 



Database Programming with SQL

* 1. : Using ROLLUP and CUBE Operations and GROUPING SETS Practice Activities

# Objectives

* + - Use ROLLUP to produce subtotal values
    - Use CUBE to produce cross-tabulation values
    - Use GROUPING SETS to produce a single result set
    - Use the GROUPING function to identify the extra row values created by either a ROLLUP or CUBE operation

# Vocabulary

Identify the vocabulary word for each definition below.

|  |  |
| --- | --- |
| **ROLLUP** | Used to create subtotals that roll up from the most detailed level to a grand total, following a grouping list specified in the clause |
| **CUBE** | An extension to the GROUP BY clause like ROLLUP that produces cross-tabulation reports |
| **GROUPING SETS** | Used to specify multiple groupings of data |

# 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.

**SELECT manager\_id, job\_id, SUM(salary) "total salary", GROUPING(manager\_id), GROUPING(job\_id)**

**FROM employees**

**GROUP BY ROLLUP(manager\_id, job\_id);**

**Изображение выглядит как стол

Автоматически созданное описание**

1. Amend the previous query to also include a subtotal salary for each job\_id regardless of the manager\_id.

**SELECT manager\_id, job\_id, SUM(salary) "total salary", GROUPING(manager\_id), GROUPING(job\_id)**

**FROM employees**

**GROUP BY CUBE(manager\_id, job\_id)**

**Изображение выглядит как стол

Автоматически созданное описание**

1. 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

**SELECT department\_id, manager\_id, job\_id, SUM(salary) "total salary", GROUPING(department\_id), GROUPING(manager\_id), GROUPING(job\_id)**

**FROM employees**

**GROUP BY GROUPING SETS((department\_id, manager\_id, job\_id), (manager\_id, job\_id), (department\_id, manager\_id));**

Изображение выглядит как стол

Автоматически созданное описание

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.