Murach Chapter 5   Part 2

How to Code Summary Queries

Week 5, Part 2

Knowledge Points in this lecture

- GROUP BY
- HAVING
- WHERE clause and HAVING clause in summary query

The syntax with GROUP BY clauses

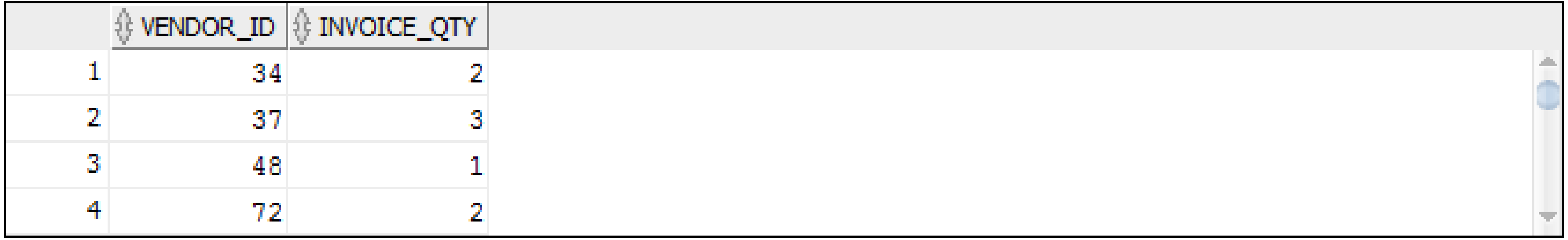
```
SELECT select_list
FROM table_source
[WHERE search_condition]
[GROUP BY group_by_list]
[ORDER BY order_by_list]
```

- **GROUP BY clause**
  - **Group rows in the result set based on 1 or more columns or expressions**
- **WHERE** conditions are applied to each individual row BEFORE grouping.

A summary query that counts the number of invoices by vendor

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

The result set



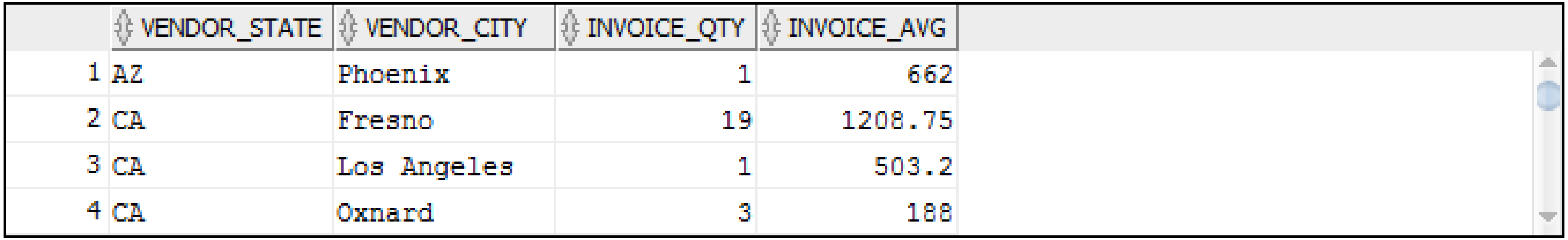
	VENDOR_ID	INVOICE_QTY
1	34	2
2	37	3
3	48	1
4	72	2

(34 rows selected)

A summary query with a join and GROUP BY

```
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
ORDER BY vendor_state, vendor_city
```

The result set



	VENDOR_STATE	VENDOR_CITY	INVOICE_QTY	INVOICE_AVG
1	AZ	Phoenix	1	662
2	CA	Fresno	19	1208.75
3	CA	Los Angeles	1	503.2
4	CA	Oxnard	3	188

(20 rows selected)

SELECT Clause and GROUP BY clause

- GROUP BY clause places a restriction on the list in SELECT clause.
- For a query containing GROUP BY clause, its SELECT clause can include the following items:
  - Aggregate functions
  - Expressions in GROUP BY clause or expressions involving GROUP BY columns
  - Constant values or expressions resulting in a constant value

The syntax with GROUP BY and HAVING clauses

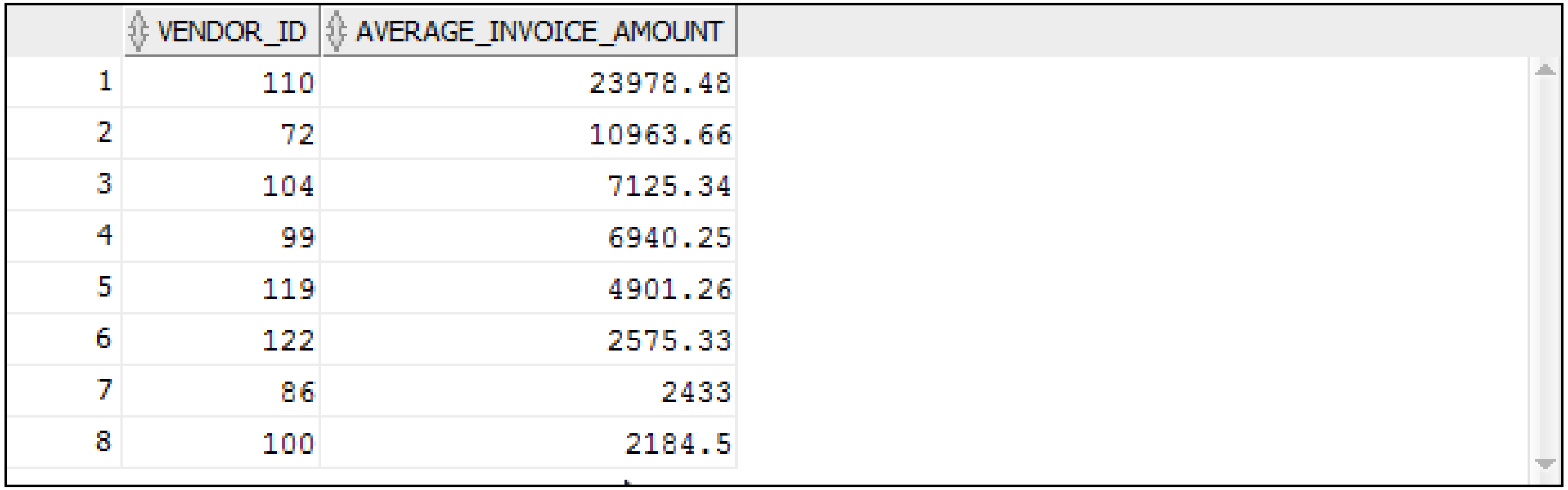
```
SELECT select_list
FROM table_source
[WHERE search_condition]
[GROUP BY group_by_list]
[HAVING search_condition]
[ORDER BY order_by_list]
```

- **HAVING clause**
  - **Specify condition(s) that a group or an aggregate must satisfy**
  - **Can only refer to a column in the base table that is included in the SELECT clause**
- **WHERE** conditions are applied to each individual row BEFORE grouping.
- **HAVING** conditions are applied to each group AFTER grouping.

A summary query that calculates 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
```

The result set



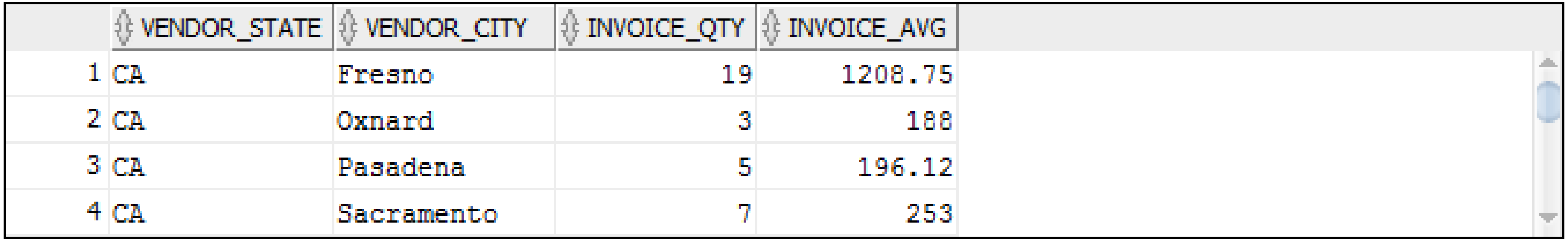
	VENDOR_ID	AVERAGE_INVOICE_AMOUNT
1	110	23978.48
2	72	10963.66
3	104	7125.34
4	99	6940.25
5	119	4901.26
6	122	2575.33
7	86	2433
8	100	2184.5

```
(8 rows selected)
```

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
ORDER BY vendor_state, vendor_city
```

The result set



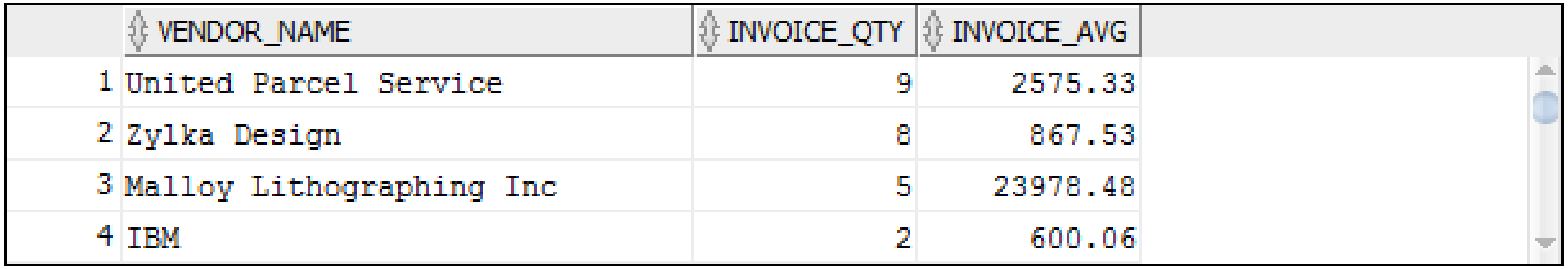
	VENDOR_STATE	VENDOR_CITY	INVOICE_QTY	INVOICE_AVG
1	CA	Fresno	19	1208.75
2	CA	Oxnard	3	188
3	CA	Pasadena	5	196.12
4	CA	Sacramento	7	253

(12 rows selected)

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
```

The result set



	VENDOR_NAME	INVOICE_QTY	INVOICE_AVG
1	United Parcel Service	9	2575.33
2	Zylka Design	8	867.53
3	Malloy Lithographing Inc	5	23978.48
4	IBM	2	600.06

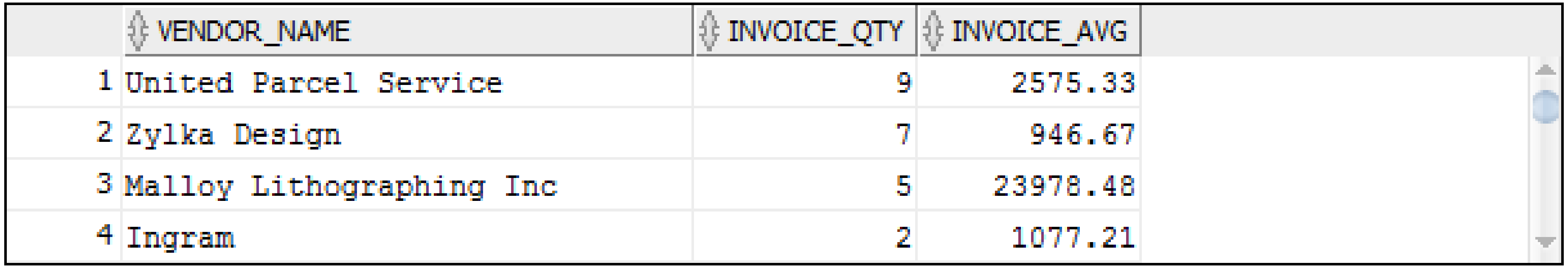
(19 rows selected)

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
```

WHERE clause canNOT include aggregate functions!

The result set



	VENDOR_NAME	INVOICE_QTY	INVOICE_AVG
1	United Parcel Service	9	2575.33
2	Zylka Design	7	946.67
3	Malloy Lithographing Inc	5	23978.48
4	Ingram	2	1077.21

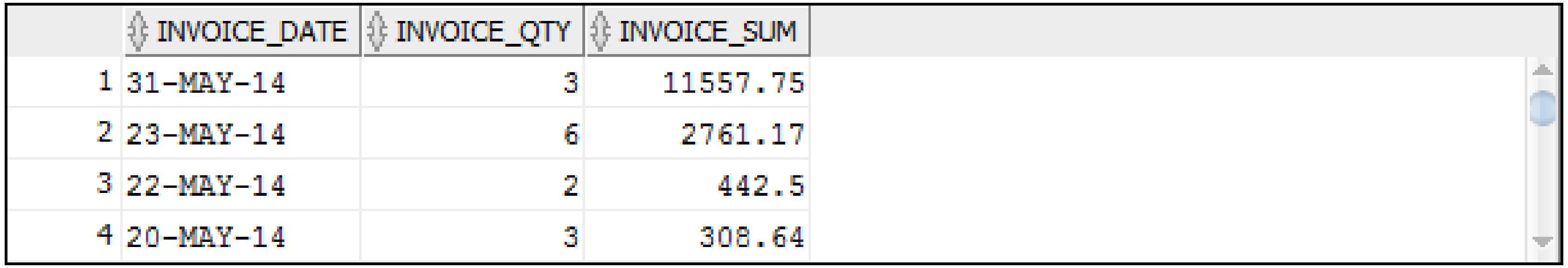
(20 rows selected)

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 '01-MAY-2014' AND '31-MAY-2014'
   AND COUNT(*) > 1
   AND SUM(invoice_total) > 100
ORDER BY invoice_date DESC
```

invoice_date is a column in GROUP BY clause.
So it can be in either HAVING or WHERE.

The result set



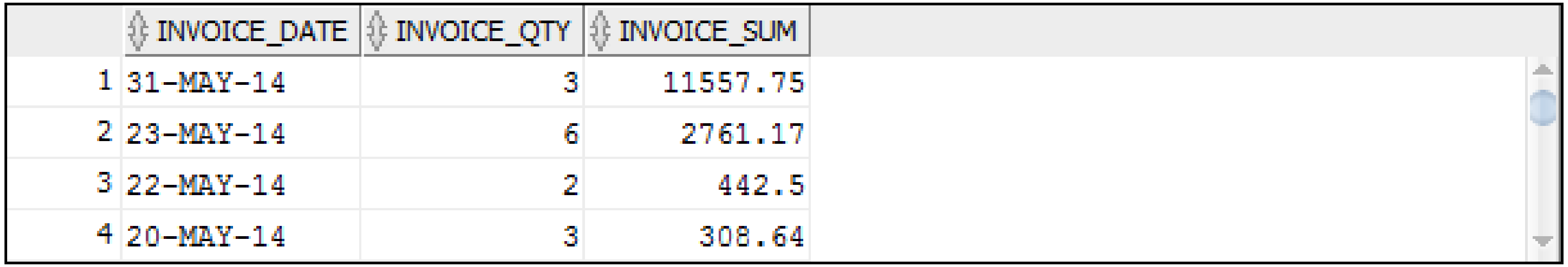
	INVOICE_DATE	INVOICE_QTY	INVOICE_SUM
1	31-MAY-14	3	11557.75
2	23-MAY-14	6	2761.17
3	22-MAY-14	2	442.5
4	20-MAY-14	3	308.64

(15 rows selected)

The same query with a WHERE clause

```
SELECT
   invoice_date,
   COUNT(*) AS invoice_qty,
   SUM(invoice_total) AS invoice_sum
FROM invoices
WHERE invoice_date
    BETWEEN '01-MAY-2014' AND '31-MAY-2014'
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
1	31-MAY-14	3	11557.75
2	23-MAY-14	6	2761.17
3	22-MAY-14	2	442.5
4	20-MAY-14	3	308.64

(15 rows selected)

HAVING Condition and WHERE Condition

	WHERE condition	HAVING condition
Purpose	Restrict individual rows included in result set (query result)	Restrict groups (sets of rows) included in result set (query result)
Include columns in base tables?	Yes, any column in base tables	Yes, but ONLY columns included in SELECT clause.
Include aggregate functions ?	NO	Yes
When to apply the condition	BEFORE individual rows are grouped based on GROUP BY list	AFTER individual rows are grouped based on GROUP BY list