Ethiopian

የኢትዮጵያ

Ethiopian Airlines Database Design

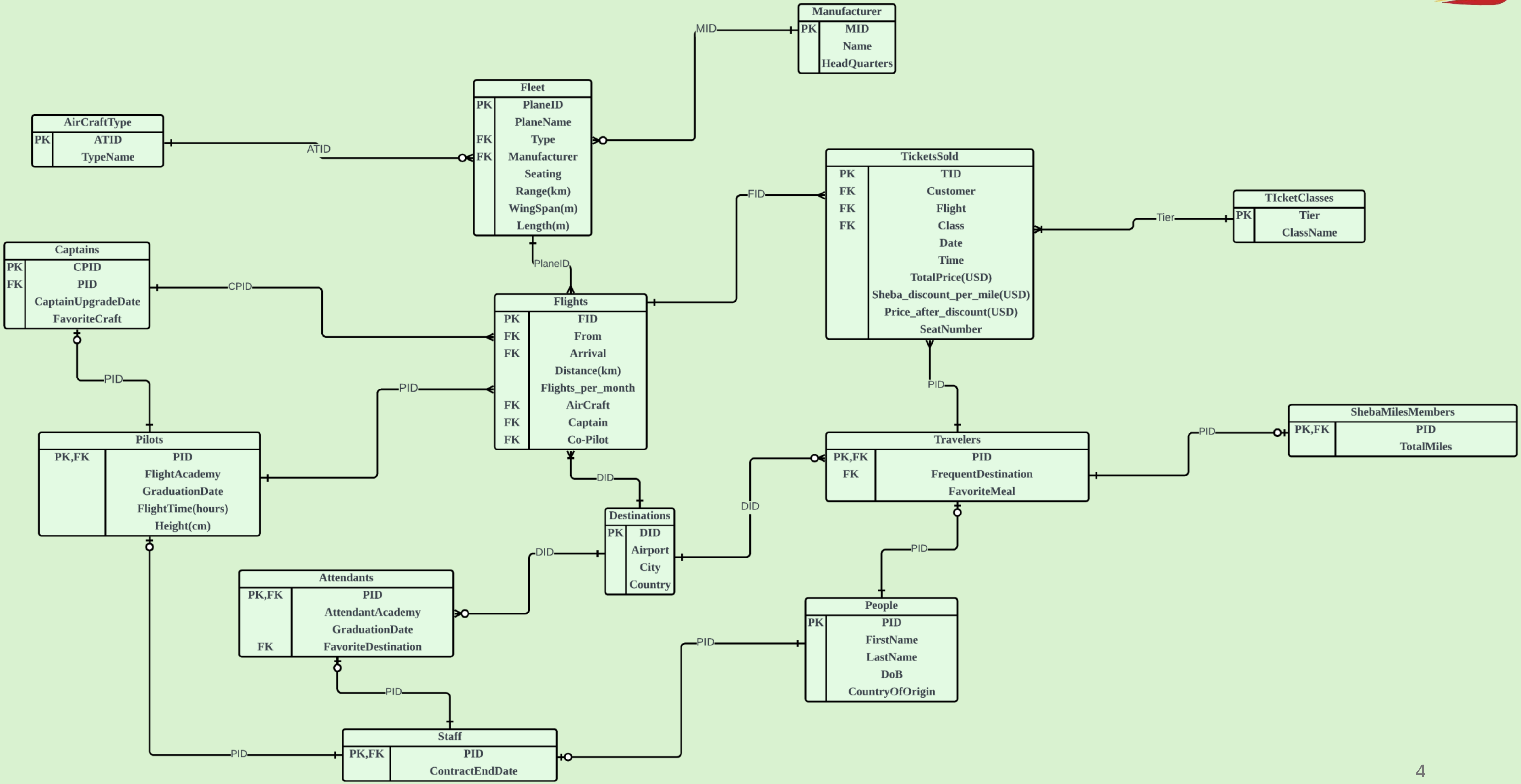
Robsen G. Gelalcha

Table of Contents

Executive Summary

- Ethiopian Airlines is the largest airline in Africa. Its award-winning crew and service are highly rated in the aviation world. Founded in 1945, in the more than seventy years of its existence, the airlines has both expanded into passenger and cargo services. It joined the Star Airline Alliance in 2011, becoming the first African airline to do so.
- This document presents a database proposal that seeks to organize and facilitate the airline’s daily operations. It outlines the different steps that go into the design and implementation of a database. The set of tables, views, triggers, stored procedures, and roles that are presented in the following pages seek to make the best use of the available data and to enable the airline to work in an efficient manner. In final implementation only eligible aircraft fly given flights and only club members have access to perks and discounts, ensuring only accurate and rule-abiding information is stored. The end goal is to provide an excellent database that will enable the airline live up to its slogan: “The new spirit of Africa.”

Entity-Relationship Diagram



	Manufacturer
PK	MID
	Name
	HeadQuarters

	Fleet
PK	PlaneID
	PlaneName
FK	Type
FK	Manufacturer
	Seating
	Range(km)
	WingSpan(m)
	Length(m)

	AirCraftType
PK	ATID
	TypeName

	TicketsSold
PK	TID
FK	Customer
FK	Flight
FK	Class
	Date
	Time
	TotalPrice(USD)
	Sheba_discount_per_mile(USD)
	Price_after_discount(USD)
	SeatNumber

	TIcketClasses
PK	Tier
	ClassName

	Captains
PK	CPID
FK	PID
	CaptainUpgradeDate
	FavoriteCraft

	Flights
PK	FID
FK	From
FK	Arrival
	Distance(km)
	Flights_per_month
FK	AirCraft
FK	Captain
FK	Co-Pilot

	Pilots
PK,FK	PID
	FlightAcademy
	GraduationDate
	FlightTime(hours)
	Height(cm)

	ShebaMilesMembers
PK,FK	PID
	TotalMiles

	Travelers
PK,FK	PID
FK	FrequentDestination
	FavoriteMeal

	Destinations
PK	DID
	Airport
	City
	Country

	Attendants
PK,FK	PID
	AttendantAcademy
	GraduationDate
FK	FavoriteDestination

	People
PK	PID
	FirstName
	LastName
	DoB
	CountryOfOrigin

	Staff
PK,FK	PID
	ContractEndDate

Tables

—

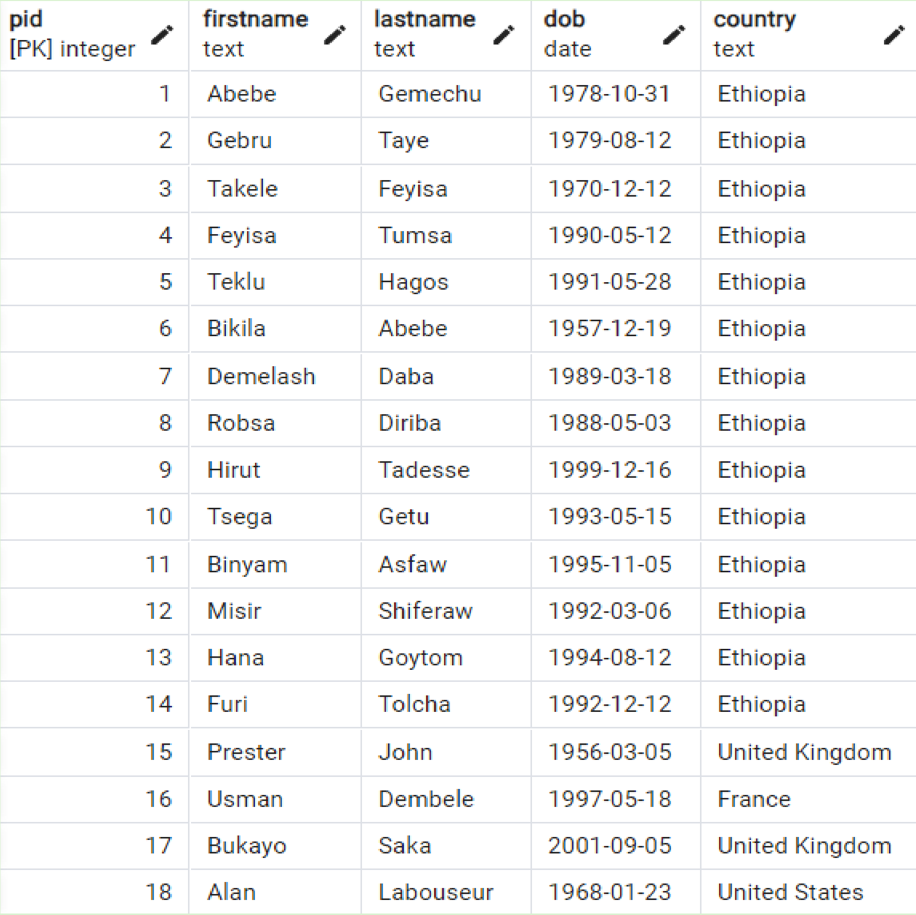
People

The people table represents a strong entity and includes all the basic attributes that are shared by all people subtypes: pilots, attendants, travelers, captains, and ShebaMileMembers.

```
 CREATE TABLE People (
 pid                            int not null,
 firstName                      text,
 lastName                       text,
 DoB                            date,
 Country                        text,
primary key(pid)
 );
```

Functional Dependencies:

pid→ firstName, lastName, DoB, Country



pid [PK] integer	firstname text	lastname text	dob date	country text
1	Abebe	Gemechu	1978-10-31	Ethiopia
2	Gebru	Taye	1979-08-12	Ethiopia
3	Takele	Feyisa	1970-12-12	Ethiopia
4	Feyisa	Tumsa	1990-05-12	Ethiopia
5	Teklu	Hagos	1991-05-28	Ethiopia
6	Bikila	Abebe	1957-12-19	Ethiopia
7	Demelash	Daba	1989-03-18	Ethiopia
8	Robsa	Diriba	1988-05-03	Ethiopia
9	Hirut	Tadesse	1999-12-16	Ethiopia
10	Tsega	Getu	1993-05-15	Ethiopia
11	Binyam	Asfaw	1995-11-05	Ethiopia
12	Misir	Shiferaw	1992-03-06	Ethiopia
13	Hana	Goytom	1994-08-12	Ethiopia
14	Furi	Tolcha	1992-12-12	Ethiopia
15	Prester	John	1956-03-05	United Kingdom
16	Usman	Dembele	1997-05-18	France
17	Bukayo	Saka	2001-09-05	United Kingdom
18	Alan	Labouseur	1968-01-23	United States

Sample data 1: People

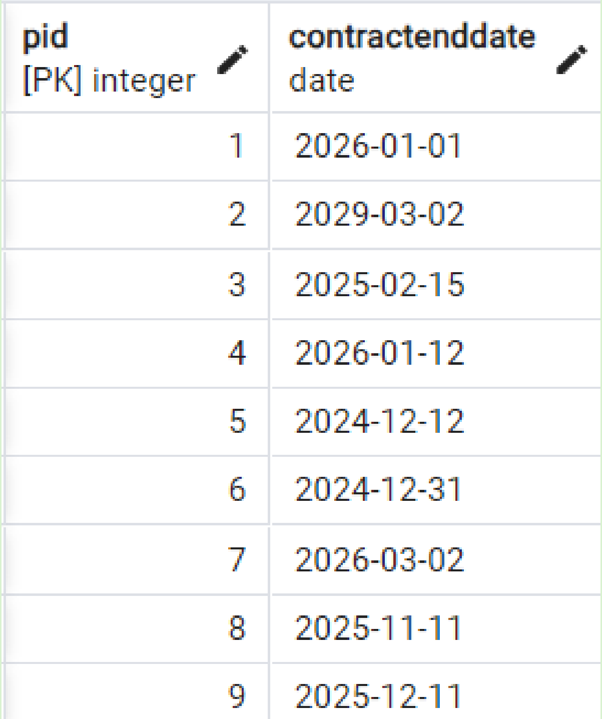
Staff

The staff table stores airline employees, i.e., attendants and pilots. In addition, it includes their contract end date.

```
CREATE TABLE Staff(
pid                 int not null references People(pid),
ContractEndDate     date,
primary key(pid));
```

Functional Dependencies:

pid→ ContractEndDate



pid [PK] integer	contractenddate date
1	2026-01-01
2	2029-03-02
3	2025-02-15
4	2026-01-12
5	2024-12-12
6	2024-12-31
7	2026-03-02
8	2025-11-11
9	2025-12-11

Sample Data 2: Staff

Destinations

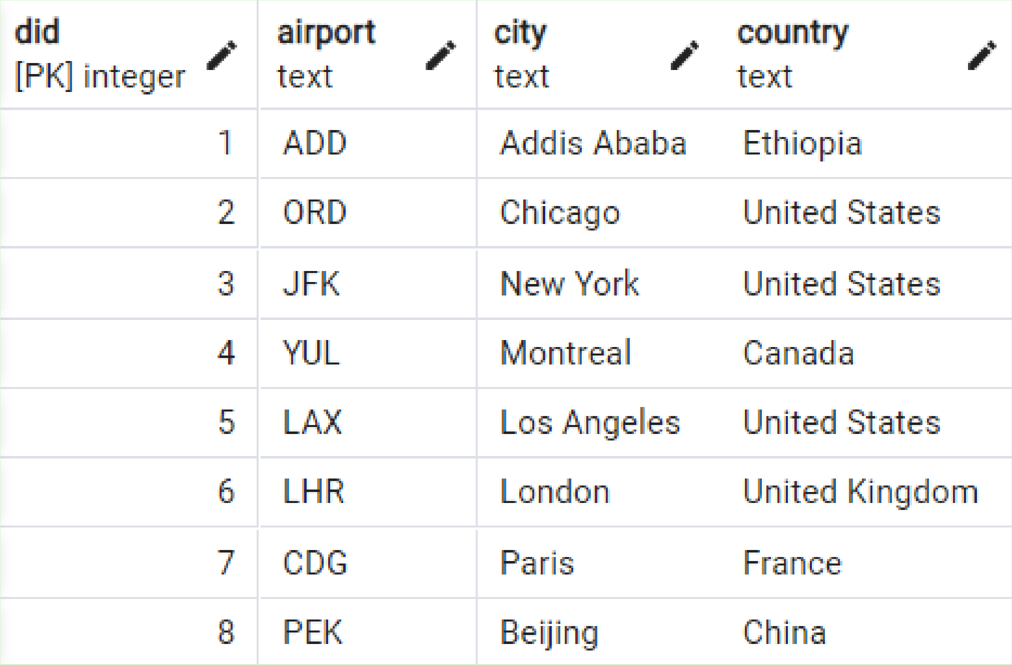
The destinations table stores a list of destinations that the airline flies to, along with necessary information about the destination.

```
CREATE TABLE Destinations(
did          int not null,
airport        text,
city           text,
country        text,
primary key(did));
```

Functional Dependencies:

did→ airport, city, country

Sample Data 3: Destinations



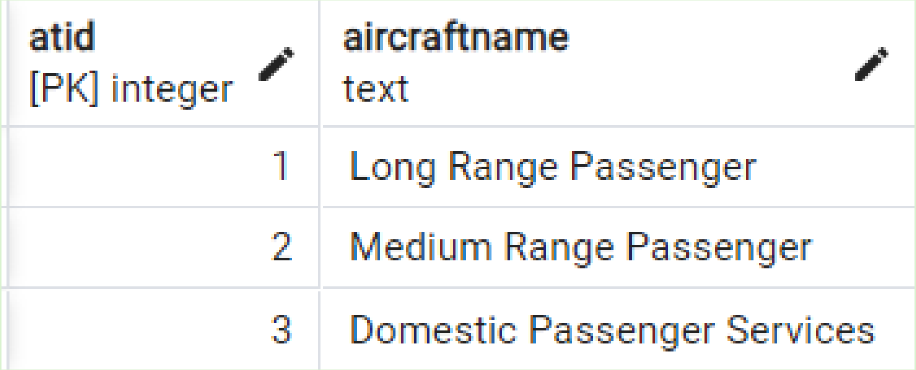
did [PK] integer	airport text	city text	country text
1	ADD	Addis Ababa	Ethiopia
2	ORD	Chicago	United States
3	JFK	New York	United States
4	YUL	Montreal	Canada
5	LAX	Los Angeles	United States
6	LHR	London	United Kingdom
7	CDG	Paris	France
8	PEK	Beijing	China

AircraftTypes

There are different types of aircraft, one being opted over another depending on the flight. The AircraftTypes stores the types of aircraft owned by the airline.

```
CREATE TABLE AircraftTypes(
atid            int not null,
AircraftName            text,
primary key(atid));
```

Sample Data 4: AircraftTypes



atid [PK] integer	aircraftname text
1	Long Range Passenger
2	Medium Range Passenger
3	Domestic Passenger Services

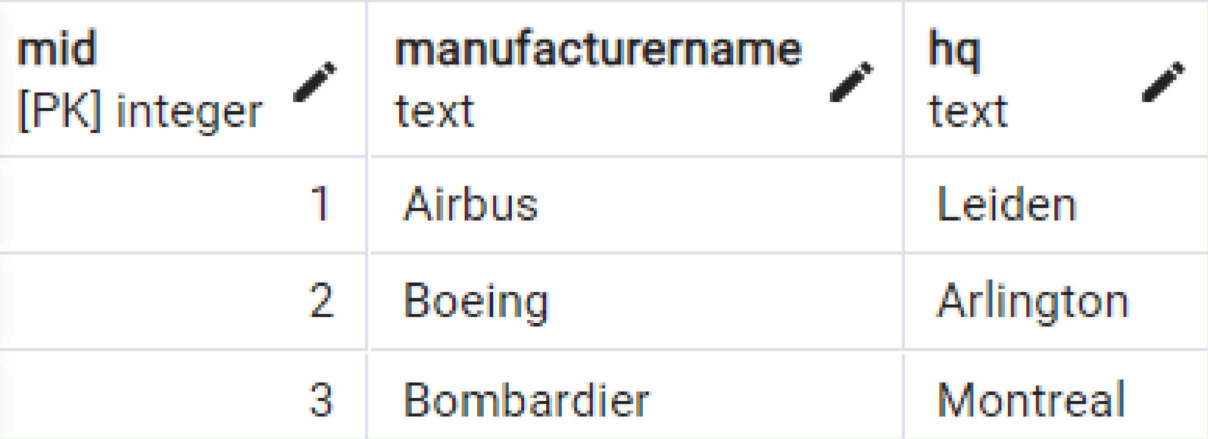
Functional Dependencies:

atid→ AircraftName

Manufacturers

As a major client to airline manufacturers, Ethiopian Airlines stores some vital information about all its suppliers. The Manufacturers table contains the names of aircraft manufacturers that supply to the airline’s fleet, along with their headquarters.

```
CREATE TABLE Manufacturers(
mid              int not null,
ManufacturerName  text,
hq               text,
primary key(mid));
```



mid [PK] integer	manufacturername text	hq text
1	Airbus	Leiden
2	Boeing	Arlington
3	Bombardier	Montreal

Sample Data 5: Manufacturers

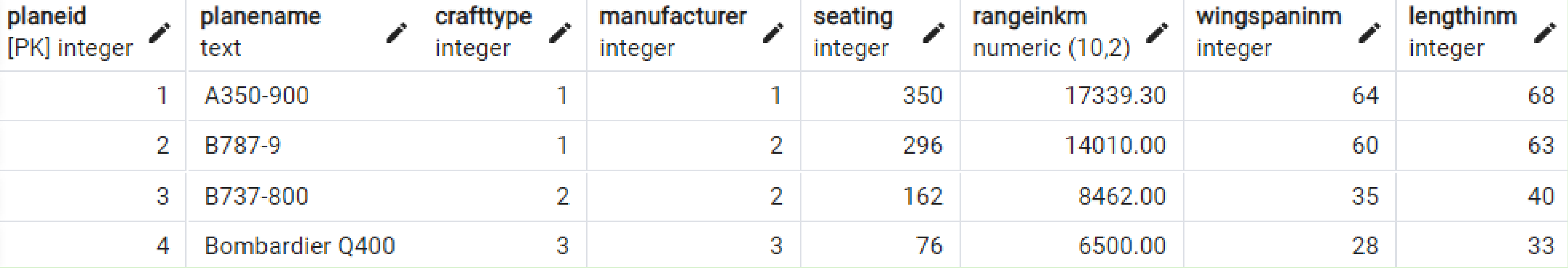
Functional Dependencies:

mid→ ManufacturerName, hq

Fleet

- The fleet table stores all the airplanes owned and operated by the airline. It also includes additional information about each plane: dimensions, number of seats, type, and manufacturer.

```
CREATE TABLE Fleet(
planeID       int not null,
PlaneName     text,
CraftType        int not null references AircraftTypes(atid),
Manufacturer   int not null references Manufacturers(mid),
Seating       int,
RangeInKM     decimal(10,2),
WingspanInM    int,
lengthInM      int,
primary key(planeID)
);
```



planeid [PK] integer	planename text	crafttype integer	manufacturer integer	seating integer	rangeinkm numeric (10,2)	wingspaninm integer	lengthinm integer
1	A350-900	1	1	350	17339.30	64	68
2	B787-9	1	2	296	14010.00	60	63
3	B737-800	2	2	162	8462.00	35	40
4	Bombardier Q400	3	3	76	6500.00	28	33

Sample Data 6: Fleet

Functional Dependencies: Fleet→ planeID, PlaneName, CraftType, Manufacturer, Seating, RangeInKM, WingspanInM, lengthInM

Classes

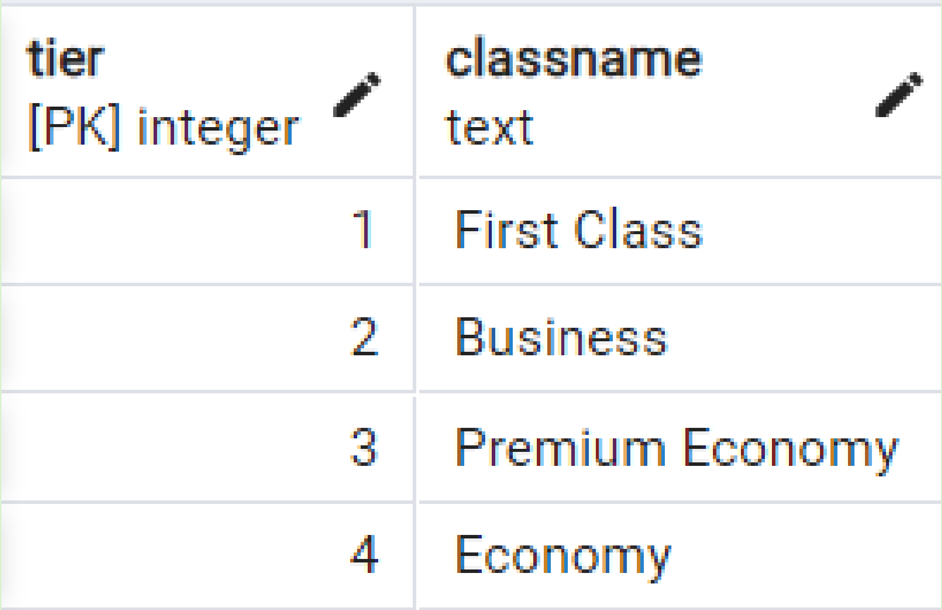
Whenever customers look to buy a plane ticket, they are offered four different ticket tiers. And this table stores each tier of tickets.

```
CREATE TABLE Classes(
Tier          int not null,
className     text,
primary key(tier));
```

Functional Dependencies:

Tier→ className

Sample Data 7: Classes

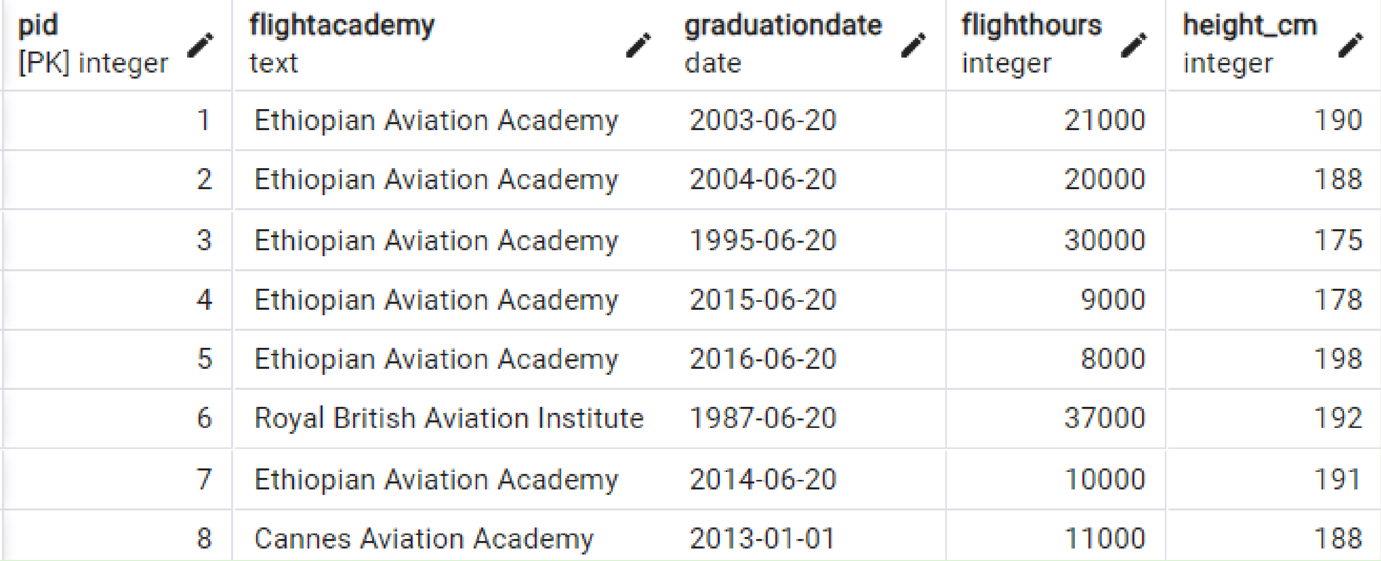


tier [PK] integer	classname text
1	First Class
2	Business
3	Premium Economy
4	Economy

Pilots

As of now, we do not have self-flying planes. Therefore, we must utilize people. The pilots table stores all the staff members that work as pilots. In addition to their people data in the People's table, the pilot's data has vital information about the pilots' profession: the academy they went to and graduation date, their total flight hours, and their height in cm.

```
CREATE TABLE Pilots(
PID       int not null references Staff(pid),
FlightAcademy       text,
GraduationDate       date,
FlightHours     int,
Height_cm          int,
primary key(PID));
```



pid [PK] integer	flightacademy text	graduationdate date	flighthours integer	height_cm integer
1	Ethiopian Aviation Academy	2003-06-20	21000	190
2	Ethiopian Aviation Academy	2004-06-20	20000	188
3	Ethiopian Aviation Academy	1995-06-20	30000	175
4	Ethiopian Aviation Academy	2015-06-20	9000	178
5	Ethiopian Aviation Academy	2016-06-20	8000	198
6	Royal British Aviation Institute	1987-06-20	37000	192
7	Ethiopian Aviation Academy	2014-06-20	10000	191
8	Cannes Aviation Academy	2013-01-01	11000	188

Sample Data 8: Pilots

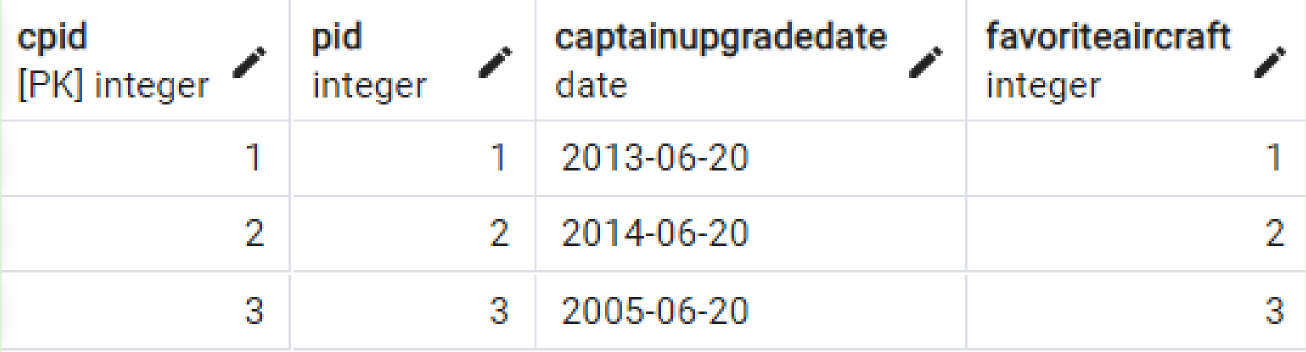
Functional Dependencies:

PID→ FlightAcademy, GraduationDate, FlightHours, Height_cm

Captains

The captains are a pilot subtype. These pilots have more experience, money. The captains’ table has a special primary key, CPID. In addition, it includes other helpful information about the Captains: their promotion to captain date, and their favorite aircraft.

```
CREATE TABLE Captains
(CPID      int not null,
PID       int not null references Pilots(PID),
CaptainUpgradeDate      date,
FavoriteAircraft       int references Fleet(planeID),
primary key(CPID));
```



cpid [PK] integer	pid integer	captainupgradedate date	favoriteaircraft integer
1	1	2013-06-20	1
2	2	2014-06-20	2
3	3	2005-06-20	3

Sample Data 9: Captains

Functional Dependencies:

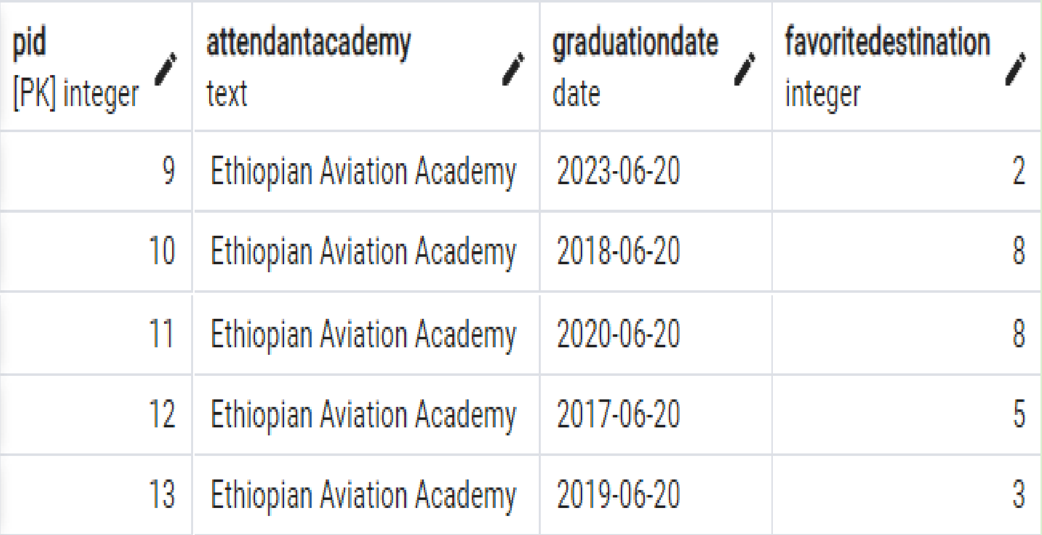
CPID→ PID, CaptainUpgradeDate, FavoriteAircraft

Attendants

Attendants are the other staff members that are vital for a flight. The attendants table stores all employees that work as flight attendants. It also provides other interesting information like their favorite destination.

```
CREATE TABLE Attendants(
PID        int not null references People(PID),
AttendantAcademy text,
GraduationDate   date,
FavoriteDestination int references Destinations(did),
primary key(PID));
```

Sample Data 10: Attendants



pid [PK] integer	attendantacademy text	graduationdate date	favoritedestination integer
9	Ethiopian Aviation Academy	2023-06-20	2
10	Ethiopian Aviation Academy	2018-06-20	8
11	Ethiopian Aviation Academy	2020-06-20	8
12	Ethiopian Aviation Academy	2017-06-20	5
13	Ethiopian Aviation Academy	2019-06-20	3

Functional Dependencies:

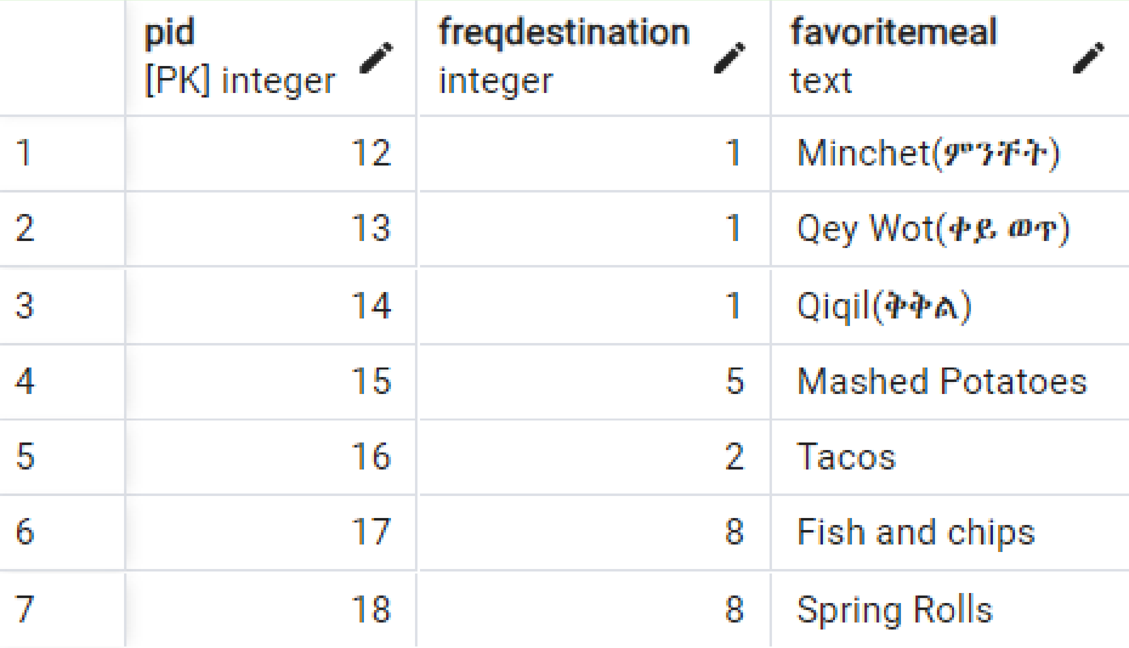
PID→ AttendantAcademy, GraduationDate, FavoriteDestination

Travelers

The existence of travelers is what birthed airlines. Ethiopian airlines, as it is the best airline on its continent, keeps necessary information about its passengers, aka travelers. This table, a subtype of people, includes information that can be used to predict and prepare for a customer’s next trip.

```
CREATE TABLE Travelers(
pid          int not null references People(PID),
FreqDestination int references Destinations(DID),
FavoriteMeal   text,
Primary key(pid));
```

Sample Data 11: Travelers



	pid [PK] integer	freqdestination integer	favoritemeal text
1	12	1	Minchet(ምንቸት)
2	13	1	Qey Wot(ቀይ ወጥ)
3	14	1	Qiqil(ቅቅል)
4	15	5	Mashed Potatoes
5	16	2	Tacos
6	17	8	Fish and chips
7	18	8	Spring Rolls

Functional Dependencies:

Pid→ FreqDestination, FavoriteMeal

* Do not be scared by the weird looking characters on the table. It is the Geez script, the only African alphabet that is officially in use. Input this to Google translate: በአላን የተሰጠውን የውሂብ ጎታ ክፍል እወዳለሁ።

ShebaMembers

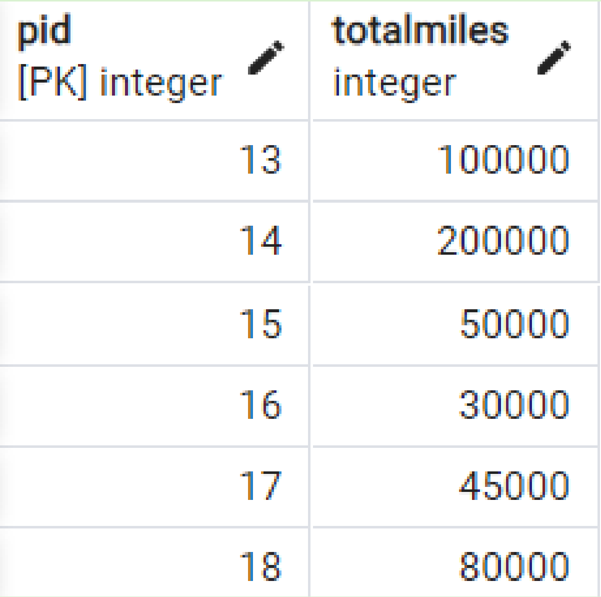
Sheba is the mile service offered by Ethiopian Airlines. ShebaMembers table is a subtype of travelers, and it stores the total miles accumulated by its members.

```
CREATE TABLE ShebaMembers(
pid      int not null references Travelers(pid),
TotalMiles  int,
primary key(pid));
```

Functional Dependencies:

Pid→ TotalMiles

Sample Data 12: ShebaMembers

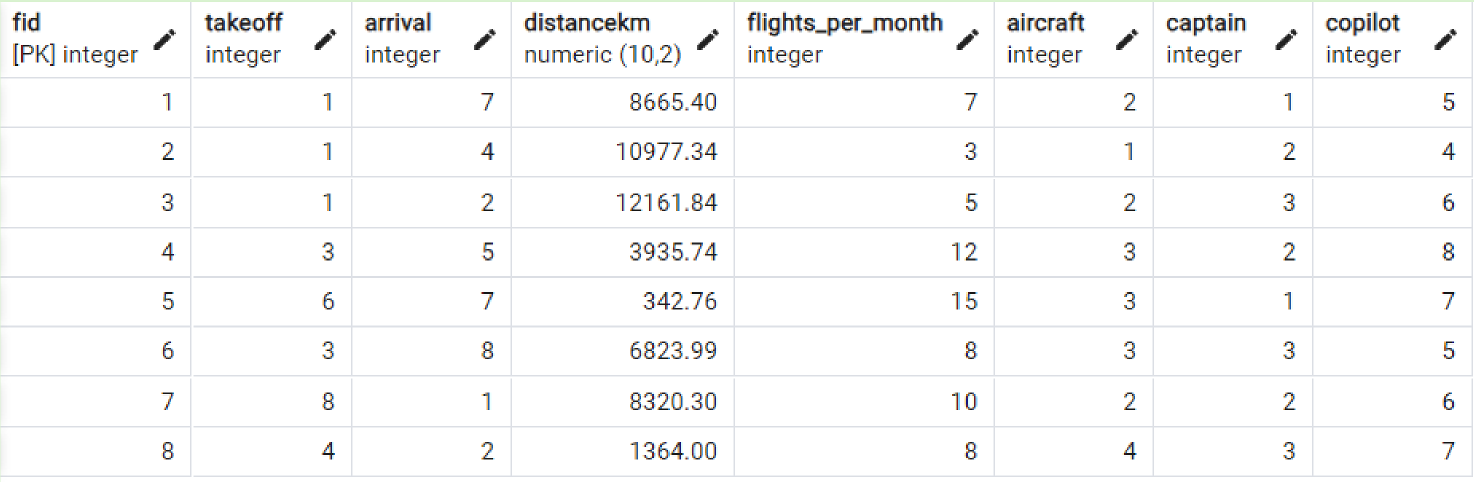


pid [PK] integer	totalmiles integer
13	100000
14	200000
15	50000
16	30000
17	45000
18	80000

Flights

The flights table stores all flights that the airline offers. In addition, it includes necessary information about each flight. It references many tables that were created before it, most notably the destinations table.

```
CREATE TABLE Flights(
fid       int not null,
Takeoff   int not null references Destinations(did),
arrival   int not null references Destinations(did),
distanceKM  decimal(10,2),
Flights_per_month  int,
Aircraft    int references Fleet(planeID),
Captain     int references Captains(CPID),
CoPilot     int references Pilots(pid),
primary key(fid));
```



fid [PK] integer	takeoff integer	arrival integer	distancekm numeric (10,2)	flights_per_month integer	aircraft integer	captain integer	copilot integer
1	1	7	8665.40	7	2	1	5
2	1	4	10977.34	3	1	2	4
3	1	2	12161.84	5	2	3	6
4	3	5	3935.74	12	3	2	8
5	6	7	342.76	15	3	1	7
6	3	8	6823.99	8	3	3	5
7	8	1	8320.30	10	2	2	6
8	4	2	1364.00	8	4	3	7

Sample Data 13: Flights

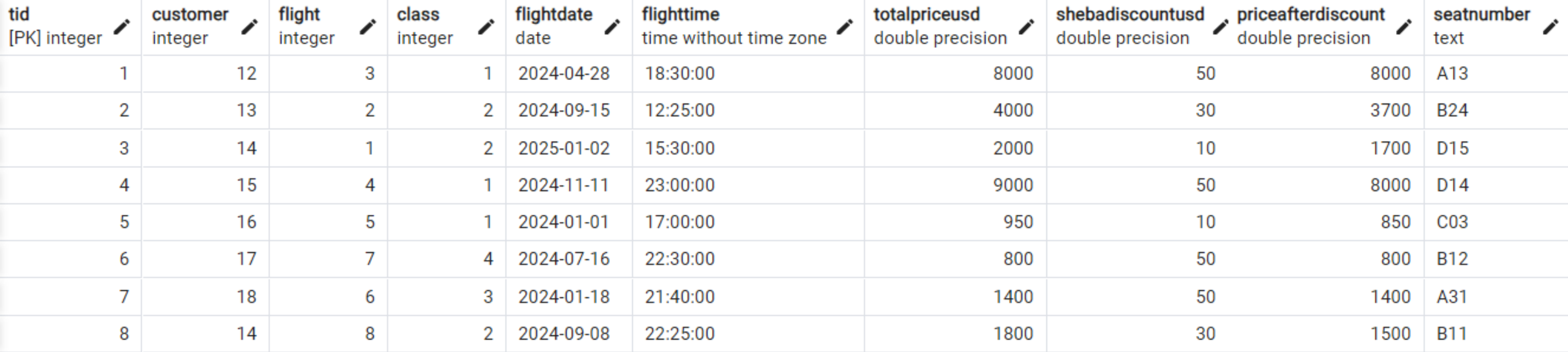
Functional Dependencies:

Fid→ Takeoff, arrival, distanceKM, Flights_per_month, Aircraft, Captain, CoPilot

Tickets

The tickets table stores all the tickets that are sold by the airlines. The ticket rows are both composed of what we would normally see on our boarding passes and other information like price. The 'ShebaDiscountUSD' column indicates the discount rate that can be redeemed for every hundred miles.

```
CREATE TABLE Tickets(
tid        int not null,
Customer   int not null references Travelers(pid),
Flight     int not null references Flights(fid),
Class      int not null references Classes(tier),
FlightDate     date,
FlightTime     time,
TotalPriceUSD   float,
ShebaDiscountUSD float,
PriceAfterDiscount float,
SeatNumber   text,
primary key(tid));
```

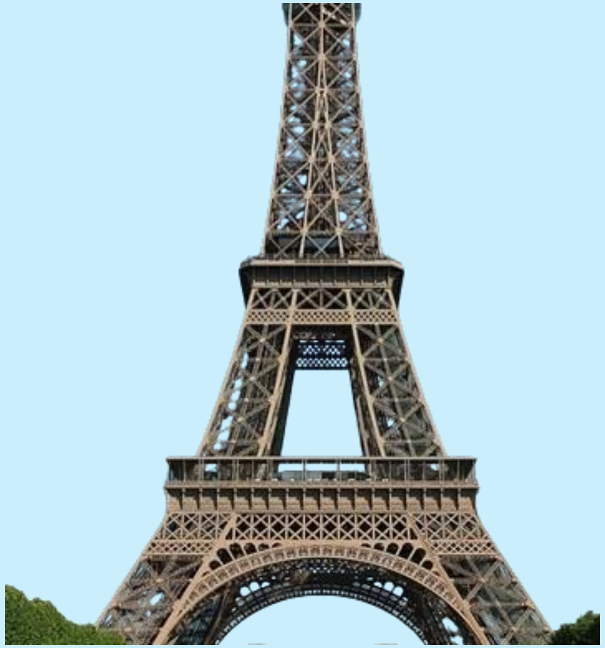


tid [PK] integer	customer integer	flight integer	class integer	flightdate date	flighttime time without time zone	totalpriceusd double precision	shebadiscountusd double precision	priceafterdiscount double precision	seatnumber text
1	12	3	1	2024-04-28	18:30:00	8000	50	8000	A13
2	13	2	2	2024-09-15	12:25:00	4000	30	3700	B24
3	14	1	2	2025-01-02	15:30:00	2000	10	1700	D15
4	15	4	1	2024-11-11	23:00:00	9000	50	8000	D14
5	16	5	1	2024-01-01	17:00:00	950	10	850	C03
6	17	7	4	2024-07-16	22:30:00	800	50	800	B12
7	18	6	3	2024-01-18	21:40:00	1400	50	1400	A31
8	14	8	2	2024-09-08	22:25:00	1800	30	1500	B11

Sample Data 14: Tickets

Functional Dependencies:

Tid→ Customer, Flight, Class, FlightDate, FlightTime, TotalPriceUSD, ShebaDiscountUSD, PriceAfterDiscount,SeatNumber

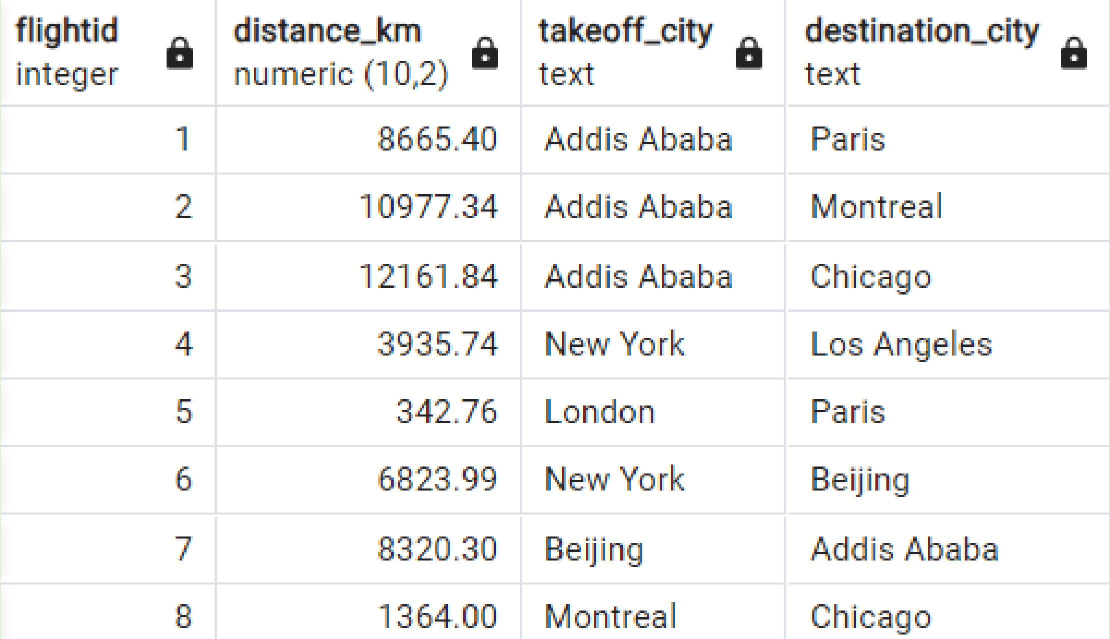


VIEWS

FlightDetails

Displays the distance, takeoff city name, and destination city name of all flights.

```
CREATE VIEW FlightDetails AS
SELECT flights.fid AS FlightID,
    flights.distancekm AS Distance_KM,
    takeoff.city AS Takeoff_City,
    arrival.city AS Destination_City
FROM Flights
JOIN Destinations takeoff ON flights.Takeoff = takeoff.did
JOIN Destinations arrival ON flights.arrival = arrival.did;
```

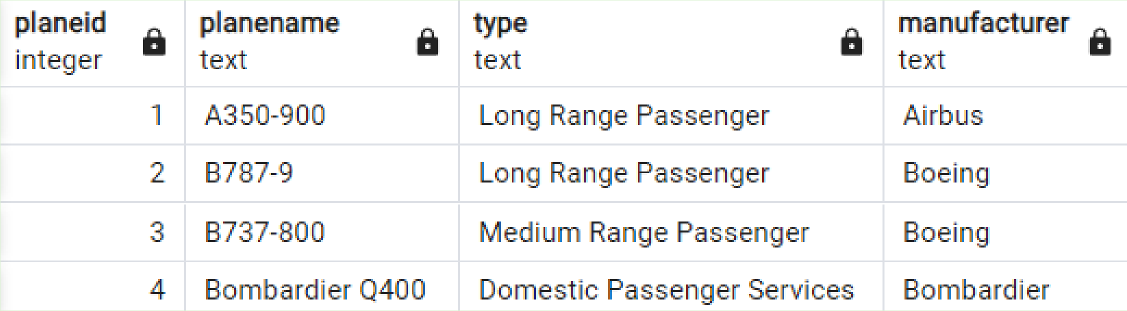


flightid integer	distance_km numeric (10,2)	takeoff_city text	destination_city text
1	8665.40	Addis Ababa	Paris
2	10977.34	Addis Ababa	Montreal
3	12161.84	Addis Ababa	Chicago
4	3935.74	New York	Los Angeles
5	342.76	London	Paris
6	6823.99	New York	Beijing
7	8320.30	Beijing	Addis Ababa
8	1364.00	Montreal	Chicago

Plane_Type_and_Manufacturer

This view displays basic details, along with aircraft names, about each aircraft owned and operated by the airlines: aircraft type and manufacturer

```
CREATE VIEW Plane_Type_And_Manufacturer AS
SELECT fleet.planeid as PlaneID,
            fleet.planename as PlaneName,
            aircrafttypes.aircraftname as Type,
            manufacturers.manufacturername as Manufacturer
FROM fleet
Join AircraftTypes on fleet.crafttype=aircrafttypes.atid
Join Manufacturers on fleet.manufacturer=manufacturers.mid;
```

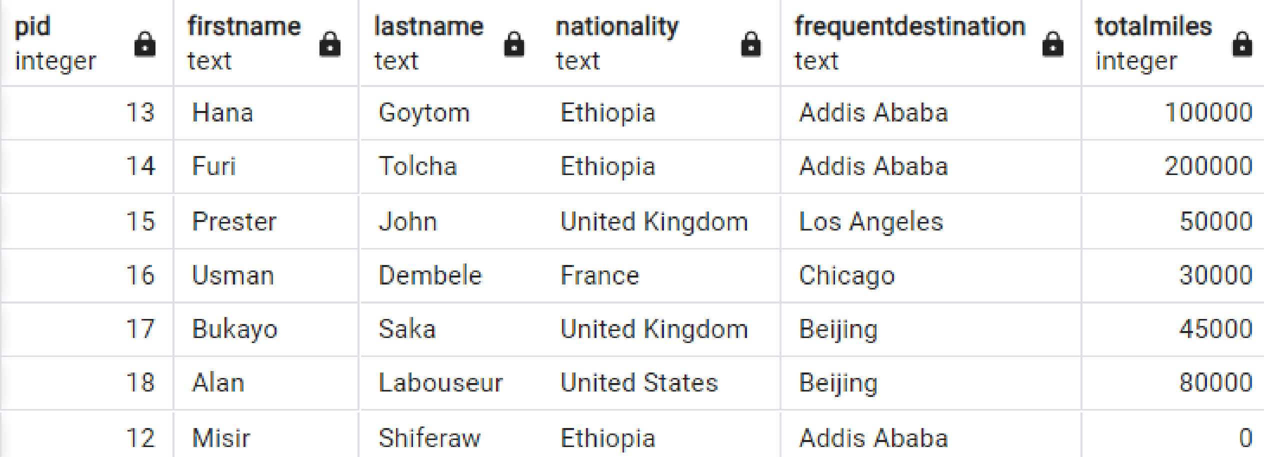


planeid integer	planename text	type text	manufacturer text
1	A350-900	Long Range Passenger	Airbus
2	B787-9	Long Range Passenger	Boeing
3	B737-800	Medium Range Passenger	Boeing
4	Bombardier Q400	Domestic Passenger Services	Bombardier

TravelerDetails

This view displays some necessary information about all travelers, regardless of their Sheba membership.

```
CREATE VIEW TravelerDetails AS
SELECT p.pid, p.firstname, p.lastname, p.country AS Nationality,
    d.city AS FrequentDestination,
    COALESCE(shebaMembers.totalmiles, 0) AS TotalMiles
FROM People p
JOIN travelers t ON t.pid = p.pid
JOIN destinations d ON t.freqdestination = d.did
LEFT JOIN shebamembers ON shebamembers.pid = t.pid;
```



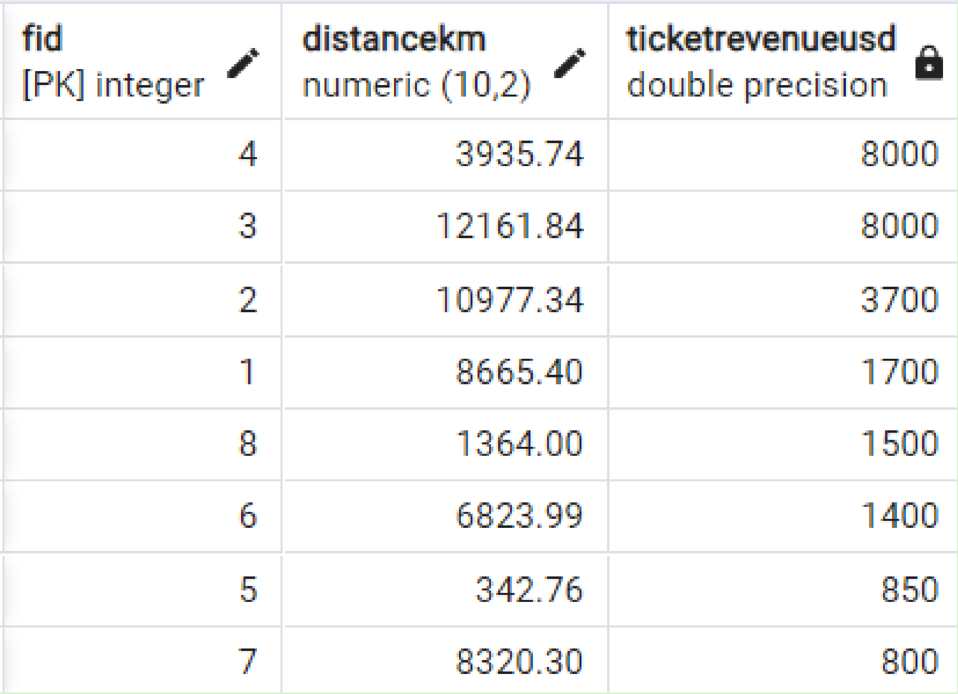
pid integer	firstname text	lastname text	nationality text	frequentdestination text	totalmiles integer
13	Hana	Goytom	Ethiopia	Addis Ababa	100000
14	Furi	Tolcha	Ethiopia	Addis Ababa	200000
15	Prester	John	United Kingdom	Los Angeles	50000
16	Usman	Dembele	France	Chicago	30000
17	Bukayo	Saka	United Kingdom	Beijing	45000
18	Alan	Labouseur	United States	Beijing	80000
12	Misir	Shiferaw	Ethiopia	Addis Ababa	0

QUERIES/REPORTS

Ticket Revenue From Each Flight

This query returns the total revenue from each flight. Distance is also included to add more context to the money made.

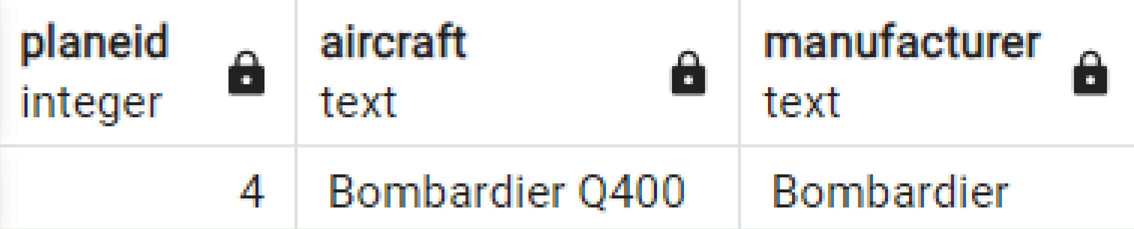
```
select flights.fid, flights.distancekm as DistanceKM,
sum(tickets.priceafterdiscount) as TicketRevenueUSD
From flights
Join tickets on flights.fid=tickets.flight
group by flights.fid
order by ticketrevenueusd desc
```



fid [PK] integer	distancekm numeric (10,2)	ticketrevenueusd double precision
4	3935.74	8000
3	12161.84	8000
2	10977.34	3700
1	8665.40	1700
8	1364.00	1500
6	6823.99	1400
5	342.76	850
7	8320.30	800

Aircraft that have been flown to or from their manufacturer’s headquarters

```
select fleet.planeid,
fleet.planename as Aircraft,
m.manufacturername as Manufacturer
from fleet
Join Manufacturers m on fleet.manufacturer=m.mid
join flights f1 on f1.aircraft=fleet.planeid
join destinations d1 on f1.takeoff=d1.did
join destinations d2 on f1.arrival=d2.did
where d1.city= m.hq or d2.city=hq;
```

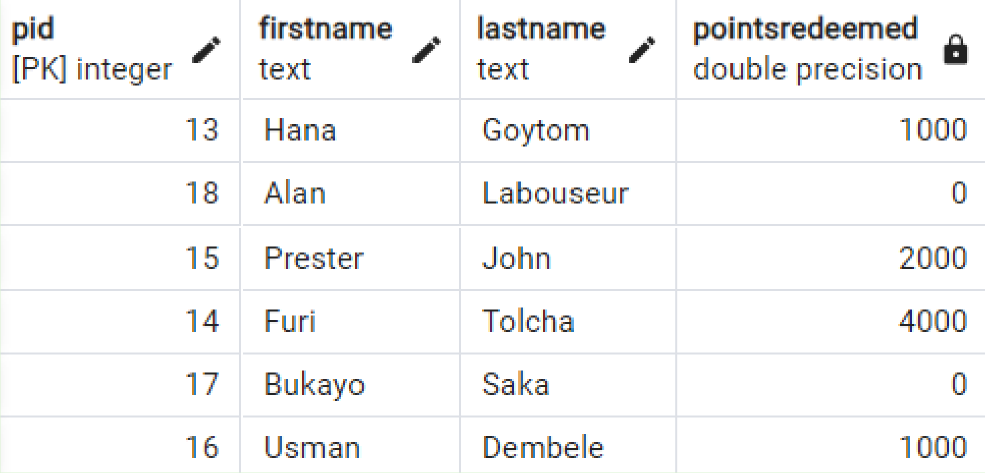


planeid integer	aircraft text	manufacturer text
4	Bombardier Q400	Bombardier

Total Miles Redeemed by Sheba Members

This query returns the total redeemed points for all ticket purchases by each Sheba member.

```
Select people.pid,people.firstName,
people.lastName,
sum(((tickets.totalpriceusd-priceafterdiscount)/shebadiscountusd)*100) as PointsRedeemed
from ShebaMembers
join people on people.pid=shebamembers.pid
join tickets on tickets.customer=shebamembers.pid
group by people.pid
```



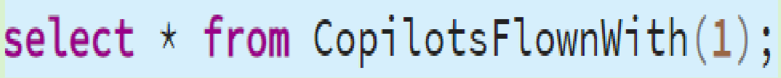
pid [PK] integer	firstname text	lastname text	pointsredeemed double precision
13	Hana	Goytom	1000
18	Alan	Labouseur	0
15	Prester	John	2000
14	Furi	Tolcha	4000
17	Bukayo	Saka	0
16	Usman	Dembele	1000

STORED PROCEDURES

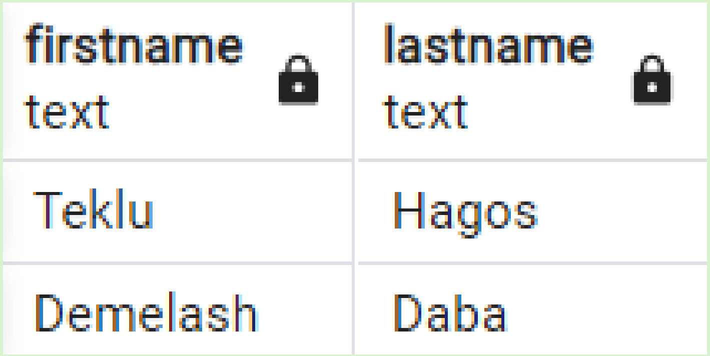
CoPilotsFlownWith

This function returns the first names and last names of co-pilots that have flown an aircraft with a given captain by inputting the CPID of the captain.

```
CREATE OR REPLACE FUNCTION
CopilotsFlownWith(input_cpid integer)
Returns Table(firstname text, lastName text)
as
$$
Begin      Return Query
           select people.firstName, people.lastName
           From People
           Join flights on people.pid=flights.copilot
           where flights.captain=input_cpid;
End;
$$
Language plpgsql;
```



```
select * from CopilotsFlownWith(1);
```

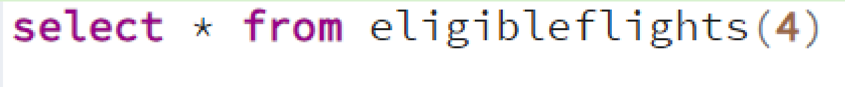


firstname text	lastname text
Teklu	Hagos
Demelash	Daba

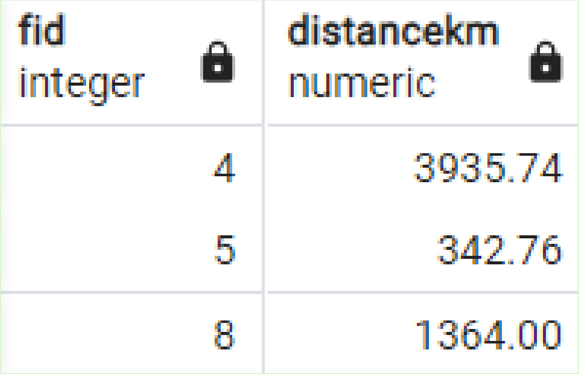
EligibleFlights

This function returns all the flights that can be conducted by a given aircraft.

```
CREATE OR REPLACE FUNCTION
EligibleFlights(input_planeid integer)
RETURNS TABLE(fid integer, distancekm decimal(10,2))
as $$
Begin
        RETURN QUERY
        SELECT f.fid, f.distancekm
        from flights f
        cross join fleet f1
        where f1.planeid=input_planeid
                and f.distancekm<=f1.rangeinkm;
end;
$$ language plpgsql;
```



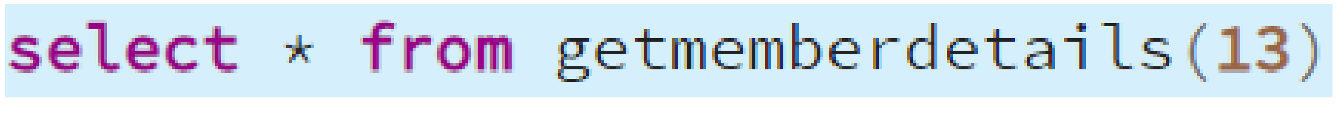
```
select * from eligibleflights(4)
```



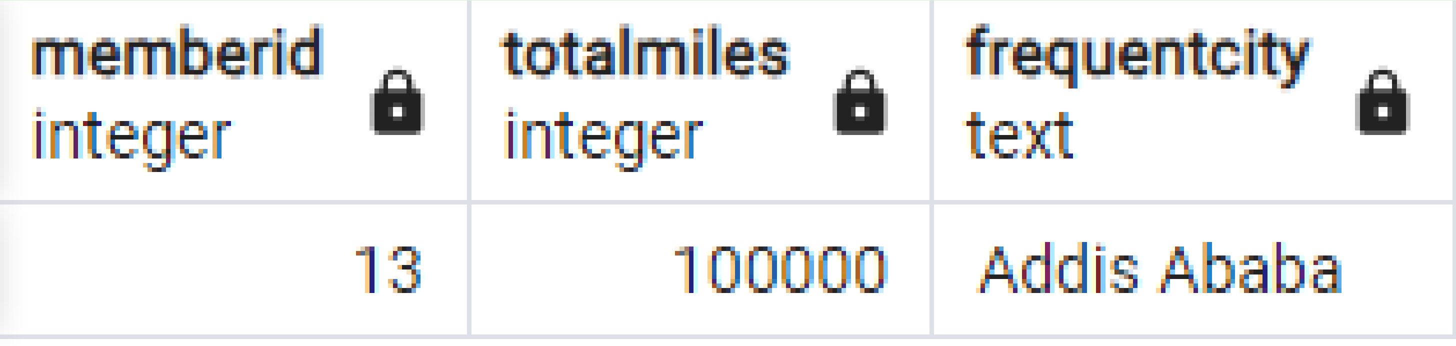
fid integer	distancekm numeric
4	3935.74
5	342.76
8	1364.00

MemberDetails

```
CREATE OR REPLACE FUNCTION GetMemberDetails(input_pid integer)
RETURNS TABLE(MemberID integer, TotalMiles int, FrequentCity text)
LANGUAGE plpgsql
AS $$
BEGIN
   RETURN QUERY
   SELECT
      SM.pid AS MemberID,
      SM.TotalMiles,
      D.city AS FrequentCity
   FROM
      ShebaMembers SM
      JOIN Travelers T ON SM.pid = T.pid
      JOIN Destinations D ON T.FreqDestination = D.did
   WHERE
      SM.pid = input_pid;
END;
$$;
```



```
select * from getmemberdetails(13)
```



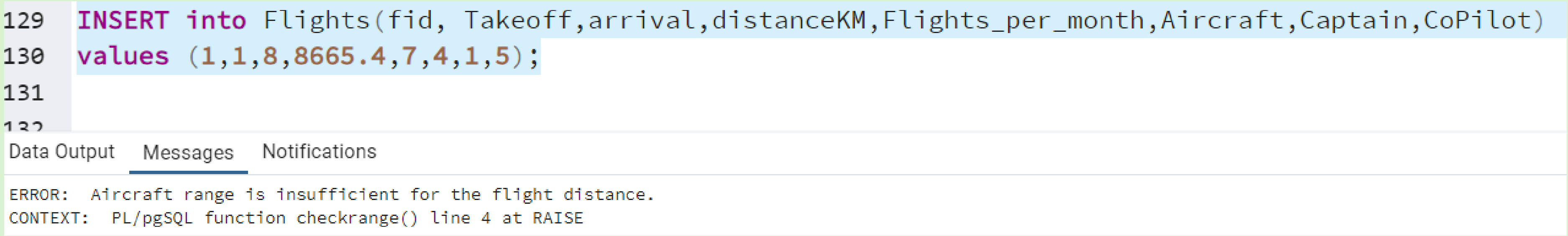
memberid integer	totalmiles integer	frequentcity text
13	100000	Addis Ababa

TRIGGERS

CheckRange

An aircraft must have enough range for a given distance. This trigger ensures that an aircraft with insufficient range can not be used for a flight.

```
CREATE OR REPLACE FUNCTION CheckRange()
RETURNS TRIGGER AS $$
BEGIN
   IF (SELECT RangeInKM FROM Fleet WHERE planeID = NEW.Aircraft) < NEW.distanceKM THEN
      RAISE EXCEPTION 'Aircraft range is insufficient for the flight distance.';
   END IF;
   RETURN NEW;
END;
$$ LANGUAGE plpgsql;
```



```
INSERT into Flights(fid, Takeoff,arrival,distanceKM,Flights_per_month,Aircraft,Captain,CoPilot)
values (1,1,8,8665.4,7,4,1,5);
```

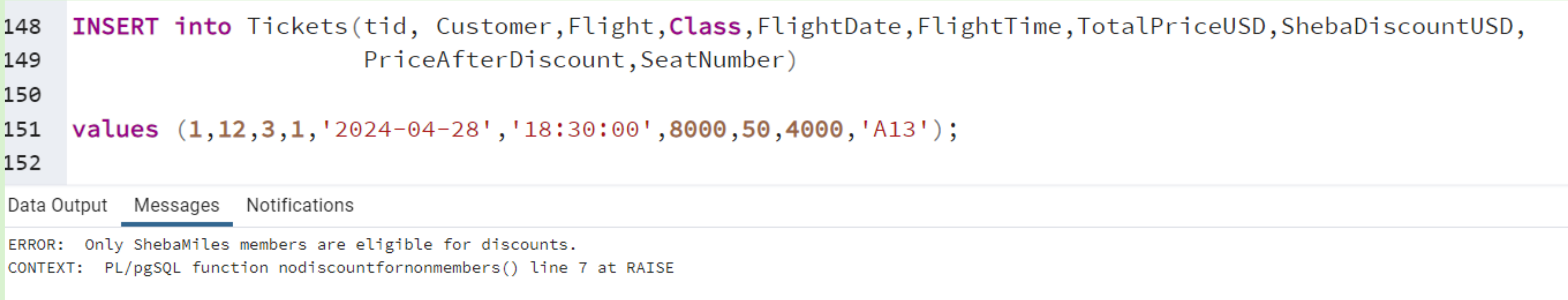
```
ERROR:  Aircraft range is insufficient for the flight distance.
CONTEXT:  PL/pgSQL function checkrange() line 4 at RAISE
```

```
CREATE TRIGGER trigger_check_aircraft_range
BEFORE INSERT OR UPDATE ON Flights
FOR EACH ROW
EXECUTE FUNCTION CheckRange();
```

NoDiscountForNonMembers

It is rule that there are no alternative discount options. Passengers must be Sheba Miles Members to get any kind of ticket discount.

```
CREATE OR REPLACE FUNCTION NoDiscountForNonMembers()
RETURNS TRIGGER AS $$
BEGIN
  IF NOT EXISTS (SELECT 1 FROM ShebaMembers WHERE pid = NEW.Customer) THEN
    IF NEW.TotalPriceUSD != NEW.PriceAfterDiscount THEN
      RAISE EXCEPTION 'Only ShebaMiles members are eligible for discounts.';
    END IF;
  END IF;
  RETURN NEW;
END;
$$ LANGUAGE plpgsql;
CREATE TRIGGER NoDiscountForNonMembers
BEFORE INSERT OR UPDATE ON Tickets
FOR EACH ROW
EXECUTE FUNCTION NoDiscountForNonMembers();
```



```
INSERT into Tickets(tid, Customer,Flight,Class,FlightDate,FlightTime,TotalPriceUSD,ShebaDiscountUSD,
                    PriceAfterDiscount,SeatNumber)

values (1,12,3,1,'2024-04-28','18:30:00',8000,50,4000,'A13');
```

```
ERROR:  Only ShebaMiles members are eligible for discounts.
CONTEXT:  PL/pgSQL function nodiscountfornonmembers() line 7 at RAISE
```

SECURITY/ROLES

Administrator

The administrator of the database can make changes to all information for all the tables in the database.

```
CREATE ROLE Admin;
GRANT ALL
ON ALL TABLES IN SCHEMA PUBLIC
TO Admin;
```

TravelAgent

The travel agent needs access to flight data and the membership status of passengers, as they figure out the best ticket option for their clients.

```
CREATE ROLE TravelAgent;
GRANT SELECT
ON Flights, Tickets, ShebaMembers
TO TravelAgent;
```

Technician

The technician needs access to different information regarding the airline fleet and be able to make changes as needed. Anytime there is a detected defect or a routine inspection, the technician can gather information about aircraft in question. They can not delete aircraft from the database, however.

```
CREATE ROLE TravelAgent;
GRANT SELECT, INSERT
ON Flights, Tickets, ShebaMembers
TO TravelAgent;
```

IMPLEMENTATION NOTES/ DESIGN WEAKNESSES AND FUTURE ENHANCEMENTS

Implementation Notes/ Design Weaknesses

- For the sake of simplicity, the sample data was limited and does not represent what an airline database looks like in the real world.
- The database uses generic datatypes in places where a specific one would be handier.
- I opted to use a different primary key for the Captains table to create an automatic trigger and ensure only certified pilots can be captains.
- Additional triggers might be necessary depending on other aspects of the aircraft and aviation regulations.

Future Enhancements

- As the airline expands and improves its customer services, other perks and deals can be added to the already existing ShebaMembers table.
- Include a table for the different ways of purchasing tickets and the varying transaction fees.
- The airline plans to reward high performing staff with Sheba memberships. We can create a Table for those staff and add a trigger that automatically inserts their name to ShebaMembers.