Data Management Technologies

Project – Flight Data Warehouse

Kai Ye 24654

Sales Datawarehouse

2 Conceptual Design: Sales Fact Measures, Sales Fact Dimensions, Dimensional Sales Fact Model

3 Logical Design: Sales Data Structures

4 Physical Design & ETL ~ 6 Visual Analysis

Mohd Kaif 24529

Customer Datawarehouse

2 Conceptual Design: Customer Fact Measures, Customer Fact Dimensions, Dimensional Customer Fact Model

3 Logical Design: Customer Data Structures

7 Physical Design & ETL ~ 9 VISUAL ANALYSIS

Contents

1 Domain Analysis and Description 1
1.1 Business Process Modelling 1
1.2 Business questions 1
2 Conceptual Design 2
2.1 Facts 2
2.1.1 Sales Fact Measures 2
2.1.2 Customer Fact Measures 2
2.2 Dimensions 2
2.2.1 Sales Fact Dimensions 2
2.2.2 Customer Fact Dimensions 4
2.3 Dimensional Fact Models 6
2.3.1 Dimensional Sales Fact Model 6
2.3.2 Dimensional Customer Fact Model 7
3 Logical Design 8
3.1 Sales Data Structures 8
3.2 Customer Data Structures 9
4 Physical Design & ETL for Sales 10
4.1 Dimension Table 10
4.1.1 Departure Date Dimension Table creation 10
4.1.2 Source Airport Dimension Table creation 10
4.1.3 Destination Airport Dimension Table creation 10
4.1.4 Aircraft Dimension Table creation 11

4.1.5 Airline Dimension Table creation 11
4.1.6 Passenger Dimension Table creation 11
4.1.7 Payment Dimension Table creation 12
4.2 Fact Table 12
4.2.1 Flight Payment Fact Table Creation 12
4.3 ETL 13
4.3.1 Extract 13
4.3.2 Transform 13
4.3.3 Load 14
5 Querying 16
5.1 ROLLUP 16
5.2 CUBE 18
5.3 GROUPING SETS 19
6 Visual Analysis 20
7 Physical Design and ETL for Customer 22
7.1 Dimension Table 22
7.1.1 Airline Dimension Table 22
7.1.2 Date Dimension Table 22
7.1.3 Time Dimension Table 22
7.1.4 Source Airport Dimension Table 23
7.1.5 Destination Airport Dimension Table 23
7.1.6 Passenger Dimension Table 23
7.1.7 Class Dimension Table 24
7.2 Fact Table 24

7.2.1 Flight Travel Fact Table 24
7.3 ETL 24
7.3.1 Extract 24
7.3.2 Transform 25
7.3.3 Load 25
8 Querying 29
8.1 Business Question 29
8.2 ROLLUP 32
8.3 CUBE 32
8.4 GROUPING SET 33
8.5 RANK 34
8.6 WINDOWING 35
8.7 NTILE 36
8.8 PERIOD-TO-PERIOD 37
9 VISUAL ANALYSIS 38
10 Difficulties 40

1 Domain Analysis and Description

1.1 Business Process Modelling

In the scenario considered, there are two Business Processes that are proposed to be modeled:

1. Sales: A sale process occurs when the customer purchases a ticket for travel.

   This Business Process is analyzed on the
2. Customer's Age: When a sale is made, the passenger's age is collected. And a part of the project is focused on the age Analysis of the passengers.

1.2 Business questions

Implemented Business Warehouse should help managers to observe, analyze data, and make decisions by providing answers to questions like following:

1. Sales BP:

   Sales Business Process is common for airline companies and even all companies, because all companies need to sell products or services to survive in the modern market.

   - How much is the income of tickets from one Source Country and different Destination Country?
   - How much is the income of tickets from each Source Country to any Destination Country, and vice versa?
2. Customer BP:
   - what is the average age of the passengers travelling from specific source country to specific destination country?
   - What is the frequency of different age groups travelling on International flights vs Domestic flights?
   - What is the average age of passengers using different classes?

2 Conceptual Design

This section provides a conceptual structure of the Data Warehouse, describing facts and dimensions it made of.

Facts are the concepts on which data users base their decision-making process, and dimensions are the parameters of facts on the basis of which the facts can be analyzed, sliced, diced, or rolled through different hierarchies.

2.1 Facts

2.1.1 Sales Fact Measures

- Original Price: the original price of the airline, provided by Airline Company.
- Discount: the absolute value of a discount from Original Price, provided by Agent Sales Company.
- Final Price: the final price of the airline and customer actually paid money, provided by Agent Sales Company.

2.1.2 Customer Fact Measures

- Age: This is the exact number of years a passenger has lived up to their flight departure date, calculated based on their date of birth.

2.2 Dimensions

2.2.1 Sales Fact Dimensions

- Departure Date
  - date
    - month
      - quarter

            ● year
  - day of week
  - day type (workday, weekend, holiday)
- Source Airport
  - airport
    - ■ city
      - ◆ region
        - ● country
  - airport portal
  - airport size
- Destination Airport
  - airport
    - ■ city
      - ◆ region
        - ● country
  - airport portal
  - airport size
- Airline
  - Airline ID
  - Aircraft
    - ■ Aircraft Name
    - ■ Manufacturer
      - ◆ Country
    - ■ Production Date
- Passenger
  - Passport ID
  - Passenger Name
  - Phone Number

- Birthday

- Payment
  - Payment Time
  - Payment Method
  - Currency

2.2.2 Customer Fact Dimensions

- Airline Dimension
  - Airline
    - Aircraft
      - Fleet Size
      - Production Date
      - Aircraft Country
      - Manufacturer
- Date Dimension
  - Date
    - Month
      - Quarter
        - Year
    - Day of week
    - Holiday
    - Season
- Time Dimension
  - Second
    - Minute
      - Hour
- Destination Airport Dimension

- Destination Airport
    - Airport Size
    - Destination city
        - City Size
        - Destination Region
            - Destination Country
                - Destination Continent
- Source Airport Dimension
    - Source Airport
        - Source City
            - Source Region
                - Source country
                    - Source Continent
- Passenger Dimension
    - Passenger Name
        - Nationality
        - Phone Number
        - E-Mail
        - Date of Birth

- Class Dimension
    - Class

2.3 Dimensional Fact Models

2.3.1 Dimensional Sales Fact Model

Conceptually the Sales fact table and it’s dimensions are shown in following:

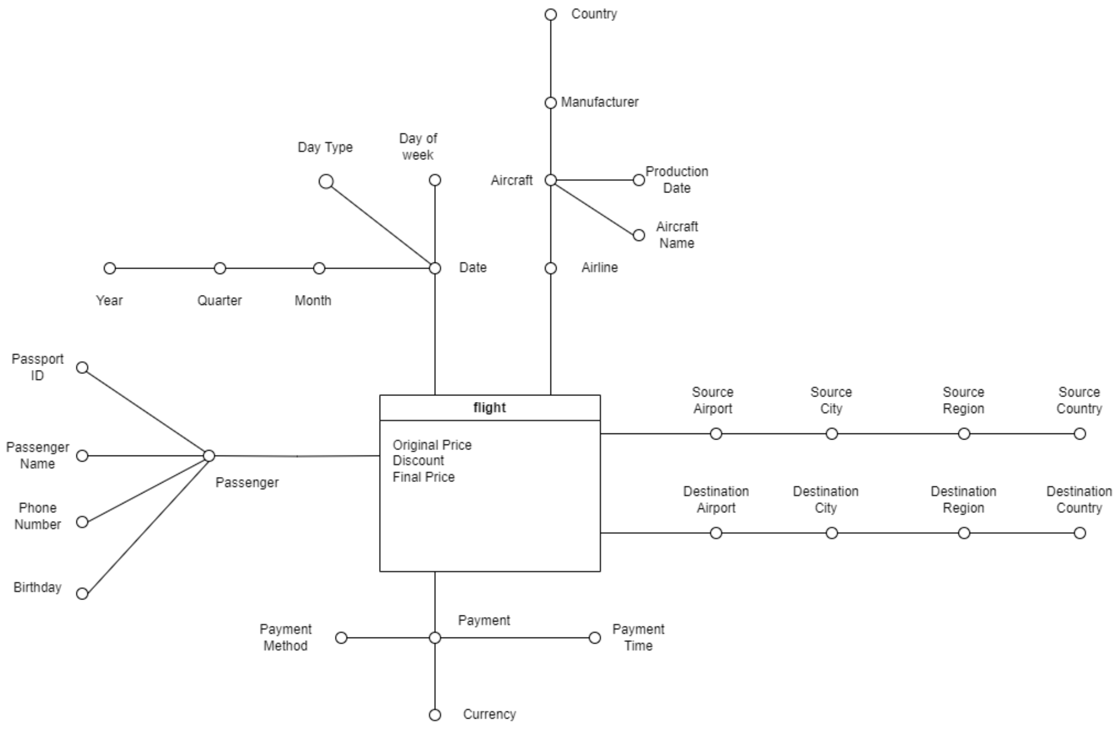


Figure 1 Conceptual design for Sales fact

2.3.2 Dimensional Customer Fact Model

Conceptually the Customer fact table and its dimensions are shown below:

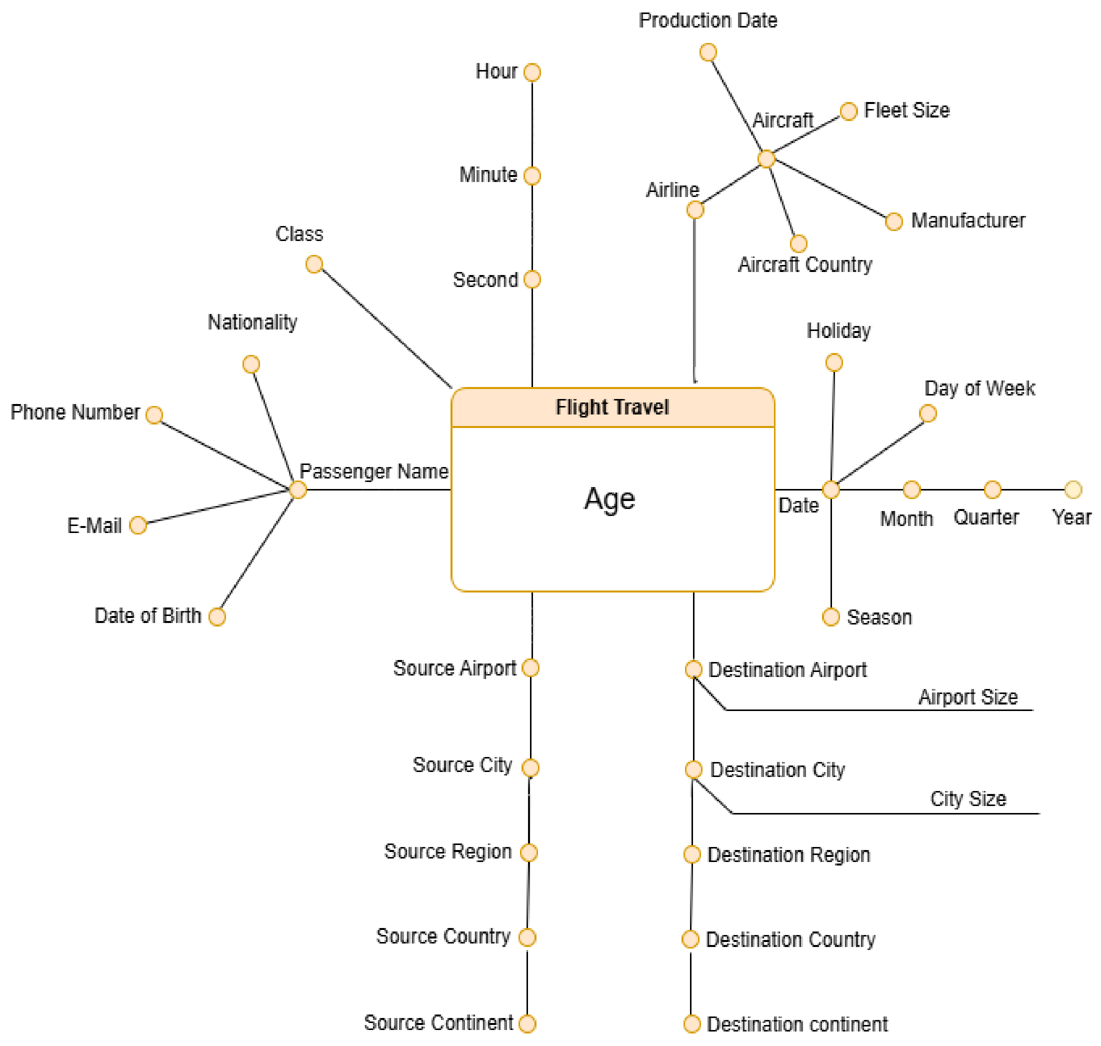


Figure 2 Conceptual design for Customer fact

3 Logical Design

3.1 Sales Data Structures

In this section, the conceptual data models of the flight databases are analyzed. These schemas are used as a base for the Data Warehouse physical design and implementation.

As conception design, we split all flight Data Warehouse into 1 fact table and 6 dimension tables.

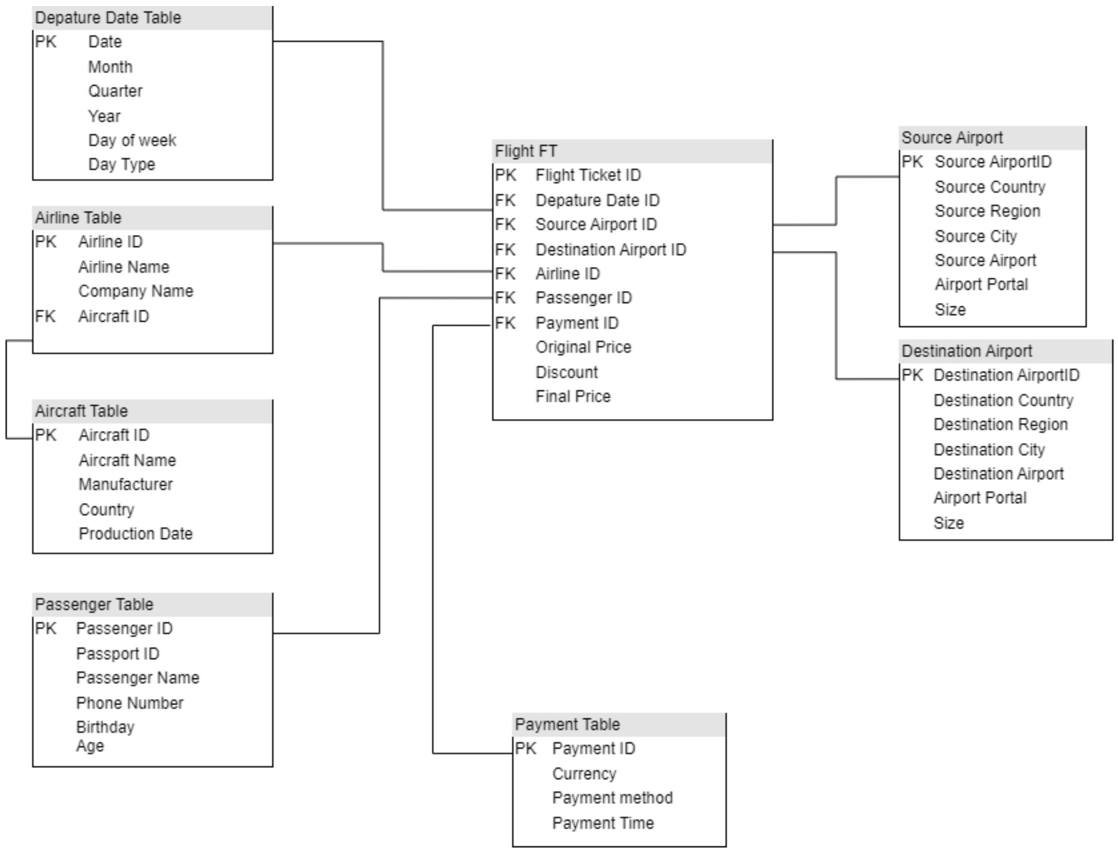


Figure 3 Logical design for Sales fact

3.2 Customer Data Structures

It shows a subset of the database used by flight travel:

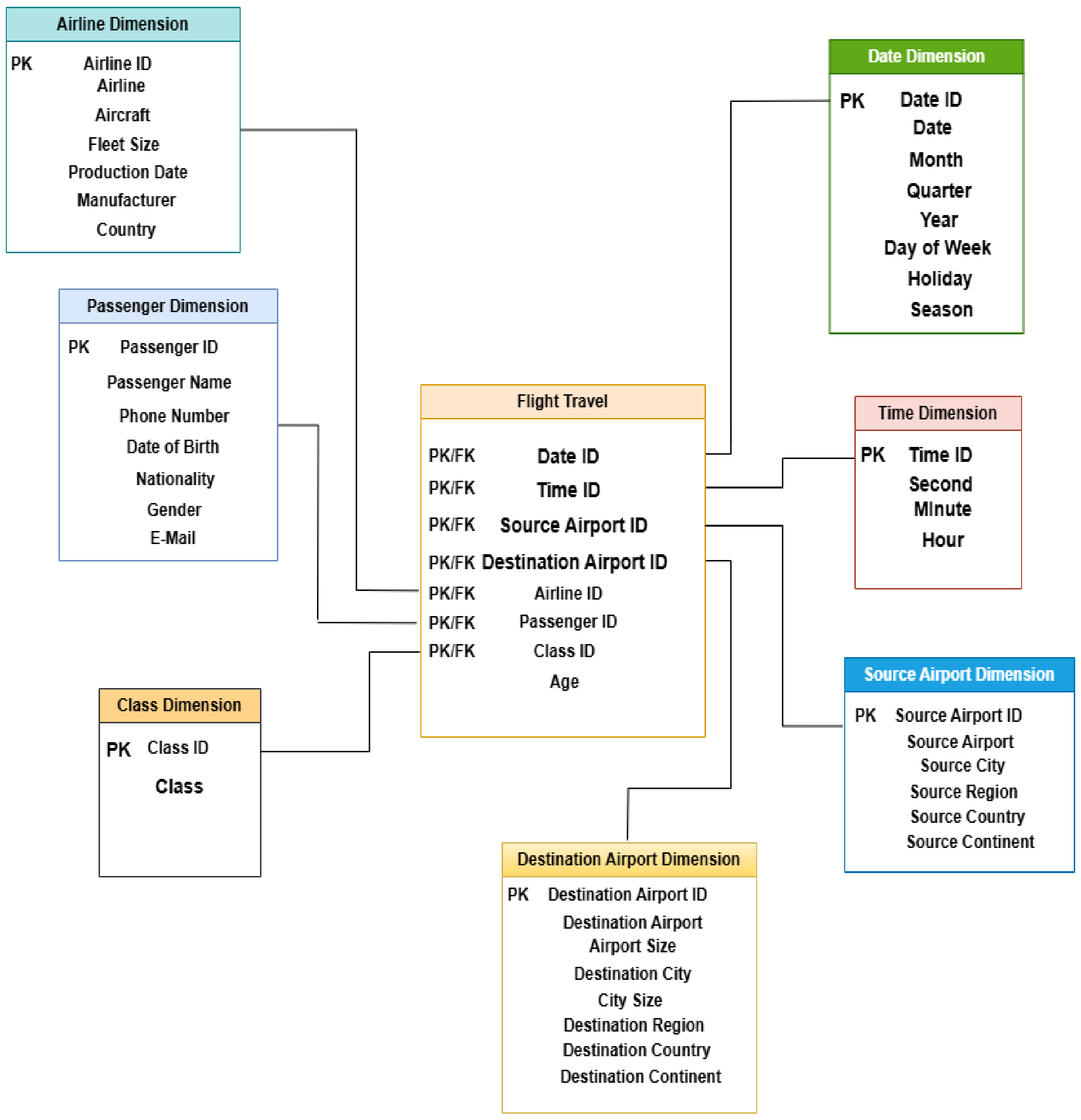


Figure 4 Logical design for Customer fact

4 Physical Design & ETL for Sales

4.1 Dimension Table

4.1.1 Departure Date Dimension Table creation

```sql
CREATE TABLE DepartureDate_Table (
    DateID DATE NOT NULL,
    Month INT NOT NULL,
    Quarter INT NOT NULL,
    Year INT NOT NULL,
    DayOfWeek INT NOT NULL,
    DayType CHAR(255),

    PRIMARY KEY (DateID)
);
```

4.1.2 Source Airport Dimension Table creation

```sql
CREATE TABLE SourceAirport_Table (
    SourceAirportID INT NOT NULL,
    Country CHAR(255),
    Region CHAR(255),
    City CHAR(255),
    AirportName CHAR(255),
    Size INT,

    PRIMARY KEY (SourceAirportID)
);
```

4.1.3 Destination Airport Dimension Table creation

```sql
CREATE TABLE DestinationAirport_Table (
    DestinationAirportID INT NOT NULL,
    Country CHAR(255),
    Region CHAR(255),
    City CHAR(255),
    AirportName CHAR(255),
    Size INT,

    PRIMARY KEY (DestinationAirportID)
);
```

4.1.4 Aircraft Dimension Table creation

```
CREATE TABLE Aircraft_Table (
    AircraftID INT NOT NULL,
    AircraftName CHAR(255),
    Manufacturer CHAR(255),
    Country CHAR(255),
    ProductionDate DATE,

    PRIMARY KEY (AircraftID)
);
```

4.1.5 Airline Dimension Table creation

```
CREATE TABLE Airline_Table (
    AirlineID INT NOT NULL,
    AirlineName CHAR(255),
    CompanyName CHAR(255),
    AircraftID INT,

    PRIMARY KEY (AirlineID),
    FOREIGN KEY (AircraftID) REFERENCES Aircraft_Table(AircraftID)
);
```

4.1.6 Passenger Dimension Table creation

```
CREATE TABLE Passenger_Table (
    PassengerID INT NOT NULL,
    PassportID CHAR(255),
    PassengerName CHAR(255),
    PhoneNumber CHAR(255),
    Birthday DATE,
    Age INT,

    PRIMARY KEY (PassengerID)
);
```

4.1.7 Payment Dimension Table creation

```
CREATE TABLE Payment_Table (
    PaymentID INT NOT NULL,
    Currency CHAR(255),
    PaymentMethod CHAR(255),
    PaymentTime TIME,

    PRIMARY KEY (PaymentID)
);
```

4.2 Fact Table

4.2.1 Flight Payment Fact Table Creation

```
CREATE TABLE FlightTicket_Table (
    TicketID INT NOT NULL,
    DateID DATE NOT NULL,
    SourceAirportID INT NOT NULL,
    DestinationAirportID INT NOT NULL,
    AirlineID INT NOT NULL,
    PassengerID INT NOT NULL,
    PaymentID INT NOT NULL,
    OriginalPrice DECIMAL(10, 2),
    Discount DECIMAL(10, 2),
    FinalPrice DECIMAL(10, 2) NOT NULL,

    PRIMARY KEY (TicketID),
    FOREIGN KEY (DateID) REFERENCES DepartureDate_Table(DateID),
    FOREIGN KEY (SourceAirportID) REFERENCES
SourceAirport_Table(SourceAirportID),
    FOREIGN KEY (DestinationAirportID) REFERENCES
DestinationAirport_Table(DestinationAirportID),
    FOREIGN KEY (AirlineID) REFERENCES Airline_Table(AirlineID),
    FOREIGN KEY (PassengerID) REFERENCES
Passenger_Table(PassengerID),
    FOREIGN KEY (PaymentID) REFERENCES Payment_Table(PaymentID)
);
```

4.3 ETL

4.3.1 Extract

Firstly, we generate a series of dataset from generatedata website, and save as csv files.

Containing below columns:

SourceCountry, SourceRegion, SourceCity, DestinationCountry, DestinationRegion, DestinationCity, OriginalPrice, Discount, DepatureDate, PaymentTime, PassengerName, PassportID, Phone

We need to extract these data into different tables:

SourceAirport_Table: SourceCountry, SourceRegion, SourceCity

DestinationAirport_Table: DestinationCountry, DestinationRegion, DestinationCity

DepartureDate_Table: DepatureDate

Payment_Table: PaymentTime

Passenger_Table: PassengerName, PassportID, Phone

FlightTicket_Table: OriginalPrice, Discount

4.3.2 Transform

Clean and format the data to meet your database schema's requirements.

- Convert dates to a standard format (YYYY-MM-DD).
- Calculate FinalPrice = OriginalPrice – Discount

4.3.3 Load

Load the transformed data into the PostgreSQL database.

We Create and import a temporary RawData_Table to load data into dimension and fact table.

```
CREATE TABLE RawData_Table (
    SourceCountry CHAR(255),
    SourceRegion CHAR(255),
    SourceCity CHAR(255),
    DestinationCountry CHAR(255),
    DestinationRegion CHAR(255),
    DestinationCity CHAR(255),
    OriginalPrice DECIMAL(10, 2),
    Discount DECIMAL(10, 2),
    DepatureDate DATE,
    PaymentTime TIME,
    PassengerName CHAR(255),
    PassportID CHAR(255),
    Phone CHAR(255)
);
```

Then insert these values into separate tables from the temporary table.

SourceAirport_Table:

```
INSERT INTO SourceAirport_Table(Country, Region, City)
SELECT SourceCountry, SourceRegion, SourceCity
FROM RawData_Table;
```

DestinationAirport_Table:

```
INSERT INTO DestinationAirport_Table(Country, Region, City)
SELECT DestinationCountry, DestinationRegion, DestinationCity
FROM RawData_Table;
```

DepartureDate_Table:

```
INSERT INTO DepartureDate_Table(DateID, Month, Quarter, Year,
DayOfWeek, DayType)
SELECT
```

```
    DepatureDate,
    EXTRACT(MONTH FROM DepatureDate) AS Month,
    EXTRACT(QUARTER FROM DepatureDate) AS Quarter,
    EXTRACT(YEAR FROM DepatureDate) AS Year,
    EXTRACT(DOW FROM DepatureDate) AS DayOfWeek,
    CASE
        WHEN EXTRACT(DOW FROM DepatureDate) IN (0, 6) THEN 'Weekend'
        ELSE 'Weekday'
    END AS DayType
FROM RawData_Table
ON CONFLICT (DateID) DO NOTHING;
```

Payment_Table:

```
INSERT INTO Payment_Table(PaymentTime)
SELECT PaymentTime
FROM RawData_Table;
```

Passenger_Table:

```
INSERT INTO Passenger_Table(PassengerName, PassportID, PhoneNumber)
SELECT PassengerName, PassportID, Phone
FROM RawData_Table;
```

FlightTicket_Table:

```
INSERT INTO FlightTicket_Table(
    DateID,
    SourceAirportID,
    DestinationAirportID,
    PassengerID,
    PaymentID,
    OriginalPrice,
    Discount,
    FinalPrice
    )
SELECT
    DepatureDate,
    SourceAirportID,
    DestinationAirportID,
    PassengerID,
    PaymentID,
    OriginalPrice,
    Discount,
    OriginalPrice - Discount AS FinalPrice
FROM RawData_Table
    JOIN SourceAirport_Table
        ON RawData_Table.SourceCountry = SourceAirport_Table.Country
        AND RawData_Table.SourceRegion = SourceAirport_Table.Region
        AND RawData_Table.SourceCity = SourceAirport_Table.City
    JOIN DestinationAirport_Table
    ON RawData_Table.DestinationCountry =
DestinationAirport_Table.Country
        AND RawData_Table.DestinationRegion =
DestinationAirport_Table.Region
        AND RawData_Table.DestinationCity =
DestinationAirport_Table.City
    JOIN Passenger_Table
    ON RawData_Table.PassengerName = Passenger_Table.PassengerName
        AND RawData_Table.PassportID = Passenger_Table.PassportID
        AND RawData_Table.Phone = Passenger_Table.PhoneNumber
    JOIN Payment_Table
    ON RawData_Table.PaymentTime = Payment_Table.PaymentTime
```

5 Querying

The purpose of this section is to solve the Business Problems mentioned above, and we use rollup, cube and grouping set queries to display the result.

5.1 ROLLUP

For the first BP: How much is the income of tickets from one Source Country and different Destination Country?

To solve this problem, at first we inner join FlightTicket table and Source/Destination

Airport Tables, and then grouped by rollup Source/Destination Country.

```
SELECT SourceAirport_Table.Country as SourceCountry,
    DestinationAirPort_Table.Country as DestinationCountry,
    sum(FinalPrice) as sumPrice,
    RANK() OVER (PARTITION BY SourceAirport_Table.Country ORDER BY
sum(FinalPrice) DESC) AS Rank
FROM FlightTicket_Table
INNER JOIN SourceAirport_Table
ON FlightTicket_Table.SourceAirportID =
SourceAirport_Table.SourceAirportID
INNER JOIN DestinationAirPort_Table
ON FlightTicket_Table.DestinationAirPortID =
DestinationAirPort_Table.DestinationAirportID
GROUP BY ROLLUP(SourceAirport_Table.Country,
DestinationAirPort_Table.Country)
```

From the result, we can see the ticket income of from one country to each country, e.g. from Australia to different countries (Costa Rica, Sweden…), and the sum of ticket income from Australia. The highest income from Australia is to Costa Rica excluding the sum of from Australia.

Table 1 Result of Rollup query

sourcecountry	destinationcountry	sumprice	rank
Australia	NULL	7552	1
Australia	Costa Rica	3560	2
Australia	Sweden	1544	3
Australia	Norway	884	4
Australia	Singapore	790	5
Australia	New Zealand	774	6
Austria	NULL	21489	1
Austria	India	4503	2
Austria	Ukraine	2310	3
…	…	…	…

5.2 CUBE

Then we try group by cube, this group is a little different from rollup, it will group all columns (e.g. both SourceAirport and DestinationAirport).

```
SELECT SourceAirport_Table.Country as SourceCountry,
    DestinationAirPort_Table.Country as DestinationCountry,
    sum(FinalPrice) as sumPrice,
    DENSE_RANK() OVER (PARTITION BY SourceAirport_Table.Country
ORDER BY sum(FinalPrice) DESC) AS Rank
FROM FlightTicket_Table
INNER JOIN SourceAirport_Table
ON FlightTicket_Table.SourceAirportID =
SourceAirport_Table.SourceAirportID
INNER JOIN DestinationAirPort_Table
ON FlightTicket_Table.DestinationAirPortID =
DestinationAirPort_Table.DestinationAirportID
GROUP BY CUBE(SourceAirport_Table.Country,
DestinationAirPort_Table.Country)
```

From the result, we can directly find the difference, compared with rollup, cube even show the sum of income to destination countries, and including the sum of income from all tickets.

Table 2 Result of Cube query

sourcecountry	destinationcountry	sumprice	rank
Vietnam	Singapore	795	9
Vietnam	New Zealand	710	10
Vietnam	United Kingdom	604	11
Vietnam	Germany	569	12
NULL	NULL	736092	1
NULL	Ireland	84444	2
NULL	Mexico	63021	3
NULL	Nigeria	42184	4
NULL	Costa Rica	36868	5
…	…	…	…

5.3 GROUPING SETS

For the second BP: How much is the income of tickets from each Source Country to any Destination Country, and vice versa?

To solve the second problem, we use group by grouping sets. This query only group one side of grouping sets, and calculate the sum of other sets excluding the chosen one.

```
SELECT SourceAirport_Table.Country as SourceCountry,
    DestinationAirPort_Table.Country as DestinationCountry,
    sum(FinalPrice) as sumPrice
FROM FlightTicket_Table
INNER JOIN SourceAirport_Table
ON FlightTicket_Table.SourceAirportID =
SourceAirport_Table.SourceAirportID
INNER JOIN DestinationAirPort_Table
ON FlightTicket_Table.DestinationAirPortID =
DestinationAirPort_Table.DestinationAirportID
GROUP BY grouping sets(SourceAirport_Table.Country,
DestinationAirPort_Table.Country)
```

From the result, we can find the sum of income from each source country, and the sum of income to each destination country.

Table 3 Result of Grouping sets query

sourcecountry	destinationcountry	sumprice
Peru	NULL	22937
Poland	NULL	27298
Costa Rica	NULL	18170
Philippines	NULL	19561
United States	NULL	7109
NULL	Turkey	10825
NULL	Nigeria	42184
NULL	Indonesia	11400
NULL	New Zealand	22199
…	…	…

6 Visual Analysis

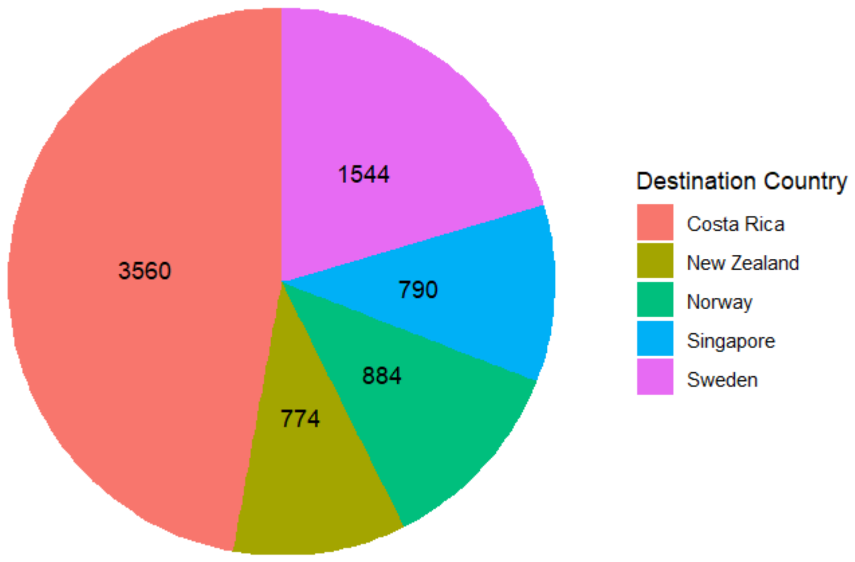


Figure 5 Income of Airline from Australian

This is the plot of rollup query result, about the income all airlines come from Australian, we can get the information of highest income of destination is Costa Rica. And the lowest income is New Zealand. It’s nearly three times more than New Zealand.

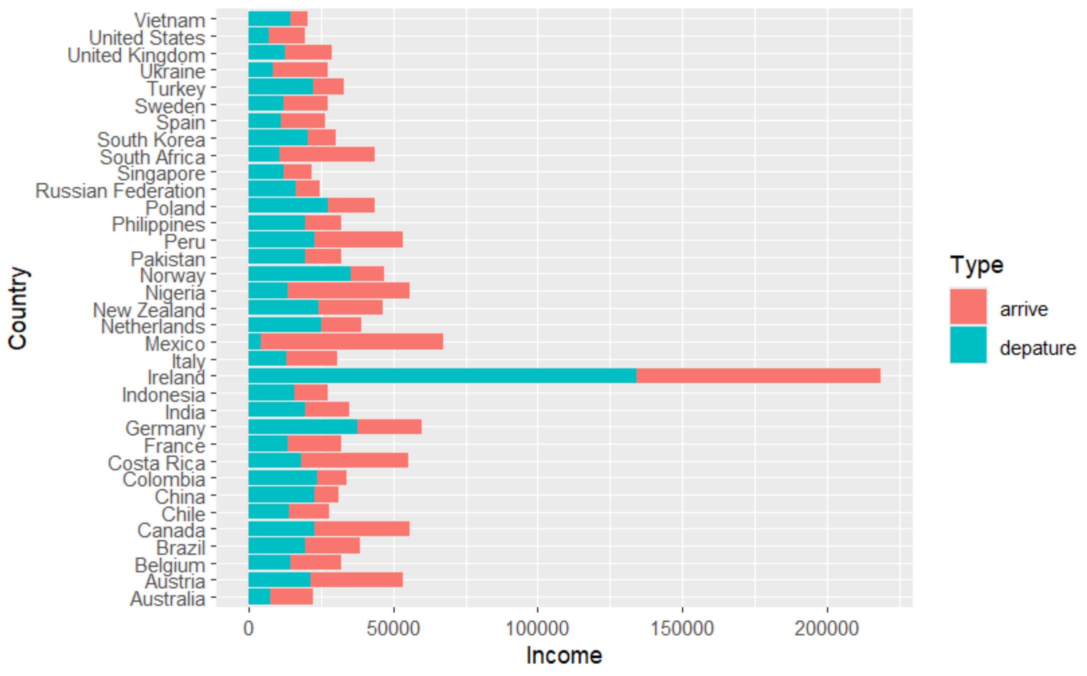


Figure 6 Airline income

Second plot belongs to grouping sets query, this plot lists all income of source and destination countries. And we can find the highest income of country is Ireland, both arriving and departing airline income are much higher than any other country.

7 Physical Design and ETL for Customer

7.1 Dimension Table

7.1.1 Airline Dimension Table

```
CREATE TABLE Airline_Dimension (
    "AirlineID" SERIAL PRIMARY KEY,
    "Airline" VARCHAR(30),
    "Aircraft" VARCHAR(12),
    "Fleet_Size" INT,
    "Manufacturer" VARCHAR(20),
    "Country" VARCHAR(15)
);
```

7.1.2 Date Dimension Table

```
CREATE TABLE Date_Dimension (
    "DateID" SERIAL PRIMARY KEY,
    "Date" DATE NOT NULL,
    "Month" SMALLINT NOT NULL,
    "Quarter" SMALLINT NOT NULL,
    "Year" SMALLINT NOT NULL,
    "DayofWeek" VARCHAR(10) NOT NULL,
    "Holiday" VARCHAR(4),
    "Season" VARCHAR(7)
);
```

7.1.3 Time Dimension Table

```
CREATE TABLE Time_Dimension (
    "TimeID" SERIAL PRIMARY KEY,
    "Second" INT NOT NULL,
    "Minute" INT NOT NULL,
    "Hour" INT NOT NULL
);
```

7.1.4 Source Airport Dimension Table

```
CREATE TABLE Source_Airport_Dimension (
"SourceAirportID" SERIAL PRIMARY KEY,
"Source_Airport" VARCHAR(30),
"Source_city" VARCHAR(15) NOT NULL,
"Source_Region" VARCHAR(15) NOT NULL,
"Source_Country" VARCHAR(15) NOT NULL,
"Source_Continent" VARCHAR(15)
);
```

7.1.5 Destination Airport Dimension Table

```
CREATE TABLE Destination_Airport_Dimension (
    "DestinationAirportID" SERIAL PRIMARY KEY,
    "Destination_Airport" VARCHAR(30),
    "Airport_Size" VARCHAR(7),
    "Destination_City" VARCHAR(15) NOT NULL,
    "City_Size" VARCHAR(7) ,
    "Destination_Region" VARCHAR(15) NOT NULL,
    "Destination_Country" VARCHAR(15) NOT NULL,
    "Destination_Continent" VARCHAR(15)
);
```

7.1.6 Passenger Dimension Table

```
CREATE TABLE    Passenger_Dimension (
    "PassengerID" SERIAL PRIMARY KEY,
    "Passenger_Name" VARCHAR(30),
    "Phone_Number" VARCHAR(15),
    "DateOfBirth" DATE NOT NULL,
    "Nationality" VARCHAR(20),
    "Gender" VARCHAR(7),
    "E-mail" VARCHAR(30)
);
```

7.1.7 **Class Dimension Table**

```
CREATE TABLE Class_Dimension (
        "ClassID" SERIAL PRIMARY KEY,
        "Class" VARCHAR(10) NOT NULL
);
```

7.2 **Fact Table**

7.2.1 **Flight Travel Fact Table**

```
CREATE TABLE Flight_Travel (
      "DateID" SERIAL NOT NULL,
      "TimeID" SERIAL NOT NULL,
      "SourceAirportID" SERIAL NOT NULL,
      "DestinationAirportID" SERIAL NOT NULL,
      "AirlineID" SERIAL NOT NULL,
      "PassengerID" SERIAL NOT NULL,
      "ClassID" SERIAL NOT NULL,
      "Age" SMALLINT,
        PRIMARY KEY ("DateID", "TimeID", "SourceAirportID", "DestinationAirportID",
"AirlineID", "PassengerID", "ClassID"),
      FOREIGN KEY ("DateID") REFERENCES Date_Dimension("DateID"),
      FOREIGN KEY ("TimeID") REFERENCES Time_Dimension("TimeID"),
      FOREIGN KEY ("SourceAirportID") REFERENCES
Source_Airport_Dimension("SourceAirportID"),
      FOREIGN KEY ("DestinationAirportID") REFERENCES
Destination_Airport_Dimension("DestinationAirportID"),
      FOREIGN KEY ("AirlineID") REFERENCES Airline_Dimension("AirlineID"),
      FOREIGN KEY ("PassengerID") REFERENCES Passenger_Dimension("PassengerID"),
      FOREIGN KEY ("ClassID") REFERENCES Class_Dimension("ClassID")
);
```

7.3 **ETL**

7.3.1 **Extract**

The data has been generated from generatedata.com website in the PostgreSQL format containing the following columns from each table:

“Airline”, “Aircraft”, “Fleet_Size”, “Manufacturer”, “Country”

“Date”, “Month”, “Quarter”, “Year”, “DayofWeek”, ‘Holiday’, “Season”

“Second”, “Minute”, “Hour”

“Source_Airport”, “Source_city”, “Source_Region”, “Source_Country”, “Source_Continent”

“Destination_Airport”, “Airport_Size”, “Destination_City”, “City_Size”, “Destination_Region”, “Destination_Country”, “Destination_Continent”

“Passenger_Name”, “Phone_Number”, “DateOfBirth”, “Nationality”, “Gender”, “E-mail”

“Class”

7.3.2 Transform

As the business questions are based on ‘Age’, so the age is calculated as the difference of Date of Travel(Date) – Date of Birth(DateofBirth)

- Age = Date – DateofBirth

7.3.3 Load

As the data has been downloaded in the PostgreSQL format, the data was in query form like below shown:

Airline_Dimension

```
INSERT INTO Airline_Dimension ("Fleetsize")
VALUES
('48'), ('72'), ('48'), ('60'), ('48'), ('60'), ('72'), ('48'), ('48'), ('72'), ('60'), ('48'), ('48'), ('48'), ('48'), ('60'),
('72'),   ('72'), ('48'), ('60'), ('48'), ('72'), ('72'), ('84'), ('84'), ('60'), ('48'), ('84'), ('72'), ('48'), ('72'), ('60'),
('60'), ('60'), ………..;
```

These are 34 values, similarly there a total 500 values inserted in the “Airline_Dimension” Table.

Date_Dimension

```
INSERT INTO Date_Dimension (Date,Year,DayofWeek)
VALUES
   ('2024-3-16','2024','Sat'), ('2024-7-6','2024','Thu'), ('2024-1-8','2024','Sat'), ('2024-8-6','2024','Sun'), ('2024-5-29','2024','Wed'), ('2024-9-21','2024','Fri'), ('2024-10-2','2024','Tue'), ('2024-1-17','2024','Sat'), ('2024-1-29','2024','Thu'), ('2024-10-4','2024','Fri'), ('2024-7-30','2024','Thu'), ('2024-3-11','2024','Fri'), ('2024-3-14','2024','Wed'), ('2024-1-11','2024','Sun'), ('2024-4-21','2024','Mon'), ('2024-6-7','2024','Tue'), ('2024-1-22','2024','Fri'), ('2024-12-1','2024','Fri'), ……..;
```

These are 20 values shown how the data has been inserted and there are a total of 500 values in the "Date_Dimension".

Time_Dimension

```
INSERT INTO Time_Dimension (Second,Minute,Hour)
VALUES
   ('25','27','16'), ('21','49','2'), ('48','28','18'), ('07','39','13'), ('35','05','21'), ('15','49','22'), ('29','02','5'), ('27','04','5'), ('26','25','7'), ('21','08','17'), ('51','50','1'), ('14','09','17'), ('31','11','23'), ('21','58','16'), ('04','39','23'), ('34','35','11'), ('18','52','5'), ('59','05','15'), ('37','34','9'),('45','18','15'), ………..;
```

These are 20 values shown how the data has been inserted and there are a total of 500 values in the "Time_Dimension".

Source_Airport_Dimension

```
INSERT INTO Source_Airport_Dimension ("Country","Region","City")
VALUES
   ('Chile','Arica y Parinacota','Arica'), ('South Korea','Gyeonggi','Incheon'), ('Australia','New South Wales','Grafton'), ('Ireland','Munster','Cork'), ('Philippines','Zamboanga Peninsula','Dapitan'), ('United Kingdom','Cardiganshire','Tregaron'), ('Vietnam','Bình Thuận','Phan Thiết'), ('Belgium','Oost-Vlaanderen','Daknam'), ('Russia Federation','Volgograd Oblast','Volgograd'), ('Singapore','North-East Region','Ang Mo Kio'), ('Nigeria','Akwa Ibom','Ikot Ekpene'), ('United Kingdom','Hampshire','Portsmouth'), ('Costa Rica','Cartago','San Nicolás'), ('Russian Federation','Bryansk Oblast','Pochep'), ('Poland','Śląskie','Katowice'), ('New Zealand','South Island','Ashburton'), ('South Korea','North Gyeongsang','Gimcheon'), ('Ireland','Ulster','Belfast'), ('Turkey','Kocaeli','Körfez'), ('India','Arunachal Pradesh','Itanagar'), ……….;
```

These are 20 values shown how the data has been inserted and there are a total of 500 values in the "Source_Airport_dimension".

Destination_Airport_Dimension

```
INSERT INTO Date_Dimension (Destination_City,Destination_Region,Destination_Country)
VALUES
  ('Bedok','East Region','Singapore'), ('Cobquecura','Biobío','Chile'), ('Oaxaca','Oaxaca','Mexico'), ('Grangemouth','Stirlingshire','United Kingdom'), ('Mascalucia','Sicilia','Italy'), ('Chepén','La Libertad','Peru'), ('Wijnegem','Antwerpen','Belgium'), ('Bazel','Oost-Vlaanderen','Belgium'), ('Palma de Mallorca','Illes Balears','Spain'), ('Reims','Champagne-Ardenne','France'), ('Oroquieta','Northern Mindanao','Philippines'), ('Wick','Caithness','United Kingdom'), ('Kharabali','Astrakhan Oblast','Russian Federation'), ('Cherkasy','Cherkasy oblast','Ukraine'), ('Ganganagar','Rajasthan','India'), ('Arequipa','Arequipa','Peru'), ('Otukpo','Benue','Nigeria'), ('Illkirch-Graffenstaden','Alsace','France'), ('Jecheon','North Chungcheong','South Korea'), ('Istanbul','Istanbul','Turkey'), ………;
```

These are 20 values shown how the data has been inserted, apart from these 20 values there are a total of 500 values present in the above table.

Passenger_Dimension

```
INSERT INTO Passenger_Dimension
("Passenger_Name","Phone_Number","DateOfBirth","Nationality","Gender","Email")
VALUES
  ('Cole O. Chen','1-469-492-9672','1964-03-09','Saint Pierre and
Miquelon','Male','coleochen6940@hotmail.edu'),
  ('Allistair Ellison','1-586-318-2060','2001-01-07','Saudi
Arabia','Male','allistairellison6802@protonmail.ca'),
  ('Kelly Thompson','431-4980','1990-10-04','Ecuador','Male','kellythompson@icloud.edu'),
  ('Magee C. Cardenas','566-5770','1965-01-
17','Malaysia','Male','mageeccardenas2423@hotmail.com'),
  ('Asher Stokes','1-644-414-5451','2005-04-12','Bouvet Island','Male','asherstokes@aol.ca'),
  ('Serina S. Dickson','522-7749','1982-11-20','Laos','Female','serinasdickson7795@icloud.org'),
  ('Robin W. Emerson','1-654-349-6033','2001-05-
20','Kuwait','Female','robinwemerson@icloud.org'),
  ('Yvette I. Navarro','1-182-885-3798','1989-02-
25','Oman','Female','yvetteinavarro@outlook.com'), ……………..;
```

These are 10 values shown how the data has been inserted and there are a total of 500 values in the “Passenger_Dimension”.

Class_Dimension

```
INSERT INTO Class_Dimension (Class)
VALUES
   ("First Class"), ("Premium Economy"), ("Business"), ("Premium Economy"), ("Premium
Economy"), ("First Class"), ("Economy"), ("Business"), ("Business"), ("Premium Economy"),
("First Class"), ("Premium Economy"), ("Premium Economy"), ("First Class"), ("Economy"),
("Economy"), ("First Class"), ("First Class"), ("First Class"), ("First Class"), ……….;
```

These are 200 values shown how the data has been inserted and there are a total of 500

```
CREATE TABLE short_term (
    "DateID" SMALLINT,
    "Date" DATE,
    "Year" SMALLINT,
    "DayofWeek" VARCHAR(15),
    "TimeID" SMALLINT,
    "Second" SMALLINT,
    "Minute" SMALLINT,
   "Hour" SMALLINT,
    "SourceAirportID" SMALLINT,
    "Source_City" VARCHAR(60),
    "Source_Region" VARCHAR(60),
    "Source_Country" VARCHAR(60),
    "DestinationAirportID" SMALLINT,
    "Destination_City" VARCHAR(60),
    "Destination_Region" VARCHAR(60),
    "Destination_Country" VARCHAR(60),
    "AirlineID" SMALLINT,
    "Fleet_Size" SMALLINT,
    "PassengerID" SMALLINT,
    "Passenger_Name" VARCHAR(60),
    "Phone_Number" VARCHAR(50),
    "DateOfBirth" DATE,
    "Nationality" VARCHAR(60),
    "Gender" VARCHAR(10),
    "Email" VARCHAR(60),
    "ClassID" SMALLINT,
    "Class" VARCHAR(20),
    "Age" SMALLINT
);
```

And in the above table the data has been imported from a CSV file containing all the columns that are present in the “short_term” table

Flight_Travel (Fact Table)

```
INSERT INTO Flight_Travel("DateID", "TimeID", "SourceAirportID", "DestinationAirportID",
"AirlineID", "PassengerID", "ClassID")
SELECT short_term."DateID", short_term."TimeID", short_term."SourceAirportID",
short_term."DestinationAirportID", short_term."AirlineID", short_term."PassengerID",
short_term."ClassID"
FROM short_term;

Below is the Query to insert all the ages to the “Flight_Travel” fact table

UPDATE Flight_Travel
SET "Age" = EXTRACT(YEAR FROM AGE(Date_Dimension."Date",
Passenger_Dimension."DateOfBirth"))
FROM Date_Dimension, Passenger_Dimension
WHERE Flight_Travel."DateID" = Date_Dimension."DateID"
AND Flight_Travel."PassengerID" = Passenger_Dimension."PassengerID";
```

8 Querying

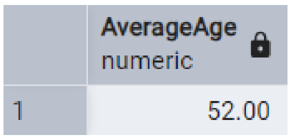
8.1 Business Question

1. what is the average age of the passengers travelling from specific source country to specific destination country?

   To solve this problem the following query has been written and specific source country is “Norway” and specific destination country is “Austria”

```
SELECT ROUND(AVG(Flight_Travel."Age"), 2) AS "AverageAge"
FROM Flight_Travel
JOIN Source_Airport_Dimension ON Flight_Travel."SourceAirportID" =
Source_Airport_Dimension."SourceAirportID"
JOIN Destination_Airport_Dimension ON Flight_Travel."DestinationAirportID" =
Destination_Airport_Dimension."DestinationAirportID"
WHERE Source_Airport_Dimension."Source_Country" = 'Norway' AND
Destination_Airport_Dimension."Destination_Country" = 'Austria';
```

OUTPUT:



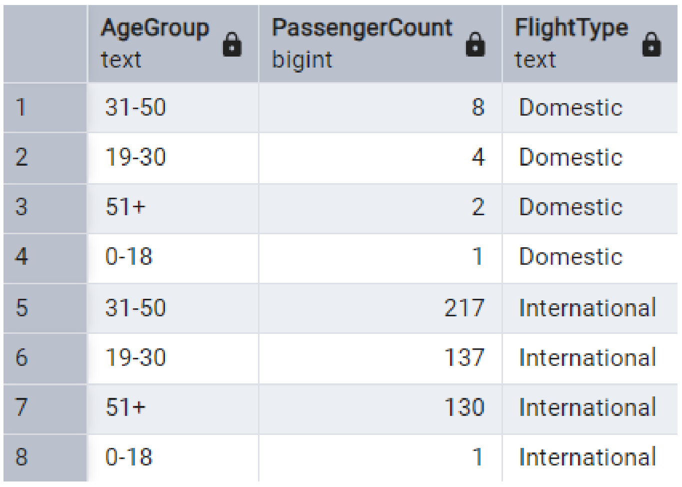
	AverageAge numeric
1	52.00

Figure 7 average age of people travelling from Norway to Austria

2. What is the frequency of different age groups travelling on International flights vs Domestic flights?

```
SELECT
CASE
WHEN Flight_Travel."Age" <= 18 THEN '0-18'
WHEN Flight_Travel."Age" BETWEEN 19 AND 30 THEN '19-30'
WHEN Flight_Travel."Age" BETWEEN 31 AND 50 THEN '31-50'
ELSE '51+'
END AS "AgeGroup",
COUNT(*) AS "PassengerCount",
CASE
WHEN Source_Airport_Dimension."Source_Country" =
Destination_Airport_Dimension."Destination_Country" THEN 'Domestic'
ELSE 'International'
END AS "FlightType"
FROM Flight_Travel
JOIN Source_Airport_Dimension ON Flight_Travel."SourceAirportID" =
Source_Airport_Dimension."SourceAirportID"
JOIN Destination_Airport_Dimension ON Flight_Travel."DestinationAirportID" =
Destination_Airport_Dimension."DestinationAirportID"
GROUP BY "AgeGroup", "FlightType"
ORDER BY "FlightType", "PassengerCount" DESC;
```

OUTPUT:



	AgeGroup text	PassengerCount bigint	FlightType text
1	31-50	8	Domestic
2	19-30	4	Domestic
3	51+	2	Domestic
4	0-18	1	Domestic
5	31-50	217	International
6	19-30	137	International
7	51+	130	International
8	0-18	1	International

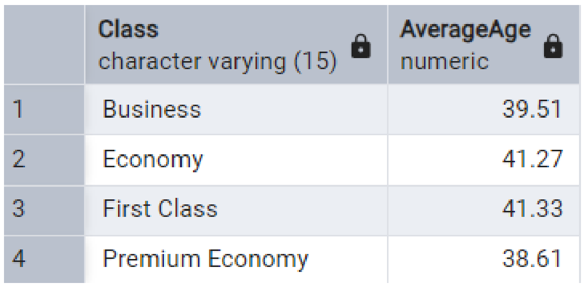
Figure 8 Frequency of different age groups on International vs Domestic Flights

3. What is the average age of passengers using different classes?

   To solve this problem following query has been written

```
SELECT Class_Dimension."Class", ROUND(AVG(Flight_Travel."Age"), 2) AS "AverageAge"
FROM Flight_Travel
JOIN Class_Dimension ON Flight_Travel."ClassID" = Class_Dimension."ClassID"
GROUP BY Class_Dimension."Class"
ORDER BY "Class";
```

OUTPUT:



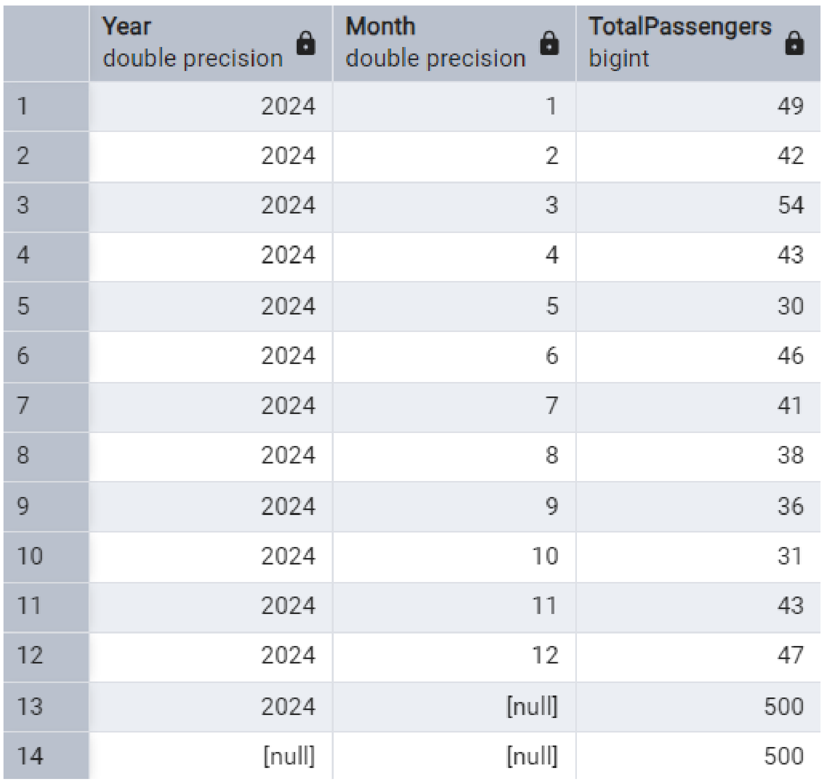
	Class character varying (15)	AverageAge numeric
1	Business	39.51
2	Economy	41.27
3	First Class	41.33
4	Premium Economy	38.61

Figure 9 Average age by travel class

8.2 ROLLUP

```
SELECT
EXTRACT(YEAR FROM Date_Dimension."Date") AS "Year",
EXTRACT(MONTH FROM Date_Dimension."Date") AS "Month",
COUNT(Flight_Travel."PassengerID") AS "TotalPassengers"
FROM Flight_Travel
JOIN Date_Dimension    ON Flight_Travel."DateID" = Date_Dimension."DateID"
GROUP BY ROLLUP (EXTRACT(YEAR FROM Date_Dimension."Date"), EXTRACT(MONTH FROM
Date_Dimension."Date"))
ORDER BY "Year", "Month";
```

OUTPUT:



	Year double precision	Month double precision	TotalPassengers bigint
1	2024	1	49
2	2024	2	42
3	2024	3	54
4	2024	4	43
5	2024	5	30
6	2024	6	46
7	2024	7	41
8	2024	8	38
9	2024	9	36
10	2024	10	31
11	2024	11	43
12	2024	12	47
13	2024	[null]	500
14	[null]	[null]	500

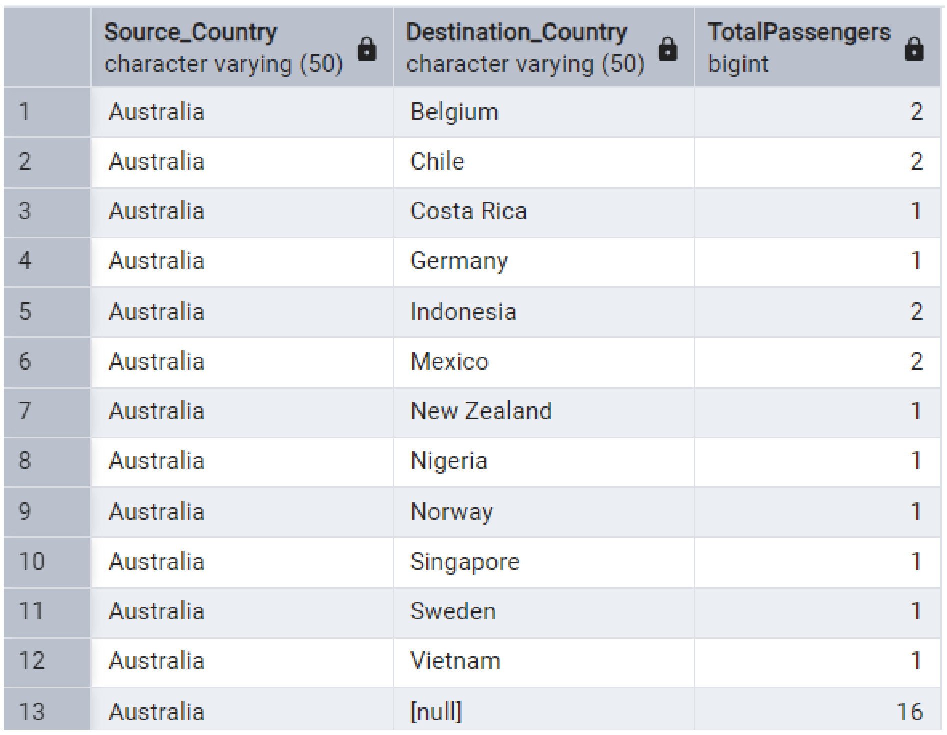
Figure 10 monthly and yearly total of passengers

8.3 CUBE

This query will return the total number of passengers traveling between different countries, along with subtotals and a grand total

```
SELECT
Source_Airport_Dimension."Source_Country",
Destination_Airport_Dimension."Destination_Country",
COUNT(Flight_Travel."PassengerID") AS "TotalPassengers"
FROM Flight_Travel
JOIN Source_Airport_Dimension ON Flight_Travel."SourceAirportID" =
Source_Airport_Dimension."SourceAirportID"
JOIN Destination_Airport_Dimension ON Flight_Travel."DestinationAirportID" =
Destination_Airport_Dimension."DestinationAirportID"
GROUP BY CUBE (Source_Airport_Dimension."Source_Country",
Destination_Airport_Dimension."Destination_Country")
ORDER BY Source_Airport_Dimension."Source_Country",
Destination_Airport_Dimension."Destination_Country";
```

OUTPUT:



	Source_Country character varying (50)	Destination_Country character varying (50)	TotalPassengers bigint
1	Australia	Belgium	2
2	Australia	Chile	2
3	Australia	Costa Rica	1
4	Australia	Germany	1
5	Australia	Indonesia	2
6	Australia	Mexico	2
7	Australia	New Zealand	1
8	Australia	Nigeria	1
9	Australia	Norway	1
10	Australia	Singapore	1
11	Australia	Sweden	1
12	Australia	Vietnam	1
13	Australia	[null]	16

Figure 11 passengers travelling from specific source to different destinations

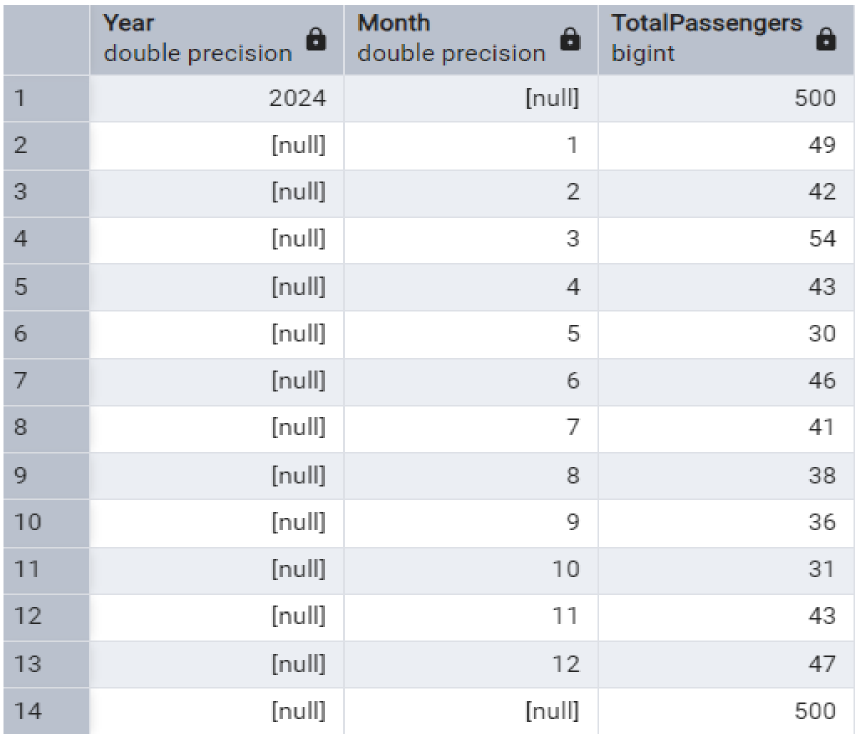
This is only the result about Australia, but there are other countries as well and the output contains a total of 488 rows.

8.4 GROUPING SET

This query will return total passengers for each year(since the data is of 2024 so the total would be for year 2024), total passenger for each month, and final grand total

```
SELECT
EXTRACT(YEAR FROM Date_Dimension."Date") AS "Year",
EXTRACT(MONTH FROM Date_Dimension."Date") AS "Month",
COUNT(Flight_Travel."PassengerID") AS "TotalPassengers"
FROM Flight_Travel
JOIN Date_Dimension ON Flight_Travel."DateID" = Date_Dimension."DateID"
GROUP BY GROUPING SETS ((EXTRACT(YEAR FROM Date_Dimension."Date")),
(EXTRACT(MONTH FROM Date_Dimension."Date")), ())
ORDER BY "Year", "Month";
```

OUTPUT:



	Year double precision	Month double precision	TotalPassengers bigint
1	2024	[null]	500
2	[null]	1	49
3	[null]	2	42
4	[null]	3	54
5	[null]	4	43
6	[null]	5	30
7	[null]	6	46
8	[null]	7	41
9	[null]	8	38
10	[null]	9	36
11	[null]	10	31
12	[null]	11	43
13	[null]	12	47
14	[null]	[null]	500

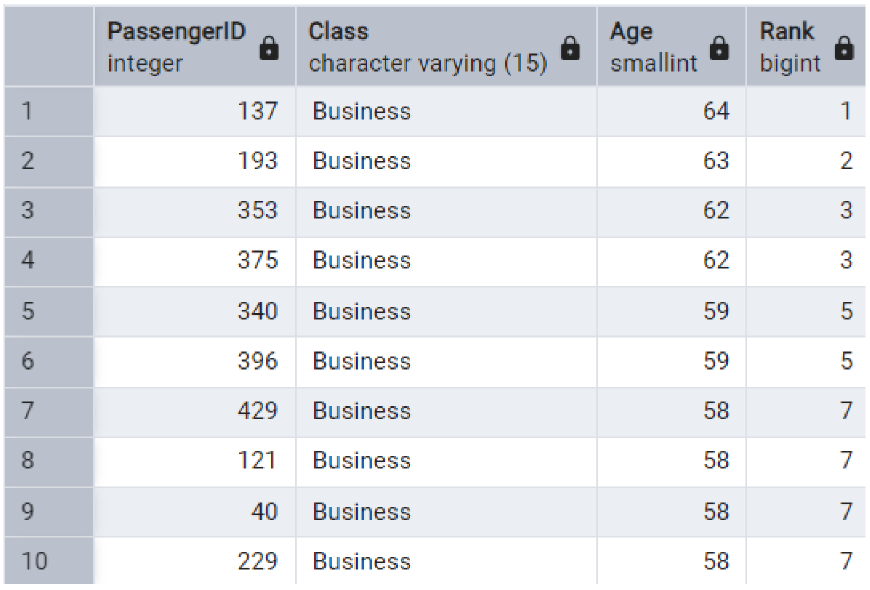
Figure 12 monthly and yearly total of passengers using grouping set

8.5 RANK

This query will assign a rank to each passenger based on their age within each flight class, with the oldest passenger ranked first.

```
SELECT Flight_Travel."PassengerID",
Class_Dimension."Class",
Flight_Travel."Age",
RANK() OVER (PARTITION BY Class_Dimension."Class" ORDER BY Flight_Travel."Age"
DESC) AS "Rank"
FROM Flight_Travel
JOIN Class_Dimension ON Flight_Travel."ClassID" = Class_Dimension."ClassID"
ORDER BY Class_Dimension."Class", "Rank";
```

OUTPUT:



	PassengerID integer	Class character varying (15)	Age smallint	Rank bigint
1	137	Business	64	1
2	193	Business	63	2
3	353	Business	62	3
4	375	Business	62	3
5	340	Business	59	5
6	396	Business	59	5
7	429	Business	58	7
8	121	Business	58	7
9	40	Business	58	7
10	229	Business	58	7

Figure 13 Rank by age per travel class

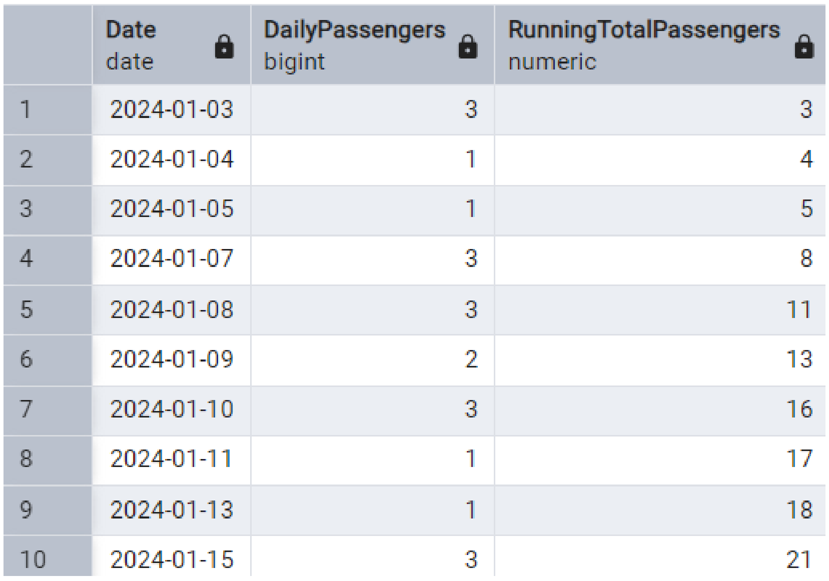
This is only 10 rows from the result of business class, there are a total of 500 rows in the output

8.6 WINDOWING

This query will give a daily count of passengers and a running total of passengers over time.

```
SELECT
Date_Dimension."Date",
COUNT(Flight_Travel."PassengerID") AS "DailyPassengers",
SUM(COUNT(Flight_Travel."PassengerID")) OVER (ORDER BY Date_Dimension."Date") AS
"RunningTotalPassengers"
FROM Flight_Travel
JOIN Date_Dimension ON Flight_Travel."DateID" = Date_Dimension."DateID"
GROUP BY Date_Dimension."Date"
ORDER BY Date_Dimension."Date";
```

OUTPUT:



	Date date	DailyPassengers bigint	RunningTotalPassengers numeric
1	2024-01-03	3	3
2	2024-01-04	1	4
3	2024-01-05	1	5
4	2024-01-07	3	8
5	2024-01-08	3	11
6	2024-01-09	2	13
7	2024-01-10	3	16
8	2024-01-11	1	17
9	2024-01-13	1	18
10	2024-01-15	3	21

Figure 14 Daily and and running total of passengers

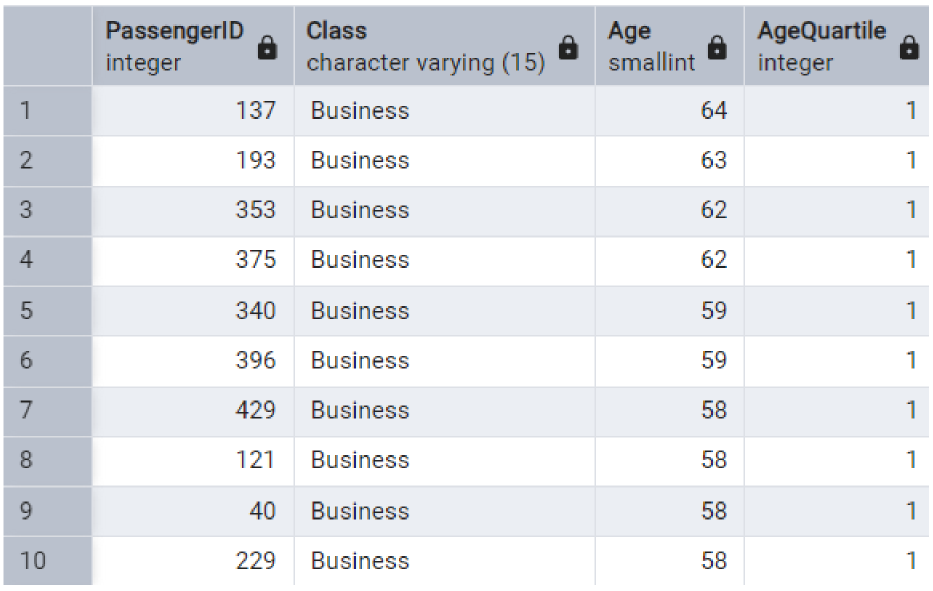
These are only 10 rows from the result, but there are a total of 265 rows more in the above given output.

8.7 NTILE

This query will divide passengers into 4 groups (quartiles) based on their age within each flight class. The quartiles are ordered from the oldest to the youngest passenger in each class.

```
SELECT
Flight_Travel."PassengerID",
Class_Dimension."Class",
Flight_Travel."Age",
NTILE(4) OVER (PARTITION BY Class_Dimension."Class" ORDER BY Flight_Travel."Age"
DESC) AS "AgeQuartile"
FROM Flight_Travel
JOIN Class_Dimension ON Flight_Travel."ClassID" = Class_Dimension."ClassID"
ORDER BY Class_Dimension."Class", "AgeQuartile";
```

OUTPUT:



	PassengerID integer	Class character varying (15)	Age smallint	AgeQuartile integer
1	137	Business	64	1
2	193	Business	63	1
3	353	Business	62	1
4	375	Business	62	1
5	340	Business	59	1
6	396	Business	59	1
7	429	Business	58	1
8	121	Business	58	1
9	40	Business	58	1
10	229	Business	58	1

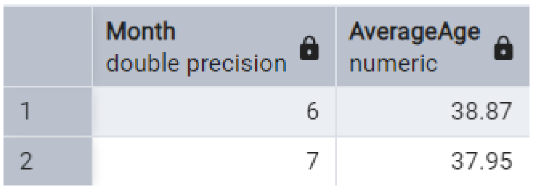
Figure 15 Quartiles of passengers based on age

8.8 **PERIOD-TO-PERIOD**

This query will give the average age of passengers for the months of June (6) and July

```
SELECT
EXTRACT(MONTH FROM d."Date") AS "Month", AVG(ft."Age") AS "AverageAge"
FROM Flight_Travel ft
JOIN Date_Dimension d ON ft."DateID" = d."DateID"
WHERE EXTRACT(MONTH FROM d."Date") IN (6, 7)
GROUP BY EXTRACT(MONTH FROM d."Date")
ORDER BY "Month";
```

OUTPUT:



	Month double precision	AverageAge numeric
1	6	38.87
2	7	37.95

Figure 16 Average age of passengers travelling in June and July

9 VISUAL ANALYSIS

Below is the plot for Business Question 2(Domestic Users)

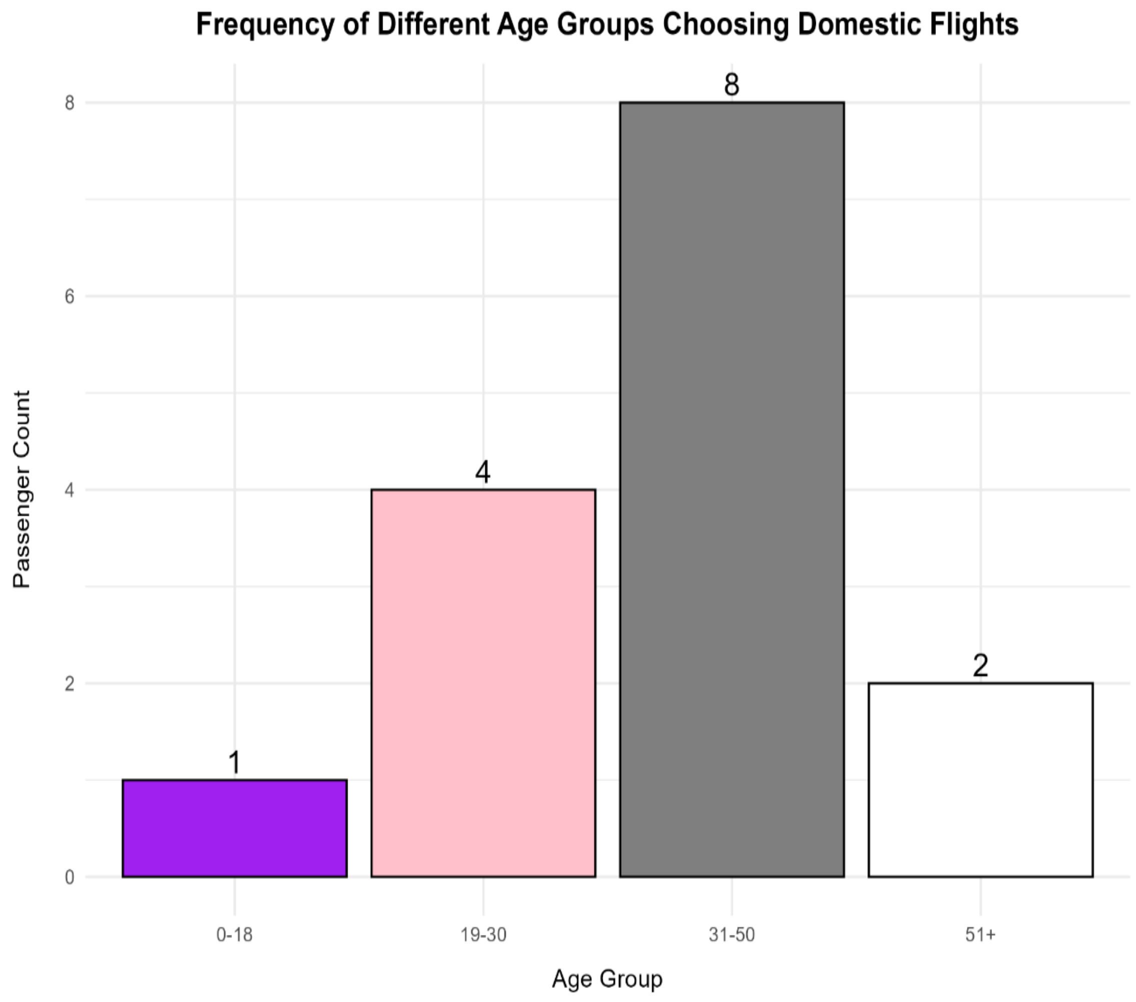


Figure 17 Domestic flight users by age group

This is the plot for Business Question 2(International Users)

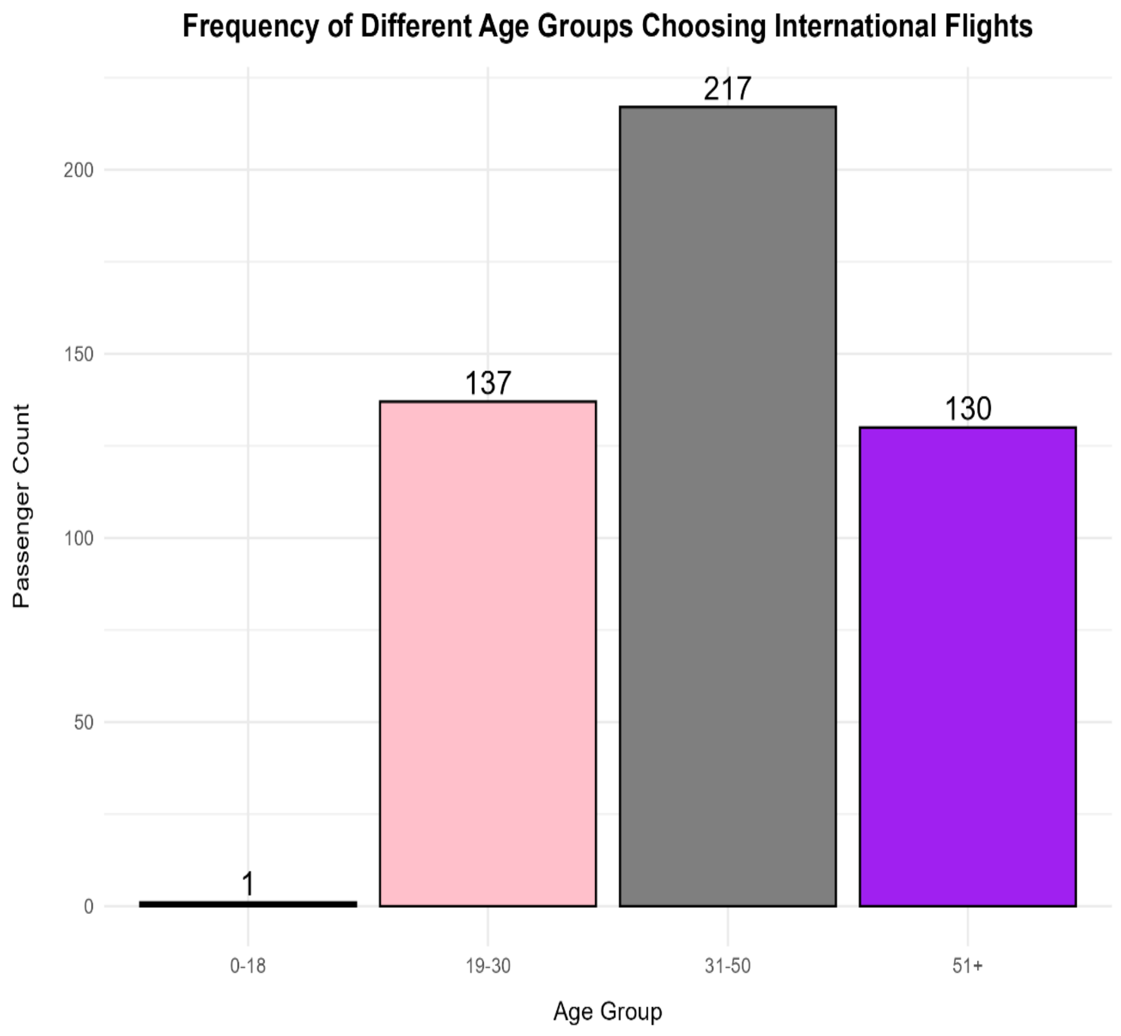


Figure 18 International flight users by age group

This is the plot for Business Question 3

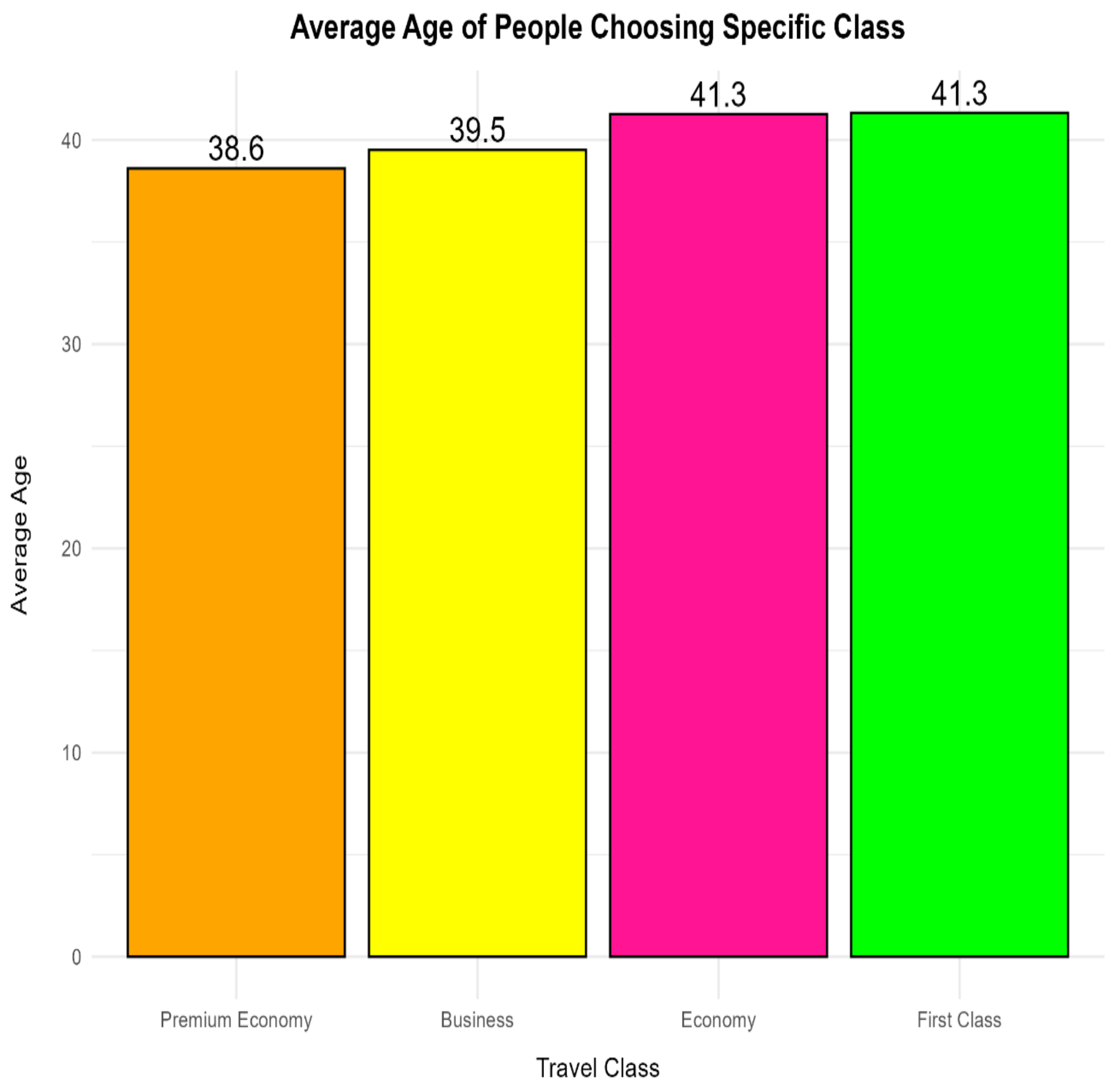


Figure 19 Average age by class

10 Difficulties

About ETL process, we have to design how to process and transfer dataset into database server, we decide to use a temporary table and then separately fulfill these dimension tables and fact table. When we connect Postgresql server, we meet few technical issues, passwords reset few times.

The most problematic part was to fill the data into the Flight_Travel fact table which i did by selecting and importing the already saved entire data from the CSV file which I created, to the short_term table (a temporary table made for the purpose to fill the data into Flight_Travel fact table) with the same column names as in the CSV file and then inserted the data to the Flight_Travel fact table from the short_term table and then filled the “Age” column of Flight_Travel fact table .