• Problem21:

− Create a list of department names and their monthly costs (salaries added up)

• Tables Used: − Employees, Department

SELECT

d.DepartmentName,

SUM(e.Salary) AS MonthlyCost

FROM

Employees e

JOIN

Department d ON e.DepartmentID = d.DepartmentID

GROUP BY

d.DepartmentName;

• Problem21:

− Create a list of department names, and job\_ids

− Calculate the monthly salary cost for each job\_idwithin a department, for each department, and for all departments added together

• Tables Used: − Employees, Department

-- Calculate the monthly salary cost for each job\_id within each department

SELECT

d.DepartmentName,

e.JobID,

SUM(e.Salary) AS MonthlyCost

FROM

Employees e

JOIN

Department d ON e.DepartmentID = d.DepartmentID

GROUP BY

d.DepartmentName,

e.JobID

UNION ALL

-- Calculate the monthly salary cost for each department (all job\_ids within the department)

SELECT

d.DepartmentName,

'ALL' AS JobID,

SUM(e.Salary) AS MonthlyCost

FROM

Employees e

JOIN

Department d ON e.DepartmentID = d.DepartmentID

GROUP BY

d.DepartmentName

UNION ALL

-- Calculate the total monthly salary cost for all departments together

SELECT

'ALL DEPARTMENTS' AS DepartmentName, 'ALL' AS JobID, SUM(e.Salary) AS MonthlyCost

FROM Employees e;

•Problem22:

−Create a list of department names, and job\_ids

−Calculate the monthly salary cost for each job\_idwithin a department, for each department, for each group of job\_idsirrespective of the department, and for all departments added together

(Hint: Cube)

•Tables Used: −Employees, Departments

SELECT

COALESCE(d.DepartmentName, 'ALL DEPARTMENTS') AS DepartmentName,

COALESCE(e.JobID, 'ALL JOBS') AS JobID,

SUM(e.Salary) AS MonthlyCost

FROM

Employees e

JOIN

Department d ON e.DepartmentID = d.DepartmentID

GROUP BY

CUBE(d.DepartmentName, e.JobID)

ORDER BY

DepartmentName ASC,

JobID ASC;

• Problem23:

− Expand the previous list to also show if the department\_idor job\_id was used to create the subtotals shown in the output (Hint: Cube, Grouping)

• Tables Used: − Employees, Department

SELECT

COALESCE(d.DepartmentName, 'ALL DEPARTMENTS') AS DepartmentName,

COALESCE(e.JobID, 'ALL JOBS') AS JobID,

SUM(e.Salary) AS MonthlyCost,

GROUPING(d.DepartmentName) AS IsDepartmentSubtotal,

GROUPING(e.JobID) AS IsJobSubtotal

FROM

Employees e

JOIN

Department d ON e.DepartmentID = d.DepartmentID

GROUP BY

CUBE(d.DepartmentName, e.JobID)

ORDER BY

DepartmentName ASC,

JobID ASC;

• Problem24:

− Create a list that includes the monthly salary costs for each job title within a department

− In the same list, display the monthly salary cost per city. (Hint: Grouping Sets)

• Tables Used: − Employees, Departments, Locations

**SELECT**

**COALESCE(d.DepartmentName, 'ALL DEPARTMENTS') AS DepartmentName,**

**COALESCE(e.JobID, 'ALL JOBS') AS JobID,**

**COALESCE(l.City, 'ALL CITIES') AS City,**

**SUM(e.Salary) AS MonthlyCost**

**FROM**

**Employees e**

**JOIN**

**Departments d ON e.DepartmentID = d.DepartmentID**

**JOIN**

**Locations l ON d.LocationID = l.LocationID**

**GROUP BY**

**GROUPING SETS (**

**(d.DepartmentName, e.JobID), -- Group by Department and JobID**

**(l.City) -- Group by City**

**)**

**ORDER BY**

**DepartmentName ASC,**

**JobID ASC,**

**City ASC;**

**•Problem25:**

**−Create a list of employee names as shown and department ids**

**−In the same report, list the department ids and department names. And finally, list the cities**

**−The rows should not be joined, just listed in the same report. (Hint: Union)**

**•Tables Used: −Employees, Departments, Location**

**-- List of employee names and department IDs**

**SELECT**

**e.Name AS DisplayName,**

**e.DepartmentID AS DisplayValue,**

**'Employee' AS DataType**

**FROM**

**Employees e**

**UNION ALL**

**-- List of department IDs and department names**

**SELECT**

**d.DepartmentID AS DisplayName,**

**d.DepartmentName AS DisplayValue,**

**'Department' AS DataType**

**FROM**

**Departments d**

**UNION ALL**

**-- List of cities**

**SELECT**

**l.City AS DisplayName,**

**NULL AS DisplayValue,**

**'City' AS DataType**

**FROM**

**Locations l**

**ORDER BY**

**DataType,**

**DisplayName;**

**• Problem26:**

**− Create a list of each employee's first initial and last name, salary, and department name for each employee earning more than the average for his department**

**• Tables Used: − Departments, Employee**

**SELECT**

**CONCAT(SUBSTRING(e.FirstName, 1, 1), '. ', e.LastName) AS EmployeeName,**

**e.Salary,**

**d.DepartmentName**

**FROM**

**Employees e**

**JOIN**

**Departments d ON e.DepartmentID = d.DepartmentID**

**WHERE**

**e.Salary > (**

**SELECT**

**AVG(e2.Salary)**

**FROM**

**Employees e2**

**WHERE**

**e2.DepartmentID = e.DepartmentID**

**)**

**ORDER BY**

**e.LastName ASC;**