

# Alaska Airlines Database

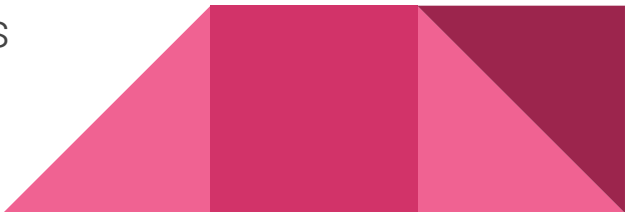
Allison Bennett, Andrew Gabra, Jenny Rong, Wanyu Guan

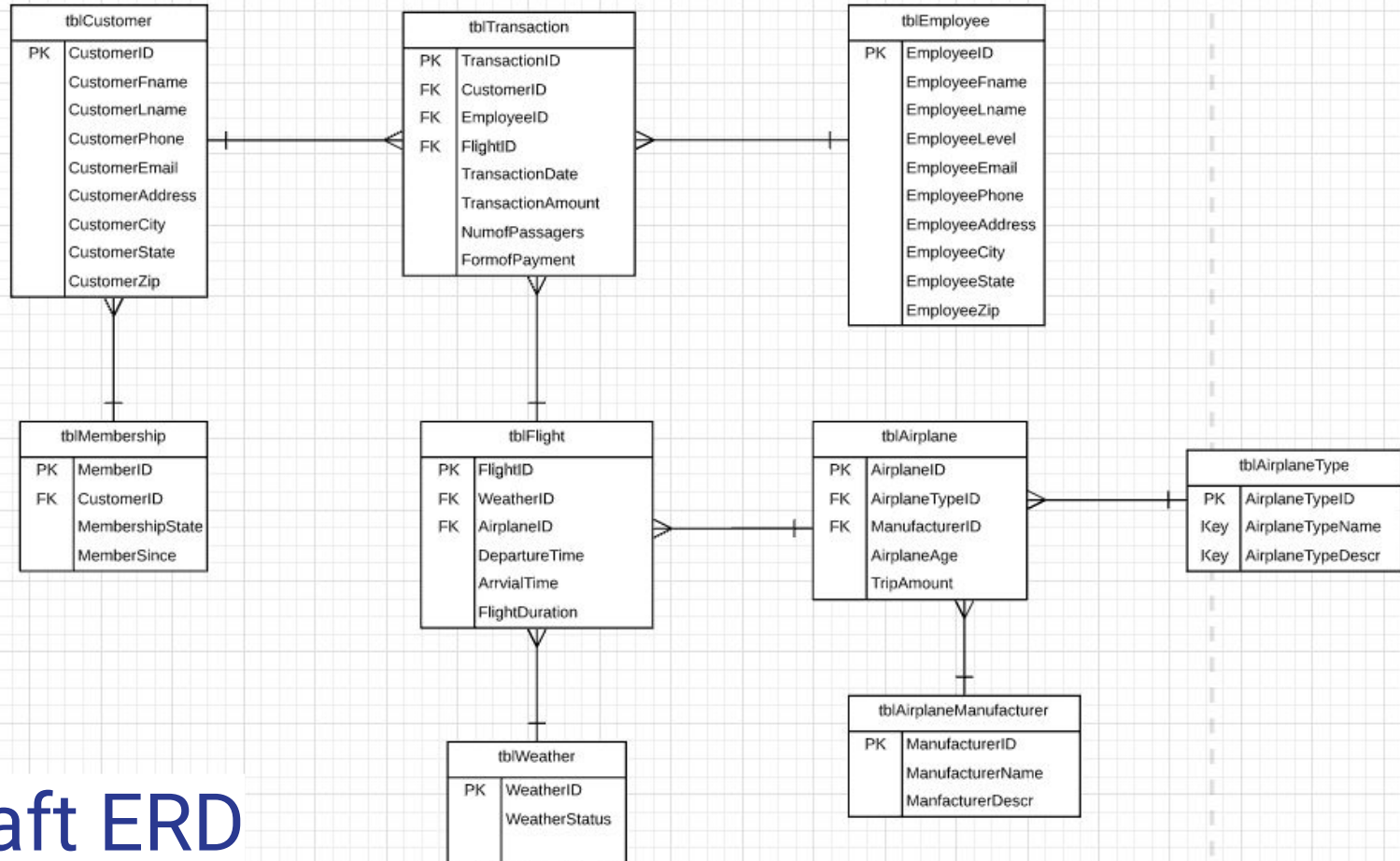
# Business Problem Space

## Context

- COVID-19 travel restrictions have negatively impacted Alaska Airlines
  - Focus on Seattle airport, Alaska Airlines' headquarters
- Challenges in booking tickets and finding flight information
- Net loss of \$122 million per quarter

## Needs

- To provide a centralized platform for Alaska Airlines
    - Handle flight bookings out of Seattle (SeaTac)
    - Support millions of customers by providing flight information
    - Compare sale performances pre and post COVID-19
  - Creative strategies to maintain or increase flight sales
- 



Draft ERD

# Revised ERD

## Flaws with draft:

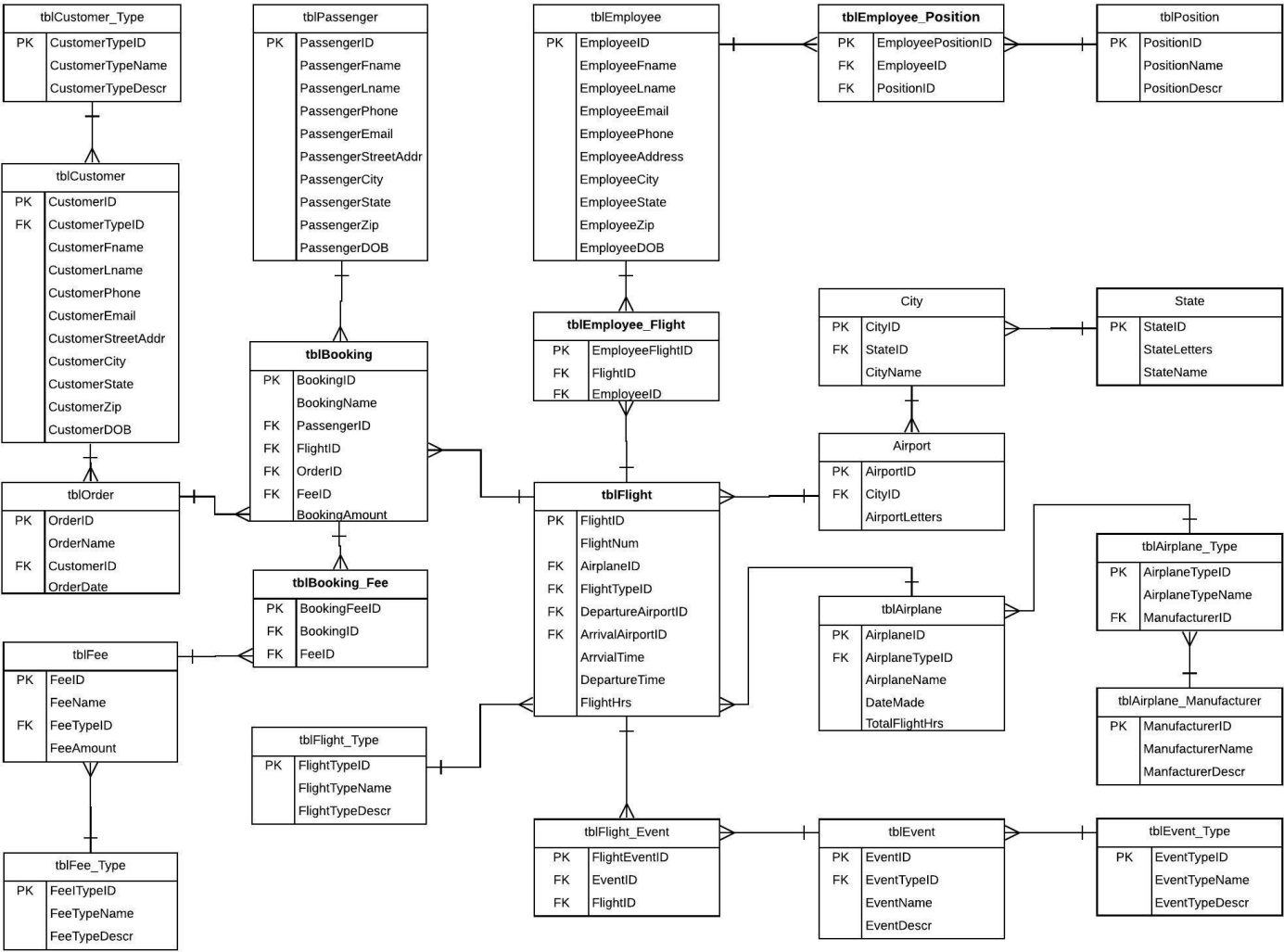
- Not storing much data.
- Inefficient connections

## Changes made:

- More tables
- More data
- Many to Many
- One to Many

## Why were changes made

- Normalized ERD
- More in depth data analysis.



# Physical Database Design

## Creation

- Split up elements, came together to ensure they worked together
- Used synthetic transactions for most inserts
  - Future use in testing how much the database can handle
  - Allowed for more data and more interesting results from queries

## Challenges

- Altering tables to align with database changes
  - Showed importance of spending time on ERD



# Inserting Data Using Synthetic Transaction

tblFee	
PK	FeeID
	FeeName
FK	FeeTypeID
	FeeAmount

```
CREATE PROCEDURE newFee
    @FeeName VARCHAR(50),
    @FeeAmount FLOAT,
    @FeeTypeName VARCHAR(50)
AS
    DECLARE @FeeTypeID INT

    EXEC getFeeTypeID
    @Fee_Type_Name = @FeeTypeName,
    @Fee_Type_ID = @FeeTypeID OUTPUT

    -- Error handle @FeeTypeID in case if it's NULL/ empty
    IF @FeeTypeID IS NULL
    BEGIN
        PRINT '@FeeTypeID is empty and this will cause the transaction to be failed';
        THROW 51000, '@FeeTypeID cannot be NULL', 1;
    END

    BEGIN TRAN G1
        INSERT INTO tblFee (FeeName, FeeTypeID, FeeAmount)
        VALUES (@FeeName, @FeeTypeID, @FeeAmount)
        IF @@ERROR <> 0
        BEGIN
            PRINT 'There is an error; need to rollback this transaction'
            ROLLBACK TRAN G1
        END
        ELSE
            COMMIT TRAN G1
    GO
```

```

CREATE PROCEDURE usp_Wrapper_NewFee
@Run INT
AS
DECLARE @FeeTypeCount INT = (SELECT COUNT(*) FROM tblFee_Type)
DECLARE @FeeType_Name VARCHAR(100)

```

```

DECLARE @Fee_Name VARCHAR(100), @Fee_Amount FLOAT

```

```

DECLARE @PK INT
DECLARE @Rand INT

```

```

WHILE @Run > 0

```

```

BEGIN

```

```

    SET @PK = (SELECT RAND()* @FeeTypeCount + 1)

```

```

    SET @FeeType_Name = (SELECT FeeTypeName FROM tblFee_Type WHERE FeeTypeID = @PK)

```

```

    SET @Rand = (SELECT RAND()* 100)

```

```

    SET @Fee_Amount = (SELECT ROUND(RAND() * 100, 2))

```

```

    SET @Fee_Name = (CASE

```

```

        WHEN @Rand < 10 THEN 'Additional checked bag'

```

```

        WHEN @Rand BETWEEN 11 AND 15 THEN 'Overweight bag'

```

```

        WHEN @Rand BETWEEN 16 AND 20 THEN 'Oversize bag'

```

```

        WHEN @Rand BETWEEN 21 AND 25 THEN 'Call center service charge'

```

```

        WHEN @Rand BETWEEN 26 AND 30 THEN 'Paper itinerary mailing service charge'

```

```

        WHEN @Rand BETWEEN 31 AND 35 THEN 'Partner award booking fee'

```

```

        WHEN @Rand BETWEEN 36 AND 40 THEN 'Unaccompanied minor service fee'

```

```

        WHEN @Rand BETWEEN 41 AND 45 THEN 'Pet travel fee'

```

```

        WHEN @Rand BETWEEN 46 AND 50 THEN 'Ticket receipt research fee'

```

```

        WHEN @Rand BETWEEN 51 AND 55 THEN 'Same day confirmed changes'

```

```

        WHEN @Rand BETWEEN 56 AND 60 THEN 'Cancellation fee'

```

```

        WHEN @Rand BETWEEN 61 AND 65 THEN 'Premium class seats'

```

```

        WHEN @Rand BETWEEN 66 AND 70 THEN 'First class paid upgrades'

```

```

        WHEN @Rand BETWEEN 71 AND 75 THEN 'Inflight meals and snack boxes'

```

```

        WHEN @Rand BETWEEN 76 AND 80 THEN 'Beer, wine, and cocktails'

```

```

        WHEN @Rand BETWEEN 81 AND 85 THEN 'Inflight WiFi'

```

```

        WHEN @Rand BETWEEN 86 AND 90 THEN 'Left on board item return fee'

```

```

        ELSE 'No fee'

```

```

    END)

```

```

EXEC [newFee]

```

```

@FeeName = @Fee_Name,

```

```

@FeeAmount = @Fee_Amount,

```

```

@FeeTypeName = @FeeType_Name

```

```

SET @Run = @Run - 1

```

```

END

```

```

GO

```

```

EXEC usp_wrapper_NewFee @Run = 20;

```

```

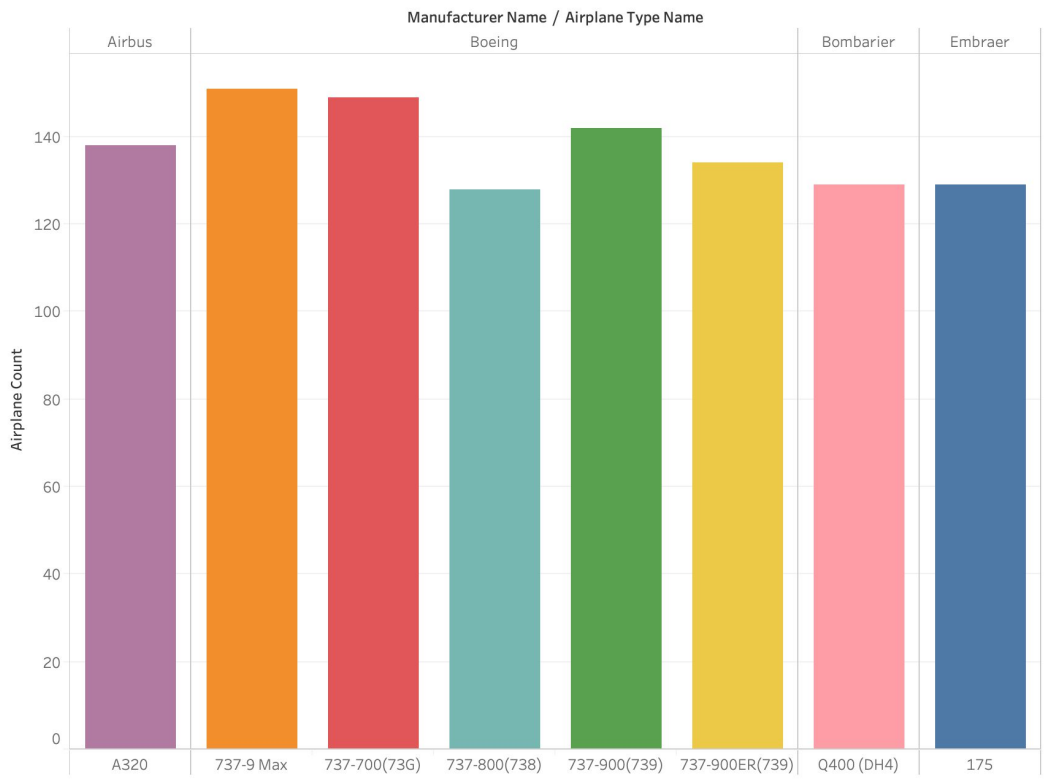
GO

```

# Inserting Data Using Synthetic Transaction Cont.

	FeeID	FeeName	FeeTypeID	FeeAmount
1	21	Overweight bag	2	29.86
2	22	Premium class seats	3	21.22
3	23	Inflight WiFi	4	85.59
4	24	Inflight WiFi	1	96.72
5	25	Paper itinerary mailing serv...	2	30.52
6	26	Unaccompanied minor service ...	1	32.8
7	27	Pet travel fee	2	67.65
8	28	First class paid upgrades	1	95.1
9	29	Same day confirmed changes	1	61.07
10	30	First class paid upgrades	2	7.08
11	31	Beer, wine, and cocktails	3	66.82
12	32	Unaccompanied minor service ...	2	15.71
13	33	Ticket receipt research fee	3	76.16
14	34	No fee	1	3.02
15	35	Ticket receipt research fee	3	39.73
16	36	Inflight WiFi	4	2.33
17	37	Left on board item return fee	1	23.72
18	38	Premium class seats	2	87.34
19	39	Additional checked bag	4	66
20	40	Overweight bag	4	88.17

# Data Visualizations



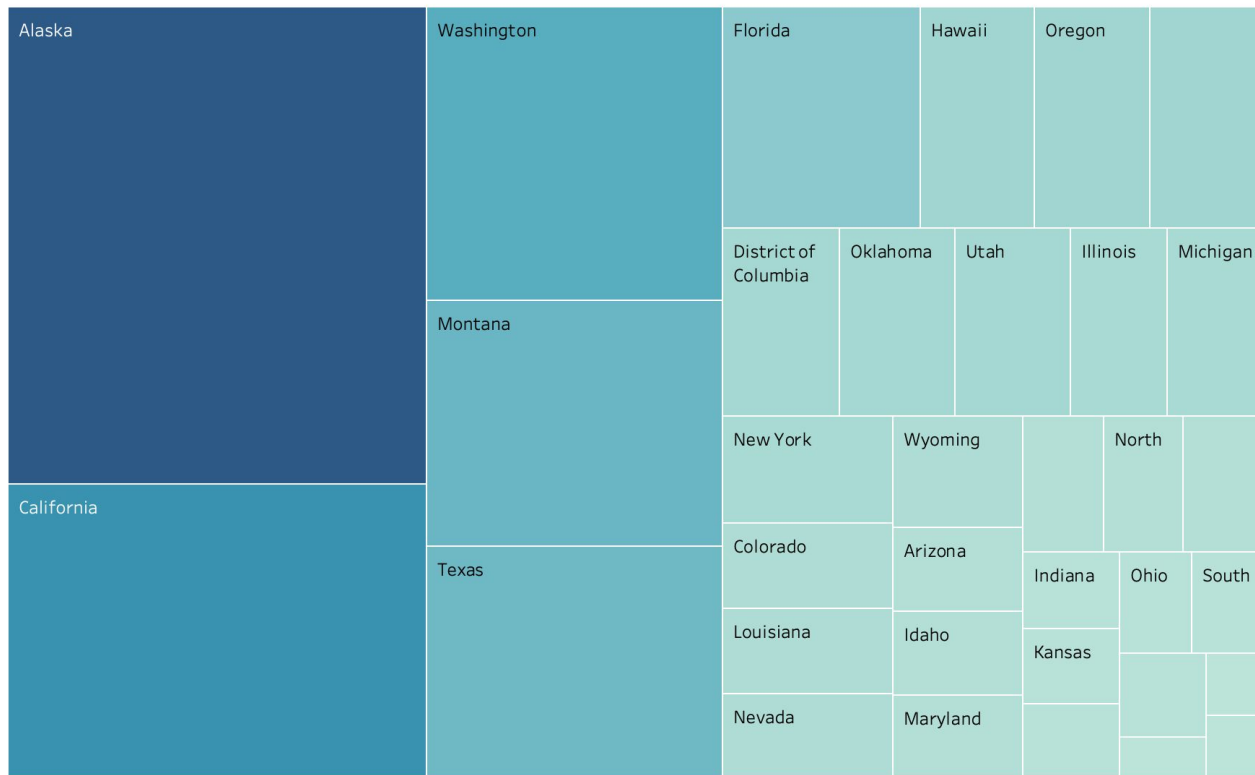
Number of each type of airplanes for each manufacturer

```
-- Number of each type of airplanes for each manufacturer
CREATE VIEW view_airplanes_per_manufacture
AS
SELECT AM.ManufacturerName, ATP.AirplaneTypeName, COUNT(A.AirplaneID) AS airplane_count
FROM tblAirplane A
JOIN tblAirplane_Type ATP ON A.AirplaneTypeID = ATP.AirplaneTypeID
JOIN tblAirplane_Manufacturer AM ON ATP.ManufacturerID = AM.ManufacturerID
GROUP BY AM.ManufacturerName, ATP.AirplaneTypeName
GO
```



# Data Visualizations

## Distribution of Destination States



--Views by different states

```
CREATE VIEW vwMost_Frequent_To_difState
```

```
AS
```

```
SELECT TOP(50) S.StateName, COUNT(*) AS NumFlights
```

```
FROM tblState S
```

```
JOIN tblCity C ON C.StateID = S.StateID
```

```
JOIN tblAirport A ON A.CityID = C.CityID
```

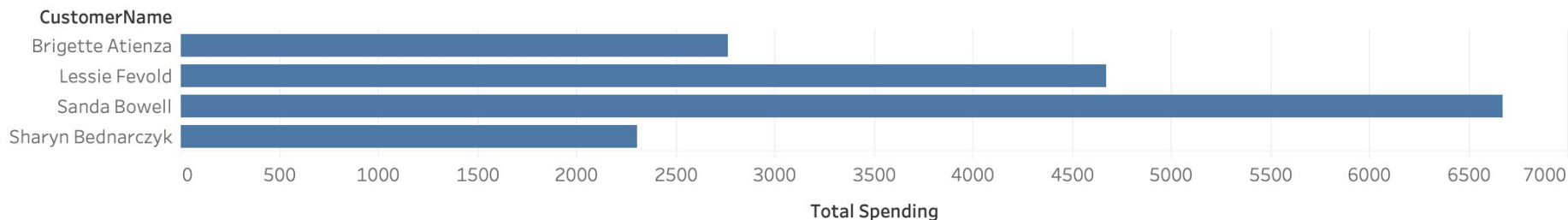
```
JOIN tblFlight F ON F.ArrivalAirportID = A.AirportID
```

```
GROUP BY S.StateName
```

```
GO
```

# Data Visualizations

## Top Customer Spending



```
CREATE VIEW vwCustomer_Spending
AS
SELECT CONCAT(C.CustomerFname, ' ', C.CustomerLname) AS CustomerName, SUM(B.BookingAmount) AS TotalSpending
FROM tblBooking B
    JOIN tblOrder O ON O.OrderID = B.OrderID
    JOIN tblCustomer C ON C.CustomerID = O.CustomerID
GROUP BY C.CustomerID, C.CustomerFname, C.CustomerLname
GO
```

CustomerName	
Brigette Atienza	2,765
Lessie Fevold	4,668
Sanda Howell	6,671
Sharyn Bednarczyk	2,303



Thank you!