

Database Programming with SQL

* 1. : Using GROUP BY and HAVING Clauses Practice Activities

# Objectives

* + - Construct and execute a SQL query using GROUP BY
    - Construct and execute a SQL query using GROUP BY … HAVING
    - Construct and execute a GROUP BY on more than one column
    - Nest group functions

# Vocabulary

Identify the vocabulary word for each definition below.

|  |  |
| --- | --- |
| **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 |

# Try It / Solve It

1. In the SQL query shown below, which of the following is true about this query?

**T** a. Kimberly Grant would not appear in the results set.

**F** b. The GROUP BY clause has an error because the manager\_id is not listed in the SELECT clause.

**F** c. Only salaries greater than 16001 will be in the result set.

**F** d. Names beginning with Ki will appear after names beginning with Ko.

**F** e. Last names such as King and Kochhar will be returned even if they don’t have salaries > 16000.

SELECT last\_name, MAX(salary) FROM employees

WHERE last\_name LIKE 'K%' GROUP BY manager\_id, last\_name HAVING MAX(salary) >16000

ORDER BY last\_name DESC ;

1. Each of the following SQL queries has an error. Find the error and correct it. Use Oracle Application Express to verify that your corrections produce the desired results.
   1. SELECT manager\_id FROM employees

WHERE AVG(salary) <16000

GROUP BY manager\_id;

Above query will say ORA-00934: **group function is not allowed here**.

If 'desired' result is getting manger ids, under whom, average salary is less than 16000:

**SELECT manager\_id,AVG(salary)**

**FROM employees**

**GROUP BY manager\_id**

**HAVING AVG(salary) <16000;**

Изображение выглядит как стол

Автоматически созданное описание

* 1. SELECT cd\_number, COUNT(title) FROM d\_cds

WHERE cd\_number < 93;

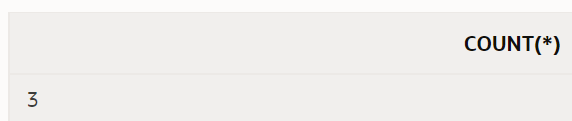
**Above query will say ORA-00937: not a single-group group function**

**If ‘desired’ result is getting row count, excluding anything greater than or equal to 93:**

**SELECT COUNT(\*)**

**FROM d\_cds**

**WHERE cd\_number < 93;**



* 1. SELECT ID, MAX(ID), artist AS Artist FROM d\_songs

WHERE duration IN('3 min', '6 min', '10 min') HAVING ID < 50

GROUP by ID;

**Above query will say ORA-00979: not a GROUP BY expression**

**id is primary key of the table, so most likely my intention may not be to group by id here.**

**Desired may have been grouping by type\_code and getting maximum duration**

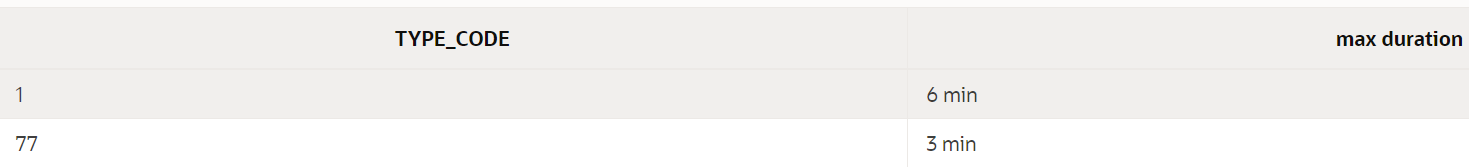
**but limiting the input to group by with duration IN('3 min', '6 min', '10 min')  and ID < 50**

**SELECT type\_code, MAX(TO\_NUMBER(REPLACE(duration,' min',''))) || ' min' as "max duration"**

**FROM d\_songs**

**WHERE duration IN('3 min', '6 min', '10 min') AND id < 50**

**GROUP BY type\_code;**

****

* 1. SELECT loc\_type, rental\_fee AS Fee FROM d\_venues

WHERE id <100 GROUP BY "Fee" ORDER BY 2;

**1) Above query will say ORA-00904: "Fee": invalid identifier**

**2) Even if I replace "Fee" alias in group by [don't use alias in group by clause] it will give me error : ORA-00979: not a GROUP BY expression**

**3) Seems to be question wanted average rental\_fee of each loc\_type with id less than 100 before grouping.**

**I assumed rental\_fee  is a number field:**

**SELECT loc\_type, AVG(rental\_fee) AS Fee**

**FROM d\_venues**

**WHERE id <100**

**GROUP BY loc\_type**

**ORDER BY 2;**

**But above query gives error ORA-01722: invalid number, because rental\_fee is a VARCHAR2(50)**

**4) Now, I need assumptions to view the calculations with same eyes: /hour, /flat fee, /per person.**

***Let’s assume that we are group of*10 people*and event I am organizing occurs for*5 hours*.***

**SELECT loc\_type, AVG(**

**CASE**

**WHEN INSTR(rental\_fee, '/hour') != 0 THEN TO\_NUMBER(REPLACE(rental\_fee,'/hour',''))\*5**

**WHEN INSTR(rental\_fee, '/flat fee') != 0 THEN TO\_NUMBER(REPLACE(rental\_fee,'/flat fee',''))**

**WHEN INSTR(rental\_fee, '/per person') != 0 THEN TO\_NUMBER(REPLACE(rental\_fee,'/per person',''))\*10**

**ELSE 0**

**END**

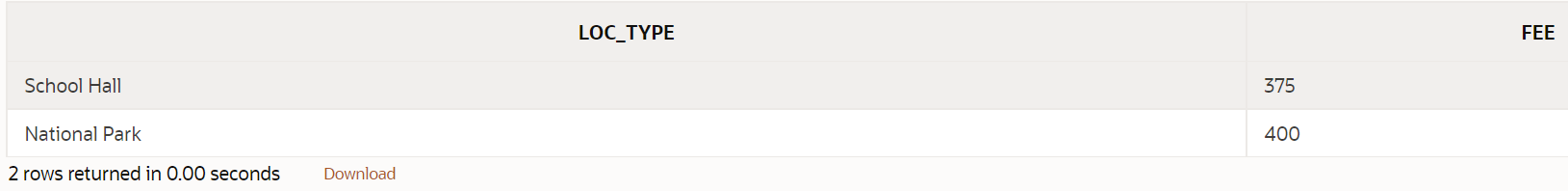
**) AS Fee**

**FROM d\_venues**

**WHERE id <100**

**GROUP BY loc\_type**

**ORDER BY 2;**

****

1. **Rewrite** the following query to accomplish the same result: SELECT DISTINCT MAX(song\_id)

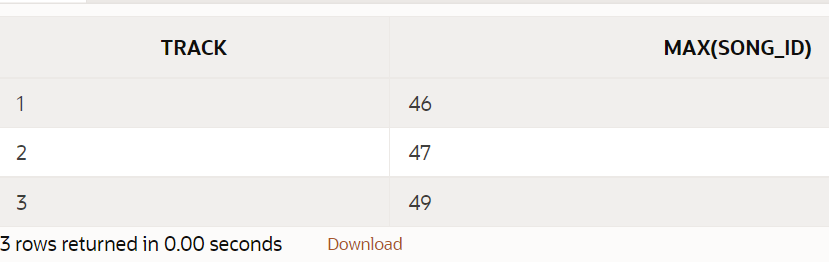
FROM d\_track\_listings WHERE track IN ( 1, 2, 3);

**SELECT track, MAX(song\_id)**

**FROM d\_track\_listings**

**WHERE track IN ( 1, 2, 3)**

**GROUP BY track;**

****

1. Indicate True or False

**T** a. If you include a group function and any other individual columns in a SELECT clause, then each individual column must also appear in the GROUP BY clause.

**F** b. You can use a column alias in the GROUP BY clause.

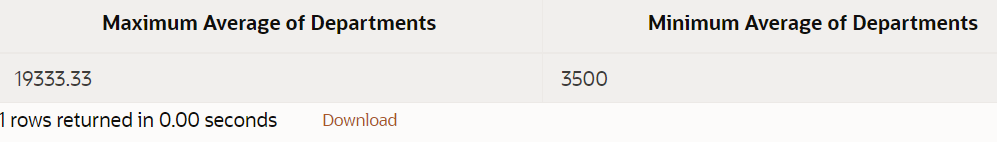
**F** c. The GROUP BY clause always includes a group function.

1. Write a query that will return both the maximum and minimum average salary grouped by department from the employees table.

**SELECT ROUND(MAX(AVG(salary)),2) as "Maximum Average of Departments", ROUND(MIN(AVG(salary)),2) "Minimum Average of Departments"**

**FROM employees**

**GROUP BY department\_id;**

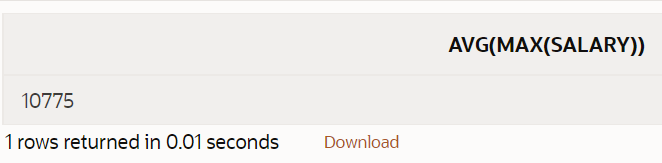
****

1. Write a query that will return the average of the maximum salaries in each department for the employees table.

**SELECT AVG(MAX(salary))**

**FROM employees**

**GROUP BY department\_id;**

****

Copyright © 2020, Oracle and/or its affiliates. All rights reserved. Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.