

# Aviation Industry Queries

FREE COPY

# 40+ SQL PROJECTS FOR AIRLINES

(WITH SOLUTIONS)

2025



**40+**

**SQL PROJECTS**

**FOR**

**AIRLINES**

**(WITH SOLUTIONS)**

**2025**

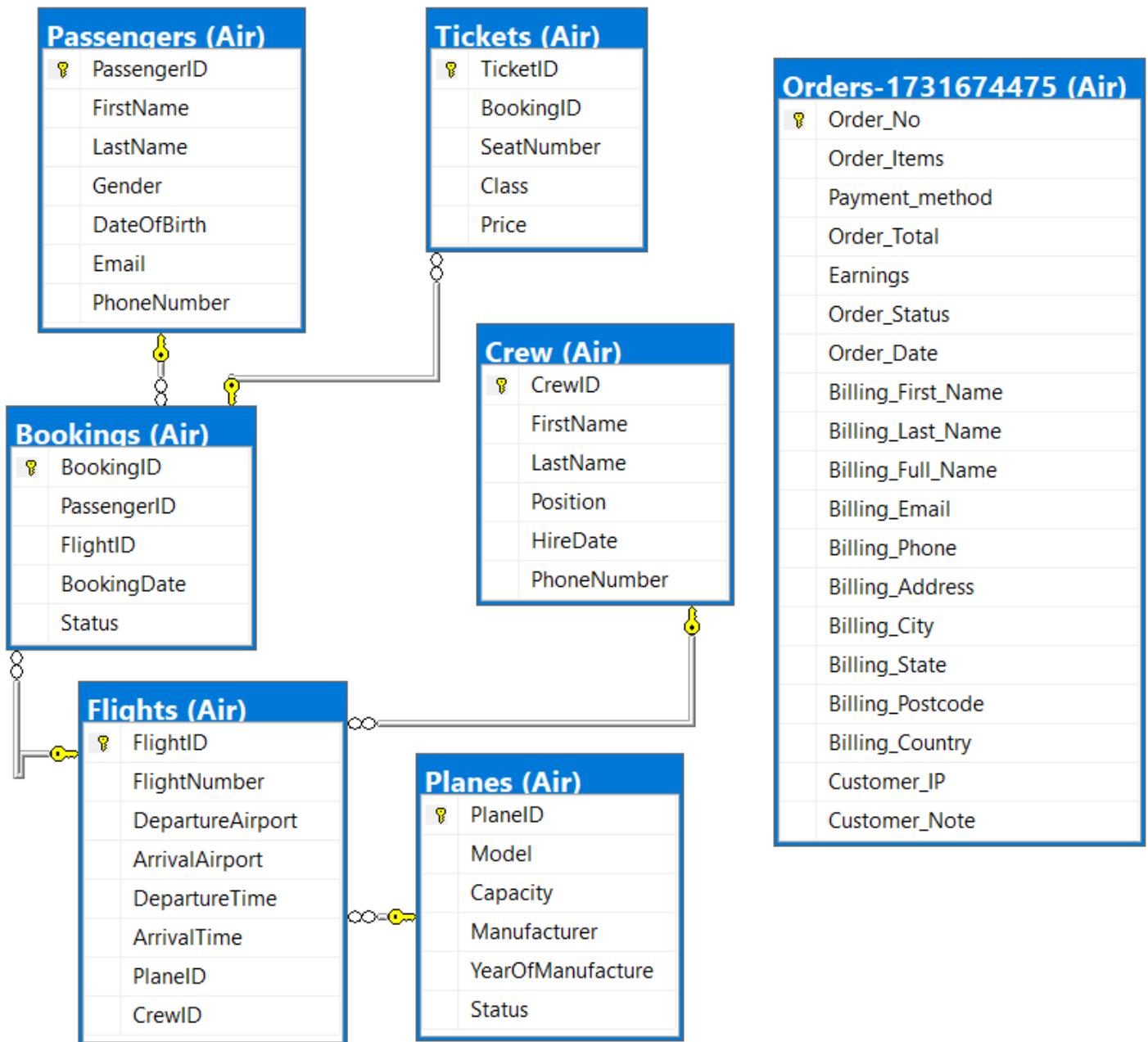
[AfriqIQ.com](http://AfriqIQ.com)

*Copyright © 2025 Afriq IQ*

*All rights reserved.*

*No part of this book shall be reproduced, stored in a retrieval system, or transmitted by any means, electronic, mechanical, photocopying, recording, or otherwise, without written permission from the publisher. No patent liability is assumed with respect to the use of the information contained herein. Although every precaution has been taken in the preparation of this book, the author assumes no responsibility for errors or omissions. Nor is any liability assumed for damages resulting from the use of the information contained herein.*

# Database Diagram



## Table of Contents

<b>About the Author</b> .....	2
<b>Important information</b> .....	3
<b>Business Scenarios</b> .....	5

AfriqIQ.com

## About Afriq IQ

[AfriqIQ.com](https://AfriqIQ.com) is the pioneering platform dedicated to unearthing and amplifying the AI revolution driven by African and diaspora innovators.

AfriqIQ serves as a dynamic directory and hub for over 2000+ startups, prototypes, and beta tools, bridging continental creators Africa and beyond.

Born from a mission to democratise visibility in the AI world, AfriqIQ aggregates public insights, fosters community connections, and spotlights pan-African and Diaspora solutions that tackle real-world challenges like healthcare equity, sustainable agritech, and financial inclusion.

Through partnerships with accelerators, hubs and features such as VC matching, job boards, and ecosystem analytics, the platform empowers users to claim profiles, unlock premium tools, and collaborate across borders.

This ebook is AfriqIQ's invitation to the tech movement. Whether you're a founder seeking exposure, an investor hunting trends, or a researcher mapping the landscape, AfriqIQ equips you to navigate and shape Africa's AI odyssey.

Dive in at [AfriqIQ.com](https://AfriqIQ.com) and submit your startup for free, explore the directory, and join the network. Africa's AI story is global, collaborative, and unstoppable. Together, we're building it.

## **Important information**

Welcome to "**SQL Projects on Aviation**". This book is designed to be your comprehensive guide to mastering the art of SQL using practical business scenarios. Whether you're a beginner looking to build a strong foundation or an experienced practitioner aiming to sharpen your skills.

SQL (Structured Query Language) is the backbone of managing and manipulating data in modern databases, making it an essential skill for anyone working with data-driven applications.

This book covers the commands used in Data Query language (DQL) in most parts. Other categories of SQL commands namely, Data Manipulation Language (DML), Data Definition Language (DDL) and Data Control Language (DCL) will be covered in a separate book.

Solutions to the problems in this book are given at the end of this book (before the Appendix).

***NOTE:** The commands employed in the "sample" solutions primarily align with Microsoft's SQL Server conventions, although suggestions are provided on how the syntax might be adapted to suit other frequently used SQL platforms such as MySQL, Oracle, PostgreSQL, etc.*

### **Database used in this book.**

The database used in this book is based solely on the aviation business as the title suggests. Access to the database should have already been acquired when the book was downloaded. The Database Diagram given above (after the copyright page) can help in understanding the relationships between the tables.

## **SQL Server installation and database restoration**

In the Appendix, you'll find comprehensive instructions for installing SQL Server 2019 (or later) on Windows 10 and later versions, as well as the step-by-step process on how to restore a database using the Microsoft AdventureWorks database as an example.

## **Who This Book Is For**

This "**SQL Projects on Aviation**" assumes some fundamental knowledge of SQL's basic commands at the least. Nevertheless, it is intended for a wide audience, including beginners, students, educators and database professionals and anyone who uses SQL in their everyday lives and business.

## **Feedback**

We value your feedback and suggestions. If you have any comments, questions, or ideas for improvement, please don't hesitate to reach out on any of the following platforms:

- Website: [AfriqlQ.com](http://AfriqlQ.com)
- Twitter:
- LinkedIn

Your input will help us enhance future editions of this book.

Thank you for choosing " SQL Projects on Aviation". We hope this book serves as a valuable resource in your journey to become a proficient SQL practitioner.

Happy querying!

# **Business Scenarios**

## **1. Business Scenario Q1**

### **Passenger Flight Details Lookup**

An airline company wants to implement a simple query to allow their customer service agents to quickly retrieve information about all flights for Michelle Taylor. The agents need to know which flight a passenger is booked on, including the flight date, departure time, and the plane they will be flying on. This information helps the agents provide accurate assistance to passengers regarding their travel details.

## **2. Business Scenario Q2**

### **Booking Confirmation Details**

An airline company wants to provide a system where passengers can look up their booking confirmation details, including their seat number, class, flight information, and the price paid. This allows passengers to verify their booking and ensure all details are correct. Get these information for Booking ID B0019.

## **3. Business Scenario Q3**

### **Crew Schedule Overview**

An airline company needs a system for their crew members to quickly check their work schedules. This includes the flights they are assigned to, the departure and arrival times, and the plane they will be operating. This information is crucial for crew members to manage their work shifts efficiently. Get this information for a crew with ID C0123

## **4. Business Scenario Q4**

### **Flight Occupancy Check**

An airline company needs a system to check the occupancy of their flights. This includes knowing how many seats are booked and how many are still available for a particular flight (ID F0897). This information helps the airline manage overbooking, seat allocation, and plan for future demand.

## **5. Business Scenario Q5**

### **Passenger Contact Information for Upcoming Flights**

An airline company needs a system to quickly retrieve the contact information of passengers for upcoming flights. Focus on retrieving the contact information of passengers booked on a specific flight, identified by its flight number, 778. This is crucial for sending notifications about flight changes, delays, or other important information.

## **6. Business Scenario Q6**

### **Passenger List for a Specific Flight**

An airline company needs to retrieve a list of all passengers on a flight 505. This is useful for check-in procedures, boarding, and ensuring all passengers are accounted for.

## **7. Business Scenario Q7**

### **List of All Planes and Their Assigned Flights**

An airline company wants to generate a list of all planes along with the details of their assigned flights. This includes the flight number, departure and arrival times, and the

plane model. This information helps in managing the fleet and scheduling maintenance.

## **8. Business Scenario Q8**

### **Passenger List for Specific Flight Date**

An airline company wants to retrieve a list of all passengers booked on flights departing on 14<sup>th</sup> May 2022. This is useful for operational planning and ensuring that all passengers are accounted for.

## **9. Business Scenario Q9**

### **Passenger Information for All Booked Flights**

An airline company wants to retrieve a list of all passengers along with their booked flights. This is useful for operational planning and ensuring that all passengers are accounted for.

## **10. Business Scenario Q10**

### **Monthly Flight Revenue Analysis**

An airline company wants to analyse the revenue generated from flights on a monthly basis. This includes calculating the total revenue from ticket sales for each flight, grouping the results by month, and identifying the top-performing flights in terms of revenue. This information is useful for financial analysis, planning, and decision-making.

## **11. Business Scenario Q11**

### **Crew Utilisation and Assignment Analysis**

An airline company wants to analyse the utilisation of their crew members. This includes identifying the number of flights each crew member has been assigned to for the year 2022, the total flight hours logged by each crew member, and ranking the crew members based on their utilisation. This information helps in workforce planning, identifying overworked crew members, and ensuring balanced crew assignments.

## **12. Business Scenario Q12**

### **Passenger Loyalty Program Analysis**

An airline company wants to analyse the loyalty of their passengers by tracking the number of flights taken by each passenger for the year 2021 and calculating the total amount spent on tickets. This helps in identifying the most frequent flyers and rewarding them with loyalty points or other benefits.

## **13. Business Scenario Q13**

### **Revenue Analysis by Flight Class**

An airline company wants to analyse the revenue generated from ticket sales for each flight class (e.g., Economy, Business, First Class). This includes calculating the total revenue per class, the number of tickets sold per class, and identifying the most profitable class. This information helps in pricing strategy and capacity planning.

## **14. Business Scenario Q14**

### **Passenger Demographic Analysis for Marketing**

An airline company wants to analyse the demographics of their passengers to tailor marketing campaigns and improve customer service. This includes understanding the gender distribution, age distribution, and contact information of passengers who have travelled in the year 2021. This information helps in targeting specific passenger segments with personalized offers and services.

## **15. Business Scenario Q15**

### **Revenue and Seat Utilisation Analysis for Each Flight**

An airline company wants to analyse the revenue and seat utilization for each flight. This includes calculating the total revenue generated from ticket sales, the number of seats booked, and the percentage of seat utilization for each flight. This information helps in understanding the financial performance and occupancy rate of different flights, informing decisions on pricing, scheduling, and capacity planning.

## **16. Business Scenario Q16**

### **Frequent Flyer Program Analysis (Simplified)**

An airline company wants to analyse the activity of their frequent flyer program. This includes identifying passengers who have taken the most flights and determining their eligibility for different tiers (Platinum for total flights of 10 and above; Gold for total flights of 5 and above, Silver for total flights of 2 and above and Bronze for anything else) of the frequent flyer program based on the number of flights taken in the year 2021. This information helps in rewarding loyal customers and promoting the frequent flyer program.

## **17. Business Scenario Q17**

### **Monthly Revenue and Flight Count per Airport**

An airline company wants to analyse the monthly revenue and number of flights for each airport. This includes calculating the total revenue generated from ticket sales for flights departing from each airport and the number of flights departing each month. This information helps in understanding the performance of different airports and informs decisions on resource allocation and scheduling.

## **18. Business Scenario Q18**

### **Passenger Booking Trends Analysis**

An airline company wants to analyse passenger booking trends to understand the peak booking periods and the average booking lead time. This includes calculating the number of bookings made each month, the average time between booking and flight departure, and identifying patterns in booking behaviour. This information helps in optimising marketing strategies and managing operational efficiency.

#### **Additional Queries for Comprehensive Analysis:**

- **Daily Booking Trends:**
- Booking Trends by Passenger Segment (e.g., Business vs. Economy Class)

## **19. Business Scenario Q19**

### **Analysis of Flight Delays and Their Impact on Revenue**

An airline company wants to analyse the impact of flight delays on revenue. This includes identifying the average delay time for each flight, the total revenue lost due to

delays, and the number of delayed flights. This information helps in understanding the financial impact of delays and improving operational efficiency.

**Additional Queries for Comprehensive Analysis:**

- **Monthly Impact of Delays:**
- Impact of Delays by Airport:

**20. Business Scenario Q20**

**Top Revenue-Generating Flights**

An airline company wants to identify the top revenue-generating flights. This includes calculating the total revenue generated from ticket sales for each flight and ranking the flights based on their revenue. This information helps in identifying the most profitable routes and optimising flight schedules.

**Additional Queries for Comprehensive Analysis:**

- **Top Revenue-Generating Flights by Month:**
- Top Revenue-Generating Flights by Airport:

**21. Business Scenario Q21**

**Frequent Flyer Analysis by Total Flights and Total Spend**

An airline company wants to identify their most frequent flyers by analysing both the number of flights taken and the total amount spent on tickets. This information is useful for designing loyalty programs and targeting top customers with special offers and benefits.

## **22. Business Scenario Q22**

### **Analysis of Plane Utilisation and Efficiency**

An airline company wants to analyse the utilisation and efficiency of their planes. This includes calculating the total number of flights each plane has completed, the total hours flown, and identifying the planes with the highest and lowest utilisation rates (i.e. if the number of flights is less than zero then that is considered underutilised and when greater than 3 then that is overutilised and anything else is well-utilised). This information helps in optimising the fleet management and scheduling.

#### **Additional Queries for Comprehensive Analysis:**

- **Plane Utilisation by Month:**
- Top 5 Most Utilised Planes:
- Under-utilised Planes:

## **23. Business Scenario Q23**

### **Passenger Demographics and Travel Patterns**

An airline company wants to analyse the demographics and travel patterns of their passengers. This includes identifying the gender distribution, age distribution, and the most popular travel routes. This information helps in tailoring marketing campaigns and improving customer service by understanding the passenger base better.

## **24. Business Scenario Q24**

### **Age Distribution of Passengers**

You are a data analyst for an airline company. The business wants to understand the age distribution of passengers who have confirmed bookings to tailor marketing campaigns by age group. Age should be calculated based on the current date and grouped into these bands: 0-17, 18-24, 25-34, 35-44, 45-54, 55-64, and 65+. Retrieve the age group and the count of confirmed passengers in each group

## **25. Business Scenario Q25**

### **Most Popular Travel Routes**

You are a data analyst for an airline company. The business wants to identify the most popular travel routes based on confirmed bookings to optimise flight scheduling. Write a query to retrieve the departure airport, arrival airport, and the count of confirmed flights for each route.

## **26. Business Scenario Q26**

### **Booking Status Analysis**

An airline company wants to analyse the booking statuses over a specific period to understand trends and identify potential issues with booking confirmations and cancellations. This includes calculating the total number of bookings, confirmed bookings, pending and cancelled bookings for each month. This information helps in improving booking management and customer service.

## **27. Business Scenario Q27**

### **Daily Booking Status Analysis**

You are a data analyst for an airline company. The business wants to analyse daily booking status to monitor performance and identify trends. Write a query to retrieve the booking day, total bookings, confirmed bookings, cancelled bookings, and pending bookings for each day, ordered by booking day.

## **28. Business Scenario Q28**

### **Booking Status Breakdown by Status Type**

You are a data analyst for an airline company. The business wants to analyse the breakdown of booking statuses to assess overall booking health. Write a query to retrieve each status type and the total number of bookings for that status, ordered by the total bookings.

## **29. Business Scenario Q29**

### **Weekly Booking Status Analysis**

You are a data analyst for an airline company. The business wants to analyse weekly booking status to monitor performance and identify trends. Write a query to retrieve the year, week, total bookings, confirmed bookings, cancelled bookings, and pending bookings for each week.

## **30. Business Scenario Q30**

### **Booking Volume Trends Analysis**

An airline company wants to analyse the volume of bookings over time to understand trends in passenger demand. This includes identifying peak booking periods, monthly booking trends, and the overall growth or decline in booking volume. This information helps in planning marketing campaigns, adjusting flight schedules, and optimising resource allocation.

## **31. Business Scenario Q31**

### **Daily Booking Volume Analysis**

You are a data analyst for an airline company. The business wants to analyse daily booking volume to monitor performance and identify trends. Write a query to retrieve the booking day and total bookings for each day.

## **32. Business Scenario Q32**

### **Weekly Booking Volume Analysis**

You are a data analyst for an airline company. The business wants to analyse weekly booking volume to monitor performance and identify trends. Write a query to retrieve the year, week, and total bookings for each week.

### **33. Business Scenario Q33**

#### **Annual Booking Volume Analysis**

The You are a data analyst for an airline company. The business wants to analyse annual booking volume to monitor performance and identify trends. Write a query to retrieve the year and total bookings for each year.

### **34. Business Scenario Q34**

#### **Fleet Composition and Status Analysis**

An airline company wants to analyse the composition and status of their fleet. This includes identifying the distribution of planes by manufacturer, the age of the planes, and the current operational status of each plane. This information helps in making decisions about fleet upgrades, maintenance schedules, and retirement of older planes.

### **35. Business Scenario Q35**

#### **Current Operational Status of Planes**

You are a data analyst for an airline company. The business wants to analyse the current operational status of planes to monitor fleet availability. Write a query to retrieve each status and the total number of planes for that status.

## **36. Business Scenario Q36**

### **Details of Planes Manufactured by a Specific Manufacturer**

You are a data analyst for an airline company. The business wants to retrieve details of planes manufactured by a specific manufacturer, such as 'Boeing', to assess the fleet composition. Write a query to select the plane ID, model, capacity, year of manufacture, and status for planes made by Boeing.

## **37. Business Scenario Q37**

### **Capacity Distribution of Planes**

You are a data analyst for an airline company. The business wants to analyse the capacity distribution of planes to assess fleet efficiency. Write a query to retrieve the capacity group (i.e. Capacity less than or equal to 100, between 101 and 200, between 201 and 300 and greater than 300) and total planes for each group.

## **38. Business Scenario Q38**

### **Age distribution of Planes**

You are a data analyst for an airline company. The business wants to analyse the age distribution of planes to assess fleet maintenance needs. Age should be calculated based on the current date and grouped into these bands: 0-5 years, 6-10 years, 11-20 years, and 20+. Write a query to retrieve the age group and total planes for each group.

## **39. Business Scenario Q39**

### **Revenue Analysis by Ticket Class**

An airline company wants to analyse the revenue generated from ticket sales by different classes (e.g., Economy, Business, First Class). This includes calculating the total revenue for each class, the average price per ticket, and the number of tickets sold per class. This information helps in understanding the financial performance of different classes and optimising pricing strategies.

## **40. Business Scenario Q40**

### **Revenue Analysis by Ticket Class and Month**

You are a data analyst for an airline company. The business wants to analyse revenue by ticket class and month to optimise pricing strategies and identify seasonal trends. Write a query to retrieve the class, year, month, total tickets sold, total revenue, and average ticket price for each combination.

## **41. Business Scenario Q41**

### **Top 5 Most Expensive Tickets Sold**

You are a data analyst for an airline company. The business wants to identify the top 5 most expensive tickets sold to review premium offerings. Write a query to retrieve the ticket ID, class, and price for the 5 most expensive tickets.

**42. Business Scenario Q42**

**Revenue Distribution by Price Range**

You As the data analyst for an Airline, you're reviewing ticket sales data to understand how revenue is performing across different price segments. This will help identify high-performing price ranges and guide future dynamic pricing adjustments. Craft a SQL query that categorises tickets into price ranges (0-100, 101-200, 201-300, 301-400, and 400+) and calculates the total tickets sold and total revenue for each range.

# SOLUTIONS

So

---

```
1 --Q1.
2 SELECT
3     CONCAT(p.FirstName, ' ', p.LastName) AS PassengerName,
4     f.FlightNumber,
5     f.DepartureTime,
6     f.ArrivalTime,
7     pl.Model AS PlaneModel
8 FROM
9     Air.Passengers p
10 JOIN
11     Air.Bookings b ON p.PassengerID = b.PassengerID
12 JOIN
13     Air.Flights f ON b.FlightID = f.FlightID
14 JOIN
15     Air.Planes pl ON f.PlaneID = pl.PlaneID
16 WHERE
17     p.FirstName = 'Michelle' AND p.LastName = 'Taylor';
18
19
20 --Q2.
21 SELECT
22     CONCAT(p.FirstName, ' ', p.LastName) AS PassengerName,
23     f.FlightNumber,
24     f.DepartureTime,
25     t.SeatNumber,
26     t.Class,
27     t.Price,
28     pl.Model AS PlaneModel
29 FROM
30     Air.Passengers p
31 JOIN
32     Air.Bookings b ON p.PassengerID = b.PassengerID
33 JOIN
34     Air.Tickets t ON b.BookingID = t.BookingID
35 JOIN
36     Air.Flights f ON b.FlightID = f.FlightID
37 JOIN
38     Air.Planes pl ON f.PlaneID = pl.PlaneID
39 WHERE
40     b.BookingID = 'B0019';
41
42
43 --Q3.
44
45 SELECT
46     CONCAT(c.FirstName, ' ', c.LastName) AS CrewMemberName,
47     f.FlightNumber,
48     f.DepartureTime,
49     f.ArrivalTime,
50     pl.Model AS PlaneModel
51 FROM
52     Air.Crew c
53 JOIN
```

# SOLUTIONS

So

---

```
54     Air.Flights f ON c.CrewID = f.CrewID
55 JOIN
56     Air.Planes pl ON f.PlaneID = pl.PlaneID
57 WHERE
58     c.CrewID = 'C0123';
59
60
61
62 --Q4.
63 SELECT
64     f.FlightNumber,
65     COUNT(t.TicketID) AS SeatsBooked,
66     pl.Capacity AS TotalCapacity,
67     (pl.Capacity - COUNT(t.TicketID)) AS AvailableSeats
68 FROM
69     Air.Flights f
70 JOIN
71     Air.Bookings b ON f.FlightID = b.FlightID
72 JOIN
73     Air.Tickets t ON b.BookingID = t.BookingID
74 JOIN
75     Air.Planes pl ON f.PlaneID = pl.PlaneID
76 WHERE
77     f.FlightID = 'F0897'
78 GROUP BY
79     f.FlightNumber, pl.Capacity;
80
81
82 --Q5.
83 SELECT
84     CONCAT(p.FirstName, ' ', p.LastName) AS PassengerName,
85     p.Email,
86     p.PhoneNumber,
87     f.FlightNumber,
88     f.DepartureTime,
89     f.ArrivalTime
90 FROM
91     Air.Passengers p
92 JOIN
93     Air.Bookings b ON p.PassengerID = b.PassengerID
94 JOIN
95     Air.Flights f ON b.FlightID = f.FlightID
96 WHERE
97     f.FlightNumber = '778';
98
99
100 --Q6.
101 SELECT
102     CONCAT(p.FirstName, ' ', p.LastName) AS PassengerName,
103     p.Gender,
104     p.DateOfBirth,
105     p.Email,
106     p.PhoneNumber,
```

# SOLUTIONS

So

---

```
107      f.FlightNumber,
108      f.DepartureTime,
109      f.ArrivalTime
110  FROM
111      Air.Passengers p
112  JOIN
113      Air.Bookings b ON p.PassengerID = b.PassengerID
114  JOIN
115      Air.Flights f ON b.FlightID = f.FlightID
116  WHERE
117      f.FlightNumber = '505';
118
119  --Q7.
120  SELECT
121      pl.PlanetID,
122      pl.Model,
123      pl.Manufacturer,
124      pl.YearOfManufacture,
125      f.FlightNumber,
126      f.DepartureAirport,
127      f.ArrivalAirport,
128      f.DepartureTime,
129      f.ArrivalTime
130  FROM
131      Air.Planes pl
132  LEFT JOIN
133      Air.Flights f ON pl.PlanetID = f.PlanetID
134  ORDER BY
135      pl.PlanetID, f.DepartureTime;
136
137
138  --Q8.
139  SELECT
140      CONCAT(p.FirstName, ' ', p.LastName) AS PassengerName,
141      p.Email,
142      p.PhoneNumber,
143      f.FlightNumber,
144      f.DepartureTime,
145      f.ArrivalTime
146  FROM
147      Air.Passengers p
148  JOIN
149      Air.Bookings b ON p.PassengerID = b.PassengerID
150  JOIN
151      Air.Flights f ON b.FlightID = f.FlightID
152  WHERE
153      CAST(f.DepartureTime AS DATE) = '2022-05-14'
154
155  --Q9.
156  SELECT
157      CONCAT(p.FirstName, ' ', p.LastName) AS PassengerName,
158      p.Email,
159      p.PhoneNumber,
```

# SOLUTIONS

So

---

```
160     f.FlightNumber,
161     f.DepartureTime,
162     f.ArrivalTime,
163     f.DepartureAirport,
164     f.ArrivalAirport
165 FROM
166     Air.Passengers p
167 JOIN
168     Air.Bookings b ON p.PassengerID = b.PassengerID
169 JOIN
170     Air.Flights f ON b.FlightID = f.FlightID;
171
172
173 --Q10.
174 WITH MonthlyRevenue AS (
175     SELECT
176         f.FlightNumber,
177         YEAR(f.DepartureTime) AS Year,
178         MONTH(f.DepartureTime) AS Month,
179         SUM(t.Price) AS TotalRevenue
180     FROM
181         Air.Flights f
182     JOIN
183         Air.Bookings b ON f.FlightID = b.FlightID
184     JOIN
185         Air.Tickets t ON b.BookingID = t.BookingID
186     WHERE
187         b.Status = 'Confirmed'    -- Only consider confirmed bookings
188     GROUP BY
189         f.FlightNumber,
190         YEAR(f.DepartureTime),
191         MONTH(f.DepartureTime)
192 )
193     SELECT
194         Year,
195         Month,
196         FlightNumber,
197         TotalRevenue,
198         RANK() OVER (PARTITION BY Year, Month ORDER BY TotalRevenue DESC) AS RevenueRank
199     FROM
200         MonthlyRevenue
201     ORDER BY
202         Year,
203         Month,
204         RevenueRank;
205
206
207 --Q11.
208 WITH CrewFlightHours AS (
209     SELECT
210         c.CrewID,
211         CONCAT(c.FirstName, ' ', c.LastName) AS CrewMemberName,
```

# SOLUTIONS

So

```
212         c.Position,
213         COUNT(f.FlightID) AS TotalFlights,
214         SUM(DATEDIFF(HOUR, f.DepartureTime, f.ArrivalTime)) AS
215             TotalFlightHours
216     FROM
217         Air.Crew c
218     JOIN
219         Air.Flights f ON c.CrewID = f.CrewID
220     WHERE
221         f.DepartureTime BETWEEN '2022-01-01' AND '2022-12-31' -- Specify
222             the time period
223     GROUP BY
224         c.CrewID,
225         c.FirstName,
226         c.LastName,
227         c.Position
228 )
229 SELECT
230     CrewID,
231     CrewMemberName,
232     Position,
233     TotalFlights,
234     TotalFlightHours,
235     RANK() OVER (ORDER BY TotalFlightHours DESC) AS UtilisationRank
236 FROM
237     CrewFlightHours
238 ORDER BY
239     UtilisationRank;
240
241 --Q12.
242 WITH PassengerFlights AS (
243     SELECT
244         p.PassengerID,
245         CONCAT(p.FirstName, ' ', p.LastName) AS PassengerName,
246         COUNT(b.FlightID) AS TotalFlights,
247         SUM(t.Price) AS TotalAmountSpent
248     FROM
249         Air.Passengers p
250     JOIN
251         Air.Bookings b ON p.PassengerID = b.PassengerID
252     JOIN
253         Air.Tickets t ON b.BookingID = t.BookingID
254     WHERE
255         b.Status = 'Confirmed'
256         AND b.BookingDate BETWEEN '2021-01-01' AND '2021-12-31'
257     GROUP BY
258         p.PassengerID,
259         p.FirstName,
260         p.LastName
261 )
262 SELECT
263     PassengerID,
```

# SOLUTIONS

So

---

```
263     PassengerName,  
264     TotalFlights,  
265     TotalAmountSpent,  
266     RANK() OVER (ORDER BY TotalFlights DESC, TotalAmountSpent DESC) AS      ↵  
267     LoyaltyRank  
268   FROM  
269   PassengerFlights  
270 ORDER BY  
271     LoyaltyRank;  
272  
273 --Q13.  
274 WITH ClassRevenue AS (  
275     SELECT  
276         t.Class,  
277         SUM(t.Price) AS TotalRevenue,  
278         COUNT(t.TicketID) AS TicketsSold  
279     FROM  
280         Air.Tickets t  
281     JOIN  
282         Air.Bookings b ON t.BookingID = b.BookingID  
283     JOIN  
284         Air.Flights f ON b.FlightID = f.FlightID  
285     WHERE  
286         b.Status = 'Confirmed'    -- Only consider confirmed bookings  
287     GROUP BY  
288         t.Class  
289 )  
290     SELECT  
291     Class,  
292     TotalRevenue,  
293     TicketsSold,  
294     RANK() OVER (ORDER BY TotalRevenue DESC) AS RevenueRank  
295   FROM  
296     ClassRevenue  
297 ORDER BY  
298     RevenueRank;  
299  
300 --Q14.  
301 WITH PassengerDemographics AS (  
302     SELECT  
303         p.PassengerID,  
304         CONCAT(p.FirstName, ' ', p.LastName) AS PassengerName,  
305         p.Gender,  
306         DATEDIFF(YEAR, p.DateOfBirth, GETDATE()) AS Age,  
307         p.Email,  
308         p.PhoneNumber  
309     FROM  
310         Air.Passengers p  
311     JOIN  
312         Air.Bookings b ON p.PassengerID = b.PassengerID  
313     WHERE  
314         b.Status = 'Confirmed'    -- Only consider confirmed bookings
```

# SOLUTIONS

So

```
315      AND b.BookingDate BETWEEN '2021-01-01' AND '2021-12-31' -- Specify ↵
            the time period
316  )
317  SELECT
318      Gender,
319      COUNT(*) AS PassengerCount,
320      AVG(Age) AS AverageAge
321  FROM
322      PassengerDemographics
323  GROUP BY
324      Gender
325  ORDER BY
326      PassengerCount DESC;
327
328
329  --Q15.
330  WITH FlightRevenue AS (
331      SELECT
332          f.FlightID,
333          f.FlightNumber,
334          f.DepartureAirport,
335          f.ArrivalAirport,
336          f.DepartureTime,
337          f.ArrivalTime,
338          p.Capacity,
339          COUNT(t.TicketID) AS SeatsBooked,
340          SUM(t.Price) AS TotalRevenue,
341          CAST(COUNT(t.TicketID) AS FLOAT) / p.Capacity * 100 AS ↵
                SeatUtilizationPercentage
342  FROM
343      Air.Flights f
344  JOIN
345      Air.Bookings b ON f.FlightID = b.FlightID
346  JOIN
347      Air.Tickets t ON b.BookingID = t.BookingID
348  JOIN
349      Air.Planes p ON f.PlaneID = p.PlaneID
350  WHERE
351      b.Status = 'Confirmed' -- Only consider confirmed bookings
352  GROUP BY
353      f.FlightID,
354      f.FlightNumber,
355      f.DepartureAirport,
356      f.ArrivalAirport,
357      f.DepartureTime,
358      f.ArrivalTime,
359      p.Capacity
360  )
361  SELECT
362      FlightID,
363      FlightNumber,
364      DepartureAirport,
365      ArrivalAirport,
```

# SOLUTIONS

So

---

```
366     DepartureTime,
367     ArrivalTime,
368     Capacity,
369     SeatsBooked,
370     TotalRevenue,
371     SeatUtilizationPercentage
372 FROM
373     FlightRevenue
374 ORDER BY
375     DepartureTime;
376
377 --Q16.
378 WITH FrequentFlyer AS (
379     SELECT
380         p.PassengerID,
381         CONCAT(p.FirstName, ' ', p.LastName) AS PassengerName,
382         p.Email,
383         p.PhoneNumber,
384         COUNT(b.FlightID) AS TotalFlights
385     FROM
386         Air.Passengers p
387     JOIN
388         Air.Bookings b ON p.PassengerID = b.PassengerID
389     WHERE
390         b.Status = 'Confirmed' -- Only consider confirmed bookings
391         AND b.BookingDate BETWEEN '2021-01-01' AND '2022-12-31' -- Specify ↗
392             the time period
393     GROUP BY
394         p.PassengerID,
395         p.FirstName,
396         p.LastName,
397         p.Email,
398         p.PhoneNumber
399     )
400     SELECT
401         PassengerID,
402         PassengerName,
403         Email,
404         PhoneNumber,
405         TotalFlights,
406         CASE
407             WHEN TotalFlights >= 10 THEN 'Platinum'
408             WHEN TotalFlights >= 5 THEN 'Gold'
409             WHEN TotalFlights >= 2 THEN 'Silver'
410             ELSE 'Bronze'
411         END AS FrequentFlyerTier
412     FROM
413         FrequentFlyer
414     ORDER BY
415         TotalFlights DESC;
416
417 --Q17.
```

# SOLUTIONS

So

---

```
418 SELECT
419     f.DepartureAirport,
420     YEAR(f.DepartureTime) AS Year,
421     MONTH(f.DepartureTime) AS Month,
422     COUNT(f.FlightID) AS NumberOfFlights,
423     SUM(t.Price) AS TotalRevenue
424 FROM
425     Air.Flights f
426 JOIN
427     Air.Bookings b ON f.FlightID = b.FlightID
428 JOIN
429     Air.Tickets t ON b.BookingID = t.BookingID
430 WHERE
431     b.Status = 'Confirmed' -- Only consider confirmed bookings
432 GROUP BY
433     f.DepartureAirport,
434     YEAR(f.DepartureTime),
435     MONTH(f.DepartureTime)
436 ORDER BY
437     f.DepartureAirport,
438     Year,
439     Month;
440
441 --Q18.
442 SELECT
443     YEAR(b.BookingDate) AS Year,
444     MONTH(b.BookingDate) AS Month,
445     COUNT(b.BookingID) AS NumberOfBookings,
446     AVG(DATEDIFF(DAY, b.BookingDate, f.DepartureTime)) AS AverageLeadTime
447 FROM
448     Air.Bookings b
449 JOIN
450     Air.Flights f ON b.FlightID = f.FlightID
451 WHERE
452     b.Status = 'Confirmed' -- Only consider confirmed bookings
453 GROUP BY
454     YEAR(b.BookingDate),
455     MONTH(b.BookingDate)
456 ORDER BY
457     Year,
458     Month;
459
460
461 --Q18-1
462
463 SELECT
464     YEAR(b.BookingDate) AS BookingDate,
465     COUNT(b.BookingID) AS NumberOfBookings,
466     AVG(DATEDIFF(DAY, b.BookingDate, f.DepartureTime)) AS AverageLeadTime
467 FROM
468     Air.Bookings b
469 JOIN
470     Air.Flights f ON b.FlightID = f.FlightID
```

# SOLUTIONS

---

```
471 WHERE
472     b.Status = 'Confirmed'
473 GROUP BY
474     YEAR(b.BookingDate)
475 ORDER BY
476     BookingDate;
477
478
479 --Q18-2
480 SELECT
481     t.Class,
482     YEAR(b.BookingDate) AS Year,
483     MONTH(b.BookingDate) AS Month,
484     COUNT(b.BookingID) AS NumberOfBookings,
485     AVG(DATEDIFF(DAY, b.BookingDate, f.DepartureTime)) AS AverageLeadTime
486 FROM
487     Air.Bookings b
488 JOIN
489     Air.Flights f ON b.FlightID = f.FlightID
490 JOIN
491     Air.Tickets t ON b.BookingID = t.BookingID
492 WHERE
493     b.Status = 'Confirmed'
494 GROUP BY
495     t.Class,
496     YEAR(b.BookingDate),
497     MONTH(b.BookingDate)
498 ORDER BY
499     t.Class,
500     Year,
501     Month;
502
503 --Q19.
504 SELECT
505     f.FlightID,
506     f.FlightNumber,
507     f.DepartureAirport,
508     f.ArrivalAirport,
509     f.DepartureTime,
510     f.ArrivalTime,
511     COUNT(t.TicketID) AS TotalTickets,
512     SUM(t.Price) AS TotalRevenue,
513     AVG(DATEDIFF(MINUTE, f.DepartureTime, f.ArrivalTime)) AS
514         AverageDelayMinutes,
515     SUM(CASE WHEN DATEDIFF(MINUTE, f.DepartureTime, f.ArrivalTime) > 0 THEN
516             t.Price ELSE 0 END) AS RevenueLostDueToDelays
517 FROM
518     Air.Flights f
519 JOIN
520     Air.Bookings b ON f.FlightID = b.FlightID
521 JOIN
522     Air.Tickets t ON b.BookingID = t.BookingID
523 WHERE
```

# SOLUTIONS

---

```
522     b.Status = 'Confirmed' -- Only consider confirmed bookings
523 GROUP BY
524     f.FlightID,
525     f.FlightNumber,
526     f.DepartureAirport,
527     f.ArrivalAirport,
528     f.DepartureTime,
529     f.ArrivalTime
530 ORDER BY
531     AverageDelayMinutes DESC;
532
533
534 --Q19-1
535 SELECT
536     YEAR(f.DepartureTime) AS Year,
537     MONTH(f.DepartureTime) AS Month,
538     COUNT(f.FlightID) AS NumberOfDelayedFlights,
539     AVG(DATEDIFF(MINUTE, f.DepartureTime, f.ArrivalTime)) AS
540         AverageDelayMinutes,
541     SUM(CASE WHEN DATEDIFF(MINUTE, f.DepartureTime, f.ArrivalTime) > 0 THEN ↵
542             t.Price ELSE 0 END) AS RevenueLostDueToDelays
543 FROM
544     Air.Flights f
545 JOIN
546     Air.Bookings b ON f.FlightID = b.FlightID
547 JOIN
548     Air.Tickets t ON b.BookingID = t.BookingID
549 WHERE
550     b.Status = 'Confirmed'
551     GROUP BY
552         YEAR(f.DepartureTime),
553         MONTH(f.DepartureTime)
554 ORDER BY
555     Year,
556     Month;
557
558
559 --Q19-2
560
561 SELECT
562     f.DepartureAirport,
563     COUNT(f.FlightID) AS NumberOfDelayedFlights,
564     AVG(DATEDIFF(MINUTE, f.DepartureTime, f.ArrivalTime)) AS
565         AverageDelayMinutes,
566     SUM(CASE WHEN DATEDIFF(MINUTE, f.DepartureTime, f.ArrivalTime) > 0 THEN ↵
567             t.Price ELSE 0 END) AS RevenueLostDueToDelays
568 FROM
569     Air.Flights f
570 JOIN
571     Air.Bookings b ON f.FlightID = b.FlightID
572 JOIN
573     Air.Tickets t ON b.BookingID = t.BookingID
574 WHERE
```

# SOLUTIONS

---

```
571     b.Status = 'Confirmed'  
572 GROUP BY  
573     f.DepartureAirport  
574 ORDER BY  
575     AverageDelayMinutes DESC;  
576  
577  
578 --Q20.  
579 SELECT  
580     f.FlightID,  
581     f.FlightNumber,  
582     f.DepartureAirport,  
583     f.ArrivalAirport,  
584     f.DepartureTime,  
585     f.ArrivalTime,  
586     SUM(t.Price) AS TotalRevenue,  
587     COUNT(t.TicketID) AS TotalTickets  
588 FROM  
589     Air.Flights f  
590 JOIN  
591     Air.Bookings b ON f.FlightID = b.FlightID  
592 JOIN  
593     Air.Tickets t ON b.BookingID = t.BookingID  
594 WHERE  
595     b.Status = 'Confirmed' -- Only consider confirmed bookings  
596 GROUP BY  
597     f.FlightID,  
598     f.FlightNumber,  
599     f.DepartureAirport,  
600     f.ArrivalAirport,  
601     f.DepartureTime,  
602     f.ArrivalTime  
603 ORDER BY  
604     TotalRevenue DESC;  
605  
606 --Q20-1.  
607 SELECT  
608     YEAR(f.DepartureTime) AS Year,  
609     MONTH(f.DepartureTime) AS Month,  
610     f.FlightID,  
611     f.FlightNumber,  
612     f.DepartureAirport,  
613     f.ArrivalAirport,  
614     SUM(t.Price) AS TotalRevenue,  
615     COUNT(t.TicketID) AS TotalTickets  
616 FROM  
617     Air.Flights f  
618 JOIN  
619     Air.Bookings b ON f.FlightID = b.FlightID  
620 JOIN  
621     Air.Tickets t ON b.BookingID = t.BookingID  
622 WHERE  
623     b.Status = 'Confirmed'
```

# SOLUTIONS

---

```
624 GROUP BY
625     YEAR(f.DepartureTime),
626     MONTH(f.DepartureTime),
627     f.FlightID,
628     f.FlightNumber,
629     f.DepartureAirport,
630     f.ArrivalAirport
631 ORDER BY
632     Year,
633     Month,
634     TotalRevenue DESC;
635
636 --Q20-2
637 SELECT
638     f.DepartureAirport,
639     f.ArrivalAirport,
640     f.FlightID,
641     f.FlightNumber,
642     SUM(t.Price) AS TotalRevenue,
643     COUNT(t.TicketID) AS TotalTickets
644 FROM
645     Air.Flights f
646 JOIN
647     Air.Bookings b ON f.FlightID = b.FlightID
648 JOIN
649     Air.Tickets t ON b.BookingID = t.BookingID
650 WHERE
651     b.Status = 'Confirmed'
652 GROUP BY
653     f.DepartureAirport,
654     f.ArrivalAirport,
655     f.FlightID,
656     f.FlightNumber
657 ORDER BY
658     TotalRevenue DESC;
659
660
661 --Q21.
662 SELECT
663     p.PassengerID,
664     CONCAT(p.FirstName, ' ', p.LastName) AS PassengerName,
665     p.Email,
666     COUNT(b.FlightID) AS TotalFlights,
667     SUM(t.Price) AS TotalSpend
668 FROM
669     Air.Passengers p
670 JOIN
671     Air.Bookings b ON p.PassengerID = b.PassengerID
672 JOIN
673     Air.Tickets t ON b.BookingID = t.BookingID
674 WHERE
675     b.Status = 'Confirmed' -- Only consider confirmed bookings
676 GROUP BY
```

# SOLUTIONS

---

```
677     p.PassengerID,
678     p.FirstName,
679     p.LastName,
680     p.Email
681 ORDER BY
682     TotalFlights DESC,
683     TotalSpend DESC;
684
685
686 --Q22.
687 SELECT
688     p.PlaneID,
689     p.Model,
690     p.Manufacturer,
691     COUNT(f.FlightID) AS TotalFlights,
692     SUM(DATEDIFF(HOUR, f.DepartureTime, f.ArrivalTime)) AS
693     TotalFlightHours,
694     AVG(DATEDIFF(HOUR, f.DepartureTime, f.ArrivalTime)) AS
695     AverageFlightDuration,
696     CASE
697         WHEN COUNT(f.FlightID) < 0 THEN 'Under-utilised'
698         WHEN COUNT(f.FlightID) > 3 THEN 'Over-utilised'
699         ELSE 'Well-utilised'
700     END AS UtilizationStatus
701 FROM
702     Air.Planes p
703 LEFT JOIN
704     Air.Flights f ON p.PlaneID = f.PlaneID
705 GROUP BY
706     p.PlaneID,
707     p.Model,
708     p.Manufacturer
709 ORDER BY
710     TotalFlights DESC;
711
712 --Q22-1
713 SELECT
714     YEAR(f.DepartureTime) AS Year,
715     MONTH(f.DepartureTime) AS Month,
716     p.PlaneID,
717     p.Model,
718     COUNT(f.FlightID) AS TotalFlights,
719     SUM(DATEDIFF(HOUR, f.DepartureTime, f.ArrivalTime)) AS TotalFlightHours
720 FROM
721     Air.Planes p
722 JOIN
723     Air.Flights f ON p.PlaneID = f.PlaneID
724 GROUP BY
725     YEAR(f.DepartureTime),
726     MONTH(f.DepartureTime),
727     p.PlaneID,
728     p.Model
```

# SOLUTIONS

---

```
728 ORDER BY
729     Year,
730     Month,
731     TotalFlights DESC;
732
733
734 --Q22-2
735 SELECT
736     TOP 5
737     p.PlaneID,
738     p.Model,
739     p.Manufacturer,
740     COUNT(f.FlightID) AS TotalFlights,
741     SUM(DATEDIFF(HOUR, f.DepartureTime, f.ArrivalTime)) AS TotalFlightHours
742 FROM
743     Air.Planes p
744 JOIN
745     Air.Flights f ON p.PlaneID = f.PlaneID
746 GROUP BY
747     p.PlaneID,
748     p.Model,
749     p.Manufacturer
750 ORDER BY
751     TotalFlights DESC;
752
753
754 --Q22-3
755 SELECT
756     p.PlaneID,
757     p.Model,
758     p.Manufacturer,
759     COUNT(f.FlightID) AS TotalFlights,
760     SUM(DATEDIFF(HOUR, f.DepartureTime, f.ArrivalTime)) AS TotalFlightHours
761 FROM
762     Air.Planes p
763 LEFT JOIN
764     Air.Flights f ON p.PlaneID = f.PlaneID
765 GROUP BY
766     p.PlaneID,
767     p.Model,
768     p.Manufacturer
769 HAVING
770     COUNT(f.FlightID) < 1
771 ORDER BY
772     TotalFlights ASC;
773
774
775 --Q23.
776 -- Gender Distribution of Passengers
777 SELECT
778     p.Gender,
779     COUNT(p.PassengerID) AS PassengerCount
780 FROM
```

# SOLUTIONS

---

```
781     Air.Passengers p
782 JOIN
783     Air.Bookings b ON p.PassengerID = b.PassengerID
784 WHERE
785     b.Status = 'Confirmed'
786 GROUP BY
787     p.Gender
788 ORDER BY
789     PassengerCount DESC;
790
791
792 --Q24.
793
794 -- Age Distribution of Passengers
795 SELECT
796     CASE
797         WHEN DATEDIFF(YEAR, p.DateOfBirth, GETDATE()) BETWEEN 0 AND 17 THEN '0-17'
798         WHEN DATEDIFF(YEAR, p.DateOfBirth, GETDATE()) BETWEEN 18 AND 24 THEN '18-24'
799         WHEN DATEDIFF(YEAR, p.DateOfBirth, GETDATE()) BETWEEN 25 AND 34 THEN '25-34'
800         WHEN DATEDIFF(YEAR, p.DateOfBirth, GETDATE()) BETWEEN 35 AND 44 THEN '35-44'
801         WHEN DATEDIFF(YEAR, p.DateOfBirth, GETDATE()) BETWEEN 45 AND 54 THEN '45-54'
802         WHEN DATEDIFF(YEAR, p.DateOfBirth, GETDATE()) BETWEEN 55 AND 64 THEN '55-64'
803         ELSE '65+'
804     END AS AgeGroup,
805     COUNT(p.PassengerID) AS PassengerCount
806 FROM
807     Air.Passengers p
808 JOIN
809     Air.Bookings b ON p.PassengerID = b.PassengerID
810 WHERE
811     b.Status = 'Confirmed'
812 GROUP BY
813     CASE
814         WHEN DATEDIFF(YEAR, p.DateOfBirth, GETDATE()) BETWEEN 0 AND 17 THEN '0-17'
815         WHEN DATEDIFF(YEAR, p.DateOfBirth, GETDATE()) BETWEEN 18 AND 24 THEN '18-24'
816         WHEN DATEDIFF(YEAR, p.DateOfBirth, GETDATE()) BETWEEN 25 AND 34 THEN '25-34'
817         WHEN DATEDIFF(YEAR, p.DateOfBirth, GETDATE()) BETWEEN 35 AND 44 THEN '35-44'
818         WHEN DATEDIFF(YEAR, p.DateOfBirth, GETDATE()) BETWEEN 45 AND 54 THEN '45-54'
819         WHEN DATEDIFF(YEAR, p.DateOfBirth, GETDATE()) BETWEEN 55 AND 64 THEN '55-64'
820         ELSE '65+'
821     END
```

# SOLUTIONS

---

```
822 ORDER BY
823     PassengerCount DESC;
824
825
826 --Q25.
827 -- Most Popular Travel Routes
828 SELECT
829     f.DepartureAirport,
830     f.ArrivalAirport,
831     COUNT(f.FlightID) AS FlightCount
832 FROM
833     Air.Flights f
834 JOIN
835     Air.Bookings b ON f.FlightID = b.FlightID
836 WHERE
837     b.Status = 'Confirmed'
838 GROUP BY
839     f.DepartureAirport,
840     f.ArrivalAirport
841 ORDER BY
842     FlightCount DESC;
843
844
845 --Q26.
846 -- Booking Status Analysis by Month
847 SELECT
848     YEAR(BookingDate) AS Year,
849     MONTH(BookingDate) AS Month,
850     COUNT(BookingID) AS TotalBookings,
851     SUM(CASE WHEN Status = 'Confirmed' THEN 1 ELSE 0 END) AS ↗
852         ConfirmedBookings,
853     SUM(CASE WHEN Status = 'Canceled' THEN 1 ELSE 0 END) AS ↗
854         CanceledBookings,
855     SUM(CASE WHEN Status = 'Pending' THEN 1 ELSE 0 END) AS PendingBookings
856 FROM
857     Air.Bookings
858 GROUP BY
859     YEAR(BookingDate),
860     MONTH(BookingDate)
861 ORDER BY
862     Year,
863     Month;
864
865
866 --Q27.
867 SELECT
868     CAST(BookingDate AS DATE) AS BookingDay,
869     COUNT(BookingID) AS TotalBookings,
870     SUM(CASE WHEN Status = 'Confirmed' THEN 1 ELSE 0 END) AS ↗
871         ConfirmedBookings,
872     SUM(CASE WHEN Status = 'Canceled' THEN 1 ELSE 0 END) AS ↗
873         CanceledBookings,
```

# SOLUTIONS

```
871     SUM(CASE WHEN Status = 'Pending' THEN 1 ELSE 0 END) AS PendingBookings
872 FROM
873     Air.Bookings
874 GROUP BY
875     CAST(BookingDate AS DATE)
876 ORDER BY
877     BookingDay;
878
879
880 --Q28.
881 SELECT
882     Status,
883     COUNT(BookingID) AS TotalBookings
884 FROM
885     Air.Bookings
886 GROUP BY
887     Status
888 ORDER BY
889     TotalBookings DESC;
890
891
892 --Q29.
893 SELECT
894     YEAR(BookingDate) AS Year,
895     DATEPART(WEEK, BookingDate) AS Week,
896     COUNT(BookingID) AS TotalBookings,
897     SUM(CASE WHEN Status = 'Confirmed' THEN 1 ELSE 0 END) AS ConfirmedBookings,
898     SUM(CASE WHEN Status = 'Canceled' THEN 1 ELSE 0 END) AS CanceledBookings,
899     SUM(CASE WHEN Status = 'Pending' THEN 1 ELSE 0 END) AS PendingBookings
900 FROM
901     Air.Bookings
902 GROUP BY
903     YEAR(BookingDate),
904     DATEPART(WEEK, BookingDate)
905 ORDER BY
906     Year,
907     Week;
908
909
910 --Q30.
911 -- Monthly Booking Volume Analysis
912 SELECT
913     YEAR(BookingDate) AS Year,
914     MONTH(BookingDate) AS Month,
915     COUNT(BookingID) AS TotalBookings
916 FROM
917     Air.Bookings
918 GROUP BY
919     YEAR(BookingDate),
920     MONTH(BookingDate)
921 ORDER BY
```

# SOLUTIONS

---

```
922     Year,  
923     Month;  
924  
925  
926 --Q31.  
927 SELECT  
928     CAST(BookingDate AS DATE) AS BookingDay,  
929     COUNT(BookingID) AS TotalBookings  
930 FROM  
931     Air.Bookings  
932 GROUP BY  
933     CAST(BookingDate AS DATE)  
934 ORDER BY  
935     BookingDay;  
936  
937  
938 --Q32.  
939 SELECT  
940     YEAR(BookingDate) AS Year,  
941     DATEPART(WEEK, BookingDate) AS Week,  
942     COUNT(BookingID) AS TotalBookings  
943 FROM  
944     Air.Bookings  
945 GROUP BY  
946     YEAR(BookingDate),  
947     DATEPART(WEEK, BookingDate)  
948 ORDER BY  
949     Year,  
950     Week;  
951  
952  
953 --Q33.  
954 SELECT  
955     YEAR(BookingDate) AS Year,  
956     COUNT(BookingID) AS TotalBookings  
957 FROM  
958     Air.Bookings  
959 GROUP BY  
960     YEAR(BookingDate)  
961 ORDER BY  
962     Year;  
963  
964  
965 --Q34.  
966 -- Fleet Composition by Manufacturer  
967 SELECT  
968     Manufacturer,  
969     COUNT(PlaneID) AS TotalPlanes,  
970     SUM(Capacity) AS TotalCapacity  
971 FROM  
972     Air.Planes  
973 GROUP BY  
974     Manufacturer
```

# SOLUTIONS

---

```
975 ORDER BY
976     TotalPlanes DESC;
977
978
979 --Q35.
980 SELECT
981     Status,
982     COUNT(PlaneID) AS TotalPlanes
983 FROM
984     Air.Planes
985 GROUP BY
986     Status
987 ORDER BY
988     TotalPlanes DESC;
989
990
991 --Q36.
992 SELECT
993     PlaneID,
994     Model,
995     Capacity,
996     YearOfManufacture,
997     Status
998 FROM
999     Air.Planes
1000 WHERE
1001     Manufacturer = 'Boeing' -- Replace with the actual manufacturer name
1002 ORDER BY
1003     YearOfManufacture DESC;
1004
1005
1006 --Q37.
1007 SELECT
1008     CASE
1009         WHEN Capacity <= 100 THEN '0-100 seats'
1010         WHEN Capacity BETWEEN 101 AND 200 THEN '101-200 seats'
1011         WHEN Capacity BETWEEN 201 AND 300 THEN '201-300 seats'
1012         ELSE '300+ seats'
1013     END AS CapacityGroup,
1014     COUNT(PlaneID) AS TotalPlanes
1015 FROM
1016     Air.Planes
1017 GROUP BY
1018     CASE
1019         WHEN Capacity <= 100 THEN '0-100 seats'
1020         WHEN Capacity BETWEEN 101 AND 200 THEN '101-200 seats'
1021         WHEN Capacity BETWEEN 201 AND 300 THEN '201-300 seats'
1022         ELSE '300+ seats'
1023     END
1024 ORDER BY
1025     TotalPlanes DESC;
1026
1027
```

# SOLUTIONS

---

```
1028 --Q38.
1029 SELECT
1030     CASE
1031         WHEN YEAR(GETDATE()) - YEAR(YearOfManufacture) <= 5 THEN '0-5'      ↵
1032             'years'
1033         WHEN YEAR(GETDATE()) - YEAR(YearOfManufacture) BETWEEN 6 AND 10      ↵
1034             '6-10 years'
1035         WHEN YEAR(GETDATE()) - YEAR(YearOfManufacture) BETWEEN 11 AND 20      ↵
1036             '11-20 years'
1037         ELSE '20+ years'
1038     END AS AgeGroup,
1039     COUNT(PlaneID) AS TotalPlanes
1040 FROM
1041     Air.Planes
1042 GROUP BY
1043     CASE
1044         WHEN YEAR(GETDATE()) - YEAR(YearOfManufacture) <= 5 THEN '0-5'      ↵
1045             'years'
1046         WHEN YEAR(GETDATE()) - YEAR(YearOfManufacture) BETWEEN 6 AND 10      ↵
1047             '6-10 years'
1048         WHEN YEAR(GETDATE()) - YEAR(YearOfManufacture) BETWEEN 11 AND 20      ↵
1049             '11-20 years'
1050         ELSE '20+ years'
1051     END
1052 ORDER BY
1053     TotalPlanes DESC;
1054
1055 --Q39.
1056 -- Revenue Analysis by Ticket Class
1057 SELECT
1058     Class,
1059     COUNT(TicketID) AS TotalTicketsSold,
1060     SUM(Price) AS TotalRevenue,
1061     AVG(Price) AS AverageTicketPrice
1062 FROM
1063     Air.Tickets
1064 GROUP BY
1065     Class
1066 ORDER BY
1067     TotalRevenue DESC;
1068
1069 --Q40.
1070 SELECT
1071     Class,
1072     YEAR(BookingDate) AS Year,
1073     MONTH(BookingDate) AS Month,
1074     COUNT(TicketID) AS TotalTicketsSold,
1075     SUM(Price) AS TotalRevenue,
1076     AVG(Price) AS AverageTicketPrice
1077 FROM
1078     Air.Tickets t
1079 JOIN
```

# SOLUTIONS

---

```
1075     Air.Bookings b ON t.BookingID = b.BookingID
1076 GROUP BY
1077     Class,
1078     YEAR(BookingDate),
1079     MONTH(BookingDate)
1080 ORDER BY
1081     Year,
1082     Month,
1083     TotalRevenue DESC;
1084
1085
1086 --Q41.
1087 SELECT
1088     TOP 5
1089     TicketID,
1090     Class,
1091     Price
1092 FROM
1093     Air.Tickets
1094 ORDER BY
1095     Price DESC;
1096
1097
1098 --Q42.
1099 SELECT
1100     CASE
1101         WHEN Price <= 100 THEN '0-100'
1102         WHEN Price BETWEEN 101 AND 200 THEN '101-200'
1103         WHEN Price BETWEEN 201 AND 300 THEN '201-300'
1104         WHEN Price BETWEEN 301 AND 400 THEN '301-400'
1105         ELSE '400+'
1106     END AS PriceRange,
1107     COUNT(TicketID) AS TotalTicketsSold,
1108     SUM(Price) AS TotalRevenue
1109 FROM
1110     Air.Tickets
1111 GROUP BY
1112     CASE
1113         WHEN Price <= 100 THEN '0-100'
1114         WHEN Price BETWEEN 101 AND 200 THEN '101-200'
1115         WHEN Price BETWEEN 201 AND 300 THEN '201-300'
1116         WHEN Price BETWEEN 301 AND 400 THEN '301-400'
1117         ELSE '400+'
1118     END
1119 ORDER BY
1120     TotalRevenue DESC;
1121
1122
1123
1124
```

# Appendix

## SQL Server Installation Guide

**Phase 1:** Installing SQL Server 2019 on a Windows 10 operating system.

1. To set up SQL Server 2019, obtain the necessary files by clicking the provided link:  
<https://www.microsoft.com/en-gb/sql-server/sql-server-downloads>
2. Select "Download Now" for the developer edition.



Or, download a free specialised edition



3. After the file has finished downloading, double-click on it to initiate the installation.
4. In the window that appears, choose the "Basic" installation type.

# Developer Edition

Select an installation type:

## Basic

Select Basic installation type to install the SQL Server Database Engine feature with default configuration.

## Custom

Select Custom installation type to step through the SQL Server installation wizard and choose what you want to install. This installation type is detailed and takes longer than running the Basic install.

## Download Media

Download SQL Server setup files now and install them later on a machine of your choice.

SQL Server transmits information about your installation experience, as well as other usage and performance data, to Microsoft to help improve the product. To learn more about data processing and privacy controls, and to turn off the collection of this information after installation, see the [documentation](#)

5. Press "Next," agree to the Terms and Conditions, and then click "Install."
6. After the installation is finished, you will receive a link to download SQL Server Management Studio. If you don't spot the link, please click on this provided link:  
<https://aka.ms/ssmsfullsetup>
7. Download SQL Server Management Studio and proceed to install it.

## Phase 2: AdventureWorks (2019 or 2022) Database

1. Upon the successful installation of SQL Server 2019, you'll require a database for practice.  
Please follow the link below to download the AdventureWorks2019 or 2022 Database:  
<https://github.com/Microsoft/sql-server-samples/releases/tag/adventureworks>
2. On the webpage, locate and select the highlighted option to download the AdventureWorks2019.bak file or AdventureWorks2022.bak file.

## AdventureWorks (OLTP) full database backups

[AdventureWorks2022.bak](#)

[AdventureWorks2019.bak](#)

[AdventureWorks2017.bak](#)

[AdventureWorks2016.bak](#)

[AdventureWorks2016\\_EXT.bak](#)

Download size is 883 MB. This is an extended version of AdventureWorks, Server 2016 sample scripts on this database.

[AdventureWorks2014.bak](#)

[AdventureWorks2012.bak](#)

3. Navigate to the folder where the AdventureWorks2012.bak file has been downloaded and proceed to make a copy of the file.
4. Paste the file into the Backup folder within your freshly installed SQL system, which should be situated in a location resembling the one described below:

**C:\Program Files\Microsoft SQL Server\MSSQL11.SQLSERVERB\MSSQL\Backup**

5. Next, open SQL Server Management Studio from either the Programs Menu or the Applications Desktop (Windows 8).
6. Now click on the following link to restore the database on SQL Server:

<https://learn.microsoft.com/en-us/sql/samples/adventureworks-install-configure?view=sql-server-ver16&tabs=ssms>

### Phase 3: AdventureWorks Data Warehouse Version (2019 or 2022) Database

1. The following link allows you to download the AdventureWorksDW2019.bak or AdventureWorksDW2022.bak versions: <https://learn.microsoft.com/en-us/sql/samples/adventureworks-install-configure?view=sql-server-ver16&tabs=ssms>
2. After downloading the file, following similar steps as in **Phase 2** to restore the database.



OLTP	Data Warehouse	Lightweight
AdventureWorks2022.bak ↗	AdventureWorksDW2022.bak ↗	AdventureWorksLT2(
AdventureWorks2019.bak ↗	AdventureWorksDW2019.bak ↗	AdventureWorksLT2(
AdventureWorks2017.bak ↗	AdventureWorksDW2017.bak ↗	AdventureWorksLT2(
AdventureWorks2016.bak ↗	AdventureWorksDW2016.bak ↗	AdventureWorksLT2(
AdventureWorks2016_EXT.bak ↗	AdventureWorksDW2016_EXT.bak ↗	N/A
AdventureWorks2014.bak ↗	AdventureWorksDW2014.bak ↗	AdventureWorksLT2(
AdventureWorks2012.bak ↗	AdventureWorksDW2012.bak ↗	AdventureWorksLT2(
AdventureWorks2008R2.bak ↗	AdventureWorksDW2008R2.bak ↗	N/A

### Phase 4: WideWorldImporters Database:

1. The link to download the WideWorldImporters database is: <https://github.com/Microsoft/sql-server-samples/releases/tag/wide-world-importers-v1.0>
2. Ensure to follow the steps as outlined in Phase 2 to restore the database.

▼ Assets 17

⌚Daily.ETL.ispac	61.2 KB	Aug 12, 2016
⌚sample-scripts.zip	23.1 KB	Jun 8, 2016
⌚WideWorldImporters-Full.bacpac	58.5 MB	Oct 7, 2022
⌚WideWorldImporters-Full.bak	121 MB	Oct 7, 2022
⌚WideWorldImporters-Full_old.bacpac	59.1 MB	Nov 16, 2016
⌚WideWorldImporters-Full_old.bak	121 MB	Aug 13, 2016
⌚WideWorldImporters-Standard.bacpac	58.2 MB	Oct 7, 2022
⌚WideWorldImporters-Standard.bak	121 MB	Oct 7, 2022
⌚WideWorldImporters-Standard_old.bacpac	58.5 MB	Jun 8, 2016
⌚WideWorldImporters-Standard_old.bak	121 MB	Aug 15, 2016
⌚WideWorldImportersDW-Full.bacpac	19.6 MB	Nov 16, 2016
⌚WideWorldImportersDW-Full.bak	47.7 MB	Jun 8, 2016
⌚WideWorldImportersDW-Standard.bacpac	21.4 MB	Jun 8, 2016
⌚WideWorldImportersDW-Standard.bak	51.4 MB	Jun 8, 2016

