FREE COPY

REAL BUSINESS SCENARIOS

250+

SQL TUTORIAL EXCERCISES AND SOLUTIONS

COVERS MULTIPLE DATABASES:

- FINANCE & BANKING
- HEALTHCARE
- MANUFACTURING
- WAREHOUSE
- HR EMPLOYEE
- AND MORE...

GRAB YOUR FREE COPY

Dr. Eyo Eyo

250+ SQL TUTORIAL EXCERCISES AND SOLUTIONS

Dr. Eyo Eyo

Website: eyowhite.com

Twitter: twitter.com/Eyowhite3

Table of Contents

About the Author

Dr. Eyo Eyo (PhD) is a University Lecturer and accomplished researcher in the fields of machine learning, data analysis, and engineering.

Eyo holds a Doctor of Philosophy (PhD) degree, which stands as a testament to his dedication to advancing knowledge in data science and engineering. His academic journey has been marked by a relentless pursuit of excellence, resulting in numerous scholarly achievements and contributions that have enriched his discipline.

Eyo's commitment to education goes beyond the classroom, as he continually seeks innovative ways to engage students and foster a deeper understanding of complex subjects.

In the realm of research, Eyo has made significant strides. His work in machine learning and data analysis has led to groundbreaking insights and practical applications, contributing to the advancement of knowledge through numerous published works.

As you delve into the pages of Eyo's book, "250+ SQL Tutorial Exercises and Solutions", you will discover the depth of his expertise and the invaluable insights he brings to the world of Structured Query Language and data analysis in general.

Important information

Welcome to "**250+ SQL Tutorial Exercises and Solutions**." 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.

NOTE: The commands employed in the "sample" solutions primarily align with 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.

Various databases have been utilised in this book covering diverse sectors of business operations such as finance, banking, healthcare, manufacturing, warehouse, HR employee, and more. Access to these databases can be acquired as follows:

- ***Eyowhite database*** (HR Employee and ProductOrders Tables) – to obtain the ".bak" file for this database please use this link.

- ***Finance and Banking database*** – the ".bak" file for this database can be obtained using this link.
- ***Healthcare database*** – the ".bak" file for this database can be obtained using this link.
- ***AdventureWorks2019 or 2022, AdventureWorksDW2022 and WideWorldImporters databases***. Each of these databases can be obtained from Microsoft website as outlined in the instructions given in Appendix B of this book.

SQL Server installation and database restoration

In Appendix B, you'll find comprehensive instructions for installing SQL Server 2019 on Windows 10, as well as the step-by-step process for restoring the databases mentioned earlier, using the AdventureWorks databases as a reference. Additional documentation for the AdventureWorks databases, including the database dictionary and table relationships, can also be accessed on the Microsoft website or datedo.com.

Appendix A provides the tables and their relationships for the rest of the databases.

Who This Book Is For

This "**250+ SQL Tutorial Exercises and Solutions**" assumes some fundamental knowledge of SQL at the least. Nevertheless, it is intended for a wide audience, including:

- **SQL Beginners**: If you're new to SQL, don't worry. We'll start from the basics and gradually progress to more advanced topics. You'll find step-by-step exercises and solutions that build your SQL knowledge systematically.

- **Students and Educators**: Whether you're a student studying database management or an educator looking for resources to teach SQL, this book provides a rich set of exercises that can be used for learning or teaching purposes.
- **Database Professionals**: Experienced database administrators, analysts, and developers can benefit from this book by using it as a reference and for honing their SQL skills with challenging exercises.

What You'll Find Inside

In "**250+ SQL Tutorial Exercises and Solutions**," you'll discover:

- A structured approach: The book is organised into parts (Parts 1 – 3) starting with the fundamentals and progressing through intermediate and advanced topics.
- Part 1 (Basic or Entry-Level exercises) deals with fundamental concepts in the Data Query language (DQL) category. This is particularly useful for beginners in SQL. Syntax for the most common SQL statements, clauses, operators, etc are first presented and exercises given for business scenarios that do explore these concepts.
- Part 2 and part 3 deal with Intermediate and Advanced Level business scenario exercises respectively. Experienced SQL practitioners may choose to proceed directly to Parts 2 and 3 of the books, unless they find it beneficial to review their skills, in which case they can start with Part 1.
- Detailed solutions: After attempting the exercises, you can refer to the “sample” solutions provided at the end of the book to check your answers and learn from best practices. The provided sample solutions acknowledge that there are multiple

approaches to addressing and resolving SQL challenges, which is why they are presented as sample solutions.

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:

- Twitter: https://twitter.com/Eyowhite3
- Website: https://eyowhite.com/contact/

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

Thank you for choosing "250+ SQL Tutorial Exercises and Solutions." We hope this book serves as a valuable resource in your journey to become a proficient SQL practitioner.

Happy querying!

[Eyo Eyo]

PART 1

Basic (Entry-Level) Exercises

(Single-table queries)

Database: Eyowhite

The SELECT Statement

Basic syntax

```
SELECT column1, column2, ...

FROM table_name
```

Special Cases:

-- Using DISTINCT to retrieve unique values:

```
SELECT DISTINCT column1, column2, ...

FROM table_name
```

-- Using ALL to include all values:

```
SELECT ALL column1, column2, ...

FROM table_name
```

-- Retrieving all columns from a specific table:

```
SELECT *

FROM table_name
```

1. Business scenario Q1

Product Cost Distribution Analysis

You work for a retail company that sells various products. Your manager wants to analyse the cost distribution of products in the inventory. Using the CustomerProductTabl table, **r**etrieve the details of all products. CLICK TO SHOW ANSWER Q1

2. Business scenario Q2

Product Quantity Analysis I

You are working for a company that sources products from various suppliers. Your task is to analyse the quantities of the orders made. Retrieve a list of all unique product orders. Use the CustomerOrdersTabl table. CLICK TO SHOW ANSWER Q2

3. Business scenario Q3

Product Quantity Analysis II

You are working for a company that sources products from various suppliers. Your task is to analyse the quantities of the orders made. Retrieve a list of all product orders and display only their identity. Use the CustomerOrdersTabl table. CLICK TO SHOW ANSWER Q3.

The WHERE clause

Basic syntax

```
SELECT column1, column2, ...

FROM table_name

WHERE condition1

AND/OR condition2

AND/OR (condition3 AND condition4)

AND/OR NOT condition5

AND/OR (condition6 OR condition7)
```

4. Business Scenario Q4

Price Range Analysis

You work for an online marketplace that offers a wide range of products. Your task is to analyse products within specific price ranges to assist customers in finding products that match their budget. Using the CustomerProductTabl table,

a. Retrieve a list of products that are affordable for budget-conscious customers (cost under $1per product).

b. Find products that fall within a moderate price range (cost between $2 and $5 per product).

c. Identify high-end products that cater to customers looking for premium items (cost over $6 per product). CLICK TO SHOW ANSWER Q4

5. Business Scenario Q5

Custom Product Selection

You're tasked with selecting specific products for a custom order based on a list of desired product IDs. Using the CustomerProductTabl Table, retrieve the details of products with specific product IDs (2, 567, and 89) to fulfil a custom order. CLICK TO SHOW ANSWER Q5

6. Business Scenario Q6

Product Search by exact Name

You're working for an e-commerce company, and you want to find products with a specific name to assist a customer who is looking for those items. Your task is to retrieve product details for products that match the specific name provided by the customer. Using the CustomerProductTabl table, **r**etrieve the details of products with a specific name, such as "Instant Camera" or "Phone Stand". CLICK TO SHOW ANSWER Q6

7. Business scenario Q7

Items orders

You manage inventory for an online store and want to identify products that have been ordered by customers. Your superior already has a knowledge of 5 items of a specific product that were ordered by customers on a certain date. But you are now tasked with retrieving the product details for the quantity of items that were ordered which in this case likely had other purchased dates using the CustomerOrdersTabl Table. CLICK TO SHOW ANSWER Q7

8. Business scenario Q8

Product Search by Description

You're working for an online store, and you want to assist customers in finding products that match certain descriptions. Your task is to retrieve product details for items that have a specific keyword in their product description. Using the CustomerProductTabl table, retrieve the details of products that have "Set" as part of their product description. CLICK TO SHOW ANSWER Q8

9. Business scenario Q9

Order Tracking I

You're managing an employee database for an online retail store, and you want to identify the employees whose salary have not entered on the payroll. Using the Employee.Pay table, retrieve the JobTitle, and IDs for those whose names have not appeared on the payroll. CLICK TO SHOW ANSWER Q9

10. Business scenario Q10

Order Tracking II

You're managing an employee database for an online retail store, and you want to identify only those employees whose salary have been entered on the payroll. Using the Employee.Pay table, retrieve the JobTitle, and IDs for those whose names have appeared on the payroll. CLICK TO SHOW ANSWER Q10

11. Business scenario Q11

Price Existence Check (using EXISTS)

You're managing an e-commerce platform and want to determine if there are products with a specific price value in the CustomerProductTabl table. Your task is to check if there are products with a price above $6. CLICK TO SHOW ANSWER Q11

12. Business scenario Q12

Price Threshold Check (with ALL)

You're managing an e-commerce platform and want to determine if all products have prices exceeding a certain threshold in the " CustomerProductTabl Table." Your task is to check if all products have cost greater than $4. CLICK TO SHOW ANSWER Q12

13. Business scenario Q13

Price Threshold Check (with ANY)

You're managing an e-commerce platform and want to determine if there's at least one product with a price exceeding a certain threshold in the "Products Table." Your task is to check if there's any product with a price greater than $4. CLICK TO SHOW ANSWER Q13

14. Business scenario Q14

Total Income Threshold Calculation and Filtering

You're managing employee payroll data, and you want to calculate the total income (Salary + Bonus) for all employees and filter those who have a total income exceeding a certain threshold. Your task is to calculate the total income and retrieve Employee IDs, Job Titles, and the calculated Total Incomes for employees whose total monthly income exceeds $3,000. Use the Employee.Pay table. CLICK TO SHOW ANSWER Q14

15. Business scenario Q15

Hourly Rate Threshold Calculation and Filtering

You're managing employee payroll data in the, and you want to calculate a threshold value for hourly rate by multiplying it by 100 and filter employees whose calculated threshold exceeds a predefined limit. Your task is to calculate the threshold and retrieve Employee IDs, Job Titles, Hourly Rates, and the calculated Thresholds for employees whose threshold exceeds $1,500. Use the Employee.Pay table. CLICK TO SHOW ANSWER Q15.

The ORDER BY clause

Basic syntax

```
SELECT column1, column2, ...

FROM table_name

WHERE condition (may be optional depending on the question)

ORDER BY column1 [ASC | DESC], column2 [ASC | DESC], ...
```

Note: **[ASC | DESC]**: Specifies the sorting order. Use **ASC** for ascending (default) and **DESC** for descending.

16. Business scenario Q16

Price Range Product Listing

You manage an online marketplace and want to provide users with a product listing sorted by price range. Your task is to retrieve a list of products with their product descriptions and prices, sorted in ascending order of cost using the CustomerProductTabl table. CLICK TO SHOW ANSWER Q16

17. Business scenario Q17

Expensive Product Listing

Continuing with the online marketplace example, you now want to provide users with a product listing sorted by descending price order. Your task is to retrieve a list of products with their product descriptions and cost, sorted in descending order of cost using the CustomerProductTabl table. CLICK TO SHOW ANSWER Q17.

18. Business scenario Q18

Low-Cost Product Listing

You want to create a list of low-cost products available in your online store. Your task is to retrieve a list of products with their descriptions and prices, sorted by ascending price. You also want to limit the list to products with cost under $6. Use the CustomerProductTabl table. CLICK TO SHOW ANSWER Q18.

19. Business scenario Q19

High-Cost Product Listing

You want to provide customers with a list of high-cost products available in your online store. Your task is to retrieve a list of products with their descriptions and prices, sorted by descending price. You also want to limit the list to products with prices above $3. Use the CustomerProductTabl table. CLICK TO SHOW ANSWER Q19.

20. Business scenario Q20

Affordable Products with Alphabetical Order

You want to create a list of affordable products available in your online store, sorted by their product descriptions in ascending alphabetical order. Your task is to retrieve a list of products with their descriptions and costs, sorted by ascending product description. You also want to limit the list to products with prices under $5. Use the CustomerProductTabl table. CLICK TO SHOW ANSWER Q20.

21. Business scenario Q21

Affordable Products with Reverse Alphabetical Order

You want to create a list of affordable products available in your online store, sorted by their product descriptions in descending alphabetical order. Your task is to retrieve a list of products with their descriptions and prices, sorted by descending product description. You also want to limit the list to products with prices under $5. Use the CustomerProductTabl table. CLICK TO SHOW ANSWER Q21.

22. Business scenario Q22

Employee Performance Ranking

You're managing an HR database for a company and want to generate a performance ranking report for employees. You want to order the employees first by their Job title in ascending order and then by their salary in descending order within each department. Your task is to retrieve employee details, including their IDs, Job title, and salaries, sorted according to these criteria. Use the Employee.Pay table. CLICK TO SHOW ANSWER Q22.

23.Business scenario Q23

High-Salary Employees in Specific Roles

You're managing an HR database for a company and want to generate a report on high-salary employees with specific job titles. You want to order the employees first by their roles and then by their salary within each department. Your task is to retrieve employee details, including their IDs, Job titles (specifically Salesman and Shipper), Hire date and salaries. Use the Employee.Pay table. CLICK TO SHOW ANSWER Q23

24. Business scenario Q24

Balanced Product Display

You're managing an online store's product display and want to provide a balanced presentation of products. You want to retrieve a list of products ordered first by their description in ascending order and then by their price in ascending order as well. Your task is to retrieve product details, including IDs, description, and prices, sorted according to these criteria. Use the CustomerProductTabl table. CLICK TO SHOW ANSWER Q24.

25. Business scenario Q25

Product Inventory Reorder

You're managing inventory for an e-commerce company and want to generate a reorder list for products. You want to prioritise products first by their description in descending order and then by their unit cost in ascending order. Your task is to retrieve product details, including IDs, description, and cost, sorted according to these criteria. Use the CustomerProductTabl table.

CLICK TO SHOW ANSWER Q25.

Summarising Functions

COUNT function

Basic syntax

```
SELECT
   COUNT (*) -- Count all rows
   -- or
   -- COUNT (DISTINCT column_name) -- Count distinct values in a specific column
   -- or
   -- COUNT (ALL column_name) -- Count all values, including duplicates
FROM
   table_name
```

26. Business scenario Q26

Employee Payroll Analysis

You're managing employee payroll data and need to perform various counts to analyse the dataset. Your task is to count the following:

a. The total number of employees, which is equivalent to counting the EmployeeID column.

b. The number of distinct job titles in the organisation.

c. The total number of salaries records available in the table.

d. The total number of rows in the Employee table. CLICK TO SHOW ANSWER Q26

SUM function

Syntax

```
SELECT SUM (column_name)

FROM table_name
```

27. Business scenario Q27

Total Salary Calculation

You're managing employee payroll data, and you need to calculate the total sum of all salaries to understand the organisation's payroll expenses. Your task is to calculate the sum of all salaries in the "Employee.Pay" table. CLICK TO SHOW ANSWER Q27.

Average (AVG) Function

Syntax

```
SELECT AVG (column_name)
```

FROM table_name

28. Business scenario Q28

Average Salary Calculation

You're managing employee payroll data, and you need to calculate the average of all salaries to understand the organisation's payroll expenses. Your task is to calculate the sum of all salaries in the "Employee.Pay" table. CLICK TO SHOW ANSWER Q28.

MIN. AND MAX. Functions

Syntax

```
SELECT MAX (column_name)

FROM table_name

And

SELECT MIN (column_name)

FROM table_name
```

29. Business scenario Q29

Salary Analysis

You're managing employee payroll data and need to analyse the salary information for your employees. Your task is to find the following:

a. The minimum salary among all employees.

b. The maximum salary among all employees. CLICK TO SHOW ANSWER Q29.

30. Business scenario Q30

Total Compensation Calculation

You're managing employee payroll data and you need to calculate the total compensation for each employee. Total compensation includes the sum of Salary and Bonus for each employee. Your task is to calculate the total compensation for each employee and find the average total compensation for all employees. CLICK TO SHOW ANSWER Q30.

GROUP BY Clause

Syntax

```
SELECT column1, column2, ...

FROM table_name

WHERE condition (may be optional depending on the question)

GROUP BY column1, column2, ...
```

31. Business scenario Q31

Employee City Analysis

You're managing employee data and you want to analyse the distribution of employees across different cities. Your task is to list the unique combinations of EmployeeID and City to determine which employees are in each city. Use the Employee.Employee table. CLICK TO SHOW ANSWER Q31.

32.Business scenario Q32

Employee Salary Grouping

You're managing employee payroll data, and you want to analyse the distribution of employee salaries while also showing the total salary for each employee. Your task is to calculate the total salary for each employee and list the unique combinations of Salary and EmployeeID to understand how salaries are distributed among employees. Use the Employee.Pay table. CLICK TO SHOW ANSWER Q32.

33.Business Scenario Q33

Employee City Population Analysis

You're managing employee data, and you want to analyse the population of employees in different cities. Your task is to count the number of employees in each city and list the cities along with their respective employee counts. Use the Employee.Employee table. CLICK TO SHOW ANSWER Q33.

34. Business scenario Q34

Employee City Population Analysis I (with Sorting)

You're managing employee data and you want to analyse the population of employees in different cities. Your task is to count the number of employees in each city, list the cities along with their respective employee counts, and sort the results in descending order of employee counts. Use the Employee.Employee table. CLICK TO SHOW ANSWER Q34.

35.Business scenario Q35

Employee City Population Analysis II

Following on from question 34, you're managing employee data and you want to analyse the population of employees in different cities. Your task is to count the number of employees in each city, list the cities along with their respective employee counts, and sort the results by multiple columns using numbers in any sorting order. CLICK TO SHOW ANSWER Q35.

HAVING Clause

Syntax

```
SELECT column1, column2, ...

FROM table_name

GROUP BY column1, column2, ...

HAVING aggregate_function(column) condition;
```

36. Business scenario Q36

Employee Compensation Analysis and Salary Filtering for Payroll Management

You're managing employee payroll data table, and you want to analyse the distribution of employee total compensation while also filtering out specific groups of employees based on salary conditions. Your task is to: Calculate the total compensation (salary plus bonus) for each employee, list the unique combinations of Salary and EmployeeID to understand how salaries are distributed among employees and filter out employees whose total income is above or equal to $3000. CLICK TO SHOW ANSWER Q36.

37.Business scenario Q37

City-wise Employee Population Analysis and Filtering I

You're managing employee data, and you want to analyse the population of employees in different cities. Your task is to count the number of employees in each city, list the cities along with their respective employee counts, filter out cities with more than 3 employees, and sort the results by the number of employees in descending order. CLICK TO SHOW ANSWER Q37.

38.Business scenario Q38

City-wise Employee Population Analysis and Filtering II

Following on from question 38, you're managing employee data, and you want to analyse the population of employees in different cities. Your task is to count the number of employees in each city, list the cities along with their respective employee counts, filter out cities with more than 3 employees, and sort the results by multiple columns using numbers in any sorting order. CLICK TO SHOW ANSWER Q38

The CONCAT function

Syntax

(depending on DB implementations)

SQL Server:

```
SELECT column1 + column2

FROM table_name;
```

MySQL:

```
SELECT CONCAT (column1, column2)

FROM table_name;
```

Oracle:

```
SELECT column1 || column2

FROM table_name;
```

OR

```
SELECT CONCAT (column1, column2)

FROM table_name;
```

39. Business scenario Q39

Employee Full Name Concatenation

You're managing employee data, and you want to create a report that displays the full names of employees by concatenating their first names and last names. Your task is to generate a list of employee IDs along with their full names. Use the Employeee.Employee table. CLICK TO SHOW ANSWER Q39.

TRANSLATE function.

Syntax

TRANSLATE (input_string, from_string, to_string)

40. Business scenario Q40

Protecting Phone Number Privacy

You're managing employee data in the "Employee" table, and you want to protect the privacy of phone numbers by replacing the last 4 digits of each phone number with 'xxx' for security reasons. Your task is to create a report that displays the phone numbers 'xxx' for all employees.

CLICK TO SHOW ANSWER Q40.

UPPER and LOWER Functions

Syntax

UPPER (input_string)

And

LOWER (input_string)

41. Business scenario Q41

Name Uppercase Conversion

You're managing employee data, and you want to convert the last names and first names of employees to uppercase for uniformity in your reports. Your task is to create a report that displays the employee information with uppercase last names and first names. CLICK TO SHOW ANSWER Q41.

SUBSTR(ING) Function

Syntax

MySQL and Oracle:

SUBSTR (Column Name, start_position, length)

SQL Server:

SUBSTRING (Column Name, start_position, length)

42. Business scenario Q42

Phone Number Extraction

You're managing employee data, and you want to extract a portion of the phone numbers for a report. Your task is to create a report that displays a substring of the phone numbers, starting from the 6th character, with a length of 4 characters. Use the Employee.Employee table. CLICK TO SHOW ANSWER Q42.

LTRIM and RTRIM Functions

Syntax

LTRIM (string [, characters])

And

RTRIM (string [, characters])

43. Business scenario Q43

Address Cleanup

You're managing employee data, and you've noticed that some addresses have leading or trailing spaces that need to be removed for consistency. Your task is to create a report that displays employee information with cleaned and trimmed employee addresses. CLICK TO SHOW ANSWER Q43

LENGTH function

Syntax

SQL Server:

LEN (string_expression)

MySQL:

CHAR_LENGTH (string_expression)

PostgreSQL:

LENGTH (string_expression)

44. Business scenario Q44

Name Length Calculation

You're managing employee data, and you need to determine the length of both the last names and first names of employees for reporting purposes. Your task is to create a report that displays the employee IDs along with the lengths of their last names and first names. Use Employee.Employee table. CLICK TO SHOW ANSWER Q44.

The COALESCE Function

Syntax

COALESCE (expression1, expression2, ..., expressionN)

45. Business scenario Q45

Employee Compensation Calculation

You're managing employee compensation data which includes information about hourly pay, salary, and bonuses. Your task is to create a report that calculates the total compensation for each employee, considering all possible compensation components, and handle NULL values gracefully. Use the Employee.Pay table. CLICK TO SHOW ANSWER Q45.

REPLICATE Function

Syntax

RIGHT (REPLICATE (string, length, padding_character))

OR

LEFT (REPLICATE (string, length, padding_character))

Others:

LPAD (string, length, padding_character)

OR

RPAD (string, length, padding_character)

46. Business scenario Q46

Phone Number Formatting

You're managing employee data, and you need to format the "Phone" column to ensure that all phone numbers have a consistent length of 15 characters by adding trailing spaces where needed. This is essential for data consistency and reporting purposes. Use the Employee.Employee table. CLICK TO SHOW ANSWER Q46.

47. Business scenario Q47

Product Description Formatting

You're managing product data in the "Product" table, and you need to format the "ProductDescription" column to ensure that all product descriptions have a consistent length by adding dots or periods at the end. This is useful for creating a uniform display of product descriptions. Use the CustomerProductTabl table. CLICK TO SHOW ANSWER Q47.

Conversion Functions (CAST AND CONVERT)

General syntax for Cast:

CAST (expression AS data_type)

Syntax for Convert:

SQL Server and Sybase:

CONVERT (data_type, expression, [style])

MySQL:

CONVERT (expression, data_type)

Oracle:

TO_<data_type>(expression)

48. Business scenario Q48

Price Conversion

You're managing product data, and the "Cost" column is currently stored as strings or money. You need to convert the "Cost" column to a numeric data type (e.g., DECIMAL or FLOAT) to perform accurate calculations and comparisons based on cost. Your goal is to convert the prices for all products. Use the CustomerProductTabl table. CLICK TO SHOW ANSWER Q48.

Combining character functions (CONCAT AND SUBSTRING)

49. Business scenario Q49

Employee Data Transformation for Reporting

You're managing employee data, and you need to perform data transformations on the names and phone numbers for reporting purposes. Specifically, you want to: Create a new column called "FullName" that concatenates the "FirstName" and "LastName" columns with a space in between. Format the "Phone" column to separate the numbers with dashes (e.g., "555-555-1234"). Use the Employee.Employee table. CLICK TO SHOW ANSWER Q49.

DATE Functions

(With CASE-WHEN-THEN-ELSE conditions)

50. Business scenario Q50

Years of Service Calculation

You're managing employee data, and you need to calculate the years of service for each employee based on their "HireDate" and "LastDate." You want to identify employees who have worked for the company for a certain number of years. Use the Employee.Pay table. Calculate the years of service for each employee based and Identify employees who have worked for the company for at least 7 years. CLICK TO SHOW ANSWER Q50.

51. Business scenario Q51

Employee Salary Projection

You're managing employee data, and you need to project the potential future salaries of employees based on their current salary, bonus, and a salary increase rate. Employees are eligible for a salary increase if they have been with the company for at least 2 years and their Last Date is between 2011 and 2012. The salary increase rate is 5%. CLICK TO SHOW ANSWER Q51.

52. Business scenario Q52

Performance Bonus Calculation

You manage employee data, and you need to calculate performance bonuses for employees based on their job titles and salary. Employees with the job title "Manager" are eligible for a 10% bonus on their current salary, while other employees receive a 5% bonus on their salary. However, the bonus should only be calculated for employees whose last date is between 2011 and 2012. CLICK TO SHOW ANSWER Q52.

53.Business scenario Q53

Performance Bonus Calculation

Still following on from the previous question, you manage employee data and need to calculate bonuses for employees based on their job titles, salary, and the current month. Employees with the job title "Manager" receive a 10% bonus on their current salary if the current month is January. For other employees, if the current month is July, they receive a 7% bonus on their salary. Otherwise, employees receive a standard 5% bonus on their salary. CLICK TO SHOW ANSWER Q53.

54. Business scenario Q54

Employee Salary Adjustment

You manage employee data, and you need to calculate and update salaries for employees based on their job titles and years of service. Employees with the job title "Salesman" are eligible for a 10% salary increase if they have been with the company for more than 5 years. Employees with the job title "Shipper" receive a 5% salary increase if they have been with the company for more than 3 years. For all other employees, no salary adjustment is made. CLICK TO SHOW ANSWER Q54.

55. Business scenario Q55

Overtime Pay Calculation

You manage employee data, and you need to calculate overtime pay for employees who have worked more than 40 hours a week. Employees are paid their hourly rate for the first 40 hours and 1.5 times their hourly rate for any additional hours worked beyond 40 hours in a week. Calculate and update the "Salary" column for employees to include overtime pay based on their weekly hours worked. CLICK TO SHOW ANSWER Q55.

56. Business scenario Q56

Vacation Days Calculation

In your organisation, employees accrue vacation days based on their years of service. Employees with less than 5 years of service receive 10 vacation days per year, and employees with 5 or more years of service receive 15 vacation days per year. Calculate and display the number of vacation days accrued by each employee based on their years of service assuming they are still in employment up till today's date. Employees with less than 5 years of service receive 10 vacation days per year, and employees with 5 or more years of service receive 15 vacation days per year. Use the Employee.Pay table. CLICK TO SHOW ANSWER Q56.

57.Business scenario Q57

Retirement Eligibility Calculation

In your organisation, employees are eligible for retirement after they reach a certain number of years of service. You need to determine which employees are eligible for retirement based on their hire dates. Identify and display the employees who are eligible for retirement based on reaching service years of 20 years or more. CLICK TO SHOW ANSWER Q57.

58.Business scenario Q58

Date Conversion for Reporting

In your organisation, you need to generate a report that includes the employee's hire date and last working date in a specific date format for reporting purposes. The required date format is "YYYY-MM-DD. Use the Employee.Pay table. CLICK TO SHOW ANSWER Q58.

59. Business scenario Q59

Employee Anniversary Recognition

In your organisation, you want to generate a report to recognise employees' work anniversaries. The report should include the employee's job title and a message congratulating them on their work anniversary. The message should specify the number of years they have been with the company. Use the Employee.Pay table. CLICK TO SHOW ANSWER Q59.

(Multiple-table queries)

Table JOINS

INNER JOIN

Syntax

SELECT *

FROM Table1

INNER JOIN Table2 ON Table1.column = Table2.column.

60. Business scenario Q60

Employee Salary and Job Information

In a company, you need to generate a report that combines employee personal information and their job-related details. You have two tables, Employee.Employee and Employee.pay. Create a report that combines employee personal information (name, address, and contact) with their job-related details (job title, hire date, salary, and bonus). CLICK TO SHOW ANSWER Q60.

61. Business scenario Q61

Employee Performance Evaluation

In a company, you want to evaluate employee performance by combining their personal information and job-related details. You have two tables, Employee.Employee and Employee.pay. Create a report that combines employee personal information (name, address, and contact) with their job-related details (job title, hire date, salary, and bonus) for performance evaluation. Additionally, calculate the total compensation (Salary + Bonus) for each employee. CLICK TO SHOW ANSWER Q61.

62. Business scenario Q62

Analysis of Employee Promotion Eligibility

In a company, you want to identify employees eligible for promotion based on their job titles and the duration of their employment. You have two tables: Employee.Employee and Employee.pay. Create a report that identifies employees eligible for promotion based on the following criteria:

- Employees with the job title "Sales Support" or "Salesman."
- Employees who have been with the company for at least 5 years. CLICK TO SHOW ANSWER Q62.

63. Business scenario Q63

Employee Bonus Eligibility

In a company, you want to identify employees eligible for a year-end bonus based on their job titles and bonus amounts. You have two tables: Employee.Employee and Employee.pay. Create a report that lists employee details and sort the results by job title and bonus amount. You want to identify employees who are eligible for a bonus based on the following criteria:

- Employees with the job title "Salesman" or "Sales Support"
- Employees who have received a bonus greater than or equal to $1000. CLICK TO SHOW ANSWER Q63.

64. Business scenario Q64

Employee Tenure and Salary Analysis

In a company, you want to analyse the tenure and salaries of employees in a specific role and also sorted by job title and salary amount. You have two tables: Employee.Employee and Employee.pay. Create a report that lists employee details for the role of 'Salesman' or sales support' and sorts the results by job title and salary. Additionally, calculate the tenure (in years) of each employee from their Hire Date to the current date. CLICK TO SHOW ANSWER Q64.

65. Business scenario Q65

Customer Order Analysis

In a retail business, you want to analyse customer orders, including the product details, for the cities from which orders are made, sorted by order date and quantity. You have three tables: CustomerTab.CustomerTabl, CustomerTab.CustomerOrdersTabl, and CustomerTab.CustomerProductTabl. Create a report that lists customer orders and includes product details. Sort the results by order date and order quantity in any order. CLICK TO SHOW ANSWER Q65.

LEFT JOIN

Syntax

```
SELECT *

FROM Table1

LEFT JOIN Table2 ON Table1.column = Table2.column.
```

66. Business scenario Q66

Customer Order Analysis with Missing Orders

In a retail business, you want to analyse customer orders, including the product details, for all customers, even if they haven't placed any orders. You have three tables: CustomerTab.CustomerTabl, CustomerTab.CustomerOrdersTabl, and CustomerTab.CustomerProductTabl. Create a report that lists all customers and their orders (if any), including product details. Sort the results by customer name, order date, and product description. CLICK TO SHOW ANSWER Q66.

RIGHT JOIN

Syntax

```
SELECT *
FROM Table1
RIGHT JOIN Table2 ON Table1.column = Table2.column.
```

67. Business scenario Q67

Customer Order Analysis with Optional Customers

In a retail business, you want to analyse customer orders, including the product details, for all orders, even if the customers' information is missing. You have three tables: CustomerTab.CustomerTabl, CustomerTab.CustomerOrdersTabl, and CustomerTab.CustomerProductTabl. Create a report that lists all customer orders (if any) and includes customer information (if available), along with product details. Sort the results by customer name, order date, and product description. CLICK TO SHOW ANSWER Q67.

FULL JOIN

Syntax

SELECT *

FROM Table1

FULL JOIN Table2 ON Table1.column = Table2.column.

68. Business scenario Q68

Customer-Order Relationship Analysis

In your business, you want to analyse the relationship between customers and their orders. You have two tables: CustomerTab.CustomerTabl and CustomerTab.CustomerOrdersTabl. Create a comprehensive report that lists all customers and their orders, including those customers who haven't placed any orders yet and those orders that have no associated customer information. Sort the results alphabetically by customer name and by order date.

CLICK TO SHOW ANSWER Q68.

SELF JOIN

Syntax

SELECT t1.column1, t2.column2

FROM TableName t1

JOIN TableName t2 ON t1.relatedColumn = t2.relatedColumn;

69. Business scenario Q69

Employee Carpooling Analysis

In your company, you want to promote carpooling among employees who live in the same city to reduce transportation costs and environmental impact. Identify pairs of employees who live in the same city and can potentially carpool together. Create a list of these employee pairs, along with their contact information. Use the Employee.Employee table. CLICK TO SHOW ANSWER Q69.

OUTER JOIN

Syntax

FROM TABLE1 {RIGHT | LEFT | FULL} [OUTER] JOIN ON TABLE2

70. Business scenario Q70

Product Analysis with Missing Order Quantity

In a retail business, you want to analyse customer orders, including the product details, even if there might be missing orders for certain products. You have two tables: CustomerTab.CustomerOrdersTabl, and CustomerTab.CustomerProductTabl. Create a report that lists all product descriptions and their orders (if any). CLICK TO SHOW ANSWER Q70.

The Cartesian Product or CROSS JOIN

Syntax

```
SELECT *
FROM table1
CROSS JOIN table2;
```

71. Business scenario Q71

Total Compensation Report for HR Analysis: Salary and Bonus

In a human resources analysis project, you have two tables: Employee.Employee and Employee.Pay. You want to calculate the total compensation for each employee, which includes their salary and bonus. You need to create a report that shows the total compensation for each employee. CLICK TO SHOW ANSWER Q71.

SUBQUERY

Syntax

```
SELECT column_name
FROM table
WHERE column_name = (SELECT column_name
                     FROM table
                     WHERE conditions);
```

72. Business scenario Q72

Identifying Employees with Highest Bonuses

In a company's human resources analysis project, you have two tables: Employee.Employee and Employee.Pay. The company is interested in identifying employees with the highest bonuses. They want to create a report that lists the top five employees with the highest bonuses and includes their names and bonus amounts. CLICK TO SHOW ANSWER Q72.

73. Business scenario Q73

Identifying Customers with High Total Order Quantities

You want to find customers who have placed orders with a total quantity greater than a certain threshold. Assuming we want to find customers, who have placed orders with a total order quantity greater than 25. Use the CustomerTabl and CustomerOrdersTabl tables. CLICK TO SHOW ANSWER Q73.

74. Business scenario Q74

Customer Order Analysis: Total Product Cost and Top Spender

Calculate the total cost of products ordered by each customer and identify the customers who have spent the most on their orders. Use the OrdersTabl and ProductTabl. CLICK TO SHOW ANSWER Q74.

75. Business scenario Q75

Identifying Unordered Products: Customer and Product Analysis

Find the products that have never been ordered by any customer. Use the OrdersTabl and ProductTabl. CLICK TO SHOW ANSWER Q75.

76. Business scenario Q76

Customer Average Order Cost Analysis with Threshold

Calculate the average cost of products ordered by each customer, and then identify customers whose average order cost is above a threshold of 50. Use the OrdersTabl and ProductTabl. CLICK TO SHOW ANSWER Q76.

77. Business scenario Q77

Average Salary Analysis for Employees

Find the average salary for employees who joined the company in the same year as the employee with the highest salary, and also provide the count of such employees. Use the Employee.Pay table. CLICK TO SHOW ANSWER Q77.

78. Business scenario Q78

Promotion Eligibility Analysis

Identify employees who are eligible for a promotion based on the following criteria: they must have been with the company for at least five years, and their current salary is below the average salary of all employees in the same job title. Use the Employee and Employee.Pay tables. CLICK TO SHOW ANSWER Q78.

79. Business scenario Q79

Identifying High-Value Customers

A retail company wants to analyse its customer base to identify high-value customers. To do this, they need to find the top 10 customers who have placed the highest total quantity of orders. Additionally, they want to know the addresses of these top 10 customers. Use the CustomerTabl and OrdersTabl. CLICK TO SHOW ANSWER Q79.

UNION Operator

Syntax

```
SELECT column1, column2, ...

FROM table1

UNION

SELECT column1, column2, ...

FROM table2.
```

UNION ALL Operator

Syntax

```
SELECT column1, column2, ...

FROM table1

UNION ALL

SELECT column1, column2, ...

FROM table2.
```

INTERSECT Operator

Syntax

```
SELECT column1, column2, ...

FROM table1

INTERSECT
```

```
SELECT column1, column2, ...

FROM table2;
```

EXCEPT Operator

Syntax

```
SELECT column1, column2, ...

FROM table1

EXCEPT

SELECT column1, column2, ...

FROM table2.
```

80. Business scenario Q80

Analyse top selling products

A retail company wants to create a consolidated list of top-selling and least-selling products from their inventory. They also want to include the product cost in the report. However, the data is stored in a single table: The Product table. To create this report, they need to combine the necessary information from the Product table into a single result set. CLICK TO SHOW ANSWER Q80.

81. Business scenario Q81

Combining Product and Customer Data for Marketing in a Specific City

A company wants to create a single list of all products and customers that are associated with a particular address for marketing purposes. However, the data is stored in two separate tables: Product table and Customer table. To achieve this, they need to combine the relevant information from both tables into a single result set for a specific city. CLICK TO SHOW ANSWER Q81.

82. Business scenario Q82

Identify Customers Who Have Not Ordered

A company wants to identify customers who have never placed an order. They have customer information stored in the Customer table and order information in the Order table. To find customers who have not placed orders, they need to combine information from both sources and identify customers who do not have any corresponding orders. CLICK TO SHOW ANSWER Q82.

83. Business scenario Q83

Identifying Unordered Products

A company wants to identify products that have never been ordered by any customers. They have product information stored in the Product table and order information in the Order table. To find products that have not been ordered, they need to combine information from both sources and identify products that do not have any corresponding orders. CLICK TO SHOW ANSWER Q83.

ROLLUP AND CUBE Expressions

Syntax

```
SELECT column1, column2, ..., aggregate_function(column)

FROM table
```

GROUP BY ROLLUP (column1, column2, ...);

84. Business scenario Q84

Payroll Expense Analysis

A company wants to analyse its payroll expenses and generate a report that provides insights into the total salary expenditure by job title and department. The company stores employee pay-related information in the Employee Pay table. To create a comprehensive payroll report, they need to calculate subtotals and a grand total of salary expenses, considering both job titles and departments. CLICK TO SHOW ANSWER Q84.

85. Business scenario Q85

Analysis of customer purchase behaviour

A company wants to know the total number of orders placed by each customer, as well as the total quantity of each product ordered by each customer using the Order table. CLICK TO SHOW ANSWER Q85.

86. Business scenario Q86

Analyse Customer Purchase Behaviour by Month.

A company wants to know the total number of orders placed by each customer, as well as the total quantity of each product ordered by each customer, grouped by month using the Order table. CLICK TO SHOW ANSWER Q86.

87. Business scenario Q87

Analyse Employee Salaries by Job Title.

A company wants to know the total salary paid to each job title using the Employee Pay table.

CLICK TO SHOW ANSWER Q87.

88. Business scenario Q88

Compensation and HR Analytics

A company wants to analyse its employee compensation data to gain insights into salary trends, job titles, and compensation expenses. They have two tables, the "Employee" table and the "EmployeePay" table. CLICK TO SHOW ANSWER Q88.

89. Business scenario Q89

Sales and Market Expansion

A retail company wants to analyse its sales data and customer information to make informed decisions about expanding its market presence. They have two tables, the "Order" table, and the "Customer" table. And maybe the product table too. CLICK TO SHOW ANSWER Q89.

Other uses of CASE-WHEN-ELSE-END

90. Business scenario Q90

Customer Loyalty Program Analysis

A retail company, "ShopSmart," wants to analyse customer purchases to identify loyal customers and provide them with exclusive discounts. They have two tables, "Product" and

"Order," . Determine whether a customer is a loyal customer based on the total amount spent. A loyal customer is defined as someone who has spent more than 200 in total. CLICK TO SHOW ANSWER Q90.

91. Business scenario Q91

Product Promotion Analysis

A retail company, "SaleSmart," wants to analyse product sales data to identify which products should be included in a new promotional campaign. They have two tables, "Product" and "Order," and need to identify products that meet specific criteria for promotion. Identify products that have generated total sales revenue exceeding 1,000. Determine which products have been sold more than 100 times. Identify products with a profit margin of at least 30%. Profit margin is defined as (Profit / Revenue) * 100, where Profit = Total Sales Revenue - Total Cost. Find products that have not been sold at all. CLICK TO SHOW ANSWER Q91.

Part 2

Intermediate Level Exercises

Database: AdventureWorks (2019 or 2022)

92. Business scenario Q92

Product Selection and Pricing Analysis

A retail company wants to analyse its product inventory to identify products with specific colours and price ranges for a sales promotion. They need to retrieve information about products with non-null colours that are not silver, black, or white, and whose list prices fall between $75 and $750. Additionally, they want to rename the "StandardCost" column as "Price" for clarity. The results should be sorted by list price in descending order. CLICK TO SHOW ANSWER Q92.

93. Business scenario Q93

Identifying the Most Expensive Products with a Specific Product Number Prefix

A retail company wants to identify and list the top 10 most expensive products with product numbers that begin with 'BK'. They are interested in obtaining the product ID, name, and colour of these high-value products for inventory management and marketing purposes. CLICK TO SHOW ANSWER Q93.

94. Business scenario Q94

Contact Information Analysis

A business with a database of contact information First, they want to create a list of all the contact people where the first 4 characters of their last name are equal to the first 4 characters of their email address.

Second, they want to find all the contact people whose first name and the last name begin with the same character, and then create a new column called full name combining their first name and their last name. Finally, they would like to add a column with the length of the full name. CLICK TO SHOW ANSWER Q94.

95. Business scenario Q95

Product Subcategory Manufacturing Analysis

A manufacturing company wants to identify product subcategories that take an average of 3 days or longer to manufacture. They need this information to optimise production processes and possibly make improvements in subcategories where manufacturing time is a concern. CLICK TO SHOW ANSWER Q95.

96. Business scenario Q96

Price Segmentation for Products

A retail company wants to segment its products based on their prices into different categories: low value, mid value, mid to high value, and higher value. They also want to filter the results to include only products with colours black, silver, and red. This price segmentation will help them tailor marketing strategies and promotions for different product categories. If price is less than $200 then it's low value. If price is between $201 and $750 then it is mid value. If price is between $751 and $1250 then it is mid to high value. All else is higher value. They want to filter the results only for products that are black, silver and red (colour). CLICK TO SHOW ANSWER Q96.

97. Business scenario Q97

Long Service Award Eligibility Analysis

A company with an employee database, wants to identify how many employees will be eligible for a Long Service Award in the next 5 years. The eligibility criterion for this award is a long service of 20 years or more with the company. This information is important for HR planning and recognition programs. CLICK TO SHOW ANSWER Q97.

98. Business scenario Q98

Calculating Years to Retirement for Employees

A company with an employee database wants to calculate how many more years each employee has to work before reaching the retirement age of 65. This information is important for retirement planning and HR management. CLICK TO SHOW ANSWER Q98.

99. Business scenario Q99

Implementing a New Pricing Policy

A retail company wants to implement a new pricing policy for its products based on the colour of each item. The policy includes adjusting prices for different colours and calculating a commission for each item based on the new prices. The specific pricing adjustments are as follows:

- If the colour is white, increase the price by 8%.
- If the colour is yellow, reduce the price by 7.5%.
- If the colour is black, increase the price by 17.2%.
- If the colour is multi, silver, silver/black, or blue, take the square root of the price and double that value.

The commission for each item is calculated as 37.5% of the new prices. CLICK TO SHOW ANSWER Q99.

100. Business scenario Q100

Retrieving Sales Personnel Information

A sales-focused company wants to retrieve information about its sales personnel and their sales quotas. Specifically, they need to display the FirstName, LastName, HireDate, SickLeaveHours, and the Region where each salesperson works. CLICK TO SHOW ANSWER Q100.

101. Business scenario Q101

Retrieving Order Details with Customer and Salesman

A company with an order database wants to retrieve detailed information about their orders. Specifically, they want to display the order number, order date, order amount, the customer who placed the order, the salesman who works for that customer, and the commission the salesman receives for the order. CLICK TO SHOW ANSWER Q101

102. Business scenario Q102

Calculating Commission and Margin for Products

A retail company wants to calculate commission and margin for all of its products based on the standard cost and product colours. The commission should be calculated as 14.790% of the standard cost, and the margin should be adjusted depending on the product's colour following the specified rules. CLICK TO SHOW ANSWER Q102.

General Sales Analysis and Customer Demographics

Possible database Tables and Columns required:

- Table: Sales.SalesOrderHeader

  Columns: SalesOrderID, OrderDate, TotalDue, CustomerID

- Table: Sales.SalesOrderDetail

  Columns: SalesOrderID, ProductID, OrderQty, UnitPrice, LineTotal

- Table: Production.Product

  Columns: ProductID, Name, ProductNumber

- Table: Person.Person
- Table: Sales.Customer

103. Business scenario Q103

Total Profit Calculation for Product Portfolio Analysis

A retail company wants to calculate the total profit generated by their products to assess overall profitability. To do this, they need to sum up the profit for each product in their database. CLICK TO SHOW ANSWER Q103.

104. Business scenario Q104

Sales Analysis for Identifying Top-Selling Products and Trends

A retail company needs to conduct sales analysis to gain insights into their sales performance, trends, and key metrics. They aim to make data-driven decisions to identify top-selling products. CLICK TO SHOW ANSWER Q104.

105. Business scenario Q105

Sales and Customer Demographics Analysis

A retail company needs to conduct sales analysis to gain insights into their sales performance, trends, and key metrics. They aim to make data-driven decisions to understand and analyse customer demographics. CLICK TO SHOW ANSWER Q105.

106. Business scenario Q106

Sales Analysis for High-Value Customer Identification and Insights

A retail company needs to conduct sales analysis to gain insights into theirsales performance, trends, and key metrics. They aim to make data-driven decisions to identify high-value customers. CLICK TO SHOW ANSWER Q106.

Inventory Management

Possible database Tables and Columns required:

- Table: Production.Product

  Columns: ProductID, Name, StandardCost

- Table: Purchasing.ProductVendor

  Columns: ProductID, VendorID, MinOrderQty

- Table: Purchasing.PurchaseOrderDetail

  Columns: PurchaseOrderID, ProductID, OrderQty, UnitPrice

- Table: Sales.SalesOrderDetail

  Columns: SalesOrderID, ProductID, OrderQty

107. Business scenario Q107

Inventory Optimisation through Value Analysis

A retail company needs to optimise its inventory management processes to reduce carrying costs while ensuring product availability and customer satisfaction. They aim to make data-driven decisions regarding by determining the total inventory value of products. CLICK TO SHOW ANSWER Q107.

108. Business scenario Q108

Identifying Slow-Moving and Obsolete Inventory

A retail company needs to optimise its inventory management processes to reduce carrying costs while ensuring product availability and customer satisfaction. They aim to make data-driven decisions regarding by Identifying slow-moving or obsolete inventory. CLICK TO SHOW ANSWER Q108.

General Supplier Evaluation

Possible Database Tables and Columns required:

- Table: Purchasing.ProductVendor

Columns: ProductID, VendorID, StandardPrice

- Table: Purchasing.PurchaseOrderHeader

Columns: PurchaseOrderID, VendorID, OrderDate, ShipDate

- Table: Purchasing.PurchaseOrderDetail

Columns: PurchaseOrderID, ProductID, OrderQty, UnitPrice

- Table: Purchasing.Vendor

Columns: VendorID, Name

- Table: Purchasing.PurchaseOrderDetail

109. Business scenario Q109

Supplier Performance Evaluation: Average Delivery Time Analysis

A retail company wants to evaluate its suppliers to ensure they meet quality, delivery, and cost requirements. They aim to assess supplier performance and identify opportunities for improvement while maintaining strong supplier relationships. The task is to calculate the average delivery time for each supplier. CLICK TO SHOW ANSWER Q109.

110. Business scenario Q110

Supplier Performance Evaluation: On-Time Delivery Percentage Analysis

A retail company wants to evaluate its suppliers to ensure they meet quality, delivery, and cost requirements. They aim to assess supplier performance and identify opportunities for improvement while maintaining strong supplier relationships. The task is to calculate the on-time delivery percentage for each supplier. CLICK TO SHOW ANSWER Q110.

111.Business scenario Q111

Supplier Performance Evaluation: Average Cost Performance Analysis

A retail company wants to evaluate its suppliers to ensure they meet quality, delivery, and cost requirements. They aim to assess supplier performance and identify opportunities for improvement while maintaining strong supplier relationships. The task is to calculate the average cost performance for each supplier. CLICK TO SHOW ANSWER Q111.

General Financial Reporting

A retail company needs to prepare comprehensive financial reports to assess its financial performance, monitor expenses, and track profitability. The objectives are to generate financial statements, including income statements and balance sheets, to support decision-making and financial planning.

Possible Database Tables and Columns Required:

- Table: Sales.SalesOrderHeader

  Columns: SalesOrderID, OrderDate, TotalDue
- Table: Purchasing.PurchaseOrderHeader

  Columns: PurchaseOrderID, OrderDate, TotalDue
- Table: HumanResources.Employee

  Columns: EmployeeID, FirstName, LastName
- Table: HumanResources.EmployeePayHistory

  Columns: EmployeeID, Rate, PayFrequency, PayDate

112. Business scenario Q112

The task is to first determine the total sales revenue. CLICK TO SHOW ANSWER Q112.

113. Business scenario Q113

The task is to calculate total expenses (e.g., employee salaries, purchase costs). CLICK TO SHOW ANSWER Q113.

114. Business scenario Q114

The task is to calculate net income (revenue - expenses). CLICK TO SHOW ANSWER Q114.

115. Business scenario Q115

The task is to calculate total actual expenses (e.g., actual employee salaries). CLICK TO SHOW ANSWER Q115.

Business Case Problem: Supply Chain Optimisation

A manufacturing and retail company, seeks to optimize its supply chain logistics to achieve cost efficiency, analyse and mitigate supply chain disruptions, and determine optimal transportation routes and distribution centres. These efforts are crucial for enhancing overall supply chain performance and ensuring timely delivery to customers.

Possible Database Tables and Columns required:

- Table: Production.Product

  Columns: ProductID, Name, StandardCost
- Table: Production.Location

  Columns: LocationID, Name, CostToShip
- Table: Production.Supplier

  Columns: SupplierID, Name, Country
- Table: Production.PurchaseOrderDetail

  Columns: PurchaseOrderID, ProductID, OrderQty
- Table: Production.WorkOrder

  Columns: WorkOrderID, ProductID, OrderQty

116. Business scenario Q116

Your task is to determine the total cost of products in inventory. CLICK TO SHOW ANSWER Q116

117. Business scenario Q117

Your task is to calculate the total cost of shipping from different locations. CLICK TO SHOW ANSWER Q117

118. Business scenario Q118

Your task is to identify the location with the highest shipping cost. CLICK TO SHOW ANSWER Q118

119. Business scenario Q119

Your task is to identify products with the highest number of delayed purchases orders. CLICK TO SHOW ANSWER Q119.

Market Expansion and Segmentation

A retail company aims to expand its product offerings into new markets while effectively segmenting existing and potential markets based on demographics, geography, or psychographics. They seek to identify opportunities for growth and tailor marketing strategies to specific market segments.

Possible required Database Tables and Columns:

- Table: Sales.Customer

  Columns: CustomerID, FirstName, LastName, EmailAddress, Phone

- Table: Sales.SalesOrderHeader

  Columns: SalesOrderID, CustomerID, OrderDate, TotalDue

- Table: Sales.CustomerAddress

  Columns: CustomerID, AddressID
- Table: Person.Address

  Columns: AddressID, City, StateProvince, CountryRegion, PostalCode
- Table: Sales.SalesTerritory

  Columns: TerritoryID, Name

120. Business scenario Q120

Your task is to determine total sales by territory to identify potential markets. CLICK TO SHOW ANSWER Q120.

121. Business scenario Q121

Your task is to segment markets based on territory demographics. CLICK TO SHOW ANSWER Q121.

Employee Retention Analysis

A growing company is concerned about high employee turnover rates. They aim to analyse employee attrition, identify the factors contributing to it, and develop effective retention strategies to retain valuable talent.

Possible Database Tables and Columns:

- Table: HumanResources.Employee

  Columns: EmployeeID, FirstName, LastName, HireDate
- Table: HumanResources.EmployeePayHistory

  Columns: BusinessEntityID (corresponds to EmployeeID), RateChangeDate, Rate
- Table: HumanResources.EmployeeDepartmentHistory

Columns: BusinessEntityID (corresponds to EmployeeID), DepartmentID, ShiftID, StartDate, EndDate

- Table: HumanResources.Department

  Columns: DepartmentID, Name

122. Business scenario Q122

Your task is to calculate overall turnover rate. CLICK TO SHOW ANSWER Q122

123. Business scenario Q123

Your task is to identify departments with the highest turnover rates. CLICK TO SHOW ANSWER Q123

124. Business scenario Q124

Your task is to identify employees with the highest rate of pay increase. CLICK TO SHOW ANSWER Q124

125. Business scenario Q125

Long Service Award Calculation.

A company that values employee loyalty and commitment, wants to recognize and reward employees who have completed a significant number of years in service. They aim to calculate the number of employees eligible for a Long Service Award who have 5+ years of service. CLICK TO SHOW ANSWER Q125

126. Business scenario Q126

Retirement Age Analysis

A company that values workforce planning and employee well-being, wants to assess the retirement readiness of its employees. They aim to calculate the number of years each employee has to work before reaching the company's defined retirement age, if the retirement age is set at 65 years ensuring a smooth transition and succession planning. CLICK TO SHOW ANSWER Q126

127. Business scenario Q127

Employee Productivity Ranking

A company with a sales team and employees in various roles, wants to assess and rank employee productivity. They aim to rank employees based on specific productivity metrics, such as total sales generated or total hours worked, to identify high-performing employees and recognize their contributions. Your task is to rank employees based on their productivity (e.g., total sales generated). CLICK TO SHOW ANSWER Q127.

Possible Database Tables and Columns

- Table: Sales.SalesPerson

  Columns: BusinessEntityID (corresponds to EmployeeID), FirstName, LastName
- Table: Sales.SalesOrderHeader

  Columns: SalesOrderID, SalesPersonID
- Table: HumanResources.Employee

  Columns: BusinessEntityID (corresponds to EmployeeID), FirstName, LastName
- Table: HumanResources.EmployeePayHistory

  Columns: BusinessEntityID (corresponds to EmployeeID), RateChangeDate, Rate

128. Business scenario Q128

Marketing Campaign Effectiveness

A company that runs various marketing campaigns, wants to assess the effectiveness of its marketing efforts. They aim to measure the success of marketing campaigns by tracking customer response rates and conversion rates to understand how well these campaigns are performing and to optimise future marketing strategies. They want to measure the effectiveness of marketing campaigns by tracking customer response rates and conversion rates. CLICK TO SHOW ANSWER Q128.

Possible Database Tables and Columns:

- Table: Sales.Customer

  Columns: CustomerID, FirstName, LastName
- Table: Sales.SalesOrderHeader

  Columns: SalesOrderID, CustomerID

129. Business scenario Q129

Promotion Effectiveness Analysis

A retail company wants to assess the effectiveness of its promotional campaigns. They aim to evaluate the impact of various promotional efforts on sales and customer acquisition. This analysis will help optimise marketing spend by identifying which promotions are the most successful in driving revenue and acquiring new customers. They want to evaluate the impact of promotional campaigns on sales and customer acquisition. Use Sales.SpecialOffer and Sales.SalesOrderHeader. CLICK TO SHOW ANSWER Q129.

130. Business scenario Q130

Customer Retention Analysis

A retail company wants to understand and improve customer retention rates. They aim to analyse customer churn, identify factors that influence customer retention, and develop strategies to reduce customer churn. This analysis will help build stronger customer relationships and increase customer loyalty. They want to analyse customer retention rates and factors influencing retention. Use Sales.Customer and Sales.SalesOrderHeader tables. CLICK TO SHOW ANSWER Q130.

131. Business scenario Q131

Product Life Cycle Analysis

A retail company wants to assess the life cycle stages of various products in its catalogue. They aim to determine whether each product is in the introduction, growth, maturity, or decline stage of its life cycle. This analysis will inform inventory management decisions and help shape marketing strategies for each product. Use the Production.Product table. CLICK TO SHOW ANSWER Q131.

132. Business scenario

Customer Lifetime Value (CLV) Calculation (hard)

A retail company wants to assess the value of each customer over their lifetime of engagement with the company. They aim to calculate the Customer Lifetime Value (CLV) for each customer to identify high-value customers and tailor marketing efforts, loyalty programs, and customer service strategies accordingly. Use the Sales.Customer and Sales.SalesOrderHeader tables. CLICK TO SHOW ANSWER Q132.

133. Business scenario Q133

Product Quality Analysis

A manufacturing company wants to monitor and analyse product quality data to ensure that its products meet or exceed quality standards. They aim to implement a comprehensive quality analysis system that can help identify issues early in the manufacturing process and continuously improve product quality. They want to analyse product quality based on customer reviews. Use Production.Product and Production.ProductReview. CLICK TO SHOW ANSWER Q133.

134. Business scenario Q134

Sales Forecasting

A retail company wants to predict future sales based on historical data and market trends. They aim to implement a sales forecasting system that can provide accurate sales predictions, allowing for better inventory planning and resource allocation. This analysis will help optimise stock levels, meet customer demand, and allocate resources efficiently. The want to predict future sales using historical data for the last 10 years. Use Sales.SalesOrderHeader table. CLICK TO SHOW ANSWER Q134.

135. Business scenario Q135

E-commerce Conversion Rate Optimisation

An e-commerce company aims to optimise its website's conversion rate to enhance the shopping experience and ultimately increase sales. They want to analyse conversion rates for the last 20 years to identify areas for improvement and implement strategies to convert more visitors into customers. Use Sales.SalesOrderHeader and Sales.SalesPerson tables. CLICK TO SHOW ANSWER Q135

136. Business scenario Q136

Analysing Customer Locations for Sales Optimisation

You work for a retail company. Your company is looking to optimise its sales strategy by analysing customer locations to determine which regions have the highest sales volumes. This information will help the company focus its marketing efforts, inventory management, and customer support more effectively. Your task is to create an SQL query that calculates the total sales amount for each city. You should focus on customers' shipping addresses and determine the cities with the highest total sales. CLICK TO SHOW ANSWER Q136.

Possible Database Tables and Columns required:

- Person.Address Table:

  Columns: AddressID, City, StateProvinceID, PostalCode
- Person.BusinessEntityAddress Table:

  Columns: BusinessEntityID, AddressID, AddressTypeID
- Sales.SalesOrderHeader Table:

  Columns: SalesOrderID, CustomerID, ShipToAddressID

137. Business scenario Q137

Customer Geographical Analysis for Sales Expansion

You are working for a retail company which is considering expanding its business to new locations and wants to analyse customer demographics to identify potential areas for expansion. This analysis will help the company make informed decisions about where to open new stores. Your task is to create an SQL query that provides demographic insights by calculating the total number of customers and the average order value for each city. CLICK TO SHOW ANSWER Q137.

Possible Database Tables and Columns required:

- Person.Address Table:

  Columns: AddressID , City, StateProvinceID, PostalCode
- Person.BusinessEntityAddress Table:

  Columns: BusinessEntityID, AddressID, AddressTypeID
- Sales.SalesOrderHeader Table:

  Columns: SalesOrderID, CustomerID, ShipToAddressID

138. Business scenario Q138

Order Analysis for Customer Segmentation

Your retail company wants to segment its customers based on their order history. You need to identify customer groups based on the total number of orders they've placed and the total order value. This information will help you tailor marketing and promotional strategies to different customer segments. Your task is to create an SQL query that segments customers into different groups based on their order history. Specifically, you want to categorize customers into three groups:

a. **High-Value Customers:** Customers with a total order value greater than 5,000.

b. **Regular Customers:** Customers with a total order value between 1,000 and 5,000.

c. **Occasional Customers:** Customers with a total order value less than 1,000. CLICK TO SHOW ANSWER Q138.

Possible Database Tables and Columns required:

- Person.Address Table:

  Columns: AddressID , City, StateProvinceID, PostalCode
- Person.BusinessEntityAddress Table:

  Columns: BusinessEntityID, AddressID, AddressTypeID

- Sales.SalesOrderHeader Table:

  Columns: SalesOrderID, CustomerID, ShipToAddressID

139. Business scenario Q139

Vendor and Store Analysis

Your retail company wants to perform an analysis of vendors and their relationships with various stores. Your task is to create an SQL query that analyses the relationships between vendors and stores.

a. Identify the top five vendors who supply the most products to stores.

b. Determine the stores with the highest number of vendor relationships.

c. Calculate the average number of vendor relationships per store. CLICK TO SHOW ANSWER Q139.

Possible database tables and columns required:

- Person.BusinessEntity Table:

  Columns: BusinessEntityID, BusinessEntityName

- Person.Person Table:

  Columns: BusinessEntityID, PersonType, FirstName, LastName

- Purchasing.Vendor Table:

  Columns: BusinessEntityID, AccountNumber

- Sales.Store Table:

  Columns: BusinessEntityID, Name

140. Business scenario Q140

Currency and Sales Territory Analysis

In this business scenario, you are tasked with analysing the relationships between country regions, states, currencies, and sales territories. The goal is to gain insights into how currency and sales territories are associated with specific regions and states. The task is to Identify distinct country regions and their associated states and to determine the currencies used in each country region and sales territory. CLICK TO SHOW ANSWER Q140.

Possible database tables required:

- Person.CountryRegion
- Person.StateProvince
- Sales.CountryRegionCurrency
- Sales.SalesTerritory

141. Business scenario Q141

Password Complexity Assessment

In this business scenario, you need to assess the complexity of passwords used by individuals. The objective is to identify patterns or weaknesses in password creation to improve security measures. The Person.Password and Person.Person tables. CLICK TO SHOW ANSWER Q141.

142. Business scenario Q142

Employee and Customer Contact Information

In this business scenario, you are responsible for managing and consolidating contact information for employees and customers. Your goal is to retrieve a comprehensive list of contact details, i.e phone numbers, for both employees and customers. This information will be used for communication and customer support purposes. CLICK TO SHOW ANSWER Q142.

Possible Database Tables required:

- Person.Person
- HumanResources.Employee
- Sales.Customer
- Person.PersonPhone

143. Business scenario Q143

Sales Tax Rates by State

In this business scenario, you are tasked with calculating and organising the sales tax rates for different states. The goal is to determine the applicable sales tax rate for each state and City, which is crucial for accurate pricing and tax calculations for sales transactions. Use the Person.StateProvince and the Sales.SalesTaxRate tables. CLICK TO SHOW ANSWER Q143.

144. Business scenario Q144

Employee Department History Tracking

In this business scenario, you are responsible for tracking the history of employee department assignments. The company needs to keep a record of employee department changes to maintain an accurate historical log of department assignments. Use both HumanResources.EmployeeDepartmentHistory and HumanResources.Employee tables. CLICK TO SHOW ANSWER Q144.

145. Business scenario Q145

Job Candidate Evaluation

In this business scenario, you are tasked with evaluating job candidates who have applied for positions at a company. HR needs to determine which candidates meet the criteria for certain job roles based on their qualifications and experience. Your task is to Evaluate job candidates for specific roles. Use the HumanResources.JobCandidate and HumanResources.Employee tables. CLICK TO SHOW ANSWER Q145.

146. Business scenario Q146

Document Usage Analysis

A manufacturing company wants to analyse the usage of its documents across different products. They have a table that records document associations with products. The company needs to determine which documents are associated with the most products and which products have the highest number of associated documents. Your task is to analyse document usage across products. Use the Production.Document and Production.ProductDocument tables. CLICK TO SHOW ANSWER Q146.

147. Business scenario Q147

Analysing Product Details and Unit Measures

Your company wants to analyse product details along with their corresponding unit measures and subcategories. This analysis will help in understanding which product subcategories are most common, the number of products in each subcategory, and the unit of measure used for these products. You should provide a report that summarises (centimetres, weight, etc) this information. CLICK TO SHOW ANSWER Q147.

Possible Database Tables required:

- Production.Product
- Production.ProductSubcategory
- Production.ProductCategory
- Production.ProductModel
- Production.ProductProductPhoto
- Production.UnitMeasure

148. Business scenario 148

Product and Subcategory Analysis

Your company is interested in gaining insights into the products available. You want to understand the distribution of products across different subcategories, their unit measures, and associated costs. Additionally, you need to identify which subcategories have the highest and lowest product counts and the most expensive products. CLICK TO SHOW ANSWER Q148.

Possible database tables required:

- Production.Product
- Production.ProductSubcategory
- Production.ProductCategory
- Production.UnitMeasure

149. Business scenario Q149

Product and Subcategory Inventory Analysis

Your company wants to analyse the inventory of products within different subcategories. The goal is to understand the quantity of each product, the unit of measure, and the total inventory value for each subcategory. Additionally, you need to identify the subcategories with the highest and lowest total inventory values. CLICK TO SHOW ANSWER Q149.

Possible Database tables required:

- Production.Product
- Production.ProductSubcategory
- Production.ProductCategory
- Production.UnitMeasure

150. Business scenario Q150

Product Sales Analysis by Subcategory

Your company wants to analyse product sales within different subcategories. The goal is to calculate the total sales amount, average sales price, and the number of units sold for each product subcategory. Additionally, you need to identify the subcategories with the highest and lowest total sales amounts. CLICK TO SHOW ANSWER Q150.

Possible Database tables required:

- Production.Product
- Production.ProductSubcategory
- Production.ProductCategory
- Sales.SalesOrderDetail
- Sales.SalesOrderHeader

151. Business scenario Q151

Tracking Work Order Progress

In this business scenario, we need to track the progress of work orders through various stages of manufacturing, including information about which machines and manufacturing locations they are routed to. This helps in monitoring and optimising the manufacturing process. The task is to Track Work Order Progress (Scheduled start, actual start, scheduled end, actual end, etc). CLICK TO SHOW ANSWER Q151.

Possible Database Tables required:

- Production.WorkOrder
- Production.WorkOrderRouting
- Production.Location

152. Business scenario Q152

Tracking Sales Quota History

In this business scenario, we need to track the historical changes in sales quotas for salespeople over time. This information is crucial for understanding the performance and growth of individual salespeople within the organisation. The task requires tracking Sales Quota History. Use the Sales.SalesPersonQuotaHistory and Sales.SalesPerson tables. CLICK TO SHOW ANSWER Q152.

153. Business scenario Q153

Identifying Popular Products

In this business scenario, we aim to identify the most popular products based on the number of times they have been added to shopping carts. This analysis can help in making decisions about inventory management, promotions, and marketing strategies. CLICK TO SHOW ANSWER Q153.

Possible Database Tables required:

- Production.Product
- Sales.ShoppingCartItem

Database: Finance and Banking

154. Business scenario Q154

Optimising Cross-Selling in a Banking Environment

In a banking institution, there is a need to optimise cross-selling of financial products to its customers. The bank wants to identify opportunities for cross-selling financial products to its customers based on their account types and balances. Specifically, they want to find customers who have a certain type of account (e.g., savings account) with a balance above 5000 and offer them a related financial product (e.g., a Certificate of Deposit or Investment Plan). Use the Finance.UniqueCustomers and Finance.UniqueAccounts tables. CLICK TO SHOW ANSWER Q154.

155. Business scenario Q155

Fraud Detection and Prevention in Banking

In a banking institution, there is a pressing need to detect and prevent fraudulent activities within customer accounts. The bank wants to proactively identify potential instances of fraud by monitoring account transactions and identifying suspicious patterns. Specifically, they want to find accounts where the balance suddenly drops significantly or where there are multiple large withdrawals in a short period of time. The following instances will be analysed:

Balance < 0) -- Detect negative balances, which could indicate overdraft or unauthorized transactions.

OR

(Amount < 1000) -- Detect large withdrawals (you can adjust the threshold)

OR

(Amount > 5000); -- Detect large deposits (you can adjust the threshold)

Use Finance.UniqueCustomers, Finance.UniqueAccounts and Finance.UniqueTransations tables. CLICK TO SHOW ANSWER Q155.

156. Business scenario Q156

Customer Segmentation for Targeted Marketing

A bank wants to improve its marketing efforts by segmenting its customers based on their account behaviour and demographics. The bank aims to create targeted marketing campaigns tailored to specific customer segments. They want to categorise customers based on their average account balances. The segments are defined as follows:

- "High-Value Customer" if the average balance is greater than or equal to 5,000.
- "Medium-Value Customer" if the average balance is between 1,000 and 5,000.

- "Low-Value Customer" if the average balance is below 1,000.

Use the Finance.UniqueCustomers and Finance.Accounts tables. CLICK TO SHOW ANSWER Q156.

157. Business scenario Q157

Account Type Analysis

A bank wants to analyse the distribution of different account types among its customers and determine the average balance for each account type. This analysis will help the bank understand the popularity of different account types and their average account balances. CLICK TO SHOW ANSWER Q157.

158. Business scenario Q158

Overdraft Alert

A bank wants to identify accounts that are at risk of overdrawing, i.e., having a negative balance. The bank wants to notify customers whose account balances are close to zero to prevent overdraft fees and ensure better customer satisfaction. Specifically, the bank wants to find accounts with balances below 1000. Use the Finance.UniqueAccounts table and Finance.UniqueCustomers table. CLICK TO SHOW ANSWER Q158.

159. Business scenario Q159

Inactive Account Analysis

A bank wants to identify and analyse inactive accounts to improve customer engagement and reduce account maintenance costs. An account is considered inactive if there have been no transactions (deposits or withdrawals) in the last 48 months. The bank aims to identify these inactive accounts and notify customers to encourage account activity. Use the

Finance.UniqueAccounts table and Finance.UniqueCustomers table and the Finance.UniqueTansactions table. CLICK TO SHOW ANSWER Q159.

160. Business scenario Q160

Branch Transaction Analysis

A bank wants to analyse transaction patterns at its branches to assess the performance of each branch in terms of transaction volume. The bank aims to identify branches with the highest and lowest transaction counts for a specified period. The task is to analyse branch transaction counts for the periods between 2022 and 2023. Use the Finance.UniqueAccounts, Finance.UniqueTransactions and Finance.UniqueAccountBranches tables. CLICK TO SHOW ANSWER Q160.

161. Business scenario Q161

Account Branch Analysis

A bank wants to analyse the distribution of its customer accounts across its branches to assess branch popularity and account allocation. The bank aims to identify branches with the highest and lowest numbers of associated accounts and understand whether certain branches attract more customers. Use the Finance.UniqueAccounts, Finance.UniqueTransactions and Finance.UniqueAccountBranches tables. CLICK TO SHOW ANSWER Q161.

162. Business scenario Q162

Branch-wise Account Balance Summary

A bank wants to generate a summary of account balances for each branch to assess the financial performance of its branches. The bank aims to determine the total balance of accounts associated with each branch. Use the Finance.UniqueAccounts, Finance.UniqueTransactions and Finance.UniqueAccountBranches tables. CLICK TO SHOW ANSWER Q162.

Database: Healthcare

163. Business scenario Q163

Doctor-Patient Appointment Tracking

A healthcare facility wants to track and manage appointments between patients and doctors efficiently. The facility aims to create a system that records appointment details, including patient information, doctor information, appointment date and time, and the purpose of the appointment. Use the Medical.Appointments, Medical.Patients and the Medical.Doctors tables. CLICK TO SHOW ANSWER Q163.

164. Business scenario Q164

Medication Prescription and Management

A healthcare facility wants to manage medication prescriptions for patients efficiently. The facility aims to create a system that records medication details, including medication names, dosages, prescribing doctors, prescription dates, and expiration dates. This system will help in tracking medication usage and ensuring patients receive the right medications at the right time. Use the Medical.Appointments, Medical.Patients, Medical.Doctors and Medical.Medications tables. CLICK TO SHOW ANSWER Q164.

165. Business scenario Q165

Patient Prescription History Management

A healthcare facility aims to efficiently manage and track the prescription history of patients. Patients should be able to access their prescription history, including prescribed medications, dosages, prescribing doctors, prescription dates, and medication expiration dates. Doctors should have access to their patients' prescription histories to make informed medical decisions. Use the Medical.Patients, Medical.Medications and Medical.Doctors tables. CLICK TO SHOW ANSWER Q165.

166. Business scenario Q166

Benefitting from New Medication

A pharmaceutical company is developing a new medication called Surgeonix to treat a specific medical condition. They want to know which patients in their database are most likely to benefit from this medication. Use the Medical.Patients and Medical.Medications tables. CLICK TO SHOW ANSWER Q166.

167. Business scenario Q167

Commonly prescribed Age-group based medication.

A health insurance company wants to know which medications are most commonly prescribed to patients under a certain age group. The task is to find the medications that are most commonly prescribed to patients under the age of 18. Use the Medical.Medications and Medical.Patients tables. CLICK TO SHOW ANSWER Q167.

168. Business scenario Q168

Determine which Doctor is prescribing medicine for certain purpose of hospital visits.

A hospital wants to know which doctors are prescribing the most medications to patients with a certain purpose of hospital visits. The task is to find the doctors who are prescribing the most medications to patients with 'Child infections'. Use the Medical.Medications and Medical.Appointments tables. CLICK TO SHOW ANSWER Q168.

169. Business scenario Q169.

Determine which Doctor is prescribing a particular medicine.

A pharmaceutical company wants to know which doctors are prescribing their new medication, called " Meditrex ", the most often. Use the Medical.Medications table. CLICK TO SHOW ANSWER Q169.

170. Business scenario Q170

Determine Average Prescription per Doctor.

Find all doctors who have prescribed more than the average number of medications to their patients. Use the Medical.Doctors, Medical.Appointments and Medical.Medications tables. CLICK TO SHOW ANSWER Q170.

171. Business scenario Q171

Determine Hospital Visits.

A hospital administrator wants to see a list of all patients who have seen a doctor in the past year 2023. Use the Medical.Patients and Medical.Appointments tables. CLICK TO SHOW ANSWER Q171.

172. Business scenario Q172.

Determine Hospital Visits for specific purpose.

A hospital administrator wants to see a list of all patients who have seen a doctor for the purpose of Joint pain in the past two years. Use the Medical.Patients and Medical.Appointments tables. CLICK TO SHOW ANSWER Q172.

173. Business scenario Q173

Sales Representative Assignment

A hospital administrator wants to see a list of all patients who have been prescribed a certain medication and have seen a doctor for the same purpose in the past year. This problem is relevant to healthcare because it can help hospital administrators to identify patients who are at risk of not taking their medication as prescribed. This can be important for ensuring that patients receive the care they need and preventing medication errors. The medication name is 'PainAway' the Purpose of visit is 'Child infections' and the appointment should be within 10 years. Use the Medical.Patients, Medical.Medications, Medical.Appointments and MedicationName tables. CLICK TO SHOW ANSWER Q173.

174. Business scenario Q174

Appointment Scheduling Efficiency

In this business case, we aim to improve the efficiency of appointment scheduling for doctors by identifying the busiest appointment days and times. By doing so, healthcare providers can optimise doctor schedules and enhance patient experience. Use the Medical.Appointments table. CLICK TO SHOW ANSWER Q174.

175. Business scenario Q175

Medication Expiry Monitoring

In this business scenario, healthcare providers aim to enhance medication safety by proactively identifying medications with expiry dates in the next one year. This allows for timely renewal or replacement of medications to avoid patient health risks. Use the Medical.Medications, Medical.Doctors and Medical.Patients tables. CLICK TO SHOW ANSWER Q175.

176. Business scenario- Q176

Doctor-Patient Appointment Analysis

In this business scenario, a healthcare facility seeks to optimise its appointment scheduling by analysing the distribution of appointments across various doctors and days of the week. The goal is to ensure efficient resource allocation and patient satisfaction. CLICK TO SHOW ANSWER Q176.

Database: Adventureworks Warehouse (Version 2022)

177. Business scenario Q177

Sales Representative Performance Analysis for a Specific Year

A retail company wants to analyse the performance of its sales representatives by calculating the total sales amount generated by each salesperson within year 2012. This analysis will help identify the top-performing sales representatives. Use the FactResellerSales and DimEmployee tables. CLICK TO SHOW ANSWER Q177.

178. Business scenario Q178

Yearly Sales Performance Analysis

A company wants to perform a yearly sales performance analysis for its products and needs to calculate the total sales for each year. This analysis will help the company identify trends and patterns in sales over the years. Use the FactInternetSales table. CLICK TO SHOW ANSWER Q178.

179. Business scenario Q179

Customer Sales Analysis

A retail company wants to analyse customer sales data from its Data Warehousing database to identify their top-performing customers in terms of sales revenue. The task is to match customer identity with the total sales amount. Use the dbo.FactInternetSales and dbo.DimCustomer tables. CLICK TO SHOW ANSWER Q179.

Database: Wideworldimporters

180. Business scenario Q180

Sales Representative Assignment

A company has hired a new sales representative and needs to assign them to a specific city. To do this, they want to find a city with the highest number of potential customers. Your task is to find the city with the highest number of potential customers. Use the Application.Cities and Sales.Customers tables. CLICK TO SHOW ANSWER Q180.

181. Business scenario Q181

Customer Location Analysis

A company wants to analyse the distribution of its customers across different state provinces to identify regions with high and low customer density. This information will help them make informed decisions regarding marketing strategies and resource allocation. Use the Application.StateProvinces, Application.Cities and Sales.Customers tables. CLICK TO SHOW ANSWER Q181.

182. Business scenario Q182

Delivery Method Contact List

A company needs a simple contact list of their delivery method contacts. This list should include the names and email addresses of individuals associated with each delivery method. Use the Application.DeliveryMethods and Application.People tables. CLICK TO SHOW ANSWER Q182.

183. Business scenario Q183

Supplier Performance Analysis

A company wants to analyse the performance of its suppliers. They need a report that shows the total number of purchase orders and the total amount spent with each supplier, as well as the number of invoices received from customers for products supplied by each supplier. Use the Purchasing.Suppliers, Purchasing.PurchaseOrders and Sales.Invoices tables. CLICK TO SHOW ANSWER Q183.

184. Business scenario Q184

Transaction Analysis by Payment Method

A company wants to analyse their supplier and customer transactions based on payment methods. They need to understand how different payment methods are used for both purchasing from suppliers and receiving payments from customers. This analysis will help them make informed decisions regarding their payment processes and financial management. Use the Application.PaymentMethods, Purchasing.SupplierTransactions, and Sales.CustomerTransactions tables. CLICK TO SHOW ANSWER Q184.

185. Business scenario Q185

Payment Trends Analysis

A company wants to analyse payment trends for their supplier and customer transactions. They aim to identify patterns in payment methods used, such as credit cards, electronic transfers, or checks, to better understand their financial operations. This analysis will assist them in optimising their payment processes and improving financial decision-making. Use the Application.PaymentMethods, Purchasing.SupplierTransactions, and Sales.CustomerTransactions tables. CLICK TO SHOW ANSWER Q185.

186. Business scenario Q186

Stock Item Colour Analysis

A company wants to analyse the distribution of stock items by colour to make informed inventory decisions. They need to know how many stock items are available in each colour category. Use the Warehouse.Colors and Warehouse.StockItems tables. CLICK TO SHOW ANSWER Q186.

187. Business scenario Q187

Stock Item Colour Distribution Analysis

A company wants to analyse the distribution of stock items by colour in their inventory. They need to know the percentage of stock items available in each colour category. Use the Warehouse.Colors and Warehouse.StockItems tables. CLICK TO SHOW ANSWER Q187.

188. Business scenario Q188

Average Purchase Order Line Quantity by Package Type

A company wants to calculate the average quantity of items ordered in purchase order lines for each package type. Understanding the average order line quantity by package type can help them optimize their purchasing and inventory management strategies. Use the Warehouse.PackageTypes, Warehouse.StockItems and Purchasing.PurchaseOrderLines tables. CLICK TO SHOW ANSWER Q188.

189. Business scenario Q189

Package Quantity Comparison

A company wants to compare the quantities of items ordered in purchase orders with the quantities sold in sales invoices for each package type. This analysis helps them understand how their purchasing aligns with sales for different package types. Use Warehouse.PackageTypes, Warehouse.StockItems, Purchasing.PurchaseOrderLines and Sales.InvoiceLines tables. CLICK TO SHOW ANSWER Q189.

190. Business scenario Q190

Stock Group Analysis

A company wants to analyse the stock items and stock groups to understand how they are associated with people or employees. They want to know which stock items belong to specific stock groups and which people are associated with those stock groups. Use Warehouse.StockItemStockGroups, Warehouse.StockGroups, Warehouse.StockItems and Application.People tables. CLICK TO SHOW ANSWER Q190.

191. Business scenario Q191

Stock Group Assignment Audit

A company needs to audit stock item assignments to stock groups. They want to review the assignment history, including who performed each assignment and when. Use Warehouse.StockItemStockGroups, Warehouse.StockGroups, Warehouse.StockItems and Application.People tables. CLICK TO SHOW ANSWER Q191.

PART 3

Advanced Level Exercises

Database: AdventureWorks (2019 or 2022)

192. Business scenario Q192

Demand Forecasting

A manufacturing company wants to forecast the demand for its products to optimise production and inventory management. They aim to predict future product demand accurately to ensure that they have the right amount of inventory on hand, minimise excess stock, and meet customer demand efficiently. They want to forecast demand for products to optimise production and inventory management by exploring historical behaviour going back 20 years which will help in calculating monthly demand forecasts. CLICK TO SHOW ANSWER Q192.

Possible database tables and columns required:

- Table: Production.Product

  Columns: ProductID, Name
- Table: Sales.SalesOrderHeader

  Columns: SalesOrderID, OrderDate, CustomerID
- Table: Sales.SalesOrderDetail

  Columns: SalesOrderDetailID, SalesOrderID, ProductID, OrderQty, LineTotal

193. Business scenario Q193

Product Line Expansion

A company considering product line expansion wants to evaluate the potential profitability of expanding or diversifying its existing product line. They aim to assess the financial viability, market demand, and potential risks associated with introducing new products or product variations. The task is to evaluate potential profitability of product line expansion and calculate profitability metrics (Total revenue, gross profit, gross margin percentage). CLICK TO SHOW ANSWER Q193.

Possible database Tables and Columns required:

- Table: Production.Product

  Columns: ProductID, Name, StandardCost, ListPrice

- Table: Sales.SalesOrderDetail

  Columns: SalesOrderDetailID, SalesOrderID, ProductID, OrderQty, LineTotal

194. Business scenario Q194

Product Return Analysis

A retail company aims to reduce product return rates by analysing product return data. They want to identify common reasons for returns and implement strategies to improve product quality, customer satisfaction, and overall return rates. The task is to analyse product return data to identify common reasons for returns up to year 2014 with the order date stretching back to 2009 and to further retrieve product return analysis results. CLICK TO SHOW ANSWER Q194.

Possible Database Tables and Columns required.

- Table: Sales.SalesOrderHeader

  Columns: SalesOrderID, OrderDate, CustomerID

- Table: Sales.SalesOrderDetail

  Columns: SalesOrderDetailID, SalesOrderID, ProductID, OrderQty, LineTotal

- Table: Production.Product

  Columns: ProductID, Name

- Table: Production.ProductReview

  Columns: ProductReviewID, ProductID, ReviewerName, Rating, Comments, ReviewDate

195. Business scenario Q195

E-commerce Conversion Rate Optimisation

An e-commerce company aims to optimise its website's conversion rate to enhance the shopping experience and ultimately increase sales. They want to analyse conversion rates for the last 20 years to identify areas for improvement, retrieve website conversion analysis results and implement strategies to convert more visitors into customers. Use Sales.SalesOrderHeader and Sales.SalesPerson tables. CLICK TO SHOW ANSWER Q195.

196. Business scenario Q196

Sales Forecasting

A retail company wants to predict future sales based on historical data and market trends. They aim to implement a sales forecasting system that can provide accurate sales predictions, allowing for better inventory planning and resource allocation. This analysis will help optimise stock levels, meet customer demand, and allocate resources efficiently. The want to predict future sales using historical data for the last 10 years as well as retrieve sales forecast for the 2 years that follows. Use the Sales.SalesOrderHeader table. CLICK TO SHOW ANSWER Q196.

197. Business scenario Q197

Product Quality Analysis

A manufacturing company wants to monitor and analyse product quality data to ensure that its products meet or exceed quality standards. They aim to implement a comprehensive quality analysis system that can help identify issues early in the manufacturing process and continuously improve product quality. The tasks are to analyse product quality based on customer reviews and retrieve product quality analysis results. CLICK TO SHOW ANSWER Q197.

Possible Database Tables and Columns required:

- Table: Production.Product

  Columns: ProductID, Name, ProductNumber

- Table: Production.ProductReview

  Columns: ProductReviewID, ProductID, ReviewerName, Rating, Comments, ReviewDate

198. Business scenario Q198

Customer Retention Analysis

A retail company wants to understand and improve customer retention rates. They aim to analyse customer churn, identify factors that influence customer retention, and develop strategies to reduce customer churn. This analysis will help build stronger customer relationships and increase customer loyalty. The tasks are to analyse customer retention rates and factors influencing retention and to calculate customer retention rates and analyse factors. CLICK TO SHOW ANSWER Q198.

Possible Database Tables and Columns required:

- Table: Sales.Customer

  Columns: CustomerID, FirstName, LastName

- Table: Sales.SalesOrderHeader

  Columns: SalesOrderID, CustomerID, OrderDate

199. Business scenario Q199

Promotion Effectiveness Analysis

A retail company wants to assess the effectiveness of its promotional campaigns. They aim to evaluate the impact of various promotional efforts on sales and customer acquisition. This analysis will help optimise marketing spend by identifying which promotions are the most successful in driving revenue and acquiring new customers. The tasks are to evaluate the impact of promotional campaigns on sales and customer acquisition and further retrieve promotion effectiveness information. CLICK TO SHOW ANSWER Q199.

Possible Database Tables and Columns required:

- Table: Sales.SpecialOffer

  Columns: SpecialOfferID, Description, DiscountPct
- Table: Sales.SalesOrderHeader

  Columns: SalesOrderID, CustomerID, OrderDate, TotalDue

200. Business scenario Q200

Customer Lifetime Value (CLV) Calculation

A retail company wants to assess the value of each customer over their lifetime of engagement with the company. They aim to calculate the Customer Lifetime Value (CLV) for each customer to identify high-value customers and tailor marketing efforts, loyalty programs, and customer service strategies accordingly. The tasks are to calculate Customer Lifetime Value (CLV) for each customer and retrieve Customer Lifetime Value (CLV) information. CLICK TO SHOW ANSWER Q200.

Possible Database Tables and Columns required:

- Table: Sales.Customer

  Columns: CustomerID, FirstName, LastName
- Table: Sales.SalesOrderHeader

  Columns: SalesOrderID, CustomerID, OrderDate, TotalDue

201. Business scenario Q201

Product Life Cycle Analysis

A retail company wants to assess the life cycle stages of various products in its catalogue. They aim to determine whether each product is in the introduction, growth, maturity, or decline stage of its life cycle. This analysis will inform inventory management decisions and help shape marketing strategies for each product. The required tasks are to determine the life cycle stage of products and retrieve product life cycle information. CLICK TO SHOW ANSWER Q201.

Possible Database Tables and Columns required:

- Table: Production.Product

  Columns: ProductID, Name, ProductNumber, StandardCost, ListPrice, SellStartDate, DiscontinuedDate

202. Business scenario Q202

Employee Productivity Ranking

A company with a sales team and employees in various roles, wants to assess and rank employee productivity. They aim to rank employees based on specific productivity metrics, such as total sales generated or total hours worked, to identify high-performing employees and recognize their contributions. The task is to rank employees based on their productivity (e.g., total sales generated) and further rank employees by productivity in descending order. CLICK TO SHOW ANSWER Q202.

Possible Database Tables and Columns required:

- Table: Sales.SalesPerson

  Columns: BusinessEntityID (corresponds to EmployeeID), FirstName, LastName

- Table: Sales.SalesOrderHeader

  Columns: SalesOrderID, SalesPersonID

- Table: HumanResources.Employee

  Columns: BusinessEntityID (corresponds to EmployeeID), FirstName, LastName

- Table: HumanResources.EmployeePayHistory

  Columns: BusinessEntityID (corresponds to EmployeeID), RateChangeDate, Rate

203. Business scenario Q203

Customer Address Validation for Shipping Efficiency

Your company, a retailer has been facing issues with shipping efficiency due to incorrect customer addresses. This problem has resulted in increased shipping costs and customer dissatisfaction. To address this issue, you want to validate and correct customer addresses based on available data in the database. Your task is to create an SQL query that identifies customers with potentially incorrect addresses by comparing their "ShipToAddress" with the address data in the "Person.Address" table. The query should return a list of customers with potentially incorrect addresses, along with the order count for each customer. CLICK TO SHOW ANSWER Q203.

Possible Database Tables and Columns required:

- Person.Address Table:

  Columns: AddressID, City, StateProvinceID, PostalCode

- Person.BusinessEntityAddress Table:

  Columns: BusinessEntityID, AddressID, AddressTypeID

- Sales.SalesOrderHeader Table:

  Columns: SalesOrderID, CustomerID, ShipToAddressID

204. Business scenario Q204

Sales Region Analysis for Targeted Marketing

Your company, a retailer wants to improve its marketing strategies by identifying regions with the highest sales. This information will help you allocate marketing resources effectively. Your task is to create an SQL query that categorizes regions (based on the "StateProvinceID" from the "Person.Address" table) by their total sales amount. The query should return a list of regions, their total sales amounts, and the number of sales orders placed from highest to lowest sales. CLICK TO SHOW ANSWER Q204.

Possible Database Tables and Columns required:

- Person.Address Table:

  Columns: AddressID, City, StateProvinceID, PostalCode
- Person.BusinessEntityAddress Table:

  Columns: BusinessEntityID, AddressID, AddressTypeID
- Sales.SalesOrderHeader Table:

  Columns: SalesOrderID, CustomerID, ShipToAddressID

205. Business scenario Q205

Department Employee History Analysis

In this business scenario, you are tasked with analysing the history of employee department assignments to gain insights into department changes and trends within the organisation. Your analysis will help HR and management make informed decisions about workforce management and departmental restructuring. Use both HumanResources.EmployeeDepartmentHistory and HumanResources.Department tables. CLICK TO SHOW ANSWER Q205.

206. Business scenario Q206

Department Change Analysis

In this business scenario, you are tasked with analysing the history of department changes for employees in an organisation. The HR department wants to understand how frequently employees change departments and identify any trends or patterns in these changes. Use both HumanResources.EmployeeDepartmentHistory and HumanResources.Department tables. CLICK TO SHOW ANSWER Q206.

207. Business scenario Q207

Employee Department Tenure

In this business scenario, you are tasked with calculating the average tenure of employees in each department of an organisation. The HR department wants to understand how long, on average, employees have been working in their respective departments. Use both HumanResources.EmployeeDepartmentHistory and HumanResources.Department tables. CLICK TO SHOW ANSWER Q207.

208. Business scenario Q208

Employee Shift Analysis

In this business scenario, you are tasked with analysing the shift preferences of employees in an organisation. The HR department wants to know which shifts are the most preferred among employees and the distribution of employees across different shifts. Use the HumanResources.EmployeeDepartmentHistory and HumanResources.Shift tables. CLICK TO SHOW ANSWER Q208.

209. Business scenario Q209

Product Model Localisation Check

In this business scenario, a company with a global presence, wants to ensure that its product descriptions are correctly localised for different cultures and languages. They have a Product Model that contains descriptions, and they want to check which cultures are missing localised descriptions for specific product models. This will help them identify gaps in their localization efforts. Your task is to check which cultures are missing localised descriptions for product models. Use Production.Culture, Production.ProductModelProductDescriptionCulture and Production.ProductModel tables. CLICK TO SHOW ANSWER Q209.

210. Business scenario Q210

Inventory Valuation and Usage Analysis

Your company needs to perform an analysis of inventory valuation and usage for its products. You are tasked with calculating the total inventory valuation, the total quantity of products used in production, and the total quantity of products sold for each product. This analysis will help in optimising inventory levels and understanding the cost and usage patterns of different products. CLICK TO SHOW ANSWER Q210.

Possible Database table required:

- Production.Product
- Production.BillOfMaterials
- Production.ProductCostHistory
- Production.WorkOrder

211. Business scenario Q211

Cost Analysis and Profitability of Product Models

Your company wants to perform a cost analysis and profitability assessment for different product models. You need to calculate the total cost of manufacturing each product model and compare it with the total revenue generated from selling those products. This analysis will help identify which product models are the most profitable and may guide decisions on pricing and production. CLICK TO SHOW ANSWER Q211.

Possible Database Tables required:

- Production.ProductModel
- Production.Product
- Production.BillOfMaterials
- Production.ProductCostHistory
- Production.TransactionHistory
- Production.ProductListPriceHistory

212. Business scenario Q212

Product Description Localisation Gap Analysis

Your company aims to ensure that product descriptions are effectively localised for various cultures to enhance the customer experience. However, there may be gaps or missing translations in certain cultures. To address this, you need to identify which product descriptions are incomplete or missing for specific cultures. CLICK TO SHOW ANSWER Q212.

Possible Database Tables required:

- Production.ProductModelProductDescriptionCulture
- Production.Culture

- Production.ProductDescription

213. Business scenario Q213

Work Order Scrap Analysis

Your manufacturing company wants to analyse work orders and identify common scrap reasons to improve production efficiency. You need to create a report that lists the most frequent scrap reasons along with the number of work orders associated with each reason. CLICK TO SHOW ANSWER Q213.

Possible Database Tables required:

- Production.ScrapReason
- Production.WorkOrder

214. Business scenario Q214

Work Order Scrap Rate Analysis

Your manufacturing company is concerned about the scrap rates in the production process for different products. You want to assess the scrap rates for each product and identify the primary scrap reasons contributing to higher scrap rates. Create a report that lists products with their scrap rates and the top scrap reasons associated with each product. CLICK TO SHOW ANSWER Q214.

Possible Database Tables required:

- Production.Product
- Production.WorkOrder
- Production.ScrapReason

215. Business scenario Q215

Calculating Total Scrap Costs by Product

In this business scenario, we want to calculate the total scrap costs incurred by each product. The database contains information about work orders, products, and scrap reasons. We need to determine how much each product has cost the company due to scrap. CLICK TO SHOW ANSWER Q215.

Possible database tables required:

- Production.Product
- Production.WorkOrder
- Production.ScrapReason

216. Business scenario Q216

Analysing Scrap Reasons for Production Delays

In this business scenario, we want to analyse the reasons for production delays by examining scrap reasons for work orders. The database contains information about work orders, products, and scrap reasons. We aim to identify the top scrap reasons that have caused production delays and determine their impact. CLICK TO SHOW ANSWER Q216.

Possible Database Tables Required:

- Production.WorkOrder
- Production.ScrapReason

217. Business scenario Q217

Identifying Work Orders with Longest Routing Times

In this business scenario, we want to identify specific work orders that have the longest routing times in the manufacturing process. By analysing the Production.WorkOrder and Production.WorkOrderRouting tables, we can find these work orders, which may require special attention or investigation to optimize the manufacturing workflow. CLICK TO SHOW ANSWER Q217.

218. Business scenario Q218

Analysing Work Order Completion Rates

In this business scenario, we aim to analyse the completion rates of work orders within different manufacturing stages. We want to understand how many work orders are completed successfully and on time at each routing operation. This analysis can help identify bottlenecks and areas for process improvement. The task is to analyse Work Order Completion Rates (Complete on time, completed delayed and total work orders). Use Production.WorkOrder and Production.WorkOrderRouting tables. CLICK TO SHOW ANSWER Q218.

219. Business scenario Q219

Optimising Work Order Routing

In this business scenario, we aim to optimise the routing of work orders by considering the location of manufacturing facilities. We want to ensure that work orders are assigned to the most suitable manufacturing location based on a factor such as proximity. CLICK TO SHOW ANSWER Q219.

Possible Database Tables Required:

- Production.WorkOrderRouting

- Production.Location
- Production.WorkOrder

220. Business scenario Q220

Monitoring Sales Performance Against Quotas

In this business scenario, we need to monitor the performance of salespeople by comparing their actual sales against their assigned sales quotas. This information is crucial for evaluating individual sales performance and identifying whether sales targets are being met. CLICK TO SHOW ANSWER Q220.

Possible database tables required:

- Sales.SalesPerson
- Sales.SalesPersonQuotaHistory
- Sales.SalesOrderHeader

221. Business scenario Q221

Evaluating Sales Quota Achievement

In this business scenario, we want to evaluate the achievement of sales quotas for salespeople over a specified period. This analysis will help identify salespersons who have met or exceeded their assigned quotas and those who may need additional support. CLICK TO SHOW ANSWER Q221

Possible database tables required:

- Sales.SalesPerson
- Sales.SalesPersonQuotaHistory
- Sales.SalesOrderHeader

222. Business scenario Q222

Inventory Replenishment

In this business scenario, we need to determine when and how many units of a product should be replenished in the inventory based on the sales history and current stock levels. This is essential for maintaining optimal inventory levels and avoiding stockouts. The task is to determine inventory replenishment quantity from Total sales and current stock. Use the Production.Product and Sales.ShoppingCartItem tables. CLICK TO SHOW ANSWER Q222.

223. Business scenario Q223

Customer Shopping Behaviour Analysis

In this business scenario, we want to analyse customer shopping behaviour by identifying the top-selling products and their corresponding categories. Understanding which products and categories are most popular can help in tailoring marketing strategies and optimising inventory. CLICK TO SHOW ANSWER Q223.

Possible Database Tables required:

- Production.Product
- Sales.ShoppingCartItem
- Production.ProductCategory
- Production.ProductSubcategory

Database: Wideworldimporters

224. Business scenario Q224

Supplier Evaluation for On-Time Deliveries

A company wants to evaluate its suppliers based on their on-time delivery performance. They need to calculate the percentage of purchase orders that were delivered on time for each supplier and identify suppliers with a consistently high on-time delivery rate. Use the Purchasing.Suppliers, Purchasing.PurchaseOrders and Purchasing.SupplierTransactions tables. CLICK TO SHOW ANSWER Q224

225. Business scenario Q225

Popular Stock Items by Colour

A company wants to identify the most popular stock items for each colour category in their inventory. They need to know which stock items are frequently sold and available in each colour. Use the Warehouse.Colors and Warehouse.StockItems tables. CLICK TO SHOW ANSWER Q225.

226. Business scenario Q226

Average Order Line Amount by Package Type

A company wants to calculate the average order line amount for each package type. They need to understand the typical order line value associated with different packaging options to optimise their inventory and pricing strategies. Your task is to determine the average order line amount for the packaged items. Use Warehouse.PackageTypes, Warehouse.StockItems, and Sales.OrderLines tables. CLICK TO SHOW ANSWER Q226.

227. Business scenario Q227

Top Stock Items Analysis

A company wants to identify the top-selling stock items within each stock group. They also want to calculate the average and total quantities sold for these top items. This analysis will help them focus on the most profitable products within each stock group. Use Warehouse.PackageTypes, Warehouse.StockItems, and Sales.OrderLines tables. CLICK TO SHOW ANSWER Q227.

228. Business scenario Q228.

Stock Group Analysis with Ranking and Growth

A company wants to analyse the performance of stock groups by ranking them based on their total sales quantities. In addition, they want to calculate the sales growth rate for each stock group over time. This analysis will help them identify which stock groups are performing well and which ones need improvement. Use Warehouse.PackageTypes, Warehouse.StockItems, and Sales.OrderLines tables. CLICK TO SHOW ANSWER Q228.

229. Business scenario Q229

Loan Application and Approval

A bank wants to streamline its loan application and approval process for its customers. The bank aims to make the loan application process more efficient for customers and reduce manual work for its staff. They want to automate the loan application and approval process based on predefined criteria, making it faster and more convenient for both customers and the bank. the loan application and approval process based on the following criteria:

- **LoanEligibility:**

- "Eligible" if the customer's savings account balance is 5,000 or more.
- "Not Eligible" for all other cases.

- **LoanStatus:**
  - "Approved" if the customer is eligible for a loan.
  - "Rejected" if the customer is not eligible for a loan.

Use the Finance.UniqueCustomers and Finance.UniqueAccounts tables. CLICK TO SHOW ANSWER Q229.

230. Business scenario Q230

Account Type Distribution Analysis

A bank wants to analyse the distribution of different account types among its customers in order to better understand their preferences and tailor marketing strategies accordingly. The bank aims to determine the percentage of customers who have savings accounts versus checking accounts. Use the Use the Finance.UniqueCustomers and Finance.UniqueAccounts tables. CLICK TO SHOW ANSWER Q230.

231. Business scenario Q231

Yearly Transaction Summary

A bank wants to generate a yearly summary of customer transactions to provide insights into customer behaviour and financial activity. The bank aims to calculate the total transaction count and the sum of transaction amounts for each customer for a specific year. Use the Finance.UniqueCustomers, Finance.UniqueAccounts and Finance.UniqueTransactions tables. CLICK TO SHOW ANSWER Q231.

232. Business scenario Q232

Account Closure Analysis

A bank wants to analyse the reasons for account closures to improve customer retention and service quality. The bank aims to identify accounts that have been closed and determine the primary reasons for closure based on transaction history. The task requires identifying accounts that have been closed based on two primary closure reasons:

1. "No Transactions": Accounts with no transaction history.
2. "Zero Balance": Accounts with a zero balance.

The query should calculate the last transaction date for each account and categorizes accounts into the closure reasons mentioned above.

Use the Finance.UniqueCustomers, Finance.UniqueAccounts and Finance.UniqueTransactions tables. CLICK TO SHOW ANSWER Q232.

233. Business scenario Q233

Yearly Transaction Trend Analysis

A bank wants to analyse transaction trends by year to understand the annual patterns of deposit and withdrawal activities among its customers. The bank aims to identify whether there are specific years when customers tend to make more transactions and if there's any correlation between transaction patterns and account types. Use the Finance.UniqueCustomers, Finance.UniqueAccounts and Finance.UniqueTransactions tables. CLICK TO SHOW ANSWER Q233.

234. Business scenario Q234

Customer Transaction Analysis

A bank wants to analyse the transaction behaviour of its customers to identify their preferences for deposit or withdrawal transactions. The bank aims to determine which transaction type (deposit or withdrawal) is more common among customers and if there's any correlation between transaction behaviour and account balances. Use the Finance.UniqueCustomers, Finance.UniqueAccounts and Finance.UniqueTransactions tables. CLICK TO SHOW ANSWER Q234.

235. Business scenario Q235

Customer Transaction Classification

A bank wants to classify its customers based on their transaction behaviour. The bank aims to categorize customers into different segments (e.g., active, occasional, and inactive) based on the frequency of their transactions and their account balances. They want to categorise customers into different segments based on their transaction behaviour and account balances:

- "Active" customers have at least 10 transactions and an average balance of 5,000 or more.
- "Occasional" customers have at least 5 transactions and an average balance of 1,000 or more.
- "Inactive" customers do not meet the criteria for the "Active" or "Occasional" segments.

Use the Finance.UniqueCustomers, Finance.UniqueAccounts and Finance.UniqueTransactions tables. CLICK TO SHOW ANSWER Q235.

236. Business scenario Q236

Transaction Anomaly Detection

A bank wants to detect unusual or potentially fraudulent transactions within its accounts. The bank aims to identify transactions that deviate significantly from a customer's typical transaction behaviour based on their transaction history. The task is to identify transactions as either "Normal" or "Anomaly Detected" based on the following criteria:

- A transaction is considered an anomaly if its amount deviates significantly from the average amount for that account, where deviation is defined as more than two times the standard deviation.

Use the Finance.UniqueCustomers, Finance.UniqueAccounts and Finance.UniqueTransactions tables. CLICK TO SHOW ANSWER Q236.

237. Business scenario Q237

Branch Transaction Comparison

A bank wants to compare the transaction volumes between its branches for a specific period to identify branches with significant differences in transaction activity. The bank aims to find branches that have notably higher or lower transaction volumes relative to others. The analysis should cover the periods between 2022 and 2023. Use the Finance.UniqueAccounts, Finance.UniqueBranches, and Finance.UniqueTransactions tables. CLICK TO SHOW ANSWER Q237.

238. Business scenario Q238

Customer Branch Preference

A bank wants to understand the preferences of its customers when it comes to branch locations. The bank aims to identify which branches are favoured by specific customer segments based on transaction data. This analysis will help the bank tailor its marketing and service offerings. Use the Finance.UniqueCustomers, Finance.UniqueAccounts, Finance.UniqueAccountBranches and Finance.UniqueTransactions tables. CLICK TO SHOW ANSWER Q238.

239. Business scenario Q239

Branch Profitability Analysis

A bank wants to assess the profitability of its branches based on the total transaction amounts and account balances. The bank aims to identify branches that generate the highest and lowest profits by considering both transaction revenue and account balance fees. The task is to determine transaction revenue, account balance fees and total profit. Use the Finance.UniqueAccounts, Finance.UniqueAccountBranches and Finance.UniqueTransactions tables. CLICK TO SHOW ANSWER Q239.

Database: Healthcare

240. Business scenario Q240

The Prescription Adherence Problem

A healthcare organization wants to improve the prescription adherence rate of its patients. Prescription adherence is the extent to which a patient takes their medication as prescribed by their doctor. Low prescription adherence can lead to poor health outcomes, such as increased risk of hospitalization and death. The task is to identify patients who are at risk of poor prescription adherence. Use the Medical.Patients, Medical.Medications and Medical.Appointments tables. CLICK TO SHOW ANSWER Q240.

241. Business scenario Q241.

Optimising Medication Prescriptions in Healthcare

In a healthcare system that utilises electronic medical records (EMR), there's a growing concern about optimising medication prescriptions to ensure patient safety, reduce medication errors, and improve overall healthcare quality. The objective is to identify potential issues with medication prescriptions, monitor prescription patterns, and enhance communication between patients and healthcare providers. The task involves the design of an SQL query to identify all instances where patients have been prescribed medications by doctors within the healthcare system, focusing on cases where medication prescriptions are nearing or have exceeded their expiry dates. Additionally, generate a report highlighting doctors who have prescribed medications that are commonly associated with adverse reactions or contraindications for patients based on their medical history. The specific task would be to identify prescriptions nearing or exceeding expiry dates in the nest 6 months. Identify doctors prescribing medications with potential contraindications. Use the Medical.Medications, Medical.Patients and Medical.Doctors tables. CLICK TO SHOW ANSWER Q241.

242. Business scenario Q242

Doctor Appointment Utilisation Analysis

In this business scenario, a healthcare facility aims to optimise the scheduling of doctor appointments to maximise doctor availability and minimise patient waiting times. The goal is to analyse appointment utilization patterns and identify opportunities for improved appointment scheduling. The task is to query total appointments, last appointment date and days since last appointment. Use the Medical.Doctors and Medical.Appointments tables. CLICK TO SHOW ANSWER Q242.

Database: Adventureworks Warehouse (Version 2022)

243. Business scenario Q243

Customer Loyalty Analysis (Simplified)

In this simplified business scenario, a retail company wants to analyse customer loyalty based on their purchasing behaviour and demographics. The goal is to identify loyal customers and understand their characteristics to develop targeted marketing strategies. The task involves determination of customer age group and loyalty status. Use the DimCustomer and FactInternetSales tables. CLICK TO SHOW ANSWER Q243.

244. Business scenario Q244

Customer Sales Analysis

A company wants to analyse its customer sales data to gain insights into customer purchasing behaviour. Specifically, they want to identify the top-spending customers and categorise them based on their purchase history. The code should determine total sales amount and Customer Category. Use the FactInternetSales and DimCustomer tables. CLICK TO SHOW ANSWER Q244.

245. Business scenario Q245

Product Inventory Analysis

A retail company wants to analyse its product inventory data from its Warehousing database to optimise stock levels. Specifically, they want to identify products that are running low on inventory and need to be replenished. Use the FactProductInventory and DimProduct tables. CLICK TO SHOW ANSWER Q245.

246. Business scenario Q246

Employee Performance Evaluation

A human resources department wants to evaluate employee performance based on various factors, including sales and customer satisfaction scores. The task is to determine the performance for year 2012. Use the DimEmployee and FactInternetSales tables. CLICK TO SHOW ANSWER Q246.

247. Business scenario Q247

Product Sales Performance Analysis by Category and Subcategory

A retail company wants to analyse the sales performance of its products based on different product categories and subcategories. They aim to identify which product categories and subcategories are performing exceptionally well and which ones may need additional marketing efforts. Use the FactInternetSales, DimProduct, DimProductSubcategory and DimProductCategory tables. CLICK TO SHOW ANSWER Q247.

248. Business scenario Q248

Product Sales Analysis by Region for a Specific Year

A manufacturing company wants to analyse its product sales across different regions for year 2012. They are interested in finding out which regions had the highest and lowest sales during that year. Use the FactInternetSales and DimGeography tables. CLICK TO SHOW ANSWER Q248.

249. Business scenario Q249

Sales Analysis by Product Category for a Specific Period

A retail company wants to analyse the sales performance of its products in different product categories over the period 2012 and 2013. They aim to identify the top-selling product categories during this period. Use the FactInternetSales, DimProduct, DimProductSubcategory, DimProductCategory and DimDate tables. CLICK TO SHOW ANSWER Q249.

250. Business scenario Q250

Sales Performance Analysis for Top Products in Each Territory in 2012

A retail company wants to analyse the sales performance of its products in 2012 across different sales territories and wants to find the top-performing products in each territory. The task is to work out sales performance analysis for EnglishProduct in each territory in 2012. Use the FactInternetSales, DimProduct, DimSalesTerritory and DimDate tables. CLICK TO SHOW ANSWER Q250.

251. Business scenario Q251

Promotion Analysis

A company is running a promotion called either 'Touring-1000 Promotion' or 'Touring-3000 Promotion' on various products on the English promotion channel. The goal is to analyse the basic performance of this promotion, including the total sales generated during the promotion period. Use the DimPromotion and the FactInternetSales tables. CLICK TO SHOW ANSWER Q251.

SAMPLE SOLUTIONS

```
--Q1. Sample Solution

SELECT *
FROM CustomerTab.CustomerProductTabl

--Back to Business scenario Q1


--Q2. Sample Solution SELECT

DISTINCT (ProductID)
FROM CustomerTab. CustomerOrdersTabl

--Back to Business scenario Q2


--Q3. Sample SolutionSELECT ALL

(ProductID)
FROM CustomerTab. CustomerOrdersTabl

--Back to Business scenario Q3


--Q4. Sample Solution

--a)

SELECT ProductID, ProductDescription, CostFROM
CustomerTab.CustomerProductTabl WHERE Cost < 1;

--b)

SELECT ProductID, ProductDescription, CostFROM
CustomerTab.CustomerProductTabl WHERE Cost
BETWEEN 2 AND 5;

--c)

SELECT ProductID, ProductDescription, CostFROM
CustomerTab.CustomerProductTabl WHERE Cost > 6;

--Back to Business scenario Q4


--Q5. Sample Solution

SELECT ProductID, ProductDescription, CostFROM
CustomerTab.CustomerProductTabl WHERE ProductID
IN (2, 567, 89);

--Back to Business scenario Q5
```

```
--Q6. Sample Solution

SELECT ProductID, ProductDescription, CostFROM
CustomerTab.CustomerProductTabl
WHERE ProductDescription = 'Instant Camera' OR ProductDescription = 'Phone Stand';

--Back to Business scenario Q6


--Q7. Sample Solution

SELECT OrderID, CustomerID, ProductID, OrderQuantity, OrderDateFROM
CustomerTab. CustomerOrdersTabl
WHERE NOT OrderQuantity = 5;

--Back to Business scenario Q7


--Q8. Sample Solution

SELECT ProductID, ProductDescription, CostFROM CustomerTab.CustomerProductTabl WHERE
ProductDescription LIKE '%Set%';

--Back to Business scenario Q8


--Q9. Sample Solution

SELECT JobTitle, EmployeeIDFROM
Employee.Pay
WHERE Salary IS NULL;

--Back to Business scenario Q9


--Q10. Sample Solution

SELECT JobTitle, EmployeeIDFROM
Employee.Pay
WHERE Salary IS NOT NULL;

--Back to Business scenario Q10


--Q11. Sample SolutionSELECT Cost
FROM CustomerTab.CustomerProductTablWHERE
EXISTS (SELECT Cost
FROM CustomerTab.CustomerProductTablWHERE
Cost > 6);

--Back to Business scenario Q11


--Q12. Sample SolutionSELECT *
FROM CustomerTab.CustomerProductTabl
WHERE Cost > ALL (SELECT Cost
            FROM CustomerTab.CustomerProductTablWHERE
            Cost < 4);
```

--Back to Business scenario Q12

```
--Q13. Sample SolutionSELECT *
FROM CustomerTab.CustomerProductTabl
WHERE 4 < ANY (SELECT Cost
             FROM CustomerTab.CustomerProductTabl);
```

--Back to Business scenario Q13

```
--Q14. Sample Solution

SELECT EmployeeID, JobTitle, (Salary + Bonus) AS TotalIncomeFROM
Employee.Pay
WHERE (Salary + Bonus) > 3000
```

--Back to Business scenario Q14

```
--Q15. Sample Solution

SELECT EmployeeID, JobTitle, HourlyRate, (HourlyRate * 100) AS ThresholdFROM
Employee.Pay
WHERE (HourlyRate * 100) > 1500;
```

--Back to Business scenario Q15

```
--Q16. Sample Solution

SELECT ProductID, ProductDescription, CostFROM
CustomerTab.CustomerProductTabl ORDER BY Cost
ASC;
```

--Back to Business scenario Q16

```
--Q17. Sample Solution

SELECT ProductID, ProductDescription, CostFROM
CustomerTab.CustomerProductTabl ORDER BY Cost
DESC;
```

--Back to Business scenario Q17

```
--Q18. Sample Solution

SELECT ProductID, ProductDescription, CostFROM
CustomerTab.CustomerProductTabl WHERE Cost < 6
ORDER BY Cost ASC;
```

--Back to Business scenario Q18

```
--Q19. Sample Solution

SELECT ProductID, ProductDescription, CostFROM
CustomerTab.CustomerProductTabl WHERE Cost > 3
ORDER BY Cost DESC;
```

--Back to Business scenario Q19

```
--Q20. Sample Solution

SELECT ProductID, ProductDescription, CostFROM
CustomerTab.CustomerProductTabl WHERE Cost < 5
ORDER BY ProductDescription ASC;
```

--Back to Business scenario Q20

```
--Q21. Sample Solution

SELECT ProductID, ProductDescription, CostFROM
CustomerTab.CustomerProductTabl WHERE Cost < 5
ORDER BY ProductDescription DESC;
```

--Back to Business scenario Q21

```
--Q22. Sample Solution

SELECT EmployeeID, EmployeeID, JobTitle, SalaryFROM
Employee.Pay
ORDER BY JobTitle ASC, Salary DESC;
```

--Back to Business scenario Q22

```
--Q23. Sample Solution

SELECT EmployeePayID, JobTitle, Hiredate, SalaryFROM
Employee.Pay
WHERE JobTitle IN ('Salesman', 'Shipper')ORDER BY
JobTitle, Salary;
```

--Back to Business scenario Q23

```
--Q24. Sample Solution

SELECT ProductID, ProductDescription, CostFROM
CustomerTab.CustomerProductTabl ORDER BY 2, 3;
```

--Back to Business scenario Q24

```
--Q25. Sample Solution

SELECT ProductID, ProductDescription, Cost
FROM CustomerTab.CustomerProductTabl
```

```
ORDER BY 2 DESC, 3 ASC;

--Back to Business scenario Q25


--Q26. Sample Solution

--a)

SELECT COUNT(EmployeeID) AS EmployeeFROM
Employee.Pay;

--b)

SELECT COUNT (DISTINCT JobTitle)
FROM Employee.Pay;

--c)

SELECT COUNT(Salary) AS TotalSalariesFROM
Employee.Pay;

--d)

SELECT COUNT (*) AS TotalRowsFROM
Employee.Pay;

--Back to Business scenario Q26


--Q27. Sample Solution

SELECT SUM(Salary) AS TotalSalariesFROM
Employee.Pay;

--Back to Business scenario Q27


--Q28. Sample Solution

SELECT AVG(Salary) AS TotalSalariesFROM
Employee.Pay;

--Back to Business scenario Q28


--Q29. Sample Solution

--a)

SELECT MIN(Salary) AS MinimumSalaryFROM
Employee.Pay;

--b)

SELECT MAX(Salary) AS MaximumSalaryFROM
Employee.Pay;
```

--Back to Business scenario Q29

```
--Q30. Sample Solution

-- Calculate total compensation for each employee (Salary + Bonus)

SELECT EmployeeID, JobTitle, Salary, Bonus, (Salary + Bonus) AS TotalCompensationFROM
Employee.Pay;

-- Find the average total compensation for all employees

SELECT AVG(Salary + Bonus) AS AverageTotalCompensationFROM
Employee.Pay;
```

--Back to Business scenario Q30

```
--Q31. Sample Solution

SELECT EmployeeID, City FROM
Employee.Employee GROUP BY
EmployeeID, City;
```

--Back to Business scenario Q31

```
--Q32. Sample Solution

SELECT EmployeeID, SUM(Salary) AS TotalSalaryFROM
Employee.Pay
GROUP BY Salary, EmployeeID;
```

--Back to Business scenario Q32

```
--Q33. Sample Solution

SELECT City, COUNT(*) AS EmployeeCountFROM
Employee.Employee
GROUP BY City;
```

--Back to Business scenario Q33

```
--Q34. Sample Solution

SELECT City, COUNT (*) AS EmployeeCountFROM
Employee.Employee
GROUP BY City
ORDER BY EmployeeCount DESC;
```

--Back to Business scenario Q34

```
--Q35. Sample Solution

SELECT City, COUNT(*) AS EmployeeCountFROM
```

```
Employee.Employee
GROUP BY CityORDER BY
1, 2;

--Back to Business scenario Q35


--Q36. Sample Solution

SELECT EmployeeID, SUM (Salary + Bonus) AS TotalCompensationFROM
Employee.Pay
GROUP BY EmployeeID
HAVING SUM (Salary + Bonus) <= 3000;

--Back to Business scenario Q36


--Q37. Sample Solution

SELECT City, COUNT (*) AS EmployeeCountFROM
Employee.Employee
GROUP BY City
HAVING COUNT (*) <= 3
ORDER BY EmployeeCount DESC;

--Back to Business scenario Q37


--Q38. Sample Solution

SELECT City, COUNT (*) AS EmployeeCountFROM
Employee.Employee
GROUP BY City
HAVING COUNT (*) <= 3ORDER BY
2,1;

--Back to Business scenario Q38


--Q39. Sample Solution

SELECT EmployeeID, CONCAT (FirstName, ' ', LastName)  AS FullNameFROM
Employee.Employee

--Back to Business scenario Q39


--Q40. Sample Solution

SELECT EmployeeID, LastName, FirstName,
    TRANSLATE (Phone, '0123456789', 'xxxxxxxxxx') AS ProtectedPhoneFROM
Employee.Employee;

--Back to Business scenario Q40


--Q41. Sample SolutionSELECT
```

```
EmployeeID,
       UPPER(LastName) AS UppercaseLastName,
       UPPER(FirstName) AS UppercaseFirstName,
       EmployeeAddress, City, PostCode, Phone
FROM Employee.Employee

--Back to Business scenario Q41


--Q42. Sample Solution

SELECT EmployeeID, LastName, FirstName,
       SUBSTRING (Phone, 6, 4) AS ExtractedPhoneSubstringFROM
Employee.Employee;

--Back to Business scenario Q42


--Q43. Sample Solution

SELECT EmployeeID, LastName, FirstName,
       LTRIM(RTRIM(EmployeeAddress)) AS CleanedAddress,City,
       PostCode, Phone
FROM Employee.Employee;

--Back to Business scenario Q43


--Q44. Sample SolutionSELECT

EmployeeID,
       LEN(LastName) AS LastNameLength,
       LEN(FirstName) AS FirstNameLength
FROM Employee.Employee;

--Back to Business scenario Q44


--Q45. Sample Solution

SELECT EmployeeID, JobTitle, HireDate,
       COALESCE (HourlyRate, 0) + COALESCE (Salary, 0) + COALESCE(Bonus, 0) AS
         TotalCompensationFROM
Employee.Pay;

--Back to Business scenario Q45


--Q46. Sample Solution

SELECT EmployeeID, LastName, FirstName,
       EmployeeAddress, City, PostCode,
       RIGHT(Phone + REPLICATE('-', 15), 15) AS FormattedPhoneFROM
Employee.Employee;

--Back to Business scenario Q46
```

```
--Q47. Sample Solution

SELECT ProductID,
       LEFT(REPLICATE('.', 30) + ProductDescription, 30) AS
         FormattedProductDescription,
       Cost
FROM CustomerTab.CustomerProductTabl;

--Back to Business scenario Q47


--Q48. Sample Solution

SELECT ProductID, ProductDescription,
       CAST (Cost AS DECIMAL (10, 2)) AS ConvertedPriceFROM
CustomerTab.CustomerProductTabl;

--Back to Business scenario Q48


--Q49. Sample SolutionSELECT

EmployeeID,
       CONCAT (FirstName, ' ', LastName) AS FullName,
       CONCAT (SUBSTRING (Phone, 1, 3), '-', SUBSTRING (Phone, 4, 3), '-',
         SUBSTRING (Phone, 7, 4)) AS FormattedPhoneFROM
Employee.Employee;

--Back to Business scenario Q49


--Q50. Sample SolutionSELECT

EmployeeID,
       JobTitle,
       HireDate,
       LastDate,
       DATEDIFF (YEAR, HireDate, LastDate) AS YearsOfServiceFROM
Employee.Pay
WHERE DATEDIFF (YEAR, HireDate, LastDate) >= 7AND
      YEAR(HireDate) BETWEEN 2001 AND 2009
      AND YEAR(LastDate) BETWEEN 2011 AND 2012.

--Back to Business scenario Q50


--Q51. Sample SolutionSELECT

EmployeeID,
       JobTitle,
       HireDate,
       LastDate,
       Salary AS CurrentSalary,Bonus,
       CASE
           WHEN DATEDIFF (YEAR, HireDate, GETDATE ()) >= 2AND
                YEAR(LastDate) BETWEEN 2011 AND 2012 THEN
```

```
               Salary * 1.05 -- Apply 5% salary increase.
           ELSE Salary -- No salary increases.
       END AS FutureSalaryFROM
Employee.Pay

--Back to Business scenario Q51


--Q52. Sample SolutionSELECT

EmployeeID,
       JobTitle,
       Salary AS CurrentSalary,
          Bonus AS CurrentBonus,
          CASE
              WHEN JobTitle = 'Manager' THEN Salary * 0.10
              ELSE Salary * 0.05
          END AS PerformanceBonus
   FROM Employee.Pay
   WHERE YEAR(LastDate) BETWEEN 2011 AND 2012
```

--Back to Business scenario Q52

```
--Q53. Sample Solution

SELECT EmployeeID,
 JobTitle,
 Salary AS CurrentSalary,
 Bonus AS CurrentBonus,
 Salary *
 (CASE
   WHEN MONTH (GETDATE ()) = 1 AND JobTitle = 'Manager' THEN 0.10
   WHEN MONTH (GETDATE ()) = 7 AND JobTitle <> 'Manager' THEN 0.07
   ELSE 0.05
   END) AS PerformanceBonus
FROM Employee.Pay
```

--Back to Business scenario Q53

```
--Q54. Sample Solution

SELECT EmployeeID,
JobTitle,
HireDate,
Salary AS OriginalSalary,
CASE
WHEN JobTitle = 'Salesman' AND DATEDIFF (YEAR, HireDate, GETDATE ()) > 5 THEN
  Salary * 1.10
WHEN JobTitle = 'Shipper' AND DATEDIFF (YEAR, HireDate, GETDATE ()) > 3 THEN Salary
   * 1.05
ELSE Salary
END AS AdjustedSalary
FROM Employee.Pay
```

--Back to Business scenario Q54

```
--Q55. Sample Solution

SELECT EmployeeID,
JobTitle,
HireDate,
HourlyRate,
LastDate,
CASE
WHEN HourlyRate IS NOT NULL AND DATEDIFF (WEEK, HireDate, GETDATE ()) > 0 THEN
(40 * HourlyRate) + ((CASE WHEN HourlyRate > 0 THEN (HourlyRate * 1.5) ELSE 0 END)
  * (DATEDIFF (WEEK, HireDate, GETDATE ()) - 1) * 40)
ELSE
Salary
END AS CalculatedSalary
```

```
FROM Employee.Pay;
--Back to Business scenario Q55


--Q56. Sample Solution

SELECT EmployeeID,

       JobTitle,
       HireDate,
       DATEDIFF(YEAR, HireDate, GETDATE()) AS YearsOfService,
       CASE
           WHEN DATEDIFF(YEAR, HireDate, GETDATE()) < 5 THEN 10
           ELSE 15
       END AS VacationDaysAccrued
FROM Employee.Pay

--Back to Business scenario Q56


--Q57. Sample Solution

SELECT EmployeeID,
       JobTitle,
       HireDate,
       DATEDIFF (YEAR, HireDate, GETDATE ()) AS YearsOfService,
       CASE
           WHEN DATEDIFF (YEAR, HireDate, GETDATE ()) >= 20 THEN 'Eligible for
             Retirement'
           ELSE 'Not Eligible for Retirement'
       END AS RetirementStatus
FROM Employee.Pay

--Back to Business scenario Q57


--Q58. Sample Solution

SELECT EmployeeID,
       JobTitle,
       CONVERT (NVARCHAR (10), HireDate, 120) AS HireDateFormatted,
       CONVERT (NVARCHAR (10), LastDate, 120) AS LastDateFormatted
FROM Employee.Pay

--Back to Business scenario Q58


--Q59. Sample Solution

SELECT JobTitle,
       CONCAT ('Congratulations, "', JobTitle, '", on your ', DATEDIFF (YEAR,
         HireDate, GETDATE ()), '-year work anniversary with our company!') AS
         AnniversaryMessage
FROM Employee.Pay

--Back to Business scenario Q59
```

```
--Q60. Sample Solution

SELECT E.LastName,
       E.FirstName,
       E.EmployeeAddress,
       E.City,
       E.PostCode,
       E.Phone,
       EP.JobTitle,
       EP.HireDate,
       EP.Salary,
       EP.Bonus
FROM Employee.Employee E
INNER JOIN Employee.Pay EP ON E.EmployeeID = EP.EmployeeID

--Back to Business scenario Q60


--Q61. Sample Solution

SELECT E.LastName,

       E.FirstName,
       E.EmployeeAddress,
       E.City,
       E.PostCode,
       E.Phone,
       EP.JobTitle,
       EP.HireDate,
       EP.Salary,
       EP.Bonus,
       (EP.Salary + EP.Bonus) AS TotalCompensation
FROM Employee.Employee E
INNER JOIN Employee.Pay EP ON E.EmployeeID = EP.EmployeeID

--Back to Business scenario Q61


--Q62. Sample Solution

SELECT E.LastName,
       E.FirstName,
       EP.JobTitle,
       EP.Salary
FROM Employee.Employee E
INNER JOIN Employee.Pay EP ON E.EmployeeID = EP.EmployeeID
WHERE (EP.JobTitle = 'Sales Support' OR EP.JobTitle = 'Salesman')
  AND DATEDIFF (YEAR, EP.HireDate, EP.LastDate) >= 5;

--Back to Business scenario Q62


--Q63. Sample Solution

SELECT E.LastName,
       E.FirstName,
       E.City,
       EP.JobTitle,
       EP.Bonus
```

```
FROM Employee.Employee E
INNER JOIN Employee.Pay EP ON E.EmployeeID = EP.EmployeeID
WHERE EP.JobTitle = 'Salesman' OR EP.JobTitle = 'Sales Support'
  AND EP.Bonus >= 1000
ORDER BY EP.JobTitle, EP.Bonus;

--Back to Business scenario Q63


--Q64. Sample Solution

SELECT E.LastName,
       E.FirstName,
       E.City,
       EP.JobTitle,
       EP.Salary,
       DATEDIFF (YEAR, EP.HireDate, GETDATE()) AS TenureInYears
FROM Employee.Employee E
INNER JOIN Employee.Pay EP ON E.EmployeeID = EP.EmployeeID
WHERE EP.JobTitle = 'Salesman' or EP.JobTitle = 'Sales Support'
ORDER BY EP.JobTitle, EP.Salary;

--Back to Business scenario Q64


--Q65. Sample Solution

SELECT C.CustomerName,
       C.City,
       CO.OrderDate,
       P.ProductDescription,
       CO.OrderQuantity
FROM CustomerTab.CustomerTabl C
INNER JOIN CustomerTab.CustomerOrdersTabl CO ON C.CustomerID = CO.CustomerID
INNER JOIN CustomerTab.CustomerProductTabl P ON CO.ProductID = P.ProductID
ORDER BY CO.OrderDate, CO.OrderQuantity;

--Back to Business scenario Q65


--Q66 Sample Solution

SELECT C.CustomerName,
       C.City,
       CO.OrderDate,
       P.ProductDescription,
       CO.OrderQuantity
FROM CustomerTab.CustomerTabl C
LEFT JOIN CustomerTab.CustomerOrdersTabl CO ON C.CustomerID = CO.CustomerID
LEFT JOIN CustomerTab.CustomerProductTabl P ON CO.ProductID = P.ProductID
ORDER BY C.CustomerName, CO.OrderDate, P.ProductDescription;

--Back to Business scenario Q66


--Q67. Sample Solution

SELECT C.CustomerName,
       C.City,
       CO.OrderDate,
       P.ProductDescription,
       CO.OrderQuantity
FROM CustomerTab.CustomerOrdersTabl CO
RIGHT JOIN CustomerTab.CustomerTabl C ON CO.CustomerID = C.CustomerID
LEFT JOIN CustomerTab.CustomerProductTabl P ON CO.ProductID = P.ProductID
ORDER BY C.CustomerName, CO.OrderDate, P.ProductDescription;

--Back to Business scenario Q67


--Q68. Sample Solution

SELECT C.CustomerName,
       C.Address,
       C.City,
       C.ZipCode,
       C.Phone,
       CO.OrderID,
       CO.OrderQuantity,
       CO.OrderDate
FROM CustomerTab.CustomerTabl C
FULL JOIN CustomerTab.CustomerOrdersTabl CO ON C.CustomerID = CO.CustomerID
ORDER BY C.CustomerName, CO.OrderDate;

--Back to Business scenario Q68


--Q69. Sample Solution

SELECT E1. EmployeeID AS Employee1ID,
       E1. LastName AS Employee1LastName,
       E1. FirstName AS Employee1FirstName,
       E1. Phone AS Employee1Phone,
       E2. EmployeeID AS Employee2ID,
       E2. LastName AS Employee2LastName,
       E2. FirstName AS Employee2FirstName,
       E2. Phone AS Employee2Phone,
       E1. City AS CommonCity
FROM Employee.Employee E1
INNER JOIN Employee.Employee E2 ON E1. City = E2. City AND E1. EmployeeID < E2.
  EmployeeID;

--Back to Business scenario Q69


--Q70. Sample Solution

SELECT P.ProductDescription, O.OrderQuantity
FROM CustomerTab.CustomerProductTabl P
LEFT OUTER JOIN CustomerTab.CustomerOrdersTabl O
ON P.ProductID = O.ProductID;

--Back to Business scenario Q70


--Q71. Sample Solution

SELECT E.EmployeeID,
       E.LastName,
       E.FirstName,
       (EP.Salary + EP.Bonus) AS TotalCompensation
FROM Employee.Employee E
CROSS JOIN Employee.Pay EP
ORDER BY E.EmployeeID;

--Back to Business scenario Q71


--Q72. Sample Solution

SELECT
    E.LastName,
    E.FirstName,
    EP.Bonus
FROM Employee.Employee E
INNER JOIN Employee.Pay EP ON E.EmployeeID = EP.EmployeeID
WHERE EP.Bonus IN (
    SELECT TOP 5 Bonus
    FROM Employee.Pay
)
ORDER BY EP.Bonus DESC;

--Back to Business scenario Q72


--Q73. Sample Solution

SELECT CustomerName
FROM CustomerTab.CustomerTabl
WHERE CustomerID IN (
    SELECT CustomerID
    FROM CustomerTab.CustomerOrdersTabl
    GROUP BY CustomerID
    HAVING SUM(OrderQuantity) > 25
);

--Back to Business scenario Q73


--Q74. Sample Solution

SELECT
    CustomerID,
    SUM(order_total) AS total_spent
FROM (
    SELECT
        o.CustomerID,
        p.Cost * o.OrderQuantity AS order_total
    FROM
        CustomerTab.CustomerOrdersTabl o
    JOIN
        CustomerTab.CustomerProductTabl p ON o.ProductID = p.ProductID
) AS customer_orders
GROUP BY
    CustomerID
ORDER BY
    total_spent DESC;

--Back to Business scenario Q74
```

```
--Q75. Sample Solution


SELECT
    p.ProductID,
    p.ProductDescription
FROM
    CustomerTab.CustomerProductTabl p
LEFT JOIN
    (
        SELECT DISTINCT
            ProductID
        FROM
            CustomerTab.CustomerOrdersTabl
    ) AS ordered_products
ON
    p.ProductID = ordered_products.ProductID
WHERE
    ordered_products.ProductID IS NULL;

--Back to Business scenario Q75


--Q76. Sample Solution

SELECT
    CustomerID,
    AVG(order_total) AS average_order_cost
FROM (
    SELECT
        o.CustomerID,
        p.Cost * o.OrderQuantity AS order_total
    FROM
        CustomerTab.CustomerOrdersTabl o
    JOIN
        CustomerTab.CustomerProductTabl p ON o.ProductID = p.ProductID
) AS customer_orders
GROUP BY
    CustomerID
HAVING
    AVG(order_total) > 50;

--Back to Business scenario Q76


--Q77. Sample Solution

SELECT AVG(EP.Salary) AS AverageSalary, COUNT(*) AS EmployeeCount
FROM Employee.Pay AS EP
WHERE YEAR(EP.HireDate) = (
    SELECT YEAR(HireDate)
    FROM Employee.Pay
    WHERE Salary = (
        SELECT MAX(Salary)
        FROM Employee.Pay
    )
);
```

--Back to Business scenario Q77

```
--Q78. Sample Solution

SELECT E.EmployeeID, E.FirstName, E.LastName, EP.JobTitle, EP.Salary
FROM Employee.Employee AS E
INNER JOIN Employee.Pay AS EP ON E.EmployeeID = EP.EmployeeID
WHERE EP.HireDate <= DATEADD(YEAR, -5, GETDATE())
  AND EP.Salary < (
    SELECT AVG (EP2.Salary)
    FROM Employee.Pay AS EP2
    WHERE EP2.JobTitle = EP.JobTitle
);
```

--Back to Business scenario Q78

```
--Q79. Sample Solution

SELECT
    c.CustomerName,
    c.Address
FROM
    CustomerTab.CustomerTabl c
WHERE
    c.CustomerID IN (
        SELECT TOP 10
            o.CustomerID
        FROM
            CustomerTab.CustomerOrdersTabl o
        GROUP BY
            o.CustomerID
        ORDER BY
            SUM(o.OrderQuantity) DESC
    );
```

--Back to Business scenario Q79

```
--Q80. Sample Solution

SELECT
    ProductID,
    ProductDescription,
    Cost,
    'Top-Selling' AS SalesCategory
FROM
    CustomerTab.CustomerProductTabl
WHERE
    ProductID IN (
        SELECT TOP 5
            ProductID
        FROM
            CustomerTab.CustomerProductTabl
        ORDER BY
            Cost DESC
    )
UNION
```

```
-- Query to retrieve least-selling products
SELECT
    ProductID,
    ProductDescription,
    Cost,
    'Least-Selling' AS SalesCategory
FROM
    CustomerTab.CustomerProductTabl
WHERE
    ProductID IN (
        SELECT TOP 5
            ProductID
        FROM
            CustomerTab.CustomerProductTabl
        ORDER BY
            Cost ASC
    );

--Back to Business scenario Q80


--Q81. Sample Solution

SELECT
    ProductID,
    ProductDescription AS Name,
    NULL AS Address
FROM
    CustomerTab.CustomerProductTabl
UNION
-- Query to retrieve customers from the specified city
SELECT
    CustomerID,
    CustomerName AS Name,
    City AS Location
FROM
    CustomerTab.CustomerTabl

--Back to Business scenario Q81


--Q82. Sample Solution

SELECT
    c.CustomerID,
    c.CustomerName,
    c.Address,
    c.City
FROM
    CustomerTab.CustomerTabl c

EXCEPT

-- Query to retrieve customers who have placed orders
SELECT
    c.CustomerID,
    c.CustomerName,
    c.Address,
    c.City
FROM
```

```
    CustomerTab.CustomerTabl c
JOIN
    CustomerTab.CustomerOrdersTabl o
ON
    c.CustomerID = o.CustomerID;

--Back to Business scenario Q82


--Q83. Sample Solution

SELECT
    p.ProductID,
    p.ProductDescription,
    p.Cost
FROM
    CustomerTab.CustomerProductTabl p

EXCEPT

-- Query to retrieve products that have been ordered
SELECT
    p.ProductID,
    p.ProductDescription,
    p.Cost
FROM
    CustomerTab.CustomerProductTabl p
JOIN
    CustomerTab.CustomerOrdersTabl o
ON
    p.ProductID = o.ProductID;

--Back to Business scenario Q83


--Q84. Sample Solution

SELECT
    JobTitle,
    EmployeeID,
    SUM(Salary) as TotalSalary
FROM
    Employee.Pay
GROUP BY ROLLUP (JobTitle, EmployeeID );

--Back to Business scenario Q84


--Q85. Sample Solution

SELECT CustomerID, ProductID,
    COUNT(*) AS TotalOrders,
    SUM(OrderQuantity) AS TotalQuantity
FROM CustomerTab.CustomerOrdersTabl
GROUP BY CustomerID, ProductID
WITH ROLLUP;

--Back to Business scenario Q85
```

```
--Q86. Sample Solution

SELECT CustomerID, ProductID, MONTH(OrderDate) AS OrderMonth,
    COUNT(*) AS TotalOrders,
    SUM(OrderQuantity) AS TotalQuantity
FROM CustomerTab.CustomerOrdersTabl
GROUP BY CustomerID, ProductID, MONTH(OrderDate)
WITH ROLLUP;


--Back to Business scenario Q86
```

```
--Q87. Sample Solution

SELECT JobTitle, SUM(Salary) AS TotalSalary
FROM Employee.Pay
GROUP BY JobTitle
WITH ROLLUP;

--Back to Business scenario Q87
```

```
--Q88. Sample Solution

SELECT
    JobTitle AS Job_Title,
    City  AS  City,
    YEAR(HireDate) AS Hire_Year,
    SUM(Salary) AS Total_Compensation_Expense
FROM
    Employee.Pay
JOIN
    Employee.Employee ON Employee.Pay.EmployeeID = Employee.Employee.EmployeeID
GROUP BY
    CUBE (JobTitle, City, YEAR(HireDate))
ORDER BY
    Job_Title, City, Hire_Year;

--Back to Business scenario Q88
```

```
--Q89. Sample Solution

SELECT
    p.ProductID AS Product,
    c.City AS City,
    YEAR(o.OrderDate) AS Order_Year,
    SUM(o.OrderQuantity * p.Cost) AS "Revenue"
FROM
    CustomerTab.CustomerOrdersTabl o
JOIN
    CustomerTab.CustomerTabl c ON o.CustomerID = c.CustomerID
JOIN
    CustomerTab.CustomerProductTabl p ON o.ProductID = p.ProductID
GROUP BY
    CUBE (p.ProductID, c.City, YEAR(o.OrderDate))
ORDER BY
```

```
    Product, City, Order_Year;

--Back to Business scenario Q89


--Q90. Sample Solution

SELECT
    CustomerID,
    CASE
        WHEN SUM(OrderQuantity * Cost) > 500 THEN 'Loyal Customer'
        ELSE 'Regular Customer'
    END AS CustomerStatus,
    SUM(OrderQuantity * Cost) AS TotalAmountSpent
FROM CustomerTab.CustomerOrdersTabl
JOIN CustomerTab.CustomerProductTabl ON CustomerTab.CustomerOrdersTabl.ProductID = 
  CustomerTab.CustomerProductTabl.ProductID
GROUP BY CustomerID
HAVING SUM(OrderQuantity * Cost) > 200;

--Back to Business scenario Q90


--Q91. Sample Solution

SELECT

    ProductDescription,
    Cost,
    SUM(OrderQuantity * Cost) AS TotalRevenue,
    SUM(OrderQuantity) AS TotalQuantitySold,
    ((SUM(OrderQuantity * Cost) - COUNT(OrderID) * Cost) / SUM(OrderQuantity * 
      Cost)) * 100 AS ProfitMargin,
    CASE
        WHEN SUM(OrderQuantity * Cost) > 500 THEN 'High Sales'
        ELSE 'Low Sales'
    END AS SalesCategory,
    CASE
        WHEN SUM(OrderQuantity) > 100 THEN 'High Demand'
        ELSE 'Low Demand'
    END AS DemandCategory,
    CASE
        WHEN ((SUM(OrderQuantity * Cost) - COUNT(OrderID) * Cost) / SUM
          (OrderQuantity * Cost)) * 100 >= 30 THEN 'High Profit Margin'
        ELSE 'Low Profit Margin'
    END AS ProfitMarginCategory,
    CASE
        WHEN SUM(OrderQuantity) = 0 THEN 'Not Sold'
        ELSE 'Sold'
    END AS SoldStatus
FROM CustomerTab.CustomerOrdersTabl
JOIN CustomerTab.CustomerProductTabl ON CustomerTab.CustomerOrdersTabl.ProductID = 
  CustomerTab.CustomerProductTabl.ProductID
GROUP BY ProductDescription, Cost
HAVING SUM(OrderQuantity * Cost) > 500 OR SUM(OrderQuantity) > 100 OR ((SUM 
  (OrderQuantity * Cost) - COUNT(OrderID) * Cost) / SUM(OrderQuantity * Cost)) * 
  100 >= 30 OR SUM(OrderQuantity) = 0;

--Back to Business scenario Q91
```

```
--Q92. Sample Solution

SELECT
    ProductID,
    Name,
    Color,
    ListPrice AS Price
FROM production.Product
WHERE Color IS NOT NULL
    AND Color NOT IN ('Silver', 'Black', 'White')
    AND ListPrice BETWEEN 75 AND 750
ORDER BY ListPrice DESC;

--Back to Business scenario Q92



--Q93. Sample Solution

SELECT TOP 10
    ProductID,
    Name,
    Color
FROM Production.Product
WHERE ProductNumber LIKE 'BK%'
ORDER BY ListPrice DESC;

--Back to Business scenario Q93


--Q94. Sample Solution

SELECT DISTINCT
FirstName
,LastName
,CONCAT(LEFT([LastName],4), '.', '@ADVENTUREWORKS.COM') AS EMAILADDRESS
,CONCAT ([FirstName], ' ',LastName) AS FULL_NAME
,LEN (CONCAT ([FirstName], '',LastName)) NAME_LENGTH
FROM [Person].[Person]
WHERE LEFT (FirstName,1) = LEFT (LastName,1)

--Back to Business scenario Q94


--Q95. Sample Solution

SELECT
ProductID
,PP.Name
,DaysToManufacture
FROM [Production].[Product] PP
INNER JOIN [Production].[ProductSubcategory] PPS
ON PP.ProductSubcategoryID = PPS.ProductSubcategoryID
WHERE DaysToManufacture >=3

--Back to Business scenario Q95
```

```
--Q96. Sample Solution

SELECT
    ProductID,
    Name,
    Color,
    ListPrice,
    CASE
        WHEN ListPrice < 200 THEN 'Low Value'
        WHEN ListPrice BETWEEN 201 AND 750 THEN 'Mid Value'
        WHEN ListPrice BETWEEN 751 AND 1250 THEN 'Mid to High Value'
        ELSE 'Higher Value'
    END AS PriceSegment
FROM Production.Product
WHERE Color IN ('Black', 'Silver', 'Red');

--Back to Business scenario Q96


--Q97. Sample Solution

SELECT
COUNT (DATEDIFF (YY, [BirthDate], GETDATE ()) + 5) AS NO_FOR_AWARD
FROM [HumanResources].[Employee]
WHERE (DATEDIFF (YY, [BirthDate], GETDATE ()) + 5) >= 20

--Back to Business scenario Q97


--Q98. Sample Solution

SELECT
[BusinessEntityID]
,NationalIDNumber
,65-Datediff(yy,[BirthDate],GETDATE()) as Years_to_work_before_Sentiment
FROM[HumanResources].[Employee]
WHERE 65-Datediff(yy,[BirthDate],GETDATE()) > 0

--Back to Business scenario Q98


--Q99. Sample Solution

SELECT
[Color]
,[ListPrice]
,CASE
       WHEN [Color] ='White' THEN CAST([ListPrice] * 1.08 AS
       Decimal(18,3))
       WHEN [Color] ='Yellow' THEN CAST([ListPrice] * 0.925 AS
       Decimal(18,3))
WHEN [Color] ='Black' THEN CAST([ListPrice] * 1.172 AS Decimal(18,3))
WHEN [Color] IN ('Multi','Silver','Silver/Black','Blue') THEN CAST(SQRT
  ([ListPrice]) *2 AS Decimal(18,3))
ELSE [ListPrice]
END AS NewPrice
,CASE WHEN [Color] ='White' THEN CAST([ListPrice] * 1.08 * 0.375 AS Decimal(18,3))
WHEN [Color] ='Yellow' THEN CAST([ListPrice] * 1.075 * 0.375 AS Decimal(18,3))
                     WHEN [Color] ='Black' THEN CAST([ListPrice] * 1.172  * 0.375
                     AS
```

```
                        Decimal(18,3))
                        WHEN [Color] IN ('Multi','Silver','Silver/Black','Blue') THEN
                        CAST(SQRT([ListPrice]) * 2  * 0.375 AS Decimal(18,3))
                        ELSE CAST([ListPrice] * 0.375 AS Decimal(18,3))
                END AS NewCommission
        FROM [Production].[Product]
        WHERE [Color] IS NOT NULL;
```

--Back to Business scenario Q99

--Q100. Sample Solution

```
SELECT
 FirstName + ' ' + LastName AS SalesPerson
,JobTitle
,C.HireDate
,C.SickLeaveHours
,D.Name AS Region
,SalesQuota FROM Sales.SalesPerson AS A INNER JOIN Person.Person AS B
 ON B.BusinessEntityID= A.BusinessEntityID
 INNER JOIN HumanResources.Employee AS C
 ON A.BusinessEntityID = C.BusinessEntityID
 LEFT JOIN Sales.SalesTerritory AS D ON D.TerritoryID = A.TerritoryID
```

--Back to Business scenario Q100

--Q101. Sample Solution

```
SELECT
SOH.[SalesOrderNumber]
,SOH.[OrderDate]
,SOD.[OrderQty]
,PP.FirstName AS SALES_PERSON
,PP.PersonType
,SSP.CommissionPct
FROM [Sales].[SalesOrderHeader] SOH
INNER JOIN [Sales].[SalesOrderDetail] SOD
ON SOH.[SalesOrderID] = SOD.[SalesOrderID]
INNER JOIN [Sales].[Customer] SC
ON SOH.[CustomerID] = SC.[CustomerID]
INNER JOIN [Sales].[SalesPerson] SSP
ON SSP.[TerritoryID] = SOH.[TerritoryID]
INNER JOIN [Person].[Person] PP
ON PP.[BusinessEntityID] = SSP.[BusinessEntityID]
```

--Back to Business scenario Q101

--Q102. Sample Solution

```
SELECT
Color
,StandardCost
,StandardCost * 0.14790 AS COMMISSION
,CASE
WHEN Color = 'BLACK' THEN ((StandardCost * 1.22) - StandardCost)
WHEN Color = 'RED' THEN ((StandardCost * 0.88) - StandardCost)
WHEN Color = 'SILVER' THEN ((StandardCost * 1.15) - StandardCost)
WHEN Color = 'MULTI' THEN ((StandardCost * 1.05) - StandardCost)
WHEN Color = 'WHITE' THEN ((StandardCost * 2) / (SQRT(StandardCost)) * 
  (StandardCost)) - StandardCost
ELSE StandardCost *1
end as [MARGIN]
FROM [Production].[Product]
WHERE Color IS NOT NULL

--Back to Business scenario Q102


--Q103. Sample Solution

SELECT
    P.ProductID,
    SUM (SOD.LineTotal) AS TotalSales,
    SUM ((SOD.UnitPrice - SOD.UnitPriceDiscount) * SOD.OrderQty) AS TotalCost,
    SUM (SOD.LineTotal - (SOD.UnitPrice - SOD.UnitPriceDiscount) * SOD.OrderQty) AS 
       TotalProfit
FROM Production.Product P
JOIN Sales.SalesOrderDetail SOD ON P.ProductID = SOD.ProductID
GROUP BY P.ProductID;

--Back to Business scenario Q103


--Q104. Sample Solution

SELECT TOP 10
    P.ProductID,
    P.Name AS ProductName,
    SUM(SOD.OrderQty) AS TotalQuantitySold
FROM Sales.SalesOrderDetail SOD
JOIN Production.Product P ON SOD.ProductID = P.ProductID
GROUP BY P.ProductID, P.Name
ORDER BY TotalQuantitySold DESC;

--Back to Business scenario Q104


--Q105. Sample Solution

SELECT
    C.CustomerID,
    COUNT(*) AS CustomerCount,
    AVG(TotalDue) AS AvgPurchaseAmount
FROM Sales.SalesOrderHeader SOH
JOIN Sales.Customer C ON SOH.CustomerID = C.CustomerID
JOIN Person.Person P ON C.PersonID = P.BusinessEntityID
GROUP BY C.CustomerID;

--Back to Business scenario Q105


--Q106. Sample Solution

SELECT TOP 10
    C.CustomerID,
    P.FirstName,
    P.LastName,
    SUM(SOH.TotalDue) AS TotalPurchases
FROM Sales.SalesOrderHeader SOH
JOIN Sales.Customer C ON SOH.CustomerID = C.CustomerID
JOIN Person.Person P ON C.PersonID = P.BusinessEntityID
GROUP BY C.CustomerID, P.FirstName, P.LastName
ORDER BY TotalPurchases DESC;

--Back to Business scenario Q106


--Q107. Sample Solution

SELECT
    SUM(PD.TotalCost) AS TotalInventoryValue
FROM (
    SELECT
        P.ProductID,
        P.Name AS ProductName,
        P.StandardCost,
        SUM(POD.OrderQty) AS TotalOrderedQuantity,
        MIN(PV.MinOrderQty) AS MinOrderQuantity,
        (SUM(POD.OrderQty) / MIN(PV.MinOrderQty)) AS ReorderCount,
        (SUM(POD.OrderQty) / MIN(PV.MinOrderQty)) * P.StandardCost AS TotalCost
    FROM Production.Product P
    LEFT JOIN Purchasing.ProductVendor PV ON P.ProductID = PV.ProductID
    LEFT JOIN Purchasing.PurchaseOrderDetail POD ON P.ProductID = POD.ProductID
    GROUP BY P.ProductID, P.Name, P.StandardCost
) PD;

--Back to Business scenario Q107


--Q108. Sample Solution

SELECT
    P.ProductID,
    P.Name AS ProductName,
    (SUM(SOD.OrderQty) - SUM(POD.OrderQty)) AS InventoryChange
FROM Production.Product P
LEFT JOIN Sales.SalesOrderDetail SOD ON P.ProductID = SOD.ProductID
LEFT JOIN Purchasing.PurchaseOrderDetail POD ON P.ProductID = POD.ProductID
GROUP BY P.ProductID, P.Name
HAVING (SUM(SOD.OrderQty) - SUM(POD.OrderQty)) <= 0;

--Back to Business scenario Q108


--Q109. Sample Solution

SELECT
    PV.BusinessEntityID,
    V.Name AS VendorName,
    AVG(DATEDIFF(DAY, POH.OrderDate, POH.ShipDate)) AS AvgDeliveryTime
FROM Purchasing.ProductVendor PV
INNER JOIN Purchasing.PurchaseOrderDetail POD ON PV.ProductID = POD.ProductID
INNER JOIN Purchasing.PurchaseOrderHeader POH ON POD.PurchaseOrderID = POH.VendorID
INNER JOIN Purchasing.Vendor V ON PV.BusinessEntityID = V.BusinessEntityID
GROUP BY PV.BusinessEntityID, V.Name;

--Back to Business scenario Q109
```

```
--Q110. Sample Solution

SELECT
    PV.BusinessEntityID,
    V.Name AS VendorName,
    SUM(CASE WHEN DATEDIFF(DAY, POH.OrderDate, POH.ShipDate) <= 0 THEN 1 ELSE 0
      END) AS OnTimeDeliveries,
    COUNT(*) AS TotalDeliveries,
    (SUM(CASE WHEN DATEDIFF(DAY, POH.OrderDate, POH.ShipDate) <= 0 THEN 1 ELSE 0
      END) * 100.0) / NULLIF(COUNT(*), 0) AS OnTimeDeliveryPercentage
FROM Purchasing.ProductVendor PV
INNER JOIN Purchasing.PurchaseOrderDetail POD ON PV.ProductID = POD.ProductID
INNER JOIN Purchasing.PurchaseOrderHeader POH ON POD.PurchaseOrderID = POH.VendorID
INNER JOIN Purchasing.Vendor V ON PV.BusinessEntityID = V.BusinessEntityID
GROUP BY PV.BusinessEntityID, V.Name;

--Back to Business scenario Q110


--Q111. Sample Solution

SELECT
    PV.BusinessEntityID,
    V.Name AS VendorName,
    AVG((POD.UnitPrice - PV.StandardPrice) * POD.OrderQty) AS AvgCostPerformance
FROM Purchasing.ProductVendor PV
INNER JOIN Purchasing.PurchaseOrderDetail POD ON PV.ProductID = POD.ProductID
INNER JOIN Purchasing.PurchaseOrderHeader POH ON POD.PurchaseOrderID = POH.VendorID
INNER JOIN Purchasing.Vendor V ON PV.BusinessEntityID = V.BusinessEntityID
GROUP BY PV.BusinessEntityID, V.Name;

--Back to Business scenario Q111


--Q112. Sample Solution

SELECT
    SUM(TotalDue) AS TotalSalesRevenue
FROM Sales.SalesOrderHeader;

--Back to Business scenario Q112


--Q113. Sample Solution

SELECT
    SUM(TotalDue) AS TotalExpenses
FROM Purchasing.PurchaseOrderHeader;

--Back to Business scenario Q113


--Q114. Sample Solution

SELECT
    TotalSalesRevenue - TotalExpenses AS NetIncome
FROM (
    SELECT
        (SELECT SUM(TotalDue) FROM Sales.SalesOrderHeader) AS TotalSalesRevenue,
        (SELECT SUM(TotalDue) FROM Purchasing.PurchaseOrderHeader) AS TotalExpenses
) FinancialData;
```

--Back to Business scenario Q114

```
--Q115. Sample Solution

SELECT
    SUM(TotalDue) AS TotalActualExpenses
FROM Purchasing.PurchaseOrderHeader;
```

--Back to Business scenario Q115

```
--Q116. Sample Solution

SELECT
    SUM(P.StandardCost * PPIV.Quantity) AS TotalInventoryCost
FROM Production.Product P
INNER JOIN Production.ProductInventory PPIV ON P.ProductID = PPIV.ProductID
```

--Back to Business scenario Q116

```
--Q117. Sample Solution

SELECT
    L.LocationID,
    L.Name AS LocationName,
    SUM(L.CostRate) AS TotalShippingCost
FROM Production.Location L
GROUP BY L.LocationID, L.Name;
```

--Back to Business scenario Q117

```
--Q118. Sample Solution

SELECT
    LocationID,
    LocationName,
    TotalShippingCost
FROM (
    SELECT
        L.LocationID,
        L.Name AS LocationName,
        SUM(L.CostRate) AS TotalShippingCost
    FROM Production.Location L
    GROUP BY L.LocationID, L.Name
) Subquery
ORDER BY TotalShippingCost DESC;
```

--Back to Business scenario Q118

```
--Q119. Sample Solution

SELECT TOP 5
    POD.ProductID,
    P.Name AS ProductName,
    COUNT(*) AS DelayedPurchaseOrders
FROM Purchasing.PurchaseOrderDetail POD
JOIN Production.Product P ON POD.ProductID = P.ProductID
WHERE POD.DueDate < GETDATE()
GROUP BY POD.ProductID, P.Name
ORDER BY DelayedPurchaseOrders DESC;

--Back to Business scenario Q119


--Q120. Sample Solution

SELECT
    ST.TerritoryID,
    ST.Name AS TerritoryName,
    SUM(SOH.TotalDue) AS TotalSales
FROM Sales.SalesOrderHeader SOH
JOIN Sales.Customer C ON SOH.CustomerID = C.CustomerID
JOIN Sales.SalesTerritory ST ON C.TerritoryID = ST.TerritoryID
GROUP BY ST.TerritoryID, ST.Name
ORDER BY TotalSales DESC;

--Back to Business scenario Q120


--Q121. Sample Solution

SELECT
    ST.TerritoryID,
    ST.Name AS TerritoryName,
    COUNT(C.CustomerID) AS CustomerCount
FROM Sales.SalesTerritory ST
JOIN Sales.Customer C ON ST.TerritoryID = C.TerritoryID
JOIN Person.Address A ON C.CustomerID = A.AddressID
GROUP BY ST.TerritoryID, ST.Name;

--Back to Business scenario Q121


--Q122. Sample Solution

SELECT
    (COUNT(DISTINCT E.BusinessEntityID) * 1.0 / (DATEDIFF(YEAR, MIN(E.HireDate),
      GETDATE()) + 1)) AS TurnoverRate
FROM HumanResources.Employee E;

--Back to Business scenario Q122


--Q123. Sample Solution

SELECT
    D.Name AS DepartmentName,
    (COUNT (DISTINCT E.BusinessEntityID) * 1.0 / (DATEDIFF(YEAR, MIN(E.HireDate),
      GETDATE()) + 1)) AS TurnoverRate
FROM HumanResources.Employee E
JOIN HumanResources.EmployeeDepartmentHistory EDH ON E.BusinessEntityID =
  EDH.BusinessEntityID
JOIN HumanResources.Department D ON EDH.DepartmentID = D.DepartmentID
```

```
GROUP BY D.Name
ORDER BY TurnoverRate DESC;

--Back to Business scenario Q123


--Q124. Sample Solution

SELECT TOP 5
    E.BusinessEntityID,
     (MAX(EPH.Rate) - MIN(EPH.Rate)) AS PayIncrease
FROM HumanResources.Employee E
JOIN HumanResources.EmployeePayHistory EPH ON E.BusinessEntityID =
  EPH.BusinessEntityID
GROUP BY E.BusinessEntityID
ORDER BY PayIncrease DESC;

--Back to Business scenario Q124


--Q125. Sample Solution

SELECT
      E.BusinessEntityID,
    MIN(E.HireDate) AS StartDate,
    DATEDIFF(YEAR, MIN(E.HireDate), GETDATE()) AS YearsWorked
FROM HumanResources.Employee E
GROUP BY  E.BusinessEntityID
HAVING DATEDIFF(YEAR, MIN(E.HireDate), GETDATE()) >= 5;

--Back to Business scenario Q125


--Q126. Sample Solution

DECLARE @RetirementAge INT = 60

SELECT
     E.BusinessEntityID,
    DATEDIFF(YEAR, E.HireDate, GETDATE()) AS YearsWorked,
    CASE
        WHEN (DATEDIFF(YEAR, E.HireDate, GETDATE()) + @RetirementAge) >=
          @RetirementAge THEN 0
        ELSE @RetirementAge - (DATEDIFF(YEAR, E.HireDate, GETDATE()) +
          @RetirementAge)
    END AS YearsToRetirement
FROM HumanResources.Employee E;

--Back to Business scenario Q126


--Q127. Sample Solution

SELECT
        E.BusinessEntityID AS EmployeeID,

        COUNT(SOH.SalesOrderID) AS TotalSalesOrders
    FROM Sales.SalesPerson SP
    JOIN HumanResources.Employee E ON SP.BusinessEntityID = E.BusinessEntityID
    LEFT JOIN Sales.SalesOrderHeader SOH ON SP.BusinessEntityID = SOH.SalesPersonID
    GROUP BY E.BusinessEntityID;

--Back to Business scenario Q127


--Q128. Sample Solution

SELECT
SO.SpecialOfferID,
COUNT(DISTINCT S.SalesOrderID) AS TotalCustomersReached, SUM(CASE
WHEN S.SalesOrderID IS NOT NULL THEN 1 ELSE 0 END) AS
  CustomersConverted,
NULLIF(SUM(CASE WHEN S.SalesOrderID IS NOT NULL THEN 1 ELSE 0 END) *
  100.0 / NULLIF(COUNT(DISTINCT S.SalesOrderID), 0), 0) AS ConversionRate
    FROM Sales.SpecialOffer SO
    LEFT JOIN Sales.SalesOrderDetail S ON SO.SpecialOfferID = S.SpecialOfferID
    GROUP BY SO.SpecialOfferID;

--Back to Business scenario Q128


--Q129. Sample Solution

SELECT

SO.SpecialOfferID,
SO.Description AS PromotionDescription,SO.DiscountPct,
COUNT(DISTINCT OH.SalesOrderID) AS CustomersAcquired,
SUM(OH.LineTotal) AS TotalSalesAmount
    FROM Sales.SpecialOffer SO
    LEFT JOIN Sales.SalesOrderDetail OH ON SO.SpecialOfferID = OH.SpecialOfferID
    GROUP BY SO.SpecialOfferID, SO.Description, SO.DiscountPct;

--Back to Business scenario Q129


--Q130. Sample Solution

SELECT

C.CustomerID,
MIN(OH.OrderDate) AS FirstPurchaseDate, MAX(OH.OrderDate)
AS LatestPurchaseDate, DATEDIFF(DAY, MIN(OH.OrderDate),
MAX(OH.OrderDate)) AS
  DaysSinceFirstPurchase,
        CASE
            WHEN COUNT(OH.SalesOrderID) > 1 THEN 'Returning Customer'
            ELSE 'New Customer'
        END AS CustomerType
    FROM Sales.Customer C
    LEFT JOIN Sales.SalesOrderHeader OH ON C.CustomerID = OH.CustomerID
    GROUP BY C.CustomerID

--Back to Business scenario Q130
```

```
--Q131. Sample Solution

SELECT
P.ProductID,
P.Name AS ProductName,
P.SellStartDate,
P.DiscontinuedDate,
CASE
    WHEN P.SellStartDate IS NULL THEN 'Unknown'
    WHEN P.DiscontinuedDate IS NULL THEN 'Introduction'
    WHEN P.DiscontinuedDate IS NOT NULL AND DATEDIFF(DAY, P.SellStartDate,
      P.DiscontinuedDate) <= 365 THEN 'Growth'
    WHEN P.DiscontinuedDate IS NOT NULL AND DATEDIFF(DAY, P.SellStartDate,
      P.DiscontinuedDate) <= 1095 THEN 'Maturity'
    ELSE  'Decline'
 END AS LifeCycleStage

FROM Production.Product P
```

--Back to Business scenario Q131

```
--Q132. Sample Solution

 SELECT
        C.CustomerID,
        COUNT(OH.SalesOrderID) AS TotalOrders,
        SUM(OH.TotalDue) AS TotalSpent,
        DATEDIFF (DAY, MIN(OH.OrderDate), MAX(OH.OrderDate)) AS DaysAsCustomer,
        CASE
            WHEN COUNT(OH.SalesOrderID) > 0 AND DATEDIFF(DAY, MIN(OH.OrderDate),
              MAX(OH.OrderDate)) > 0 THEN
                SUM(OH.TotalDue) / NULLIF(DATEDIFF(DAY, MIN(OH.OrderDate), MAX
                  (OH.OrderDate)), 0)
            ELSE
                0
        END AS AverageDailySpend,
        SUM(OH.TotalDue) / NULLIF(COUNT(OH.SalesOrderID), 0) AS AverageOrderValue,
        SUM(OH.TotalDue) * 365 / NULLIF(DATEDIFF(DAY, MIN(OH.OrderDate), MAX
          (OH.OrderDate)), 0) AS AnnualCLV
    FROM Sales.Customer C
    LEFT JOIN Sales.SalesOrderHeader OH ON C.CustomerID = OH.CustomerID
    GROUP BY C.CustomerID
```

--Back to Business scenario Q132

```
--Q133. Sample Solution

SELECT
P.ProductID,
P.Name AS ProductName, AVG(PR.Rating) AS
AverageRating,
COUNT(PR.ProductReviewID) AS TotalReviews,
        SUM(CASE WHEN PR.Rating >= 4 THEN 1 ELSE 0 END) AS PositiveReviews,
        SUM(CASE WHEN PR.Rating < 4 THEN 1 ELSE 0 END) AS NegativeReviews
    FROM Production.Product P
    LEFT JOIN Production.ProductReview PR ON P.ProductID = PR.ProductID
    GROUP BY P.ProductID, P.Name
```

--Back to Business scenario Q133

```
--Q134. Sample Solution

SELECT
        YEAR(OH.OrderDate) AS SalesYear,
        SUM(OH.TotalDue) AS AnnualSales
    FROM Sales.SalesOrderHeader OH
    WHERE OH.OrderDate >= DATEADD(YEAR, -10, GETDATE()) -- Sales data from the last
       year
    GROUP BY YEAR(OH.OrderDate)
```

--Back to Business scenario Q134

```
--Q135. Sample Solution

SELECT
        YEAR(OH.OrderDate) AS SalesYear,
        COUNT(DISTINCT OH.CustomerID) AS TotalCustomers,
        COUNT(SO.SalesOrderID) AS TotalOrders,
        100.0 * COUNT(SO.SalesOrderID) / COUNT(DISTINCT OH.CustomerID) AS
          ConversionRate
```

```
    FROM Sales.SalesOrderHeader OH
    LEFT JOIN Sales.SalesOrderHeader SO ON OH.CustomerID = SO.CustomerID
    WHERE YEAR(OH.OrderDate) >= YEAR(GETDATE()) - 20
    GROUP BY YEAR(OH.OrderDate)
```

--Back to Business scenario Q135

```
--Q136. Sample Solution

SELECT
    A.City,
    SUM(H.TotalDue) AS TotalSalesAmount
FROM
    Person.Address AS A
INNER JOIN
    Person.BusinessEntityAddress AS B
    ON A.AddressID = B.AddressID
INNER JOIN
    Sales.SalesOrderHeader AS H
    ON B.BusinessEntityID = H.CustomerID
GROUP BY
    A.City
ORDER BY
    TotalSalesAmount DESC;
```

--Back to Business scenario Q136

```
--Q137. Sample Solution

SELECT
    A.City,
    COUNT(DISTINCT H.CustomerID) AS TotalCustomers,
    AVG(H.TotalDue) AS AverageOrderValue
FROM
    Person.Address AS A
INNER JOIN
    Person.BusinessEntityAddress AS B
    ON A.AddressID = B.AddressID
INNER JOIN
    Sales.SalesOrderHeader AS H
    ON B.BusinessEntityID = H.CustomerID
GROUP BY
    A.City
ORDER BY
    TotalCustomers DESC;
```

--Back to Business scenario Q137

```
--Q138. Sample Solution

SELECT
    CASE
        WHEN TotalDue > 5000 THEN 'High-Value Customers'
        WHEN TotalDue BETWEEN 1000 AND 5000 THEN 'Regular Customers'
        ELSE 'Occasional Customers'
    END AS CustomerSegment,
    COUNT(*) AS CustomerCount
FROM
    Sales.SalesOrderHeader
GROUP BY
    CASE
        WHEN TotalDue > 5000 THEN 'High-Value Customers'
        WHEN TotalDue BETWEEN 1000 AND 5000 THEN 'Regular Customers'
        ELSE 'Occasional Customers'
    END;

--Back to Business scenario Q138


--Q139. Sample Solution

SELECT TOP 5
    v.BusinessEntityID AS VendorID,
    COUNT(DISTINCT p.ProductID) AS ProductCount
FROM
    Purchasing.ProductVendor AS pv
JOIN
    Purchasing.Vendor AS v ON pv.BusinessEntityID = v.BusinessEntityID
JOIN
    Production.Product AS p ON pv.ProductID = p.ProductID
GROUP BY
    v.BusinessEntityID
ORDER BY
    ProductCount DESC;

SELECT TOP 1
    s.Name AS StoreName,
    COUNT(DISTINCT v.BusinessEntityID) AS VendorCount
FROM
    Sales.Customer AS c
JOIN
    Sales.Store AS s ON c.StoreID = s.BusinessEntityID
JOIN
    Purchasing.Vendor AS v ON c.StoreID = v.BusinessEntityID
GROUP BY
    s.Name
ORDER BY
    VendorCount DESC;

-- Average number of vendor relationships per store
SELECT
    AVG(VendorCount) AS AvgVendorRelationshipsPerStore
FROM (
    SELECT
        c.StoreID,
        COUNT(DISTINCT v.BusinessEntityID) AS VendorCount
    FROM
        Sales.Customer AS c
    JOIN
        Purchasing.Vendor AS v ON c.StoreID = v.BusinessEntityID
    GROUP BY
        c.StoreID
) AS Subquery;
```

--Back to Business scenario Q139

```sql
--Q140. Sample Solution

-- Identify distinct country regions and their associated states
SELECT DISTINCT
    cr.CountryRegionCode,
    cr.Name AS CountryRegion,
    sp.StateProvinceCode,
    sp.Name AS StateProvince
FROM
    Person.CountryRegion AS cr
JOIN
    Person.StateProvince AS sp ON cr.CountryRegionCode = sp.CountryRegionCode
ORDER BY
    cr.CountryRegionCode, sp.StateProvinceCode;

-- Determine the currencies used in each country region and sales territory

SELECT
    cr.CountryRegionCode,
    cr.Name AS CountryRegion,
    st.TerritoryID,
    st.Name AS SalesTerritory,
    crc.CurrencyCode

FROM
    Sales.CountryRegionCurrency AS crc
JOIN
    Person.CountryRegion AS cr ON crc.CountryRegionCode = cr.CountryRegionCode
JOIN
    Sales.SalesTerritory AS st ON cr.CountryRegionCode = st.CountryRegionCode
ORDER BY
    cr.CountryRegionCode, crc.CurrencyCode;
```

--Back to Business scenario Q140

```sql
--Q141. Sample Solution

SELECT
    p.BusinessEntityID AS PersonID,
    p.FirstName,
    p.LastName,
    ps.PasswordHash
FROM
    Person.Person AS p
JOIN
    Person.Password AS ps ON p.BusinessEntityID = ps.BusinessEntityID
ORDER BY
    p.BusinessEntityID;
```

--Back to Business scenario Q141

```
--Q142. Sample Solution

SELECT
    p.BusinessEntityID,
    p.FirstName,
    p.LastName,
    ISNULL(pp.PhoneNumber, '') AS PhoneNumber
FROM
    Person.Person AS p
LEFT JOIN
    HumanResources.Employee AS e ON p.BusinessEntityID = e.BusinessEntityID
LEFT JOIN
    Sales.Customer AS c ON p.BusinessEntityID = c.PersonID
LEFT JOIN
    Person.EmailAddress AS ea ON p.BusinessEntityID = ea.BusinessEntityID
LEFT JOIN
    Person.PersonPhone AS pp ON p.BusinessEntityID = pp.BusinessEntityID
ORDER BY
    p.BusinessEntityID;

--Back to Business scenario Q142


--Q143. Sample Solution

SELECT
    a.City,
    sp.Name AS StateName,
    str.TaxRate AS SalesTaxRate
FROM
    Person.Address AS a
INNER JOIN
    Person.StateProvince AS sp ON a.StateProvinceID = sp.StateProvinceID
LEFT JOIN
    Sales.SalesTaxRate AS str ON sp.StateProvinceID = str.StateProvinceID
ORDER BY
    a.City;

--Back to Business scenario Q143


--Q144. Sample Solution

SELECT
    edh.DepartmentID,
    e.JobTitle,
    e.NationalIDNumber,
    d.Name AS DepartmentName,
    edh.StartDate,
    edh.EndDate
FROM
    HumanResources.EmployeeDepartmentHistory AS edh
INNER JOIN
    HumanResources.Employee AS e ON edh.BusinessEntityID = e.BusinessEntityID
INNER JOIN
    HumanResources.Department AS d ON edh.DepartmentID = d.DepartmentID
ORDER BY
```

```
    edh.DepartmentID,
    edh.StartDate;

--Back to Business scenario Q144


--Q145. Sample Solution

SELECT
    jc.JobCandidateID,
    jc.Resume AS CandidateResume,
    e.JobTitle
FROM
    HumanResources.JobCandidate AS jc
LEFT JOIN
    HumanResources.Employee AS e ON jc.JobCandidateID = e.BusinessEntityID
WHERE
    e.JobTitle IS NOT NULL;

--Back to Business scenario Q145


--Q146. Sample Solution

SELECT
    d.DocumentNode,
    d.Title AS DocumentTitle,
    COUNT(pd.ProductID) AS AssociatedProductCount
FROM
    Production.Document AS d
LEFT JOIN
    Production.ProductDocument AS pd
ON
    d.DocumentNode = pd.DocumentNode
GROUP BY
    d.DocumentNode, d.Title
ORDER BY
    AssociatedProductCount DESC, d.DocumentNode;

--Back to Business scenario Q146


--Q147. Sample Solution

SELECT
    pc.Name AS CategoryName,
    ps.Name AS SubcategoryName,
    p.Name AS ProductName,
    pm.Name AS ModelName,
    p.StandardCost,
    p.ListPrice,
    um.Name AS UnitOfMeasure,
    p.Weight
FROM
    Production.Product AS p
INNER JOIN
    Production.ProductSubcategory AS ps
ON
```

```
    p.ProductSubcategoryID = ps.ProductSubcategoryID
INNER JOIN
    Production.ProductCategory AS pc
ON
    ps.ProductCategoryID = pc.ProductCategoryID
INNER JOIN
    Production.ProductModel AS pm
ON
    p.ProductModelID = pm.ProductModelID
INNER JOIN
    Production.ProductProductPhoto AS ppp
ON
    p.ProductID = ppp.ProductID
INNER JOIN
    Production.ProductPhoto AS pp
ON
    ppp.ProductPhotoID = pp.ProductPhotoID
INNER JOIN
    Production.UnitMeasure AS um
ON
    p.SizeUnitMeasureCode = um.UnitMeasureCode
ORDER BY
    CategoryName, SubcategoryName, ProductName;

--Back to Business scenario Q147


--Q148. Sample Solution

SELECT
    pc.Name AS CategoryName,
    ps.Name AS SubcategoryName,
    COUNT(p.ProductID) AS ProductCount,
    MAX(p.ListPrice) AS MaxListPrice,
    MIN(p.ListPrice) AS MinListPrice,
    AVG(p.ListPrice) AS AvgListPrice,
    um.Name AS UnitOfMeasure
FROM
    Production.Product AS p
INNER JOIN
    Production.ProductSubcategory AS ps
ON
    p.ProductSubcategoryID = ps.ProductSubcategoryID
INNER JOIN
    Production.ProductCategory AS pc
ON
    ps.ProductCategoryID = pc.ProductCategoryID
INNER JOIN
    Production.UnitMeasure AS um
ON
    p.SizeUnitMeasureCode = um.UnitMeasureCode
GROUP BY
    pc.Name, ps.Name, um.Name
ORDER BY
    CategoryName, SubcategoryName;

--Back to Business scenario Q148
```

```
--Q149. Sample Solution

SELECT
    pc.Name AS CategoryName,
    ps.Name AS SubcategoryName,
    COUNT(p.ProductID) AS ProductCount,
    SUM(p.StandardCost) AS TotalInventoryValue,
    um.Name AS UnitOfMeasure
FROM
    Production.Product AS p
INNER JOIN
    Production.ProductSubcategory AS ps
ON
    p.ProductSubcategoryID = ps.ProductSubcategoryID
INNER JOIN
    Production.ProductCategory AS pc
ON
    ps.ProductCategoryID = pc.ProductCategoryID
INNER JOIN
    Production.UnitMeasure AS um
ON
    p.SizeUnitMeasureCode = um.UnitMeasureCode
GROUP BY
    pc.Name, ps.Name, um.Name
ORDER BY
    TotalInventoryValue DESC;

--Back to Business scenario Q149


--Q150. Sample Solution

SELECT
    pc.Name AS CategoryName,
    ps.Name AS SubcategoryName,
    p.Name AS ProductName,
    SUM(sod.OrderQty) AS TotalUnitsSold,
    SUM(soh.TotalDue) AS TotalSalesAmount,
    AVG(soh.TotalDue * 1.0 / sod.OrderQty) AS AvgSalesPrice
FROM
    Production.Product AS p
INNER JOIN
    Production.ProductSubcategory AS ps
ON
    p.ProductSubcategoryID = ps.ProductSubcategoryID
INNER JOIN
    Production.ProductCategory AS pc
ON
    ps.ProductCategoryID = pc.ProductCategoryID
INNER JOIN
    Sales.SalesOrderDetail AS sod
ON
    p.ProductID = sod.ProductID
INNER JOIN
    Sales.SalesOrderHeader AS soh
ON
    sod.SalesOrderID = soh.SalesOrderID
```

```
GROUP BY
    pc.Name, ps.Name, p.Name
ORDER BY
    TotalSalesAmount DESC;
```

--Back to Business scenario Q150

```
--Q151. Sample Solution

SELECT
    wo.WorkOrderID,
    wo.ProductID,
    loc.LocationID,
    loc.Name AS LocationName,
    wr.ScheduledStartDate,
    wr.ActualStartDate,
    wr.ScheduledEndDate,
    wr.ActualEndDate,
    wr.ActualResourceHrs
FROM
    Production.WorkOrder AS wo
INNER JOIN
    Production.WorkOrderRouting AS wr
ON
    wo.WorkOrderID = wr.WorkOrderID
INNER JOIN
    Production.Location AS loc
ON
    wr.LocationID = loc.LocationID
```

--Back to Business scenario Q151

```
--Q152. Sample Solution

SELECT
    sp.BusinessEntityID AS SalesPersonID,
    sqh.QuotaDate,
    sqh.SalesQuota
FROM
    Sales.SalesPerson AS sp
INNER JOIN
    Sales.SalesPersonQuotaHistory AS sqh
ON
    sp.BusinessEntityID = sqh.BusinessEntityID
ORDER BY
    sp.BusinessEntityID, sqh.QuotaDate;
```

--Back to Business scenario Q152

```
--Q153. Sample Solution

SELECT
    p.ProductID,
    p.Name AS ProductName,
    p.ProductNumber,
    COUNT(sci.ShoppingCartItemID) AS TimesAddedToCart
FROM
    Production.Product AS p
LEFT JOIN
    Sales.ShoppingCartItem AS sci
ON
    p.ProductID = sci.ProductID
GROUP BY
    p.ProductID,
    p.Name,
    p.ProductNumber
ORDER BY
    TimesAddedToCart DESC;

--Back to Business scenario Q153



--Q154. Sample Solution
SELECT
   c.CustomerID,
    c.FirstName,
    c.LastName,
    c.Email,
    c.PhoneNumber,
    a.AccountID,
    a.AccountType,
    a.Balance
FROM
    Finance.UniqueCustomers c
INNER JOIN
    Finance.UniqueAccounts a ON c.CustomerID = a.CustomerID
WHERE
    a.AccountType = 'Savings' -- Specify the desired account type
    AND a.Balance >= 5000; -- Specify the desired balance threshold

--Back to Business scenario Q154


--Q155. Sample Solution

SELECT
    c.CustomerID,
    c.FirstName,
    c.LastName,
    a.AccountID,
    a.AccountType,
    a.Balance,
    t.TransactionDate,
    t.Amount
FROM
    Finance.UniqueCustomers c
INNER JOIN
    Finance.UniqueAccounts a ON c.CustomerID = a.CustomerID
INNER JOIN
    Finance.UniqueTransactions t ON a.AccountID = t.AccountID
WHERE
    (a.Balance < 0)
    OR
    (t.Amount < 1000)
    OR
    (t.Amount > 5000);
```

--Back to Business scenario Q155

```
--Q156. Sample Solution

SELECT
    c.CustomerID,
    c.FirstName,
    c.LastName,

    AVG(a.Balance) AS AverageBalance,
    CASE
        WHEN AVG(a.Balance) >= 5000 THEN 'High-Value Customer'
        WHEN AVG(a.Balance) >= 1000 AND AVG(a.Balance) < 5000 THEN 'Medium-Value
          Customer'
        ELSE 'Low-Value Customer'
    END AS CustomerSegment
FROM
    Finance.UniqueCustomers c
INNER JOIN
    Finance.UniqueAccounts a ON c.CustomerID = a.CustomerID
GROUP BY
    c.CustomerID, c.FirstName, c.LastName
```

--Back to Business scenario Q156

```
--Q157. Sample Solution

-- Count the number of customers for each account type
SELECT
    A.AccountType,
    COUNT(DISTINCT C.CustomerID) AS NumberOfCustomers
FROM
    Finance.UniqueAccounts A
INNER JOIN
    Finance.UniqueCustomers C ON A.CustomerID = C.CustomerID
GROUP BY
    A.AccountType
ORDER BY
    A.AccountType;

-- Calculate the average balance for each account type
SELECT
    A.AccountType,
    AVG(A.Balance) AS AverageBalance
FROM
    Finance.UniqueAccounts A
GROUP BY
    A.AccountType
ORDER BY
    A.AccountType;
```

--Back to Business scenario Q157

```
--Q158. Sample Solution

DECLARE @Threshold DECIMAL(10, 2) = 1000.00;

-- Identify accounts at risk of overdrawing
SELECT
    C.CustomerID,
    C.FirstName,
    C.LastName,
    A.AccountID,
    A.AccountType,
    A.Balance
FROM
    Finance.UniqueAccounts A
INNER JOIN
    Finance.UniqueCustomers C ON A.CustomerID = C.CustomerID
WHERE
    A.Balance < @Threshold;
```

--Back to Business scenario Q158

```
--Q159. Sample Solution

SELECT
    C.CustomerID,
    C.FirstName,
    C.LastName,
    A.AccountID,
    A.AccountType,
    A.Balance
FROM
    Finance.UniqueCustomers C
INNER JOIN
    Finance.UniqueAccounts A ON C.CustomerID = A.CustomerID
LEFT JOIN
    Finance.UniqueTransactions T ON A.AccountID = T.AccountID
GROUP BY
    C.CustomerID, C.FirstName, C.LastName, A.AccountID, A.AccountType, A.Balance
HAVING
    MAX(T.TransactionDate) IS NULL OR
    MAX(T.TransactionDate) < DATEADD(MONTH, -48, GETDATE());
```

--Back to Business scenario Q159

```
--Q160. Sample Solution

DECLARE @StartDate DATE = '2022-01-01';
DECLARE @EndDate DATE = '2023-12-31';

SELECT
    B.BranchID,
    B.BranchName,
    B.Location,
    COUNT(T.TransactionID) AS TransactionCount
FROM
    Finance.UniqueBranches B
LEFT JOIN
    Finance.UniqueAccountBranches AB ON B.BranchID = AB.BranchID
    JOIN Finance.UniqueTransactions T ON AB.AccountID = T.AccountID
WHERE
    T.TransactionDate BETWEEN @StartDate AND @EndDate
GROUP BY
    B.BranchID, B.BranchName, B.Location
ORDER BY
    TransactionCount DESC

--Back to Business scenario Q160


--Q161. Sample Solution

SELECT
    B.BranchID,
    B.BranchName,
    B.Location,
    COUNT(AB.AccountID) AS AccountCount
FROM
    Finance.UniqueBranches B
LEFT JOIN
    Finance.UniqueAccountBranches AB ON B.BranchID = AB.BranchID
GROUP BY
    B.BranchID, B.BranchName, B.Location
ORDER BY
    AccountCount DESC

--Back to Business scenario Q161


--Q162. Sample Solution
SELECT
    B.BranchName,
    B.Location,
    B.City,
    SUM(A.Balance) AS TotalBalance
FROM
    Finance.UniqueBranches B
LEFT JOIN
    Finance.UniqueAccountBranches AB ON B.BranchID = AB.BranchID
LEFT JOIN
    Finance.UniqueAccounts A ON AB.AccountID = A.AccountID
GROUP BY
    B.BranchName, B.Location, B.City
ORDER BY
    B.BranchName

--Back to Business scenario Q162


--Q163. Sample Solution

SELECT
```

```
    A.AppointmentID,
    P.FirstName AS PatientFirstName,
    P.LastName AS PatientLastName,
    P.DateOfBirth AS PatientDateOfBirth,
    P.Gender AS PatientGender,
    P.PhoneNumber AS PatientPhoneNumber,
    P.Email AS PatientEmail,
    D.FirstName AS DoctorFirstName,
    D.LastName AS DoctorLastName,
    D.Specialisation AS DoctorSpecialisation,
    D.PhoneNumber AS DoctorPhoneNumber,
    D.Email AS DoctorEmail,
    A.AppointmentDateTime,
    A.Purpose
FROM
    Medical.Appointments A
INNER JOIN
    Medical.Patients P ON A.PatientID = P.PatientID
INNER JOIN
    Medical.Doctors D ON A.DoctorID = D.DoctorID
ORDER BY
    A.AppointmentDateTime;

--Back to Business scenario Q163


--Q164. Sample Solution

SELECT
    P.FirstName AS PatientFirstName,
    P.LastName AS PatientLastName,
    P.DateOfBirth AS PatientDateOfBirth,
    P.Gender AS PatientGender,
    P.PhoneNumber AS PatientPhoneNumber,
    P.Email AS PatientEmail,
    D.FirstName AS DoctorFirstName,
    D.LastName AS DoctorLastName,
    D.Specialisation AS DoctorSpecialisation,
    A.AppointmentDateTime,
    A.Purpose AS AppointmentPurpose,
    M.MedicationName,
    M.Dosage,
    M.PrescriptionDate,
    M.ExpiryDate
FROM
    Medical.Patients P
INNER JOIN
    Medical.Appointments A ON P.PatientID = A.PatientID
INNER JOIN
    Medical.Doctors D ON A.DoctorID = D.DoctorID
LEFT JOIN
    Medical.Medications M ON A.PatientID = M.PatientID
ORDER BY
    A.AppointmentDateTime, M.PrescriptionDate;

--Back to Business scenario Q164
```

```sql
--Q165. Sample Solution

SELECT
    P.FirstName AS PatientFirstName,
    P.LastName AS PatientLastName,
    P.DateOfBirth AS PatientDateOfBirth,
    P.Gender AS PatientGender,
    P.PhoneNumber AS PatientPhoneNumber,
    P.Email AS PatientEmail,
    D.FirstName AS DoctorFirstName,
    D.LastName AS DoctorLastName,
    D.Specialisation AS DoctorSpecialisation,
    M.MedicationName,
    M.Dosage,
    M.PrescriptionDate,
    M.ExpiryDate
FROM
    Medical.Patients P
INNER JOIN
    Medical.Medications M ON P.PatientID = M.PatientID
INNER JOIN
    Medical.Doctors D ON M.PrescribingDoctorID = D.DoctorID
ORDER BY
    P.PatientID, M.PrescriptionDate;
```

--Back to Business scenario Q165

```sql
--Q166. Sample Solution

SELECT PatientID, FirstName, LastName, DateOfBirth, Gender, PhoneNumber, Email
FROM Medical.Patients
WHERE PatientID IN (
    SELECT PatientID
    FROM Medical.Medications
    WHERE MedicationName = 'Surgeonix'
);
```

--Back to Business scenario Q166

```sql
--Q167. Sample Solution

SELECT MedicationName, COUNT(*) AS NumPrescriptions
FROM Medical.Medications
WHERE PatientID IN (
    SELECT PatientID
    FROM Medical.Patients
    WHERE DateOfBirth < '2004-01-01'
)
GROUP BY MedicationName
ORDER BY NumPrescriptions DESC;
```

--Back to Business scenario Q167

```sql
--Q168. Sample Solution
```

```
SELECT PrescribingDoctorID, COUNT(*) AS NumPrescriptions
FROM Medical.Medications
WHERE PatientID IN (
    SELECT PatientID
    FROM Medical.Appointments
    WHERE Purpose = 'Child infections'
)
GROUP BY PrescribingDoctorID
ORDER BY NumPrescriptions DESC;

--Back to Business scenario Q168


--Q169. Sample Solution

SELECT PrescribingDoctorID, COUNT(*) AS NumPrescriptions
FROM Medical.Medications
WHERE MedicationName = 'Meditrex'
GROUP BY PrescribingDoctorID
ORDER BY NumPrescriptions DESC;

--Back to Business scenario Q169


--Q170. Sample Solution

SELECT d.DoctorID, AVG(m.MedicationID) AS AvgMedications
FROM Medical.Doctors d
JOIN Medical.Appointments a ON d.DoctorID = a.DoctorID
JOIN Medical.Medications m ON a.PatientID = m.PatientID
GROUP BY d.DoctorID
HAVING AVG(m.MedicationID) > (SELECT AVG(m.MedicationID)
FROM Medical.Medications m);

--Back to Business scenario Q170


--Q171. Sample Solution

SELECT p.PatientID, p.FirstName, p.LastName
FROM Medical.Patients p
INNER JOIN Medical.Appointments a
ON p.PatientID = a.PatientID
WHERE a.AppointmentDateTime > '2023-09-01';

--Back to Business scenario Q171


--Q172. Sample Solution

SELECT p.PatientID, p.FirstName, p.LastName
FROM Medical.Patients p
INNER JOIN Medical.Appointments a
ON p.PatientID = a.PatientID
WHERE a.Purpose = 'Joint pain'
AND a.AppointmentDateTime > '2022-09-01';
```

--Back to Business scenario Q172

```
--Q173. Sample Solution

SELECT p.PatientID, p.FirstName, p.LastName, m.MedicationName
FROM Medical.Patients p
INNER JOIN Medical.Medications m
ON p.PatientID = m.PatientID
INNER JOIN Medical.Appointments a
ON p.PatientID = a.PatientID
WHERE m.MedicationName = 'PainAway'
AND a.Purpose = 'Child infections'
AND a.AppointmentDateTime > '2013-09-01';
```

--Back to Business scenario Q173

```
--Q174. Sample Solution

SELECT
    DATENAME(dw, A.AppointmentDateTime) AS DayOfWeek,
    DATEPART(HOUR, A.AppointmentDateTime) AS HourOfDay,
    COUNT(*) AS AppointmentCount
FROM Medical.Appointments A
GROUP BY DATENAME(dw, A.AppointmentDateTime), DATEPART(HOUR, A.AppointmentDateTime)
ORDER BY COUNT(*) DESC;
```

--Back to Business scenario Q174

```
--Q175. Sample Solution

SELECT
    M.MedicationName,
    M.ExpiryDate,
    D.FirstName AS PrescribingDoctorFirstName,
    D.LastName AS PrescribingDoctorLastName,
    P.FirstName AS PatientFirstName,
    P.LastName AS PatientLastName
FROM Medical.Medications M
INNER JOIN Medical.Doctors D ON M.PrescribingDoctorID = D.DoctorID
INNER JOIN Medical.Patients P ON M.PatientID = P.PatientID
WHERE M.ExpiryDate >= DATEADD(YEAR, 1, GETDATE());
```

--Back to Business scenario Q175

```
--Q176. Sample Solution

SELECT
    D.FirstName AS DoctorFirstName,
    D.LastName AS DoctorLastName,
    COUNT(A.AppointmentID) AS NumberOfAppointments,
    DATEPART(WEEKDAY, A.AppointmentDateTime) AS DayOfWeek
FROM Medical.Doctors D
INNER JOIN Medical.Appointments A ON D.DoctorID = A.DoctorID
```

```
GROUP BY D.FirstName, D.LastName, DATEPART(WEEKDAY, A.AppointmentDateTime)
ORDER BY DoctorLastName, DayOfWeek;
```

--Back to Business scenario Q176

```
--Q177. Sample Solution

DECLARE @AnalysisYear INT = 2012; -- Specify the year for analysis

SELECT
    E.FirstName + ' ' + E.LastName AS SalesRepresentative,
    SUM(FIS.SalesAmount) AS TotalSalesAmount
FROM
    dbo.FactResellerSales AS FIS
INNER JOIN
    dbo.DimEmployee AS E
    ON FIS.EmployeeKey = E.EmployeeKey
WHERE
    DATEPART(YEAR, FIS.OrderDate) = @AnalysisYear
GROUP BY
    E.FirstName, E.LastName
ORDER BY
    TotalSalesAmount DESC;
```

--Back to Business scenario Q177

```
--Q178. Sample Solution

SELECT
    DATEPART(YEAR, FIS.OrderDate) AS SalesYear,
    SUM(FIS.SalesAmount) AS TotalSalesAmount
FROM
    dbo.FactInternetSales AS FIS
GROUP BY
    DATEPART(YEAR, FIS.OrderDate)
ORDER BY
    SalesYear;
```

--Back to Business scenario Q178

```
--Q179. Sample Solution

SELECT
    C.CustomerKey,
    CONCAT(C.FirstName, ' ', C.LastName) AS CustomerName,
    SUM(F.SalesAmount) AS TotalSalesAmount
FROM
    dbo.FactInternetSales AS F
INNER JOIN dbo.DimCustomer AS C ON F.CustomerKey = C.CustomerKey
GROUP BY
    C.CustomerKey, C.FirstName, C.LastName
ORDER BY
    TotalSalesAmount DESC;
```

--Back to Business scenario Q179

```
--Q180. Sample Solution

SELECT
    C.CityName,
    COUNT(Cu.CustomerID) AS CustomerCount
FROM
    Application.Cities AS C
LEFT JOIN
    Sales.Customers AS Cu ON C.CityID = Cu.DeliveryCityID
GROUP BY
    C.CityName
ORDER BY
    CustomerCount DESC;
```

--Back to Business scenario Q180

```
--Q181. Sample Solution

SELECT
    SP.StateProvinceName,
    COUNT(Cu.CustomerID) AS CustomerCount
FROM
    Application.StateProvinces AS SP
LEFT JOIN
    Application.Cities AS C ON SP.StateProvinceID = C.StateProvinceID
LEFT JOIN
    Sales.Customers AS Cu ON C.CityID = Cu.DeliveryCityID
GROUP BY
    SP.StateProvinceName
ORDER BY
    CustomerCount DESC;
```

--Back to Business scenario Q181

```
--Q182. Sample Solution

SELECT
    DM.DeliveryMethodName,
    P.FullName AS ContactName,
    P.EmailAddress
FROM
    Application.DeliveryMethods AS DM
JOIN
    Application.People AS P ON DM.DeliveryMethodID = P.PersonID
ORDER BY
    DM.DeliveryMethodName;
```

--Back to Business scenario Q182

```
--Q183. Sample Solution
```

```sql
SELECT
    S.SupplierName,
    COUNT(DISTINCT PO.PurchaseOrderID) AS TotalPurchaseOrders,
    COUNT(DISTINCT I.InvoiceID) AS TotalInvoices
FROM
    Purchasing.Suppliers AS S
LEFT JOIN
    Purchasing.PurchaseOrders AS PO ON S.SupplierID = PO.SupplierID
LEFT JOIN
    Sales.Invoices AS I ON S.DeliveryMethodID = I.DeliveryMethodID
GROUP BY
    S.SupplierName
ORDER BY
    S.SupplierName;

--Back to Business scenario Q183


--Q184. Sample Solution

SELECT
    PM.PaymentMethodName,
    SUM(CASE WHEN ST.SupplierTransactionID IS NOT NULL THEN 1 ELSE 0 END) AS
      SupplierTransactionCount,
    SUM(CASE WHEN CT.CustomerTransactionID IS NOT NULL THEN 1 ELSE 0 END) AS
      CustomerTransactionCount
FROM
    Application.PaymentMethods AS PM
LEFT JOIN
    Purchasing.SupplierTransactions AS ST ON PM.PaymentMethodID =
      ST.PaymentMethodID
LEFT JOIN
    Sales.CustomerTransactions AS CT ON PM.PaymentMethodID = CT.PaymentMethodID
GROUP BY
    PM.PaymentMethodName
ORDER BY
    SupplierTransactionCount DESC, CustomerTransactionCount DESC;

--Back to Business scenario Q184



--Q185. Sample Solution

SELECT
    PM.PaymentMethodName,
    YEAR(ST.TransactionDate) AS TransactionYear,
    SUM(CASE WHEN ST.SupplierTransactionID IS NOT NULL THEN 1 ELSE 0 END) AS
      SupplierTransactionCount,
    SUM(CASE WHEN CT.CustomerTransactionID IS NOT NULL THEN 1 ELSE 0 END) AS
      CustomerTransactionCount
FROM
    Application.PaymentMethods AS PM
LEFT JOIN
    Purchasing.SupplierTransactions AS ST ON PM.PaymentMethodID =
      ST.PaymentMethodID
LEFT JOIN
```

```
    Sales.CustomerTransactions AS CT ON PM.PaymentMethodID = CT.PaymentMethodID
GROUP BY
    PM.PaymentMethodName, YEAR(ST.TransactionDate)
ORDER BY
    TransactionYear, SupplierTransactionCount DESC, CustomerTransactionCount DESC;

--Back to Business scenario Q185


--Q186. Sample Solution

SELECT
    C.ColorName, COUNT(SI.StockItemID)
    AS ItemCount
FROM
    Warehouse.Colors AS C
LEFT JOIN
    Warehouse.StockItems AS SI
ON
    C.ColorID = SI.ColorID
GROUP BY
    C.ColorName
ORDER BY
    ItemCount DESC;

--Back to Business scenario Q186


--Q187. Sample Solution

SELECT
     C.ColorName,
    COUNT(SI.StockItemID) AS ItemCount,
    (COUNT(SI.StockItemID) * 100.0 / SUM(COUNT(SI.StockItemID)) OVER ()) AS
      Percentage
FROM
    Warehouse.Colors AS C
LEFT JOIN
    Warehouse.StockItems AS SI
ON
    C.ColorID = SI.ColorID
GROUP BY
    C.ColorName
ORDER BY
    ItemCount DESC;

--Back to Business scenario Q187


--Q188. Sample Solution

SELECT
    PT.PackageTypeName,
    AVG(SI.QuantityPerOuter) AS AverageOrderLineQuantity
FROM
    Warehouse.PackageTypes AS PT
LEFT JOIN
    Warehouse.StockItems AS SI
```

```
ON
    PT.PackageTypeID = SI.UnitPackageID
LEFT JOIN
    Purchasing.PurchaseOrderLines AS POL
ON
    SI.StockItemID = POL.StockItemID
GROUP BY
    PT.PackageTypeName
ORDER BY
    AverageOrderLineQuantity DESC;

--Back to Business scenario Q188


--Q189. Sample Solution

SELECT
    PT.PackageTypeName,
    SUM(SI.QuantityPerOuter) AS PurchaseOrderQuantity,
    SUM(IL.Quantity) AS SalesInvoiceQuantity
FROM
    Warehouse.PackageTypes AS PT
LEFT JOIN
    Warehouse.StockItems AS SI ON PT.PackageTypeID = SI.UnitPackageID
LEFT JOIN
    Purchasing.PurchaseOrderLines AS POL ON SI.StockItemID = POL.StockItemID
LEFT JOIN
    Sales.InvoiceLines AS IL ON SI.StockItemID = IL.StockItemID
GROUP BY
    PT.PackageTypeName
ORDER BY
    PT.PackageTypeName;

--Back to Business scenario Q189


--Q190. Sample Solution

SELECT
    SG.StockGroupName,
    SI.StockItemName,
    P.FullName AS PersonName
FROM
    Warehouse.StockItemStockGroups AS SISG
INNER JOIN
    Warehouse.StockGroups AS SG ON SISG.StockGroupID = SG.StockGroupID
INNER JOIN
    Warehouse.StockItems AS SI ON SISG.StockItemID = SI.StockItemID
LEFT JOIN
    Application.People AS P ON SG.StockGroupID = P.PersonID
ORDER BY
    SG.StockGroupName, SI.StockItemName;

--Back to Business scenario Q190



--Q191. Sample Solution
```

```
SELECT

    SG.StockGroupName,
    SI.StockItemName,
    P.FullName AS AssignedBy,
    SS.LastEditedWhen AS AssignmentDate
FROM
    Warehouse.StockItemStockGroups AS SS
JOIN
    Warehouse.StockGroups AS SG ON SS.StockGroupID = SG.StockGroupID
JOIN
    Warehouse.StockItems AS SI ON SS.StockItemID = SI.StockItemID
JOIN
    Application.People AS P ON SS.LastEditedBy = P.PersonID
ORDER BY
    SS.LastEditedWhen DESC;

--Back to Business scenario Q191


--Q192. Sample SolutionWITH

DemandForecast AS (
    SELECT
        P.ProductID,
        P.Name AS ProductName,
        YEAR(SOH.OrderDate) AS SalesYear,
        MONTH(SOH.OrderDate) AS SalesMonth,
        SUM(SOD.OrderQty) AS TotalOrderQuantity
    FROM Production.Product P
    LEFT JOIN Sales.SalesOrderDetail SOD ON P.ProductID = SOD.ProductID
    LEFT JOIN Sales.SalesOrderHeader SOH ON SOD.SalesOrderID = SOH.SalesOrderID
    WHERE SOH.OrderDate >= DATEADD(YEAR, -12, GETDATE())  -- Forecast for the last ↩
      12 months
    GROUP BY P.ProductID, P.Name, YEAR(SOH.OrderDate), MONTH(SOH.OrderDate)
)
-- Now calculate monthly demand forecasts
SELECT
    ProductID,
    ProductName,
    SalesYear,
    SalesMonth,
    SUM(TotalOrderQuantity) AS DemandForecast
FROM DemandForecast
GROUP BY ProductID, ProductName, SalesYear, SalesMonth
ORDER BY ProductID, SalesYear, SalesMonth;

--Back to Business scenario Q192


--Q193. Sample Solution

-- Evaluate potential profitability of product line expansion
WITH ProductExpansionAnalysis AS (
    SELECT
        P.ProductID,
        P.Name AS ProductName,
        AVG(P.StandardCost) AS AvgStandardCost,
        AVG(P.ListPrice) AS AvgListPrice,
        SUM(SOD.OrderQty) AS TotalUnitsSold,
        SUM(SOD.LineTotal) AS TotalRevenue
    FROM Production.Product P
    LEFT JOIN Sales.SalesOrderDetail SOD ON P.ProductID = SOD.ProductID
    GROUP BY P.ProductID, P.Name
)
-- Calculate profitability metrics
SELECT
    ProductID,
    ProductName,
    AvgStandardCost,
    AvgListPrice,
    TotalUnitsSold,
    TotalRevenue,
    (TotalRevenue - (AvgStandardCost * TotalUnitsSold)) AS GrossProfit,
    (TotalRevenue - (AvgStandardCost * TotalUnitsSold)) * 100.0 / TotalRevenue AS 
      GrossMarginPercentage
FROM ProductExpansionAnalysis
ORDER BY GrossProfit DESC;

--Back to Business scenario Q193


--Q194. Sample Solution

 -- Analyze product return data to identify common reasons for returns
WITH ProductReturnAnalysis AS (
    SELECT
        P.ProductID,
        P.Name AS ProductName,
        COUNT(SO.SalesOrderID) AS TotalOrders,
        SUM(SOD.OrderQty) AS TotalUnitsSold,
        COUNT(DISTINCT PR.ProductReviewID) AS TotalReviews,
        SUM(CASE WHEN SO.OrderDate >= '2014-01-01' THEN 1 ELSE 0 END) AS 
          RecentOrders
    FROM Production.Product P
    LEFT JOIN Sales.SalesOrderDetail SOD ON P.ProductID = SOD.ProductID
    LEFT JOIN Sales.SalesOrderHeader SO ON SOD.SalesOrderID = SO.SalesOrderID
    LEFT JOIN Production.ProductReview PR ON P.ProductID = PR.ProductID
    WHERE SO.OrderDate >= '2009-01-01'
    GROUP BY P.ProductID, P.Name
)
-- Retrieve product return analysis results
SELECT
    ProductID,
    ProductName,
    TotalOrders,
    TotalUnitsSold,
    TotalReviews,
    RecentOrders,
    CASE
        WHEN TotalUnitsSold = 0 THEN 0
        ELSE (TotalOrders - RecentOrders) * 100.0 / TotalUnitsSold
    END AS ReturnRatePercentage
FROM ProductReturnAnalysis
ORDER BY ReturnRatePercentage DESC;
```

--Back to Business scenario Q194

```sql
--Q195. Sample Solution

-- Analyze website traffic and conversion rates for e-commerce optimization
WITH WebsiteConversion AS (
SELECT
        YEAR(OH.OrderDate) AS SalesYear,
        COUNT (DISTINCT OH.CustomerID) AS TotalCustomers,
        COUNT(SO.SalesOrderID) AS TotalOrders,
        100.0 * COUNT(SO.SalesOrderID) / COUNT(DISTINCT OH.CustomerID) AS
          ConversionRate
    FROM Sales.SalesOrderHeader OH
    LEFT JOIN Sales.SalesOrderHeader SO ON OH.CustomerID = SO.CustomerID
    WHERE YEAR(OH.OrderDate) >= YEAR(GETDATE()) - 20
    GROUP BY YEAR(OH.OrderDate)
)
-- Retrieve website conversion analysis results
SELECT
    SalesYear,
    TotalCustomers,
    TotalOrders,
    ConversionRate
FROM WebsiteConversion
ORDER BY SalesYear;
```

--Back to Business scenario Q195

```sql
--Q196. Sample Solution

-- Predict future sales using historical data (in years)
WITH SalesForecast AS (
    SELECT
        YEAR(OH.OrderDate) AS SalesYear,
        SUM(OH.TotalDue) AS AnnualSales
    FROM Sales.SalesOrderHeader OH
    WHERE OH.OrderDate >= DATEADD(YEAR, -10, GETDATE())
    GROUP BY YEAR(OH.OrderDate)
)
-- Retrieve sales forecast for the next 2 years
SELECT
    YEAR(GETDATE ()) AS ForecastYear,
    SF.SalesYear,
    SF.AnnualSales
FROM SalesForecast SF
WHERE SF.SalesYear BETWEEN YEAR(GETDATE()) AND YEAR(GETDATE()) -8
ORDER BY SF.SalesYear;
```

--Back to Business scenario Q196

```sql
--Q197. Sample Solution

-- Analyze product quality based on customer reviews
WITH ProductQualityAnalysis AS (
    SELECT
```

```
        P.ProductID,
        P.Name AS ProductName,
        AVG(PR.Rating) AS AverageRating,
        COUNT(PR.ProductReviewID) AS TotalReviews,
        SUM(CASE WHEN PR.Rating >= 4 THEN 1 ELSE 0 END) AS PositiveReviews,
        SUM(CASE WHEN PR.Rating < 4 THEN 1 ELSE 0 END) AS NegativeReviews
    FROM Production.Product P
    LEFT JOIN Production.ProductReview PR ON P.ProductID = PR.ProductID
    GROUP BY P.ProductID, P.Name
)
-- Retrieve product quality analysis results
SELECT
    ProductID,
    ProductName,
    AverageRating,
    TotalReviews,
    PositiveReviews,
    NegativeReviews,
    CASE
        WHEN AverageRating >= 4.5 THEN 'Excellent'
        WHEN AverageRating >= 4.0 THEN 'Good'
        WHEN AverageRating >= 3.0 THEN 'Average'
        ELSE 'Below Average'
    END AS QualityRating
FROM ProductQualityAnalysis
ORDER BY AverageRating DESC;
```

--Back to Business scenario Q197

```
--Q198. Sample Solution

-- Analyze customer retention rates and factors influencing retention
WITH CustomerRetention AS (
    SELECT
        C.CustomerID,
        MIN(OH.OrderDate) AS FirstPurchaseDate,
        MAX(OH.OrderDate) AS LatestPurchaseDate,
        DATEDIFF(DAY, MIN(OH.OrderDate), MAX(OH.OrderDate)) AS
          DaysSinceFirstPurchase,
        CASE
            WHEN COUNT(OH.SalesOrderID) > 1 THEN 'Returning Customer'
            ELSE 'New Customer'
        END AS CustomerType
    FROM Sales.Customer C
    LEFT JOIN Sales.SalesOrderHeader OH ON C.CustomerID = OH.CustomerID
    GROUP BY C.CustomerID
)
-- Calculate customer retention rates and analyze factors
SELECT
    CustomerType,
    COUNT(CustomerID) AS TotalCustomers,
    SUM(CASE WHEN DaysSinceFirstPurchase > 365 THEN 1 ELSE 0 END) AS
      RetainedCustomers,
    100.0 * SUM(CASE WHEN DaysSinceFirstPurchase > 365 THEN 1 ELSE 0 END) / COUNT
      (CustomerID) AS RetentionRate
FROM CustomerRetention
GROUP BY CustomerType;
```

--Back to Business scenario Q198

```
--Q199. Sample Solution

WITH PromotionAnalysis AS (
    SELECT
        SO.SpecialOfferID,
        SO.Description AS PromotionDescription,
        SO.DiscountPct,
        COUNT(DISTINCT OH.CustomerID) AS CustomersAcquired,
        SUM(OH.TotalDue) AS TotalSalesAmount
    FROM Sales.SpecialOffer SO
    LEFT JOIN Sales.SalesOrderDetail SOD ON SO.SpecialOfferID = SOD.SpecialOfferID
    JOIN SALES.SalesOrderHeader OH ON SOD.SalesOrderID = OH.SalesOrderID
    GROUP BY SO.SpecialOfferID, SO.Description, SO.DiscountPct
)
-- Retrieve promotion effectiveness information
SELECT
    SpecialOfferID,
    PromotionDescription,
    DiscountPct,
    CustomersAcquired,
    TotalSalesAmount
FROM PromotionAnalysis
ORDER BY TotalSalesAmount DESC;
```

--Back to Business scenario Q199

```
--Q200. Sample Solution

-- Calculate Customer Lifetime Value (CLV) for each customer
WITH CustomerCLV AS (
    SELECT
        C.CustomerID,
        COUNT(OH.SalesOrderID) AS TotalOrders,
        SUM(OH.TotalDue) AS TotalSpent,
        DATEDIFF(DAY, MIN(OH.OrderDate), MAX(OH.OrderDate)) AS DaysAsCustomer,
        CASE
            WHEN COUNT(OH.SalesOrderID) > 0 AND DATEDIFF(DAY, MIN(OH.OrderDate),
              MAX(OH.OrderDate)) > 0 THEN
                SUM(OH.TotalDue) / NULLIF(DATEDIFF(DAY, MIN(OH.OrderDate), MAX
                  (OH.OrderDate)), 0)
            ELSE
                0
        END AS AverageDailySpend,
        SUM(OH.TotalDue) / NULLIF(COUNT(OH.SalesOrderID), 0) AS AverageOrderValue,
        SUM(OH.TotalDue) * 365 / NULLIF(DATEDIFF(DAY, MIN(OH.OrderDate), MAX
          (OH.OrderDate)), 0) AS AnnualCLV
    FROM Sales.Customer C
    LEFT JOIN Sales.SalesOrderHeader OH ON C.CustomerID = OH.CustomerID
    GROUP BY C.CustomerID
)
-- Retrieve Customer Lifetime Value (CLV) information
SELECT
    CustomerID,
    TotalOrders,
    TotalSpent,
    DaysAsCustomer,
    AverageDailySpend,
    AverageOrderValue,
    AnnualCLV
FROM CustomerCLV
ORDER BY AnnualCLV DESC;

--Back to Business scenario Q200


--Q201. Sample Solution

-- Determine the life cycle stage of products
WITH ProductLifeCycle AS (
    SELECT
        P.ProductID,
        P.Name AS ProductName,
        P.SellStartDate,
        P.DiscontinuedDate,
        CASE
            WHEN P.SellStartDate IS NULL THEN 'Unknown'
            WHEN P.DiscontinuedDate IS NULL THEN 'Introduction'
            WHEN P.DiscontinuedDate IS NOT NULL AND DATEDIFF(DAY, P.SellStartDate, ↩
              P.DiscontinuedDate) <= 365 THEN 'Growth'
            WHEN P.DiscontinuedDate IS NOT NULL AND DATEDIFF(DAY, P.SellStartDate, ↩
              P.DiscontinuedDate) <= 1095 THEN 'Maturity'
            ELSE 'Decline'
        END AS LifeCycleStage
    FROM Production.Product P
)
-- Retrieve product life cycle information
SELECT
    ProductID,
    ProductName,
    SellStartDate,
    DiscontinuedDate,
    LifeCycleStage
FROM ProductLifeCycle
ORDER BY ProductID;

--Back to Business scenario Q201


--Q202. Sample Solution

-- Rank employees based on their productivity (e.g., total sales generated)
WITH EmployeeProductivity AS (
    SELECT
        E.BusinessEntityID AS EmployeeID,
        COUNT(SOH.SalesOrderID) AS TotalSalesOrders
    FROM Sales.SalesPerson SP
    JOIN HumanResources.Employee E ON SP.BusinessEntityID = E.BusinessEntityID
    LEFT JOIN Sales.SalesOrderHeader SOH ON SP.BusinessEntityID = SOH.SalesPersonID
    GROUP BY E.BusinessEntityID
)
-- Rank employees by productivity in descending order
```

```
SELECT
    EmployeeID,
    TotalSalesOrders,
    RANK() OVER (ORDER BY TotalSalesOrders DESC) AS ProductivityRank
FROM EmployeeProductivity
ORDER BY ProductivityRank;
```

--Back to Business scenario Q202

```
--Q203. Sample Solution

WITH IncorrectAddressCustomers AS (
    SELECT
        H.CustomerID,
        A.City,
        A.StateProvinceID,
        A.PostalCode,
        COUNT(H.SalesOrderID) AS OrderCount
    FROM
        Sales.SalesOrderHeader AS H
    INNER JOIN
        Person.Address AS A
        ON H.ShipToAddressID = A.AddressID
    WHERE
        A.City IS NULL OR A.StateProvinceID IS NULL OR A.PostalCode IS NULL
    GROUP BY
        H.CustomerID, A.City, A.StateProvinceID, A.PostalCode
)
SELECT
    IAC.CustomerID,
    MAX(COALESCE(IAC.City, 'N/A')) AS City,
    MAX(COALESCE(IAC.StateProvinceID, 'N/A')) AS StateProvinceID,
    MAX(COALESCE(IAC.PostalCode, 'N/A')) AS PostalCode,
    MAX(IAC.OrderCount) AS OrderCount
FROM
    IncorrectAddressCustomers AS IAC
GROUP BY
    IAC.CustomerID
ORDER BY
    OrderCount DESC;
```

--Back to Business scenario Q203

```
--Q204. Sample Solution

WITH RegionSales AS (
    SELECT
        A.StateProvinceID,
        SUM(OD.LineTotal) AS TotalSalesAmount,
        COUNT(DISTINCT H.SalesOrderID) AS OrderCount
    FROM
        Sales.SalesOrderHeader AS H
    INNER JOIN
        Sales.SalesOrderDetail AS OD
        ON H.SalesOrderID = OD.SalesOrderID
    INNER JOIN
```

```
        Person.Address AS A
        ON H.ShipToAddressID = A.AddressID
    GROUP BY
        A.StateProvinceID
)
SELECT
    R.StateProvinceID,
    MAX(SPA.Name) AS StateProvinceName,
    SUM(R.TotalSalesAmount) AS TotalSalesAmount,
    SUM(R.OrderCount) AS OrderCount
FROM
    RegionSales AS R
LEFT JOIN
    Person.StateProvince AS SPA
    ON R.StateProvinceID = SPA.StateProvinceID
GROUP BY
    R.StateProvinceID
ORDER BY
    TotalSalesAmount DESC;

--Back to Business scenario Q204


--Q205. Sample Solution

WITH DepartmentHistoryCTE AS (
    SELECT
        edh.DepartmentID,
        d.Name AS DepartmentName,
        edh.StartDate,
        edh.EndDate,
        DATEDIFF(day, edh.StartDate, edh.EndDate) AS DaysInDepartment
    FROM
        HumanResources.EmployeeDepartmentHistory AS edh
    INNER JOIN
        HumanResources.Department AS d ON edh.DepartmentID = d.DepartmentID
)
SELECT
    DepartmentName,
    AVG(DaysInDepartment)  AS  AvgDaysInDepartment,
    MAX(DaysInDepartment)  AS  MaxDaysInDepartment,
    MIN(DaysInDepartment) AS MinDaysInDepartment
FROM
    DepartmentHistoryCTE
GROUP BY
    DepartmentName
ORDER BY
    AvgDaysInDepartment DESC;

--Back to Business scenario Q205


--Q206. Sample Solution

WITH DepartmentChanges AS (
    SELECT
        edh.DepartmentID,
        d.Name AS DepartmentName,
        edh.StartDate,
        edh.EndDate,
        ROW_NUMBER() OVER (PARTITION BY edh.DepartmentID ORDER BY edh.StartDate) AS
           ChangeNumber
    FROM
        HumanResources.EmployeeDepartmentHistory AS edh
    INNER JOIN
        HumanResources.Department AS d ON edh.DepartmentID = d.DepartmentID
)
SELECT
    DepartmentName,
    COUNT(DISTINCT DepartmentID) AS TotalEmployees,
    MAX(ChangeNumber) AS MaxDepartmentChanges,
    AVG(ChangeNumber * 1.0) AS AvgDepartmentChanges
FROM
    DepartmentChanges
GROUP BY
    DepartmentName
ORDER BY
    AvgDepartmentChanges DESC;


--Back to Business scenario Q206


--Q207. Sample Solution

WITH DepartmentTenure AS (
    SELECT
        edh.BusinessEntityID,
        d.Name AS DepartmentName,
        DATEDIFF(YEAR, edh.StartDate, ISNULL(edh.EndDate, GETDATE())) AS
          TenureInYears
    FROM
        HumanResources.EmployeeDepartmentHistory AS edh
    INNER JOIN
        HumanResources.Department AS d ON edh.DepartmentID = d.DepartmentID
)
SELECT
    DepartmentName,
    AVG(TenureInYears) AS AvgTenureInYears
FROM
    DepartmentTenure
GROUP BY
    DepartmentName
ORDER BY
    AvgTenureInYears DESC;

--Back to Business scenario Q207


--Q208. Sample Solution

WITH ShiftPreferences AS (
    SELECT
        edh.BusinessEntityID,
        s.Name AS ShiftName,
        COUNT(*) AS ShiftCount
    FROM
        HumanResources.EmployeeDepartmentHistory AS edh
    INNER JOIN
        HumanResources.Shift AS s ON edh.ShiftID = s.ShiftID
    GROUP BY
        edh.BusinessEntityID, s.Name
)
SELECT
    ShiftName,
    COUNT(DISTINCT BusinessEntityID) AS EmployeeCount,
    SUM(ShiftCount) AS TotalShifts
FROM
    ShiftPreferences
GROUP BY
    ShiftName
ORDER BY
    TotalShifts DESC;

--Back to Business scenario Q208


--Q209. Sample Solution

-- Check which cultures are missing localized descriptions for product models
SELECT
    pm.ProductModelID,
    pm.Name AS ProductModelName,
    c.CultureID,
    c.Name AS CultureName,
    CASE
        WHEN pmpdc.ProductDescriptionID IS NOT NULL THEN 'Localized'
        ELSE 'Missing Localization'
    END AS LocalizationStatus
FROM
    Production.ProductModel AS pm
CROSS JOIN
    Production.Culture AS c
LEFT JOIN
    Production.ProductModelProductDescriptionCulture AS pmpdc
    ON pm.ProductModelID = pmpdc.ProductModelID
    AND c.CultureID = pmpdc.CultureID
ORDER BY
    pm.ProductModelID,
    c.CultureID;

--Back to Business scenario Q209


--Q210. Sample Solution

WITH InventoryAnalysis AS (
    SELECT
        p.ProductID,
        p.Name AS ProductName,
        SUM(pch.StandardCost * bom.PerAssemblyQty) AS InventoryValuation,
        SUM(wor.OrderQty) AS QuantityUsedInProduction,
        SUM(th.Quantity) AS QuantitySold
    FROM
```

```
        Production.Product AS p
    INNER JOIN
        Production.BillOfMaterials AS bom
    ON
        p.ProductID = bom.ProductAssemblyID
    LEFT JOIN
        Production.ProductCostHistory AS pch
    ON
        p.ProductID = pch.ProductID
    LEFT JOIN
        Production.WorkOrder AS wor
    ON
        p.ProductID = wor.ProductID
    LEFT JOIN
        Production.TransactionHistory AS th
    ON
        p.ProductID = th.ProductID
    WHERE
        th.TransactionType = 'S' -- Sales transaction
    GROUP BY
        p.ProductID, p.Name
)
SELECT
    ProductName,
    InventoryValuation,
    QuantityUsedInProduction,
    QuantitySold
FROM
    InventoryAnalysis;

--Back to Business scenario Q210


--Q211. Sample Solution

WITH ProductModelAnalysis AS (
    SELECT
        pm.ProductModelID,
        pm.Name AS ProductModelName,
        SUM(pch.StandardCost * bom.PerAssemblyQty) AS TotalCost,
        SUM(th.Quantity * plph.ListPrice) AS TotalRevenue
    FROM
        Production.ProductModel AS pm
    LEFT JOIN
        Production.Product AS p
    ON
        pm.ProductModelID = p.ProductModelID
    LEFT JOIN
        Production.BillOfMaterials AS bom
    ON
        p.ProductID = bom.ProductAssemblyID
    LEFT JOIN
        Production.ProductCostHistory AS pch
    ON
        p.ProductID = pch.ProductID
    LEFT JOIN
        Production.TransactionHistory AS th
    ON
```

```
        p.ProductID = th.ProductID
    LEFT JOIN
        Production.ProductListPriceHistory AS plph
    ON
        p.ProductID = plph.ProductID
    WHERE
        th.TransactionType = 'S' -- Sales transaction
    GROUP BY
        pm.ProductModelID, pm.Name
)
SELECT
    ProductModelName,
    TotalCost,
    TotalRevenue,
    (TotalRevenue - TotalCost) AS Profit
FROM
    ProductModelAnalysis;

--Back to Business scenario Q211


--Q212. Sample Solution

SELECT DISTINCT
    pdpc.CultureID,
    pdpc.CultureName,
    pdpc.DescriptionStatus
FROM (
    SELECT
        pmpdc.CultureID,
        c.Name AS CultureName,
        CASE
            WHEN pd.Description IS NULL THEN 'Missing'
            WHEN pd.Description = '' THEN 'Incomplete'
            ELSE 'Complete'
        END AS DescriptionStatus
    FROM
        Production.ProductModelProductDescriptionCulture AS pmpdc
    INNER JOIN
        Production.Culture AS c
    ON
        pmpdc.CultureID = c.CultureID
    LEFT JOIN
        Production.ProductDescription AS pd
    ON
        pmpdc.ProductDescriptionID = pd.ProductDescriptionID
) AS pdpc
ORDER BY
    pdpc.CultureName, pdpc.DescriptionStatus;

--Back to Business scenario Q212


--Q213. Sample Solution

WITH ScrapSummary AS (
    SELECT
        sr.ScrapReasonID,
        sr.Name AS ScrapReason,
        COUNT(wo.WorkOrderID) AS WorkOrderCount
    FROM
        Production.ScrapReason AS sr
    LEFT JOIN
        Production.WorkOrder AS wo
    ON
        sr.ScrapReasonID = wo.ScrapReasonID
    GROUP BY
        sr.ScrapReasonID, sr.Name
)
SELECT
    ss.ScrapReason,
    ss.WorkOrderCount
FROM
    ScrapSummary AS ss
ORDER BY
    ss.WorkOrderCount DESC;

--Back to Business scenario Q213


--Q214. Sample Solution

WITH ScrapRateSummary AS (
    SELECT
        p.ProductID,
        p.Name AS ProductName,
        sr.Name AS ScrapReason,
        COUNT(wo.WorkOrderID) AS ScrapCount,
        CASE
            WHEN COUNT(DISTINCT wo.WorkOrderID) = 0 THEN 0  -- Check for divide by 
              zero
            ELSE COUNT(wo.WorkOrderID) * 1.0 / COUNT(DISTINCT wo.WorkOrderID)
        END AS ScrapRate
    FROM
        Production.Product AS p
    LEFT JOIN
        Production.WorkOrder AS wo
    ON
        p.ProductID = wo.ProductID
    LEFT JOIN
        Production.ScrapReason AS sr
    ON
        wo.ScrapReasonID = sr.ScrapReasonID
    GROUP BY
        p.ProductID, p.Name, sr.Name
)
SELECT
    srs.ProductName,
    srs.ScrapReason,
    srs.ScrapRate
FROM
    ScrapRateSummary AS srs
WHERE
    srs.ScrapRate IS NOT NULL
ORDER BY
    srs.ProductName, srs.ScrapRate DESC;
```

--Back to Business scenario Q214

```
--Q215. Sample Solution

WITH ScrapCosts AS (
    SELECT
        p.ProductID,
        p.Name AS ProductName,
        ISNULL(sr.Name, 'No Scrap Reason') AS ScrapReason,
        SUM(ISNULL(wo.ScrappedQty, 0) * ISNULL(p.StandardCost, 0)) AS
          TotalScrapCost
    FROM
        Production.Product AS p
    LEFT JOIN
        Production.WorkOrder AS wo
    ON
        p.ProductID = wo.ProductID
    LEFT JOIN
        Production.ScrapReason AS sr
    ON
        wo.ScrapReasonID = sr.ScrapReasonID
    GROUP BY
        p.ProductID, p.Name, sr.Name
)
SELECT
    ProductName,
    ScrapReason,
    TotalScrapCost
FROM
    ScrapCosts
ORDER BY
    ProductName, TotalScrapCost DESC;
```

--Back to Business scenario Q215

```
--Q216. Sample Solution

WITH ProductionDelays AS (
    SELECT
        sr.Name AS ScrapReason,
        COUNT(DISTINCT wo.WorkOrderID) AS DelayedWorkOrders
    FROM
        Production.WorkOrder AS wo
    LEFT JOIN
        Production.ScrapReason AS sr
    ON
        wo.ScrapReasonID = sr.ScrapReasonID
    WHERE
        wo.EndDate > wo.DueDate
    GROUP BY
        sr.Name
)
SELECT
    ScrapReason,
    DelayedWorkOrders
```

```
FROM
    ProductionDelays
ORDER BY
    DelayedWorkOrders DESC;

--Back to Business scenario Q216


--Q217. Sample Solution

WITH WorkOrderRoutingTimes AS (
    SELECT
        wo.WorkOrderID,
        MAX(DATEDIFF(SECOND, wr.ActualStartDate, wr.ActualEndDate)) AS
          MaxRoutingTimeInSeconds
    FROM
        Production.WorkOrder AS wo
    INNER JOIN
        Production.WorkOrderRouting AS wr
    ON
        wo.WorkOrderID = wr.WorkOrderID
    GROUP BY
        wo.WorkOrderID
)
SELECT TOP 5
    wo.WorkOrderID,
    wort.MaxRoutingTimeInSeconds
FROM
    WorkOrderRoutingTimes AS wort
INNER JOIN
    Production.WorkOrder AS wo
ON
    wort.WorkOrderID = wo.WorkOrderID
ORDER BY
    wort.MaxRoutingTimeInSeconds DESC;

--Back to Business scenario Q217


--Q218. Sample Solution

WITH WorkOrderCompletion AS (
    SELECT
        wr.WorkOrderID,
        wr.OperationSequence,
        CASE
            WHEN wr.ActualEndDate <= wo.DueDate THEN 'On Time'
            ELSE 'Delayed'
        END AS CompletionStatus
    FROM
        Production.WorkOrder AS wo
    INNER JOIN
        Production.WorkOrderRouting AS wr
    ON
        wo.WorkOrderID = wr.WorkOrderID
)
SELECT
    OperationSequence,
    SUM(CASE WHEN CompletionStatus = 'On Time' THEN 1 ELSE 0 END) AS
      CompletedOnTime,
    SUM(CASE WHEN CompletionStatus = 'Delayed' THEN 1 ELSE 0 END) AS
      CompletedDelayed,
    COUNT(*) AS TotalWorkOrders
FROM
    WorkOrderCompletion
GROUP BY
    OperationSequence
ORDER BY
    OperationSequence;
```

--Back to Business scenario Q218

```
--Q219. Sample Solution

WITH WorkOrderLocation AS (
    SELECT
        wr.WorkOrderID,
        wr.OperationSequence,
        wr.LocationID,
        loc.Name AS LocationName

    FROM
        Production.WorkOrderRouting AS wr
    INNER JOIN
        Production.Location AS loc
    ON
        wr.LocationID = loc.LocationID
)
SELECT
    wo.WorkOrderID,
    wo.DueDate,
    wol.OperationSequence,
    wol.LocationName
FROM
    Production.WorkOrder AS wo
INNER JOIN
    WorkOrderLocation AS wol
ON
    wo.WorkOrderID = wol.WorkOrderID
ORDER BY
    wo.WorkOrderID, wol.OperationSequence;
```

--Back to Business scenario Q219

```
--Q220. Sample Solution

SELECT
    sp.BusinessEntityID AS SalesPersonID,
    sqh.QuotaDate,
    sqh.SalesQuota,
    SUM(soh.TotalDue) AS ActualSales
FROM
    Sales.SalesPerson AS sp
INNER JOIN
```

```
    Sales.SalesPersonQuotaHistory AS sqh
ON
    sp.BusinessEntityID = sqh.BusinessEntityID
LEFT JOIN
    Sales.SalesOrderHeader AS soh
ON
    sp.BusinessEntityID = soh.SalesPersonID
    AND sqh.QuotaDate = soh.OrderDate
GROUP BY
    sp.BusinessEntityID,
    sqh.QuotaDate,
    sqh.SalesQuota
ORDER BY
    sp.BusinessEntityID, sqh.QuotaDate;

--Back to Business scenario Q220


--Q221. Sample Solution

SELECT
    sp.BusinessEntityID AS SalesPersonID,
    sqh.QuotaDate,
    sqh.SalesQuota,
    SUM(CASE WHEN soh.TotalDue >= sqh.SalesQuota THEN 1 ELSE 0 END) AS
      QuotaMetCount,
    SUM(CASE WHEN soh.TotalDue < sqh.SalesQuota THEN 1 ELSE 0 END) AS
      QuotaNotMetCount
FROM
    Sales.SalesPerson AS sp
INNER JOIN
    Sales.SalesPersonQuotaHistory AS sqh
ON
    sp.BusinessEntityID = sqh.BusinessEntityID
LEFT JOIN
    Sales.SalesOrderHeader AS soh
ON
    sp.BusinessEntityID = soh.SalesPersonID
    AND sqh.QuotaDate = soh.OrderDate
GROUP BY
    sp.BusinessEntityID,
    sqh.QuotaDate,
    sqh.SalesQuota
 ORDER BY
    sp.BusinessEntityID, sqh.QuotaDate;

--Back to Business scenario Q221


--Q222. Sample Solution

WITH ProductSales AS (
    SELECT
        p.ProductID,
        p.Name AS ProductName,
        SUM(sci.Quantity) AS TotalSales
     FROM
```

```
        Production.Product AS p
    LEFT JOIN
        Sales.ShoppingCartItem AS sci
    ON
        p.ProductID = sci.ProductID
    GROUP BY
        p.ProductID, p.Name
),
CurrentInventory AS (
    SELECT
        ProductID,
        SUM(Quantity) AS CurrentStock
    FROM
        Production.ProductInventory
    GROUP BY
        ProductID
)
SELECT
    ps.ProductID,
    ps.ProductName,
    ps.TotalSales,
    COALESCE(ci.CurrentStock, 0) AS CurrentStock,
    CASE
        WHEN ci.CurrentStock IS NULL THEN ps.TotalSales
        ELSE ps.TotalSales - ci.CurrentStock
    END AS ReplenishQuantity
FROM
    ProductSales AS ps
LEFT JOIN
    CurrentInventory AS ci
ON
    ps.ProductID = ci.ProductID
ORDER BY
    ReplenishQuantity DESC;

--Back to Business scenario Q222


--Q223. Sample Solution

WITH ProductSales AS (
    SELECT
        p.ProductID,
        p.Name AS ProductName,
        pc.Name AS CategoryName,
        SUM(sci.Quantity) AS TotalSales
    FROM
        Production.Product AS p
    INNER JOIN
        Production.ProductSubcategory AS psc
    ON
        p.ProductSubcategoryID = psc.ProductSubcategoryID

    JOIN Production.ProductCategory pc ON psc.ProductCategoryID =
      pc.ProductCategoryID

    LEFT JOIN
        Sales.ShoppingCartItem AS sci
    ON
        p.ProductID = sci.ProductID
    GROUP BY
        p.ProductID, p.Name, pc.Name
)
SELECT
    ProductID,
    ProductName,
    CategoryName,
    TotalSales
FROM
    ProductSales
ORDER BY
    TotalSales DESC;

--Back to Business scenario Q223


--Q224. Sample Solution

WITH SupplierDelivery AS (
    SELECT
        S.SupplierID,
        S.SupplierName,
        COUNT(PO.PurchaseOrderID) AS TotalPurchaseOrders,
        SUM(CASE WHEN PO.ExpectedDeliveryDate >= ST.TransactionDate THEN 1 ELSE 0 
          END) AS OnTimeDeliveries
    FROM
        Purchasing.Suppliers AS S
    LEFT JOIN
        Purchasing.PurchaseOrders AS PO ON S.DeliveryMethodID = PO.DeliveryMethodID

    JOIN Purchasing.SupplierTransactions ST ON ST.SupplierID = PO.SupplierID

    GROUP BY
        S.SupplierID, S.SupplierName
)
SELECT
    SD.SupplierName,
    SD.TotalPurchaseOrders,
    SD.OnTimeDeliveries,
    CAST((SD.OnTimeDeliveries * 100.0) / SD.TotalPurchaseOrders AS DECIMAL(10, 2)) 
      AS OnTimeDeliveryPercentage
FROM
    SupplierDelivery AS SD
ORDER BY
    OnTimeDeliveryPercentage DESC;

--Back to Business scenario Q224


--Q225. Sample Solution

WITH RankedStockItems AS (
    SELECT
        C.ColorName,
        SI.StockItemName,
        SI.UnitPrice,
        ROW_NUMBER() OVER (PARTITION BY C.ColorName ORDER BY SI.UnitPrice DESC) AS 
          Rank
    FROM
        Warehouse.Colors AS C
    LEFT JOIN
        Warehouse.StockItems AS SI
    ON
        C.ColorID = SI.ColorID
)
SELECT
    ColorName,
    StockItemName,
    UnitPrice
FROM
    RankedStockItems
WHERE
    Rank = 1
ORDER BY
    ColorName;

--Back to Business scenario Q225


--Q226. Sample Solution

WITH OrderLineDetails AS (
    SELECT
        PT.PackageTypeName,
        OL.UnitPrice * OL.Quantity AS LineAmount
    FROM
        Warehouse.PackageTypes AS PT
    JOIN
        Warehouse.StockItems AS SI
    ON
        PT.PackageTypeID = SI.UnitPackageID
    LEFT JOIN
        Sales.OrderLines AS OL
    ON
        SI.StockItemID = OL.StockItemID
)
SELECT
    PackageTypeName,
    AVG(LineAmount) AS AverageOrderLineAmount
FROM
    OrderLineDetails
GROUP BY
    PackageTypeName
ORDER BY
    AverageOrderLineAmount DESC;

--Back to Business scenario Q226


--Q227. Sample Solution

WITH RankedStockItems AS (
    SELECT
        SISG.StockGroupID,
        SI.StockItemID,
        SI.StockItemName,
        SUM(OL.Quantity) AS TotalQuantitySold,
        ROW_NUMBER() OVER (PARTITION BY SISG.StockGroupID ORDER BY SUM(OL.Quantity)
           DESC) AS ItemRank
    FROM
        Warehouse.StockItemStockGroups AS SISG
    INNER JOIN
        Warehouse.StockItems AS SI
    ON
        SISG.StockItemID = SI.StockItemID
    LEFT JOIN
        Sales.OrderLines AS OL
    ON
        SI.StockItemID = OL.StockItemID
    WHERE
        OL.OrderID IS NOT NULL
    GROUP BY
        SISG.StockGroupID,
        SI.StockItemID,
        SI.StockItemName
)
SELECT
    RSI.StockGroupID,
    SG.StockGroupName,
    RSI.StockItemID,
    RSI.StockItemName,
    RSI.TotalQuantitySold,
    AVG(RSI.TotalQuantitySold) OVER (PARTITION BY RSI.StockGroupID) AS
      AvgQuantitySold,
    SUM(RSI.TotalQuantitySold) OVER (PARTITION BY RSI.StockGroupID) AS
      TotalQuantitySoldInGroup
FROM
    RankedStockItems AS RSI
INNER JOIN
    Warehouse.StockGroups AS SG
ON
    RSI.StockGroupID = SG.StockGroupID
WHERE
    RSI.ItemRank = 1;

--Back to Business scenario Q227


--Q228. Sample Solution

WITH RankedStockGroups AS (
    SELECT
        SG.StockGroupID,
        SG.StockGroupName,
        SUM(OL.Quantity) AS TotalQuantitySold,
        RANK() OVER (ORDER BY SUM(OL.Quantity) DESC) AS RankByQuantitySold,
        LAG(SUM(OL.Quantity)) OVER (ORDER BY SG.StockGroupID) AS
          PreviousQuantitySold,
        SUM(SUM(OL.Quantity)) OVER (ORDER BY SG.StockGroupID) AS
          CumulativeQuantitySold,
        LEAD(SUM(OL.Quantity)) OVER (ORDER BY SG.StockGroupID) AS NextQuantitySold
    FROM
```

```
        Warehouse.StockGroups AS SG
    LEFT JOIN
        Warehouse.StockItemStockGroups AS SISG
    ON
        SG.StockGroupID = SISG.StockGroupID
    LEFT JOIN
        Warehouse.StockItems AS SI
    ON
        SISG.StockItemID = SI.StockItemID
    LEFT JOIN
        Sales.OrderLines AS OL
    ON
        SI.StockItemID = OL.StockItemID
    GROUP BY
        SG.StockGroupID,
        SG.StockGroupName
)
SELECT
    RSG.StockGroupID,
    RSG.StockGroupName,
    RSG.TotalQuantitySold,
    RSG.RankByQuantitySold,
    CASE
        WHEN RSG.PreviousQuantitySold IS NULL THEN NULL
        ELSE (RSG.TotalQuantitySold - RSG.PreviousQuantitySold) * 100.0 /
          RSG.PreviousQuantitySold
    END AS SalesGrowthRate,
    RSG.CumulativeQuantitySold,
    RSG.NextQuantitySold
FROM
    RankedStockGroups AS RSG;

--Back to Business scenario Q228


--Q229. Sample Solution

WITH LoanApplications AS (
    SELECT
        c.CustomerID,
        c.FirstName,
        c.LastName,
        a.AccountID,
        a.AccountType,
        a.Balance,
        CASE
            WHEN a.Balance >= 5000 THEN 'Eligible'
            ELSE 'Not Eligible'
        END AS LoanEligibility
    FROM
        Finance.UniqueCustomers c
    INNER JOIN
        Finance.UniqueAccounts a ON c.CustomerID = a.CustomerID
    WHERE
        a.AccountType = 'Savings'
)
SELECT
    CustomerID,
    FirstName,
    LastName,
    AccountID,
    AccountType,
    Balance,
    LoanEligibility,
    CASE
        WHEN LoanEligibility = 'Eligible' THEN 'Approved'
        ELSE 'Rejected'
    END AS LoanStatus
FROM
    LoanApplications;

--Back to Business scenario Q229


--Q230. Sample Solution

SELECT
    CASE
        WHEN A.AccountType = 'savings' THEN 'Savings'
        WHEN A.AccountType = 'checking' THEN 'Checking'
    END AS AccountType,
    COUNT(DISTINCT C.CustomerID) AS NumberOfCustomers,
    COUNT(DISTINCT C.CustomerID) * 100.0 / (SELECT COUNT(DISTINCT CustomerID) FROM ↩
      Finance.UniqueCustomers) AS Percentage
FROM
    Finance.UniqueAccounts A
INNER JOIN
    Finance.UniqueCustomers C ON A.CustomerID = C.CustomerID
GROUP BY
    A.AccountType;

--Back to Business scenario Q230


--Q231. Sample Solution

DECLARE @TargetYear INT = 2023;

-- Generate the yearly transaction summary
SELECT
    C.CustomerID,
    C.FirstName,
    C.LastName,
    C.Email,
    C.PhoneNumber,
    SUM(CASE WHEN YEAR(T.TransactionDate) = @TargetYear THEN 1 ELSE 0 END) AS      ↩
      TransactionCount,
    SUM(CASE WHEN YEAR(T.TransactionDate) = @TargetYear THEN T.Amount ELSE 0 END)  ↩
      AS TotalAmount
FROM
    Finance.UniqueCustomers C
LEFT JOIN
    Finance.UniqueAccounts A ON C.CustomerID = A.CustomerID
LEFT JOIN
    Finance.UniqueTransactions T ON A.AccountID = T.AccountID
WHERE
```

```
    YEAR(T.TransactionDate) = @TargetYear
GROUP BY
    C.CustomerID, C.FirstName, C.LastName, C.Email, C.PhoneNumber
ORDER BY
    C.CustomerID;

--Back to Business scenario Q231


--Q232. Sample Solution

SELECT
    A.AccountID,
    C.CustomerID,
    C.FirstName,
    C.LastName,
    C.Email,
    C.PhoneNumber,
    A.AccountType,
    A.Balance,
    MAX(T.TransactionDate) AS LastTransactionDate,
    CASE
        WHEN MAX(T.TransactionDate) IS NULL THEN 'No Transactions'
        WHEN MAX(T.TransactionDate) IS NOT NULL AND A.Balance = 0 THEN 'Zero
          Balance'
        ELSE 'Other Reasons'
    END AS ClosureReason
FROM
    Finance.UniqueCustomers C
INNER JOIN
    Finance.UniqueAccounts A ON C.CustomerID = A.CustomerID
LEFT JOIN
    Finance.UniqueTransactions T ON A.AccountID = T.AccountID
GROUP BY
    A.AccountID, C.CustomerID, C.FirstName, C.LastName, C.Email, C.PhoneNumber,
      A.AccountType, A.Balance
HAVING
    MAX(T.TransactionDate) IS NULL OR A.Balance = 0
ORDER BY
    A.AccountID;

--Back to Business scenario Q232


--Q233. Sample Solution

SELECT
    YEAR(T.TransactionDate) AS TransactionYear,
    CASE
        WHEN A.AccountType = 'savings' THEN 'Savings'
        WHEN A.AccountType = 'checking' THEN 'Checking'
    END AS AccountType,
    COUNT(T.TransactionID) AS TransactionCount
FROM
    Finance.UniqueTransactions T
INNER JOIN
    Finance.UniqueAccounts A ON T.AccountID = A.AccountID
GROUP BY
```

```
    YEAR(T.TransactionDate), A.AccountType
ORDER BY
    YEAR(T.TransactionDate), A.AccountType;

--Back to Business scenario Q233


--Q234. Sample Solution

SELECT
    C.CustomerID,
    C.FirstName,
    C.LastName,
    C.Email,
    C.PhoneNumber,
    A.AccountType,
    A.Balance,
    COUNT(CASE WHEN T.TransactionType = 'Deposit' THEN 1 ELSE NULL END) AS
      DepositCount,
    COUNT(CASE WHEN T.TransactionType = 'Withdrawal' THEN 1 ELSE NULL END) AS
      WithdrawalCount
FROM
    Finance.UniqueCustomers C
INNER JOIN
    Finance.UniqueAccounts A ON C.CustomerID = A.CustomerID
LEFT JOIN
    Finance.UniqueTransactions T ON A.AccountID = T.AccountID
GROUP BY
    C.CustomerID, C.FirstName, C.LastName, C.Email, C.PhoneNumber, A.AccountType,
      A.Balance
ORDER BY
    C.CustomerID;

--Back to Business scenario Q234


--Q235. Sample Solution

SELECT
    C.CustomerID,
    C.FirstName,
    C.LastName,
    C.Email,
    C.PhoneNumber,
    CASE
        WHEN TransactionCount >= 10 AND AverageBalance >= 5000 THEN 'Active'
        WHEN TransactionCount >= 5 AND AverageBalance >= 1000 THEN 'Occasional'
        ELSE 'Inactive'
    END AS CustomerSegment
FROM
    Finance.UniqueCustomers C
LEFT JOIN (
    SELECT
        A.CustomerID,
        COUNT(T.TransactionID) AS TransactionCount,
        AVG(A.Balance) AS AverageBalance
    FROM
        Finance.UniqueAccounts A
    LEFT JOIN
        Finance.UniqueTransactions T ON A.AccountID = T.AccountID
    GROUP BY
        A.CustomerID
) AS CustomerSummary ON C.CustomerID = CustomerSummary.CustomerID
ORDER BY
    C.CustomerID;

--Back to Business scenario Q235


--Q236. Sample Solution

    SELECT
    T.TransactionID,
    C.CustomerID,
    C.FirstName,
    C.LastName,
    T.TransactionDate,
    T.Amount,
    T.TransactionType,
    A.AccountType,
    A.Balance,
    CASE
        WHEN ABS(T.Amount - AVG(T2.Amount)) > 2 * STDEV(T2.Amount) THEN 'Anomaly
          Detected'
        ELSE 'Normal'
    END AS TransactionStatus
FROM
    Finance.UniqueTransactions T
INNER JOIN
    Finance.UniqueAccounts A ON T.AccountID = A.AccountID
INNER JOIN
    Finance.UniqueCustomers C ON A.CustomerID = C.CustomerID
LEFT JOIN
    Finance.UniqueTransactions T2 ON A.AccountID = T2.AccountID AND T.TransactionID
      <> T2.TransactionID
GROUP BY
    T.TransactionID, C.CustomerID, C.FirstName, C.LastName, T.TransactionDate,
      T.Amount, T.TransactionType, A.AccountType, A.Balance
ORDER BY
    T.TransactionID;

--Back to Business scenario Q236


--Q237. Sample Solution

DECLARE @StartDate DATE = '2022-01-01';
DECLARE @EndDate DATE = '2023-12-31';

SELECT
    B.BranchID,
    B.BranchName,
    B.Location,
    COUNT(T.TransactionID) AS TransactionCount
FROM
    Finance.UniqueBranches B
```

```
LEFT JOIN
    Finance.UniqueAccountBranches AB ON B.BranchID = AB.BranchID
    JOIN Finance.UniqueTransactions T ON AB.AccountID = T.AccountID
WHERE
    T.TransactionDate BETWEEN @StartDate AND @EndDate
GROUP BY
    B.BranchID, B.BranchName, B.Location
ORDER BY
    TransactionCount DESC
SELECT
    B.BranchID,
    B.BranchName,
    B.Location,
    COUNT(AB.AccountID) AS AccountCount
FROM
    Finance.UniqueBranches B
LEFT JOIN
    Finance.UniqueAccountBranches AB ON B.BranchID = AB.BranchID
GROUP BY
    B.BranchID, B.BranchName, B.Location
ORDER BY
    AccountCount DESC

--Back to Business scenario Q237


--Q238. Sample Solution

SELECT
    C.CustomerID,
    C.FirstName,
    C.LastName,
    B.BranchName AS PreferredBranch,
    MAX(T.TransactionDate) AS LastTransactionDate
FROM
    Finance.UniqueCustomers C
LEFT JOIN
    Finance.UniqueAccounts A ON C.CustomerID = A.CustomerID
LEFT JOIN
    Finance.UniqueAccountBranches AB ON A.AccountID = AB.AccountID
LEFT JOIN
    Finance.UniqueBranches B ON AB.BranchID = B.BranchID
LEFT JOIN
    Finance.UniqueTransactions T ON A.AccountID = T.AccountID
GROUP BY
    C.CustomerID, C.FirstName, C.LastName, B.BranchName
HAVING
    MAX(T.TransactionDate) IS NOT NULL
ORDER BY
    C.CustomerID;

--Back to Business scenario Q238


--Q239. Sample Solution

SELECT
    B.BranchID,
    B.BranchName,
    B.Location,
    B.City,
    SUM(T.Amount) AS TransactionRevenue,
    SUM(CASE WHEN A.AccountType = 'savings' THEN 0.02 * A.Balance ELSE 0.01 * 
      A.Balance END) AS AccountBalanceFees,
    SUM(T.Amount) + SUM(CASE WHEN A.AccountType = 'savings' THEN 0.02 * A.Balance 
      ELSE 0.01 * A.Balance END) AS TotalProfit
FROM
    Finance.UniqueBranches B
LEFT JOIN
    Finance.UniqueAccountBranches AB ON B.BranchID = AB.BranchID
LEFT JOIN
    Finance.UniqueAccounts A ON AB.AccountID = A.AccountID
LEFT JOIN
    Finance.UniqueTransactions T ON A.AccountID = T.AccountID
GROUP BY
    B.BranchID, B.BranchName, B.Location, B.City
ORDER BY
    TotalProfit DESC;

--Back to Business scenario Q239


--Q240. Sample Solution

SELECT PatientID, FirstName, LastName, DateOfBirth, Gender, PhoneNumber, Email
FROM Medical.Patients
WHERE (
    EXISTS (
        SELECT *
        FROM Medical.Medications
        WHERE PatientID = Patients.PatientID
        AND Dosage = '10 mg'
    )
    OR
    EXISTS (
        SELECT *
        FROM Medical.Appointments
        WHERE Patients.PatientID = Appointments.PatientID
        AND Purpose = 'Critical care'
    )
);

--Back to Business scenario Q240


--Q241. Sample Solution

-- Identify prescriptions nearing or exceeding expiry dates
SELECT
    M.MedicationID,
    M.MedicationName,
    M.Dosage,
    M.PatientID,
    M.PrescribingDoctorID,
    M.PrescriptionDate,
    M.ExpiryDate,
    P.FirstName AS PatientFirstName,
    P.LastName AS PatientLastName,
    D.FirstName AS DoctorFirstName,
    D.LastName AS DoctorLastName
FROM Medical.Medications M
INNER JOIN Medical.Patients P ON M.PatientID = P.PatientID
INNER JOIN Medical.Doctors D ON M.PrescribingDoctorID = D.DoctorID
WHERE M.ExpiryDate <= DATEADD(MONTH, 6, GETDATE()) -- Medications expiring within
  the next 6 months
   OR M.ExpiryDate <= GETDATE(); -- Expired medications

-- Identify doctors prescribing medications with potential contraindications
SELECT
    D.DoctorID,
    D.FirstName,
    D.LastName,
    M.MedicationName,
    P.FirstName AS PatientFirstName,
    P.LastName AS PatientLastName
FROM Medical.Medications M
INNER JOIN Medical.Patients P ON M.PatientID = P.PatientID
INNER JOIN Medical.Doctors D ON M.PrescribingDoctorID = D.DoctorID
WHERE EXISTS (
    SELECT 1
    FROM Medical.Medications M2
    WHERE M2.PatientID = M.PatientID
      AND M2.MedicationID <> M.MedicationID
      AND M2.ExpiryDate >= GETDATE()
      AND M2.MedicationName = M.MedicationName
);

--Back to Business scenario Q241


--Q242. Sample Solution

WITH DoctorAvailability AS (
    SELECT
        D.DoctorID,
        D.FirstName AS DoctorFirstName,
        D.LastName AS DoctorLastName,
        COUNT(A.AppointmentID) AS NumAppointments
    FROM Medical.Doctors D
    LEFT JOIN Medical.Appointments A ON D.DoctorID = A.DoctorID
    GROUP BY D.DoctorID, D.FirstName, D.LastName
)
SELECT
    DA.DoctorFirstName,
    DA.DoctorLastName,
    DA.NumAppointments AS TotalAppointments,
    MAX(A.AppointmentDateTime) AS LastAppointmentDate,
    DATEDIFF(DAY, MAX(A.AppointmentDateTime), GETDATE()) AS
      DaysSinceLastAppointment
FROM DoctorAvailability DA
LEFT JOIN Appointments A ON DA.DoctorID = A.DoctorID
GROUP BY DA.DoctorID, DA.DoctorFirstName, DA.DoctorLastName, DA.NumAppointments
ORDER BY DaysSinceLastAppointment DESC;
```

--Back to Business scenario Q242

```
--Q243. Sample Solution

SELECT
    C.CustomerKey,
    C.FirstName,
    C.LastName,
    C.EmailAddress,
    CASE
        WHEN DATEDIFF(YEAR, C.BirthDate, GETDATE()) >= 18 THEN 'Adult'
        ELSE 'Minor'
    END AS CustomerAgeGroup,
    CASE
        WHEN PC.TotalPurchases >= 10 THEN 'Loyal'
        ELSE 'Regular'
    END AS CustomerLoyaltyStatus
FROM DimCustomer C
LEFT JOIN (
    SELECT CustomerKey, COUNT(DISTINCT SalesOrderNumber) AS TotalPurchases
    FROM FactInternetSales
    GROUP BY CustomerKey
) PC ON C.CustomerKey = PC.CustomerKey
ORDER BY C.CustomerKey;
```

--Back to Business scenario Q243

```
--Q244. Sample Solution

-- Customer Sales Analysis
WITH CustomerSales AS (
    SELECT
        C.FirstName,
        C.LastName,
        SUM(FS.SalesAmount) AS TotalSalesAmount
    FROM
        dbo.FactInternetSales AS FS
    INNER JOIN dbo.DimCustomer AS C ON FS.CustomerKey = C.CustomerKey
    GROUP BY
        C.FirstName,
        C.LastName
)
SELECT
    FirstName,
    LastName,
    TotalSalesAmount,
    CASE
        WHEN TotalSalesAmount >= 5000 THEN 'High-Spending'
        WHEN TotalSalesAmount >= 2000 THEN 'Medium-Spending'
        ELSE 'Low-Spending'
    END AS CustomerCategory
FROM
    CustomerSales
ORDER BY
    TotalSalesAmount DESC;
```

--Back to Business scenario Q244

```sql
--Q245. Sample Solution

-- Product Inventory Analysis
WITH ProductInventory AS (
    SELECT
        P.EnglishProductName AS ProductName,
        SUM(I.UnitsBalance) AS TotalQuantity
    FROM
        dbo.FactProductInventory AS I
    INNER JOIN dbo.DimProduct AS P ON I.ProductKey = P.ProductKey
    GROUP BY
        P.EnglishProductName
)
SELECT
    ProductName,
    TotalQuantity,
    CASE
        WHEN TotalQuantity < 50 THEN 'Low Inventory'
        WHEN TotalQuantity < 100 THEN 'Moderate Inventory'
        ELSE 'Sufficient Inventory'
    END AS InventoryStatus
FROM
    ProductInventory
ORDER BY
    TotalQuantity ASC;
```

--Back to Business scenario Q245

```sql
--Q246. Sample Solution

-- Employee Sales Performance Evaluation
SELECT
    E.EmployeeKey,
    CONCAT(E.FirstName, ' ', E.LastName) AS EmployeeName,
    YEAR(IH.OrderDate) AS EvaluationYear,
    MONTH(IH.OrderDate) AS EvaluationMonth,
    SUM(IH.SalesAmount) AS TotalSalesAmount
FROM
    DimEmployee AS E
INNER JOIN FactInternetSales AS IH ON E.SalesTerritoryKey = IH.SalesTerritoryKey
WHERE
    YEAR(IH.OrderDate) = 2012
GROUP BY
    E.EmployeeKey, E.FirstName, E.LastName, YEAR(IH.OrderDate), MONTH(IH.OrderDate)
ORDER BY
    TotalSalesAmount DESC;
```

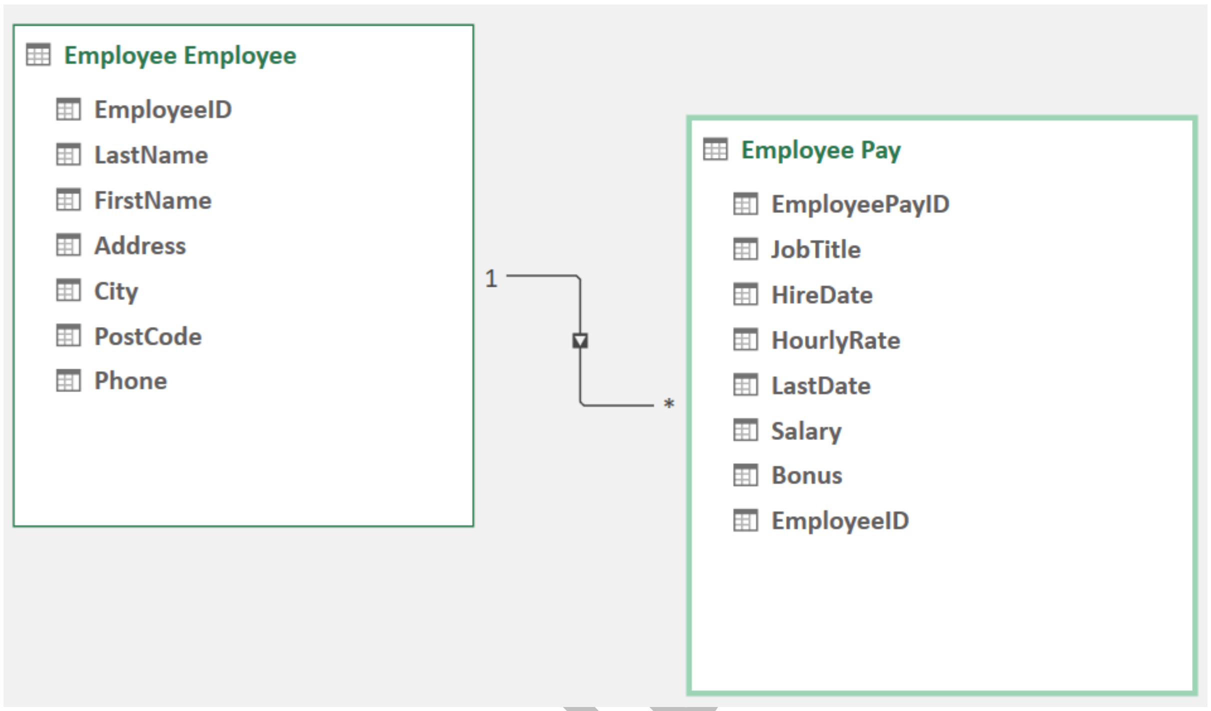
--Back to Business scenario Q246

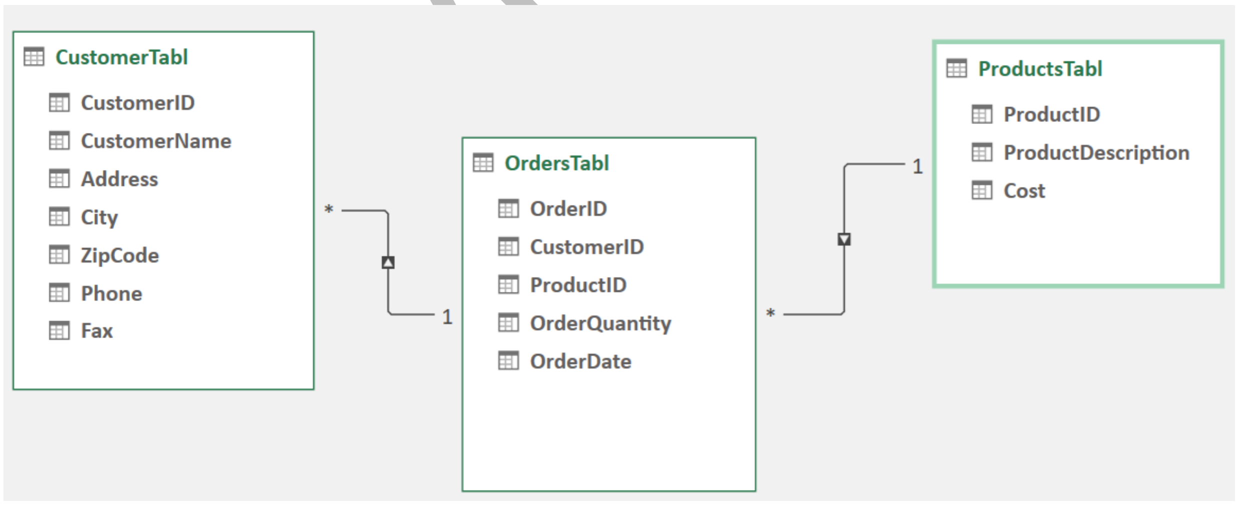
```sql
--Q247. Sample Solution

SELECT
    PC.ProductCategoryKey,
    PS.ProductSubcategoryKey,
    YEAR(IH.OrderDate) AS SalesYear,
    SUM(IH.SalesAmount) AS TotalSalesAmount
FROM
    dbo.FactInternetSales AS IH
INNER JOIN dbo.DimProduct AS DP ON IH.ProductKey = DP.ProductKey
INNER JOIN dbo.DimProductSubcategory AS PS ON DP.ProductSubcategoryKey = 
  PS.ProductSubcategoryKey
INNER JOIN dbo.DimProductCategory AS PC ON PS.ProductCategoryKey = 
  PC.ProductCategoryKey
GROUP BY
    PC.ProductCategoryKey, PS.ProductSubcategoryKey, YEAR(IH.OrderDate)
ORDER BY
    SalesYear, TotalSalesAmount DESC;

--Back to Business scenario Q247


--Q248. Sample Solution

DECLARE @AnalysisYear INT = 2012; -- Specify the year for analysis

SELECT
    DG.EnglishCountryRegionName AS Region,
    SUM(FIS.SalesAmount) AS TotalSalesAmount
FROM
    dbo.FactInternetSales AS FIS
INNER JOIN
    dbo.DimDate AS DD
    ON FIS.OrderDateKey = DD.DateKey
INNER JOIN
    dbo.DimGeography AS DG
    ON FIS.SalesTerritoryKey = DG.SalesTerritoryKey
WHERE
    DATEPART(YEAR, DD.FullDateAlternateKey) = @AnalysisYear
GROUP BY
    DG.EnglishCountryRegionName
ORDER BY
    TotalSalesAmount DESC;

--Back to Business scenario Q248



--Q249. Sample Solution

DECLARE @StartDate DATE = '2013-01-01'; -- Specify the start date of the analysis 
  period
DECLARE @EndDate DATE = '2013-12-31';   -- Specify the end date of the analysis 
  period

SELECT
    DPC.EnglishProductCategoryName AS ProductCategory,
    SUM(FIS.SalesAmount) AS TotalSalesAmount
FROM
    dbo.FactInternetSales AS FIS
INNER JOIN
    dbo.DimProduct AS DP
    ON FIS.ProductKey = DP.ProductKey
INNER JOIN
    dbo.DimProductSubcategory AS DPSC
    ON DP.ProductSubcategoryKey = DPSC.ProductSubcategoryKey
INNER JOIN
    dbo.DimProductCategory AS DPC
    ON DPSC.ProductCategoryKey = DPC.ProductCategoryKey
INNER JOIN
    dbo.DimDate AS DD
    ON FIS.OrderDateKey = DD.DateKey
WHERE
    DD.FullDateAlternateKey BETWEEN @StartDate AND @EndDate
GROUP BY
    DPC.EnglishProductCategoryName
ORDER BY
    TotalSalesAmount DESC;

--Back to Business scenario Q249


--Q250. Sample Solution

DECLARE @Year INT = 2012; -- Specify the year for analysis

WITH RankedProducts AS (
    SELECT
        DP.ProductKey,
        ST.SalesTerritoryRegion,
        DP.EnglishProductName,
        FS.SalesAmount,
        RANK() OVER(PARTITION BY ST.SalesTerritoryRegion ORDER BY FS.SalesAmount
          DESC) AS RankInTerritory
    FROM
        FactInternetSales AS FS
    INNER JOIN
        DimProduct AS DP ON FS.ProductKey = DP.ProductKey
    INNER JOIN
        DimSalesTerritory AS ST ON FS.SalesTerritoryKey = ST.SalesTerritoryKey
    INNER JOIN
        DimDate AS DD ON FS.OrderDateKey = DD.DateKey
    WHERE
        YEAR(DD.FullDateAlternateKey) = @Year
)
SELECT
    ProductKey,
    SalesTerritoryRegion,
    EnglishProductName,
    SalesAmount
FROM
    RankedProducts
WHERE
    RankInTerritory = 1;

--Back to Business scenario Q250
```

```
--Q251. Sample Solution

-- Promotion Analysis
SELECT
    DP.EnglishPromotionName,
    DP.StartDate,
    DP.EndDate,
    SUM(FS.SalesAmount) AS TotalSalesAmount
FROM
    DimPromotion AS DP
JOIN
    FactInternetSales AS FS ON DP.PromotionKey = FS.PromotionKey
WHERE
    DP.EnglishPromotionName = 'Touring-1000 Promotion' OR DP.EnglishPromotionName =
       'Touring-3000 Promotion'
GROUP BY
    DP.EnglishPromotionName,
    DP.StartDate,
    DP.EndDate;

--Back to Business scenario Q251
```

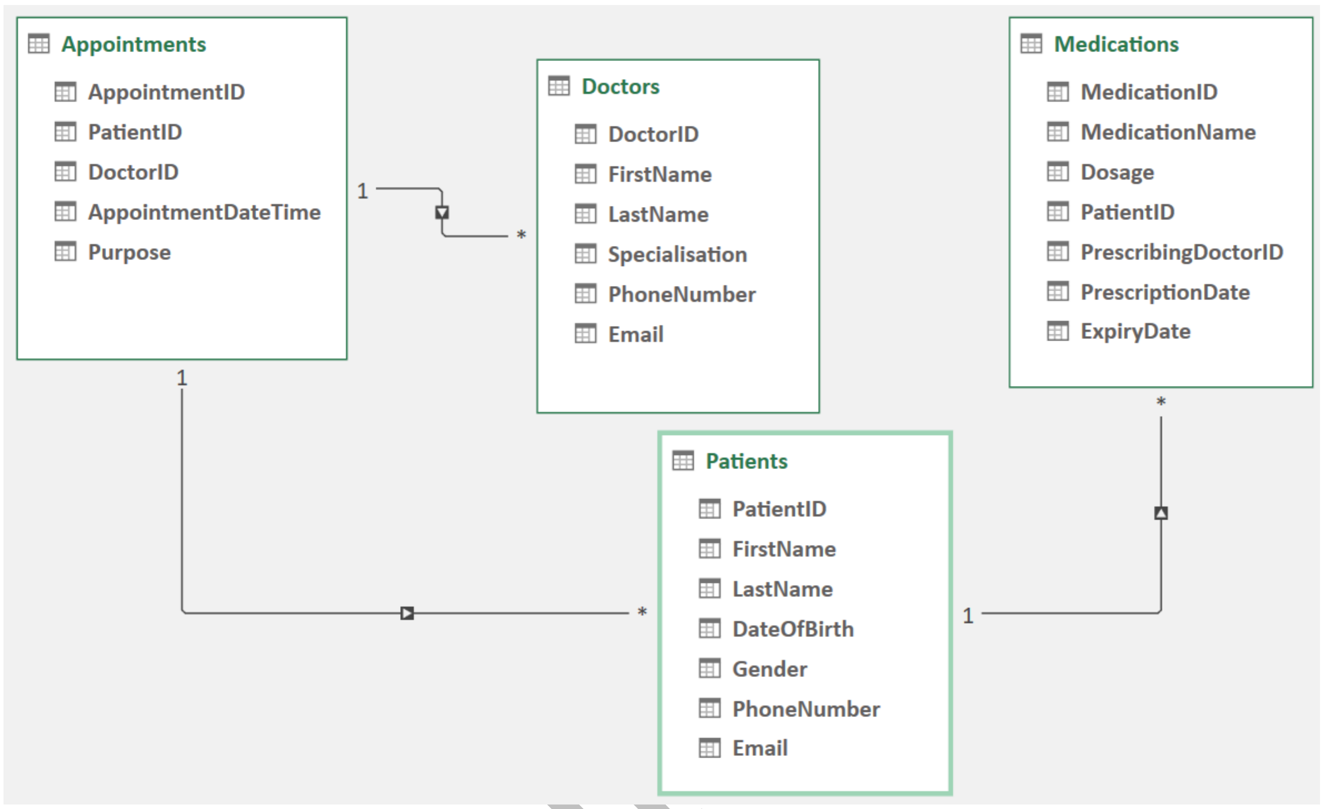
Appendix A

A1 – Eyowhite Database (HR Employee and ProductOrders)

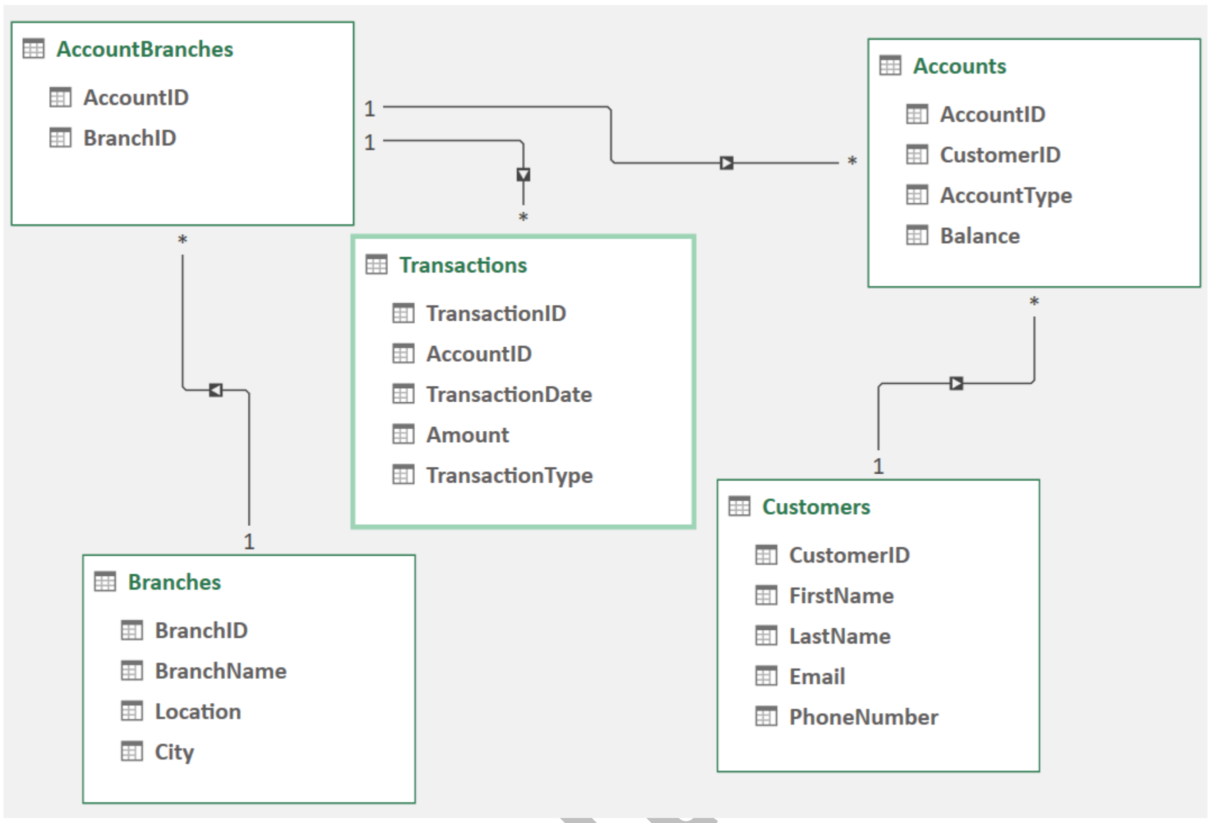




A2 – HealthCare Database



A3 – Finance and Banking Database

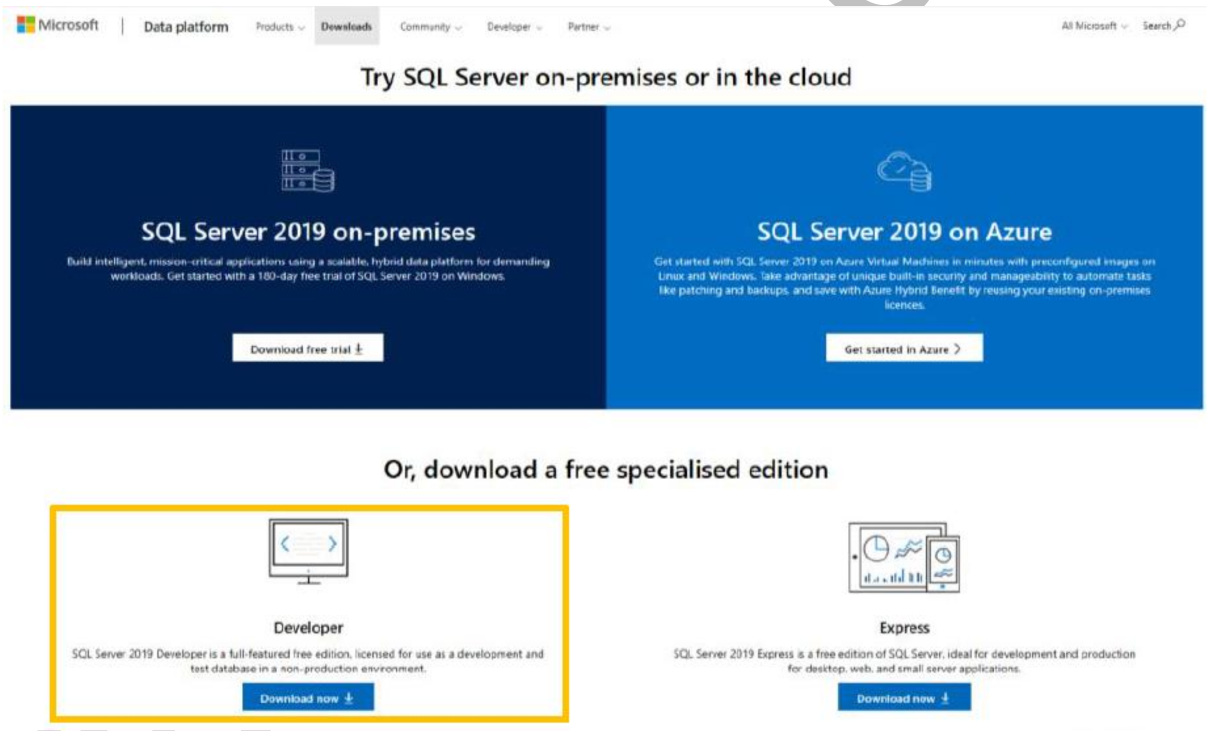


Appendix B

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.



Try SQL Server on-premises or in the cloud

SQL Server 2019 on-premises

Build intelligent, mission-critical applications using a scalable, hybrid data platform for demanding workloads. Get started with a 180-day free trial of SQL Server 2019 on Windows.

Download free trial

SQL Server 2019 on Azure

Get started with SQL Server 2019 on Azure Virtual Machines in minutes with preconfigured images on Linux and Windows. Take advantage of unique built-in security and manageability to automate tasks like patching and backups, and save with Azure Hybrid Benefit by reusing your existing on-premises licences.

Get started in Azure

Or, download a free specialised edition

Developer

SQL Server 2019 Developer is a full-featured free edition, licensed for use as a development and test database in a non-production environment.

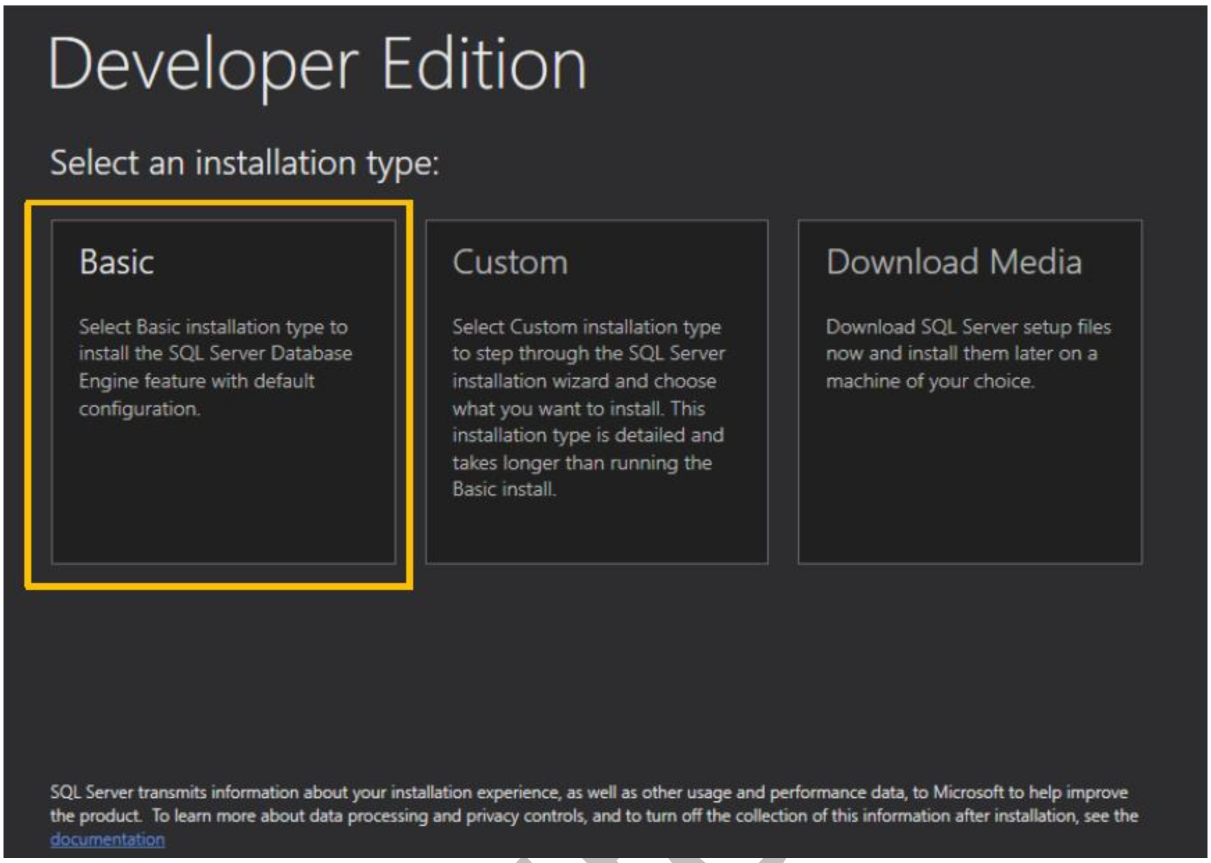
Download now

Express

SQL Server 2019 Express is a free edition of SQL Server, ideal for development and production for desktop, web, and small server applications.

Download now

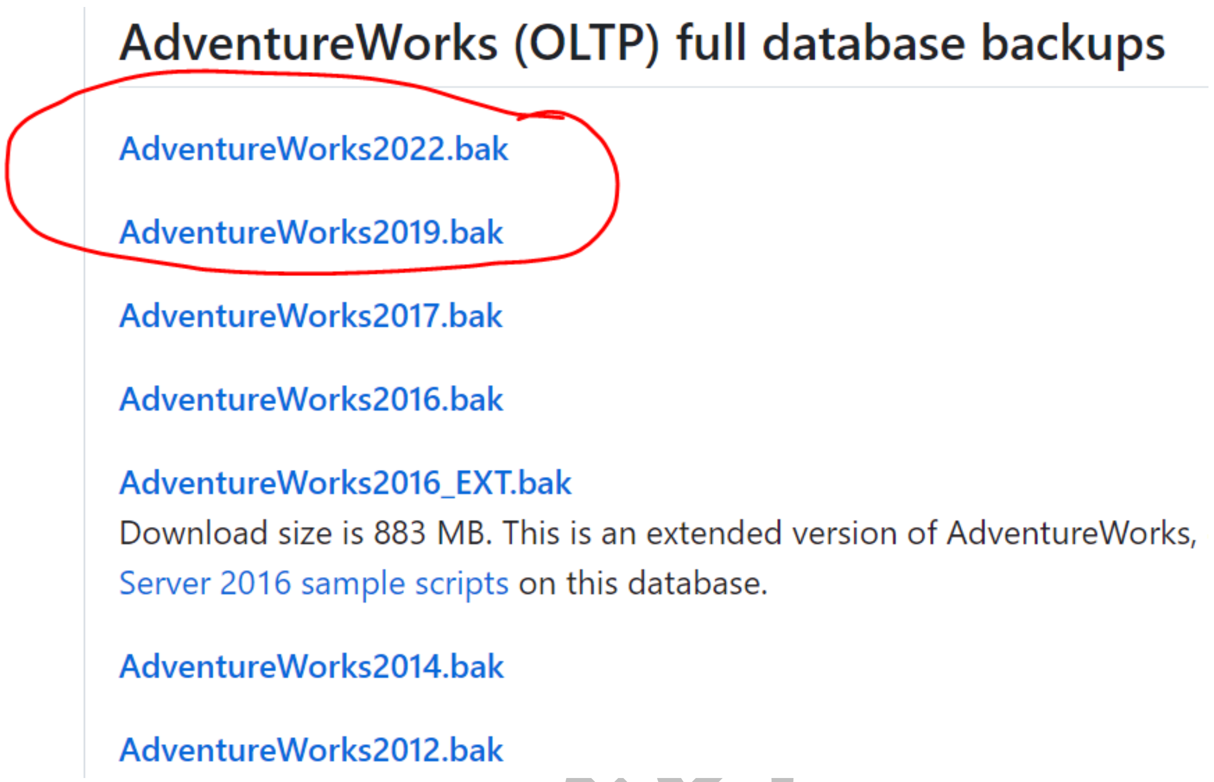
3. After the file has finished downloading, double-click on it to initiate the installation.
4. In the window that appears, choose the "Basic" installation type.


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

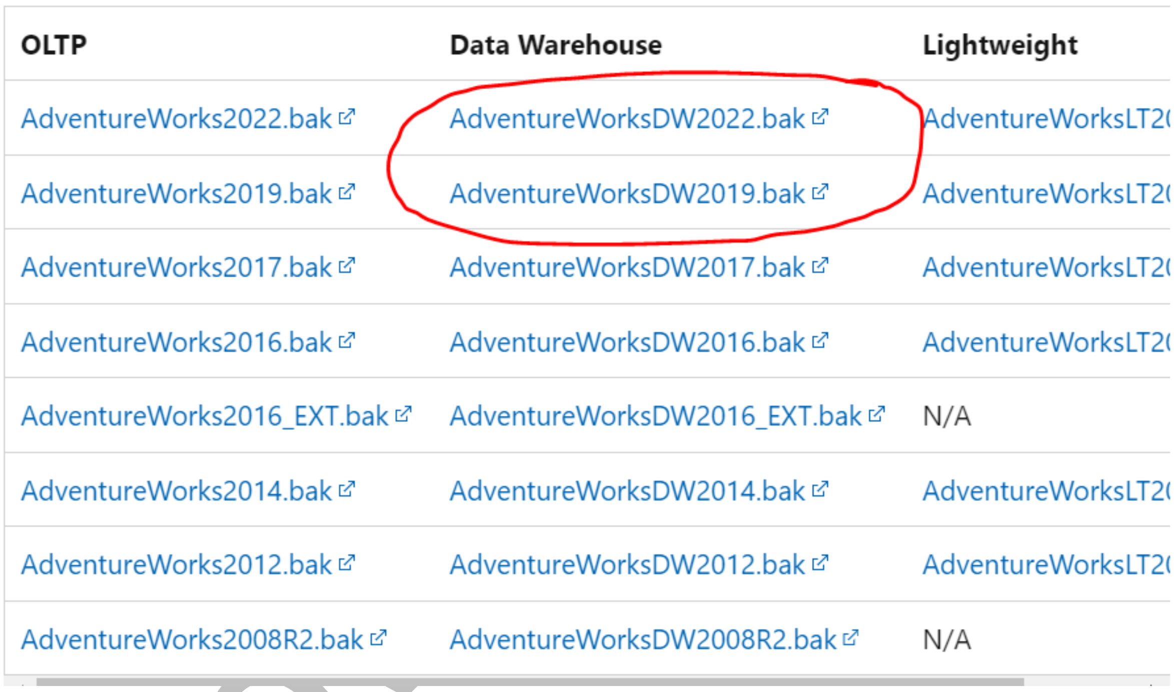
AdventureWorks2012.bak

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.SQLSERVERBI\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

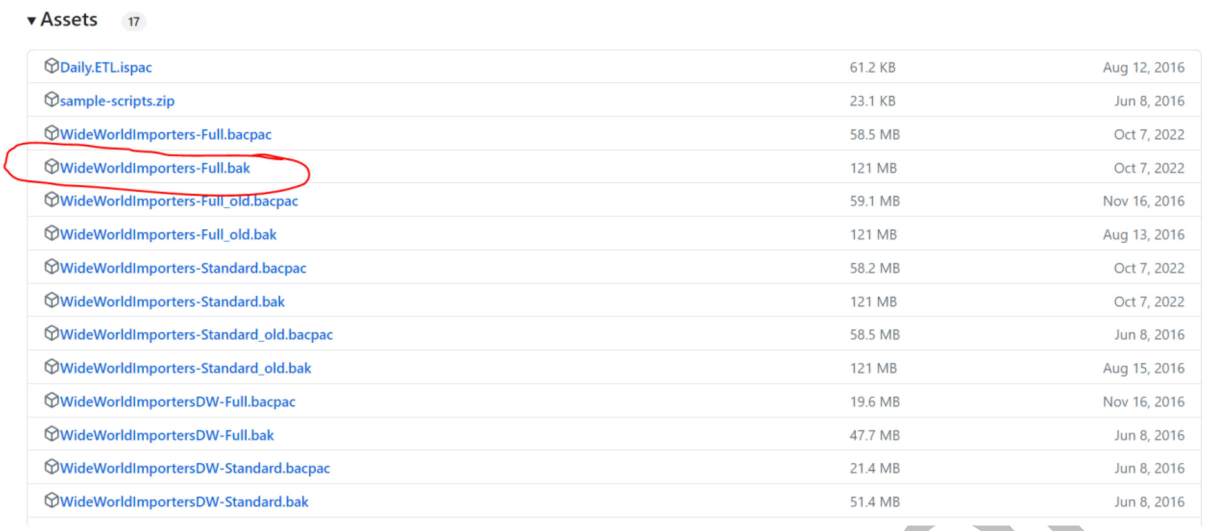
1. The following link allows you to download the AdventureWorksDW2019.bak or AdventureWorksDW2022.bak versions: https://learn.microsoft.com/en-us/sql/samples/adventureworks-install-configure?view=sql-server-ver16&tabs=ssms
2. After downloading the file, following similar steps as in **Phase 2** to restore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



OLTP	Data Warehouse	Lightweight
AdventureWorks2022.bak	AdventureWorksDW2022.bak	AdventureWorksLT20
AdventureWorks2019.bak	AdventureWorksDW2019.bak	AdventureWorksLT20
AdventureWorks2017.bak	AdventureWorksDW2017.bak	AdventureWorksLT20
AdventureWorks2016.bak	AdventureWorksDW2016.bak	AdventureWorksLT20
AdventureWorks2016_EXT.bak	AdventureWorksDW2016_EXT.bak	N/A
AdventureWorks2014.bak	AdventureWorksDW2014.bak	AdventureWorksLT20
AdventureWorks2012.bak	AdventureWorksDW2012.bak	AdventureWorksLT20
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 following 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