

Q1: Write a SELECT statement that returns two columns: VendorName and their individual PaymentAverage, where PaymentAverage is the average of the PaymentTotal column. Return 10 vendors who've been paid the most.

```
A1: SELECT TOP 10 VendorName, AVG(PaymentTotal) AS Average FROM Vendors V
JOIN PaidInvoices i ON v.VendorID = i.VendorID
GROUP BY VendorName
ORDER BY Average DESC;
```

The screenshot displays the Microsoft SQL Server Management Studio (SSMS) interface. The query editor shows the following SQL statement:

```
SELECT TOP 10 VendorName, AVG(PaymentTotal) AS Average FROM Vendors V
JOIN PaidInvoices i ON v.VendorID = i.VendorID
GROUP BY VendorName
ORDER BY Average DESC
```

The query has been executed successfully, and the results are displayed in the Results pane. The results show the top 10 vendors by their average payment total.

VendorName	Average
Malloy Lithographing Inc.	28639.74
Data Reproductions Corp	21842.00
Digital Dreamworks	7125.34
Betelmann Industry Svcs. Inc	6940.25
Yeasman, Inc.	4901.25
United Parcel Service	2575.3288
Computerworld	2433.00
Cahners Publishing Company	2184.50
Pollstar	1750.00
Franchise Tax Board	1600.00

The Properties pane on the right shows the current connection parameters, including the connection name (local) (DESKTOP-U6AC2GB\elham) and the server name (DESKTOP-U6AC2GB\elham).

Q2: Write a SELECT statement that returns: AccountDescription, LineItemCount, and LineItemSum. LineItemCount is the number of entries in the InvoiceLineItems table that have that AccountNo. LineItemSum is the sum of the InvoiceLineItemAmount column for that AccountNo. Group the result set by account description, and sort it in ascending order of LineItemSum.

A1: USE AP

GO

```
SELECT A.AccountDescription ,COUNT(1) LineItemCount, SUM(InvoiceLineItemAmount)
LineItemSum FROM GLAccounts A
JOIN InvoiceLineItems L ON A.AccountNo = L.AccountNo
GROUP BY A.AccountDescription
ORDER BY LineItemSum DESC;
```

The screenshot displays the Microsoft SQL Server Enterprise Manager interface. The central pane shows the execution results of a query. The query is as follows:

```
USE AP
GO
SELECT A.AccountDescription ,COUNT(1) LineItemCount, SUM(InvoiceLineItemAmount) LineItemSum FROM GLAccounts A
JOIN InvoiceLineItems L ON A.AccountNo = L.AccountNo
GROUP BY A.AccountDescription
ORDER BY LineItemSum DESC
```

The results pane shows a table with three columns: AccountDescription, LineItemCount, and LineItemSum. The data is sorted by LineItemSum in descending order. The first row is 'Book Printing Costs' with a LineItemCount of 8 and a LineItemSum of 148759.97. The second row is 'Freight' with a LineItemCount of 60 and a LineItemSum of 27599.65. The third row is 'Outside Services' with a LineItemCount of 3 and a LineItemSum of 13394.10. The fourth row is 'Book Production Costs' with a LineItemCount of 8 and a LineItemSum of 6175.12. The fifth row is 'Books, Dues, and Subscriptions' with a LineItemCount of 6 and a LineItemSum of 5207.32. The sixth row is 'Direct Mail Advertising' with a LineItemCount of 6 and a LineItemSum of 3900.77. The seventh row is 'Computer Equipment' with a LineItemCount of 3 and a LineItemSum of 2137.05. The eighth row is 'Building Lease' with a LineItemCount of 1 and a LineItemSum of 1750.00. The ninth row is 'UCI' with a LineItemCount of 1 and a LineItemSum of 1600.00. The tenth row is 'Business Licenses and Taxes' with a LineItemCount of 1 and a LineItemSum of 856.92. The eleventh row is 'Group Insurance' with a LineItemCount of 3 and a LineItemSum of 564.00. The twelfth row is 'Travel and Accommodations' with a LineItemCount of 1 and a LineItemSum of 503.20. The thirteenth row is 'Building Maintenance' with a LineItemCount of 1 and a LineItemSum of 450.00. The fourteenth row is 'Other Equipment' with a LineItemCount of 1 and a LineItemSum of 356.48. The fifteenth row is 'Other' with a LineItemCount of 1 and a LineItemSum of 356.48.

The status bar at the bottom indicates that the query was executed successfully, returning 21 rows in 00:00:00. The status bar also shows the server name (local), the instance name (RTM), the database name (DESKTOP-U6AC2GB\elham), the user (AP), and the session ID (57).

Q3: Write a SELECT statement that returns three columns: VendorName, InvoiceCount and InvoiceAverage. InvoiceCount is the count of the number of invoices, and InvoiceAverage is the average of the InvoiceTotal of each vendor. Filter the result set to include only those rows with InvoiceCount more than 3. Group the result set by VendorName and sort the result set in ascending order of InvoiceCount.

```
A3: SELECT Vendorname, COUNT(InvoiceID) 'Invoice Count', AVG(Invoicetotal) 'Invoice Total'
FROM Vendors
JOIN Invoices ON Vendors.VendorID = Invoices.VendorID
GROUP BY VendorName
HAVING COUNT(InvoiceID) > 3
ORDER BY COUNT(InvoiceID);
```

The screenshot displays the Microsoft SQL Server Management Studio interface. The central query editor shows the following SQL query:

```
SELECT Vendorname, COUNT(InvoiceID) 'Invoice Count', AVG(Invoicetotal) 'Invoice Total' FROM Vendors
JOIN Invoices ON Vendors.VendorID = Invoices.VendorID
GROUP BY VendorName
HAVING COUNT(InvoiceID) > 3
ORDER BY COUNT(InvoiceID);
```

The query has been executed successfully, and the results are displayed in the 'Results' tab. The results table has three columns: Vendorname, Invoice Count, and Invoice Total. The data is as follows:

	Vendorname	Invoice Count	Invoice Total
1	Roadway Package System, Inc	4	10.9175
2	Malloy Lithographing Inc	5	23978.482
3	Pacific Bell	6	28.5016
4	Zylka Design	8	867.5312
5	United Parcel Service	9	2575.3288
6	Federal Express Corporation	47	93.1493

The status bar at the bottom indicates that the query was executed successfully, returning 6 rows in 00:00:00 seconds. The right-hand pane shows the 'Properties' window with connection details for the current session.

Q4: Write a SELECT statement that answers the following question: What is the sum of sales for each “Sale Year”? Use the WITH ROLLUP operator to include a row that gives the grand sum. Use SalesTotals table from Examples database.

A4: USE Examples

Go

```
SELECT COALESCE(salesyear, 'Total') SalesYear, SUM (SalesTotal) "Sum of sales" FROM  
SalesTotals  
GROUP BY SalesYear WITH ROLLUP;
```

The screenshot shows the Microsoft SQL Server Management Studio interface. The query editor contains the following SQL statement:

```
USE Examples  
Go  
SELECT COALESCE(salesyear, 'Total') SalesYear, SUM (SalesTotal) "Sum of sales" FROM SalesTotals  
GROUP BY SalesYear WITH ROLLUP
```

The Results pane displays the output of the query:

	SalesYear	Sum of sales
1	2019	3286197.85
2	2020	4109980.00
3	2021	2003659.02
4	Total	9399836.87

The Properties pane on the right shows connection details for the current session, including connection name, elapsed time, and state.

Q5: Write a SELECT statement that return the vendor name and the total number of accounts that apply to that vendor's invoices. Filter the result set to include only the vendor who is being paid more than once. Sort the result set in ascending order of VendorName. (HINT: Use Vendors table, Invoices table and InvoiceLineItems table of AP database).

A5: USE AP

GO

```
SELECT VendorName , COUNT(*) TotalAccount FROM Vendors AS v
JOIN Invoices AS i
ON v.VendorID = i.VendorID
JOIN InvoiceLineItems l
ON i.InvoiceID = l.InvoiceID
GROUP BY VendorName
HAVING COUNT(*) > 1
ORDER BY VendorName;
```

The screenshot displays the Microsoft SQL Server Management Studio interface. The central query editor shows the following SQL code:

```
USE AP
GO
SELECT VendorName , COUNT(*) TotalAccount FROM Vendors AS v
JOIN Invoices AS i
ON v.VendorID = i.VendorID
JOIN InvoiceLineItems l
ON i.InvoiceID = l.InvoiceID
GROUP BY VendorName
HAVING COUNT(*) > 1
ORDER BY VendorName;
```

The Results pane at the bottom shows the output of the query, which is a table with two columns: VendorName and TotalAccount. The data is sorted by VendorName in ascending order.

VendorName	TotalAccount
Blue Cross	3
Cardinal Business Media, Inc.	2
Compuserve	2
Data Reproductions Corp	2
Federal Express Corporation	47
IBM	2
Ingram	2
Malloy Lithographing Inc	5
Pacific Bell	6
Roadway Package System, Inc	4
United Parcel Service	9
Wells Fargo Bank	4
Zyka Design	9

The Properties pane on the right shows the current connection parameters for the (local) (DESKTOP-U6AC2GB\elham) connection. The connection is successful, and the session version is 15.0.2080.

Q6: Write a SELECT statement that returns the distinct VendorName (i.e. VendorName should not be repeated in the result). Filter the result set to include only those vendors with invoices having a PaymentTotal that is greater than the average PaymentTotal for all invoices. Sort the result set in ascending order of VendorName.

```
A6: SELECT DISTINCT Vendorname FROM Vendors v
JOIN Invoices i ON v.VendorID = i.VendorID
GROUP BY VendorName
HAVING SUM (InvoiceTotal) > (SELECT MAX(AVG( InvoiceTotal)) OVER() FROM Invoices)
ORDER BY VendorName;
```

The screenshot displays the Microsoft SQL Server Management Studio (SSMS) interface. The main window shows a query window with the following SQL code:

```
SELECT DISTINCT Vendorname FROM Vendors v
JOIN Invoices i ON v.VendorID = i.VendorID
GROUP BY VendorName
HAVING SUM (InvoiceTotal) > (SELECT MAX(AVG( InvoiceTotal)) OVER() FROM Invoices)
ORDER BY VendorName
```

The query has been executed successfully, and the results are displayed in the Results pane. The results show a list of vendor names, sorted in ascending order. The first few vendors are:

Vendorname
Bertelsmann Industry Svcs. Inc.
Cahners Publishing Company
Computerworld
Data Reproductions Corp
Digital Dreamworks
Federal Express Corporation
Ingram
Malloy Lithographing Inc
United Parcel Service
Yarned, Inc
Zylka Design

The Properties pane on the right shows the current connection parameters, including the connection name, display name, login name, server name, server version, and session tracing ID.

Q7: Write a SELECT statement that returns the sum of the largest unpaid invoices submitted by each vendor. Use a derived table that returns MAX(InvoiceTotal) grouped by VendorID, filtering for invoices with a balance due. (HINT: Balance = InvoiceTotal – CreditTotal - PaymentTotal)

```
A7: SELECT VendorName ,  
      (  
        SELECT MAX(InvoiceTotal - CreditTotal - PaymentTotal) FROM PaidInvoices i  
        WHERE i.VendorID = v.VendorID  
      ) AS MaximumDebt  
FROM Vendors v  
ORDER BY MaximumDebt DESC;
```

The screenshot displays the Microsoft SQL Server Enterprise Manager interface. The central pane shows a query window with the following SQL code:

```
SELECT VendorName ,  
      (  
        SELECT MAX(InvoiceTotal - CreditTotal - PaymentTotal) FROM PaidInvoices i  
        WHERE i.VendorID = v.VendorID  
      ) AS MaximumDebt  
FROM Vendors v  
ORDER BY MaximumDebt DESC
```

The Results pane at the bottom shows the output of the query, which is a table with two columns: VendorName and MaximumDebt. The data is sorted by MaximumDebt in descending order. The first row is 'Fireano County Tax Collector' with a value of 0.00. The second row is 'Blue Cross' with a value of 0.00. The third row is 'Data Reproductions Corp' with a value of 0.00. The fourth row is 'Cardinal Business Media, Inc.' with a value of 0.00. The fifth row is 'Wang Laboratories, Inc.' with a value of 0.00. The sixth row is 'Petter's Scientific & Pro Books' with a value of 0.00. The seventh row is 'Ingram' with a value of 0.00. The eighth row is 'IBM' with a value of 0.00. The ninth row is 'Computerworld' with a value of 0.00. The tenth row is 'Edward Data Services' with a value of 0.00. The eleventh row is 'Evans Executone Inc.' with a value of 0.00. The twelfth row is 'Wakefield Co.' with a value of 0.00. The thirteenth row is 'Abbey Office Furnishings' with a value of 0.00. The fourteenth row is 'Pacific Bell' with a value of 0.00. The fifteenth row is 'Wash. Comm. Book' with a value of 0.00.

The Properties pane on the right shows the current connection parameters, including the connection name (local), the server name (DESKTOP-U6AC2GB\elham), and the login name (DESKTOP-U6AC2GB\elham).

Q8: Write a SELECT statement that returns the sum of the largest unpaid invoices submitted by each vendor. Use a derived table that returns MAX(InvoiceTotal) grouped by VendorID, filtering for invoices with a balance due. (HINT: Balance = InvoiceTotal – CreditTotal - PaymentTotal)

```
A8: Select * from vendors v
Join (
  Select vendorstate,vendorcity from vendors
Group by vendorstate,vendorcity
Having count(1) = 1
) vv on vv.vendorstate = v.vendorstate and vv.vendorcity = v.vendorcity
```

Query executed successfully. (local) (15.0 RTM) | DESKTOP-U6AC2GB\elham ... | Examples | 00:00:00 | 41 rows

VendorID	VendorName	VendorAddress1	VendorAddress2	VendorCity	VendorState	VendorZipCode	VendorContact
14	Bertelsmann Industry Svcs. Inc	28210 N Avenue Stanford		Valencia	CA	91355	Potter
15	Ascom Hasler Mailing Systems	PO Box 895		Shelton	CT	06484	Lewis
16	Naylor Publications Inc	PO Box 40513		Jacksonville	FL	32231	Gerald
17	Open Horizons Publishing	Book Marketing Update	PO Box 205	Fairfield	IA	52556	Damen
18	The Drawing Board	PO Box 4758		Carol Stream	IL	60197	Mckayla
19	Leslie Company	PO Box 610		Olathe	KS	66061	Alondra
20	Malloy Lithographing Inc	5411 Jackson Road	PO Box 1124	Ann Arbor	MI	48106	Regging
21	Data Reproductions Corp	4545 Glenmeade Lane		Albion Hills	MI	48326	Armando
22	Small Press	121 E Front St - 4th Floor		Traverse City	MI	49684	Colette
23	Capital Resource Credit	PO Box 39046		Minneapolis	MN	55439	Maxwell
24	Baker & Taylor Books	Five Lakepointe Plaza, Ste 5...	2709 Water Ridge Pa...	Charlotte	NC	28217	Bernardo
25	Newbridge Book Clubs	3000 Cordell Drive		Delran	NJ	08370	Maria
26	Simon Direct Inc	4 Cornwall Dr Ste 102		East Brunswick	NJ	08816	Bradlee
27	Rink Advertising	17 Daniel Road		Fairfield	NI	07004	Nad

Thank you for your time.
Sincerely,
Seyed Alireza Zarrin Mehr