

# Chapter 6

## How to code summary queries

### Objectives

1. Write a SELECT statement that returns one row for each vendor in the Invoices table that contains these columns :  
The vendor\_id column from the Vendors table  
The sum of the invoice\_total columns in the Invoices table for that vendor  
This should return 34 rows.

### Answer:

**SELECT vendor\_id, SUM(invoice\_total) AS invoice\_total\_sum**

**FROM invoices**

**GROUP BY vendor\_id**

The screenshot shows the SQL Enterprise Manager interface. The left pane displays the 'Schemas' tree with 'alexamara' expanded, showing 'Tables' and 'Views'. The 'Tables' list includes 'companies', 'customers', 'general\_ledger', 'invoice\_archive', 'invoice\_line\_item', 'invoices', 'sales', 'student\_table', 'terms', 'vendor\_contact', and 'vendors'. The 'Table: vendors' is selected, showing columns: vendor\_id, vendor\_name, vendor\_address1, vendor\_address2, vendor\_city, vendor\_state, vendor\_zip\_code, vendor\_phone, vendor\_contact\_last\_name, vendor\_contact\_first\_name, and default\_terms\_id. The main pane shows a query window with the following SQL code:

```
1 use ap;  
2 SELECT vendor_id, SUM(invoice_total) AS invoice_total_sum  
3 FROM invoices  
4 GROUP BY vendor_id
```

The 'Result Grid' shows the following data:

vendor_id	invoice_total_sum
34	1200.12
37	564.00
48	856.92
72	21927.31
80	265.36
81	936.93
82	600.00
83	2154.42
86	2433.00
88	207.78

The 'Result 4' tab shows the 'Action Output' with the following messages:

#	Time	Action	Message
9	20:55:03	select vendor_id from vendors LIMIT 0, 1000	122 row(s) returned
10	20:56:24	select * from invoices LIMIT 0, 1000	114 row(s) returned
11	20:57:15	select vendor_id, count(invoice_total) from vendors Join Invoices LIMIT 0, 1000	Error Code: 1052. Column 'vendor_id' in field list is ambiguous
12	20:57:37	select vendors.vendor_id, count(invoices.invoice_total) from vendors Join Invoices LIMIT 0, 1000	Error Code: 1140. In aggregated query without GROUP BY, expression #1
13	20:57:40	select vendors.vendor_id, count(invoices.invoice_total) from vendors Join Invoices LIMIT 0, 1000	Error Code: 1140. In aggregated query without GROUP BY, expression #1
14	20:59:13	SELECT vendor_id, SUM(invoice_total) AS invoice_total_sum FROM invoices GROUP BY vendor_id LIMIT 0, 1000	34 row(s) returned

2. Write a SELECT statement that returns one row for each vendor that contains these columns:

The vendor\_name column from the Vendors table

The sum of the payment\_total columns in the Invoices table for that vendor

Sort the result set in descending sequence by the payment total sum for each vendor.

**Answer:**

**SELECT vendor\_name, SUM(payment\_total) AS payment\_total\_sum**

**FROM vendors v JOIN invoices i**

**ON v.vendor\_id = i.vendor\_id**

**GROUP BY vendor\_name**

**ORDER BY payment\_total\_sum DESC**

The screenshot shows the Navicat SQL Editor interface. The SQL File 12\* window contains the following query:

```
1 use ap;
2 SELECT vendor_name, SUM(payment_total) AS payment_total_sum
3 FROM vendors v JOIN invoices i
4 ON v.vendor_id = i.vendor_id
5 GROUP BY vendor_name
6 ORDER BY payment_total_sum DESC
```

The Result Grid shows the following data:

vendor_name	payment_total_sum
Malloy Lithographing Inc	86069.22
United Parcel Service	23177.96
Data Reproductions Corp	21842.00
Digital Dreamworks	7125.34
Bertelsmann Industry Svcs. Inc	6940.25
Zylka Design	6740.25
Yesmed, Inc	4901.26
Federal Express Corporation	4167.13
Computerworld	2433.00
Cahners Publishing Company	2184.50

The Output window shows the execution log:

#	Time	Action	Message
25	21:07:08	use ap;	0 row(s) affected
26	21:07:08	SELECT vendor_name, SUM(payment_total) FROM invoices GROUP BY vendors LIMIT 0, 1000	Error Code: 1054. Unknown column 'vendor_name' in field list' 0 row(s) affected
27	21:07:39	use ap;	0 row(s) affected
28	21:07:39	SELECT vendors.vendor_name, SUM(payment_total) FROM invoices GROUP BY vendors LIMIT 0, 1000	Error Code: 1054. Unknown column 'vendors.vendor_name' in field list' 0 row(s) affected
29	21:10:51	use ap;	0 row(s) affected
30	21:10:51	SELECT vendor_name, SUM(payment_total) AS payment_total_sum FROM vendors v JOIN invoices i ON v.vendor_id = i.vendor_id GROUP BY vendor_name ORDER BY payment_total_sum DESC	34 row(s) returned

3. Write a SELECT statement that returns one row for each vendor that contains three columns:

The vendor\_name column from the Vendors table

The count of the invoices in the Invoices table for each vendor

The sum of the invoice\_total columns in the Invoices table for each vendor

Sort the result set so the vendor with the most invoices appears first.

**Answer:**

```
SELECT vendor_name, COUNT(*) AS invoice_count,  
       SUM(invoice_total) AS invoice_total_sum  
FROM vendors v JOIN invoices i  
  ON v.vendor_id = i.vendor_id  
GROUP BY vendor_name  
ORDER BY invoice_count DESC
```

The screenshot displays the Navicat SQL Enterprise Edition interface. The left sidebar shows the 'SCHEMAS' tree with 'alexamara' selected, containing tables like 'companies', 'customers', 'general\_ledger', 'invoice\_archive', 'invoice\_line\_item', 'invoices', 'sales', 'student\_table', 'terms', 'vendor\_contact', and 'vendors'. The 'vendors' table is selected, showing its columns: 'vendor\_id', 'vendor\_name', 'vendor\_address1', 'vendor\_address2', 'vendor\_city', 'vendor\_state', 'vendor\_zip\_code', 'vendor\_phone', 'vendor\_contact\_last\_name', 'vendor\_contact\_first\_name', and 'default\_terms\_id'.

The main window shows a SQL query in the 'SQL File 12\*' editor:

```
1 use ap;
2 SELECT vendor_name, COUNT(*) AS invoice_count,
3        SUM(invoice_total) AS invoice_total_sum
4 FROM vendors v JOIN invoices i
5      ON v.vendor_id = i.vendor_id
6 GROUP BY vendor_name
7 ORDER BY invoice_count DESC
```

The 'Result Grid' shows the following data:

vendor_name	invoice_count	invoice_total_sum
Federal Express Corporation	47	4378.02
United Parcel Service	9	23177.96
Zylka Design	8	6940.25
Pacific Bell	6	171.01
Malloy Lithographing Inc	5	119892.41
Roadway Package System, Inc	4	43.67
Blue Cross	3	564.00
IBM	2	1200.12
Cardinal Business Media, Inc.	2	265.36
Data Reproductions Corp	2	21927.31

The 'Output' window shows the execution log:

#	Time	Action	Message
27	21:07:39	use ap	0 row(s) affected
28	21:07:39	SELECT vendors.vendor_name, SUM(payment_total) FROM invoices GROUP BY vendors LIMIT 0, 1000	Error Code: 1054. Unknown column 'vendors.vendor_name' in field list
29	21:10:51	use ap	0 row(s) affected
30	21:10:51	SELECT vendor_name, SUM(payment_total) AS payment_total_sum FROM vendors v JOIN invoices i ON v.vendor_id = i.vendor_id	34 row(s) returned
31	21:13:37	use ap	0 row(s) affected
32	21:13:37	SELECT vendor_name, COUNT(*) AS invoice_count, SUM(invoice_total) AS invoice_total_sum FROM vendors v	34 row(s) returned

4. Write a SELECT statement that returns one row for each general ledger account number that contains three columns:
- The account\_description column from the General\_Ledger\_Accounts table
  - The count of the items in the Invoice\_Line\_Items table that have the same account\_number
  - The sum of the line\_item\_amount columns in the Invoice\_Line\_Items table that have the same account\_number
- Return only those rows where the count of line items is greater than 1. This should return 10 rows.
- 

*Chapter 6    How to code summary*

Group the result set by account description.

Sort the result set in descending sequence by the sum of the line item amounts.

**Answer: SELECT account\_description, COUNT(\*) AS line\_item\_count,  
SUM(line\_item\_amount) AS line\_item\_amount\_sum  
FROM general\_ledger\_accounts gl  
JOIN invoice\_line\_items li  
ON gl.account\_number = li.account\_number  
GROUP BY account\_description  
HAVING line\_item\_count > 1  
ORDER BY line\_item\_amount\_sum DESC**

The screenshot shows the Microsoft SQL Server Enterprise Manager interface. The left pane displays the 'SCHEMAS' tree with 'alexamara' expanded, showing 'ap' and 'ex' schemas. The 'ap' schema contains tables like 'companies', 'customers', 'general\_ledger\_accounts', 'invoice\_archive', 'invoice\_line\_items', 'invoices', 'sales', 'student\_table', 'terms', 'vendor\_contact', and 'vendors'. The 'vendors' table is selected, and its columns are listed: 'vendor\_id', 'vendor\_name', 'vendor\_address1', 'vendor\_address2', 'vendor\_city', 'vendor\_state', 'vendor\_zip\_code', 'vendor\_phone', 'vendor\_contact\_last\_name', 'vendor\_contact\_first\_name', and 'default\_terms\_id'.

The central pane shows a SQL query in 'SQL File 12\*':

```

1 use ap;
2 SELECT account_description, COUNT(*) AS line_item_count,
3        SUM(line_item_amount) AS line_item_amount_sum
4 FROM general_ledger_accounts gl
5      JOIN invoice_line_items li
6      ON gl.account_number = li.account_number
7 GROUP BY account_description
8 HAVING line_item_count > 1
9 ORDER BY line_item_amount_sum DESC

```

The 'Result Grid' shows the following data:

account_description	line_item_count	line_item_amount_sum
Book Printing Costs	8	148759.97
Freight	60	27599.65
Outside Services	3	13394.10
Book Production Costs	8	6175.12
Books, Dues, and Subscriptions	6	5207.32
Direct Mail Advertising	6	3900.77
Computer Equipment	3	2137.05
Computer Equipment	3	2137.05

The 'Output' pane shows the execution log:

#	Time	Action	Message
29	21:10:51	use ap	0 row(s) affected
30	21:10:51	SELECT vendor_name, SUM(payment_total) AS payment_total_sum FROM vendors v JOIN invoices i ON v.vendor_id = i.vendor_id	34 row(s) returned
31	21:13:37	use ap	0 row(s) affected
32	21:13:37	SELECT vendor_name, COUNT(*) AS invoice_count, SUM(invoice_total) AS invoice_total_sum FROM vendors v JOIN invoices i ON v.vendor_id = i.vendor_id	34 row(s) returned
33	21:17:48	use ap	0 row(s) affected
34	21:17:49	SELECT account_description, COUNT(*) AS line_item_count, SUM(line_item_amount) AS line_item_amount_sum FROM general_ledger_accounts gl JOIN invoice_line_items li ON gl.account_number = li.account_number GROUP BY account_description HAVING line_item_count > 1 ORDER BY line_item_amount_sum DESC	10 row(s) returned

5. Modify the solution to exercise 4 so it returns only invoices dated in the second quarter of 2014 (April 1, 2014 to June 30, 2014). This should still return 10 rows but with some different line item counts for each vendor. *Hint: Join to the Invoices table to code a search condition based on invoice\_date.*

**Answer:**

```

SELECT account_description, COUNT(*) AS line_item_count,
       SUM(line_item_amount) AS line_item_amount_sum
FROM general_ledger_accounts gl
JOIN invoice_line_items li

```

**ON gl.account\_number = li.account\_number**

**JOIN invoices i**

**ON li.invoice\_id = i.invoice\_id**

**WHERE invoice\_date BETWEEN '2014-04-01' AND '2014-06-30'**

**GROUP BY account\_description**

**HAVING line\_item\_count > 1**

**ORDER BY line\_item\_amount\_sum DESC**

The screenshot shows a database management tool interface with a SQL query editor and a results grid. The query is as follows:

```

1 use ap;
2 SELECT account_description, COUNT(*) AS line_item_count,
3       SUM(line_item_amount) AS line_item_amount_sum
4 FROM general_ledger_accounts gl
5 JOIN invoice_line_items li
6   ON gl.account_number = li.account_number
7 JOIN invoices i
8   ON li.invoice_id = i.invoice_id
9 WHERE invoice_date BETWEEN '2014-04-01' AND '2014-06-30'
10 GROUP BY account_description
11 HAVING line_item_count > 1
12 ORDER BY line_item_amount_sum DESC
13

```

The results grid shows the following data:

account_description	line_item_count	line_item_amount_sum
Book Printing Costs	3	66748.44
Freight	41	17624.19
Outside Services	3	13394.10
Book Production Costs	7	5174.66
Books, Dues, and Subscriptions	4	4027.90

The tool also displays a table structure for 'vendors' and an output log showing the execution of the query and the number of rows returned.





7. Write a SELECT statement that answers this question: Which vendors are being paid from more than one account? Return these columns:

The vendor name from the Vendors table

The count of distinct general ledger accounts that apply to that vendor's invoices

This should return 2 rows.

**Answer:**

```
SELECT vendor_name,  
       COUNT(DISTINCT li.account_number) AS number_of_gl_accounts  
FROM vendors v  
JOIN invoices i  
  ON v.vendor_id = i.vendor_id  
JOIN invoice_line_items li  
  ON i.invoice_id = li.invoice_id  
GROUP BY vendor_name  
HAVING number_of_gl_accounts > 1  
ORDER BY vendor_name
```

MySQL Workbench

navneet x

File Edit View Query Database Server Tools Scripting Help

Navigator SQL File 12" vendors invoices ap.vendors ap.vendors Limit to 1000 rows

**SCHEMAS**

Filter objects

alexamara

ap

Tables

- companies
- customers
- general\_ledger
- invoice\_archive
- invoice\_line\_items
- invoices
- sales
- student\_table
- terms
- vendor\_contact
- vendors

Views

Stored Procedures

Functions

ex

Tables

Administration Schemas

Information

**Table: vendors**

**Columns:**

- vendor\_id
- vendor\_name
- vendor\_address1
- vendor\_address2
- vendor\_city
- vendor\_state
- vendor\_zip\_code
- vendor\_phone
- vendor\_contact\_last\_i
- vendor\_contact\_first\_i
- default\_terms\_id

```
1 use ap;
2 SELECT vendor_name,
3        COUNT(DISTINCT li.account_number) AS number_of_gl_accounts
4 FROM vendors v
5 JOIN invoices i
6     ON v.vendor_id = i.vendor_id
7 JOIN invoice_line_items li
8     ON i.invoice_id = li.invoice_id
9 GROUP BY vendor_name
10 HAVING number_of_gl_accounts > 1
11 ORDER BY vendor_name
12
13
```

**Result Grid**

vendor_name	number_of_gl_accounts
Wells Fargo Bank	3
Zylka Design	2

**Result 10**

Output

Action Output

#	Time	Action	Message
35	21:20:15	use ap	0 row(s) affected
36	21:20:15	SELECT account_description, COUNT(*) AS line_item_count, SUM(line_item_amount) AS line_item_amount...	10 row(s) returned
37	21:22:20	use ap	0 row(s) affected
38	21:22:20	SELECT account_number, SUM(line_item_amount) AS line_item_sum FROM invoice_line_items GROUP BY acc...	22 row(s) returned
39	21:23:59	use ap	0 row(s) affected
40	21:23:59	SELECT vendor_name, COUNT(DISTINCT li.account_number) AS number_of_gl_accounts FROM vendors ...	2 row(s) returned