A man with glasses and a beard is looking directly at the camera. He is wearing a white t-shirt. The background is a blurred office setting with computer monitors. Overlaid on the background are various data visualizations, including bar charts, line graphs, and a city skyline at night. The text 'Data Analysis SQL Class Booklet' is centered over the image in a large, white, sans-serif font.

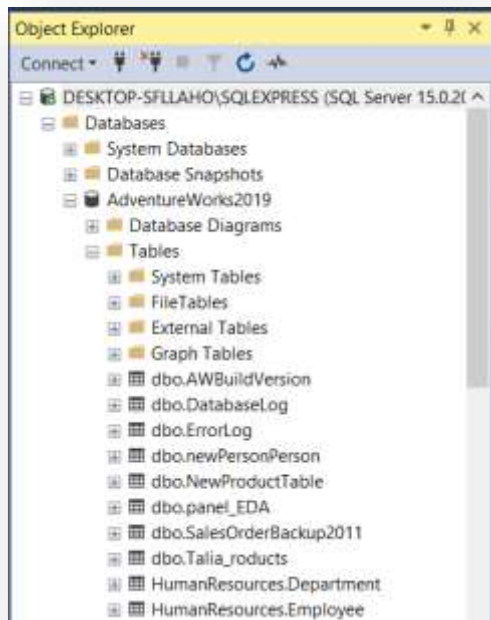
Data Analysis SQL Class Booklet

SQL Unit - Class practice booklet

Lesson 2 - Basic SQL query

Part 1 - Basic Data Retrieval

1. How many records (rows) are in the Production.ProductModel table?
2. Examine the data in the table. What is the name of the person in the 20th record of the Person.Person table?
3. Examine the table tree in Microsoft SQL Studio Management in your computer and count how many tables there are in HumanResources. (Count manually.)



Instruction: Look for the tables with the name HumanResources before the point.
E.g., HumanResources.Employee

4. On what date (OrderDate) was Order no. 43742 in the Sales.SalesOrderHeader table issued? (Scroll manually until you find the relevant record/row.)
5. Write a query that shows all the columns from the Purchasing.Vendor table.
6. Write a query that shows the following columns from the Person.Person table: BusinessEntityID, First name, Middle name and Last name.

SQL Unit - Class practice booklet

7. Write a query that shows the following columns from the Production.Product table: ProductID, [Name], ProductNumber, Color.
8. Write a query that shows all the columns from the Production.Product table.

Part 2 - Basic Filtering - WHERE clause

1. Write a query that returns the First name, Middle name and Last name of the people with the Middle name J in the Person.Person table.
2. How many orders were made by customer no. 15148 (CustomerID)? Find it in the Sales.SalesOrderHeader table.
Instruction: Write a query that returns the order details for customer 15148.
3. Write a query that returns all the orders in the Sales.SalesOrderHeader table that were issued on 31/07/2013 (OrderDate).
4. Write a query that returns all the product details from the Production.Product table for all the products with the Color Black.
5. Display the product details for the products from the Production.Product table with a List Price of 1079.99 or lower.
Display the following columns: ProductID, Color, ListPrice.
6. Display the product details for the products from the Production.Product table with a List Price above 3000.
Display the following columns: ProductID, Color, ListPrice.
How many such products are there?

Part 3 - Filtering using AND, OR, NOT Operators

1. Write a query that returns all the product details from the Production.Product table for all the products with the Colors Silver or Black.

SQL Unit - Class practice booklet

2. Display the product details for the products from the Production.Product table with a List Price of 1079.99 or lower, except the products with a price of 0. (In other words, do not display the products with a List price of 0.)
Display the following columns: ProductID, Color, ListPrice.
3. Write a query that returns all the employee details from the HumanResources.Employee table for all the employees in the following jobs:
Research and Development Engineer
Design Engineer
Display the Employee code (BusinessEntityID), Job title (JobTitle) and User ID (LoginID).

Instruction: If you are unsure which field displays the job, write a query that shows all the data in the table. Afterwards, go over it column by column and determine which is the appropriate column, in your opinion.

4. Display the First name, Last name and ID number (BusinessEntityID) of all the people in the Person.Person table whose first name is not: Diane, James, Aaron.
5. Display the Order number (SalesOrderID) and Order date from the Sales.SalesOrderHeader table for all the orders that were issued during January 2013.
Instruction: What are the dates in the month of January. Think of the simplest way to include all the dates in a condition.
6. Display the Product ID, Color, Standard Cost and List Price from the Production.Product table **only** for the products with the Color Black, Cost higher than 1,000 and List Price lower than 3,000.

SQL Unit - Class practice booklet

Lesson 3 - Calculated Columns and Sorting

Part 1 - Calculated columns and aliases

1. Margin = the difference between the price list price of the item and its cost, i.e., the company's profit from the sale of the item.

Write a query that returns the Product ID, Product name and Margin from the Production.Product table.

To calculate the margin, use the fields ListPrice and StandardCost.

2. Continuing from the previous question, note that there are products with a list price of 0.
Adjust the query so that it only displays the products with a list price that is different than 0.
3. Write a query that returns the Product ID, List price and List price+12% (a calculated column) from the Production.Product table. Display only the products with a list price that is different than 0.
*In the presentation from today's lesson, you will find the formula for calculating a percentage change.
4. The company learned that item costs will be going up by 12%, so they want to raise prices by 12%.
Write a query that returns the following columns from the Production.Product table. only for products with a list price that is different than 0.
 - a. ProductID
 - b. ListPrice
 - c. StandardCost
 - d. Current profit (calculated column: ListPrice less StandardCost)
 - e. List Price after a 12% rise (calculated column)
 - f. Cost after a 12% rise (calculated column)

*In the presentation from today's lesson, you will find the formula for calculating a percentage change.

SQL Unit – Class practice booklet

5. Challenge question:

Continuing from the previous question, the company wants to know the difference between the new profit and the old.

Add an additional column to the query that displays the difference between the old and new profits (New profit less Old profit).

6. Continuing from the previous question, examine the results. Will the company profit or lose from the 12% price rise?

Part 2 – YEAR() function self-study practice, ORDER BY Keyword

As part of your job as analyst, you will be required to study and research independently to find solutions to the queries and tasks you will be asked to execute.

In order that you may be well prepared for the job, every few lessons, as part of the practice, I will have you learn another function/sub-topic independently. Afterwards, you will have to answer questions using the new topic you have learned.

1. On the internet, search for information about the function YEAR (). Remember to add the words SQL Server to your search in order to target relevant results.

In order to understand the method and application of the function, you must find answers to the following questions:

- What is the purpose of the function? What is the result that it returns?
 - Does the function take parameters? (Parameter = information that we give the function within the parentheses) What are they?
 - What is the syntax of the function?
 - What are some examples for using the function? (Important because it really helps to understand how to use it.)
2. Write a query that returns the Order number (SalesOrderID), Order date and the Order year (calculated column) from the Sales.SalesOrderHeader table.

SQL Unit - Class practice booklet

3. Write a query that returns the BusinessEntityID, Last name and First name from the Person.Person table. Sort the results according to Last name in ascending order,
4. Write a query that returns the BusinessEntityID, Last name and First name from the Person.Person table. Sort the results according to Last name in ascending order, and secondary sort according to First name in descending order.
5. Write a query that returns the BusinessEntityID, Last name and First name from the Person.Person table. Sort the results according to Last name in descending order, and secondary sort according to First name in descending order.
6. Write a query that returns the Employee number (BusinessEntityID), Hire Date and Year of birth (a column calculated by function on the BirthYear column) from the HumanResources.Employee table. Sort the results according to Hire date from the newest employee (will appear first) to the oldest (will appear last).
7. Continuing from the previous question, add a filter to display the data only of employees that began working in 2010.

Instruction:

In which section of the query are the data filtered?

What is the value returned by the function YEAR()?

SQL Unit – Class practice booklet

Lesson 4 – Grouping and Aggregate

Part 1 – GROUP BY Keyword

Don't be alarmed if you receive a red error message. Read the message and try to understand its meaning and fix the code accordingly.

1. How much income (SubTotal) was there in 2012 (OrderDate)?
Instruction: Write a query that answers the question, is based on the Sales.SalesOrderHeader table and filters the data by year. (Use the function YEAR().)
2. How much income (SubTotal) was there in 2013 (OrderDate)?
Instruction: Write a query that answers the question, based on the Sales.SalesOrderHeader table .
3. Examine the results of the 2 previous questions and answer the following:
 - a. Was there a rise or drop in sales?
 - b. Think what the causes for this may be (based on your general knowledge and life experience).
4. Write a query that displays the amount of orders made by each customer (CustomerID). Use the Sales.SalesOrderHeader table.
Instruction: Write a query that groups the data in the Sales.SalesOrderHeader table according to Customer ID and displays the Customer ID and a count of the number of orders. Give a significant name to the column with the number of orders per customer.
5. Continuing from the previous question, sort the query results according to the number of orders from the highest to the lowest.
6. Continuing from the previous question, add code so that the query will run only on the orders with Order Date 2013.

SQL Unit - Class practice booklet

7. Write a query that displays descriptive statistics for each Color from the Production.Product table : quantity of items of the same color, maximum list price, average list price, minimum list price,

Instruction: Write a query that retrieves data from the Production.Product table, and groups it by Color field. Display the color code, and the appropriate aggregate functions. Be sure to give significant, comprehensible names to the calculated columns.

8. Continuing from the previous question, examine the results. Note that there are colors for which the minimum price is 0.
Since a product cannot have a price to the customer of 0, copy the query and add a filter to it, so that lines with List Price 0 will not be included in the calculation.

9. Continuing from the 2 previous questions, use the mouse to select the codes of both queries and run them together. Note that both results will appear in the Results window. These are the corresponding query results.
Examine the Average Price column for the colors that had a minimum price of 0. Are there discrepancies in the average?
Pay attention to this is a very important point!
Sometimes we must filter out data that skew the calculations. Therefore, it is important to verify the data and the results.

This is a significant part of the analyst's job: critical thinking.

10. What is the most common Last Name in the Person.Person table?

Instruction: Write a query that shows how many times the same last name repeats for each Last Name in the Person.Person table. Sort the results according to the number of repetitions of the last name in descending order.

Hint: Use "Group by" and pay attention to how many fields you choose to display in the query (Select).

11. Examine the Order details (Sales.SalesOrderHeader) for 2012. Check the following in the Total Payment for Order field (SubTotal):
- What is the highest Order amount?
 - What is the lowest Order amount?
 - What is the average Order amount?
 - What is the total of the Orders?

SQL Unit – Class practice booklet

- e. How many orders (separate records in Sales.SalesOrderHeader) were issued?

Part 2 – HAVING Clause

1. Write a query based on the Sales.SalesOrderDetail table , that counts how many rows there are in each order (SalesOrderID). Display the Order number, and the quantity of lines in the order. Display data only for orders that have more than 3 lines.
2. Write a query based on the Sales.SalesOrderDetail table, that adds up the Line Total for each order (SalesOrderID). Display the Order number, and the Total for Payment for orders that have a Total for Payment above 1000.
3. How many customers made more than 20 orders? (Each row in the Sales.SalesOrderHeader table represents an order.)

Instruction: Write a query based on the data in the Order details table (Sales.SalesOrderDetail) that groups the data according to CustomerID. Add a filter after aggregation, such that only the rows with a Count higher or equal to 20 will be displayed.

4. Which jobs in the company (JobTitle) have 10 employees or more in the same job?
Display the list of Jobs (JobTitle) that answer the criteria and the number of employees in that job. Base your answers on the HumanResources.Employee table.

Instruction: Write a query that displays the Job Titles and the number of employees in each job from the HumanResources.Employee table. Add a filter that will display only the jobs with 10 employees or more.

5. Write a query based on the Sales.SalesOrderDetail table that displays the amount of each product ordered only for products with an amount above 50 units.

Instruction: Write a query based on the Sales.SalesOrderDetail table. The query groups the data according to Product ID and calculates the total number of items ordered for each item. (Pay attention to which aggregate function you are using and on which field. Use ERD.)

6. Write a query that displays the Last names from the Person.Person table for people whose last name appears 100 times or more in the Person.Person table.

SQL Unit – Class practice booklet

7. Write a query that displays the Customer ID and total purchase amount (SubTotal) for customers who purchased a total amount over 100,000 in 2012 (all the orders in that year). Base your answer on the Sales.SalesOrderHeader table.
8. Write a query based on the Sales.SalesOrderDetail table that displays the Order number and the number of lines in each order only for orders with more than 3 lines and Order numbers between 45,000 and 50,000, inclusive.

Instruction: Before you begin writing the query, examine the columns and data in the Sales.SalesOrderDetail table.

9. Write a query based on the Person.Person table that displays the Last names and the number of appearances of that name. Display only the Last names that appear between 10 and 50 times.

SQL Unit – Class practice booklet

Lesson 5 – Advanced Filtering

Part 1 – SELECT DISTINCT, SELECT TOP

1. Copy and run the following queries. What is the difference between the results?

a. Query 1:

```
select ProductSubcategoryID  
from Production.Product
```

b. Query 2:

```
select distinct ProductSubcategoryID  
from Production.Product
```

2. Write a query that displays the 5 first Department names from the HumanResources.Department table.

3. Write a query that displays all the details of the 20 products with the highest cost (StandardCost) from the Production.Product table.

4. Write a query that displays the list of Colors of the products from the Production.Product table, where each color appears only once.

5. Display the 10 items with the lowest List Price in the Production.Product table. Do not include items without a price (i.e., Price = 0).

6. What does the following query do?

```
select DISTINCT TOP 5 Color  
from Production.Product
```

7. Display the 10 Customer numbers (CustomerID) and Order totals (SubTotal) for the customers with the highest order amounts in 2012 (OrderDate). Base your answer on the Sales.SalesOrderHeader table.

8. Challenge question:

What is the number of unique (non-repeating) cities in the Person.Address table?

9. Which are the 10 orders with the highest amounts (SubTotal) in 2013 (OrderDate)?

Instruction: Figure out how to ensure that the highest amounts will be at the top?

SQL Unit - Class practice booklet

10. What does the following query return?

```
select distinct top 10 firstName  
from person.Person  
order by firstName
```

Part 2 - IN, BETWEEN OPERATORS, LIKE & Aliases

1. Display the names of the people in the People table that have one of the following Last names: Adams, Kelly, Perry, Wilson. Use the "IN" operator.
2. Write a query that displays the product code (ProductID), the Product name and the subcategory code (ProductSubcategoryID) from the Production.Product table, Display the data only for products for which the subcategory is one of the following: 2,5,9,14,15,30.
Sort the data by subcategory
3. Write a query that displays the product code (ProductID) and product cost (StandardCost) from the Production.Product table for products with a cost between 100 and 500,
4. Write a query that shows the order number (SalesOrderID), order date (OrderDate) and total for payment (SubTotal), for orders generated on the dates 10/01/2012 to 10/02/2012, inclusive . Base your answer on the Order title table.
* The dates in the question are written in dd/mm/yyyy format.
5. Write a query that displays the product code (ProductID), the product name and the Product number for all the products in the Production.Product table that begin with the letter "C".
6. Write a query that displays the product code (ProductID), the product name and the Product number for all the products in the Production.Product table that begin with the letters "C", "B" or "E".
7. Write a query that displays the product code (ProductID), the product name and the Product number for all the products in the Production.Product table that end with the number "8".
8. Display the records from the Person.Address table with the word "New" in their Address 1 line (beginning/middle/end).

SQL Unit – Class practice booklet

9. Display the First names of the people in the Person.Person table where the First name has only 5 letters, the first letter is "D" and the third letter is "N". Display the names without repetition.
10. Write a query that displays the data from the Sales.SalesOrderDetail table where the total per line (LineTotal) is between 1,000 and 5,000 (use "Between") and the Carrier Tracking Number contains the sequence F89. Sort the results by Unit price in ascending order.
11. Write a query that displays the product code (ProductID) and product name from the Production.Product table for all the products that have the word "Red" in their name and their List Price is between 600 and 1,500, inclusive.

Part 3 – IS NULL

1. Write a query that displays the product details of the products with the color NULL from the Production.Product table. Check the table for the appropriate column name
2. Write a query that displays all the order details from the Sales.SalesOrderHeader table for the orders that have data in the Sales Person ID column.
3. Write a query that displays the customer code and the highest order amount (SubTotal) in the years 2012 and 2013 for each customer in the Sales.SalesOrderHeader table. Display only sales that have values both in the Sales Person column, and in the Purchase Order Number column. Check the table for the appropriate column names.

SQL Unit - Class practice booklet

Lesson 6 - Diagram and Join Tables

Part 1-Database Diagram

1. Which column connects the following tables?

Instruction: For each row in the table, open a diagram and insert Tables 1 and 2 into it.

Check the name of the connecting column in each table and write the name in the appropriate place in the question.

Table 1 Name	Connecting Column in Table 1	Table 2 Name	Connecting Column in Table 2
Sales.SalesPerson		Sales.Store	
Production.TransactionHistory		Production.Product	
Production.BillOfMaterials		Production.Product	
Person.Person		Person.BusinessEntity	

2. Look at the Person.Address and Person.BusinessEntityAddress tables.

- What is the connection between them? How can they be connected?
- What is the significance of the data in each of the tables?

SQL Unit - Class practice booklet

Part 2-Basics: Join, Left Join, Right Join, Full Outer Join

- Below are 2 tables from the database of the Analyst in the Making College.

StudentDetails

StudentNo	StudentName	Last Name	Age	Gender
1	Ken	Hamilton	23	M
2	Rob	Krebs	40	M
3	Gail	Brown	34	F
4	Bill	Gilbert	46	M
5	Dylan	McArthur	24	M
6	Diane	Simon	32	F
7	Gigi	Shoop	39	F

StudentCourses

StudentNo	CourseNo
1	1000
1	1002
2	1000
4	1000
4	1002
4	1003
6	1002
8	1005

- Look at the data in the tables and try to understand the following:
 - What does each table represent? What is significant about the content?
 - Is there a connection between the tables? What is it?
- Take paper and pen and try to arrive at the query results manually. Write down the results.
 - JOIN / INNER JOIN:

```
select s.StudentNo,
       c.CourseNo
from   StudentDetails s
       inner join StudentCourses c
       on s.StudentNo = c.StudentNo
```

SQL Unit – Class practice booklet

b. LEFT JOIN:

```
select s.StudentNo,  
       c.CourseNo  
from   StudentDetails s  
left join StudentCourses c  
on s.StudentNo = c.StudentNo
```

c. RIGHT JOIN:

```
select s.StudentNo,  
       c.CourseNo  
from   StudentDetails s  
right join StudentCourses c  
on s.StudentNo = c.StudentNo
```

d. FULL OUTER JOIN:

```
select s.StudentNo,  
       c.CourseNo  
from   StudentDetails s  
full outer join StudentCourses c  
on s.StudentNo = c.StudentNo
```

4. What is the significance of the results of the following query?

```
select COUNT(*) as WhatIsMyMeaning  
from StudentCourses crs s  
join StudentDetails dtl  
on crs.StudentNo = dtl.StudentNo  
where dtl.Gender = 'F'
```

SQL Unit - Class practice booklet

Part 3- Join, Left Join, Right Join, Full Outer Join

General Instructions:

Use the diagram, as you were taught, or the reference pages,
1-SQL-0-AdventureWorks ERD Tables Reference

1. Write a query that links the Sales.SalesOrderHeader table to the Sales.SalesOrderDetail table and displays all the columns from both tables.
2. Write a query that links the Sales.SalesOrderDetail table to the Production.Product table and displays the following columns: SalesOrderID, ProductID, Name, ProductNumber, and LineTotal.

Think which table should be used to display the ProductID column.

3. In this query we will examine the profitability of each order record:

Write a query that links the Sales.SalesOrderDetail table to the Production.Product table and displays the following columns: SalesOrderID, ProductID, LineTotal, StandardCost, OrderQty, and the profit per order record (calculated column).

4. Write a query that links the Sales.SalesOrderHeader table to the Sales.SalesOrderDetail table and displays the following columns: SalesOrderID, OrderDate, ProductID, and LineTotal.

Display only the details of the orders from 2012.

5. Write a query that links the Sales.SalesOrderDetail table to the Production.Product table.
 - a. Display the following columns: SalesOrderID, ProductID, and Name.
Display only the details of the products for which the color is "Null".
 - b. Must the Color field be selected in the Select section In order to filter the data according to color?

SQL Unit – Class practice booklet

c. If the columns are not displayed, how can the correctness of the results be verified?

6. Write a query that links between the Sales.SalesOrderDetail table, the Production.Product table and the Sales.SalesOrderHeader table and displays the following columns: SalesOrderID, OrderDate, ProductID, Color and LineTotal.
7. Write a query that displays the quantity of products ordered each year.

Instructions: Write a query that links the Sales.SalesOrderDetail table to the Sales.SalesOrderHeader table and groups the data according to year (in a column calculated from the OrderDate field). The query will display the following columns: Year and OrderQty.

8. Write a query that displays the ProductID and LineTotal only for orders from 2011 in which the total paid (LineTotal) is greater than 1,000. (Calculate the date from the OrderDate column.)
9. Write a query that displays the customer details of each order in the Sales.SalesOrderHeader table, The following columns should be displayed: SalesOrderID, Order Date, CustomerID, First Name, Last Name, and SubTotal.

Sort the data by last name and then by first name.

Instruction: Check in ERD which are the relevant tables and what are the relationships between the tables.

10. Read the following query and explain what it does: (Don't run it in SSMS.)

```
select p.ProductID,  
       p.ProductSubcategoryID,  
       c.[Name]  
  
From Production.Product  
      left join Production.ProductSubcategory c  
            on p.ProductSubcategoryID = c.ProductSubcategoryID
```

11. Read the following queries: (Don't run them.)

SQL Unit – Class practice booklet

- a. What will the following query return? Will there be one record or more in the results?

```
select prd.ProductID
from Production.product prd
where prd.ProductID = 921
```

- b. Look at the following query. Note that this query is based on the query from the previous section, but with an extra link to the Sales.SalesOrderDetail table.

What will the query return? Will one record or more appear in the query results? Why?

```
select prd.ProductID,
       prd.[Name],
       sod.SalesOrderID

from Production.product prd
      join Sales.SalesOrderDetail so
      on sod.ProductID = prd.ProductID
where prd.ProductID = 921
```

12. In order to send marketing mailings to customers, write a query that displays the following data for each BusinessEntityID from the Person.BusinessEntityAddress table, by linking to the Person.Address table:
BusinessEntityID, AddressLine1, AddressLine2, City, StateProvinceID.
13. Continuing from the previous question, can the First and Last Names be added, as well? If so, link the table and add the relevant columns to the query results.
14. Write a query that displays the customer code and the highest order amount (SubTotal) in 2012 and 2013 for each customer from the Sales.SalesOrderHeader table. Display only the orders with values in both the salesman column and the PurchaseOrderNumber column. Check the names of the appropriate columns in the table.

SQL Unit – Class practice booklet

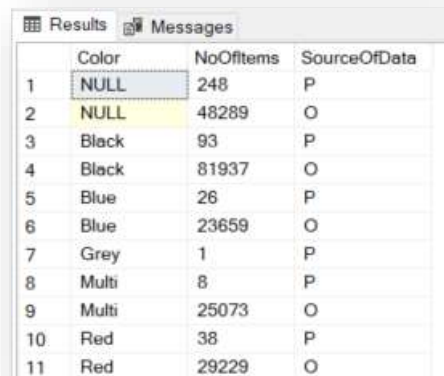
Lesson 7 – Union and Conditions

Part 1-Union, Union All

1. Write a query that returns a single list of all customer numbers from the Sales.Customer table and sales people from the Sales.SalesPerson table. Check the names of the appropriate columns in the table.
2. Write a query that displays the ProductID for the products that meet at least one of the following requirements. If the item meets more than one requirement, the product code should be displayed only once. Solve with union only:
 - a. The product was ordered (Sales.SalesOrderDetail) at a unit price after discount (calculated using the existing columns) greater than 1800, and the CarrierTrackingNumber starts with the letters 4E.
 - b. The order record is for a quantity of product greater than 10 units and the tracking number ends with the number 4.
3. In the following query, we want to compare the quantity of products of each color in the product table to the quantity of items of each color ordered, in order to understand which colors are ordered most by customers.
 - a. The query will return 3 columns: Color, number of items (a calculated column named NoOfItems), and the data source (a calculated column named SourceOfData).
 - b. The query will return a single row for each color from the product table. The row will contain the color, the number of products of that color and the constant text 'P', to show that the data came from the product table.
 - c. In addition, the query will return one row for each color from the Sales.SalesOrderDetail table. The row will contain the color, the number of products of that color ordered and the constant text 'O', to show that the data came from the orders table.
 - d. Sort the results according to color code.

SQL Unit - Class practice booklet

- e. Following is a screenshot of the beginning of the query results for reference:



	Color	NoOfItems	SourceOfData
1	NULL	248	P
2	NULL	48289	O
3	Black	93	P
4	Black	81937	O
5	Blue	26	P
6	Blue	23659	O
7	Grey	1	P
8	Multi	8	P
9	Multi	25073	O
10	Red	38	P
11	Red	29229	O

Part 2-Case When

- Write a query based on the Person.Person table, that displays the following data:
 - First name
 - Last name
 - A column named TitleEdited that will contain the following data:
If there is a value in the Title column, it will display it, and if there is no value, it will display "No Title".
- Write a query based on the Production.Product table, that displays the following data:
 - ProductID
 - Name
 - A column named StyleEdited that will contain the information to whom the model is suited, according to the value in the Style column and the following key:
 - M → Man
 - W → Woman
 - U → Unisex

SQL Unit - Class practice booklet

d. No value → Accessories

3. Write a query that ranks each row in the Sales.SalesOrderDetail table, and displays the following data:

SalesOrderID, OrderQty, Group Code (details below).

The Group code will be based on the value that appears in the Order Quantity column, and the following key:

- a. up to one item = D
 - b. 2-5 items (inclusive) = C
 - c. 6-30 items (inclusive) = B
 - d. more than 30 = A
4. Challenge question (Continuation from the previous question)

The previous query produced a list of all the order records with the rank of each record according to the quantity of items ordered.

Now, we want to refine the display to see how many times each group code appears. To do this, write a query that shows how many times each group code (A, B, C, D -according to the data in the previous question) appears in the Sales.SalesOrderDetail table.

Instruction: Look at the results of the previous query, and think how the answer could be calculated manually.

5. In order to segment employees according to gender and marital status, write a query based on the HumanResources.Employee table that shows the number of employees in each segment of gender and family status.

To make the results clearer, use the following key to change the displayed data:

- a. Gender column:
 - F → Female
 - M → Male

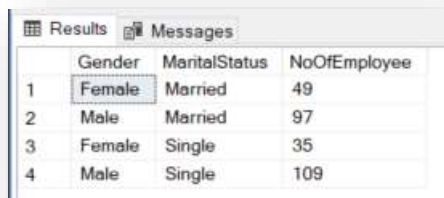
- b. Marital Status column:

SQL Unit - Class practice booklet

- S → Single
- M → Married
- Any other value → Other

Note: Currently the values in this column are only 'S' or 'M', but since there are other family statuses (e.g., widowed, divorced, etc.), the query should support the other options and classify them as 'other'.

c. Following is a preview of the results:



	Gender	MaritalStatus	NoOfEmployee
1	Female	Married	49
2	Male	Married	97
3	Female	Single	35
4	Male	Single	109

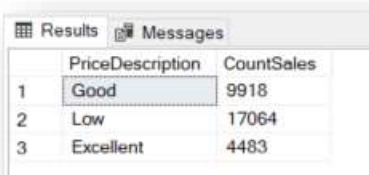
- Write a query that displays the SubTotal of every order from the Order Header table according to the following rules:
 - All orders under 1000 → Low
 - All orders of 1000 or more, but less than 3000 → Good
 - All other orders → Excellent

7. Challenge question:

Continuing from the previous question, now display how many orders of each price type there are.

Instruction: Before you start solving it, think about the way you would solve it if you were doing it manually.

Following is a preview of the results:



	PriceDescription	CountSales
1	Good	9918
2	Low	17064
3	Excellent	4483

SQL Unit – Class practice booklet

Lesson 8 – Built-in and Window Functions

Part 1 – Built-in Aggregate Functions:

Built-in Functions on Number, String and Date Fields

Use the lesson presentation to help solve the questions in this section,

1. What is the maximum OrderQty that was ordered in one order record in the Sales.SalesOrderDetail table?
2. How many different products (ProductID) were ordered in 2012?
Base your answer on the Sales.SalesOrderDetail and Sales.SalesOrderHeader tables.
3. How many letters are there in the longest FirstName in the Person.Person table?
4. In order to analyze the orders, write a query that displays the following data for each order in the Sales.SalesOrderHeader table:
SalesOrderID, OrderDate, the year of the order, the month and the day of the week.
5. Which day of the week has the highest number of orders?

In order to check the distribution of orders over the days of the week, write a query that shows how many orders were generated on each day of the week. Sort the results in descending order.

Instruction: Take the data from the Sales.SalesOrderHeader table. Use the function and operations that were taught in the lesson.

6. Which day of the week has the highest order amount?

Write a query that displays the total order amount for each day of the week.

Instruction: Take the data from the Sales.SalesOrderHeader table.
Use the function and operations that were taught in the lesson.

7. Continuing from the two previous questions, is there a correlation between the number of orders each day of the week and the profitability on that day?

If there are differences in the results of the query, what can cause this difference?

SQL Unit - Class practice booklet

8. Write a query, based on the product and order details tables, that displays the product type (a calculated field - will be defined later), the number of items ordered and the LineTotal for each type of product.

Product type definition:

ProductType is a calculated field, designated by the two left characters in the ProductNumber column.

For example:

	ProductType	ProductNumber
1	AR	AR-5381
2	BA	BA-8327
3	BB	BB-7421

9. Continuing from the previous question, in order to understand each product type, link the data to the Production.ProductSubcategory table. Start out from the previous query and add the Name column from the Subcategories table.

Instruction:

Examine the query, and consider how to add the Name column to the display so that the query will abide by the syntax rules that were taught.

Preview of the results:

	ProductType	Name	TotalQty	TotalAmount
1	BK	Mountain Bikes	28321	36445443.937380
2	BK	Road Bikes	47196	43909437.508212
3	BK	Touring Bikes	14751	14296291.259139
4	HB	Handlebars	3950	170591.320980
5	BB	Bottom Brackets	921	51826.374000
6	FB	Brakes	789	50299.311000
7	RB	Brakes	246	15719.400000
8	CH	Chains	774	9377.710144

SQL Unit - Class practice booklet

10. To be able to send marketing mailings to customers and employees, display the full name from the Person.Person table and the PhoneNumber listed in the Person.PersonPhone table.

Note that the full name consists of: FirstName, MiddleName, LastName, and should appear in one column that connects the three columns.

Define the full name column in the two ways that were taught. Examine the results and determine which way is the correct solution.

Preview of the results:

	FullNameConcat	FullName+	PhoneNumber
1	Syed E Abbas	Syed E Abbas	926-555-0182
2	Catherine R. Abel	Catherine R. Abel	747-555-0171
3	Kim Abercrombie	NULL	334-555-0137
4	Kim Abercrombie	NULL	919-555-0100
5	Kim B Abercrombie	Kim B Abercrombie	208-555-0114

11. Starting from the HumanResources.Employee table, link the Person.Person table to it, and display the following columns for each employee: Full name of the employee (in the preferred method from the previous question, concat function), date of the employee's birthday (BirthDate) and employee's age today. (Today = the day the query is run.)

Preview of the results:

(The dates that appear are correct for the query run on 05/09/2021, in dd/mm/yyyy format.)

	FullName	BirthDate	Age
1	Ken J Sánchez	1969-01-29	52
2	Terri Lee Duffy	1971-08-01	50
3	Roberto Tamburello	1974-11-12	47
4	Rob Walters	1974-12-23	47
5	Gail A Erickson	1952-09-27	69
6	Adi H Shoham	1959-03-11	62

SQL Unit – Class practice booklet

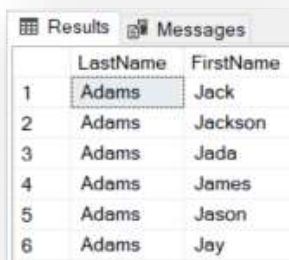
Part 2-Window Functions

1. Introductory Question:

Display the last names and first names of all the people who have the last name Adams and a first name that starts with the letter J. Sort the data by last name + first name.

Base your answer on the person.person table.

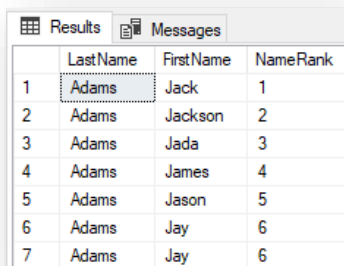
A preview of the results:



	LastName	FirstName
1	Adams	Jack
2	Adams	Jackson
3	Adams	Jada
4	Adams	James
5	Adams	Jason
6	Adams	Jay

2. Continuing from the previous question, add a column called NameRank in which you rank the results so that for each last name there is an internal ranking according to the alphabetical order of the first names.

A preview of the results:



	LastName	FirstName	NameRank
1	Adams	Jack	1
2	Adams	Jackson	2
3	Adams	Jada	3
4	Adams	James	4
5	Adams	Jason	5
6	Adams	Jay	6
7	Adams	Jay	6

SQL Unit - Class practice booklet

- Continuing on, copy the query and add another column called NameDenseRank in which you rank the results with the DENSE_RANK function, so that for each last name, there is an internal ranking according to the alphabetical order of the first name.

Examine the differences in the results between RANK and DENSE_RANK.

A preview of the results:

	LastName	FirstName	NameRank	NameDenseRank
1	Adams	Jack	1	1
2	Adams	Jackson	2	2
3	Adams	Jada	3	3
4	Adams	James	4	4
5	Adams	Jason	5	5
6	Adams	Jay	6	6
7	Adams	Jay	6	6
8	Adams	Jenna	8	7

- Display the orders generated on the dates 01/01/2013 - 02/01/2013, based on the Order heading table.
Rate each day's orders from the order with the highest SubTotal amount (rating 1) to the lowest. If there are orders with identical amounts, they receive the same rating, and then the rating continues from the next number.

A preview of the results:

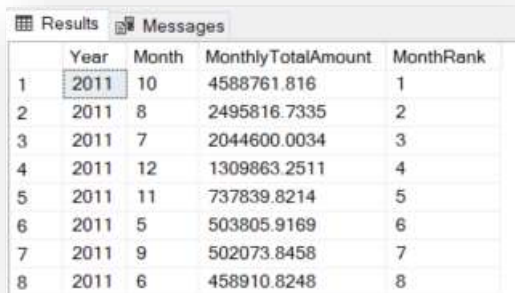
	SalesOrderID	OrderDate	SubTotal	DailyRank
1	49182	2013-01-01 00:00:00.000	2443.35	1
2	49183	2013-01-01 00:00:00.000	2443.35	1
3	49186	2013-01-01 00:00:00.000	2443.35	1
4	49187	2013-01-01 00:00:00.000	2181.5625	2
5	49181	2013-01-01 00:00:00.000	2181.5625	2
6	49188	2013-01-01 00:00:00.000	2049.0982	3
7	49184	2013-01-01 00:00:00.000	2049.0982	3
8	49185	2013-01-01 00:00:00.000	2049.0982	3
9	49190	2013-01-01 00:00:00.000	1000.4375	4

SQL Unit - Class practice booklet

5. Write a query that displays a line for each month of the year (i.e., a line for each of the months: January 2011, February 2011 ... January 2012, February 2012...), and rank the months of each year separately according to the total sales (SubTotal) in that month. (2011 has its own ranking, and the ranking starts again for 2012.)

Sort the query results by year, and ranking.

A preview of the results:



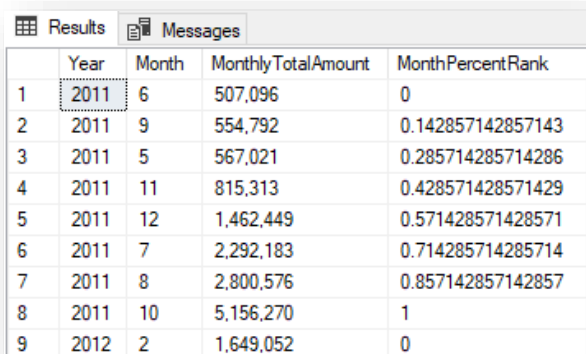
	Year	Month	MonthlyTotalAmount	MonthRank
1	2011	10	4588761.816	1
2	2011	8	2495816.7335	2
3	2011	7	2044600.0034	3
4	2011	12	1309863.2511	4
5	2011	11	737839.8214	5
6	2011	5	503805.9169	6
7	2011	9	502073.8458	7
8	2011	6	458910.8248	8

6. Continuing from the previous question, copy the query code, replace the ranking function with the percent_rank() function and run the query.
(This function does not turn pink, which is fine.)

Replace the sorting within the ranking to ascending.

What is the significance of the ranking?

A preview of the results:



	Year	Month	MonthlyTotalAmount	MonthPercentRank
1	2011	6	507,096	0
2	2011	9	554,792	0.142857142857143
3	2011	5	567,021	0.285714285714286
4	2011	11	815,313	0.428571428571429
5	2011	12	1,462,449	0.571428571428571
6	2011	7	2,292,183	0.714285714285714
7	2011	8	2,800,576	0.857142857142857
8	2011	10	5,156,270	1
9	2012	2	1,649,052	0

SQL Unit - Class practice booklet

Lesson 9 - Unrelated Nested Queries

Part 1 - Unrelated Nested Queries:

A sub-query that returns a single value or a list (one column)

1. Write a query that displays the ProductID, the ListPrice, and the average list price of all the items in the product table.
2. Continuing from the previous question, name the column with the average list price "AverageListPrice".

In addition, make sure that the average price list price is calculated only with the items with a list price greater than 0, so as not to skew the result.

3. Write a query that displays the ProductID and the Item Color from the Production.Product table for the items with the color identical to that of item number 741.
4. Write a query that displays the BusinessEntityID and Gender of all the employees in the employee table whose gender is the same as the gender of the employee with code 38.
5. Continuing from the previous question, add the first and last names of the employees from the Persons table. Use the diagram or ERD to check which column links the tables.
6. Write a query that displays the orders from the Sales.SalesOrderHeader table that have a SubTotal lower than the average of the SubTotals of all the orders. Display only the order number.
7. Continuing from the previous question, display how many orders meet the condition.
8. Write a query that displays, the product code, price per item after discount (calculated column), and the difference between the LineTotal of each order record and the average of the LineTotals (a calculated column, named DiffFromAVG) for all the records in the order details table.
9. Continuing from the previous question, write a query that displays the product codes and names of all the products in the products table that were ordered at least once in 2013.

SQL Unit – Class practice booklet

10. Continuing from the previous question, write a query that displays the product codes and names of all the products in the product table where the total quantity ordered in 2013 was at least 300 units.
11. In this query, you must check the quantity and value of orders in 2013, of the ten products with the highest quantity of orders in 2012. In other words, check how the ten products that were ordered the most in 2012 functioned in 2013. (Were they ordered many times? Not ordered at all? Are they still profitable?)

Instructions: Write a query that shows the order number, product code, product name, quantity of items in the order, and LineTotal per order record of the products ordered in 2013. The query results should show the data for only the ten best-selling products in 2012.

Think which tables and columns are involved in the query. Use the ERD for assistance.

12. Challenge Question:

Continuing from the previous question, write a query that displays the following data for each of the ten most ordered products in 2012: product code, product name, total quantity of items ordered in 2013 and total order amount in 2013.

Think which tables and columns are involved in the query. Use the ERD for assistance.

SQL Unit - Class practice booklet

13. Challenge Question:

Continuing from the previous question,
write a query showing the total quantity of items ordered and total order amount
for 2012 and 2013 of the ten most ordered products in 2012.

what can be deduced from the results of the query?
(query result attached)

	ProductID	Name	TotalQty2012	TotalQty2013	TotalAmount2012	TotalAmount2013
1	760	Road-650 Red, 60	1384	451	630305.162822	215197.597992
2	770	Road-650 Black, 52	1362	493	627096.191537	237006.923153
3	852	Women's Tights, S	1475	595	64768.473052	26464.840884
4	854	Women's Tights, L	1432	687	63301.766139	30593.902769
5	715	Long-Sleeve Logo Jersey, L	2113	2910	59545.586170	89058.134836
6	863	Full-Finger Gloves, L	2380	991	49009.451581	20760.908165
7	862	Full-Finger Gloves, M	1526	677	33206.280205	14929.815467
8	711	Sport-100 Helmet, Blue	1519	3088	29554.974982	76242.605403
9	708	Sport-100 Helmet, Black	1387	3088	27045.651028	76703.598245
10	712	AWC Logo Cap	2048	3768	10398.981116	23582.958419

14. Challenge Question:

An order for a single item is an order that has only one order line.

Write a query that displays the SalesOrderID and ProductID of single item orders.
Decide which table you should use.

15. Write a query that displays all the products from the products table that were
never ordered.

Part 2 - Unrelated Nested Queries:

A subquery that returns a table (several columns)

1. Preparation for the next question:

Write a query based on the data from the order details table that displays the
product code, total quantity ordered, and total amount to be paid (LineTotal) for
each product code.

SQL Unit - Class practice booklet

- Continuing from the previous question, write a query based on the order details and products tables, that displays the following data for each product code: product code, Name, ProductNumber, color, total quantity ordered, LineTotal.

Instructions: Write a query that returns the product details from the product table. Also, use the query you wrote in the previous question as a sub-query that returns a table, and link between the two tables using JOIN.

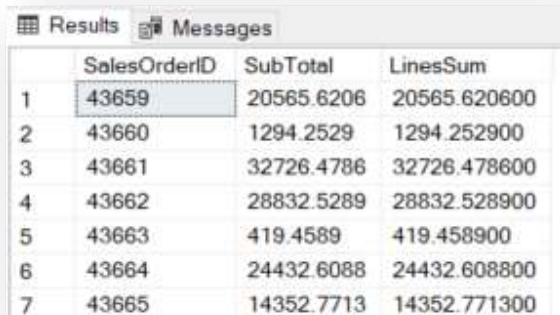
Remember, when using a sub-query as a table, the sub-query must be named.

- In this question you must examine the numerical data and their relationship with the order header and order details tables.

Question: Does the SubTotal column in the order header table contain the sum of all the rows in the LineTotal column of the Order Details table for that same order?

Instructions: Write a query, based on the Order header and Order details tables, that displays the following columns: Order Number, SubTotal from the Order header table, total of the LineTotals from the Order Details table.

A preview of the results:



	SalesOrderID	SubTotal	LinesSum
1	43659	20565.6206	20565.620600
2	43660	1294.2529	1294.252900
3	43661	32726.4786	32726.478600
4	43662	28832.5289	28832.528900
5	43663	419.4589	419.458900
6	43664	24432.6088	24432.608800
7	43665	14352.7713	14352.771300

- Continuing from the previous question, it is difficult to tell from the results whether there are lines with differences between the sums. So we will refine the query:

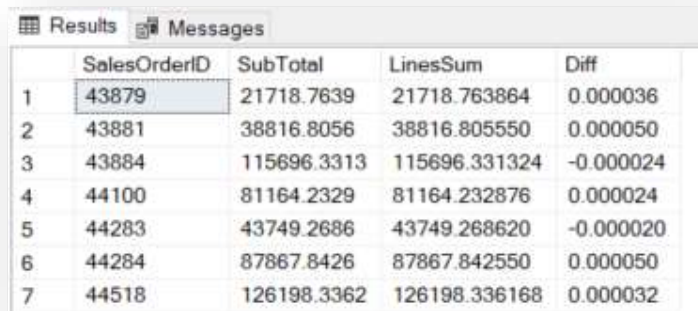
In the query, add a column called Diff, which shows the difference between the total payment from the Order details table and the total payment from the Order header table.

SQL Unit - Class practice booklet

5. Continuing from the previous question, examine the results of the previous query. Note that there are many order lines that do not have any differences, which is great.

Add an instruction to the query to display only the lines with a difference (Diff).

A preview of the results:



	SalesOrderID	SubTotal	LinesSum	Diff
1	43879	21718.7639	21718.763864	0.000036
2	43881	38816.8056	38816.805550	0.000050
3	43884	115696.3313	115696.331324	-0.000024
4	44100	81164.2329	81164.232876	0.000024
5	44283	43749.2686	43749.268620	-0.000020
6	44284	87867.8426	87867.842550	0.000050
7	44518	126198.3362	126198.336168	0.000032

6. Continuing from the previous question, examine the results. What is the range of differences? That is, what is the lowest difference and what is the highest difference?

To answer this question, simply sort the results of the previous query according to the value in the Diff column.

7. Explanation of the previous questions:

As analysts, we research the data. Sometimes we know in advance exactly what we want to check, but in most cases we realize that we can improve the query by adding columns with relevant information, grouping data and schema, adding or changing sorting, etc. only after we get the query results.

SQL is a tool aimed at helping us, the analysts, get the information in the most convenient format for analyzing the data. Therefore, when researching topics, be sure to examine the results of your analysis and see if it would be better for you to get the data in a different format.

If so, just modify the query accordingly.

SQL Unit – Class practice booklet

8. Write a query that shows the ProductID, Name, ListPrice, ProductSubcategoryID and the difference between the list price and the average list price of all the products in the same sub-category for each product in the Production.Product table.

Sort it by subcategory, in ascending order.

Include in the calculation of the average list price only products with a ListPrice and with ProductSubcategoryID that is not NULL

SQL Unit – Class practice booklet

Lesson 10 – Related Nested Queries

Part 1 – Related Nested Queries, Exists

1. Write a query that displays all the names of the products in the products table that were ordered at least once (Sales.SalesOrderDetail).

Solve this twice: once by using In, and a second time by using Exists.

2. Write a query that displays the Name of the product from the Production.Product table that has the word "Wheels" in its sub-category name in the Production.ProductSubcategory table. Solve this using Exists
3. Write a query that displays the data of all the people from the Person.Person table who ordered a product in 2013.

Instruction: Consider which tables must be used in the query. (Hint: 3 tables.) Note that each row with person details should appear only once – no more. Solve this using Exists.

4. What does the following query return?

```
select      pe.BusinessEntityID,
            pe.LastName,
            pe.FirstName
from Person.Person pe
      join Sales.Customer sc
            on pe.BusinessEntityID = sc.PersonID
where exists(
            select *
            from Sales.SalesOrderHeader sh
                  join Sales.SalesOrderDetail sd
                        on sh.SalesOrderID = sd.SalesOrderID
                  join Production.Product pr
                        on sd.ProductID = pr.ProductID
            where pr.ProductSubcategoryID in (1, 2, 3)
                  and pr.StandardCost > 600
                  and sh.CustomerID = sc.CustomerID
            )
```

SQL Unit - Class practice booklet

5. Write a query that displays all the columns from the Sales.SalesPerson table but displays only the salespeople who have sold at least one product with the word "frame" in its model name.

Instruction:

- Which tables are required for this query? (Hint: 4 tables.)
 - Consider which tables link the Sales.SalesPerson table to the Production.ProductModel table, with the knowledge that each item from the Production.Product table has its own ProductModelID.
 - Write the outer query, i.e., what is returned as the result of the query.
 - Add Exists to the filter, and write the sub-query with the connections between the tables (Join).
 - Connect the sub-query to the query that contains it.
6. Write a query that displays the first name, last name, JobTitle and the number of employees in that department from the HumanResources.Employee table.

Use the HumanResources.Employee and Person.Person tables.

Note: This may be solved in several ways. One way includes a link between the internal and outer query, without using Exists. Another solution uses Unrelated Nested Queries.

A preview of the results:

	LastName	FirstName	Job Title	AmountInDepartment
1	Moreland	Barbara	Accountant	2
2	Seamans	Mike	Accountant	2
3	Liu	David	Accounts Manager	1
4	Tomic	Dragan	Accounts Payable Specialist	2
5	Sheperdigian	Janet	Accounts Payable Specialist	2
6	Poe	Deborah	Accounts Receivable Specialist	3
7	Spoon	Candy	Accounts Receivable Specialist	3
8	Walton	Bryan	Accounts Receivable Specialist	3
9	Berg	Karen	Application Specialist	4
10	Meyyappan	Ramesh	Application Specialist	4

SQL Unit - Class practice booklet

Lesson 11 - CTE (Common Table Expressions)

Part 1 - CTE (Common Table Expressions)

1. Write a query that defines a CTE named "Sales_CTE", which is based on the Order header table, and contains the following columns: Order Number, SalesPersonID, the year in which the order was placed.

Analyze only the data that has a value in the SalesPersonID column.

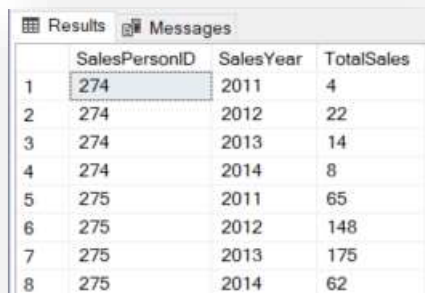
Use the CTE you defined, and write a query that shows the amount of orders that each salesperson generated each year.

Sort the query results by SalesPersonID and year.

Note: With the results of the query, it is possible to evaluate the salespeople's performance, and see whether the salesperson improved their performance or it needs to be strengthened.

In addition, take into account when examining the results that data is not provided for all the months in 2011 and 2014.

A preview of the results:

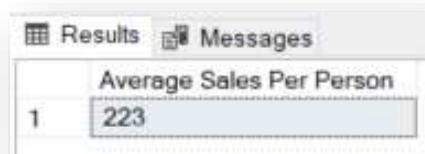


	SalesPersonID	SalesYear	TotalSales
1	274	2011	4
2	274	2012	22
3	274	2013	14
4	274	2014	8
5	275	2011	65
6	275	2012	148
7	275	2013	175
8	275	2014	62

2. Write a query that shows the average number of orders for all the years, for all the salespeople. Analyze only the data that has a value in the SalesPersonID column:
 - a. Define a CTE named "Sales_CTE", which is based on the Order header table, and contains the following columns: SalesPersonID, the amount of orders generated by the salesperson.
 - b. Write a query based on the CTE that displays the average order quantity of all the salespeople.

SQL Unit - Class practice booklet

c. A preview of the results:



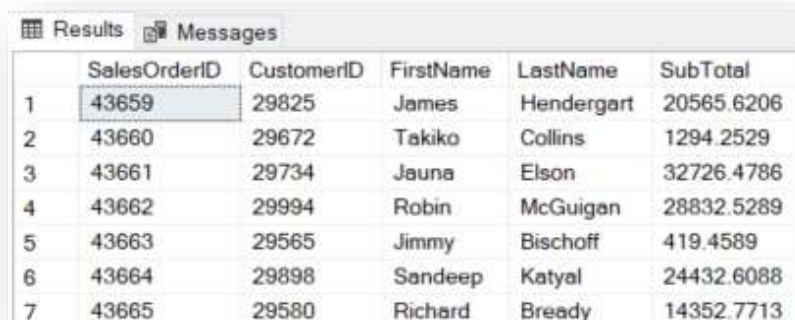
Average Sales Per Person	
1	223

3. Write a query that defines a CTE named "Person_CTE", which is based on the Sales.Customer and Person.Person tables and contains the following columns: CustomerID, first name and last name.

The purpose of the CTE is to make it easier for us to link the Sales.SalesOrderHeader table to the Person.Person table.

Use the Order header table and the Person_CTE that you defined to write a query that displays the order number, customer number, first name, last name, and SubTotal.

A preview of the results:



	SalesOrderID	CustomerID	FirstName	LastName	SubTotal
1	43659	29825	James	Hendergart	20565.6206
2	43660	29672	Takiko	Collins	1294.2529
3	43661	29734	Jauna	Elson	32726.4786
4	43662	29994	Robin	McGuigan	28832.5289
5	43663	29565	Jimmy	Bischoff	419.4589
6	43664	29898	Sandeep	Katyal	24432.6088
7	43665	29580	Richard	Bready	14352.7713

4. Write a query that displays the amount of orders per product color in 2013, sorted from highest to lowest.

Use a CTE to simplify the query:

- a. Define a CTE named "Sales_CTE", which contains the following columns: Order number, OrderDate, ProductID and OrderQty.
- b. Use the CTE and the Product table to display the amount of orders per product color in 2013, sorted from highest to lowest

SQL Unit – Class practice booklet

- c. Note: The query could also be solved without using a CTE. However, it helps to keep the code clean, tidy and legible when splitting the queries into parts.

A preview of the results:



	Color	SalesQty
1	Black	35134
2	NULL	25921
3	Yellow	19536
4	Blue	14214
5	Silver	13309
6	Multi	10721
7	Red	7455

Part 2 – Multiple CTE's (Common Table Expressions)

1. In order to prepare an annual order report by customer, which includes both the customer's details and the summarized sales data, proceed according to the following instructions:
 - a. Define a Person_CTE, based on the Sales.Customer and Person.Person tables, which displays the following columns: CustomerID, First name, and Last name.
 - b. Define a Sales_CTE that compiles the number of orders and SubTotals for each year and customer, in the following columns: CustomerID, Year, Total order quantity and SubTotal.
 - c. Join between the two CTEs you created in the two preceding sections. Display only the data for 2012.
 - d. A point to consider: Where should the filter to display only 2012 data be placed – in the query or in the CTE?

SQL Unit - Class practice booklet

2. In order to compare the quantity of items offered and the quantity ordered for each color, write a query that displays the product color, the quantity of units of each color ordered, the quantity of order rows per color, and the quantity of items of each color offered in the Products table.

Instruction:

- e. Define a CTE that is based on the Order details and Products tables and contains the following columns: color, total order quantity and quantity of order rows.
- f. Define a CTE that is based on the Products table only and contains the following columns: color, quantity of products per color.
- g. Use the two CTEs you defined, and display the following columns for all the colors (i.e., without colorless products): color, total order quantity, total orders, and quantity of products of this color in the Products table.

A preview of the results:

Results		Messages		
	Color	SalesQty	NoOfSales	NoOfItems
1	Black	81937	30099	93
2	Blue	23659	9044	26
3	Multi	25073	8502	8
4	Red	29229	13065	38
5	Silver	25023	10654	43
6	Silver/Black	3931	1517	7
7	White	5217	1429	4
8	Yellow	32556	12862	36

SQL Unit – Class practice booklet

Lesson 12 – Review and Summary Practice on Data Retrieval

Part 1 – All subjects learned so far

General instructions:

- The practice begins with very simple questions that gradually increase in difficulty and complexity.
- A short description (one line) should be added before each query.
- Refer to the document containing the ERD and read the explanations about the company's Purchase order tables carefully.
- Before starting to work, examine the tables and the data within the tables, and make sure that you understand the data, their significance and the connections between the tables in the ERD.

Research Question

The purpose of this interim practice is to examine the topic of orders from vendors. In this practice, you will write queries that will answer the following business questions:

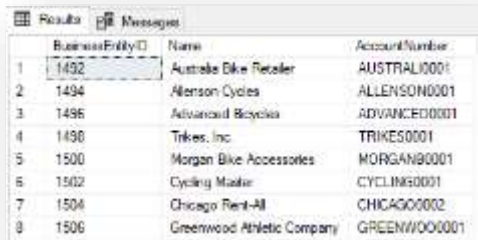
- Who are the main vendors of the store?
- Are there any vendors who supply more than one product? Which products?
- Which vendors do not deliver a large percentage of the order?
- Are there products supplied by more than one vendor ? Which vendors have the fastest average delivery time?
- Etc.

SQL Unit - Class practice booklet

The analysis

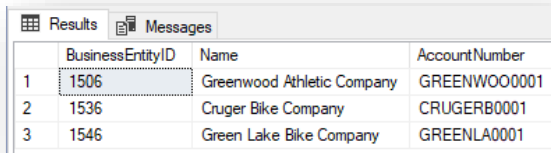
1. Write a query that displays the BusinessEntityID, Vendor name and Account number from the Vendor table.

Examine the query results. How many records are there in the table?



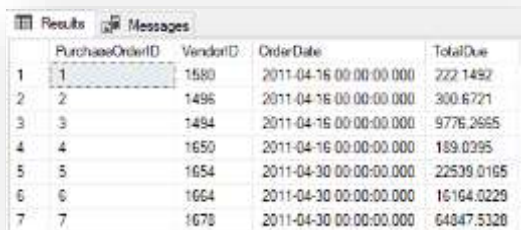
	BusinessEntityID	Name	AccountNumber
1	1492	Australe Bike Retailer	AUSTRALI0001
2	1494	Alenson Cycles	ALLENSON0001
3	1496	Advanced Bicycles	ADVANCED0001
4	1498	Trikes, Inc.	TRIKES0001
5	1500	Morgan Bike Accessories	MORGAN0001
6	1502	Cycling Master	CYCLING0001
7	1504	Chicago Rent-All	CHICAGO0002
8	1506	Greenwood Athletic Company	GREENWOO0001

2. Write a query that displays the BusinessEntityID, Vendor name and Account number from the Vendor table for vendors whose names end with the word "Company".



	BusinessEntityID	Name	AccountNumber
1	1506	Greenwood Athletic Company	GREENWOO0001
2	1536	Cruger Bike Company	CRUGERB0001
3	1546	Green Lake Bike Company	GREENLA0001

3. Write a query that displays the Purchase order ID, Vendor ID, Order date, and total cost per order from the Purchase order header table.



	PurchaseOrderID	VendorID	OrderDate	TotalDue
1	1	1580	2011-04-16 00:00:00.000	222.1492
2	2	1496	2011-04-16 00:00:00.000	300.6721
3	3	1494	2011-04-16 00:00:00.000	9776.2685
4	4	1650	2011-04-16 00:00:00.000	189.0395
5	5	1654	2011-04-30 00:00:00.000	22539.0165
6	6	1664	2011-04-30 00:00:00.000	16164.0229
7	7	1678	2011-04-30 00:00:00.000	64847.5328

SQL Unit - Class practice booklet


4. Continuing from the previous section, add also the vendor's name to the table. Consider which table can provide the vendor name, and which kind of table connection should be used. (Check the different JOIN types, and whether all the vendors appear in the Vendor table.)



	PurchaseOrderID	VendorID	Name	OrderDate	TotalDue
1	1	1580	Litware, Inc.	2011-04-16 00:00:00.000	222.1492
2	2	1496	Advanced Bicycles	2011-04-16 00:00:00.000	300.5721
3	3	1494	Alenson Cycles	2011-04-16 00:00:00.000	9776.2665
4	4	1650	American Bicycles and Wheels	2011-04-16 00:00:00.000	185.0395
5	5	1654	American Bikes	2011-04-30 00:00:00.000	22539.0165
6	6	1654	Anderson's Custom Bikes	2011-04-30 00:00:00.000	16164.0229
7	7	1678	Proseware, Inc.	2011-04-30 00:00:00.000	64847.5328
8	8	1616	Aurora Bike Center	2011-04-30 00:00:00.000	766.1827

5. Continuing from the previous question, add also the account number from the Vendors table. Display only the orders issued in 2012, and only those from vendors whose account numbers begin with the letters A-I and end with the number 2.

In order to filter by vendor name, must the "vendor name" field be chosen in select, as well?



	PurchaseOrderID	VendorID	Name	OrderDate	TotalDue	AccountNumber
1	32	1646	Electronic Bike Repair & Supplies	2012-01-08 00:00:00.000	31817.5358	ELECTRON0002
2	45	1526	International Bicycles	2012-01-16 00:00:00.000	31160.2541	INTERNAT0002
3	84	1654	American Bikes	2012-02-09 00:00:00.000	22539.0165	AMERICAN0002
4	99	1504	Chicago Rent-All	2012-03-08 00:00:00.000	609.8274	CHICAGO0002
5	103	1694	Complete, Inc.	2012-03-08 00:00:00.000	3785.0113	COMPETE0002
6	111	1646	Electronic Bike Repair & Supplies	2012-03-08 00:00:00.000	21256.3401	ELECTRON0002
7	124	1526	International Bicycles	2012-03-09 00:00:00.000	31160.2541	INTERNAT0002
8	163	1654	American Bikes	2012-05-30 00:00:00.000	22539.0165	AMERICAN0002

6. For each vendor, display the purchase orders that were ordered from them in 2012-2013.
- Display the following columns:
Purchase order ID, Vendor ID, Vendor name, Order date, and total cost per order.
 - Sort by Vendor ID and date in ascending order.

SQL Unit - Class practice booklet

	PurchaseOrderID	VendorID	Name	OrderDate	TotalDue
1	88	1492	Australia Bike Retailer	2012-02-09 00:00:00.000	725.5276
2	167	1492	Australia Bike Retailer	2012-05-30 00:00:00.000	785.6053
3	246	1492	Australia Bike Retailer	2012-09-21 00:00:00.000	623.4371
4	325	1492	Australia Bike Retailer	2013-04-25 00:00:00.000	1654.7486
5	404	1492	Australia Bike Retailer	2013-06-25 00:00:00.000	308.116
6	434	1492	Australia Bike Retailer	2013-08-05 00:00:00.000	315.3212
7	513	1492	Australia Bike Retailer	2013-08-11 00:00:00.000	445.0139
8	596	1492	Australia Bike Retailer	2013-08-18 00:00:00.000	553.8221
9	683	1492	Australia Bike Retailer	2013-08-25 00:00:00.000	396.2834

7. Write a query that collects the following data from the Purchase order header table for each vendor:

Vendor ID, Order Amount and total cost for all orders.

Sort the results by total cost per order in descending order.

	VendorID	OrderAmount	SumTotalDue
1	1576	50	5034266.74
2	1684	50	3379946.3215
3	1696	51	3347165.1965
4	1680	51	2821333.516
5	1578	50	2777684.911
6	1632	50	2675889.216
7	1678	51	2593901.3109

8. Continuing from the previous question, add a column with the total amount of items stocked from each vendor.

Note that the quantity of items entered into stock (StockedQty) is in the Purchasing.PurchaseOrderDetail table. Consider how best to add the required column.

Hint: Sub-query or CTE

	VendorID	OrderAmount	SumStockedQty	SumTotalDue
1	1590	50	117320.00	29006.25
2	1568	51	112980.00	1315488.5502
3	1696	51	96066.00	3347165.1965
4	1652	50	76898.00	2052173.6244
5	1684	50	76788.00	3379946.3215
6	1508	51	67052.00	1731662.6838
7	1544	51	66944.00	324301.4796
8	1694	51	66475.00	350018.4197

SQL Unit - Class practice booklet

9. Write a query that displays the vendor for each product.

If there are two vendors for the same product, the product will appear twice in the list: once for the first vendor and a second time for the second vendor.

	ProductID	VendorID
1	1	1580
2	2	1688
3	4	1650
4	317	1578
5	317	1678
6	318	1578
7	318	1678
8	319	1556

10. Continuing from the previous question, display the number of different products that each vendor supplies.

Instruction: Before beginning to solve, consider how to arrive at the query results and plan the method.

	VendorID	Product Supply
1	1656	30
2	1540	24
3	1536	23
4	1562	23
5	1616	19
6	1686	19
7	1492	16
8	1666	16

11. Write a query that displays, the Purchase order ID, the ShipMethodID and the Name of the shipping method of each purchase order.

	PurchaseOrderID	ShipMethodID	Name
1	1	3	OVERSEAS - DELUXE
2	2	5	CARGO TRANSPORT 5
3	3	2	ZY - EXPRESS
4	4	5	CARGO TRANSPORT 5
5	5	4	OVERNIGHT J-FAST
6	6	3	OVERSEAS - DELUXE
7	7	3	OVERSEAS - DELUXE
8	8	5	CARGO TRANSPORT 5

SQL Unit - Class practice booklet

12. Continuing from the previous question, display the total number of orders shipped by each shipping method.

Display the following columns: ShipMethodID, Shipping method Name, Number of Orders shipped by this method. Sort by number of orders shipped by this method in descending order.

	ShipMethodID	Name	NoOfOrders
1	5	CARGO TRANSPORT 5	1523
2	4	OVERNIGHT J-FAST	1085
3	2	ZY - EXPRESS	655
4	1	XRQ - TRUCK GROUND	589
5	3	OVERSEAS - DELUXE	160

13. Continuing from the previous question, copy the query and add that only the shipping methods used for more than 500 orders will be displayed.

	ShipMethodID	Name	NoOfOrders
1	5	CARGO TRANSPORT 5	1523
2	4	OVERNIGHT J-FAST	1085
3	2	ZY - EXPRESS	655
4	1	XRQ - TRUCK GROUND	589

14. Continuing from the previous question, display only the shipping methods used for more than one quarter of the orders.

	ShipMethodID	Name	NoOfOrders
1	5	CARGO TRANSPORT 5	1523
2	4	OVERNIGHT J-FAST	1085

15. Write a query that displays all the order records for the year 2012, and how many items are missing in each order record.

- Display the following columns:
Vendor ID, Product ID, OrderQty, Quantity of items missing out of the quantity ordered. (A calculated field. The fields required for the calculation are: Quantity ordered and Stocked Quantity.)

SQL Unit - Class practice booklet

b. Sort by Vendor ID in ascending order.

	VendorID	ProductID	OrderQty	LackQty
1	1492	427	3	0.00
2	1492	428	3	0.00
3	1492	429	3	0.00
4	1492	430	3	0.00
5	1492	431	3	0.00
6	1492	432	3	0.00
7	1492	433	3	0.00

16. Continuing from the previous question, copy the previous query, adding that only the order records in which all the ordered items were not provided will be displayed.

	VendorID	ProductID	OrderQty	LackQty
1	1494	530	550	82.00
2	1498	930	550	82.00
3	1498	921	550	55.00
4	1500	358	60	60.00
5	1506	935	550	82.00
6	1506	936	550	27.00
7	1508	940	550	55.00

SQL Unit - Class practice booklet

17. Continuing from the previous questions, write a query that displays the Vendor ID, the quantity of items ordered from them and the quantity items that they failed to deliver for each vendor from whom items were ordered in 2012.

Sort by the number of missing items in descending order.

	VendorID	TotalQtyOrdered	TotalLackQty
1	1538	3300	1209.00
2	1560	3300	1209.00
3	1570	4950	1209.00
4	1678	4400	1209.00
5	1586	3300	1182.00
6	1624	3300	1182.00
7	1652	3300	1182.00
8	1588	3300	1127.00

18. Continuing from the previous question, calculate the percentage of items not delivered out of all the items ordered from the vendor.

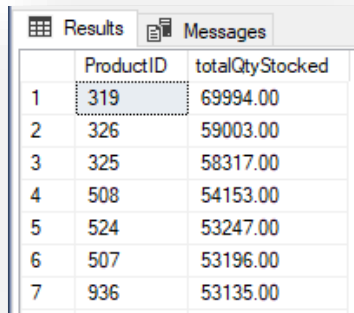
- Display the following fields:
Vendor ID, total quantity of items ordered, total quantity of missing items, percentage of missing items from all the items ordered (calculated column).
- Hint: The formula for the percentage of missing items from the ordered items: $(\text{lack_amount}) / (\text{Total_ordered_amount}) * 100$.
- Display only vendors who were short.
- Sort the data by the missing percentage in descending order.

	VendorID	TotalQtyOrdered	TotalLackQty	LackPrecent
1	1644	1650	632.00	38.303000
2	1654	1650	632.00	38.303000
3	1538	3300	1209.00	36.636300
4	1560	3300	1209.00	36.636300
5	1624	3300	1182.00	35.818100
6	1586	3300	1182.00	35.818100
7	1652	3300	1182.00	35.818100

SQL Unit - Class practice booklet

19. Write a query that shows how many items of each product (ProductID) were stocked (StockedQty).

Sort by quantity stocked in descending order.



	ProductID	totalQtyStocked
1	319	69994.00
2	326	59003.00
3	325	58317.00
4	508	54153.00
5	524	53247.00
6	507	53196.00
7	936	53135.00

20. Continuing from the previous question, display also the product name (from the Production.Product table.)



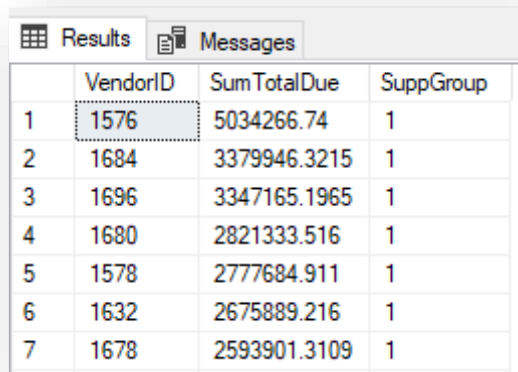
	ProductID	Name	TotalQtyStocked
1	319	HL Crankarm	69994.00
2	326	Decal 2	59003.00
3	325	Decal 1	58317.00
4	508	ML Mountain Rim	54153.00
5	524	HL Spindle/Axle	53247.00
6	507	LL Mountain Rim	53196.00
7	936	ML Mountain Pedal	53135.00
8	523	LL Spindle/Axle	52342.00

SQL Unit - Class practice booklet

21. In this question, the vendors will be sorted into 4 groups according to the quantity of items purchased from them. In this way we can distinguish who are the main vendors of the store. The groups will be formed to have the same number of suppliers in each group.

Write a query that displays the Vendor ID, the total cost of the products purchased from them (in all the years), and Group (Group 1 = the vendors who supplied the highest amount, Group 4 = the vendors who supplied the lowest amount).

Hint: Window function - ntile



	VendorID	SumTotalDue	SuppGroup
1	1576	5034266.74	1
2	1684	3379946.3215	1
3	1696	3347165.1965	1
4	1680	2821333.516	1
5	1578	2777684.911	1
6	1632	2675889.216	1
7	1678	2593901.3109	1

22. Continuing from the previous question, rank the vendors annually according to the amount purchased from them.

- Display the following fields:
Vendor ID, year, total cost of items purchased from the vendor, vendor ranking according to total cost of items in descending order (1 = the highest amount purchased in that year).

Hint: Window function - rank / dense_rank

- Display a row for each vendor and year (i.e., vendor number 1 - year 2011; vendor number 1 - year 2012; etc.)
- Sort by vendor ID.

SQL Unit - Class practice booklet

	VendorID	OrderYear	SumTotalDue	YearlyRank
1	1492	2011	767.0528	18
2	1492	2012	2134.57	50
3	1492	2013	8489.3059	61
4	1492	2014	13669.1142	70
5	1494	2014	293287.995	37
6	1494	2013	166196.5305	34
7	1494	2011	9776.2665	12
8	1494	2012	29328.7995	35

23. For each purchase order from a vendor, display the following columns:
Purchase order ID, Vendor ID, OrderDate, DueDate, number of days for the arrival of the shipment (calculated column: Due date - Order date)

Sort by number of days for the arrival of the shipment in descending order.

	PurchaseOrderID	VendorID	OrderDate	DueDate	ShipDays
1	4001	1520	2014-02-11 00:00:00.000	2014-09-22 00:00:00.000	223
2	4001	1520	2014-02-11 00:00:00.000	2014-07-02 00:00:00.000	141
3	4007	1594	2014-03-01 00:00:00.000	2014-05-14 00:00:00.000	74
4	4007	1594	2014-03-01 00:00:00.000	2014-05-14 00:00:00.000	74
5	4007	1594	2014-03-01 00:00:00.000	2014-05-14 00:00:00.000	74
6	4007	1594	2014-03-01 00:00:00.000	2014-05-14 00:00:00.000	74
7	4007	1594	2014-03-01 00:00:00.000	2014-05-14 00:00:00.000	74
8	4007	1594	2014-03-01 00:00:00.000	2014-05-14 00:00:00.000	74
9	4008	1676	2014-04-22 00:00:00.000	2014-05-22 00:00:00.000	30

24. Continuing from the previous question, display the average days for the arrival of the shipment for each vendor.

	VendorID	AvgShipDays
1	1566	14
2	1520	131
3	1658	14
4	1612	14
5	1652	14
6	1698	14
7	1506	14