

Oracle Academy

Database Programming with SQL

Instructor Resource Guide

Notes for PowerPoint Slides:

SECTION 6 LESSON 1 – GROUP By and HAVING clauses

Slide 1: GROUP BY and HAVING clauses

Lesson Preparation

Review Study Guide requirements with students. Have students complete the study guides as part of the Try It / Solve It portion of the content.

Slide 2: What Will I Learn?

No instructor notes for this slide

Slide 3: Why Learn It?

No instructor notes for this slide

Slide 4: Tell Me / Show Me – GROUP BY

You use the GROUP BY clause to divide ...

No instructor notes for this slide

Slide 5: Tell Me / Show Me – GROUP BY

What if we wanted to find the maximum salary ...

No instructor notes for this slide

Slide 6: Tell Me / Show Me – GROUP BY

Usually we want to include the GROUP ...

No instructor notes for this slide

Slide 7: Tell Me / Show Me – Group functions require that any column listed...

Answer: job_id is fine in the SELECT list, but last_name is not, because each group produces only one row of output, but employees doing the same job will probably have different last_names.

Slide 8: Tell Me / Show Me – This example shows how many students wear...

No instructor notes for this slide

Slide 9: Tell Me / Show Me – We can also use a WHERE clause...

No instructor notes for this slide

Slide 10: Tell Me / Show Me – More GROUP BY Examples:

No instructor notes for this slide

Slide 11: Tell Me / Show Me – GROUP BY Guidelines

No instructor notes for this slide

Slide 12: Tell Me / Show Me – GROUP WITHIN GROUPS

Point out that each unique combination of the GROUP BY columns provides one result row. If there were any Sales Managers working in department 60, then (60, SA_MAN) would be a different group from (80, SA_MAN).

Slide 13: Tell Me / Show Me – NESTING GROUP FUNCTIONS

No instructor notes for this slide

Slide 14: Tell Me / Show Me – HAVING

Suppose we want to find the maximum salary in ...

Answer: a WHERE clause can be used only to include/exclude individual rows, not groups of rows. Therefore we cannot use group functions in a WHERE clause.

Slide 15: Tell Me / Show Me – HAVING

In the same way you used the WHERE clause to...

No instructor notes for this slide

Slide 16: Tell Me / Show Me – HAVING

Although the HAVING clause can precede the ...

No instructor notes for this slide

Slide 17: Tell Me / Show Me - Terminology

HAVING-Used to specify which groups are to be displayed; restricts groups that do not meet group criteria

GROUP BY-Divides the rows in a table into groups

Slide 18: Summary – In this lesson you have learned to:

No instructor notes for this slide

Slide 19: Summary - Practice Guide

No instructor notes for this slide

SECTION 6 LESSON 2 – Subqueries

Slide 1: Subqueries

Lesson Preparation

Students have used the comparison operators from the very beginning of this course. Ask them why single-row subqueries can't use other comparison operator conditions such as BETWEEN...AND, IN, LIKE, or IS NOT NULL? Answer: If any of the subqueries using these operators return only one row, the subquery will execute. If more than one value is returned, an error will occur. If a null value is returned, no rows will be displayed.

Slide 2: What Will I Learn?

No instructor notes for this slide

Slide 3: Why Learn It?

No instructor notes for this slide

Slide 4: Tell Me / Show Me – Throughout this course, you have ..

No instructor notes for this slide

Slide 5: Tell Me / Show Me – A subquery is a SELECT statement...

No instructor notes for this slide

Slide 6: Tell Me / Show Me – Guidelines for using subqueries are:

No instructor notes for this slide

Slide 7: Tell Me / Show Me – There are two types of subqueries:

No instructor notes for this slide

Slide 8: Tell Me / Show Me – What if you wanted to find out the name...

Refer to the graphic. Ask students to find Monique Tuttle's birth date. Ask students who was born after Monique Tuttle? Tell students that they just did a subquery! Step through the execution of the subquery. Emphasize that the single-row operators can return only one row to the outer query. Note: The sample data shown is not in the database.

Slide 9: Tell Me / Show Me – If a subquery returns a null value..

Remind students that a WHERE condition returns only those rows for which the condition is TRUE. And NULL is not TRUE (it is not FALSE either!).

Slide 10: Tell Me /Show Me - Terminology

Outer query-It accepts a value from the inner query to solve its original query.

Multiple- row subquery>Returns more than one row from the inner query

Subquery-A SELECT statement that is embedded in a clause of another SELECT statement

Single- row subquery>Returns only one row from the inner SELECT statement

Inner query-It returns a value that is used by the outer query

Slide 11: Summary – In this lesson you have learned to:
No instructor notes for this slide

Slide 12: Summary - Practice Guide
No instructor notes for this slide

SECTION 6 LESSON 3 – Single-Row Subqueries

Slide 1: Single-Row Subqueries

Lesson Preparation

This lesson continues the discussion of subqueries from the previous lesson.

What to Watch For

Check for understanding of subqueries using group functions. Reinforce that when a group function appears in a SELECT clause, any columns not part of the group function must appear in a GROUP BY clause. Also, remind students that the HAVING clause is used in subqueries just as in other SELECT statements: to restrict the groups returned.

Connections

The concept of predefining or selecting data occurs in everyday life without our even thinking about it. An employer screening job applications wants to find the best candidate to hire. Before being able to interview anyone, the employer must identify candidates who meet the selection basic criteria. From the list of applicants, a person who meets the basic criteria can then be selected for the interview. A way to represent this process in "pseudocode" is:

```
SELECT new_employee  
FROM interview  
WHERE applicant =  
(SELECT applicant  
FROM applications  
WHERE degree = 'college' and 'qualification' IS NOT NULL);
```

Slide 2: What Will I Learn?

No instructor notes for this slide

Slide 3: Why Learn It?

Ask students how many of them use Internet search engines such as <http://www.google.com> or <http://www.yahoo.com>.

Relate the function of search engines in locating information to that of a subquery.

Slide 4: Tell Me / Show Me – Remember these facts about ...

Review the subquery rules with students.

Slide 5: Tell Me / Show Me – Also remember that:

Review the subquery rules with students. Point out that (at least for a single-row subquery) an ORDER BY clause in the subquery would not make sense; how can you order a single row?

Slide 6: Tell Me / Show Me – SUBQUERIES FROM DIFFERENT TABLES

The outer and inner queries can get data from different tables.

No instructor notes for this slide

Slide 7: Tell Me / Show Me – SUBQUERIES FROM DIFFERENT TABLES**The outer and inner queries can get data...**

Explain the graphic: the two subqueries retrieve data from two different tables and return it to the outer query. Note: Not all data is displayed in the graphic.

Point out that both the subqueries are single-row subqueries.

Slide 8: Tell Me / Show Me – GROUP FUNCTIONS IN SUBQUERIES

Review the following group functions: MIN, MAX, AVG, COUNT, SUM.

In the example shown, substitute MIN and AVG in the subquery and discuss what result would be returned.

MIN: Will someone have a salary less than the minimum salary?

AVG: Return the employees whose salary is less than the average salary for all employees.

Slide 9: Tell Me / Show Me – SUBQUERIES IN THE HAVING CLAUSE

No instructor notes for this slide

Slide 10: Tell Me / Show Me – Which departments have a ...

Work through this example with students, explaining the sequence of execution steps:

1. The subquery executes and returns the lowest salary in department 50: 2500.
2. The outer query groups the employee rows by department and calculates the lowest salary in each department.
3. The HAVING clause selects only those departments whose lowest salary is greater than 2500.

Slide 11: Summary – In this lesson you have learned to:

No instructor notes for this slide

Slide 12: Summary - Practice Guide

No instructor notes for this slide

SECTION 6 LESSON 4 – Multiple-Row Subqueries

Slide 1: Multiple-Row Subqueries

What to Watch For

Students will have difficulty knowing how to break down a database request into the outer and inner query statements. Help them identify what information is needed before the outer query can execute. Finding this unknown information is the function of the inner query. Look for the keywords in the problem, such as greater than, in, less than all, equal to any. The keywords will be the operators for the outer query. These keywords can help divide the problem into the two queries.

Slide 2: What Will I Learn?

No instructor notes for this slide

Slide 3: Why Learn It?

No instructor notes for this slide

Slide 4: Tell Me / Show Me – Whose salary is equal to the salary...

Point out that for the same reason, we cannot use <, > or <> in the WHERE clause condition. The key point is that it doesn't make sense to compare one value with several values. Is 10000 less than (13000,6000)?

Slide 5: Tell Me / Show Me – IN, ANY, and ALL

IN, ANY and ALL allow us to compare a single value with a set of values.

Slide 6: Tell Me / Show Me – IN

Point out to students the subquery results set and show how the values returned by the subquery become the values the outer query uses to restrict the rows returned. Ask students what will be returned from the outer query if there are no CD numbers less than 93.

Answer: No rows will be returned if a NULL value is returned by the subquery.

Slide 7: Tell Me / Show Me - ANY

Ask students which rows would be returned if we used <=ANY, >ANY and >=ANY.

<=ANY: The outer query would return titles whose year is less than or equal to 2001, ie 2000 or earlier.

>ANY: The outer query would return those titles whose year is greater than 2000, ie 2001 or later.

>=ANY: The outer query would return those titles whose year is greater than or equal to 2000, ie 2000 or later.

What about =ANY? Point about that =ANY will give the same results as IN.

In this case, if there are years in the d_cds table that have null values, the subquery will not return null as long as there are other non-null values in the year column.

Slide 8: Tell Me / Show Me – ALL

Explain the graphic. Ask students which rows would be returned if we used =ALL , < ALL and <> ALL.

=ALL: The outer query would return no rows. How can one value equal every one of a set of values ? For this reason, =ALL is rarely used.

<ALL: The outer query would return those titles whose year is less than 2000.

<>ALL: The outer query would return those titles whose year is not equal to all values returned by the subquery, ie whose year is not 2000 or 2001. <>ALL is equivalent to NOT IN.

Slide 9: Tell Me / Show Me – NULL VALUES

No instructor notes for this slide

Slide 10: Tell Me / Show Me – GROUP BY and HAVING

As you might suspect, the GROUP BY clause, and ...

No instructor notes for this slide

Slide 11: Tell Me / Show Me – GROUP BY and HAVING

Here is the needed SQL statement:

No instructor notes for this slide

Slide 12: Tell Me / Show Me – GROUP BY and HAVING

You can even have a GROUP BY clause in the subquery!

No instructor notes for this slide

Slide 13: Tell Me / Show Me – ONE LAST POINT ABOUT SUBQUERIES

Some subqueries may return a single row...

No instructor notes for this slide

Slide 14: Tell Me / Show Me – ONE LAST POINT ABOUT SUBQUERIES

It would be better to write a ...

Point out that when we use IN, ANY or ALL to compare with a list of values, they will still work even if there is only one value in the list. Does IT_PROG equal IT_PROG?

Yes. Is IT_PROG in a list of values which contains the single value IT_PROG? Also yes.

Slide 15: Summary – In this lesson you have learned to:

No instructor notes for this slide

Slide 16: Summary - Practice Guide

No instructor notes for this slide

Notes For Practice Activities:

Group By and Having Clauses S06 L01

Vocabulary

<u>HAVING</u>	Used to specify which groups are to be displayed; restricts groups that do not meet group criteria
---------------	---

<u>GROUP BY</u>	Divides the rows in a table into groups
-----------------	---

Try It / Solve It

1. a and d are True

2a. SELECT manager_id, AVG(salary)
FROM employees
GROUP BY manager_id;

2b. SELECT cd_number, COUNT(title)

FROM d_cds
WHERE cd_number < 93
GROUP BY cd_number;

2c. SELECT ID, MAX(ID), artist AS Artist
FROM d_songs
WHERE duration IN('3 min', '6 min', '10 min')
GROUP by ID, artist
HAVING ID < 50;

2d. SELECT loc_type, rental_fee AS Fee
FROM d_venues
WHERE id < 100
GROUP BY loc_type, rental_fee
ORDER BY 2;

3. SELECT MAX(song_id)
FROM d_track_listings
WHERE track <= 3;

4. SELECT last_name, first_name, AVG(test1) AS "Class Average"
FROM gradebook
WHERE grade_level = 10
GROUP BY gender, last_name, first_name
HAVING gender = 'M';

5a. TRUE

5b. FALSE

5c. FALSE

6. SELECT MAX(AVG(salary)) , MIN(AVG(salary))
FROM employees
GROUP BY department_id ;

7. SELECT AVG(MAX(salary))
FROM employees
GROUP BY department_id ;

Subqueries S06 L02

Vocabulary

<u>Outer query</u>	It accepts a value from the inner query to solve its original query.
<u>Multiple- row subquery</u>	Returns more than one row from the inner query
<u>Subquery</u>	A SELECT statement that is embedded in a clause of another SELECT statement
<u>Single- row subquery</u>	Returns only one row from the inner SELECT statement
<u>Inner query</u>	It returns a value that is used by the outer query

Try It / Solve It

1. A subquery is used when you don't have all the information needed to write a query.
2. A subquery is one query placed inside another.

```
3. SELECT song_id
FROM d_play_list_items
WHERE event_id=
(SELECT event_id
FROM d_play_list_items
WHERE song_id = 45);
```

```
4. SELECT name
FROM d_events
WHERE cost >
(SELECT cost
FROM d_events
WHERE id = 100);
```

```
5. SELECT track, cd_number
FROM d_track_listings
WHERE cd_number =
(SELECT cd_number
FROM d_cds
WHERE title LIKE 'Party Music for All Occasions');
```

```
6. SELECT name
FROM d_events
```

```
WHERE theme_code =  
(SELECT code  
FROM d_themes  
WHERE description = 'Tropical');
```

```
7. SELECT last_name  
FROM f_staffs  
WHERE salary >  
(SELECT salary  
FROM f_staffs  
WHERE id = 12);
```

```
8. SELECT last_name  
FROM f_staffs  
WHERE staff_type <>  
(SELECT staff_type  
FROM f_staffs  
WHERE last_name = 'Miller');
```

```
9. SELECT first_name, last_name  
FROM employees  
WHERE department_id =  
(SELECT department_id  
FROM departments  
WHERE department_name = 'IT');
```

```
10. SELECT department_name  
FROM departments  
WHERE location_id =  
(SELECT location_id  
FROM locations  
WHERE city= 'Seattle');
```

11a. True

11b. True.

Single Row Subqueries S06 L03

Try It / Solve It

```
1. SELECT *
FROM employees
WHERE salary > (SELECT salary
                FROM employees
                WHERE last_name = 'Lorentz')
AND department_id =(SELECT department_id
                    FROM employees
                    WHERE last_name = 'Abel') ;
```

```
2. SELECT *
FROM employees
WHERE job_id = (SELECT job_id
               FROM employees
               WHERE last_name = 'Rajs')
AND hire_date =(SELECT hire_date
                FROM employees
                WHERE last_name = 'Davies') ;
```

```
3. SELECT name
FROM d_events
WHERE theme_code =
(SELECT theme_code
FROM d_events
WHERE id = 100);
```

```
4. SELECT staff_type
FROM f_staffs
WHERE salary <
(SELECT max(salary)
FROM f_staffs
WHERE staff_type = 'Cook');
```

```
5. SELECT department_id, AVG(salary)
FROM employees
GROUP BY department_id
HAVING AVG(salary) > (SELECT salary
                     FROM employees
                     WHERE last_name = 'Ernst' );
```

```
6. SELECT department_id, min(salary)
FROM employees
GROUP BY department_id
```

```
HAVING min (salary)>  
(SELECT MIN (salary) FROM employees WHERE department_id <> 50);
```

Multiple-Row Subqueries S06 L04

Try It / Solve It

1. No results will be returned, the query will NOT return an error.

```
2. SELECT * FROM d_songs
WHERE type_code IN
(SELECT type_code FROM d_types
WHERE description IN ('Jazz','Pop'));
```

```
3. SELECT last_name
FROM employees
WHERE salary = ANY
(SELECT MIN(salary)
FROM employees
GROUP BY department_id);
```

4. Point out that there can be more than one way of getting the required information.

(multiple row subquery)

```
SELECT first_name, last_name
FROM f_staffs
WHERE salary <= ALL
(SELECT salary
FROM f_staffs);
```

(single-row subquery)

```
SELECT first_name, last_name
FROM f_staffs
WHERE salary=
(SELECT min(salary)
FROM f_staffs);
```

5a. <

5b. < ANY

5c. IN or = ANY

5d. > ALL

6. True: a, d

```
7. SELECT department_id, MIN(salary)
FROM employees
GROUP BY department_id
HAVING MIN(salary) <
(SELECT MIN(salary)
FROM employees
WHERE department_id = 50);
```

8. True: b, d