

Section 5 Lesson 1: Using GROUP BY and HAVING Clauses

Try It / Solve It

1. In the SQL query shown below, which of the following is true about this query?
- TRUE a. Kimberly Grant would not appear in the results set.
- _____ b. The GROUP BY clause has an error because the manager_id is not listed in the SELECT clause.
- _____ c. Only salaries greater than 16001 will be in the result set.
- TRUE d. Names beginning with Ki will appear after names beginning with Ko.
- _____ 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 ;
```

2. 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.

a. SELECT manager_id FROM employees WHERE AVG(salary) < 16000 GROUP BY manager_id;	SELECT manager_id FROM employees HAVING AVG(salary) < 16000 GROUP BY manager_id;
b. SELECT cd_number, COUNT(title) FROM d_cds WHERE cd_number < 93;	SELECT cd_number, COUNT(title) FROM d_cds HAVING cd_number < 93 GROUP BY CD_NUMBER;
c. 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;	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;
d. SELECT loc_type, rental_fee AS Fee FROM d_venues WHERE id < 100 GROUP BY "Fee" ORDER BY 2;	SELECT loc_type, rental_fee AS Fee FROM d_venues WHERE id < 100 GROUP BY rental_fee, loc_type ORDER BY 2;

3. 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 MAX(DISTINCT song_id)
FROM d_track_listings
WHERE track IN ( 1, 2, 3);
```

4. Indicate True or False

TRUE 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.

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

FALSE c. The GROUP BY clause always includes a group function.

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

The screenshot shows the Oracle SQL Workshop interface. The SQL Commands window contains the following query:

```
SELECT MAX(SALARY) "Max Salary", MIN(SALARY) "Min Salary"
FROM EMPLOYEES
GROUP BY DEPARTMENT_ID;
```

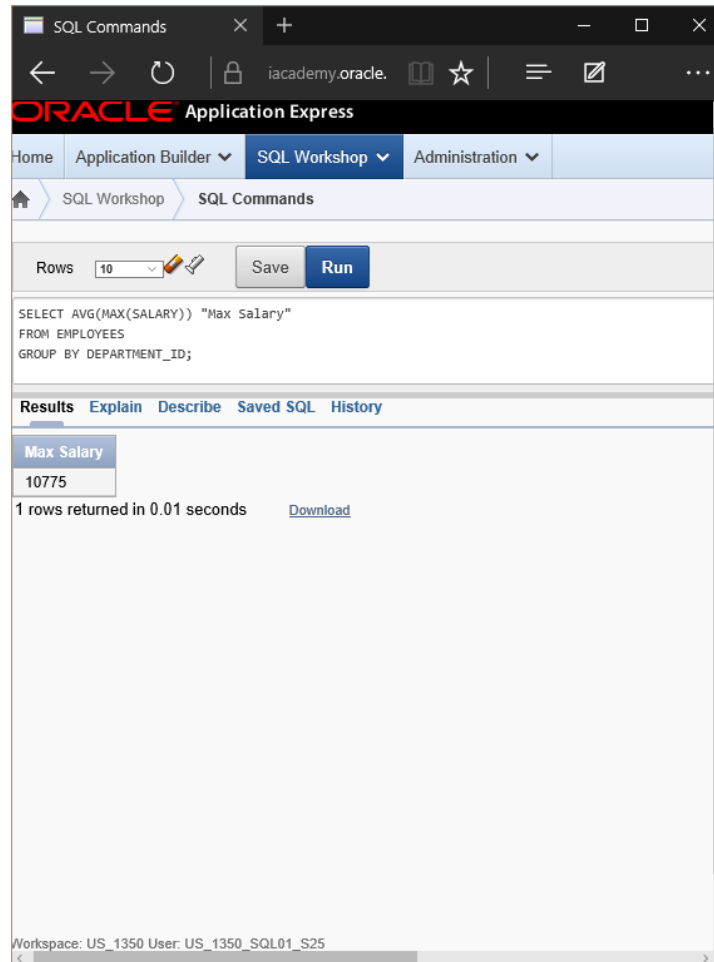
The Results tab shows the following data:

Max Salary	Min Salary
7000	7000
24000	17000
13000	6000
12000	8300
11000	8600
5800	2500
4400	4400
9000	4200

8 rows returned in 0.00 seconds

Workspace: US_1350 User: US_1350_SQL01_S25

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



The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes 'Home', 'Application Builder', 'SQL Workshop', and 'Administration'. The 'SQL Workshop' tab is active, and the 'SQL Commands' sub-tab is selected. Below the navigation bar, there is a 'Rows' dropdown set to '10', a 'Save' button, and a 'Run' button. The SQL command area contains the following query:

```
SELECT AVG(MAX(SALARY)) "Max Salary"
FROM EMPLOYEES
GROUP BY DEPARTMENT_ID;
```

Below the query, the 'Results' tab is active, displaying a table with one row:

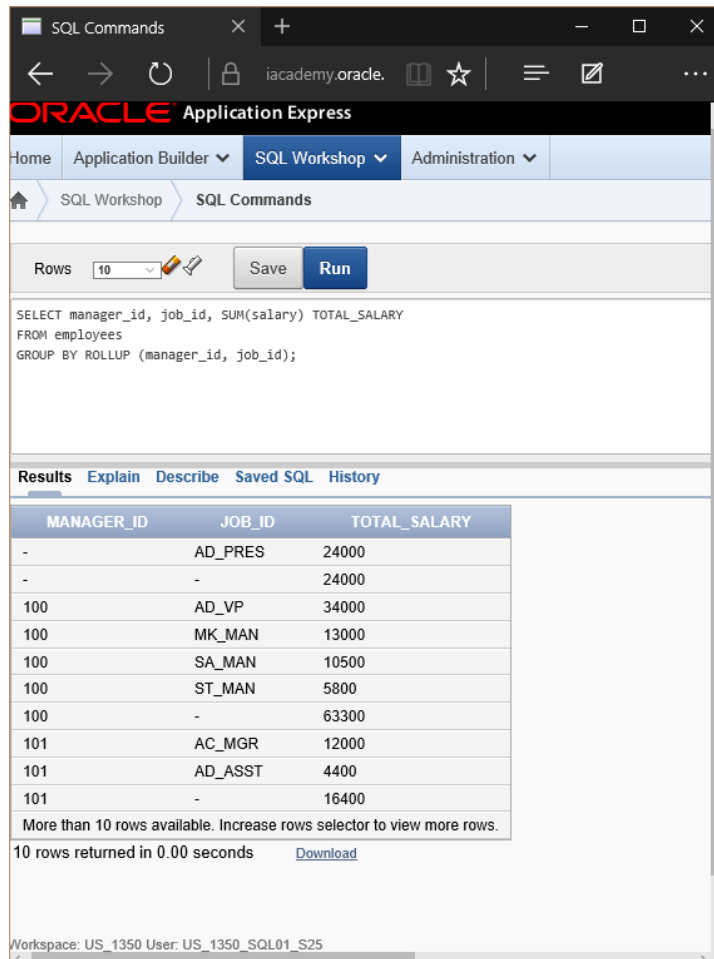
Max Salary
10775

Below the table, it states '1 rows returned in 0.01 seconds' and provides a 'Download' link. The bottom status bar shows 'Workspace: US_1350 User: US_1350_SQL01_S25'.

Section 5 Lesson 2: Using ROLLUP and CUBE Operations, and GROUPING SETS

Try It / Solve It

1. Within the Employees table, each manager_id is the manager of one or more employees who each have a job_id and earn a salary. For each manager, what is the total salary earned by all of the employees within each job_id? Write a query to display the Manager_id, job_id, and total salary. Include in the result the subtotal salary for each manager and a grand total of all salaries.



The screenshot shows the Oracle SQL Workshop interface. The SQL Commands window contains the following query:

```
SELECT manager_id, job_id, SUM(salary) TOTAL_SALARY
FROM employees
GROUP BY ROLLUP (manager_id, job_id);
```

The Results tab shows the output of the query. The results are displayed in a table with the following columns: MANAGER_ID, JOB_ID, and TOTAL_SALARY. The table contains 10 rows, showing the total salary for each manager and job combination, including subtotals for each manager and a grand total for all salaries.

MANAGER_ID	JOB_ID	TOTAL_SALARY
-	AD_PRES	24000
-	-	24000
100	AD_VP	34000
100	MK_MAN	13000
100	SA_MAN	10500
100	ST_MAN	5800
100	-	63300
101	AC_MGR	12000
101	AD_ASST	4400
101	-	16400

More than 10 rows available. Increase rows selector to view more rows.
10 rows returned in 0.00 seconds [Download](#)

Workspace: US_1350 User: US_1350_SQL01_S25

2. Amend the previous query to also include a subtotal salary for each job_id regardless of the manager_id.

The screenshot shows the Oracle SQL Workshop interface. The top navigation bar includes 'Home', 'Application Builder', 'SQL Workshop', and 'Administration'. The 'SQL Workshop' tab is active, and the 'SQL Commands' sub-tab is selected. Below the navigation bar, there is a 'Rows' selector set to 10, and 'Save' and 'Run' buttons. The SQL command area contains the following query:

```
SELECT manager_id, job_id, SUM(salary) TOTAL_SALARY
FROM employees
GROUP BY CUBE (manager_id, job_id);
```

Below the query, the 'Results' tab is active, displaying a table with the following data:

MANAGER_ID	JOB_ID	TOTAL_SALARY
-	-	24000
-	-	175500
-	AD_VP	34000
-	AC_MGR	12000
-	MK_MAN	13000
-	MK_REP	6000
-	SA_MAN	10500
-	SA_REP	26600
-	ST_MAN	5800
-	AD_ASST	4400

Below the table, a message states: 'More than 10 rows available. Increase rows selector to view more rows.' Below this, it says '10 rows returned in 0.00 seconds' with a 'Download' link. At the bottom, the workspace information is shown: 'Workspace: US_1350 User: US_1350_SQL01_S25'.

3. Using GROUPING SETS, write a query to show the following groupings:

- department_id, manager_id, job_id
- manager_id, job_id
- department_id, manager_id

The screenshot shows the Oracle SQL Workshop interface. The SQL Commands window contains the following query:

```
SELECT department_id, manager_id, job_id, SUM(salary) TOTAL_SALARY
FROM employees
GROUP BY GROUPING SETS ((department_id, manager_id, job_id),
                        (manager_id, job_id),
                        (department_id, manager_id));
```

The Results tab displays the following data:

DEPARTMENT_ID	MANAGER_ID	JOB_ID	TOTAL_SALARY
90	-	AD_PRES	24000
90	100	AD_VP	34000
20	100	MK_MAN	13000
80	100	SA_MAN	10500
50	100	ST_MAN	5800
110	101	AC_MGR	12000
10	101	AD_ASST	4400
60	102	IT_PROG	9000
60	103	IT_PROG	10200
50	124	ST_CLERK	11700

More than 10 rows available. Increase rows selector to view more rows.

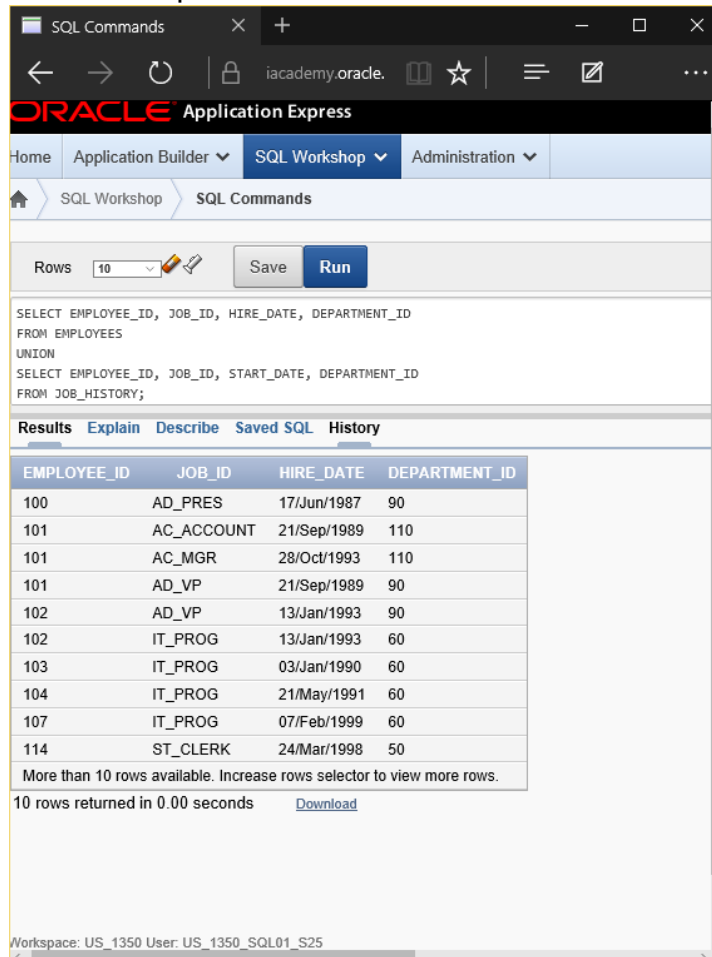
10 rows returned in 0.02 seconds [Download](#)

Workspace: US_1350 User: US_1350_SQL01_S25

Section 5 Lesson 3: Set Operators

Try It / Solve It

1. Name the different Set operators? [UNION](#), [UNION ALL](#), [INTERSECT](#), [MINUS](#)
2. Write one query to return the employee_id, job_id, hire_date, and department_id of all employees and a second query listing employee_id, job_id, start_date, and department_id from the job_hist table and combine the results as one single output. Make sure you suppress duplicates in the output.



The screenshot shows the Oracle Application Express SQL Workshop interface. The SQL Commands window contains the following query:

```
SELECT EMPLOYEE_ID, JOB_ID, HIRE_DATE, DEPARTMENT_ID
FROM EMPLOYEES
UNION
SELECT EMPLOYEE_ID, JOB_ID, START_DATE, DEPARTMENT_ID
FROM JOB_HISTORY;
```

The Results tab displays the output of the query, showing 10 rows. The columns are EMPLOYEE_ID, JOB_ID, HIRE_DATE, and DEPARTMENT_ID. The results are as follows:

EMPLOYEE_ID	JOB_ID	HIRE_DATE	DEPARTMENT_ID
100	AD_PRES	17/Jun/1987	90
101	AC_ACCOUNT	21/Sep/1989	110
101	AC_MGR	28/Oct/1993	110
101	AD_VP	21/Sep/1989	90
102	AD_VP	13/Jan/1993	90
102	IT_PROG	13/Jan/1993	60
103	IT_PROG	03/Jan/1990	60
104	IT_PROG	21/May/1991	60
107	IT_PROG	07/Feb/1999	60
114	ST_CLERK	24/Mar/1998	50

Below the table, it states: "More than 10 rows available. Increase rows selector to view more rows." and "10 rows returned in 0.00 seconds" with a [Download](#) link.

3. Amend the previous statement to not suppress duplicates and examine the output. How many extra rows did you get returned and which were they? Sort the output by employee_id to make it easier to spot.

One extra row was returned.

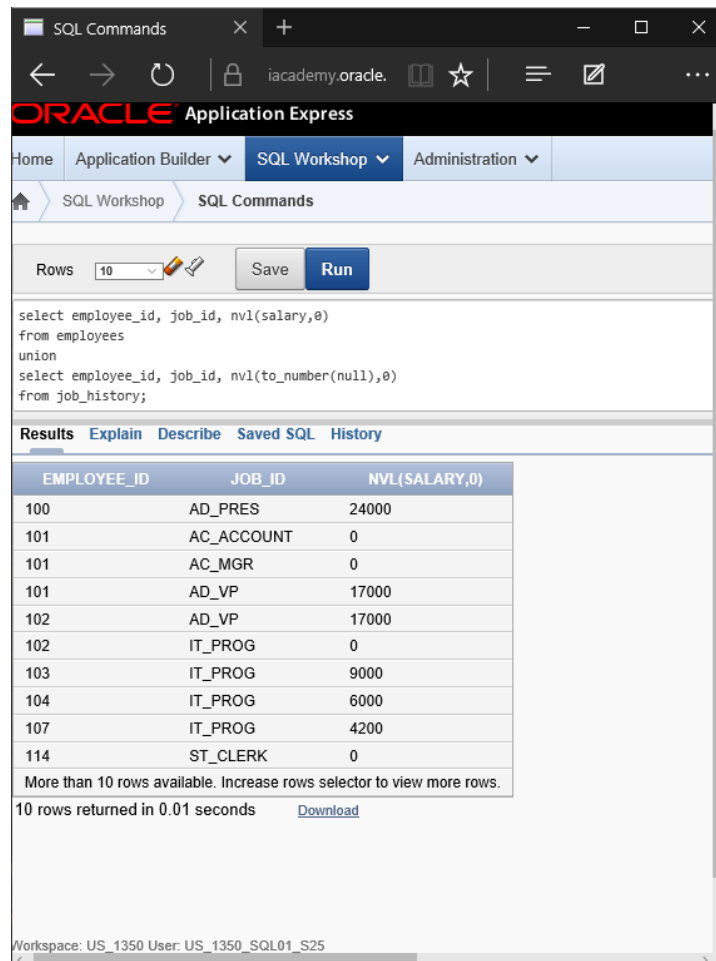
4. List all employees who have not changed jobs even once. (Such employees are not found in the job_history table)

EMPLOYEE_ID
100
103
104
107
124
141
142
143
144
149
174
178
202
205
206

5. List the employees that HAVE changed their jobs at least once.

EMPLOYEE_ID
101
102
114
122
176
200
201

6. Using the UNION operator, write a query that displays the employee_id, job_id, and salary of ALL present and past employees. If a salary is not found, then just display a 0 (zero) in its place.



The screenshot shows the Oracle SQL Workshop interface. The SQL Commands window contains the following query:

```
select employee_id, job_id, nvl(salary,0)
from employees
union
select employee_id, job_id, nvl(to_number(null),0)
from job_history;
```

The Results tab displays the following data:

EMPLOYEE_ID	JOB_ID	NVL(SALARY,0)
100	AD_PRES	24000
101	AC_ACCOUNT	0
101	AC_MGR	0
101	AD_VP	17000
102	AD_VP	17000
102	IT_PROG	0
103	IT_PROG	9000
104	IT_PROG	6000
107	IT_PROG	4200
114	ST_CLERK	0

More than 10 rows available. Increase rows selector to view more rows.
10 rows returned in 0.01 seconds [Download](#)

Workspace: US_1350 User: US_1350_SQL01_S25