Ceng 315

Information Management

Term Project Report

Term Project

"Online Flight Ticket Booking Database System"

By

Berk Atik	-	240201004
Berkay Karakoç	-	230201037
Cem Sakızcı	-	230201036
Eren Can Güleç	-	230201001
İremnur Kulaksız	-	230201041

İYTE – Computer Engineering

Fall 2018 - 2019

INTRODUCTION

In our daily lives if we want to find the cheapest flight from somewhere to somewhere we must search from different websites of different airlines, which would result in waste of time and effort, and if we consider that ticket prices changes time to time, one we thought cheapest might not be the one.

We build this database management system so our users can search for cheap plane tickets from just one platform. By this way, you don't have to search the different websites. This would save time, effort and money, therefore we designed a database for a website with these purposes.

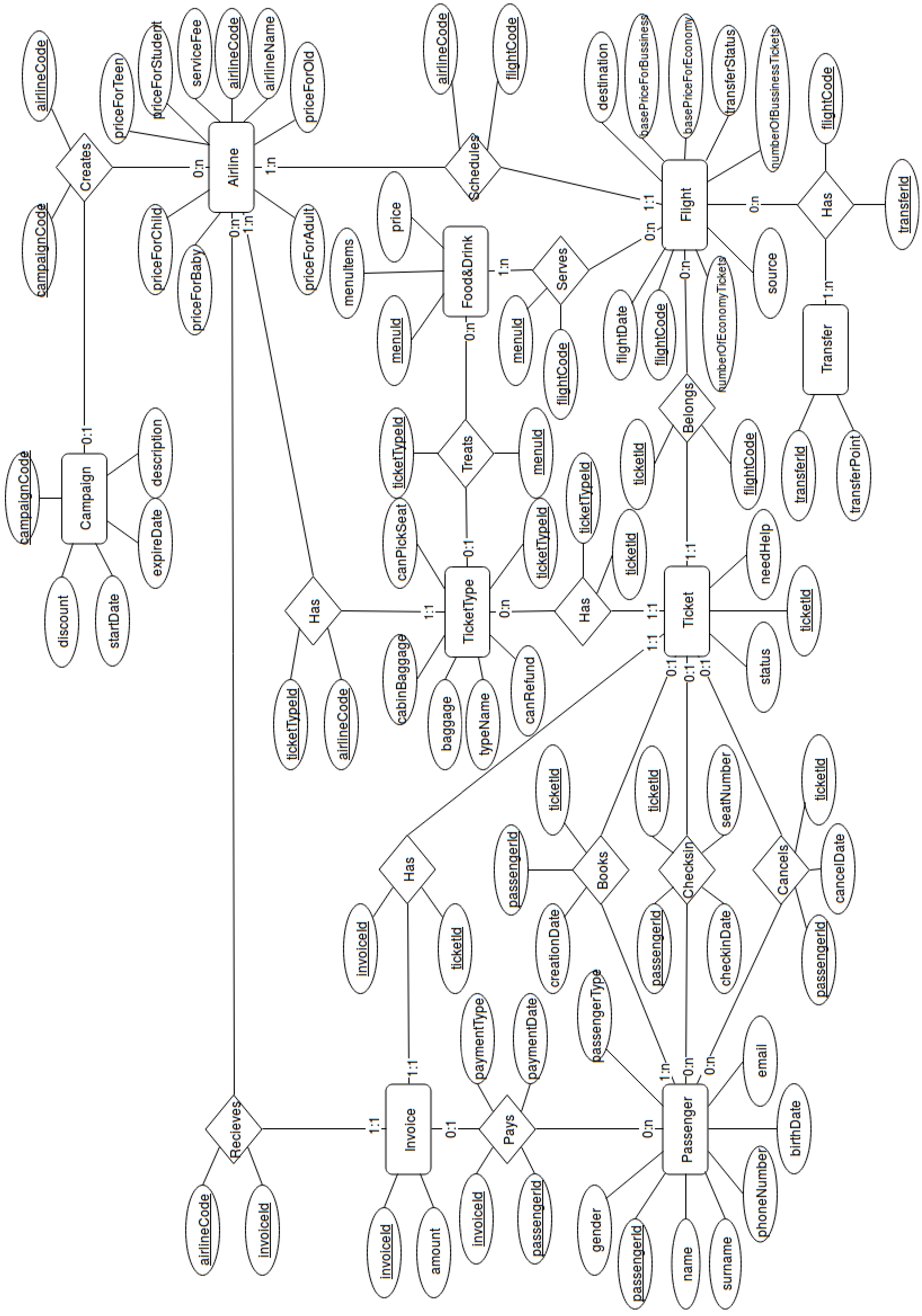
Our Database System includes the following entities:

- Passenger
- Ticket
- Ticket Type
- Flight
- Food&Drink
- Airline
- Transfer
- Invoice
- Campaign

INFORMATION ABOUT WHERE DATA IS PROVIDED FROM

Airline companies serve the ticket and flight data in their own web sites and they also provide an APIs which we can use to acquire the data that we need for our system. After agreeing on the terms of selling flight tickets through our system, we can request the data through APIs of the airlines and store these data in our database.

ER DIAGRAM


Airline
airlineCode
airlineName
serviceFee
priceForStudent
priceForTeen
priceForOld
priceForChild
priceForBaby
priceForAdult
Creates
airlineCode
campaignCode
0:n
0:1
Campaign
campaignCode
description
expireDate
discount
startDate
1:n
Schedules
airlineCode
flightCode
1:1
Flight
destination
basePriceForBussiness
basePriceForEconomy
transferStatus
numberOfBussinessTickets
numberOfEconomyTickets
flightDate
flightCode
source
0:n
Has
flightCode
transferId
1:n
Transfer
transferId
transferPoint
Serves
1:n
0:n
menuId
flightCode
Food&Drink
price
menuItems
menuId
0:n
Treats
ticketTypeId
menuId
0:1
TicketType
canPickSeat
ticketTypeId
cabinBaggage
baggage
typeName
canRefund
0:n
1:n
Has
ticketTypeId
airlineCode
1:1
0:n
Has
ticketTypeId
ticketId
1:1
Belongs
ticketId
flightCode
0:n
1:1
Ticket
needHelp
ticketId
status
1:1
0:1
0:1
0:1
Has
invoiceId
ticketId
1:1
Books
ticketId
passengerId
creationDate
Checksin
ticketId
seatNumber
passengerId
checkinDate
Cancels
ticketId
cancelDate
passengerId
1:n
0:n
0:n
Passenger
passengerType
email
birthDate
phoneNumber
surname
name
passengerId
gender
0:n
Pays
paymentType
paymentDate
invoiceId
passengerId
0:1
Invoice
invoiceId
amount
1:1
Recieves
airlineCode
invoiceId
0:n

RELATINOL MODEL

Entities:

Ticket(ticketId, status, needHelp)

TicketType(ticketTypeId, cabinBaggage, baggage, typeName, canRefund, canPickSeat)

Passenger(passengerId, birthDate, gender, name, surname, email, phoneNum, passengerType)

Invoice(invoiceId, amount)

Flight(flightCode, flightDate, source, destination, basePriceForBusiness, basePriceForEconomy, numberOfBusinessTickets, numberOfEconomyTickets, transferStatus)

Food&Drink(menuId, menuItems, price)

Airline(airlineCode, airlineName, serviceFee, priceForStudent, priceForAdult, priceForChild, priceForOld, priceForTeen, priceForOld)

Transfer(transferId, transferPoint)

Campaign(campaignCode, startDate, expireDate, discount, description)

Relations :

Has(ticketTypeId, ticketId)

Has(ticketId, InvoiceId)

Has(airlineCode, ticketTypeId)

Has(FlightCode, transferId)

Treat(menuId, typeId)

Books(passangerId, ticketId, creationDate)

Cancels(passangerId, ticketId, cancelDate)

checksIn(passangerId, ticketId, seatNumber, checkInDate)

pays(passengerId, invoiceId, paymentType, paymentDate)

recieves(invoiceId, airlineCode)

belongs(ticketId, flightCode)

schedules(flightCode, AirlineCode)

creates(campaignCode, airlineCode)

serves(menuId, FligthCode)

DEFINITIONS OF ATTRIBUTES

- Each passenger is identified by a unique id. Also, every passenger has personal information as the following : date of birth, email, gender, name, surname, phone number and passenger type. Passenger type can be baby, old, child, adult, teen and student. Gender can be either "M" for male or "F" for female.
- Each ticket is identified by a unique id. It has also additional attributes that are status and specification of help need. Status can be booked, available, cheked-in, paid. If any ticket is canceled, it becomes available.
- Each ticket type is identified by a unique id. Additionally, it has a type name to indicate the class of that ticket, baggage that indicates baggage limits for related ticket type, cabin baggage that indicates cabin baggage limits for related ticket type, canRefund attribute that checks if refund can be done to that ticket and canPickSeat attribute that checks if the passenger can pick a seat.
- Each flight is identified by a unique id and it has attributes which are date of flight, the source and the destination of the flight, base price for economy and business classes, number of business and economy tickets for the flight, status of transfer. Status of transfer can be 1 and 0. If this value is 1, there exists a transfer point which is represented in Transfer table.
- Each food & drink has a primary key as menuId. It has the attribute menuItems to define the contents of the menu and related price for that menu.
- Each airline is identified by a unique id and every airline has its base price for baby, old, child, adult, teen and student. Each airline has a name and has a service fee.
- Each transfer is identified by a unique id and has a transfer point that shows the transfer airport's name such as İzmir Adnan Menderes Airport.
- Each campaign is identified by a unique id. Also, every campaign has a starting date and expire date of itself, discount amount and description of it.
- Each invoice is identified by a unique id and has an attribute that shows the amount of it.
- Books relation has an attribute called creationDate that indicates a date when the passenger makes booking.
- Checksin relation has attributes called checkinDate and seatNumber. checkinDate indicates a date when the passenger makes check-in, after that passenger can select a seatNumber that indicates the seat that the passenger sits on.
- Cancels relation has an attribute called cancelDate that indicates a date when the passenger cancels his/her ticket.

- Pays relation has attributes paymetType and paymentDate. These indicate type of payment and a date of payment respectively.

CARDINALITY EXPLANATION

Passenger can book at least one ticket.

Passenger might not cancel any ticket or might cancel at least one ticket.

Passenger might not make any check-in of the ticket or might make many check-ins.

Passenger might not pay any invoice or might pay many invoices of tickets. For example a passenger could book but not pay the invoice of the ticket yet.

Ticket can be booked, canceled, checked-in by a passenger.

Ticket has a ticket type.

Ticket has an invoice.

Ticket belongs to a flight.

Flight can have more than one ticket.

Flight can be direct or connecting meaning that flight could not have any transfer point or could have many transfer points.

Flight is scheduled by an airline.

Flight might not serve any food and drink or might serve many food and drink.

Airline schedules at least one flight.

Airline might not create any campaign or can create more than one campaign.

Airline might not receive any invoice or might receive more than one invoice.

Airline has at least one ticket type.

Campaign can be created by an airline.

Invoice is received by an airline.

Invoice can be paid by a passenger.

Invoice belongs to a ticket.

Ticket type might not belong to any ticket ,meaning that type exists but none of the tickets does not have it for that flight, or might belong to more than one ticket.

Ticket type belongs to an airline.

Ticket type can treat food and drink.

Transfer can belong to at least one flight.

Any menu in Food&Drink might not be treated or might be treated by more than one ticket type.

Any menu in Food&Drink is served by at least one flight.

ASSUMPTIONS

Any airline has a ticket type as either business or economy.

Our database assumes only domestic flights.

CALCULATION OF AMOUNT IN INVOICE

To calculate the pricing of a ticket, there are several arguments we should consider. We get the base price of a passengerType from Airline table, we get the base price of ticketType from Flight table. We sum both base prices , if there is a discount, we subtract the discount and add our service fee from the Airline table. Result of these calculations, we find our ticket price.

USERS

- People who are looking for a cheap flight ticket (Passenger)

- Administrator

ACTIONS BY PASSENGER

- Passenger can book a ticket
- Passenger can cancel a ticket
- Passenger can make online check-in
- Passenger can reserve a ticket
- Passenger can pay the invoice
- Passenger can select the ticket type
- Passenger can pick passenger type
- Passenger can inspect and select if there exists any campaign provided by an airline
- Any person can search for available tickets by its destination, source and date of departure
- Passenger can display booked tickets by himself/herself
- Passenger can display food and drink services for a flight
- Passenger can pay a ticket by different options like credit card, debit card, easy transfer etc.
- Passenger can display and select direct flights
- Passenger can display and select transfer point for connecting flight

REDUCED RELATIONAL MODEL

Entities:

Ticket(ticketId, status, needHelp, ticketTypeId, passangerId,creationDate, cancelDate, seatNumber, checkInDate, flightCode)

TicketType(ticketTypeId, cabinBaggage, baggage, typeName, canRefund, canPickSeat, airlineCode, menuId)

Passenger(passengerId, birthDate, gender, name, surname, email, phoneNum, passengerType)

Invoice(invoiceId, amount, ticketId, paymentType, paymentDate, passengerId, airlineCode)

Flight(flightCode, flightDate, source, destination, basePriceForBusiness, basePriceForEconomy, numberOfBusinessTickets, numberOfEconomyTickets, transferStatus, AirlineCode)

Food&Drink(menuId, menuItems, price)

Airline(airlineCode, airlineName, serviceFee, priceForStudent, priceForAdult, priceForChild, priceForOld, priceForTeen, priceForOld)

Transfer(transferId, transferPoint)

Campaign(campaignCode, startDate, expireDate, discount, description, airlineCode)

Relations :

~~Has(ticketTypeId, ticketId)~~

~~Has(ticketId, InvoiceId)~~

~~Has(airlineCode, ticketTypeId)~~

Has(FlightCode, transferId)

~~Treat(menuId, typeId)~~

~~Books(passangerId, ticketId, creationDate)~~

~~Cancels(passangerId, ticketId, cancelDate)~~

~~checksIn(passangerId, ticketId, seatNumber, checkInDate)~~

~~pays(passengerId, invoiceId, paymentType, paymentDate)~~

~~recieves(invoiceId, airlineCode)~~

~~belongs(ticketId, flightCode)~~

~~schedules(flightCode, AirlineCode)~~

~~creates(campaignCode, airlineCode)~~

serves(menuId, FligthCode)

POSSIBLE QUERIES

Q1) Show all tickets for a flight

Q2) Show all passengers of a flight

Q3) Show all ticket ids by its status and its type name

Q4) Show all passengers by his/her passenger type

Q5) Show all personal information of a passenger

Q6) Show all tickets booked by a passenger

Q7) Show all direct flights

Q8) Show all tickets from somewhere to somewhere

Q9) Show all canceled tickets

Q10) Show all airline names

Q11) Show all services fees of an airline

Q12) Show the descriptions of the campaign of an airline

Q13) Show all invoices received by an airline

TABLE CREATING CODES

```
CREATE TABLE TICKET(

ticketId NUMBER(10) NOT NULL,

status VARCHAR2(10),

needHelp NUMBER(1),

ticketTypeId VARCHAR2(10),

passengerId NUMBER(10),

creationDate DATE,

cancelDate DATE,

seatNumber VARCHAR2(5),

checkInDate DATE,

flightCode VARCHAR2(10),
CONSTRAINT ticketId_pk PRIMARY KEY (ticketId)
        USING INDEX ENABLE
);

CREATE TABLE TICKET_TYPE(
ticketTypeId VARCHAR2(10) NOT NULL,
cabinBaggage NUMBER(2),
baggage NUMBER(2),
typeName VARCHAR2(10),
canRefund NUMBER(1),
canPickSeat NUMBER(1),
airlineCode VARCHAR2(10),
menuId VARCHAR2(10),
CONSTRAINT ticketTypeId_pk PRIMARY KEY (ticketTypeId)
        USING INDEX ENABLE
);

CREATE TABLE PASSENGER(
passengerId NUMBER(10) NOT NULL,
birthDate DATE,
gender VARCHAR2(1),
name VARCHAR2(20),
surname VARCHAR2(20),
email VARCHAR2(50),
phoneNumber VARCHAR2(20),
passengerType VARCHAR2(10),
CONSTRAINT passengerId_pk PRIMARY KEY (passengerId)
        USING INDEX ENABLE
);

CREATE TABLE INVOICE(
invoiceId NUMBER(10) NOT NULL,
amount NUMBER(7, 2),
ticketId NUMBER(10),
passengerId NUMBER(10),
paymentType VARCHAR2(20),
paymentDate DATE,
airlineCode VARCHAR2(10),
CONSTRAINT invoiceId_pk PRIMARY KEY (invoiceId)
        USING INDEX ENABLE
);

CREATE TABLE FLIGHT(
flightCode VARCHAR2(10) NOT NULL,
flightDate DATE,
source VARCHAR2(20),
destination VARCHAR2(20),
basePriceForBusiness NUMBER(7),
basePriceForEconomy NUMBER(7),
numberOfBusinessTickets NUMBER(5),
numberOfEconomyTickets NUMBER(5),
transferStatus NUMBER(1),
airlineCode VARCHAR2(10),
CONSTRAINT flightCode_pk PRIMARY KEY (flightCode)
        USING INDEX ENABLE
);

CREATE TABLE FOOD_DRINK(
menuId VARCHAR2(10) NOT NULL,
menuItems VARCHAR2(70),
price NUMBER(7, 2),
CONSTRAINT menuId_pk PRIMARY KEY (menuId)
        USING INDEX ENABLE
);

CREATE TABLE AIRLINE(
airlineCode VARCHAR2(10) NOT NULL,
airlineName VARCHAR2(30),
serviceFee NUMBER(7, 2),
priceForStudent NUMBER(7),
priceForAdult NUMBER(7),
priceForChild NUMBER(7),
priceForOld NUMBER(7),
priceForBaby NUMBER(7),
priceForTeen NUMBER(7),
CONSTRAINT airlineCode_pk PRIMARY KEY (airlineCode)
        USING INDEX ENABLE
);

CREATE TABLE TRANSFER(
transferId NUMBER(10) NOT NULL,
transferPoint VARCHAR2(20),
CONSTRAINT transferId_pk PRIMARY KEY (transferId)
        USING INDEX ENABLE
);

CREATE TABLE CAMPAIGN(
campaignCode VARCHAR2(10) NOT NULL,
startDate DATE,
expireDate DATE,
discount NUMBER(7, 2),
description VARCHAR2(100),
airlineCode VARCHAR2(10),
CONSTRAINT campaignCode_pk PRIMARY KEY (campaignCode)
        USING INDEX ENABLE
);

CREATE TABLE Has(
flightCode VARCHAR2(10),
transferId NUMBER(10)
```

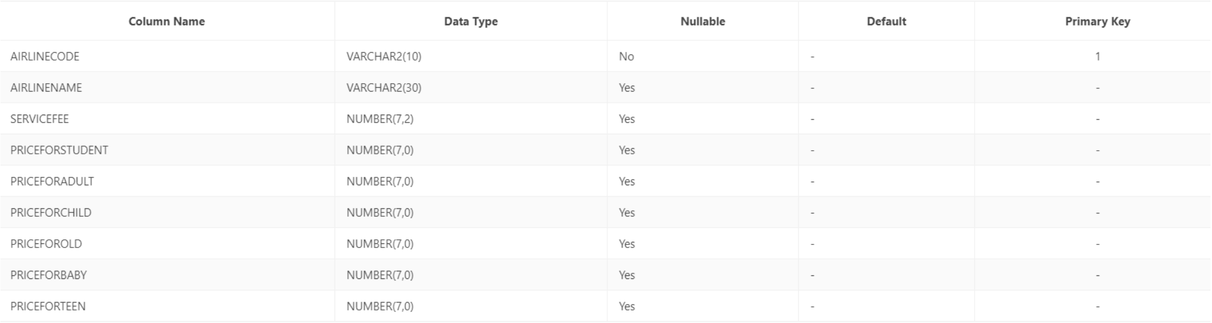
```
);


CREATE TABLE Serves(

menuId VARCHAR2(10),

flightCode VARCHAR2(10)

);


ALTER TABLE TICKET ADD CONSTRAINT Ticket_flightCode_fk FOREIGN KEY (flightCode)

   REFERENCES FLIGHT (flightCode);


ALTER TABLE TICKET ADD CONSTRAINT Ticket_ticketTypeId_fk FOREIGN KEY (ticketTypeId)

        REFERENCES TICKET_TYPE (ticketTypeId);


ALTER TABLE TICKET ADD CONSTRAINT Ticket_passengerId_fk FOREIGN KEY (passengerId)

        REFERENCES PASSENGER (passengerId);


ALTER TABLE TICKET_TYPE ADD CONSTRAINT Tickettype_airlineCode_fk FOREIGN KEY (airlineCode)

        REFERENCES AIRLINE (airlineCode);


ALTER TABLE TICKET_TYPE ADD CONSTRAINT Tickettype_menuId_fk FOREIGN KEY (menuId)

        REFERENCES FOOD_DRINK (menuId);
```

```
ALTER TABLE INVOICE ADD CONSTRAINT Invoice_ticketId_fk FOREIGN KEY
(ticketId)

        REFERENCES TICKET (ticketId);


ALTER TABLE INVOICE ADD CONSTRAINT Invoice_passengerId_fk FOREIGN KEY
(passengerId)

        REFERENCES PASSENGER (passengerId);


ALTER TABLE INVOICE ADD CONSTRAINT Invoice_airlineCode_fk FOREIGN KEY
(airlineCode)

        REFERENCES AIRLINE (airlineCode);


ALTER TABLE FLIGHT ADD CONSTRAINT Flight_airlineCode_fk FOREIGN KEY
(airlineCode)

        REFERENCES AIRLINE (airlineCode);


ALTER TABLE CAMPAIGN ADD CONSTRAINT Campaign_airlineCode_fk FOREIGN
KEY (airlineCode)

        REFERENCES AIRLINE (airlineCode);


ALTER TABLE Has ADD CONSTRAINT has_flightCode_fk FOREIGN KEY (flightCode)

        REFERENCES FLIGHT (flightCode);


ALTER TABLE Has ADD CONSTRAINT has_transferId_fk FOREIGN KEY (transferId)

        REFERENCES TRANSFER (transferId);


ALTER TABLE Serves ADD CONSTRAINT serves_flightCode_fk FOREIGN KEY
(flightCode)
```

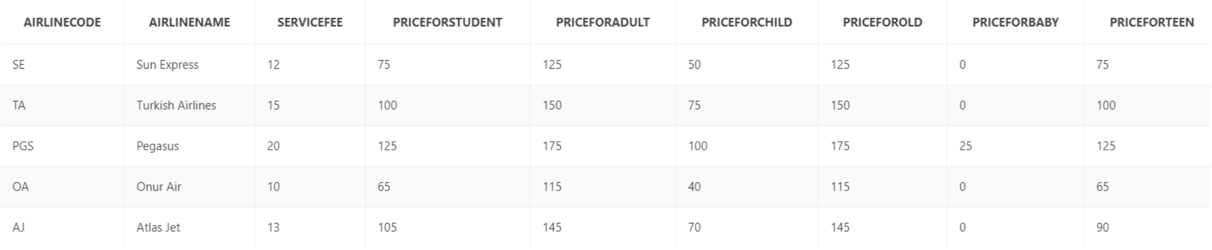
```
        REFERENCES FLIGHT (flightCode);


ALTER TABLE Serves ADD CONSTRAINT serves_menuId_fk FOREIGN KEY (menuId)
        REFERENCES FOOD_DRINK (menuId);
```

TABLES

AIRLINE

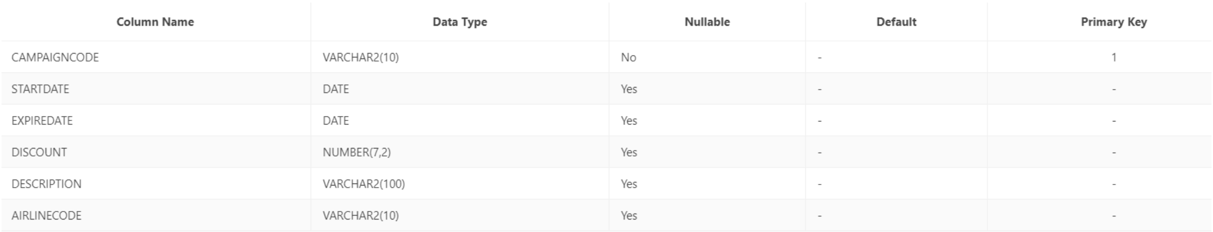


Column Name	Data Type	Nullable	Default	Primary Key
AIRLINECODE	VARCHAR2(10)	No	-	1
AIRLINENAME	VARCHAR2(30)	Yes	-	-
SERVICEFEE	NUMBER(7,2)	Yes	-	-
PRICEFORSTUDENT	NUMBER(7,0)	Yes	-	-
PRICEFORADULT	NUMBER(7,0)	Yes	-	-
PRICEFORCHILD	NUMBER(7,0)	Yes	-	-
PRICEFOROLD	NUMBER(7,0)	Yes	-	-
PRICEFORBABY	NUMBER(7,0)	Yes	-	-
PRICEFORTEEN	NUMBER(7,0)	Yes	-	-

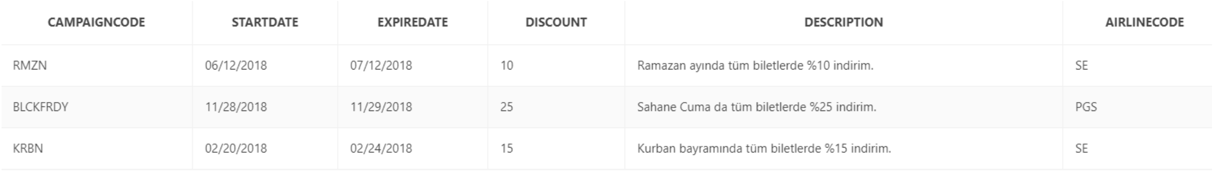


AIRLINECODE	AIRLINENAME	SERVICEFEE	PRICEFORSTUDENT	PRICEFORADULT	PRICEFORCHILD	PRICEFOROLD	PRICEFORBABY	PRICEFORTEEN
SE	Sun Express	12	75	125	50	125	0	75
TA	Turkish Airlines	15	100	150	75	150	0	100
PGS	Pegasus	20	125	175	100	175	25	125
OA	Onur Air	10	65	115	40	115	0	65
AJ	Atlas Jet	13	105	145	70	145	0	90

CAMPAIGN

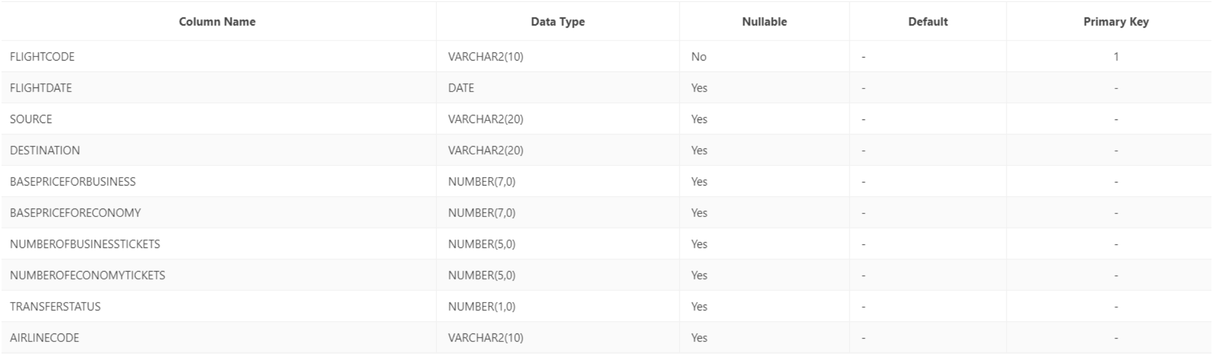


Column Name	Data Type	Nullable	Default	Primary Key
CAMPAIGNCODE	VARCHAR2(10)	No	-	1
STARTDATE	DATE	Yes	-	-
EXPIREDATE	DATE	Yes	-	-
DISCOUNT	NUMBER(7,2)	Yes	-	-
DESCRIPTION	VARCHAR2(100)	Yes	-	-
AIRLINECODE	VARCHAR2(10)	Yes	-	-

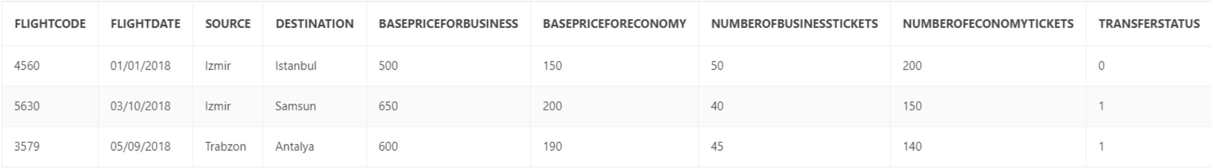


CAMPAIGNCODE	STARTDATE	EXPIREDATE	DISCOUNT	DESCRIPTION	AIRLINECODE
RMZN	06/12/2018	07/12/2018	10	Ramazan ayında tüm biletlerde %10 indirim.	SE
BLCKFRDY	11/28/2018	11/29/2018	25	Sahane Cuma da tüm biletlerde %25 indirim.	PGS
KRBN	02/20/2018	02/24/2018	15	Kurban bayramında tüm biletlerde %15 indirim.	SE

FLIGHT

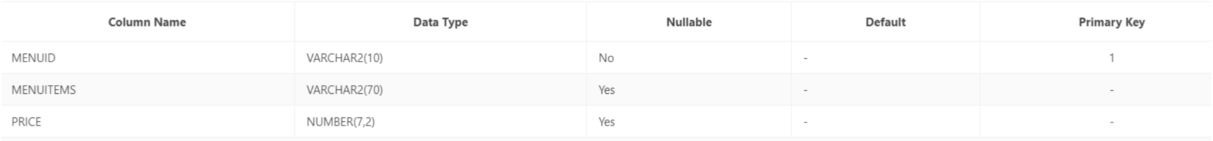


Column Name	Data Type	Nullable	Default	Primary Key
FLIGHTCODE	VARCHAR2(10)	No	-	1
FLIGHTDATE	DATE	Yes	-	-
SOURCE	VARCHAR2(20)	Yes	-	-
DESTINATION	VARCHAR2(20)	Yes	-	-
BASEPRICEFORBUSINESS	NUMBER(7,0)	Yes	-	-
BASEPRICEFORECONOMY	NUMBER(7,0)	Yes	-	-
NUMBEROFBUSINESSTICKETS	NUMBER(5,0)	Yes	-	-
NUMBEROFECONOMYTICKETS	NUMBER(5,0)	Yes	-	-
TRANSFERSTATUS	NUMBER(1,0)	Yes	-	-
AIRLINECODE	VARCHAR2(10)	Yes	-	-

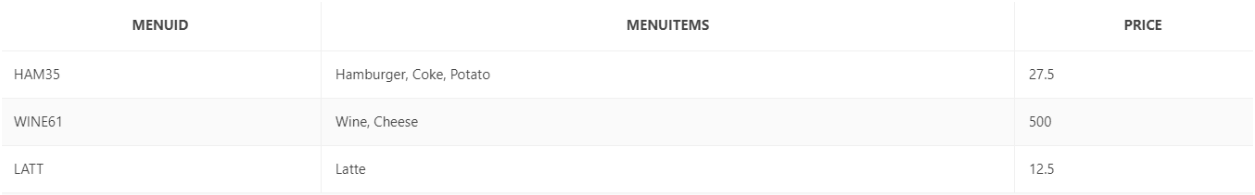


FLIGHTCODE	FLIGHTDATE	SOURCE	DESTINATION	BASEPRICEFORBUSINESS	BASEPRICEFORECONOMY	NUMBEROFBUSINESSTICKETS	NUMBEROFECONOMYTICKETS	TRANSFERSTATUS
4560	01/01/2018	Izmir	Istanbul	500	150	50	200	0
5630	03/10/2018	Izmir	Samsun	650	200	40	150	1
3579	05/09/2018	Trabzon	Antalya	600	190	45	140	1

FOOD&DRINK

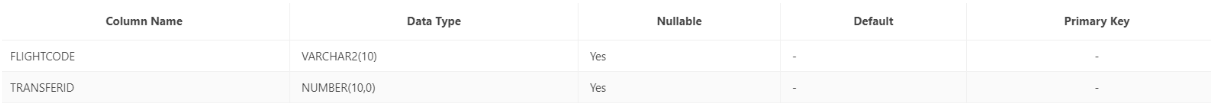


Column Name	Data Type	Nullable	Default	Primary Key
MENUID	VARCHAR2(10)	No	-	1
MENUITEMS	VARCHAR2(70)	Yes	-	-
PRICE	NUMBER(7,2)	Yes	-	-

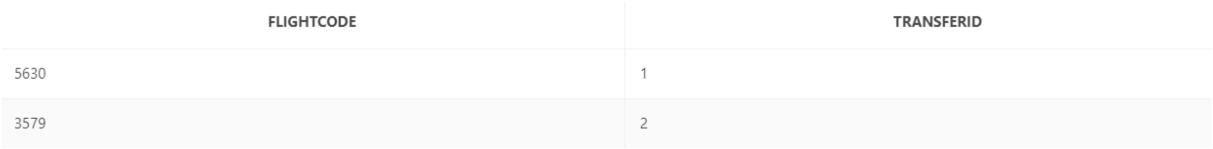


MENUID	MENUITEMS	PRICE
HAM35	Hamburger, Coke, Potato	27.5
WINE61	Wine, Cheese	500
LATT	Latte	12.5

HAS

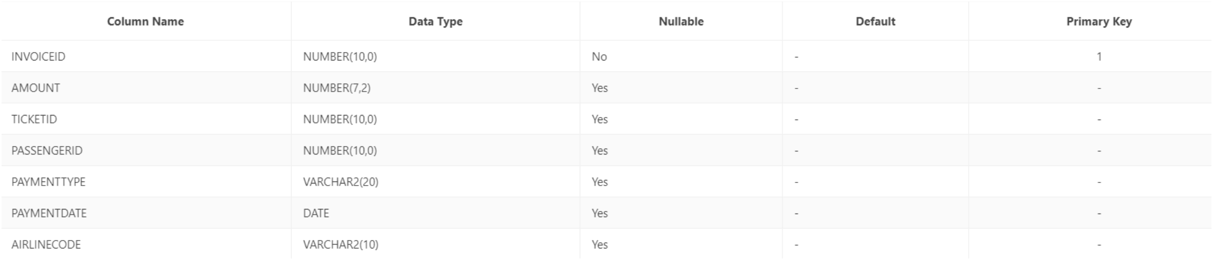


Column Name	Data Type	Nullable	Default	Primary Key
FLIGHTCODE	VARCHAR2(10)	Yes	-	-
TRANSFERID	NUMBER(10,0)	Yes	-	-

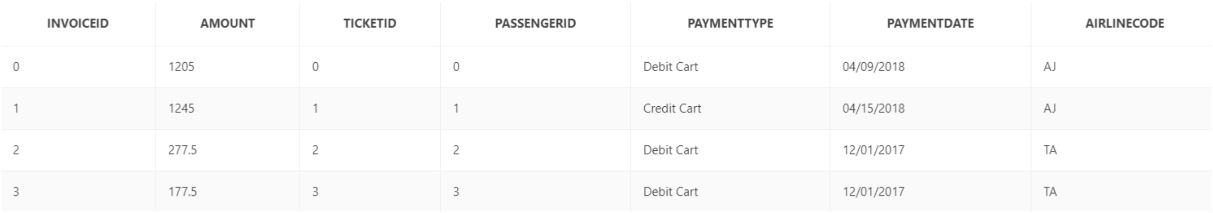


FLIGHTCODE	TRANSFERID
5630	1
3579	2

INVOICE

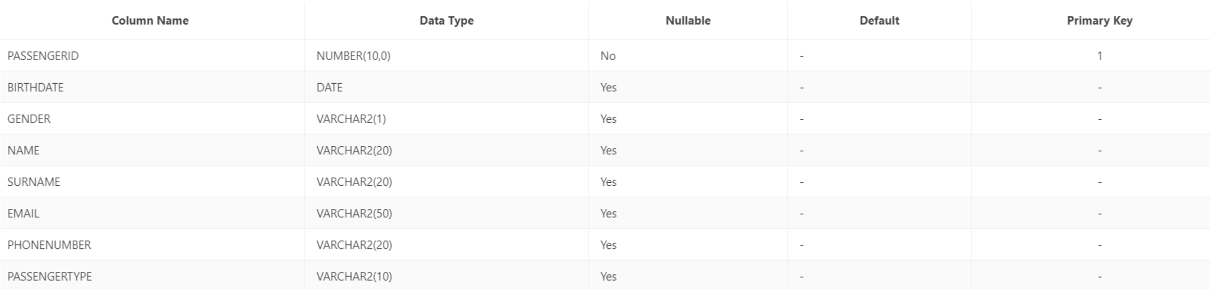


Column Name	Data Type	Nullable	Default	Primary Key
INVOICEID	NUMBER(10,0)	No	-	1
AMOUNT	NUMBER(7,2)	Yes	-	-
TICKETID	NUMBER(10,0)	Yes	-	-
PASSENGERID	NUMBER(10,0)	Yes	-	-
PAYMENTTYPE	VARCHAR2(20)	Yes	-	-
PAYMENTDATE	DATE	Yes	-	-
AIRLINECODE	VARCHAR2(10)	Yes	-	-

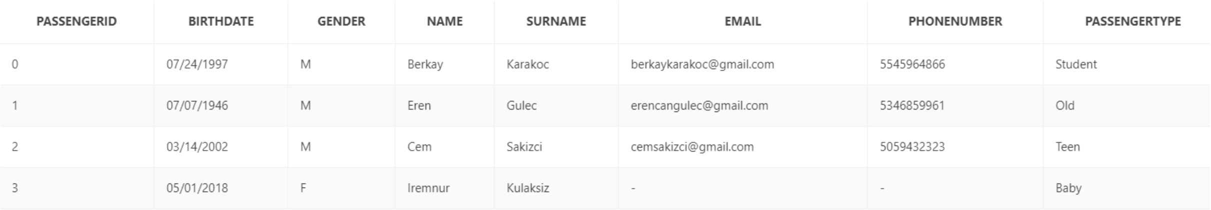


INVOICEID	AMOUNT	TICKETID	PASSENGERID	PAYMENTTYPE	PAYMENTDATE	AIRLINECODE
0	1205	0	0	Debit Cart	04/09/2018	AJ
1	1245	1	1	Credit Cart	04/15/2018	AJ
2	277.5	2	2	Debit Cart	12/01/2017	TA
3	177.5	3	3	Debit Cart	12/01/2017	TA

PASSENGER

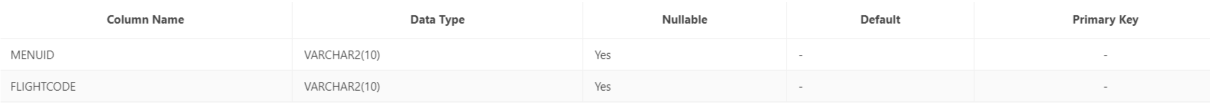


Column Name	Data Type	Nullable	Default	Primary Key
PASSENGERID	NUMBER(10,0)	No	-	1
BIRTHDATE	DATE	Yes	-	-
GENDER	VARCHAR2(1)	Yes	-	-
NAME	VARCHAR2(20)	Yes	-	-
SURNAME	VARCHAR2(20)	Yes	-	-
EMAIL	VARCHAR2(50)	Yes	-	-
PHONENUMBER	VARCHAR2(20)	Yes	-	-
PASSENGERTYPE	VARCHAR2(10)	Yes	-	-

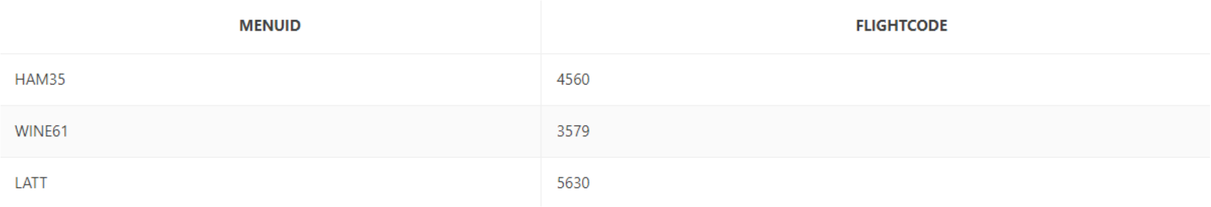


PASSENGERID	BIRTHDATE	GENDER	NAME	SURNAME	EMAIL	PHONENUMBER	PASSENGERTYPE
0	07/24/1997	M	Berkay	Karakoc	berkaykarakoc@gmail.com	5545964866	Student
1	07/07/1946	M	Eren	Gulec	erencangulec@gmail.com	5346859961	Old
2	03/14/2002	M	Cem	Sakizci	cemsakizci@gmail.com	5059432323	Teen
3	05/01/2018	F	Iremnur	Kulaksiz	-	-	Baby

SERVES

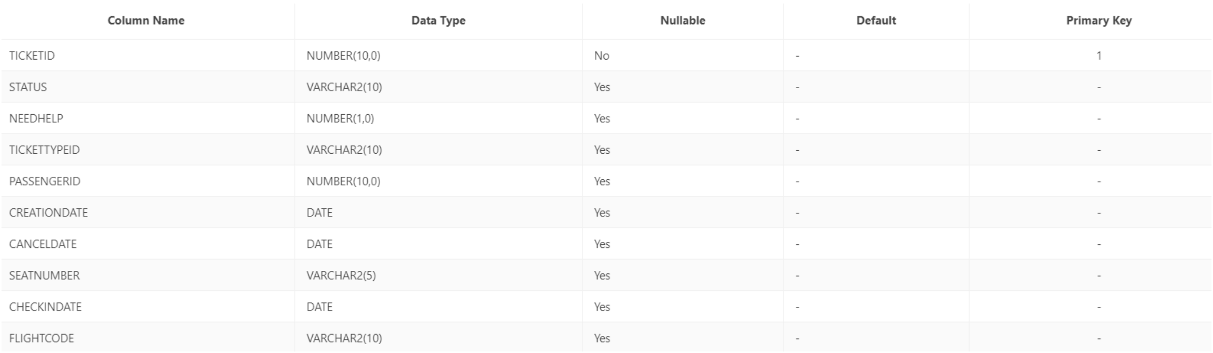


Column Name	Data Type	Nullable	Default	Primary Key
MENUID	VARCHAR2(10)	Yes	-	-
FLIGHTCODE	VARCHAR2(10)	Yes	-	-

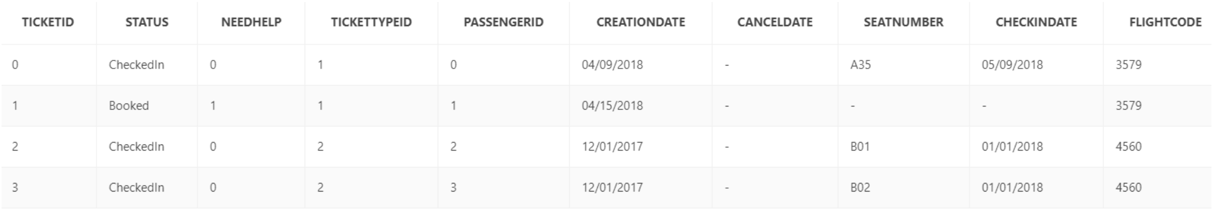


MENUID	FLIGHTCODE
HAM35	4560
WINE61	3579
LATT	5630

TICKET

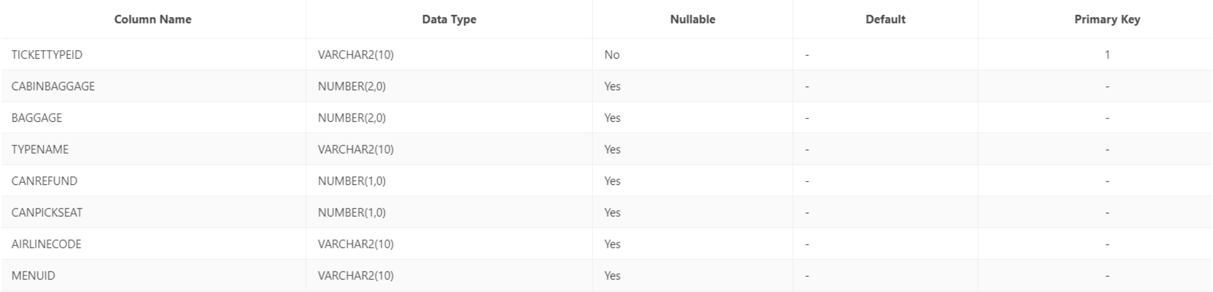


Column Name	Data Type	Nullable	Default	Primary Key
TICKETID	NUMBER(10,0)	No	-	1
STATUS	VARCHAR2(10)	Yes	-	-
NEEDHELP	NUMBER(1,0)	Yes	-	-
TICKETTYPEID	VARCHAR2(10)	Yes	-	-
PASSENGERID	NUMBER(10,0)	Yes	-	-
CREATIONDATE	DATE	Yes	-	-
CANCELDATE	DATE	Yes	-	-
SEATNUMBER	VARCHAR2(5)	Yes	-	-
CHECKINDATE	DATE	Yes	-	-
FLIGHTCODE	VARCHAR2(10)	Yes	-	-

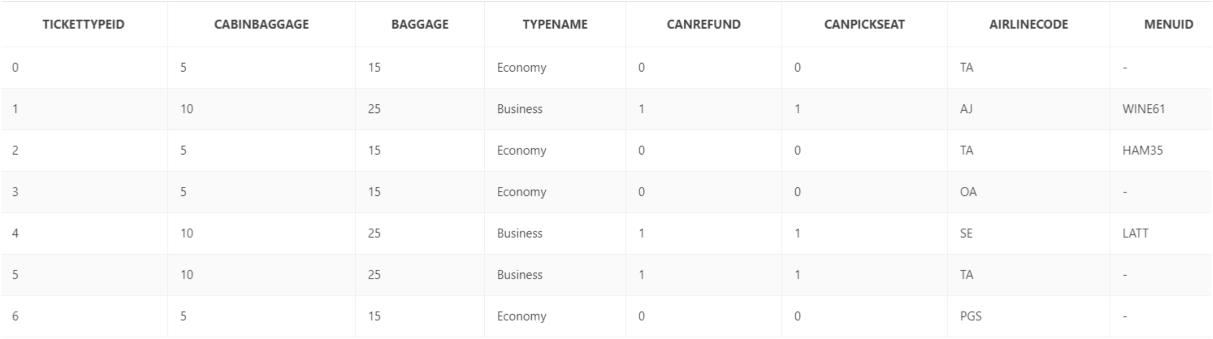


TICKETID	STATUS	NEEDHELP	TICKETTYPEID	PASSENGERID	CREATIONDATE	CANCELDATE	SEATNUMBER	CHECKINDATE	FLIGHTCODE
0	CheckedIn	0	1	0	04/09/2018	-	A35	05/09/2018	3579
1	Booked	1	1	1	04/15/2018	-	-	-	3579
2	CheckedIn	0	2	2	12/01/2017	-	B01	01/01/2018	4560
3	CheckedIn	0	2	3	12/01/2017	-	B02	01/01/2018	4560

TICKET_TYPE

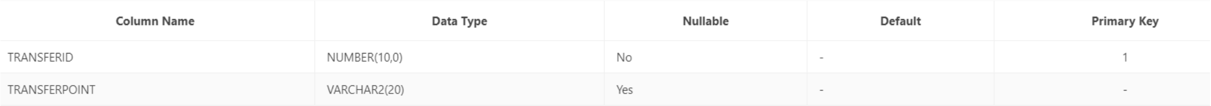


Column Name	Data Type	Nullable	Default	Primary Key
TICKETTYPEID	VARCHAR2(10)	No	-	1
CABINBAGGAGE	NUMBER(2,0)	Yes	-	-
BAGGAGE	NUMBER(2,0)	Yes	-	-
TYPENAME	VARCHAR2(10)	Yes	-	-
CANREFUND	NUMBER(1,0)	Yes	-	-
CANPICKSEAT	NUMBER(1,0)	Yes	-	-
AIRLINECODE	VARCHAR2(10)	Yes	-	-
MENUID	VARCHAR2(10)	Yes	-	-

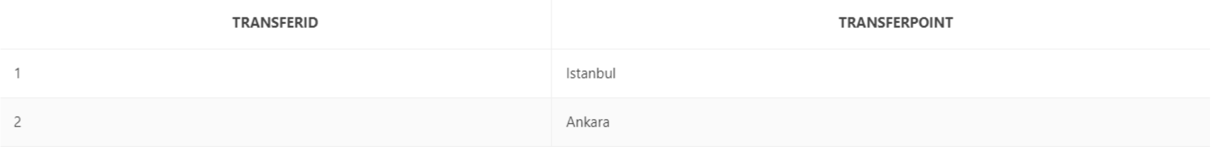


TICKETTYPEID	CABINBAGGAGE	BAGGAGE	TYPENAME	CANREFUND	CANPICKSEAT	AIRLINECODE	MENUID
0	5	15	Economy	0	0	TA	-
1	10	25	Business	1	1	AJ	WINE61
2	5	15	Economy	0	0	TA	HAM35
3	5	15	Economy	0	0	OA	-
4	10	25	Business	1	1	SE	LATT
5	10	25	Business	1	1	TA	-
6	5	15	Economy	0	0	PGS	-

TRANSFER



Column Name	Data Type	Nullable	Default	Primary Key
TRANSFERID	NUMBER(10,0)	No	-	1
TRANSFERPOINT	VARCHAR2(20)	Yes	-	-



TRANSFERID	TRANSFERPOINT
1	Istanbul
2	Ankara

ACTIVE TRIGGER EXAMPLES

1-) If a passenger books, cancels or checks-in a ticket, status of the ticket that the passenger has interacted should be updated.

```
UPDATE TICKET
SET status = <booked or available or cheked-in or paid>
WHERE ticketId = <ticket_id>;
```

2-) If a passenger uses a discount from campaign, amount in the invoice should be decreased according to the discount amount.

```
UPDATE INVOICE
SET amount = amount - (SELECT discount
                        FROM campaign
                        WHERE campaignCode = <campaign_code> ))
WHERE invoiceId = <invoice_id>
```

SQL Commands of Possible Queries

Q1) Show all tickets for a flight

```
SELECT TICKET.ticketId
FROM TICKET, FLIGHT
WHERE TICKET.flightCode=FLIGHT.flightCode;
```

Q2) Show all passengers of a flight

```
SELECT TICKET.passengerId
FROM TICKET, FLIGHT
WHERE TICKET.flightCode=FLIGHT.flightCode
AND (TICKET.status='Booked'
OR TICKET.status='CheckedIn');
```

Q3) Show all ticket ids by its status and its type name

```
SELECT TICKET.ticketId
FROM TICKET, TICKET_TYPE
WHERE status =<status>
AND TICKET_TYPE.typeName =<typeName>;
```

Q4) Show all passengers by his/her passenger type

```
SELECT passengerId
FROM PASSENGER
WHERE passengerType=<passengerType>;
```

Q5) Show all personal information of a passenger

```
SELECT *
FROM PASSENGER
WHERE passengerId=<passenger_id>;
```

Q6) Show all tickets booked by a passenger

```
SELECT ticketId
FROM TICKET
WHERE passengerId=<passenger_id>
AND status='Booked';
```

Q7) Show all direct flights

```
    SELECT *
    FROM FLIGHT
WHERE transferStatus=0;
```

Q8) Show all tickets from somewhere to somewhere

```
SELECT *
FROM FLIGHT
WHERE source=<source>
AND destination=<destination>;
```

Q9) Show all canceled tickets

```
SELECT *
FROM TICKET
WHERE cancelDate IS NOT NULL;
```

Q10) Show all airline names

```
SELECT airlineName
FROM AIRLINE;
```

Q11) Show all services fees of an airline

```
SELECT airlineName, serviceFee
FROM AIRLINE;
```

Q12) Show the descriptions of the campaign of an airline

```
SELECT description
FROM CAMPAIGN
WHERE airlineCode=<airline_code>;
```

Q13) Show all invoices received by an airline

```
SELECT *
FROM INVOICE
WHERE airlineCode=<airline_code>;
```

Changes From Initial Report

We needed to change our project’s topic according to our instructor’s feedbacks. We have changed database’s scope from Airport Management Database System to Flight Ticket Booking Website Database. Instead of City and Airport entities, we use Source and Destination in Flight entity. We have removed Plane, Store and Employee entities from our database system because we do not need to store any information related to Plane, Store and Employee. We modified Passenger, Flight, Airline and Ticket Entities. Additionally, we have added following entities: Ticket Type, Food and Drink, Invoice, Transfer and Campaign. Consequently, some relations have been changed, some of them have been removed and new ones have been created.