GSGY6083B F21

ML7136

OZ3499

Final Project Part 1

11/12/2021

**Summary**

Customer is a subtype passenger, but passenger doesn’t have to be the customer, So the passenger to customer is a partial completeness relation.

Members and the booking agent are subtypes of the customer, but the customer can be neither a member or booking agent, so there is partial completeness constraint for the customer, member and booking agent table.

Customer will purchase the insurance for her/himself or for other passengers in the flight, we assume that a passenger can’t have multiple insurance plans, so the insurance plan is one to many relationship to the passenger. In addition, it is optional for the passenger, because not every passenger will get the insurance.

In the purchase table, purchaseID  is the primary key. A customer can have multiple purchaseIDs, since customers can buy insurance for many passengers. The purchase table has the plan purchased and coverage person name to keep the record, so it is easy to check who has purchased the ticket.

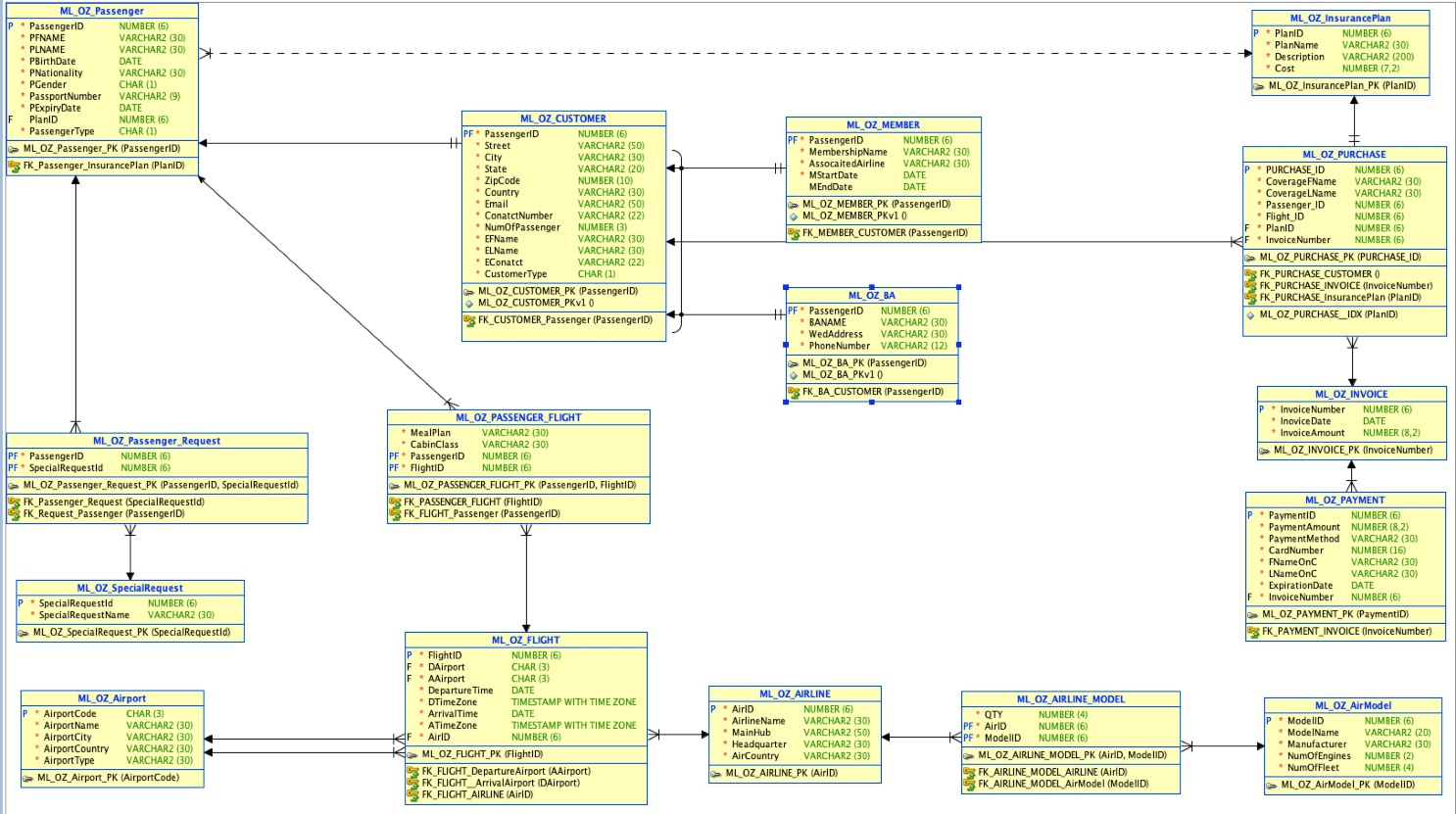
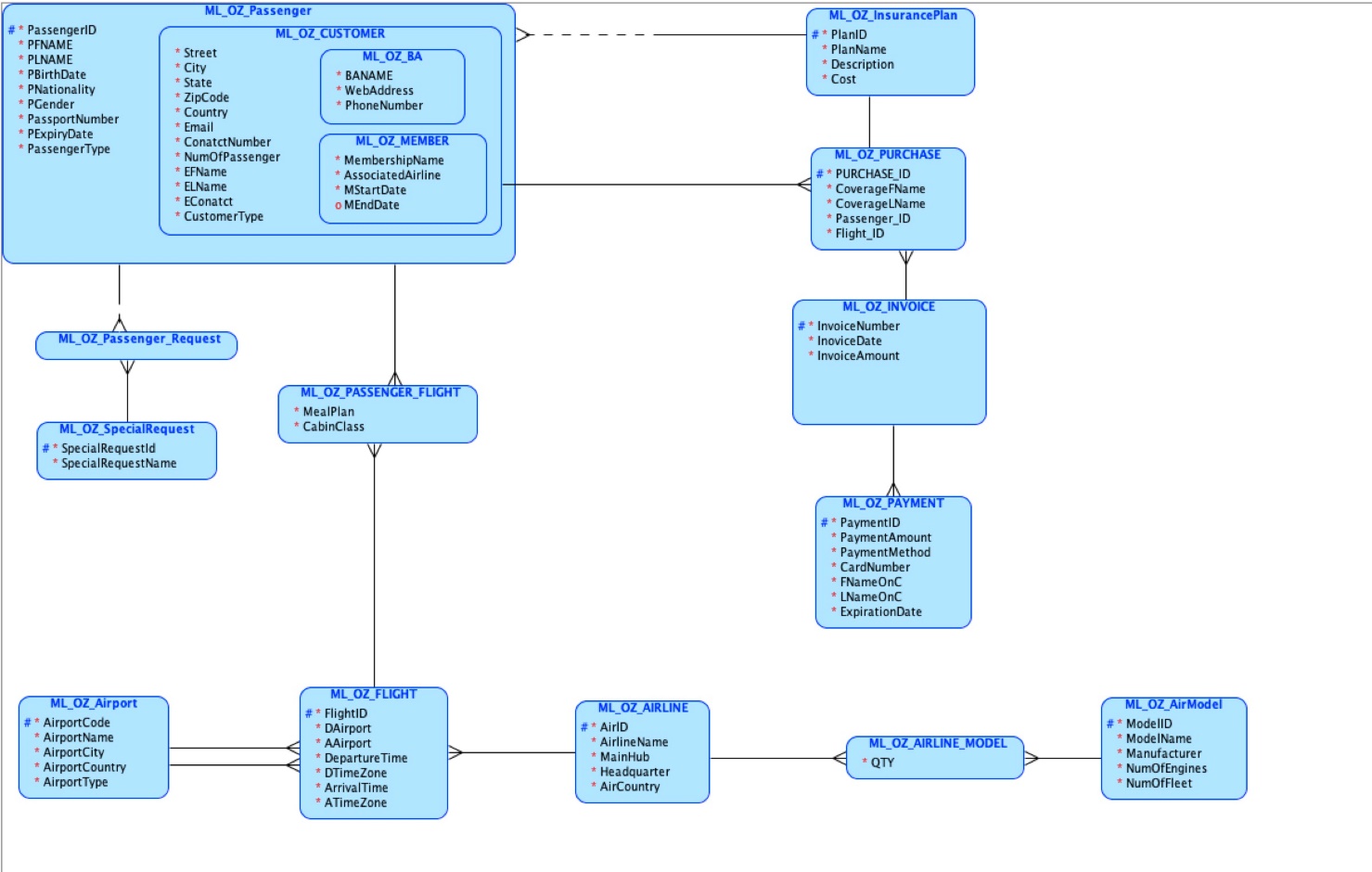
Then one or many purchases will generate an invoice for the customer. One invoice can have multiple payments, since the customer might split the payment with different cards, so there is one to many relationship between invoice and payment table. It also required that the payment date should be three days before the flight.

The passenger and the flight have many to many relationship, so we need an associate entity in between . The meal plan and cabin class will be included in the passenger\_flight entity to connect the passenger and flight entity.

There are two relationships from the flight table to the airport, one is for departure airport, another is for arrival airport. The airline and the flight have the relationship of one to many because every airline can have multiple flights, but one flight only belongs to one airline.

Airline and airplane model should have many-to-many relationship because one airline can have different models, and one model of airplane is usually owned by multiple airlines, so there is an airline\_model table to demonstrate the quantity of each airplane model in each airline.

The relationship between Passenger and SpeicialRequest is many-to-many. To solve this an associated entity is created.



**Constraints**

-- ml\_oz\_passenger

alter table ml\_oz\_passenger add constraint c\_passengerid check (passengerid between 100000 and 999999);

alter table ml\_oz\_passenger add constraint c\_passengergender check (pgender in ('M', 'F'));

alter table ml\_oz\_passenger add constraint c\_pfname check (pfname = initcap(pfname));

alter table ml\_oz\_passenger add constraint c\_pname check (plname = initcap(plname));

-- ml\_oz\_purchase

alter table ml\_oz\_purchase add constraint c\_purchaseid check (purchaseid between 100000 and 999999);

alter table ml\_oz\_purchase add constraint c\_passengerid\_pur check (passengerid between 100000 and 999999);

alter table ml\_oz\_purchase add constraint c\_flightid\_pur check (flightid between 100000 and 999999);

alter table ml\_oz\_purchase add constraint c\_invoicenumber\_pur check (invoicenumber between 100000 and 999999);

-- ml\_oz\_insuranceplan

alter table ml\_oz\_insuranceplan add constraint c\_planid check (planid between 100000 and 999999);

-- ml\_oz\_passenger\_request

alter table ml\_oz\_passenger\_request add constraint c\_passengerid\_pq check (passengerid between 100000 and 999999);

alter table ml\_oz\_passenger\_request add constraint c\_sq\_sq check (specialrequestid between 100000 and 999999);

-- ml\_oz\_passenger\_flight

alter table ml\_oz\_passenger\_flight add constraint c\_passengerid\_pf check (passengerid between 100000 and 999999);

alter table ml\_oz\_passenger\_flight add constraint c\_flightid\_pf check (flightid between 100000 and 999999);

-- ml\_oz\_invoice

alter table ml\_oz\_invoice add constraint c\_invoicenumber check (invoicenumber betweeneen 100000 and 999999);

-- ml\_oz\_specialrequest

alter table ml\_oz\_specialrequest add constraint c\_specialrequestid check (specialrequestid between 100000 and 999999);

-- ml\_oz\_payment

alter table ml\_oz\_payment add constraint c\_paymentid check (paymentid between 100000 and 999999);

alter table ml\_oz\_payment add constraint c\_invoicenumber\_pay check (invoicenumber between 100000 and 999999);

-- ml\_oz\_flight

alter table ml\_oz\_flight add constraint c\_flightid check (flightid between 100000 and 999999);

alter table ml\_oz\_flight add constraint c\_airid\_flight check (airid between 100000 and 999999);

-- ml\_oz\_airline

alter table ml\_oz\_airline add constraint c\_airid check (airid between 100000 and 999999);

-- ml\_oz\_airline\_model

alter table ml\_oz\_airline\_model add constraint c\_airid\_am check (airid between 100000 and 999999);

alter table ml\_oz\_airline\_model add constraint c\_modelid\_am check (modelid between 100000 and 999999);

-- ml\_oz\_airmodel

alter table ml\_oz\_airmodel add constraint c\_modelid check (modelid between 100000 and 999999);

**Insert Data**

-- insert into ml\_oz\_airline

insert into ml\_oz\_airline (airid, airlinename, mainhub, headquarter, aircountry) values (200001, 'Emirates Airline', 'Dubai International Airport', 'Dubai', 'United Arab Emirates');

insert into ml\_oz\_airline (airid, airlinename, mainhub, headquarter, aircountry) values ( 200002, 'Qatar Airways', 'Doha International Airport', 'Doha', 'Qatar');

insert into ml\_oz\_airline (airid, airlinename, mainhub, headquarter, aircountry) values ( 200003, 'AirAsia', 'Kuala Lumpur International Airport', 'Kuala Lumpur', 'Malaysia');

insert into ml\_oz\_airline (airid, airlinename, mainhub, headquarter, aircountry) values ( 200004, 'Air China', 'Beijing Capital International Airport', 'Beijing', 'China');

insert into ml\_oz\_airline (airid, airlinename, mainhub, headquarter, aircountry) values ( 200005, 'Jet Airways', 'Chhatrapati Shivaji International Airport', 'Mumbai', 'India');

insert into ml\_oz\_airline (airid, airlinename, mainhub, headquarter, aircountry) values ( 200006, 'JAL Group', 'Narita and Tokyo International Airport', 'Tokyo', 'Japn');

insert into ml\_oz\_airline (airid, airlinename, mainhub, headquarter, aircountry) values ( 200007, 'Jetstar', 'Melbourne Airport', 'Melbourne', 'Australia');

insert into ml\_oz\_airline (airid, airlinename, mainhub, headquarter, aircountry) values ( 200008, 'Lion Air', 'Soekarno-Hatta International Airport', 'Jakarta', 'Indonesia');

insert into ml\_oz\_airline (airid, airlinename, mainhub, headquarter, aircountry) values ( 200009, 'Korean Air', 'Incheon International Airport', 'Seoul', 'South Korea');

insert into ml\_oz\_airline (airid, airlinename, mainhub, headquarter, aircountry) values ( 200010, 'Kingfisher Airlines', 'Bengaluru International Airport', 'Bangalore', 'India');

insert into ml\_oz\_airline (airid, airlinename, mainhub, headquarter, aircountry) values ( 200011, 'Singapore Airlines', 'Singapore Changi Airport', 'Singapore', 'Singapore');

insert into ml\_oz\_airline (airid, airlinename, mainhub, headquarter, aircountry) values ( 200012, 'Thai Airways International', 'Suvarnabhumi Airport', 'Bangkok', 'Thailand');

insert into ml\_oz\_airline (airid, airlinename, mainhub, headquarter, aircountry) values ( 200013, 'Vietnam Airlines', 'Noi Bai International Airport Tan Son Nhat International Airport', 'Hanoi Ho Chi Minh City', 'Vietnam');

insert into ml\_oz\_airline (airid, airlinename, mainhub, headquarter, aircountry) values ( 200014, 'Air Berlin Group', 'Sheremetyevo International Airport', 'Berlin', 'Germany');

insert into ml\_oz\_airline (airid, airlinename, mainhub, headquarter, aircountry) values ( 200015, 'Air France KLM', 'Paris-Charles de Gaulle Airport', 'Paris', 'France');

insert into ml\_oz\_airline (airid, airlinename, mainhub, headquarter, aircountry) values ( 200016, 'Alitalia', 'Leonardo da Vinci-Fiumicino Airport', 'Rome', 'Italy');

insert into ml\_oz\_airline (airid, airlinename, mainhub, headquarter, aircountry) values ( 200017, 'Austrian Airlines Group', 'Vienna International Airport', 'Vienna', 'Austria');

insert into ml\_oz\_airline (airid, airlinename, mainhub, headquarter, aircountry) values ( 200018, 'EasyJet', 'London Luton Airport', 'London', 'United Kingdom');

insert into ml\_oz\_airline (airid, airlinename, mainhub, headquarter, aircountry) values ( 200019, 'Ryanair', 'Dublin Airport', 'Dublin', 'Ireland');

insert into ml\_oz\_airline (airid, airlinename, mainhub, headquarter, aircountry) values ( 200020, 'Avianca', 'El Dorado International Airport', 'Bogota', 'Columbia');

-- insert into ml\_oz\_airmodel

insert into ml\_oz\_airmodel (modelid, modelname, manufacturer, numofengines, numoffleet) values (100001, '737-800', 'Boeing', 2,826);

insert into ml\_oz\_airmodel (modelid, modelname, manufacturer, numofengines, numoffleet) values (100002, '737-700', 'Boeing', 2,579);

insert into ml\_oz\_airmodel (modelid, modelname, manufacturer, numofengines, numoffleet) values ( , , , ,);

-- insert into ml\_oz\_airline\_model

insert into ml\_oz\_airline\_model (qty, airid, modelid) values (100, 300001, 400001);

insert into ml\_oz\_airline\_model (qty, airid, modelid) values (100, 300002, 400002);

insert into ml\_oz\_airline\_model (qty, airid, modelid) values (100, 300003, 400003);

insert into ml\_oz\_airline\_model (qty, airid, modelid) values (100, 300004, 400004);

insert into ml\_oz\_airline\_model (qty, airid, modelid) values (100, 300005, 400005);

insert into ml\_oz\_airline\_model (qty, airid, modelid) values (100, 300006, 400006);

insert into ml\_oz\_airline\_model (qty, airid, modelid) values (100, 300007, 400007);

insert into ml\_oz\_airline\_model (qty, airid, modelid) values (100, 300008, 400008);

insert into ml\_oz\_airline\_model (qty, airid, modelid) values (100, 300009, 400009);

insert into ml\_oz\_airline\_model (qty, airid, modelid) values (100, 300010, 400010);

insert into ml\_oz\_airline\_model (qty, airid, modelid) values (100, 300011, 400011);

insert into ml\_oz\_airline\_model (qty, airid, modelid) values (100, 300012, 400012);

insert into ml\_oz\_airline\_model (qty, airid, modelid) values (100, 300013, 400013);

insert into ml\_oz\_airline\_model (qty, airid, modelid) values (100, 300014, 400014);

insert into ml\_oz\_airline\_model (qty, airid, modelid) values (100, 300015, 400015);

insert into ml\_oz\_airline\_model (qty, airid, modelid) values (100, 300016, 400016);

insert into ml\_oz\_airline\_model (qty, airid, modelid) values (100, 300017, 400017);

insert into ml\_oz\_airline\_model (qty, airid, modelid) values (100, 300018, 400018);

insert into ml\_oz\_airline\_model (qty, airid, modelid) values (100, 300019, 400019);

insert into ml\_oz\_airline\_model (qty, airid, modelid) values (100, 300020, 400020);

--insert into ml\_oz\_airport

insert into ml\_oz\_airport (airportcode, airportname, airportcity, airportcountry, airporttype) values ('JAC', 'Jackson Hole', 'Jackson', 'United States', 'International');

insert into ml\_oz\_airport (airportcode, airportname, airportcity, airportcountry, airporttype) values ('JAG', 'Shahbaz Air Base', 'Jacobabad', 'Pakistan', 'International');

insert into ml\_oz\_airport (airportcode, airportname, airportcity, airportcountry, airporttype) values ('JAI', 'Jaipur International', 'Jaipur', 'India', 'Both');

insert into ml\_oz\_airport (airportcode, airportname, airportcity, airportcountry, airporttype) values ('JAK', 'Jacmel', 'Jacmel', 'Haiti', 'Both');

insert into ml\_oz\_airport (airportcode, airportname, airportcity, airportcountry, airporttype) values ('JAX', ''Jacksonville International, 'Jacksonville', 'United States', 'Both');

insert into ml\_oz\_airport (airportcode, airportname, airportcity, airportcountry, airporttype) values ('JED', 'King Abdulaziz International', 'Jeddah', 'Saudi Arabia', 'Both');

insert into ml\_oz\_airport (airportcode, airportname, airportcity, airportcountry, airporttype) values ('JFK', 'John F Kennedy International', 'New york', 'United States', 'Both');

insert into ml\_oz\_airport (airportcode, airportname, airportcity, airportcountry, airporttype) values ('JAE', 'Shumba', 'Jaén', Perú'', 'International');

insert into ml\_oz\_airport (airportcode, airportname, airportcity, airportcountry, airporttype) values ('JAD', 'Perth Jandakot', 'Perth', 'Australia', 'International');

insert into ml\_oz\_airport (airportcode, airportname, airportcity, airportcountry, airporttype) values ('JAU', 'Francisco Carle', 'Jauja', 'Perú', 'International');

insert into ml\_oz\_airport (airportcode, airportname, airportcity, airportcountry, airporttype) values (JBQ, 'La Isabela International', 'La Isabela', 'Dominican Republic', 'International');

insert into ml\_oz\_airport (airportcode, airportname, airportcity, airportcountry, airporttype) values ('JAP', 'Chacarita', 'Puntarenas', 'Costa Rica', 'Domestic');

insert into ml\_oz\_airport (airportcode, airportname, airportcity, airportcountry, airporttype) values ('JAQ', 'Jacquinot Bay', 'Jacquinot Bay', 'Papua New Guinea', 'Domestic');

insert into ml\_oz\_airport (airportcode, airportname, airportcity, airportcountry, airporttype) values ('BAB', 'Beale Air Force Base', 'Marysville', 'United States', 'Domestic');

insert into ml\_oz\_airport (airportcode, airportname, airportcity, airportcountry, airporttype) values ('BAF', 'Westfield-Barnes Regional', 'Westfield/Springfield', 'United States', 'Domestic');

insert into ml\_oz\_airport (airportcode, airportname, airportcity, airportcountry, airporttype) values ('BAL', 'Batman', 'Batman', 'Turkey', 'Domestic');

-- insert into ml\_oz\_flight

insert into ml\_oz\_flight(flightid, dairport, aairport, departuretime, arrivaltime, airid) values (500001, 'JAC', 'JAG', TIMESTAMP '2021-01-31 09:26:56.66 +02:00', TIMESTAMP '2021-01-31 04:23:32.66 +05:00', 200001);

insert into ml\_oz\_flight(flightid, dairport, aairport, departuretime, arrivaltime, airid) values (500002, 'JBQ', 'JAU', TIMESTAMP '2021-01-31 09:26:56.66 +02:00', TIMESTAMP '2021-01-31 04:23:32.66 +05:00', 200002);

insert into ml\_oz\_flight(flightid, dairport, aairport, departuretime, arrivaltime, airid) values (500003, 'JAD', 'JAE', TIMESTAMP '2021-01-31 09:26:56.66 +02:00', TIMESTAMP '2021-01-31 04:23:32.66 +05:00', 100003);

insert into ml\_oz\_flight(flightid, dairport, aairport, departuretime, arrivaltime, airid) values (500004, 'JED', 'JFK', TIMESTAMP '2021-12-31 09:26:56.66 +02:00', TIMESTAMP '2022-01-01 04:23:32.66 +05:00', 100004);

insert into ml\_oz\_flight(flightid, dairport, aairport, departuretime, arrivaltime, airid) values (500005, 'JAX', 'JAE', TIMESTAMP '2021-11-31 09:26:56.66 +02:00', TIMESTAMP '2021-12-01 04:23:32.66 +05:00', 100005);

insert into ml\_oz\_flight(flightid, dairport, aairport, departuretime, arrivaltime, airid) values (500006, 'JAX', 'JAI', TIMESTAMP '2021-09-11 09:26:56.66 +02:00', TIMESTAMP '2021-09-12 04:23:32.66 +05:00', 100006);

insert into ml\_oz\_flight(flightid, dairport, aairport, departuretime, arrivaltime, airid) values (500007, 'JAP', 'GAQ', TIMESTAMP '2021-07-28 09:26:56.66 +02:00', TIMESTAMP '2021-07-29 04:23:32.66 +05:00', 100007);

insert into ml\_oz\_flight(flightid, dairport, aairport, departuretime, arrivaltime, airid) values (500008, 'BAL', 'BAF', TIMESTAMP '2023-07-28 09:26:56.66 +02:00', TIMESTAMP '2023-07-29 04:23:32.66 +05:00', 100008);

insert into ml\_oz\_flight(flightid, dairport, aairport, departuretime, arrivaltime, airid) values (500009, 'JAP', 'BAB', TIMESTAMP '2022-07-28 09:26:56.66 +02:00', TIMESTAMP '2022-07-29 04:23:32.66 +05:00', 100009);

insert into ml\_oz\_flight(flightid, dairport, aairport, departuretime, arrivaltime, airid) values (500010, 'JFK', 'GAQ', TIMESTAMP '2021-12-28 09:26:56.66 +02:00', TIMESTAMP '2021-12-29 04:23:32.66 +05:00', 100010);

-- insert into ml\_oz\_insuranceplan

insert into ml\_oz\_insuranceplan (planid, planname, description, cost) values (600001, 'Trip Cancellation', 'Get 3 times of refund when the trip is cancelled', 1234.92);

insert into ml\_oz\_insuranceplan (planid, planname, description, cost) values (600002, 'Trip Interruption', 'Get 2 times of refund when the trip is interrupt', 923.12);

insert into ml\_oz\_insuranceplan (planid, planname, description, cost) values (600003, 'Medical Insurance', 'Pay full medical bill if incident happen on the airplane', 1123.12);

insert into ml\_oz\_insuranceplan (planid, planname, description, cost) values (600004, 'Baggage Insurance', 'Pay up to 2000$ suitcase worth', 93.12);

insert into ml\_oz\_insuranceplan (planid, planname, description, cost) values (600005, 'Accidental Death Insurance', 'Get 1 million if death happen', 500.12);

insert into ml\_oz\_insuranceplan (planid, planname, description, cost) values (600006, 'All-inclusive Insurance', 'All the insurance included.', 2123.12);

-- insert into ml\_oz\_purchase

insert into ml\_oz\_purchase (purchaseid, coveragefname, coveragelname, planid, invoicenumber, flightid, passengerid) values ();

-- insert into ml\_oz\_passenger

insert into ml\_oz\_passenger (passengerid, pfname, plname, pbirthdate, pnationality, pgender, passportnumber, pexpirydate, passengertype)

values (100001, 'Peter', 'Parker', DATE '1998-12-25', 'United States', 'M', 'abcd12345', DATE '2026-12-25', 'P');

insert into ml\_oz\_passenger (passengerid, pfname, plname, pbirthdate, pnationality, pgender, passportnumber, pexpirydate, passengertype)

values (100006, 'Peter', 'Parker', DATE '1998-12-25', 'United States', 'M', 'abcd12345', DATE '2026-12-25', 'P');

insert into ml\_oz\_passenger (passengerid, pfname, plname, pbirthdate, pnationality, pgender, passportnumber, pexpirydate, passengertype)

values (100007, 'Tony', 'Kim', DATE '2001-2-5', 'United States', 'M', 'abrd12567', DATE '2026-9-20', 'C');

insert into ml\_oz\_passenger (passengerid, pfname, plname, pbirthdate, pnationality, pgender, passportnumber, pexpirydate, passengertype)

values (100008, 'Amy', 'Smith', DATE '1994-3-9', 'United States', 'F', 'eras2312', DATE '2022-12-20', 'C');

insert into ml\_oz\_passenger (passengerid, pfname, plname, pbirthdate, pnationality, pgender, passportnumber, pexpirydate, passengertype)

values (100009, 'Mike', 'Anderson', DATE '1988-6-6', 'Scotland', 'M', 'ebrd34567', DATE '20223-3-27', 'C');

-- insert into ml\_oz\_customer

insert into ml\_oz\_customer (passengerid, street, city, state, zipcode, country, email, contactnumber, numofpassenger, efname, elname, econtact, customertype)

values (100006, 'a', 'b', 'c', 111111, 'e', 'email', '2323-423', 2, 'harry', 'porter', '1212-3224', 'C');

insert into ml\_oz\_customer (passengerid, street, city, state, zipcode, country, email, contactnumber, numofpassenger, efname, elname, econtact, customertype)

values (100007, '100Ave', 'Boston', 'Massachusetts', 02215, 'United States', '100007@gmail.com', '+1 000-0000-001', 3, 'Lily', 'Anderson', '+1 123-634-743', 'M');

insert into ml\_oz\_customer (passengerid, street, city, state, zipcode, country, email, contactnumber, numofpassenger, efname, elname, econtact, customertype)

values (100009, ‘101Ave’, ‘New York’, ‘New York’, 11212, ‘United States’, 100009@gmail.com’, ‘+1 000-012-982’, 2, ‘Tom’, ‘Gay’, ‘+1 012-321-864’, B)

-- insert into ml\_oz\_ba

insert into ml\_oz\_ba(passengerid, bname, webaddress, phonenumber) values (100009, ‘Kayak’, ‘www.kayak.com’, ‘+1 636-123-865’);

-- insert into ml\_oz\_member

insert into ml\_oz\_member (passengerid, membershipname, assocaitedairline, mstartdate, menfdate) values (100007, ‘a’, ‘b’, DATE ‘2019-3-7’, NULL);

**List total number of records**

-- count

select count (\*) from ml\_oz\_airline;

select count (\*) from ml\_oz\_passenger;

select count (\*) from ml\_oz\_customer;

select count (\*) from ml\_oz\_member;

select count (\*) from ml\_oz\_ba;

select count (\*) from ml\_oz\_passenger\_request;

select count (\*) from ml\_oz\_specialrequest;

select count (\*) from ml\_oz\_passenger\_flight;

select count (\*) from ml\_oz\_airport;

select count (\*) from ml\_oz\_flight;

select count (\*) from ml\_oz\_airline\_model;

select count (\*) from ml\_oz\_airmodel;

select count (\*) from ml\_oz\_insuranceplan;

select count (\*) from ml\_oz\_purchase;

select count (\*) from ml\_oz\_invoice;

select count (\*) from ml\_oz\_payment;

**Data Dictionary Queries**

select \* from user\_cons\_columns;

select \* from ALL\_TAB\_COLUMNS;