Chapter 6

How to code summary queries

Objectives

Applied

- Code summary queries that use aggregate functions, including queries that use the WITH ROLLUP operator.

Knowledge

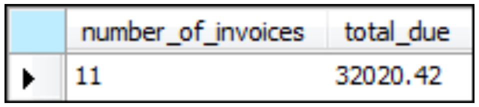
- Describe summary queries.
- Describe the differences between the HAVING clause and the WHERE clause.
- Describe the use of the WITH ROLLUP operator.

The syntax of the aggregate functions

```
AVG([ALL|DISTINCT] expression)
SUM([ALL|DISTINCT] expression)
MIN([ALL|DISTINCT] expression)
MAX([ALL|DISTINCT] expression)
COUNT([ALL|DISTINCT] expression)
COUNT(*)
```

A summary query

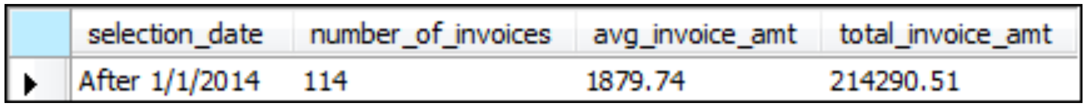
```
SELECT COUNT(*) AS number_of_invoices,
    SUM(invoice_total - payment_total - credit_total)
    AS total_due
FROM invoices
WHERE invoice_total - payment_total - credit_total > 0
```



	number_of_invoices	total_due
▶	11	32020.42

A summary query with COUNT(*), AVG, and SUM

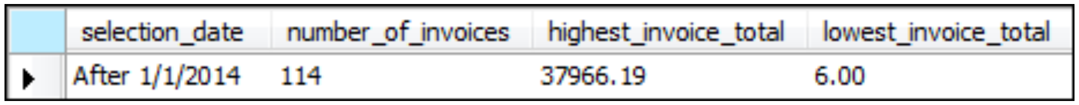
```
SELECT 'After 1/1/2014' AS selection_date,
    COUNT(*) AS number_of_invoices,
    ROUND(AVG(invoice_total), 2) AS avg_invoice_amt,
    SUM(invoice_total) AS total_invoice_amt
FROM invoices
WHERE invoice_date > '2014-01-01'
```



selection_date	number_of_invoices	avg_invoice_amt	total_invoice_amt
After 1/1/2014	114	1879.74	214290.51

A summary query with MIN and MAX

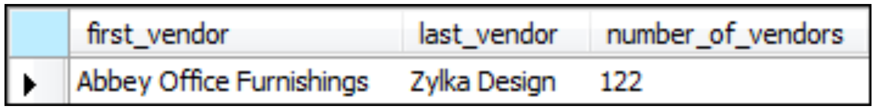
```
SELECT 'After 1/1/2014' AS selection_date,
    COUNT(*) AS number_of_invoices,
    MAX(invoice_total) AS highest_invoice_total,
    MIN(invoice_total) AS lowest_invoice_total
FROM invoices
WHERE invoice_date > '2014-01-01'
```



selection_date	number_of_invoices	highest_invoice_total	lowest_invoice_total
After 1/1/2014	114	37966.19	6.00

A summary query for non-numeric columns

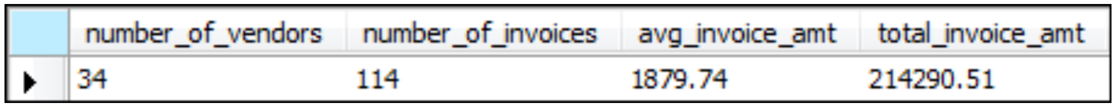
```
SELECT MIN(vendor_name) AS first_vendor,
    MAX(vendor_name) AS last_vendor,
    COUNT(vendor_name) AS number_of_vendors
FROM vendors
```



first_vendor	last_vendor	number_of_vendors
Abbey Office Furnishings	Zylka Design	122

A summary query with the DISTINCT keyword

```
SELECT COUNT(DISTINCT vendor_id) AS number_of_vendors,
    COUNT(vendor_id) AS number_of_invoices,
    ROUND(AVG(invoice_total), 2) AS avg_invoice_amt,
    SUM(invoice_total) AS total_invoice_amt
FROM invoices
WHERE invoice_date > '2014-01-01'
```



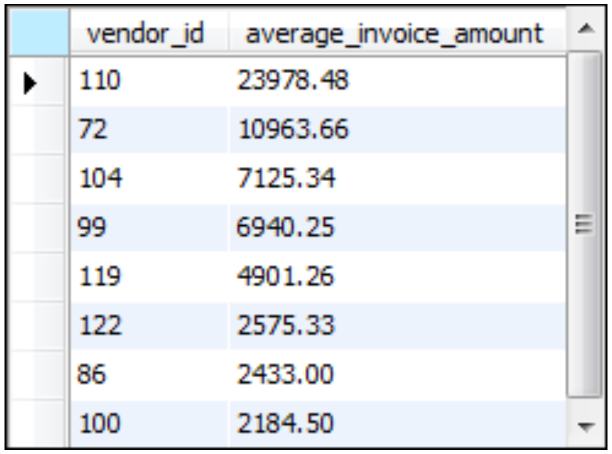
number_of_vendors	number_of_invoices	avg_invoice_amt	total_invoice_amt
34	114	1879.74	214290.51

The syntax of a SELECT statement with GROUP BY and HAVING clauses

```
SELECT select_list
FROM table_source
[WHERE search_condition]
[GROUP BY group_by_list]
[HAVING search_condition]
[ORDER BY order_by_list]
```

A summary query that calculates the average invoice amount by vendor

```
SELECT vendor_id, ROUND(AVG(invoice_total), 2)
    AS average_invoice_amount
FROM invoices
GROUP BY vendor_id
HAVING AVG(invoice_total) > 2000
ORDER BY average_invoice_amount DESC
```

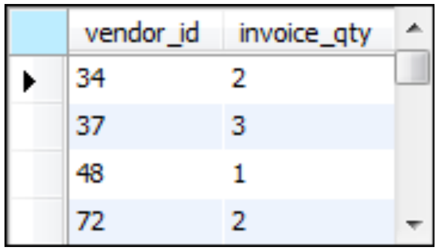


vendor_id	average_invoice_amount
110	23978.48
72	10963.66
104	7125.34
99	6940.25
119	4901.26
122	2575.33
86	2433.00
100	2184.50

```
(8 rows)
```

A summary query that counts the number of invoices by vendor

```
SELECT vendor_id, COUNT(*) AS invoice_qty
FROM invoices
GROUP BY vendor_id
```

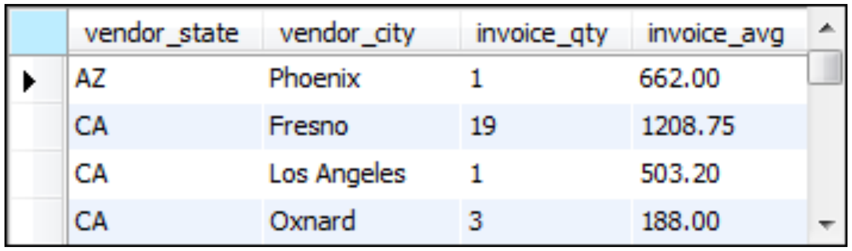


vendor_id	invoice_qty
34	2
37	3
48	1
72	2

```
(34 rows)
```

A summary query with a join

```
SELECT vendor_state, vendor_city, COUNT(*) AS invoice_qty,
    ROUND(AVG(invoice_total), 2) AS invoice_avg
FROM invoices JOIN vendors
    ON invoices.vendor_id = vendors.vendor_id
GROUP BY vendor_state, vendor_city
```

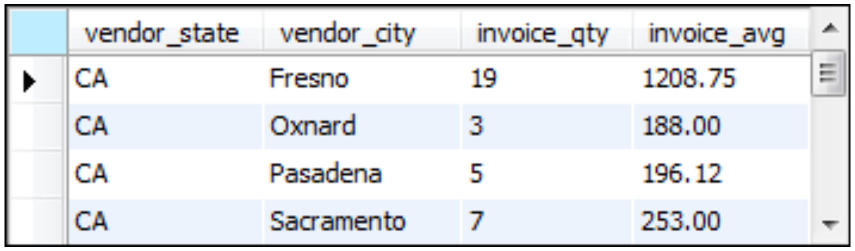


vendor_state	vendor_city	invoice_qty	invoice_avg
AZ	Phoenix	1	662.00
CA	Fresno	19	1208.75
CA	Los Angeles	1	503.20
CA	Oxnard	3	188.00

```
(20 rows)
```

A summary query that limits the groups to those with two or more invoices

```
SELECT vendor_state, vendor_city, COUNT(*) AS invoice_qty,
    ROUND(AVG(invoice_total), 2) AS invoice_avg
FROM invoices JOIN vendors
    ON invoices.vendor_id = vendors.vendor_id
GROUP BY vendor_state, vendor_city
HAVING COUNT(*) >= 2
```

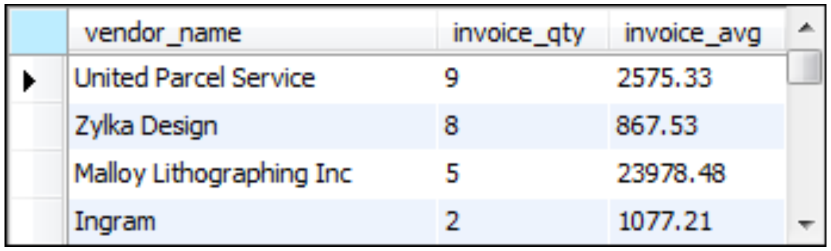


vendor_state	vendor_city	invoice_qty	invoice_avg
CA	Fresno	19	1208.75
CA	Oxnard	3	188.00
CA	Pasadena	5	196.12
CA	Sacramento	7	253.00

(12 rows)

A summary query with a search condition in the HAVING clause

```
SELECT vendor_name,
    COUNT(*) AS invoice_qty,
    ROUND(AVG(invoice_total),2) AS invoice_avg
FROM vendors JOIN invoices
    ON vendors.vendor_id = invoices.vendor_id
GROUP BY vendor_name
HAVING AVG(invoice_total) > 500
ORDER BY invoice_qty DESC
```

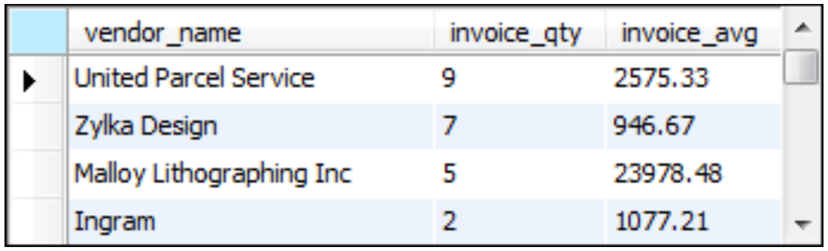


vendor_name	invoice_qty	invoice_avg
United Parcel Service	9	2575.33
Zylka Design	8	867.53
Malloy Lithographing Inc	5	23978.48
Ingram	2	1077.21

(19 rows)

A summary query with a search condition in the WHERE clause

```
SELECT vendor_name,
    COUNT(*) AS invoice_qty,
    ROUND(AVG(invoice_total),2) AS invoice_avg
FROM vendors JOIN invoices
    ON vendors.vendor_id = invoices.vendor_id
WHERE invoice_total > 500
GROUP BY vendor_name
ORDER BY invoice_qty DESC
```



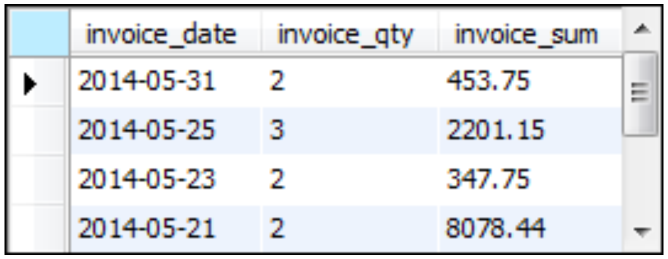
vendor_name	invoice_qty	invoice_avg
United Parcel Service	9	2575.33
Zylka Design	7	946.67
Malloy Lithographing Inc	5	23978.48
Ingram	2	1077.21

(20 rows)

A summary query with a compound condition in the HAVING clause

```
SELECT
    invoice_date,
    COUNT(*) AS invoice_qty,
    SUM(invoice_total) AS invoice_sum
FROM invoices
GROUP BY invoice_date
HAVING invoice_date BETWEEN '2014-05-01' AND '2014-05-31'
    AND COUNT(*) > 1
    AND SUM(invoice_total) > 100
ORDER BY invoice_date DESC
```

The result set



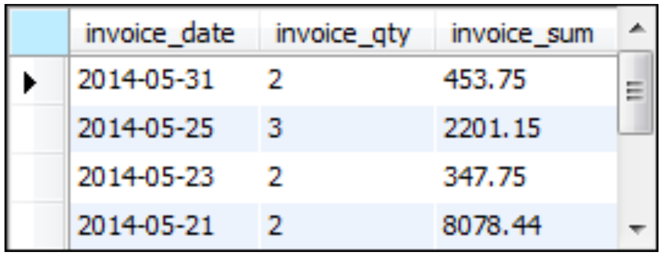
invoice_date	invoice_qty	invoice_sum
2014-05-31	2	453.75
2014-05-25	3	2201.15
2014-05-23	2	347.75
2014-05-21	2	8078.44

(7 rows)

The same query coded with a WHERE clause

```
SELECT
    invoice_date,
    COUNT(*) AS invoice_qty,
    SUM(invoice_total) AS invoice_sum
FROM invoices
WHERE invoice_date BETWEEN '2014-05-01' AND '2014-05-31'
GROUP BY invoice_date
HAVING COUNT(*) > 1
    AND SUM(invoice_total) > 100
ORDER BY invoice_date DESC
```

The same result set

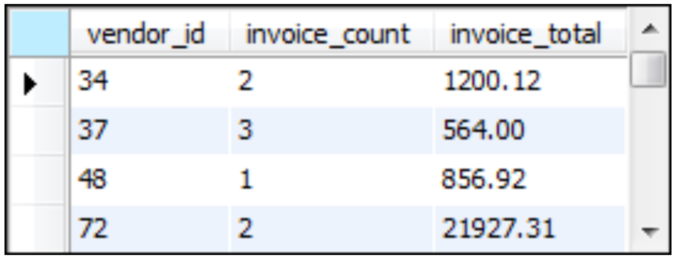


invoice_date	invoice_qty	invoice_sum
2014-05-31	2	453.75
2014-05-25	3	2201.15
2014-05-23	2	347.75
2014-05-21	2	8078.44

```
(7 rows)
```

A summary query with a final summary row

```
SELECT vendor_id, COUNT(*) AS invoice_count,
    SUM(invoice_total) AS invoice_total
FROM invoices
GROUP BY vendor_id WITH ROLLUP
```

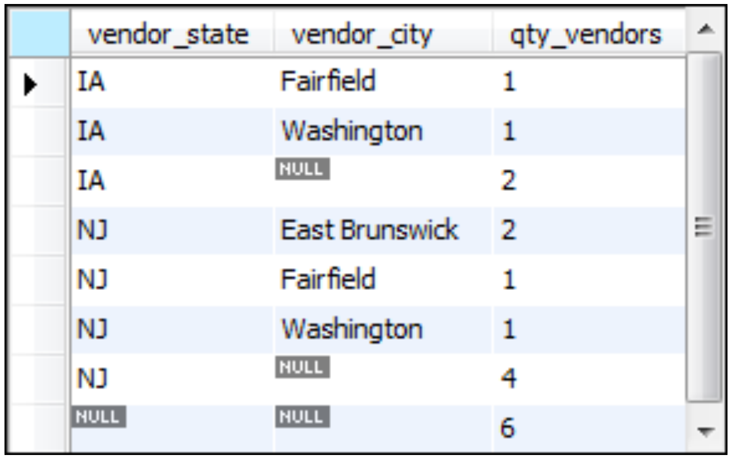


vendor_id	invoice_count	invoice_total
34	2	1200.12
37	3	564.00
48	1	856.92
72	2	21927.31

(35 rows)

A summary query with a summary row for each grouping level

```
SELECT vendor_state, vendor_city, COUNT(*) AS qty_vendors
FROM vendors
WHERE vendor_state IN ('IA', 'NJ')
GROUP BY vendor_state ASC, vendor_city ASC WITH ROLLUP
```



vendor_state	vendor_city	qty_vendors
IA	Fairfield	1
IA	Washington	1
IA	NULL	2
NJ	East Brunswick	2
NJ	Fairfield	1
NJ	Washington	1
NJ	NULL	4
NULL	NULL	6