

LAB ASSIGNMENT -11

1. Write a query to display the following for those employees whose manager Id is less than 120:
 - Manager ID
 - Job Id and total salary for every job ID for employees who report to the same manager
 - Total salary of those managers
 - Total salary of those managers, irrespective of the job IDs.

QUERY-

```
select manager_id,job_id,sum(salary)
from employees
where manager_id < 120
group by rollup (manager_id,job_id);
```

The screenshot shows the Oracle Live SQL interface. The SQL Worksheet contains the following query:

```
1 select manager_id,job_id,sum(salary)
2 from employees
3 where manager_id < 120
4 group by rollup (manager_id,job_id);
5
```

The results are displayed in a table with the following data:

MANAGER_ID	JOB_ID	SUM(SALARY)
100	AD_VP	34000
100	HR_MGR	13000
100	SA_MGR	10500
100	ST_MGR	5800
100	-	63300
101	AC_MGR	12000
101	AD_ASST	4400
101	-	16400
102	IT_PROG	9000
102	-	9000
103	IT_PROG	48000
103	-	48000
-	-	136700

Download CSV
4.3 KB, 1000 rows

© 2022 Oracle - Live SQL 22.1.3, running Oracle Database 19c Enterprise Edition - 19.0.0.0.0 - Database Documentation - Ask Tom - Dev Gym
Built with ♥ using Oracle APEX - Privacy - Terms of Use

2. Observe the output from question 1. Write a query using the GROUPING function to determine whether the NULL values in the column corresponding to the GROUP BY expressions are caused by the ROLLUP operation.

Query:

```
select manager_id MGR ,job_id JOB,
sum(salary),grouping(manager_id),grouping(job_id)
from employees
where manager_id < 120
group by rollup(manager_id,job_id);
```

The screenshot shows the Oracle Live SQL interface. The SQL Worksheet contains the following query:

```
1 select manager_id MGR ,job_id JOB,
2 sum(salary),grouping(manager_id),grouping(job_id)
3 from employees
4 where manager_id < 120
5 group by rollup(manager_id,job_id);
6
7
```

The results are displayed in a table with the following columns: MGR, JOB, SUM(SALARY), GROUPING(MANAGER_ID), and GROUPING(JOB_ID). The table shows data for managers with IDs less than 120, grouped by job ID. The GROUPING columns indicate whether the values are NULL (0) or not (1).

MGR	JOB	SUM(SALARY)	GROUPING(MANAGER_ID)	GROUPING(JOB_ID)
100	AD_VP	24000	0	0
100	PK_MAN	13000	0	0
100	SA_MAN	10500	0	0
100	ST_MAN	5800	0	0
100	-	63300	0	1
101	AC_MGR	12000	0	0
101	AD_ASST	4400	0	0
101	-	16400	0	1
102	IT_PROG	9000	0	0
102	-	9000	0	1
103	IT_PROG	48000	0	0
103	-	48000	0	1
-	-	136700	1	1

Download CSV
13 rows selected.

© 2022 Oracle - Live SQL 22.1.3, running Oracle Database 19c Enterprise Edition - 19.8.0.0.0 - Database Documentation - Ask Tom - Dev Gym
Built with ♥ using Oracle APEX - Privacy - Terms of Use

3. Write a query to display the following for those employees whose manager Id is less than 120:

- Manager Id
- Job and total salaries for every job for employees who report to the same manager
- Total salary of those managers
- Cross-tabulation values to display the total salary for every job, irrespective of the manager.
- Total salary irrespective of all job titles.

Query :

```
select manager_id, job_id, sum(salary)
from employees
where manager_id < 120
group by cube(manager_id, job_id);
```

The screenshot shows the Oracle Live SQL interface. The query entered is:

```
select manager_id MGR , job_id JOB,
sum(salary), grouping(manager_id), grouping(job_id)
from employees
where manager_id < 120
group by rollup(manager_id, job_id);

select manager_id, job_id, sum(salary)
from employees
where manager_id < 120
group by cube(manager_id, job_id);
```

The results of the second query are displayed in a table:

MANAGER_ID	JOB_ID	SUM(SALARY)
-	-	136700
-	AD_VP	34000
-	AC_MGR	12000
-	HR_MAN	13000
-	SA_MAN	10500
-	ST_MAN	5800
-	AD_ASST	4400
-	IT_PROG	57000
100	-	63300
100	AD_VP	34000
100	HR_MAN	13000
100	SA_MAN	10500
100	ST_MAN	5800
101	-	16400

At the bottom, the footer indicates: © 2022 Oracle - Live SQL 22.1.3, running Oracle Database 19c Enterprise Edition - 19.8.0.0.0 - Database Documentation - Ask Tom - Dev Gym. Built with ♥ Using Oracle APEX - Privacy - Terms of Use.

4. Observe the output from question 3. Write a query using the GROUPING function to determine whether the NULL values in the columns corresponding to the GROUP BY expressions are caused by the CUBE operation.

Query:

```
select manager_id MGR ,job_id JOB,
sum(salary),grouping(manager_id),grouping(job_id)
from employees
where manager_id < 120
group by cube(manager_id,job_id);
```

The screenshot shows the Oracle Live SQL interface. The query editor contains the following SQL code:

```
13
14
15 select manager_id, job_id, sum(salary)
16 from employees
17 where manager_id < 120
18 group by cube(manager_id, job_id);
19
20
21 select manager_id MGR ,job_id JOB,
22 sum(salary),grouping(manager_id),grouping(job_id)
23 from employees
24 where manager_id < 120
25 group by cube(manager_id,job_id);]
26
```

The results table displays the output of the query. The columns are MGR, JOB, SUM(SALARY), GROUPING(MANAGER_ID), and GROUPING(JOB_ID). The table shows data for various managers and jobs, with the GROUPING columns indicating whether the values are NULL (0) or not (1).

MGR	JOB	SUM(SALARY)	GROUPING(MANAGER_ID)	GROUPING(JOB_ID)
-	-	136700	1	1
-	AD_VP	34000	1	0
-	AC_MGR	12000	1	0
-	PK_MAN	13000	1	0
-	SA_MAN	18500	1	0
-	ST_MAN	5800	1	0
-	AD_ASST	4400	1	0
-	IT_PROG	57000	1	0
100	-	63300	0	1
100	AD_VP	34000	0	0
100	PK_MAN	13000	0	0
100	SA_MAN	18500	0	0
100	ST_MAN	5800	0	0
101	-	16400	0	1
101	AC_MGR	12000	0	0

© 2022 Oracle - Live SQL 22.1.3, running Oracle Database 19c Enterprise Edition - 19.8.0.0.0 - Database Documentation - Ask Tom - Dev Gym
Built with ❤️ using Oracle APEX - Privacy - Terms of Use

5. Using GROUPING SETS, write a query to display the following groupings:
- department_id, manager-id, job-id
 - department_id, job-id
 - manager_id , job_id

The query should calculate the sum of the salaries for each of these groups.

Query:

```
select department_id, manager_id, job_id, SUM(salary)
from employees
group by
grouping sets ((department_id, manager_id, job_id),
(department_id, job_id),(manager_id,job_id));
```

The screenshot shows the Oracle Live SQL interface. The query editor contains the following SQL code:

```
21 select manager_id MGR ,job_id JOB,
22 sum(salary),grouping(manager_id),grouping(job_id)
23 from employees
24 where manager_id < 120
25 group by cube(manager_id,job_id);
26
27
28 select department_id, manager_id, job_id, SUM(salary)
29 from employees
30 group by
31 grouping sets ((department_id, manager_id, job_id),
32 (department_id, job_id),(manager_id,job_id));
33
```

The results table displays the sum of salaries for each combination of department_id, manager_id, and job_id. The table has 16 rows and 4 columns: DEPARTMENT_ID, MANAGER_ID, JOB_ID, and SUM(SALARY).

DEPARTMENT_ID	MANAGER_ID	JOB_ID	SUM(SALARY)
10	101	AD_ASST	4400
20	100	HR_MGR	13000
20	201	HR_REP	6000
50	100	ST_MGR	5800
50	124	ST_CLERK	11700
60	102	IT_PROG	9000
60	103	IT_PROG	40000
80	100	SA_MGR	10500
80	140	SA_REP	26640
90	100	AD_VP	34000
90	-	AD_PRES	24000
110	101	AC_MGR	12000
110	205	AC_ACCOUNT	8300
-	140	SA_REP	26640
-	124	ST_CLERK	11700

© 2022 Oracle - Live SQL 22.1.3, running Oracle Database 19c Enterprise Edition - 19.8.0.0.0 - Database Documentation - Ask Tom - Dev Gym
Built with ♥ using Oracle APEX - Privacy - Terms of Use