...	False	True
6	Reilly	Watkins	1972-10-10	9219 Apple...	Feasterville ...	PA	19053	(686) 331-2...	tezbo@outl...	False	False
9	Raul	Hinton	2002-08-21	179 Oakwo...	Drexel Hill	PA	19026	(919) 230-6...	naoya@live....	False	False
10	David	Castaneda	1967-10-27	10 Sunbea...	Bangor	ME	04401	(400) 705-9...	delpino@ya...	False	False
11	Marlie	Byrd	1943-12-13	6 Oak Mead...	Prattville	AL	36067	(953) 510-6...	hahiss@veri...	True	False
12	Madeline	Massey	1989-11-24	7832 Cherry...	Kearny	NJ	07032	(672) 357-3...	wiseb@yah...	False	False
13	Adrianna	Little	1981-02-06	2 Livingston...	Bayonne	NJ	07002	(904) 270-8...	tubesteak@...	False	False
15	Clinton	Carson	1956-09-19	72 E. Redwo...	Port Jefferson	NY	11776	(735) 230-5...	mstrout@o...	False	False
16	Mira	Hunter	1988-05-16	585B South ...	Annandale	VA	22003	(393) 638-1...	nasor@live....	False	False
18	Destinee	Peck	1959-07-12	7894C West ...	North Fort ...	FL	33917	(517) 473-7...	jgwang@liv...	False	False
19	Belinda	Lawson	1985-05-26	30 East San ...	West Deptf...	NJ	08096	(686) 665-8...	gilmoure@y...	False	False
20	Jovan	Deleon	2010-09-10	741 Santa C...	Indiana	PA	15701	(427) 561-9...	jlbaumga@...	False	False
21	Nigel	Werner	1971-06-02	9337 Green ...	Mountain Vi...	CA	94043	(910) 598-4...	froodian@y...	False	False
22	Aniya	Chang	1990-10-17	5 Beach Ro...	Piedmont	SC	29673	(468) 789-8...	claesjac@at...	False	True
23	Roger	Willis	1977-01-20	957 Newbri...	Hackensack	NJ	07601	(865) 309-7...	eegsa@opt...	False	False
24	Dane	Frey	1971-05-29	92 Longfell...	Enfield	CT	06082	(858) 446-9...	mbrown@y...	False	False
25	Bryant	Adams	2006-10-15	8816 Wood ...	Phoenix	AZ	85021	(383) 779-4...	animats@ve...	False	False
26	Matias	Byrd	1989-11-15	7095 Fawn St.	Palm Bay	FL	32907	(285) 585-6...	joehall@sbc...	False	False
27	Mollie	Stokes	1998-10-21	607 Rock M...	Sheboygan	WI	53081	(906) 972-3...	boser@com...	False	False
28	Denisse	Richards	1940-02-21	626 Bridget...	Owatonna	MN	55060	(519) 620-2...	sakusha@liv...	True	False
29	Avery	Watts	1951-04-01	7847 Secon...	Catonsville	MD	21228	(618) 312-3...	burniske@...	True	False
30	Jeffery	Ferguson	2010-08-06	323 E. Wag...	Mount Pleas...	SC	29464	(778) 545-6...	aukjan@iclo...	False	False
31	Antoine	Larsen	1995-10-14	9952 Acacia...	Adda	OK	74820	(628) 710-0...	crimsane@s...	False	False
33	Zack	Kim	1962-12-23	7713 Green ...	Manchester ...	NJ	08759	(695) 982-3...	aukjan@iclo...	False	False
34	Samantha	Freeman	1952-06-16	9365 S. York...	Moncks Cor...	SC	29461	(946) 515-9...	pereinar@g...	True	False
35	Kendal	Hayes	1970-05-01	147 Wild Ro...	Decatur,	GA	30030	(509) 303-8...	pdbaby@o...	False	False
36	Yasmine	Yates	1940-01-27	3 Gonzales ...	East Haven	CT	06512	(695) 982-3...	arebenti@o...	True	False
38	Valentin	Riggs	1993-08-12	1 Valley Lane	Fenton	MI	48430	(254) 216-0...	calin@hotm...	False	False

- Fare (Fare_ID, FareType, Services)

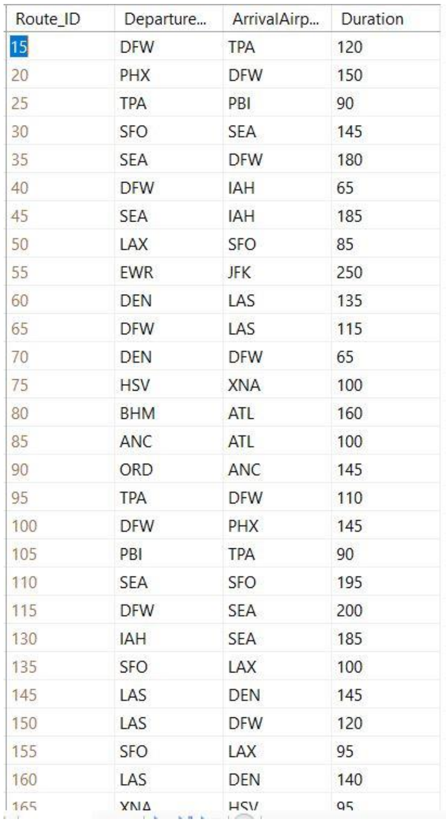
3 fare types: first, business, economy class. Customers are free to choose the class for each flight, no matter if they have transit or not. For example, customer A wants to go from A to D. Because there's no direct route from A to D, she can take one flight from A to B with economy class, then transit to the second flight from C to D with business class.



	Fare_ID	FareType	Services
▶	F	First	free upgrad...
	J	Business	free upgrad...
	Y	Economy	free basic m...
*	NULL	NULL	NULL

- Route (Route_ID, Departure Airport, Arrival Airport, Duration)

The airline will have a list of available non-stop route with time it takes to travel each route. Time to go from A to B in the route list is used to decide arrival time for each flight.



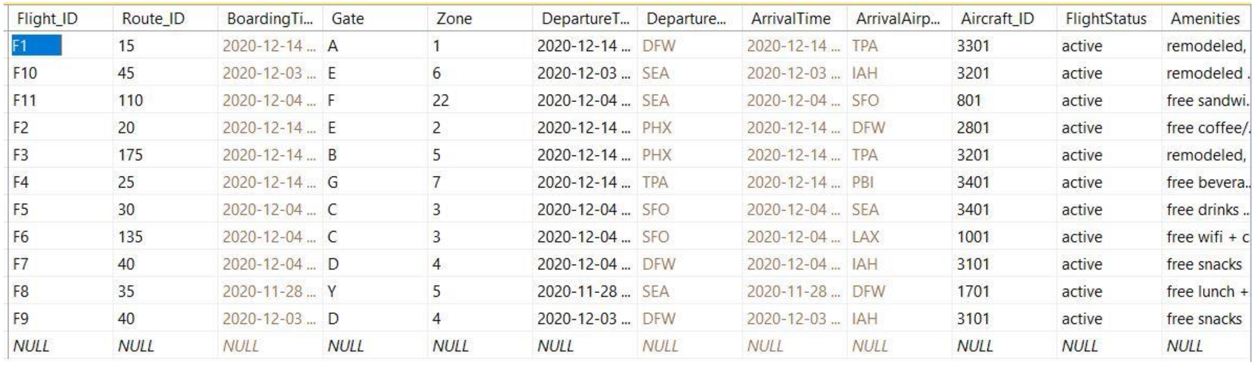
Route_ID	Departure...	ArrivalAirp...	Duration
15	DFW	TPA	120
20	PHX	DFW	150
25	TPA	PBI	90
30	SFO	SEA	145
35	SEA	DFW	180
40	DFW	IAH	65
45	SEA	IAH	185
50	LAX	SFO	85
55	EWR	JFK	250
60	DEN	LAS	135
65	DFW	LAS	115
70	DEN	DFW	65
75	HSV	XNA	100
80	BHM	ATL	160
85	ANC	ATL	100
90	ORD	ANC	145
95	TPA	DFW	110
100	DFW	PHX	145
105	PBI	TPA	90
110	SEA	SFO	195
115	DFW	SEA	200
130	IAH	SEA	185
135	SFO	LAX	100
145	LAS	DEN	145
150	LAS	DFW	120
155	SFO	LAX	95
160	LAS	DEN	140
165	XNA	HSV	95

- Flight (Flight_ID, Route_ID, BoardingTime, Gate, Zone, DepartureTime, DepartureAirport, ArrivalTime, ArrivalAirport, Aircraft_ID, FlightStatus, Amenities)

Boarding Time = DepartureTime – minimum minutes the departure airport requires for boarding

Arrival Time = DepartureTime + time it takes to travel a specific route

Departure Airport and Arrival Airport is decided based on the Route ID input.



Flight_ID	Route_ID	BoardingTi...	Gate	Zone	DepartureT...	Departure...	ArrivalTime	ArrivalAirp...	Aircraft_ID	FlightStatus	Amenities
F1	15	2020-12-14 ...	A	1	2020-12-14 ...	DFW	2020-12-14 ...	TPA	3301	active	remodeled,
F10	45	2020-12-03 ...	E	6	2020-12-03 ...	SEA	2020-12-03 ...	IAH	3201	active	remodeled .
F11	110	2020-12-04 ...	F	22	2020-12-04 ...	SEA	2020-12-04 ...	SFO	801	active	free sandwi.
F2	20	2020-12-14 ...	E	2	2020-12-14 ...	PHX	2020-12-14 ...	DFW	2801	active	free coffee/.
F3	175	2020-12-14 ...	B	5	2020-12-14 ...	PHX	2020-12-14 ...	TPA	3201	active	remodeled,
F4	25	2020-12-14 ...	G	7	2020-12-14 ...	TPA	2020-12-14 ...	PBI	3401	active	free bevera..
F5	30	2020-12-04 ...	C	3	2020-12-04 ...	SFO	2020-12-04 ...	SEA	3401	active	free drinks ..
F6	135	2020-12-04 ...	C	3	2020-12-04 ...	SFO	2020-12-04 ...	LAX	1001	active	free wifi + c
F7	40	2020-12-04 ...	D	4	2020-12-04 ...	DFW	2020-12-04 ...	IAH	3101	active	free snacks
F8	35	2020-11-28 ...	Y	5	2020-11-28 ...	SEA	2020-11-28 ...	DFW	1701	active	free lunch +
F9	40	2020-12-03 ...	D	4	2020-12-03 ...	DFW	2020-12-03 ...	IAH	3101	active	free snacks
NULL	*NULL*	*NULL*	*NULL*	*NULL*	*NULL*	*NULL*	*NULL*	*NULL*	*NULL*	*NULL*	*NULL*

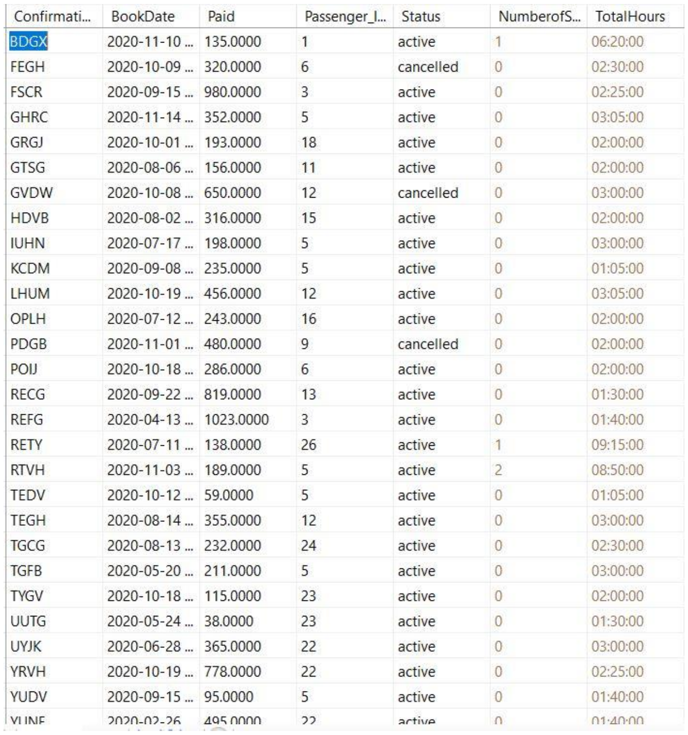
- Itinerary (Confirmation_ID, BookDate, Paid, Status, NumberofStops, TotalHours)
- FlightItinerary (Flight_ID, Itinerary_ID)

Each itinerary's confirmation ID will have multiple flight IDs, since customer may have to transit, so they have to take multiple flights.

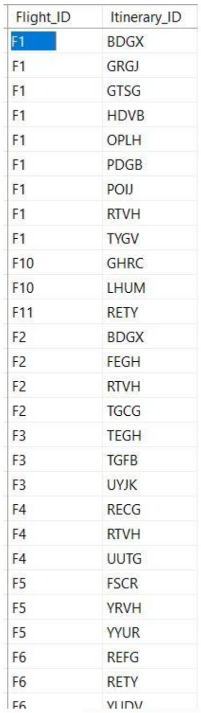
Each flight has multiple bookings, so it has multiple Itinerary_IDs.

TotalHours = time it takes a customer to go from A to B (including layover time if there is any transit).

Book date, payment amount and booking status information will be used to calculate the total amount each customer has spent in the past 6 months (to decide if they qualify for being Elite member or not)



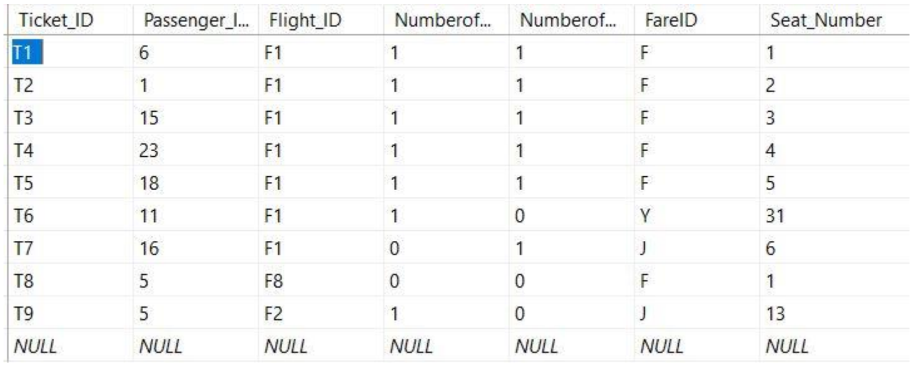
Confirmati...	BookDate	Paid	Passenger_I...	Status	NumberofS...	TotalHours
BDGX	2020-11-10 ...	135.0000	1	active	1	06:20:00
FEGH	2020-10-09 ...	320.0000	6	cancelled	0	02:30:00
FSCR	2020-09-15 ...	980.0000	3	active	0	02:25:00
GHRC	2020-11-14 ...	352.0000	5	active	0	03:05:00
GRGJ	2020-10-01 ...	193.0000	18	active	0	02:00:00
GTSG	2020-08-06 ...	156.0000	11	active	0	02:00:00
GVDW	2020-10-08 ...	650.0000	12	cancelled	0	03:00:00
HDVB	2020-08-02 ...	316.0000	15	active	0	02:00:00
IUHN	2020-07-17 ...	198.0000	5	active	0	03:00:00
KCDM	2020-09-08 ...	235.0000	5	active	0	01:05:00
LHUM	2020-10-19 ...	456.0000	12	active	0	03:05:00
OPLH	2020-07-12 ...	243.0000	16	active	0	02:00:00
PDGB	2020-11-01 ...	480.0000	9	cancelled	0	02:00:00
POIJ	2020-10-18 ...	286.0000	6	active	0	02:00:00
RECG	2020-09-22 ...	819.0000	13	active	0	01:30:00
REFG	2020-04-13 ...	1023.0000	3	active	0	01:40:00
RETY	2020-07-11 ...	138.0000	26	active	1	09:15:00
RTVH	2020-11-03 ...	189.0000	5	active	2	08:50:00
TEDV	2020-10-12 ...	59.0000	5	active	0	01:05:00
TEGH	2020-08-14 ...	355.0000	12	active	0	03:00:00
TGCG	2020-08-13 ...	232.0000	24	active	0	02:30:00
TGFB	2020-05-20 ...	211.0000	5	active	0	03:00:00
TYGV	2020-10-18 ...	115.0000	23	active	0	02:00:00
UUTG	2020-05-24 ...	38.0000	23	active	0	01:30:00
UYJK	2020-06-28 ...	365.0000	22	active	0	03:00:00
YRVH	2020-10-19 ...	778.0000	22	active	0	02:25:00
YUDV	2020-09-15 ...	95.0000	5	active	0	01:40:00
YUNE	2020-02-26	495.0000	22	active	0	01:40:00



Flight_ID	Itinerary_ID
F1	BDGX
F1	GRGJ
F1	GTSG
F1	HDVB
F1	OPLH
F1	PDGB
F1	POIJ
F1	RTVH
F1	TYGV
F10	GHRC
F10	LHUM
F11	RETY
F2	BDGX
F2	FEGH
F2	RTVH
F2	TGCG
F3	TEGH
F3	TGFB
F3	UYJK
F4	RECG
F4	RTVH
F4	UUTG
F5	FSCR
F5	YRVH
F5	YYUR
F6	REFG
F6	RETY
F6	YUDV

- Ticket (Ticket_ID, Passenger_ID, Flight_ID, NumberofCarryOn, NumberofCheckedBag, FareID, Seat_Number)

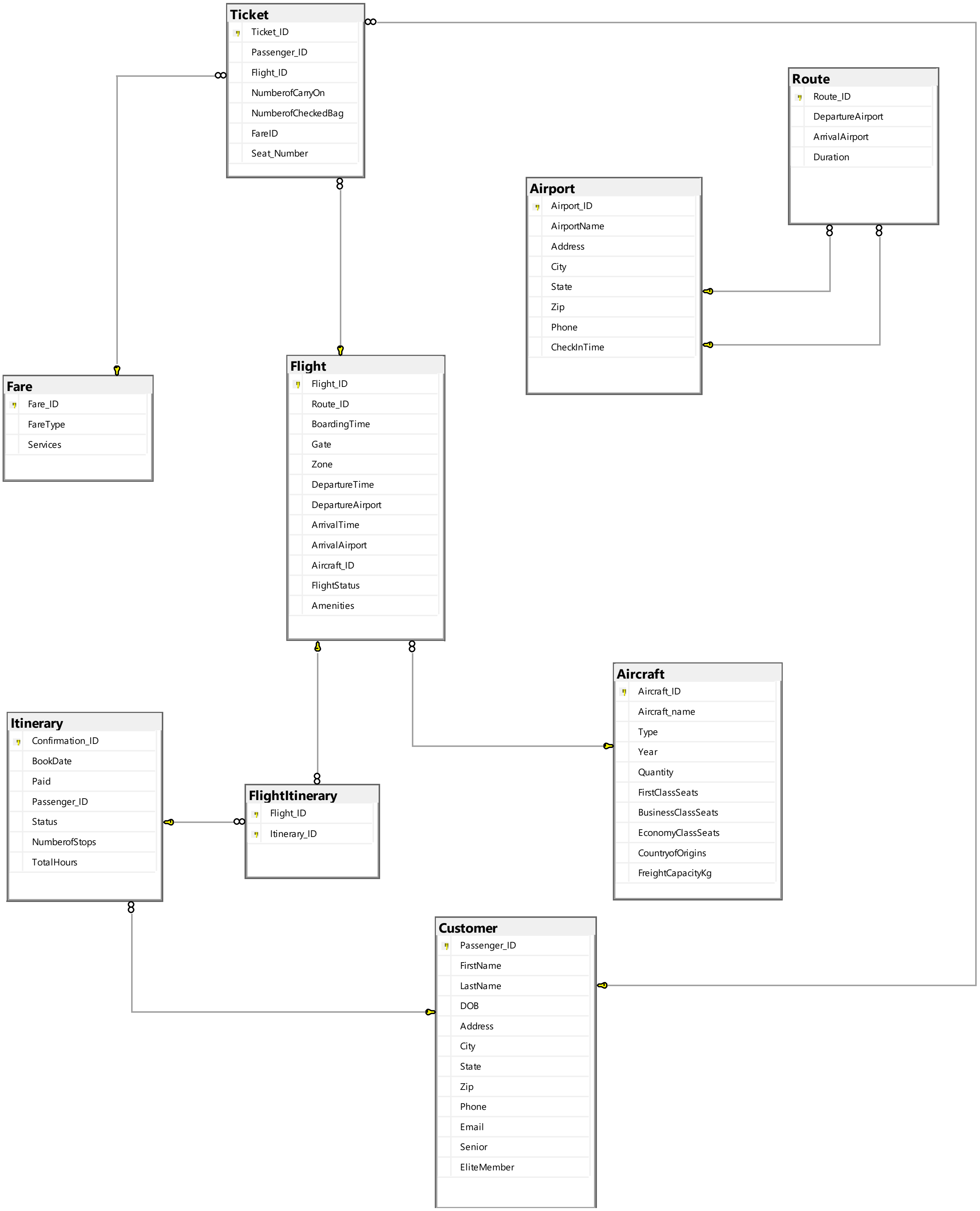
Boarding pass will be issued whenever the customer checks in for each flight, so ticket_ID, number of carry-on and number of checked bags will be given on that day. We set the trigger to check if the passenger really took the flight. If he did, we'll automatically assign the seat to the customers, also using trigger. If the passenger ID was not in the assigned flight or the flight has reach maximum number of seat for the chosen fare type, a trigger will be called with message to prevent the input.



Ticket_ID	Passenger_I...	Flight_ID	Numberof...	Numberof...	FareID	Seat_Number
T1	6	F1	1	1	F	1
T2	1	F1	1	1	F	2
T3	15	F1	1	1	F	3
T4	23	F1	1	1	F	4
T5	18	F1	1	1	F	5
T6	11	F1	1	0	Y	31
T7	16	F1	0	1	J	6
T8	5	F8	0	0	F	1
T9	5	F2	1	0	J	13
NULL	NULL	NULL	NULL	NULL	NULL	NULL

How we manage overbooking: Assuming that customers are not allowed to choose the seat number beforehand and we automatically assign seat number and make sure enough seats are available for the flight. We will reserve the maximum number of first class seats first and mark the seat number from 1 to the maximum number of first class seats. Then, the business class seat number will start from max number of first class seats + 1 to max number of first class seats + max number of business class seats. The economy seat number will start from max number of first class seats + max number of business class seats + 1 to max number of first class seats + max number of business class seats + maximum number of economy seats (total seats of the aircraft). Let say aircraft A has 2 first class seats, 5 business class seats, and 10 economy seats. The first 1st -class ticket for that aircraft will automatically has seat number 1. The first busines class ticket will be assigned seat number 3 no matter if we have any other booking for the second first-class seat or not; the first economy class ticket will be assigned seat number 8 (2+5 =7, 8 is the number right after 7). Then, we assign the seat accordingly to the fare type by incrementing the seat number by 1. Once the maximum number of seats for that fare type is reached, a message/ trigger will pop up if we put one more person in that flight.

II. EER Diagram

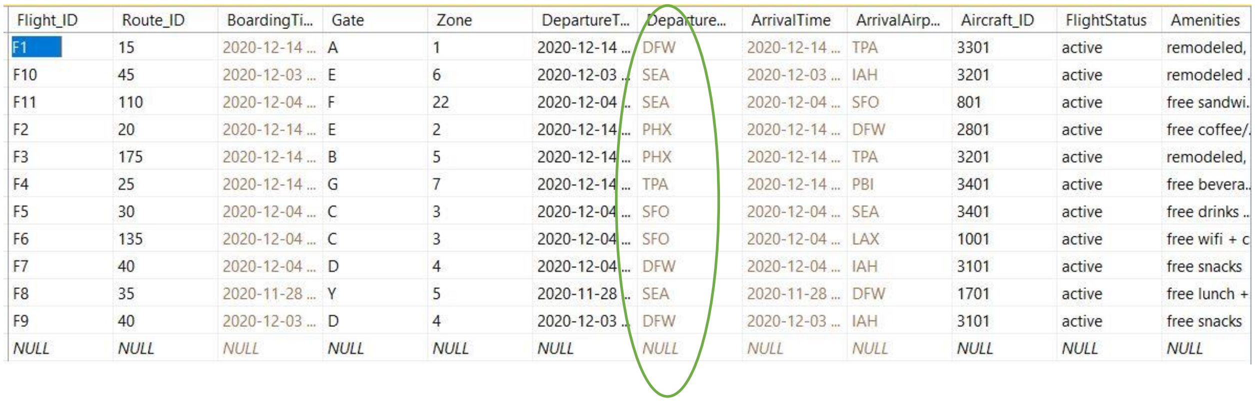


III. Queries

a. Scalar valued functions

- GetDepartureAirport.sql

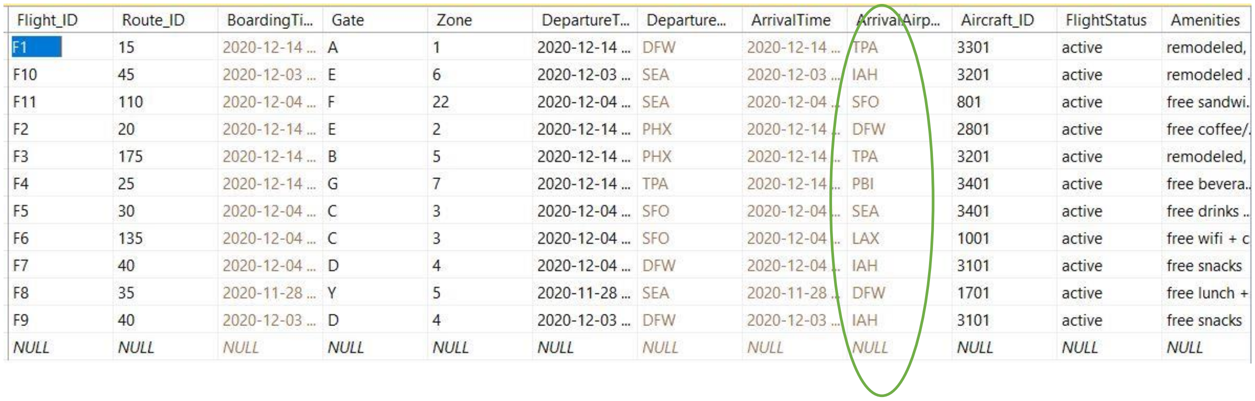
This query took Route ID as a parameter and returns the Departure Airport ID from the Route table to the Flight table. Departure Airport is called directly in the Flight table under DepartureAirport field.



Flight_ID	Route_ID	BoardingTi...	Gate	Zone	DepartureT...	Departure...	ArrivalTime	ArrivalAirp...	Aircraft_ID	FlightStatus	Amenities
F1	15	2020-12-14 ...	A	1	2020-12-14 ...	DFW	2020-12-14 ...	TPA	3301	active	remodeled,
F10	45	2020-12-03 ...	E	6	2020-12-03 ...	SEA	2020-12-03 ...	IAH	3201	active	remodeled .
F11	110	2020-12-04 ...	F	22	2020-12-04 ...	SEA	2020-12-04 ...	SFO	801	active	free sandwi.
F2	20	2020-12-14 ...	E	2	2020-12-14 ...	PHX	2020-12-14 ...	DFW	2801	active	free coffee/.
F3	175	2020-12-14 ...	B	5	2020-12-14 ...	PHX	2020-12-14 ...	TPA	3201	active	remodeled,
F4	25	2020-12-14 ...	G	7	2020-12-14 ...	TPA	2020-12-14 ...	PBI	3401	active	free bevera..
F5	30	2020-12-04 ...	C	3	2020-12-04 ...	SFO	2020-12-04 ...	SEA	3401	active	free drinks ..
F6	135	2020-12-04 ...	C	3	2020-12-04 ...	SFO	2020-12-04 ...	LAX	1001	active	free wifi + c
F7	40	2020-12-04 ...	D	4	2020-12-04 ...	DFW	2020-12-04 ...	IAH	3101	active	free snacks
F8	35	2020-11-28 ...	Y	5	2020-11-28 ...	SEA	2020-11-28 ...	DFW	1701	active	free lunch +
F9	40	2020-12-03 ...	D	4	2020-12-03 ...	DFW	2020-12-03 ...	IAH	3101	active	free snacks
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

- GetArrivalAirport.sql

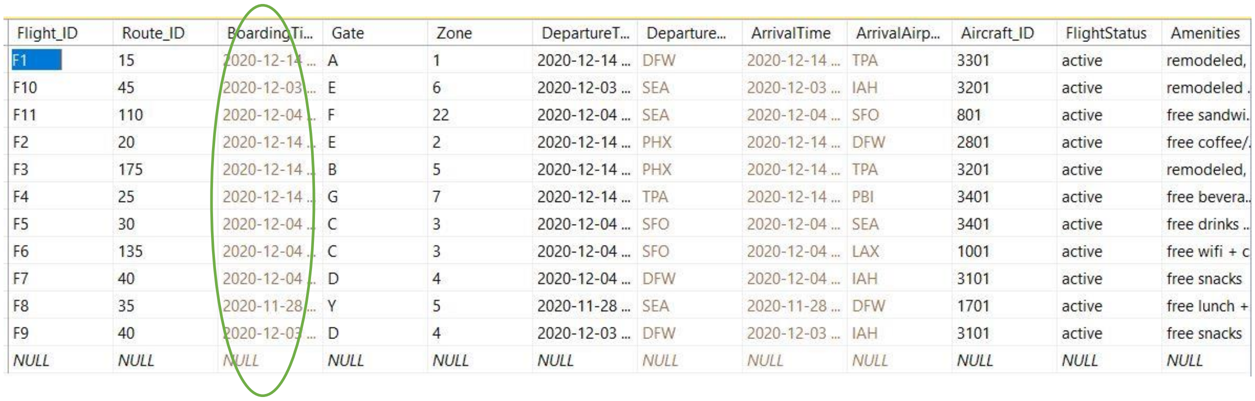
This query took Route ID as a parameter and returns the Arrival Airport ID from the Route table to the Flight table. Arrival Airport is called directly in the Flight table under ArrivalAirport field.



Flight_ID	Route_ID	BoardingTi...	Gate	Zone	DepartureT...	Departure...	ArrivalTime	ArrivalAirp...	Aircraft_ID	FlightStatus	Amenities
F1	15	2020-12-14 ...	A	1	2020-12-14 ...	DFW	2020-12-14 ...	TPA	3301	active	remodeled,
F10	45	2020-12-03 ...	E	6	2020-12-03 ...	SEA	2020-12-03 ...	IAH	3201	active	remodeled .
F11	110	2020-12-04 ...	F	22	2020-12-04 ...	SEA	2020-12-04 ...	SFO	801	active	free sandwi.
F2	20	2020-12-14 ...	E	2	2020-12-14 ...	PHX	2020-12-14 ...	DFW	2801	active	free coffee/.
F3	175	2020-12-14 ...	B	5	2020-12-14 ...	PHX	2020-12-14 ...	TPA	3201	active	remodeled,
F4	25	2020-12-14 ...	G	7	2020-12-14 ...	TPA	2020-12-14 ...	PBI	3401	active	free bevera..
F5	30	2020-12-04 ...	C	3	2020-12-04 ...	SFO	2020-12-04 ...	SEA	3401	active	free drinks ..
F6	135	2020-12-04 ...	C	3	2020-12-04 ...	SFO	2020-12-04 ...	LAX	1001	active	free wifi + c
F7	40	2020-12-04 ...	D	4	2020-12-04 ...	DFW	2020-12-04 ...	IAH	3101	active	free snacks
F8	35	2020-11-28 ...	Y	5	2020-11-28 ...	SEA	2020-11-28 ...	DFW	1701	active	free lunch +
F9	40	2020-12-03 ...	D	4	2020-12-03 ...	DFW	2020-12-03 ...	IAH	3101	active	free snacks
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

- GetBoardingTime.sql

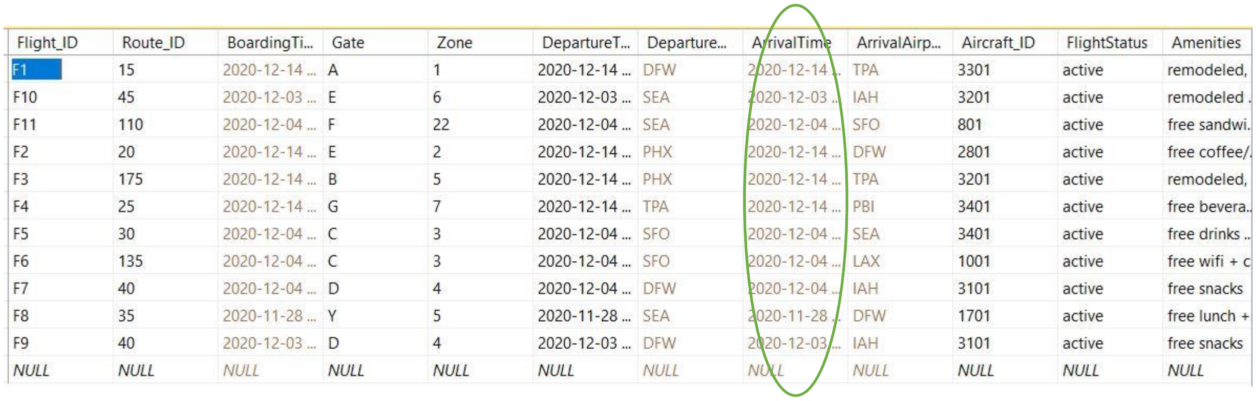
This query took DepartureTime and Airport_ID as parameters and returns the boarding time under the BoardingTime field in the Flight table. Boarding time is calculated based on input departure time – minimum time an airport required to check in provided in the Airport table



Flight_ID	Route_ID	BoardingTi...	Gate	Zone	DepartureT...	Departure...	ArrivalTime	ArrivalAirp...	Aircraft_ID	FlightStatus	Amenities
F1	15	2020-12-14 ...	A	1	2020-12-14 ...	DFW	2020-12-14 ...	TPA	3301	active	remodeled,
F10	45	2020-12-03 ...	E	6	2020-12-03 ...	SEA	2020-12-03 ...	IAH	3201	active	remodeled .
F11	110	2020-12-04 ...	F	22	2020-12-04 ...	SEA	2020-12-04 ...	SFO	801	active	free sandwi.
F2	20	2020-12-14 ...	E	2	2020-12-14 ...	PHX	2020-12-14 ...	DFW	2801	active	free coffee/.
F3	175	2020-12-14 ...	B	5	2020-12-14 ...	PHX	2020-12-14 ...	TPA	3201	active	remodeled,
F4	25	2020-12-14 ...	G	7	2020-12-14 ...	TPA	2020-12-14 ...	PBI	3401	active	free bevera..
F5	30	2020-12-04 ...	C	3	2020-12-04 ...	SFO	2020-12-04 ...	SEA	3401	active	free drinks ..
F6	135	2020-12-04 ...	C	3	2020-12-04 ...	SFO	2020-12-04 ...	LAX	1001	active	free wifi + c
F7	40	2020-12-04 ...	D	4	2020-12-04 ...	DFW	2020-12-04 ...	IAH	3101	active	free snacks
F8	35	2020-11-28 ...	Y	5	2020-11-28 ...	SEA	2020-11-28 ...	DFW	1701	active	free lunch +
F9	40	2020-12-03 ...	D	4	2020-12-03 ...	DFW	2020-12-03 ...	IAH	3101	active	free snacks
NULL	*NULL*	*NULL*	*NULL*	*NULL*	*NULL*	*NULL*	*NULL*	*NULL*	*NULL*	*NULL*	*NULL*

- GetArrivalTime.sql

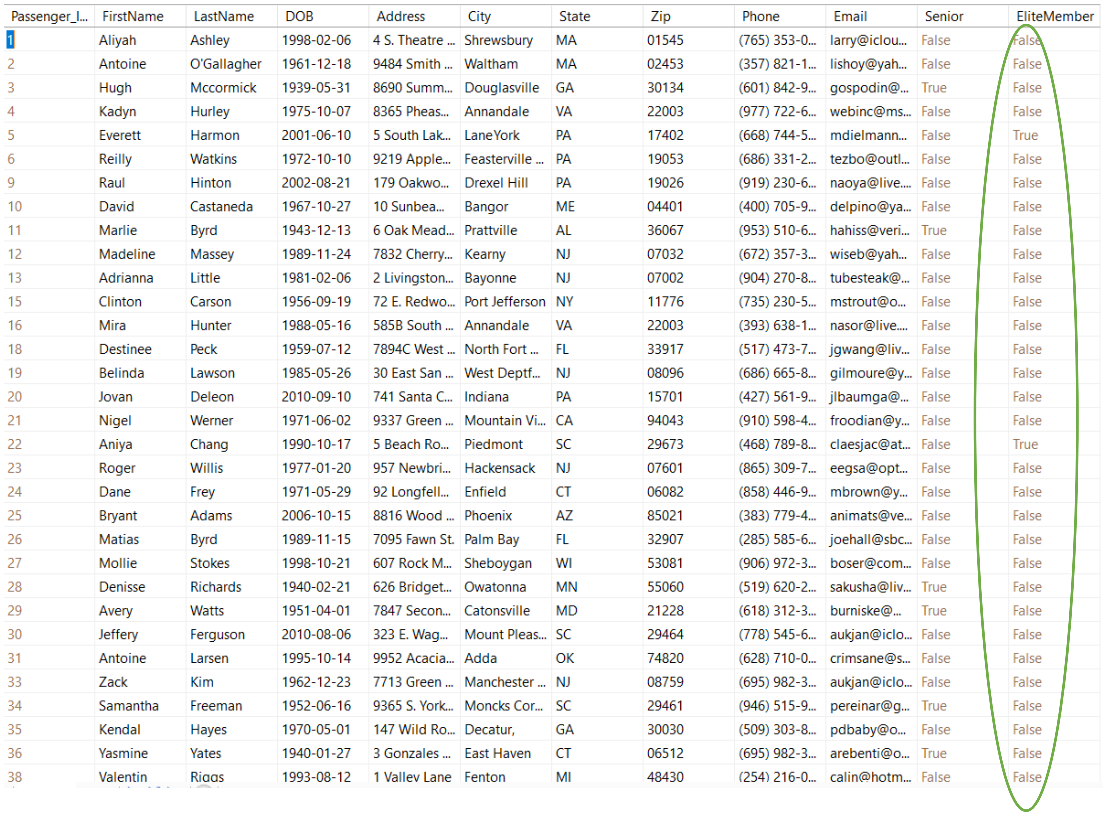
This query took DepartureTime and Route_ID as parameters and returns the arrival time of the flight under ArrivalTime flied in the Flight table. Arrival time is calculated based on the input departure time + time it took a route to travels (provided in the Route table).



Flight_ID	Route_ID	BoardingTi...	Gate	Zone	DepartureT...	Departure...	ArrivalTime	ArrivalAirp...	Aircraft_ID	FlightStatus	Amenities
F1	15	2020-12-14 ...	A	1	2020-12-14 ...	DFW	2020-12-14 ...	TPA	3301	active	remodeled,
F10	45	2020-12-03 ...	E	6	2020-12-03 ...	SEA	2020-12-03 ...	IAH	3201	active	remodeled .
F11	110	2020-12-04 ...	F	22	2020-12-04 ...	SEA	2020-12-04 ...	SFO	801	active	free sandwi.
F2	20	2020-12-14 ...	E	2	2020-12-14 ...	PHX	2020-12-14 ...	DFW	2801	active	free coffee/.
F3	175	2020-12-14 ...	B	5	2020-12-14 ...	PHX	2020-12-14 ...	TPA	3201	active	remodeled,
F4	25	2020-12-14 ...	G	7	2020-12-14 ...	TPA	2020-12-14 ...	PBI	3401	active	free bevera..
F5	30	2020-12-04 ...	C	3	2020-12-04 ...	SFO	2020-12-04 ...	SEA	3401	active	free drinks ..
F6	135	2020-12-04 ...	C	3	2020-12-04 ...	SFO	2020-12-04 ...	LAX	1001	active	free wifi + c
F7	40	2020-12-04 ...	D	4	2020-12-04 ...	DFW	2020-12-04 ...	IAH	3101	active	free snacks
F8	35	2020-11-28 ...	Y	5	2020-11-28 ...	SEA	2020-11-28 ...	DFW	1701	active	free lunch +
F9	40	2020-12-03 ...	D	4	2020-12-03 ...	DFW	2020-12-03 ...	IAH	3101	active	free snacks
NULL	*NULL*	*NULL*	*NULL*	*NULL*	*NULL*	*NULL*	*NULL*	*NULL*	*NULL*	*NULL*	*NULL*

- ClassifyCustomer.sql

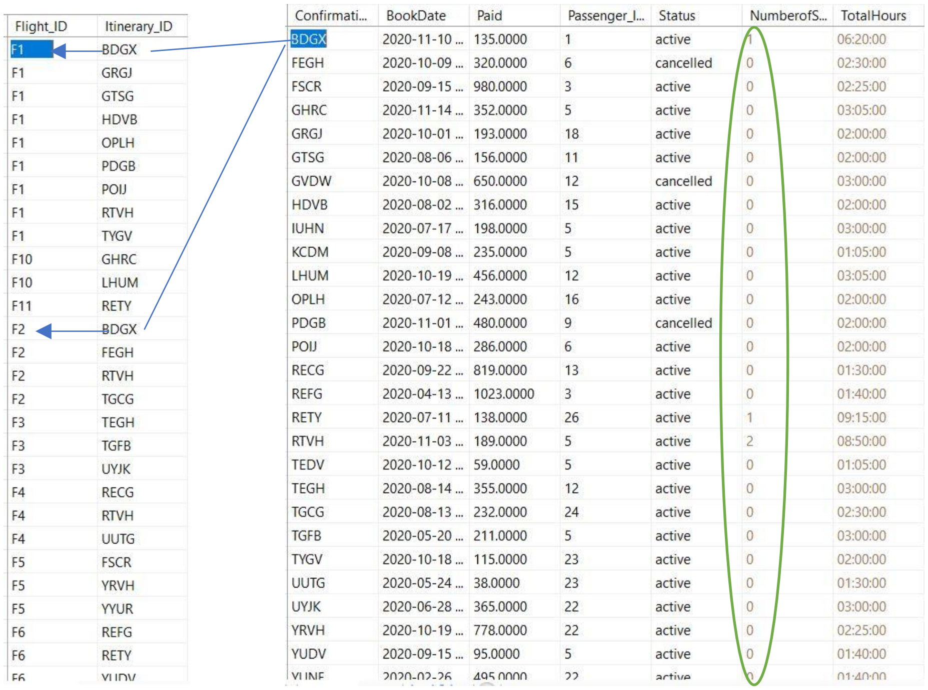
This query should be called *after* the query Get6MonthTransactions.sql from the Table valued functions folder is executed. The query took the customer ID as parameter and returns the bit value (True or False) of whether the customer is qualified for the Elite Membership program in the Customer table under EliteMember field. To decide membership, the query calculated the total successful payment a customer has made in the past 6 months from the temporary table calling from the Get6MonthTransactions.sql query. Whoever has paid over $1000 in the past 6 months will get qualified.



Passenger_I...	FirstName	LastName	DOB	Address	City	State	Zip	Phone	Email	Senior	EliteMember
1	Aliyah	Ashley	1998-02-06	4 S. Theatre ...	Shrewsbury	MA	01545	(765) 353-0...	larry@iclou...	False	False
2	Antoine	O'Gallagher	1961-12-18	9484 Smith ...	Waltham	MA	02453	(357) 821-1...	lishoy@yah...	False	False
3	Hugh	Mccormick	1939-05-31	8690 Summ...	Douglasville	GA	30134	(601) 842-9...	gospodin@...	True	False
4	Kadyn	Hurley	1975-10-07	8365 Pheas...	Annandale	VA	22003	(977) 722-6...	webinc@ms...	False	False
5	Everett	Harmon	2001-06-10	5 South Lak...	LaneYork	PA	17402	(668) 744-5...	mdielmann...	False	True
6	Reilly	Watkins	1972-10-10	9219 Apple...	Feasterville ...	PA	19053	(686) 331-2...	tezbo@outl...	False	False
9	Raul	Hinton	2002-08-21	179 Oakwo...	Drexel Hill	PA	19026	(919) 230-6...	naoya@live....	False	False
10	David	Castaneda	1967-10-27	10 Sunbea...	Bangor	ME	04401	(400) 705-9...	delpino@ya...	False	False
11	Marlie	Byrd	1943-12-13	6 Oak Mead...	Prattville	AL	36067	(953) 510-6...	hahiss@veri...	True	False
12	Madeline	Massey	1989-11-24	7832 Cherry...	Kearny	NJ	07032	(672) 357-3...	wiseb@yah...	False	False
13	Adrianna	Little	1981-02-06	2 Livingston...	Bayonne	NJ	07002	(904) 270-8...	tubesteak@...	False	False
15	Clinton	Carson	1956-09-19	72 E. Redwo...	Port Jefferson	NY	11776	(735) 230-5...	mstrout@o...	False	False
16	Mira	Hunter	1988-05-16	585B South ...	Annandale	VA	22003	(393) 638-1...	nasor@live....	False	False
18	Destinee	Peck	1959-07-12	7894C West ...	North Fort ...	FL	33917	(517) 473-7...	jgwang@liv...	False	False
19	Belinda	Lawson	1985-05-26	30 East San ...	West Deptf...	NJ	08096	(686) 665-8...	gilmoure@y...	False	False
20	Jovan	Deleon	2010-09-10	741 Santa C...	Indiana	PA	15701	(427) 561-9...	jlbaumga@...	False	False
21	Nigel	Werner	1971-06-02	9337 Green ...	Mountain Vi...	CA	94043	(910) 598-4...	froodian@y...	False	False
22	Aniya	Chang	1990-10-17	5 Beach Ro...	Piedmont	SC	29673	(468) 789-8...	claesjac@at...	False	True
23	Roger	Willis	1977-01-20	957 Newbri...	Hackensack	NJ	07601	(865) 309-7...	eegsa@opt...	False	False
24	Dane	Frey	1971-05-29	92 Longfell...	Enfield	CT	06082	(858) 446-9...	mbrown@y...	False	False
25	Bryant	Adams	2006-10-15	8816 Wood ...	Phoenix	AZ	85021	(383) 779-4...	animats@ve...	False	False
26	Matias	Byrd	1989-11-15	7095 Fawn St.	Palm Bay	FL	32907	(285) 585-6...	joehall@sbc...	False	False
27	Mollie	Stokes	1998-10-21	607 Rock M...	Sheboygan	WI	53081	(906) 972-3...	boser@com...	False	False
28	Denisse	Richards	1940-02-21	626 Bridget...	Owatonna	MN	55060	(519) 620-2...	sakusha@liv...	True	False
29	Avery	Watts	1951-04-01	7847 Secon...	Catonsville	MD	21228	(618) 312-3...	burniske@...	True	False
30	Jeffery	Ferguson	2010-08-06	323 E. Wag...	Mount Pleas...	SC	29464	(778) 545-6...	aukjan@iclo...	False	False
31	Antoine	Larsen	1995-10-14	9952 Acacia...	Adda	OK	74820	(628) 710-0...	crimsane@s...	False	False
33	Zack	Kim	1962-12-23	7713 Green ...	Manchester ...	NJ	08759	(695) 982-3...	aukjan@iclo...	False	False
34	Samantha	Freeman	1952-06-16	9365 S. York...	Moncks Cor...	SC	29461	(946) 515-9...	pereinar@g...	True	False
35	Kendal	Hayes	1970-05-01	147 Wild Ro...	Decatur,	GA	30030	(509) 303-8...	pdbaby@o...	False	False
36	Yasmine	Yates	1940-01-27	3 Gonzales ...	East Haven	CT	06512	(695) 982-3...	arebenti@o...	True	False
38	Valentin	Riggs	1993-08-12	1 Valley Lane	Fenton	MI	48430	(254) 216-0...	calin@hotm...	False	False

- GetStopNumber.sql

This query took the itinerary confirmation ID as a parameter and returns the number of stops/ transits per itinerary in the Itinerary table. The number of stops is counted based on the number of flights a confirmation ID has in the FlightItinerary table – 1.

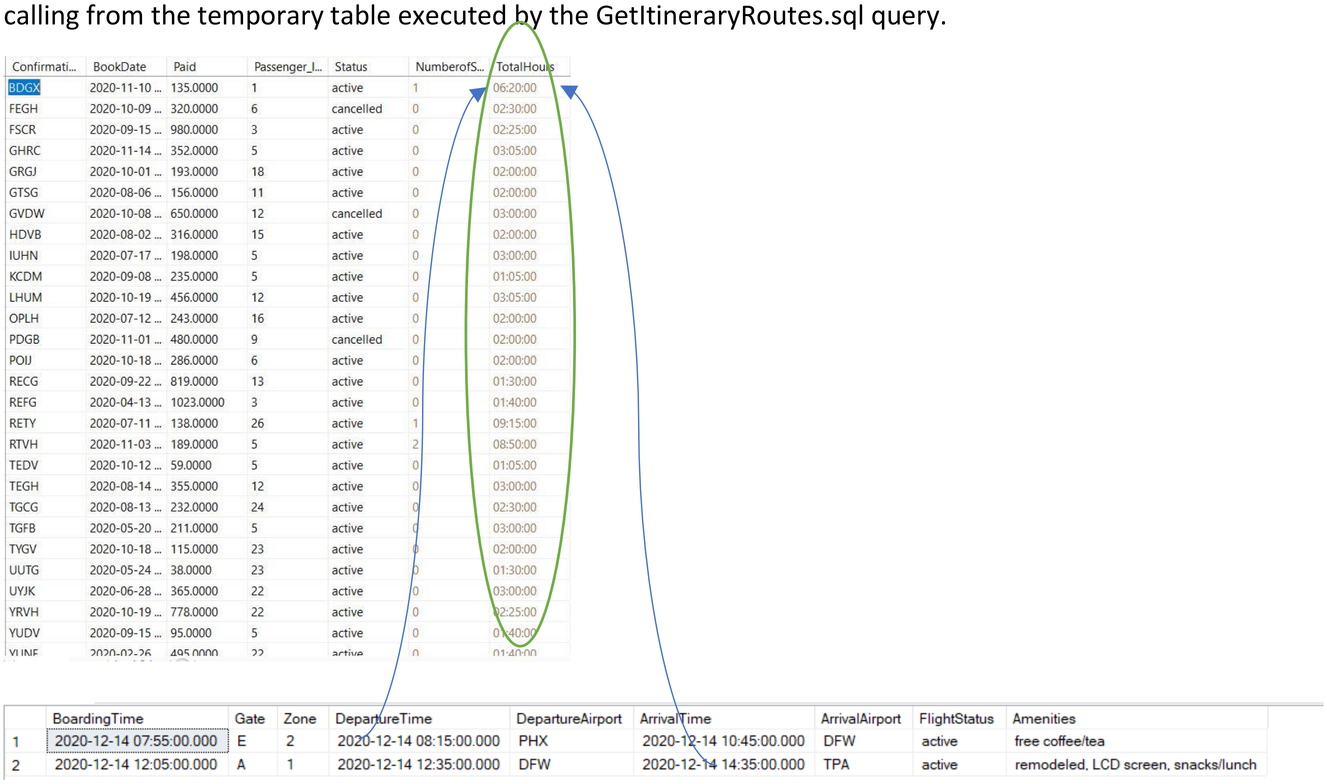


Flight_ID	Itinerary_ID
F1	BDGX
F1	GRGJ
F1	GTSG
F1	HDVB
F1	OPLH
F1	PDGB
F1	POIJ
F1	RTVH
F1	TYGV
F10	GHRC
F10	LHUM
F11	RETY
F2	BDGX
F2	FEGH
F2	RTVH
F2	TGCG
F3	TEGH
F3	TGFB
F3	UYJK
F4	RECG
F4	RTVH
F4	UUTG
F5	FSCR
F5	YRVH
F5	YYUR
F6	REFG
F6	RETY
F6	YUDV

Confirmati...	BookDate	Paid	Passenger_I...	Status	NumberofS...	TotalHours
BDGX	2020-11-10 ...	135.0000	1	active	1	06:20:00
FEGH	2020-10-09 ...	320.0000	6	cancelled	0	02:30:00
FSCR	2020-09-15 ...	980.0000	3	active	0	02:25:00
GHRC	2020-11-14 ...	352.0000	5	active	0	03:05:00
GRGJ	2020-10-01 ...	193.0000	18	active	0	02:00:00
GTSG	2020-08-06 ...	156.0000	11	active	0	02:00:00
GVDW	2020-10-08 ...	650.0000	12	cancelled	0	03:00:00
HDVB	2020-08-02 ...	316.0000	15	active	0	02:00:00
IUHN	2020-07-17 ...	198.0000	5	active	0	03:00:00
KCDM	2020-09-08 ...	235.0000	5	active	0	01:05:00
LHUM	2020-10-19 ...	456.0000	12	active	0	03:05:00
OPLH	2020-07-12 ...	243.0000	16	active	0	02:00:00
PDGB	2020-11-01 ...	480.0000	9	cancelled	0	02:00:00
POIJ	2020-10-18 ...	286.0000	6	active	0	02:00:00
RECG	2020-09-22 ...	819.0000	13	active	0	01:30:00
REFG	2020-04-13 ...	1023.0000	3	active	0	01:40:00
RETY	2020-07-11 ...	138.0000	26	active	1	09:15:00
RTVH	2020-11-03 ...	189.0000	5	active	2	08:50:00
TEDV	2020-10-12 ...	59.0000	5	active	0	01:05:00
TEGH	2020-08-14 ...	355.0000	12	active	0	03:00:00
TGCG	2020-08-13 ...	232.0000	24	active	0	02:30:00
TGFB	2020-05-20 ...	211.0000	5	active	0	03:00:00
TYGV	2020-10-18 ...	115.0000	23	active	0	02:00:00
UUTG	2020-05-24 ...	38.0000	23	active	0	01:30:00
UYJK	2020-06-28 ...	365.0000	22	active	0	03:00:00
YRVH	2020-10-19 ...	778.0000	22	active	0	02:25:00
YUDV	2020-09-15 ...	95.0000	5	active	0	01:40:00
YUNF	2020-02-26	495.0000	22	active	0	01:40:00

- GetItineraryTotalHours.sql

This query should be called *after* the query GetItineraryRoutes.sql from the Table valued function folder is executed. The query took the Itinerary_ID (or confirmation_ID) as a parameter and returns the total time it takes a customer to travel with his itinerary, including the transit/ waiting time between their flight(s) if any. The total time is calculated by taking the arrival time of the last flight – the departure time of the first flight calling from the temporary table executed by the GetItineraryRoutes.sql query.



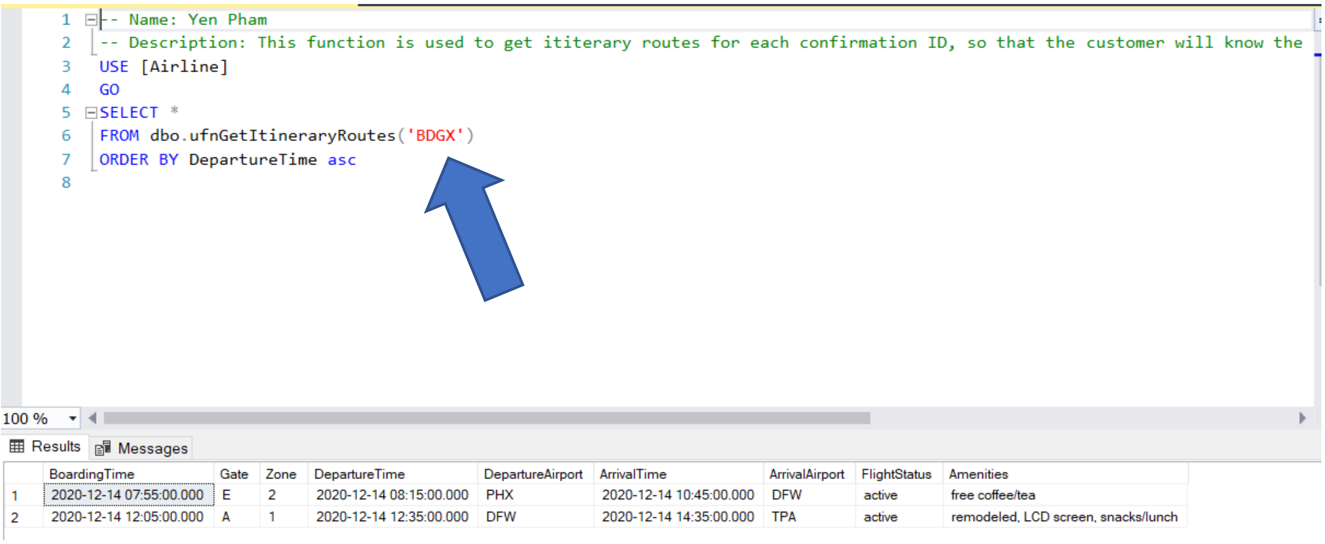
Confirmati...	BookDate	Paid	Passenger_I...	Status	NumberofS...	TotalHours
BDGX	2020-11-10 ...	135.0000	1	active	1	06:20:00
FEGH	2020-10-09 ...	320.0000	6	cancelled	0	02:30:00
FSCR	2020-09-15 ...	980.0000	3	active	0	02:25:00
GHRC	2020-11-14 ...	352.0000	5	active	0	03:05:00
GRGJ	2020-10-01 ...	193.0000	18	active	0	02:00:00
GTSG	2020-08-06 ...	156.0000	11	active	0	02:00:00
GVDW	2020-10-08 ...	650.0000	12	cancelled	0	03:00:00
HDVB	2020-08-02 ...	316.0000	15	active	0	02:00:00
IUHN	2020-07-17 ...	198.0000	5	active	0	03:00:00
KCDM	2020-09-08 ...	235.0000	5	active	0	01:05:00
LHUM	2020-10-19 ...	456.0000	12	active	0	03:05:00
OPLH	2020-07-12 ...	243.0000	16	active	0	02:00:00
PDGB	2020-11-01 ...	480.0000	9	cancelled	0	02:00:00
POIJ	2020-10-18 ...	286.0000	6	active	0	02:00:00
RECG	2020-09-22 ...	819.0000	13	active	0	01:30:00
REFG	2020-04-13 ...	1023.0000	3	active	0	01:40:00
RETY	2020-07-11 ...	138.0000	26	active	1	09:15:00
RTVH	2020-11-03 ...	189.0000	5	active	2	08:50:00
TEDV	2020-10-12 ...	59.0000	5	active	0	01:05:00
TEGH	2020-08-14 ...	355.0000	12	active	0	03:00:00
TGCG	2020-08-13 ...	232.0000	24	active	0	02:30:00
TGFB	2020-05-20 ...	211.0000	5	active	0	03:00:00
TYGV	2020-10-18 ...	115.0000	23	active	0	02:00:00
UUTG	2020-05-24 ...	38.0000	23	active	0	01:30:00
UYJK	2020-06-28 ...	365.0000	22	active	0	03:00:00
YRVH	2020-10-19 ...	778.0000	22	active	0	02:25:00
YUDV	2020-09-15 ...	95.0000	5	active	0	01:40:00
YUNF	2020-02-26	495.0000	22	active	0	01:40:00

	BoardingTime	Gate	Zone	DepartureTime	DepartureAirport	ArrivalTime	ArrivalAirport	FlightStatus	Amenities
1	2020-12-14 07:55:00.000	E	2	2020-12-14 08:15:00.000	PHX	2020-12-14 10:45:00.000	DFW	active	free coffee/tea
2	2020-12-14 12:05:00.000	A	1	2020-12-14 12:35:00.000	DFW	2020-12-14 14:35:00.000	TPA	active	remodeled, LCD screen, snacks/lunch

b. Table valued functions

- GetItineraryRoutes.sql and CallItineraryRoutes.sql

The query GetItineraryRoutes.sql took the Itinerary ID (or confirmation ID) as parameter and returns the tuples of the flights within the itinerary. GetItineraryRoutes.sql is used by the function GetItineraryTotalHours.sql from the Scalar valued functions folder. It can also be executed separately by executing the query CallItineraryRoutes.sql and putting the itinerary ID that we want to look at its routes in the desired field (marked in this arrow).

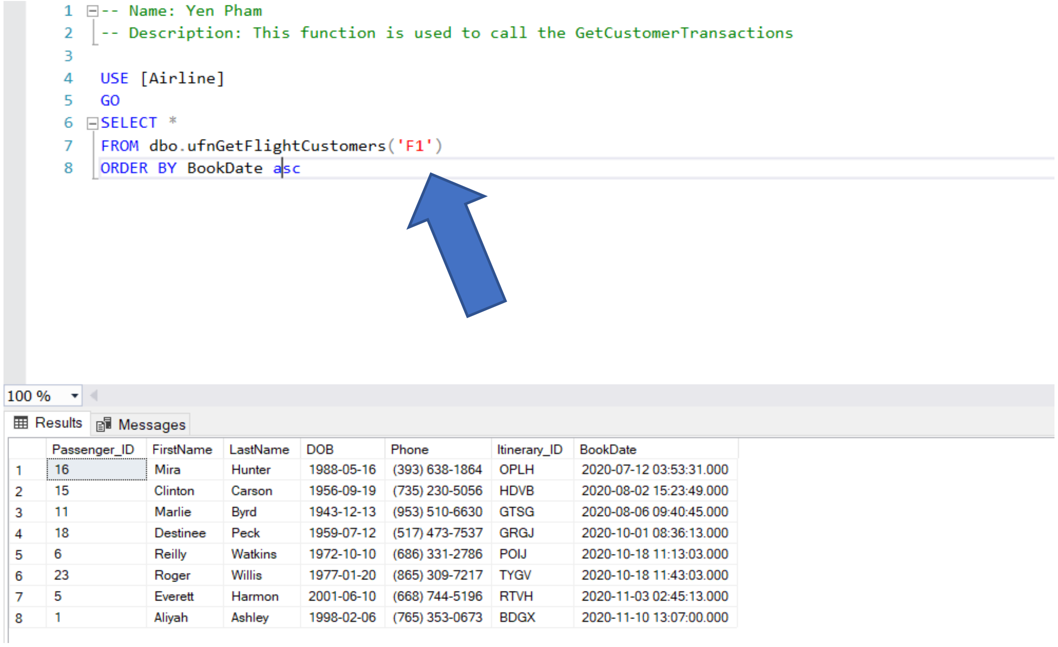


```sql
-- Name: Yen Pham
-- Description: This function is used to get ititerary routes for each confirmation ID, so that the customer will know the
USE [Airline]
GO
SELECT *
FROM dbo.ufnGetItineraryRoutes('BDGX')
ORDER BY DepartureTime asc
```

	BoardingTime	Gate	Zone	DepartureTime	DepartureAirport	ArrivalTime	ArrivalAirport	FlightStatus	Amenities
1	2020-12-14 07:55:00.000	E	2	2020-12-14 08:15:00.000	PHX	2020-12-14 10:45:00.000	DFW	active	free coffee/tea
2	2020-12-14 12:05:00.000	A	1	2020-12-14 12:35:00.000	DFW	2020-12-14 14:35:00.000	TPA	active	remodeled, LCD screen, snacks/lunch

- GetFlightCustomers.sql and CallFlightCustomers.sql

The query GetFlightCustomers.sql took the Flight ID as a parameter and returns the list of customers taking that flight. GetFlightCustomers.sql is used in the Tr_AssignedSeatCheck.sql query to set trigger for the Ticket table (to check if the customer with specific passenger ID really takes the specific flight or not). It can also be called separately by executing the query CallFlightCustomers.sql and putting the flight ID in the desired field (marked in the arrow).



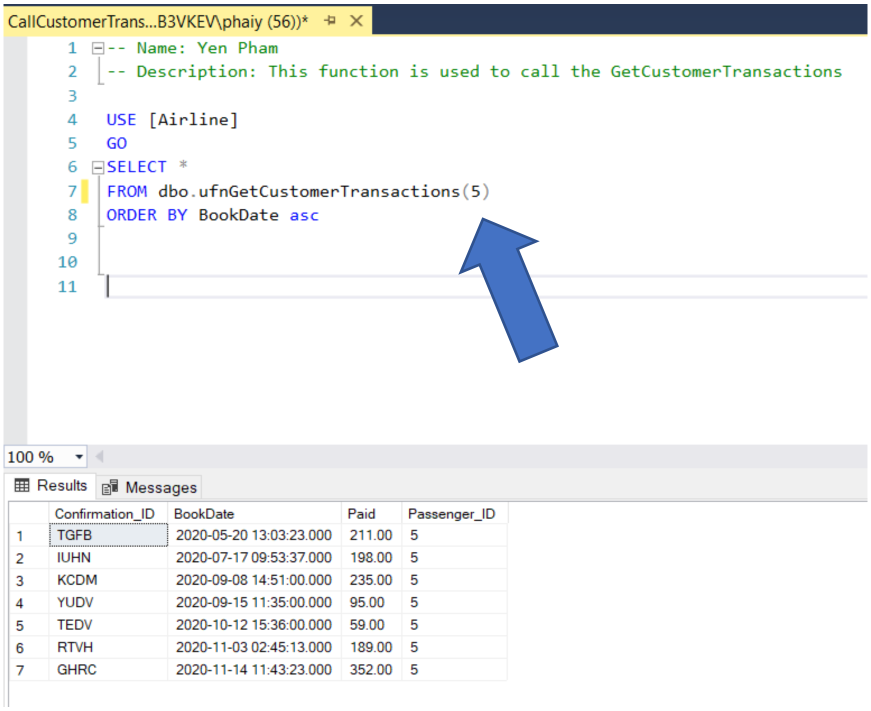
```sql
-- Name: Yen Pham
-- Description: This function is used to call the GetCustomerTransactions

USE [Airline]
GO
SELECT *
FROM dbo.ufnGetFlightCustomers('F1')
ORDER BY BookDate asc
```

	Passenger_ID	FirstName	LastName	DOB	Phone	Itinerary_ID	BookDate
1	16	Mira	Hunter	1988-05-16	(393) 638-1864	OPLH	2020-07-12 03:53:31.000
2	15	Clinton	Carson	1956-09-19	(735) 230-5056	HDVB	2020-08-02 15:23:49.000
3	11	Marlie	Byrd	1943-12-13	(953) 510-6630	GTSG	2020-08-06 09:40:45.000
4	18	Destinee	Peck	1959-07-12	(517) 473-7537	GRGJ	2020-10-01 08:36:13.000
5	6	Reilly	Watkins	1972-10-10	(686) 331-2786	POIJ	2020-10-18 11:13:03.000
6	23	Roger	Willis	1977-01-20	(865) 309-7217	TYGV	2020-10-18 11:43:03.000
7	5	Everett	Harmon	2001-06-10	(668) 744-5196	RTVH	2020-11-03 02:45:13.000
8	1	Aliyah	Ashley	1998-02-06	(765) 353-0673	BDGX	2020-11-10 13:07:00.000

- GetCustomerTransactions.sql and CallCustomerTransactions.sql

The GetCustomerTransactions.sql took Passenger_ID as a parameter and returns the list of customer's transactions since their first join in, including cancelled transactions. The query is called seperately by the CallCustomerTransactions.sql query if we put the passenger ID in the desired field (marked by blue arrow).



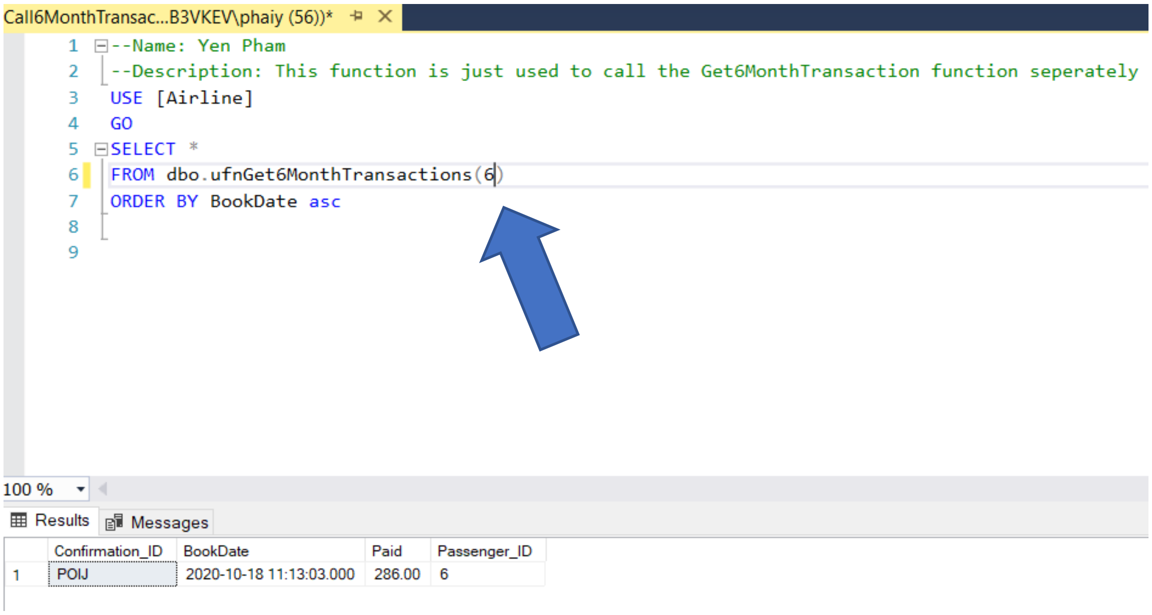
```sql
-- Name: Yen Pham
-- Description: This function is used to call the GetCustomerTransactions

USE [Airline]
GO
SELECT *
FROM dbo.ufnGetCustomerTransactions(5)
ORDER BY BookDate asc
```

	Confirmation_ID	BookDate	Paid	Passenger_ID
1	TGFB	2020-05-20 13:03:23.000	211.00	5
2	IUHN	2020-07-17 09:53:37.000	198.00	5
3	KCDM	2020-09-08 14:51:00.000	235.00	5
4	YUDV	2020-09-15 11:35:00.000	95.00	5
5	TEDV	2020-10-12 15:36:00.000	59.00	5
6	RTVH	2020-11-03 02:45:13.000	189.00	5
7	GHRC	2020-11-14 11:43:23.000	352.00	5

- Get6MonthTransactions.sql and Call6MonthTransactions.sql

The Get6MonthTransactions.sql took Passenger_ID as a parameter and returns the list of customer's successful transactions in the past 6 months, excluding cancelled transactions. The query was used by the ClassifyCustomer.sql query in the scalar valued function for the Customer table to decide whether the customer is qualified for Elite Membership or not. The query can also be called seperately by the Call6MonthTransactions.sql query if we put the passenger ID in the desired field (marked by blue arrow).



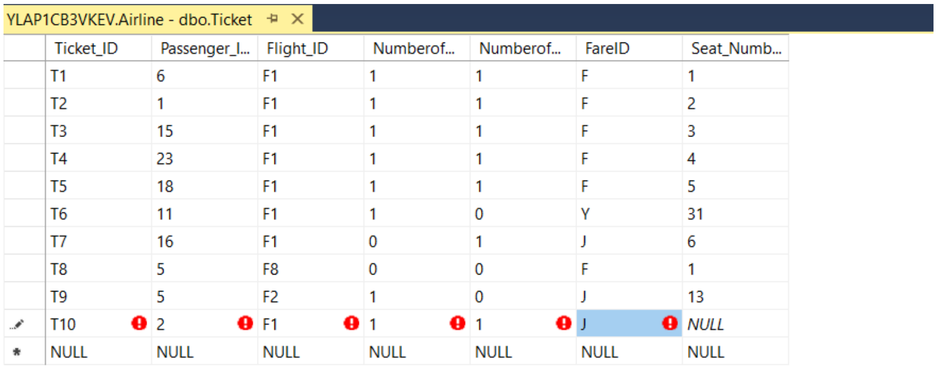
```sql
--Name: Yen Pham
--Description: This function is just used to call the Get6MonthTransaction function seperately
USE [Airline]
GO
SELECT *
FROM dbo.ufnGet6MonthTransactions(6)
ORDER BY BookDate asc
```

	Confirmation_ID	BookDate	Paid	Passenger_ID
1	POIJ	2020-10-18 11:13:03.000	286.00	6

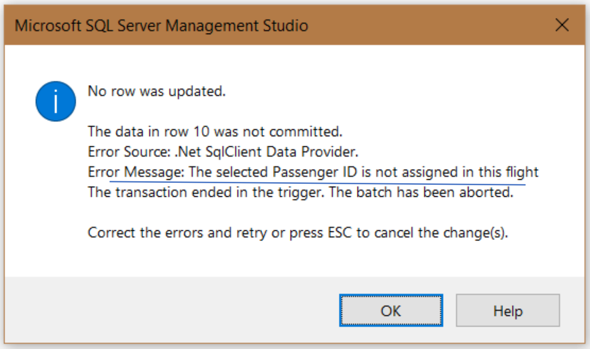
c. ***Triggers***

- Tr_AssignedSeatCheck.sql

This query set up 2 triggers, one is to decide whether the passenger really book the flight or not, and the other is to automatically assign the seat based on their chosen fare type and prevent overbooking. The query took parameter Fare Type, Flight ID, and Passenger ID from the Ticket table, max Number of First Class Seats, maxNumber of Business Class Seats, max Number of Economy class seat of a specific aircraft from the Aircraft table, and MaxAssignedSeat from a virtual temporary Inserted table. It also used the function GetFlightCustomers.sql to check the identity of the customer in the flight. The query is called in the Ticket table whenever we input new ticket information.



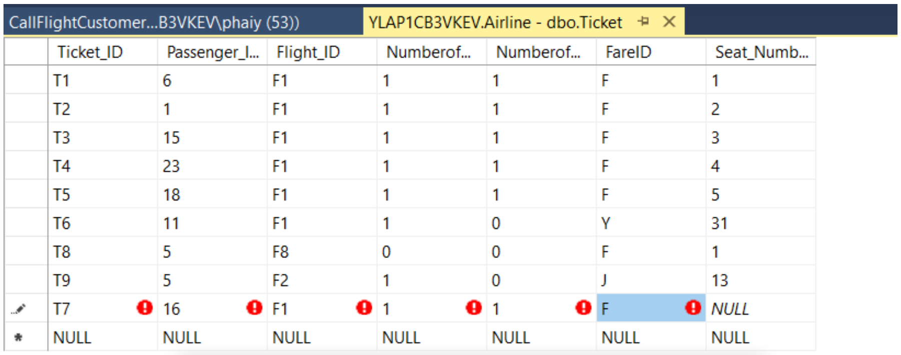
	Ticket_ID	Passenger_I...	Flight_ID	Numberof...	Numberof...	FareID	Seat_Numb...
	T1	6	F1	1	1	F	1
	T2	1	F1	1	1	F	2
	T3	15	F1	1	1	F	3
	T4	23	F1	1	1	F	4
	T5	18	F1	1	1	F	5
	T6	11	F1	1	0	Y	31
	T7	16	F1	0	1	J	6
	T8	5	F8	0	0	F	1
	T9	5	F2	1	0	J	13
✎	T10 ❗	2 ❗	F1 ❗	1 ❗	1 ❗	J ❗	*NULL*
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL



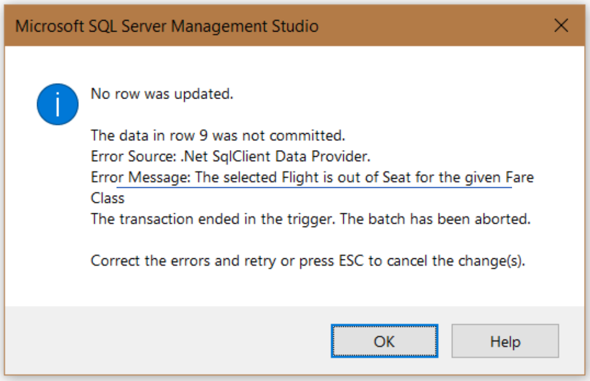
No row was updated.

The data in row 10 was not committed.
Error Source: .Net SqlClient Data Provider.
Error Message: The selected Passenger ID is not assigned in this flight
The transaction ended in the trigger. The batch has been aborted.

Correct the errors and retry or press ESC to cancel the change(s).


CallFlightCustomer...B3VKEV\phaiy (53))
YLAP1CB3VKEV.Airline - dbo.Ticket

	Ticket_ID	Passenger_I...	Flight_ID	Numberof...	Numberof...	FareID	Seat_Numb...
	T1	6	F1	1	1	F	1
	T2	1	F1	1	1	F	2
	T3	15	F1	1	1	F	3
	T4	23	F1	1	1	F	4
	T5	18	F1	1	1	F	5
	T6	11	F1	1	0	Y	31
	T8	5	F8	0	0	F	1
	T9	5	F2	1	0	J	13
✎	T7 ❗	16 ❗	F1 ❗	1 ❗	1 ❗	F ❗	*NULL*
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL



No row was updated.

The data in row 9 was not committed.
Error Source: .Net SqlClient Data Provider.
Error Message: The selected Flight is out of Seat for the given Fare Class
The transaction ended in the trigger. The batch has been aborted.

Correct the errors and retry or press ESC to cancel the change(s).

IV. Run database

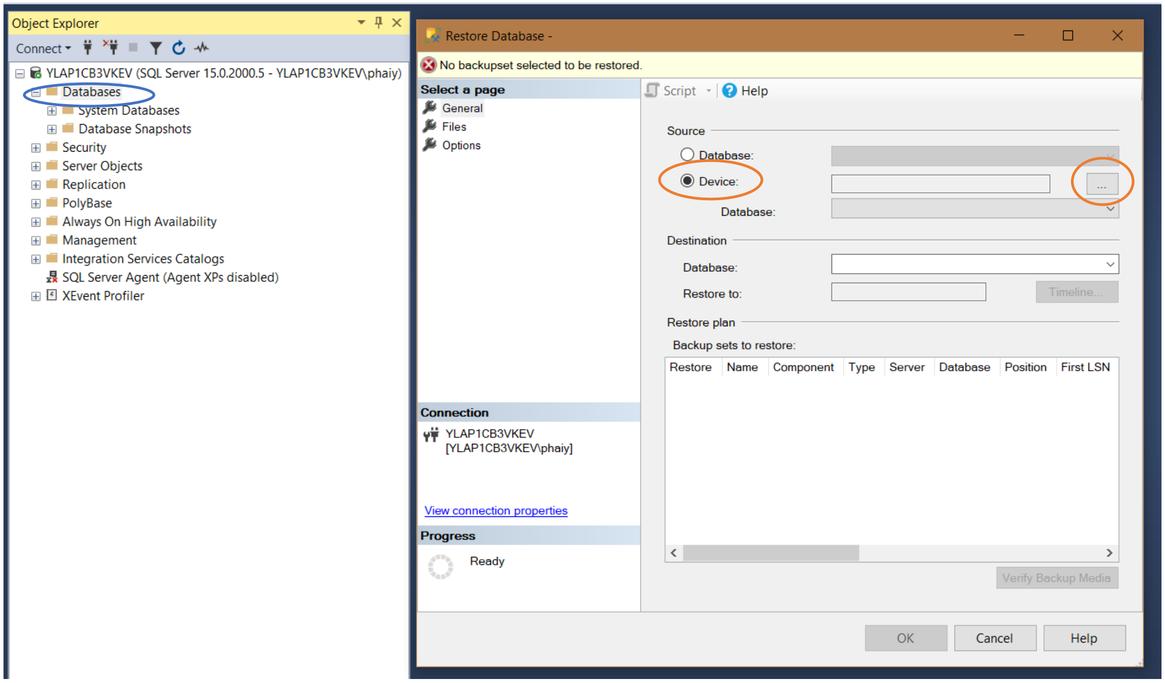
_Install Microsoft SQL Server Management Studio **version 15.0.18338.0** and SQL Server **version 15.0.2000.5.**

_Extract the Airline.zip file.

Option 1: Restore database by using backup file (with all data included).

_ To open Object Explorer: View -> Object Explorer.

_On Object Explorer, right click on Databases -> Restore Database.



Object Explorer

YLAP1CB3VKEV (SQL Server 15.0.2000.5 - YLAP1CB3VKEV\phaiy)

Restore Database -

No backupset selected to be restored.

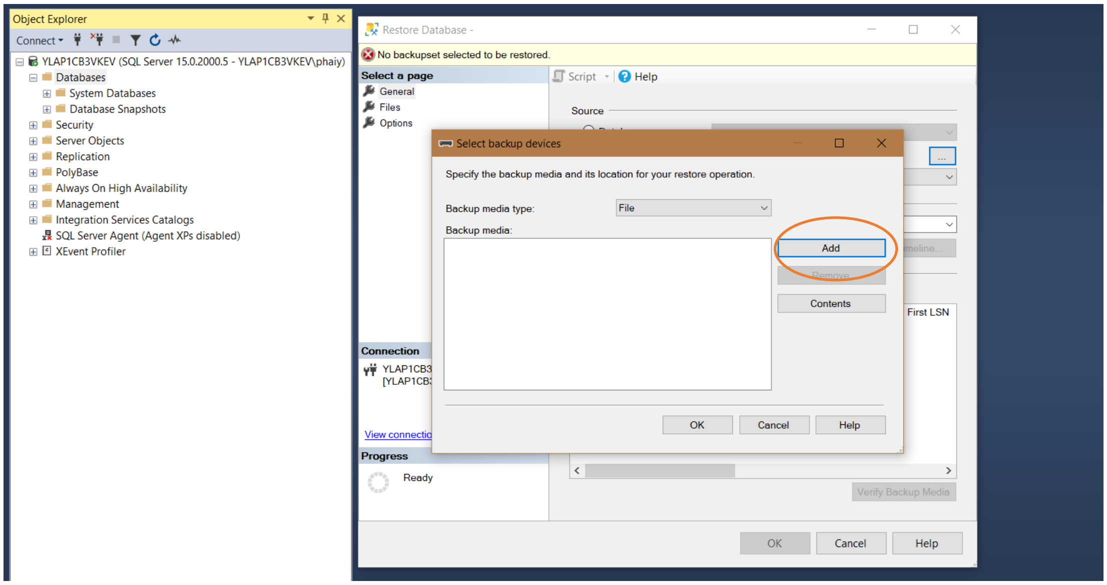
Source: ○ Database: ● Device:

Connection: YLAP1CB3VKEV [YLAP1CB3VKEV\phaiy]

Progress: Ready

_Choose option Device, then click on button …

_Click Add



Select backup devices

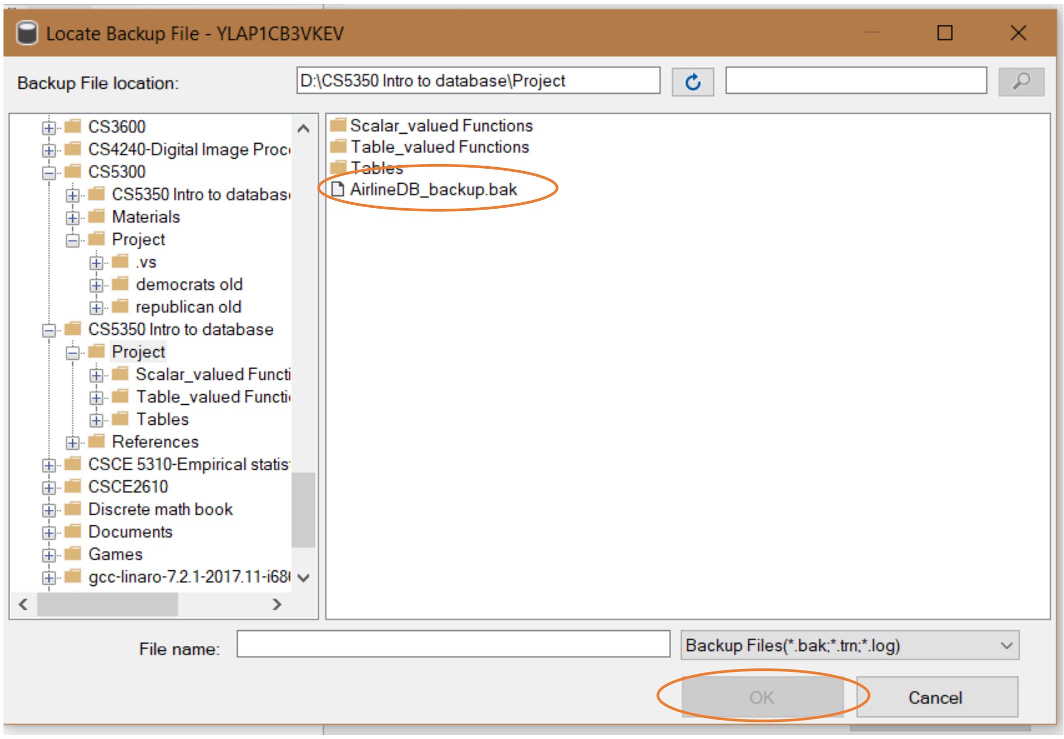
Specify the backup media and its location for your restore operation.

Backup media type: File

No backupset selected to be restored.

Progress: Ready

_Go to directory where the extracted Airline folder is located, choose the file AirlineDB_backup.bak, click OK.



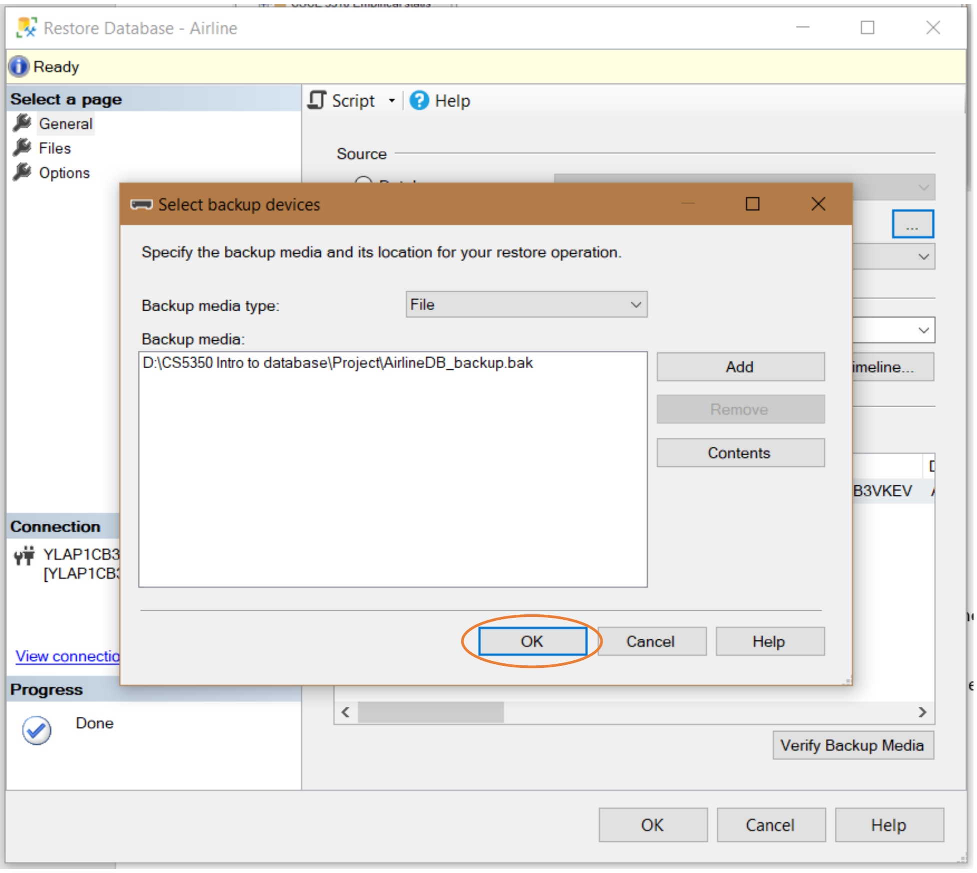
Locate Backup File - YLAP1CB3VKEV

Backup File location: D:\CS5350 Intro to database\Project

File name:

Backup Files(*.bak;*.trn;*.log)

_Then click OK again on the Select backup devices box.



Restore Database - Airline

Ready

Select backup devices

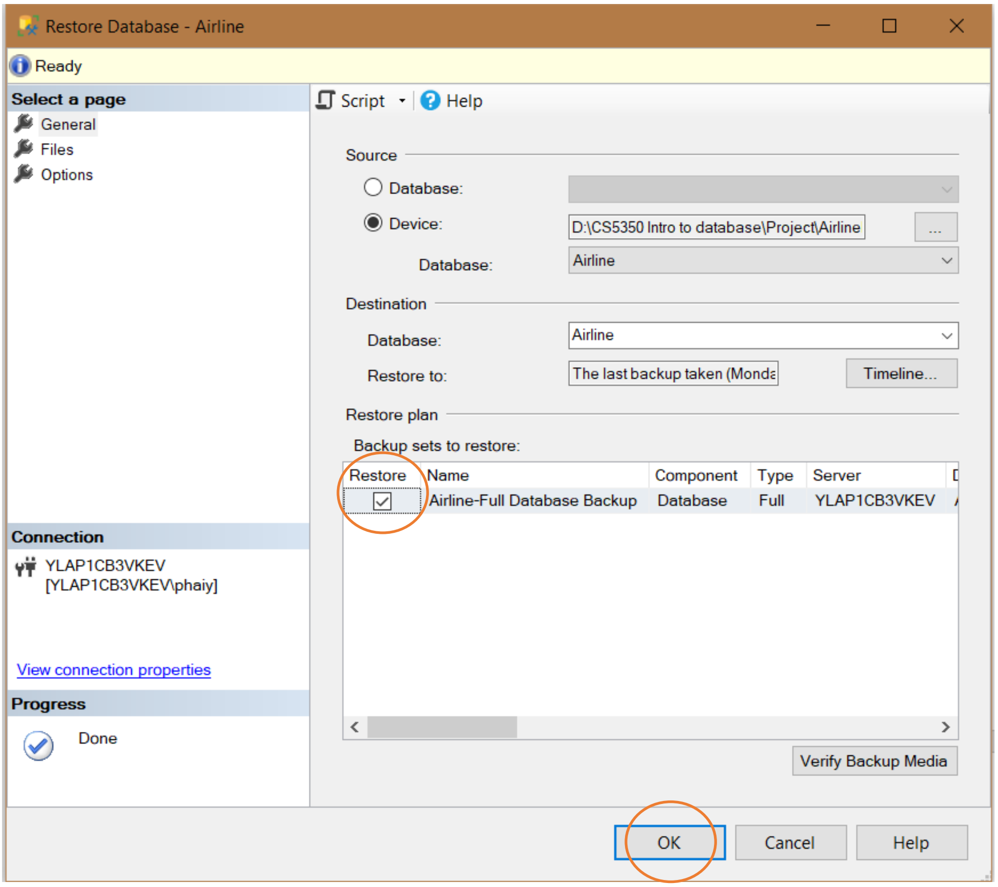
Specify the backup media and its location for your restore operation.

Backup media type: File

Backup media:

D:\CS5350 Intro to database\Project\AirlineDB_backup.bak

_On Restore – Database Airline box, choose the file and click OK.



Restore Database - Airline

Ready

Select a page
- General
- Files
- Options

Script · Help

Source

○ Database:

● Device: D:\CS5350 Intro to database\Project\Airline

Database: Airline

Destination

Database: Airline

Restore to: The last backup taken (Monda

Timeline...

Restore plan

Backup sets to restore:

Restore	Name	Component	Type	Server
☑	Airline-Full Database Backup	Database	Full	YLAP1CB3VKEV

Connection

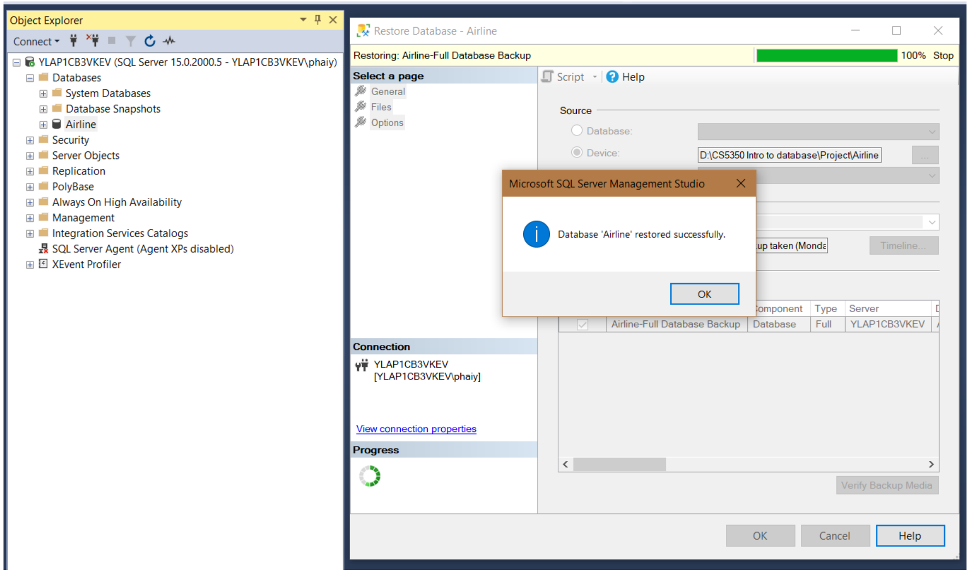
YLAP1CB3VKEV
[YLAP1CB3VKEV\phaiy]

View connection properties

Progress

Done

_All database will be restored when the message Database ‘Airline’ restored successfully shows up.



Object Explorer

YLAP1CB3VKEV (SQL Server 15.0.2000.5 - YLAP1CB3VKEV\phaiy)
- Databases
  - System Databases
  - Database Snapshots
  - Airline
- Security
- Server Objects
- Replication
- PolyBase
- Always On High Availability
- Management
- Integration Services Catalogs
- SQL Server Agent (Agent XPs disabled)
- XEvent Profiler

Restore Database - Airline

Restoring: Airline-Full Database Backup 100% Stop

Microsoft SQL Server Management Studio

Database 'Airline' restored successfully.

Option 2: Create database using SQL queries (the database will not include data – not recommended)

_Open the file Airline.sql on Microsoft SQL Management Server, click Execute.

_Open each sql files in the Tables folder, click execute:

- Aircraft.sql

- Airport.sql
- Customer.sql
- Fare.sql
- Route.sql
- Flight.sql
- Itinerary.sql
- FlightItinerary.sql
- Ticket.sql

_Some queries were mentioned in some other queries, you need to execute the mentioned query first before executing the rest (check the Queries section in this paper).

_Open each file in the Table valued Functions folder and execute only these following queries:

- GetItineraryRoutes.sql
- GetFlightCustomers.sql
- GetCustomerTransactions.sql
- Get6MonthTransactions.sql

_Open the Scalar valued Functions folder and execute these following queries:

- GetDepartureAirport.sql
- GetArrivalAirport.sql
- GetBoardingTime.sql
- GetArrivalTime.sql
- GetStopNumber.sql
- ClassifyCustomer.sql
- GetItineraryTotalHours.sql

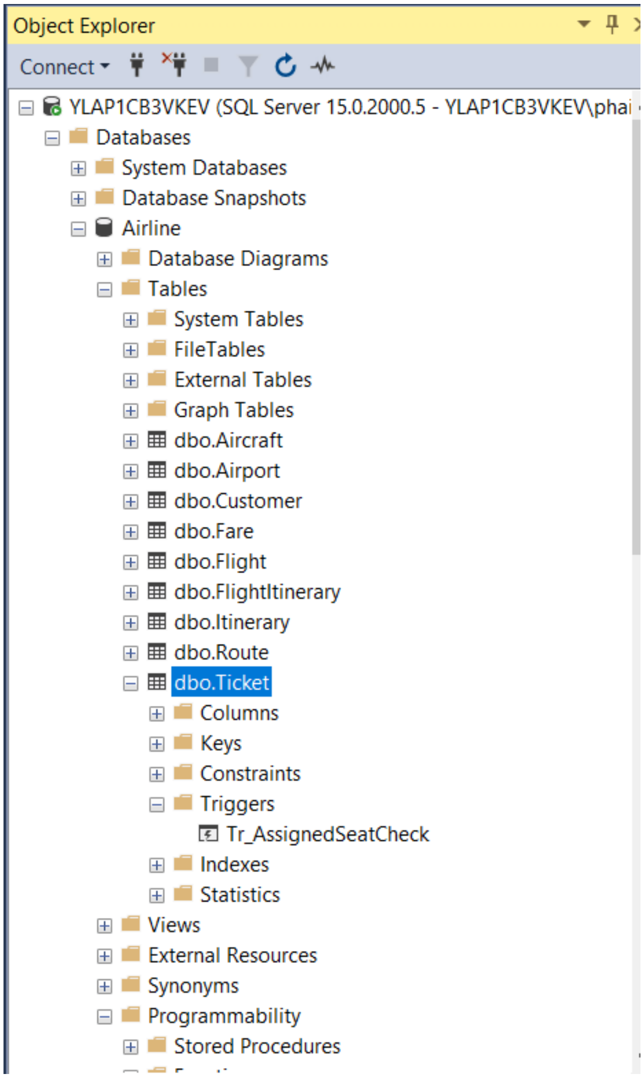
_Add data to all tables.

_Go back to the Table valued Functions\CallFunctions folder and execute these queries if you want to see the table valued function outputs:

- CallItineraryRoutes.sql
- CallFlightCustomers.sql
- CallCustomerTransactions.sql
- Call6MonthTransactions.sql

_ Open the trigger file in the \Table\Trigger for Ticket table folder and execute it.

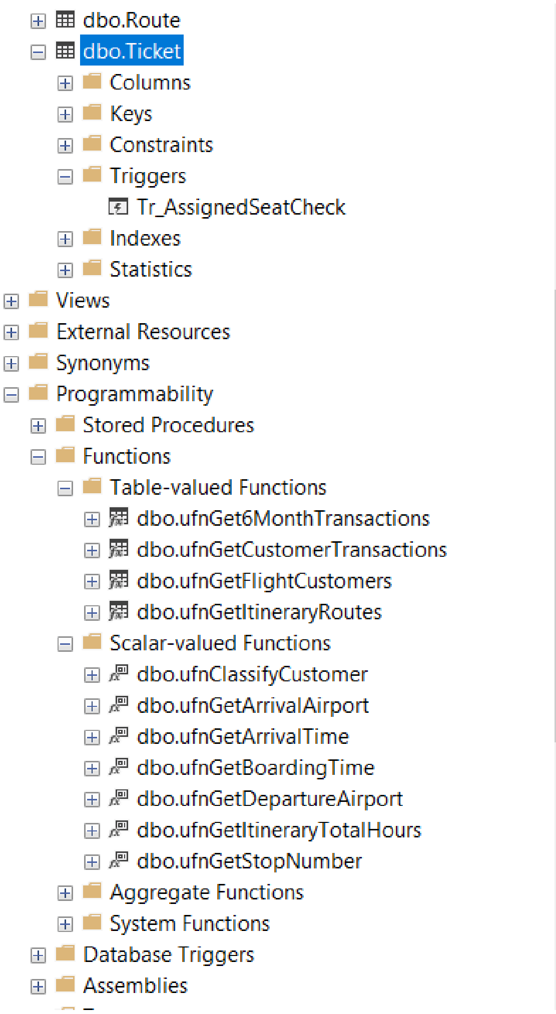
_In case there's error while executing, you'll need to copy and paste the query to a new table and execute each under the right folder as shown in these screen shots.



Object Explorer

Connect ▾

- YLAP1CB3VKEV (SQL Server 15.0.2000.5 - YLAP1CB3VKEV\phai
  - Databases
    - System Databases
    - Database Snapshots
    - Airline
      - Database Diagrams
      - Tables
        - System Tables
        - FileTables
        - External Tables
        - Graph Tables
        - dbo.Aircraft
        - dbo.Airport
        - dbo.Customer
        - dbo.Fare
        - dbo.Flight
        - dbo.FlightItinerary
        - dbo.Itinerary
        - dbo.Route
        - dbo.Ticket
          - Columns
          - Keys
          - Constraints
          - Triggers
            - Tr_AssignedSeatCheck
          - Indexes
          - Statistics
      - Views
      - External Resources
      - Synonyms
      - Programmability
        - Stored Procedures



- dbo.Route
- dbo.Ticket
  - Columns
  - Keys
  - Constraints
  - Triggers
    - Tr_AssignedSeatCheck
  - Indexes
  - Statistics
- Views
- External Resources
- Synonyms
- Programmability
  - Stored Procedures
  - Functions
    - Table-valued Functions
      - dbo.ufnGet6MonthTransactions
      - dbo.ufnGetCustomerTransactions
      - dbo.ufnGetFlightCustomers
      - dbo.ufnGetItineraryRoutes
    - Scalar-valued Functions
      - dbo.ufnClassifyCustomer
      - dbo.ufnGetArrivalAirport
      - dbo.ufnGetArrivalTime
      - dbo.ufnGetBoardingTime
      - dbo.ufnGetDepartureAirport
      - dbo.ufnGetItineraryTotalHours
      - dbo.ufnGetStopNumber
    - Aggregate Functions
    - System Functions
  - Database Triggers
  - Assemblies

VI. Limitations

The database does not support updates on the Ticket table, which means if you go back to one existed tuple and change the value, the trigger will not run. To make the database updated, we'll need to write some further cursors to update and input multiple data to the table

Another limitation is that the database does not check if the customer has mistakenly booked the same flight multiple times or not. That also means the flight may sometimes has duplicated customer appearing several times in the list with different itinerary confirmation ID because the customer mistakenly booked the flight multiple times (or clicked "make payment" multiple times). In fact in real life this can easily happen and usually the customer will be contacted to verify their booking and cancel duplicated booking, so we don't really need to delete the duplicated booking here.

References

Name & country of origins of aircrafts for aircraft table: https://www.airliners.net/aircraft-data

Airport list: https://www.world-airport-codes.com/us-top-40-airports.html