MinruiLi
Olivia Zhai
Project Part1
Fall 2021
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.

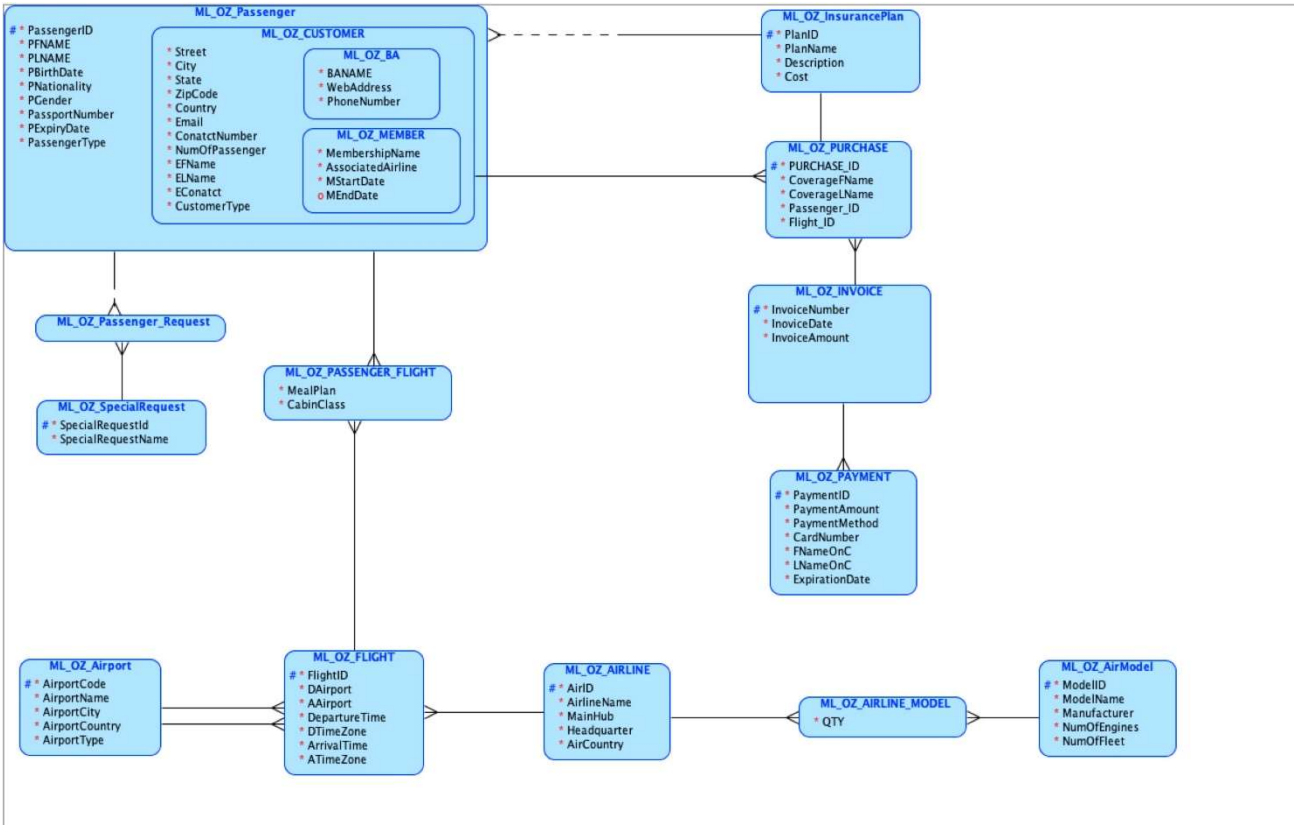
Then one or many purchases will generate an invoice for the customer. One invoice can have multiple payments, since the customer might split the payment with different cards, so there is one to many relationship between invoice and payment table. It also required that the payment date should be three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

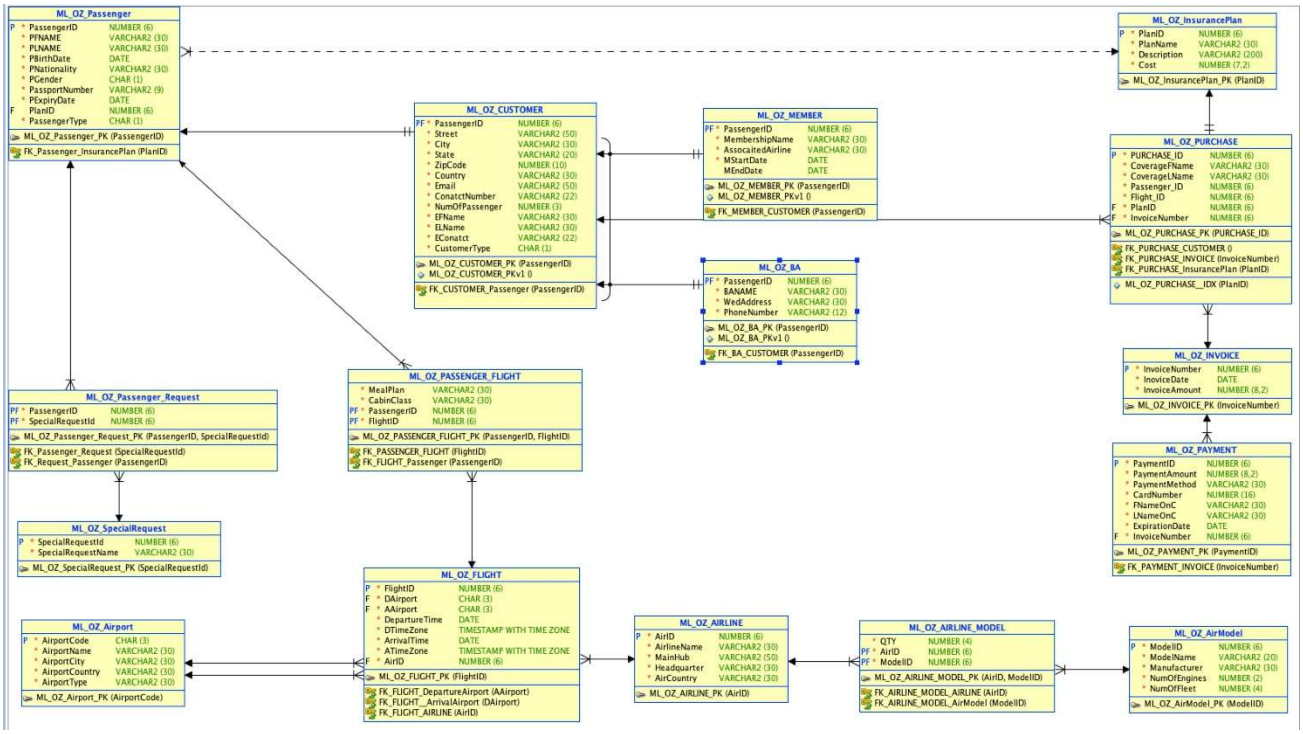
The passenger and the flight have many to many relationship, so we need an associate entity in between . The meal plan and cabin class will be included in the passenger_flight entity to connect the passenger and flight entity.

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.


ML_OZ_Passenger
* PassengerID
* PFNAME
* PLNAME
* PBirthDate
* PNationality
* PGender
* PassportNumber
* PExpiryDate
* PassengerType
ML_OZ_CUSTOMER
* Street
* City
* State
* ZipCode
* Country
* Email
* ConatctNumber
* NumOfPassenger
* EFName
* ELName
* EConatct
* CustomerType
ML_OZ_BA
* BANAME
* WebAddress
* PhoneNumber
ML_OZ_MEMBER
* MembershipName
* AssociatedAirline
* MStartDate
o MEndDate
ML_OZ_InsurancePlan
* PlanID
* PlanName
* Description
* Cost
ML_OZ_PURCHASE
* PURCHASE_ID
* CoverageFName
* CoverageLName
* Passenger_ID
* Flight_ID
ML_OZ_INVOICE
* InvoiceNumber
* InovoiceDate
* InvoiceAmount
ML_OZ_PAYMENT
* PaymentID
* PaymentAmount
* PaymentMethod
* CardNumber
* FNameOnC
* LNameOnC
* ExpirationDate
ML_OZ_Passenger_Request
ML_OZ_SpecialRequest
* SpecialRequestId
* SpecialRequestName
ML_OZ_PASSENGER_FLIGHT
* MealPlan
* CabinClass
ML_OZ_Airport
* AirportCode
* AirportName
* AirportCity
* AirportCountry
* AirportType
ML_OZ_FLIGHT
* FlightID
* DAirport
* AAirport
* DepartureTime
* DTimeZone
* ArrivalTime
* ATimeZone
ML_OZ_AIRLINE
* AirID
* AirlineName
* MainHub
* Headquarter
* AirCountry
ML_OZ_AIRLINE_MODEL
* QTY
ML_OZ_AirModel
* ModelID
* ModelName
* Manufacturer
* NumOfEngines
* NumOfFleet


ML_OZ_Passenger
P * PassengerID NUMBER (6)
* PFNAME VARCHAR2 (30)
* PLNAME VARCHAR2 (30)
* PBirthDate DATE
* PNationality VARCHAR2 (30)
* PGender CHAR (1)
* PassportNumber VARCHAR2 (9)
* PExpiryDate DATE
F PlanID NUMBER (6)
* PassengerType CHAR (1)
ML_OZ_Passenger_PK (PassengerID)
FK_Passenger_InsurancePlan (PlanID)
ML_OZ_InsurancePlan
P * PlanID NUMBER (6)
* PlanName VARCHAR2 (30)
* Description VARCHAR2 (200)
* Cost NUMBER (7,2)
ML_OZ_InsurancePlan_PK (PlanID)
ML_OZ_CUSTOMER
PF * PassengerID NUMBER (6)
* Street VARCHAR2 (50)
* City VARCHAR2 (30)
* State VARCHAR2 (20)
* ZipCode NUMBER (10)
* Country VARCHAR2 (30)
* Email VARCHAR2 (50)
* ConatctNumber VARCHAR2 (22)
* NumOfPassenger NUMBER (3)
* EFName VARCHAR2 (30)
* ELName VARCHAR2 (30)
* EConatct VARCHAR2 (22)
* CustomerType CHAR (1)
ML_OZ_CUSTOMER_PK (PassengerID)
ML_OZ_CUSTOMER_PKv1 0
FK_CUSTOMER_Passenger (PassengerID)
ML_OZ_MEMBER
PF * PassengerID NUMBER (6)
* MembershipName VARCHAR2 (30)
* AssocaitedAirline VARCHAR2 (30)
* MStartDate DATE
MEndDate DATE
ML_OZ_MEMBER_PK (PassengerID)
ML_OZ_MEMBER_PKv1 0
FK_MEMBER_CUSTOMER (PassengerID)
ML_OZ_PURCHASE
P * PURCHASE_ID NUMBER (6)
* CoverageFName VARCHAR2 (30)
* CoverageLName VARCHAR2 (30)
* Passenger_ID NUMBER (6)
* Flight_ID NUMBER (6)
F * PlanID NUMBER (6)
F * InvoiceNumber NUMBER (6)
ML_OZ_PURCHASE_PK (PURCHASE_ID)
FK_PURCHASE_CUSTOMER 0
FK_PURCHASE_INVOICE (InvoiceNumber)
FK_PURCHASE_InsurancePlan (PlanID)
ML_OZ_PURCHASE__IDX (PlanID)
ML_OZ_BA
PF * PassengerID NUMBER (6)
* BANAME VARCHAR2 (30)
* WedAddress VARCHAR2 (30)
* PhoneNumber VARCHAR2 (12)
ML_OZ_BA_PK (PassengerID)
ML_OZ_BA_PKv1 0
FK_BA_CUSTOMER (PassengerID)
ML_OZ_INVOICE
P * InvoiceNumber NUMBER (6)
* InovoiceDate DATE
* InvoiceAmount NUMBER (8,2)
ML_OZ_INVOICE_PK (InvoiceNumber)
ML_OZ_PAYMENT
P * PaymentID NUMBER (6)
* PaymentAmount NUMBER (8,2)
* PaymentMethod VARCHAR2 (30)
* CardNumber NUMBER (16)
* FNameOnC VARCHAR2 (30)
* LNameOnC VARCHAR2 (30)
* ExpirationDate DATE
F * InvoiceNumber NUMBER (6)
ML_OZ_PAYMENT_PK (PaymentID)
FK_PAYMENT_INVOICE (InvoiceNumber)
ML_OZ_PASSENGER_FLIGHT
* MealPlan VARCHAR2 (30)
* CabinClass VARCHAR2 (30)
PF * PassengerID NUMBER (6)
PF * FlightID NUMBER (6)
ML_OZ_PASSENGER_FLIGHT_PK (PassengerID, FlightID)
FK_PASSENGER_FLIGHT (FlightID)
FK_FLIGHT_Passenger (PassengerID)
ML_OZ_Passenger_Request
PF * PassengerID NUMBER (6)
PF * SpecialRequestId NUMBER (6)
ML_OZ_Passenger_Request_PK (PassengerID, SpecialRequestId)
FK_Passenger_Request (SpecialRequestId)
FK_Request_Passenger (PassengerID)
ML_OZ_SpecialRequest
P * SpecialRequestId NUMBER (6)
* SpecialRequestName VARCHAR2 (30)
ML_OZ_SpecialRequest_PK (SpecialRequestId)
ML_OZ_FLIGHT
P * FlightID NUMBER (6)
F * DAirport CHAR (3)
F * AAirport CHAR (3)
* DepartureTime DATE
* DTimeZone TIMESTAMP WITH TIME ZONE
* ArrivalTime DATE
* ATimeZone TIMESTAMP WITH TIME ZONE
F * AirID NUMBER (6)
ML_OZ_FLIGHT_PK (FlightID)
FK_FLIGHT_DepartureAirport (AAirport)
FK_FLIGHT_ArrivalAirport (DAirport)
FK_FLIGHT_AIRLINE (AirID)
ML_OZ_Airport
P * AirportCode CHAR (3)
* AirportName VARCHAR2 (30)
* AirportCity VARCHAR2 (30)
* AirportCountry VARCHAR2 (30)
* AirportType VARCHAR2 (30)
ML_OZ_Airport_PK (AirportCode)
ML_OZ_AIRLINE
P * AirID NUMBER (6)
* AirlineName VARCHAR2 (30)
* MainHub VARCHAR2 (50)
* Headquarter VARCHAR2 (30)
* AirCountry VARCHAR2 (30)
ML_OZ_AIRLINE_PK (AirID)
ML_OZ_AIRLINE_MODEL
* QTY NUMBER (4)
PF * AirID NUMBER (6)
PF * ModelID NUMBER (6)
ML_OZ_AIRLINE_MODEL_PK (AirID, ModelID)
FK_AIRLINE_MODEL_AIRLINE (AirID)
FK_AIRLINE_MODEL_AirModel (ModelID)
ML_OZ_AirModel
P * ModelID NUMBER (6)
* ModelName VARCHAR2 (20)
* Manufacturer VARCHAR2 (30)
* NumOfEngines NUMBER (2)
* NumOfFleet NUMBER (4)
ML_OZ_AirModel_PK (ModelID)

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;
```