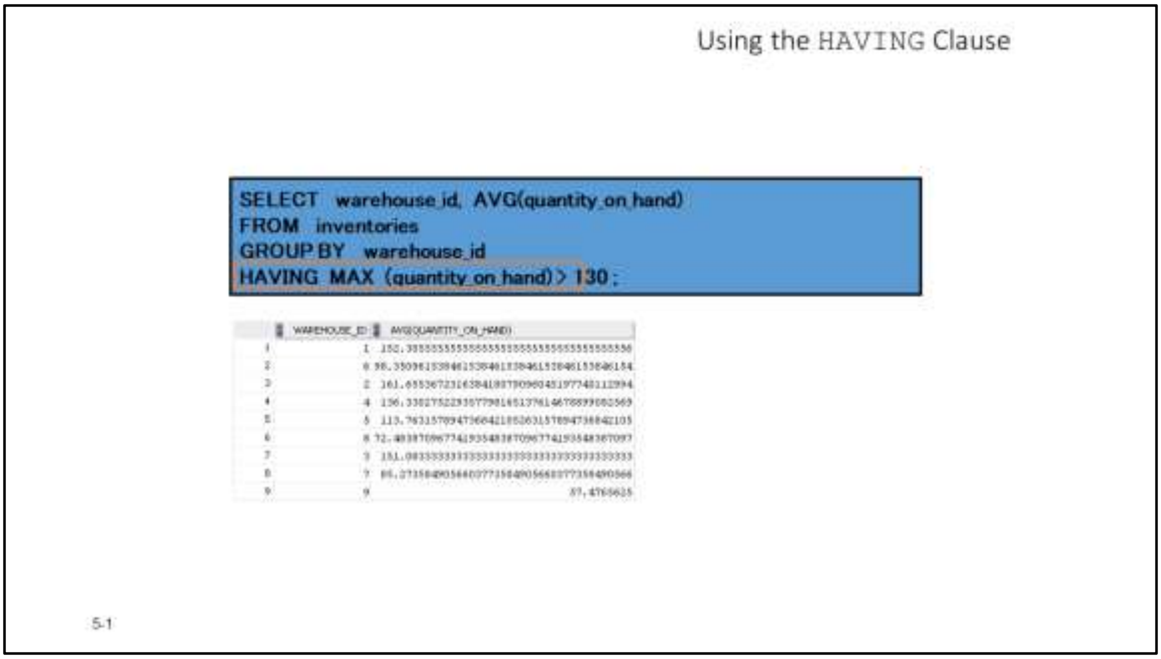


Using the HAVING Clause

```
SELECT   warehouse_id, AVG(quantity_on_hand)
FROM   inventories
GROUP BY   warehouse_id
HAVING  MAX (quantity_on_hand) > 130 ;
```

	WAREHOUSE_ID	AVG(QUANTITY_ON_HAND)
1	1	152.3555555555555555555555555555555556
2	6	98.35096153846153846153846153846153846154
3	2	161.655367231638418079096045197740112994
4	4	136.338275229357798165137614678899082569
5	5	113.763157894736842105263157894736842105
6	8	72.48387096774193548387096774193548387097
7	3	151.083333333333333333333333333333333333
8	7	85.2735849056603773584905660377358490566
9	9	57.4765625

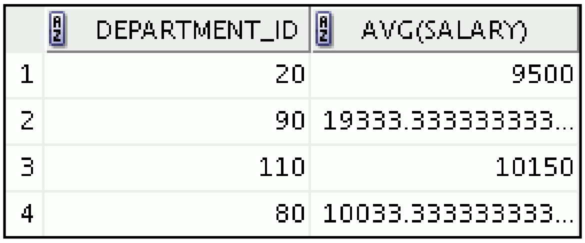
5-1

Using the HAVING Clause

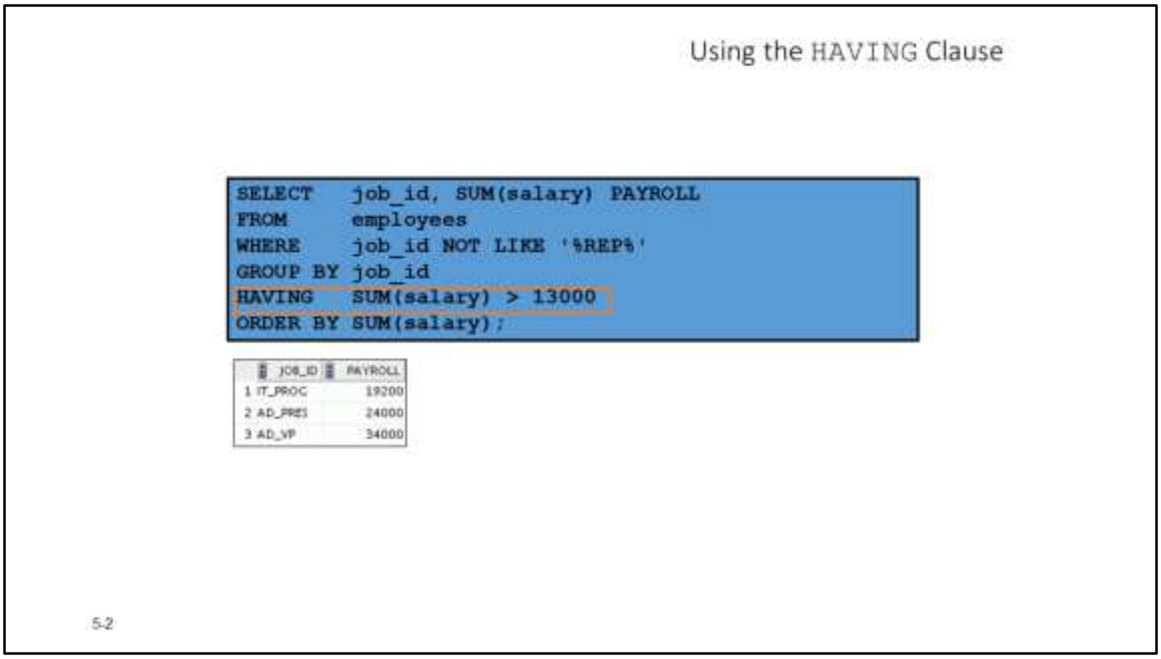
The example in the slide displays the warehouse ID and average quantity on hand for those warehouses with a maximum quantity greater than 130 130. You can use the GROUP BY clause without using a group function in the SELECT list. If you restrict rows based on the result of a group function, you must have a GROUP BY clause as well as the HAVING clause.

The following example displays the department numbers and average salaries for those departments with a maximum salary greater than $10,000:

```
SELECT   department_id, AVG(salary)
FROM     employees
GROUP BY department_id
HAVING   max(salary)>10000;
```



	DEPARTMENT_ID	AVG(SALARY)
1	20	9500
2	90	19333.333333333...
3	110	10150
4	80	10033.333333333...



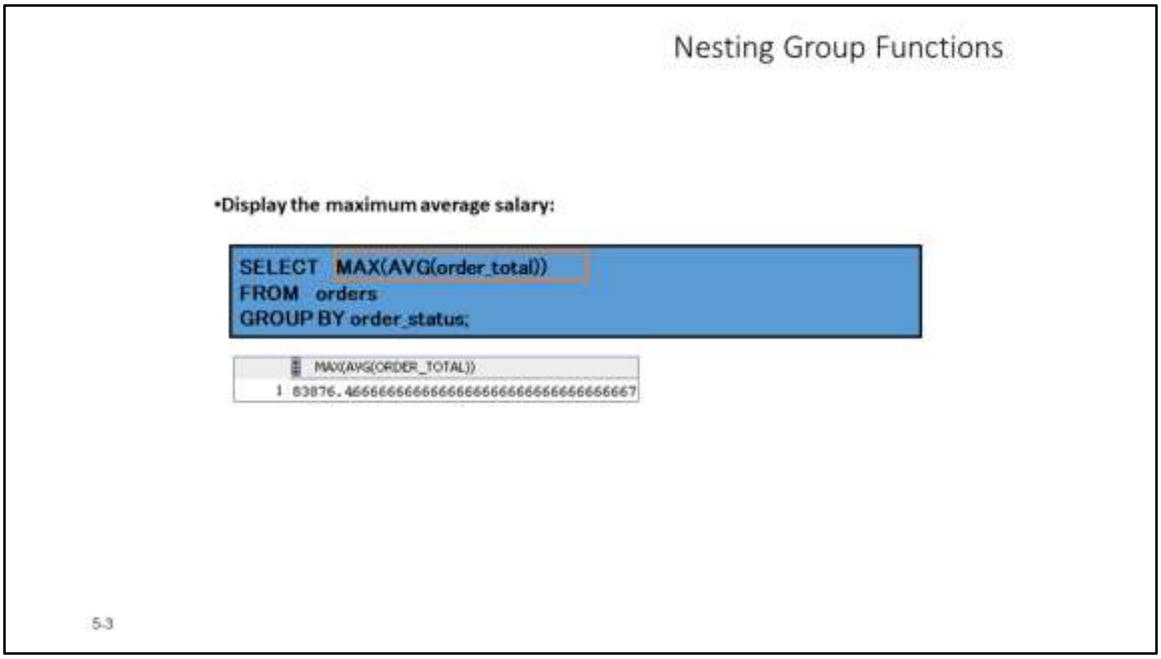
Using the HAVING Clause

```
SELECT   job_id, SUM(salary) PAYROLL
FROM     employees
WHERE    job_id NOT LIKE '%REP%'
GROUP BY job_id
HAVING   SUM(salary) > 13000
ORDER BY SUM(salary);
```

```
    JOB_ID     PAYROLL
1 IT_PROC        19200
2 AD_PRES        24000
3 AD_VP          34000
```

Using the HAVING Clause (continued)

The example in the slide displays the job ID and total monthly salary for each job that has a total payroll exceeding $13,000. The example excludes sales representatives and sorts the list by the total monthly salary.



Nesting Group Functions

- Display the maximum average salary:

```
SELECT   MAX(AVG(order_total))
FROM   orders
GROUP BY order_status;
```

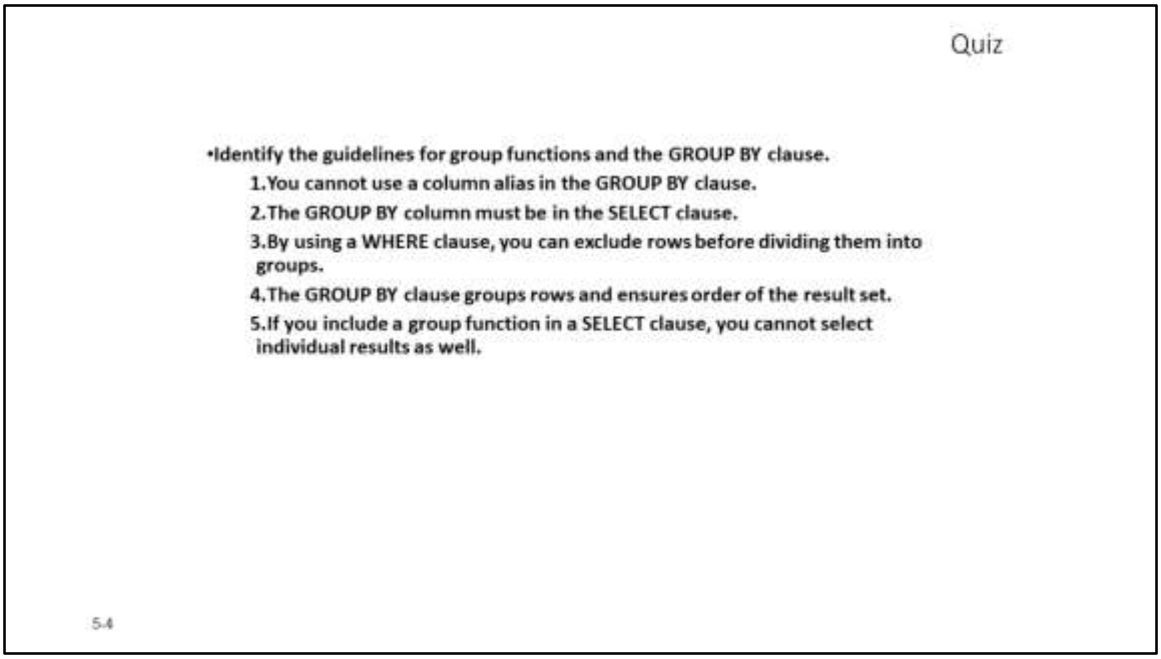
```
     MAX(AVG(ORDER_TOTAL))
   1 83876.4666666666666666666666666666666667
```

5-3

Nesting Group Functions

Group functions can be nested to a depth of two functions. The example in the slide calculates the average total order for each status and then displays the maximum average order toal.

Note that `GROUP BY` clause is mandatory when nesting group functions.

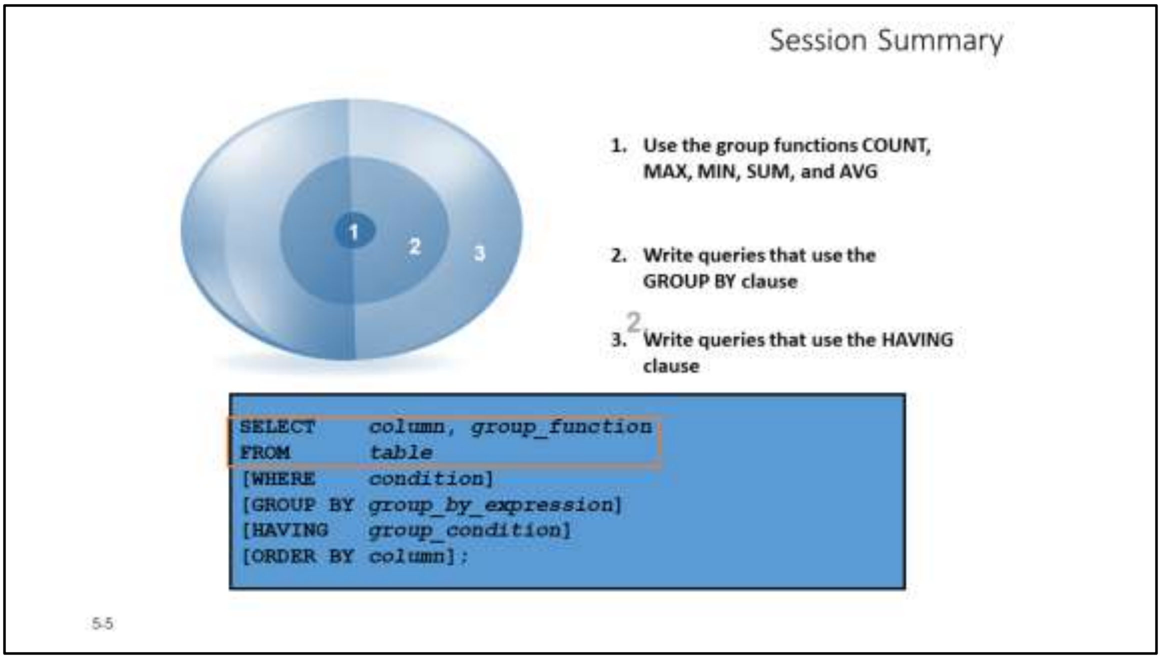


Quiz

- Identify the guidelines for group functions and the GROUP BY clause.
  1. You cannot use a column alias in the GROUP BY clause.
  2. The GROUP BY column must be in the SELECT clause.
  3. By using a WHERE clause, you can exclude rows before dividing them into groups.
  4. The GROUP BY clause groups rows and ensures order of the result set.
  5. If you include a group function in a SELECT clause, you cannot select individual results as well.

5-4

Answer: 1, 3



Session Summary

1. Use the group functions COUNT, MAX, MIN, SUM, and AVG
2. Write queries that use the GROUP BY clause
3. Write queries that use the HAVING clause

```
SELECT     column, group_function
FROM       table
[WHERE     condition]
[GROUP BY  group_by_expression]
[HAVING    group_condition]
[ORDER BY  column];
```

5-5

Summary

There are several group functions available in SQL, such as AVG, COUNT, MAX, MIN, SUM, STDDEV, and VARIANCE.

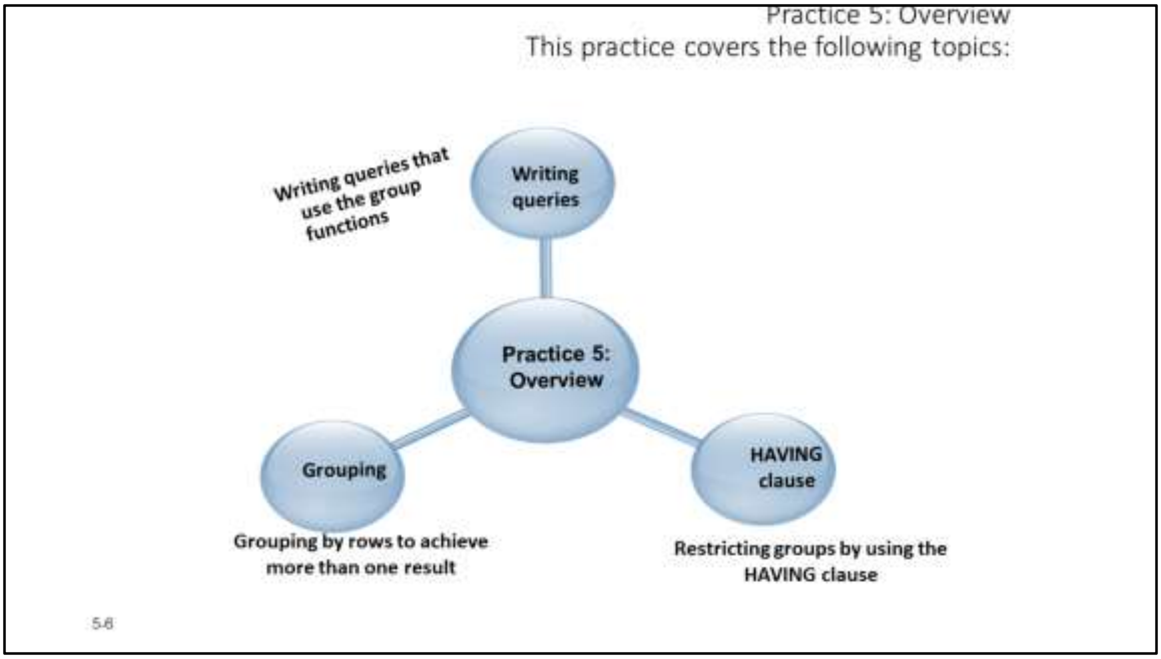
You can create subgroups by using the GROUP BY clause. Further, groups can be restricted using the HAVING clause.

Place the HAVING and GROUP BY clauses after the WHERE clause in a statement. The order of the GROUP BY and HAVING clauses following the WHERE clause is not important. Place the ORDER BY clause at the end.

The Oracle server evaluates the clauses in the following order:

1. If the statement contains a WHERE clause, the server establishes the candidate rows.
2. The server identifies the groups that are specified in the GROUP BY clause.
3. The HAVING clause further restricts result groups that do not meet the group criteria in the HAVING clause.

Note: For a complete list of the group functions, see *Oracle Database SQL Language Reference* for 10*g* or 11*g* database.



Practice 5: Overview

This practice covers the following topics:

Writing queries

Writing queries that use the group functions

Practice 5: Overview

Grouping

Grouping by rows to achieve more than one result

HAVING clause

Restricting groups by using the HAVING clause

Practice 5: Overview

In this practice, you learn to use group functions and select groups of data.