SQL Programming – Level 2 Programming Project 06

# Joins – Lunches

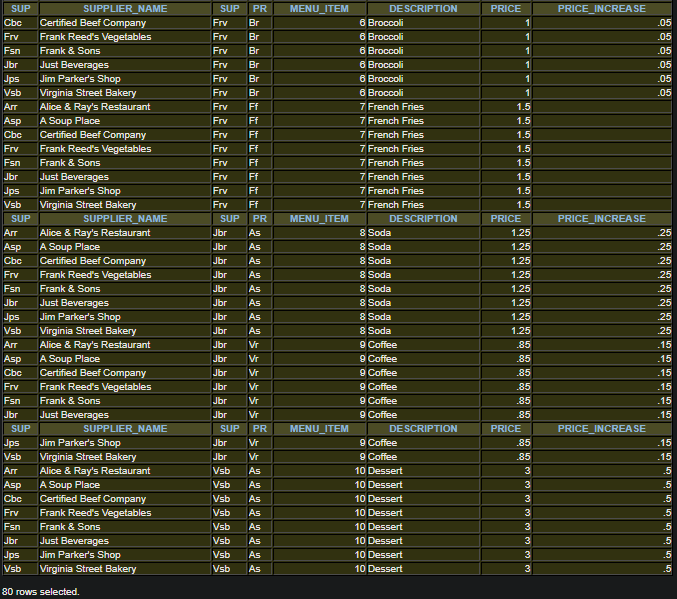
Reminder: read the Project Guidelines document for instructions on how to format and submit your assignments. It may also be advisable for you to consider formatting your output in landscape orientation so that you can better accommodate the ‘wider’ result tables that are formed in these join problems. ***In this problem set, use the FROM clause to form the join.***

## Use the Oracle 9i server for questions 1 thru 12.

1. Cross join the suppliers table (L\_suppliers) with the foods (L\_foods) table. Show all columns. [80 rows]

SELECT \* FROM L\_suppliers

CROSS JOIN L\_foods



1. Equi join the suppliers table (L\_suppliers) with the foods (L\_foods) table. Show all columns. [10 rows]

SELECT \* FROM L\_suppliers

JOIN L\_foods

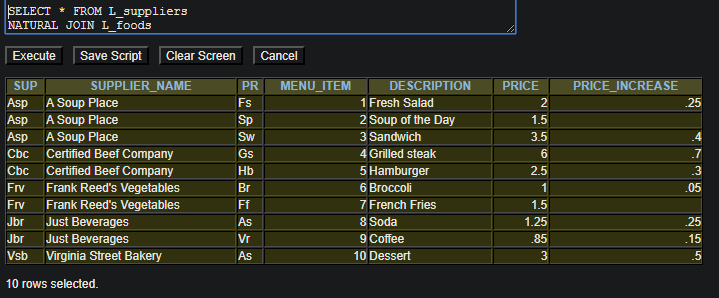
ON L\_suppliers.SUPPLIER\_ID = L\_foods.SUPPLIER\_ID



1. Natural join the suppliers table (L\_suppliers) with the foods (L\_foods) table. Show all columns. [10 rows]

SELECT \* FROM L\_suppliers

NATURAL JOIN L\_foods



1. Equi join the L-lunch\_items table with the L\_suppliers table, with the L-foods table. Show these columns in this order: lunch\_id, item\_number, product code, supplier\_id, supplier\_name. [71 rows]

SELECT L\_lunch\_items.LUNCH\_ID,

L\_lunch\_items.ITEM\_NUMBER,

L\_lunch\_items.PRODUCT\_CODE,

L\_suppliers.SUPPLIER\_ID,

L\_suppliers.SUPPLIER\_ID

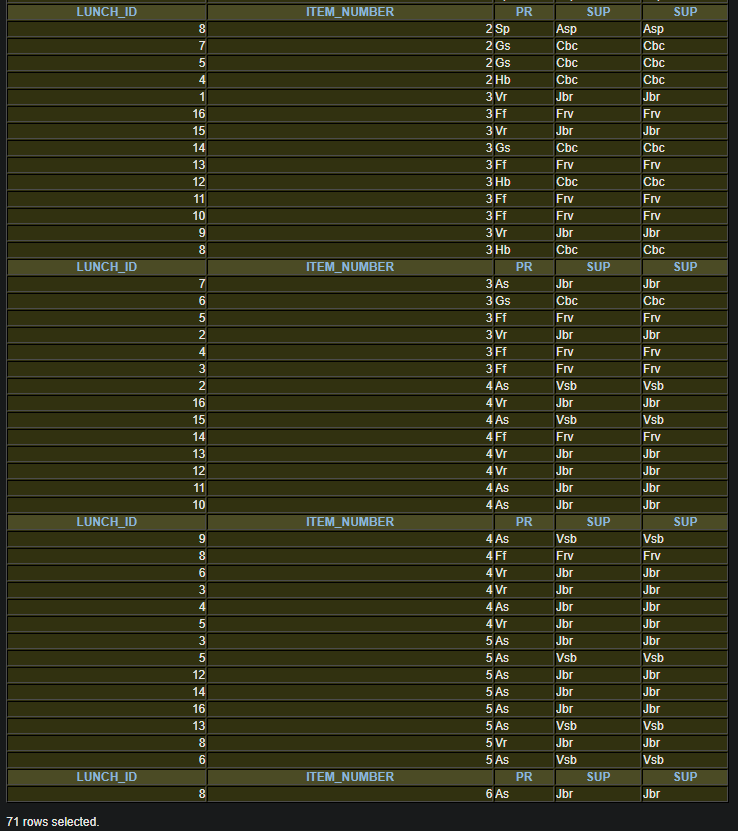
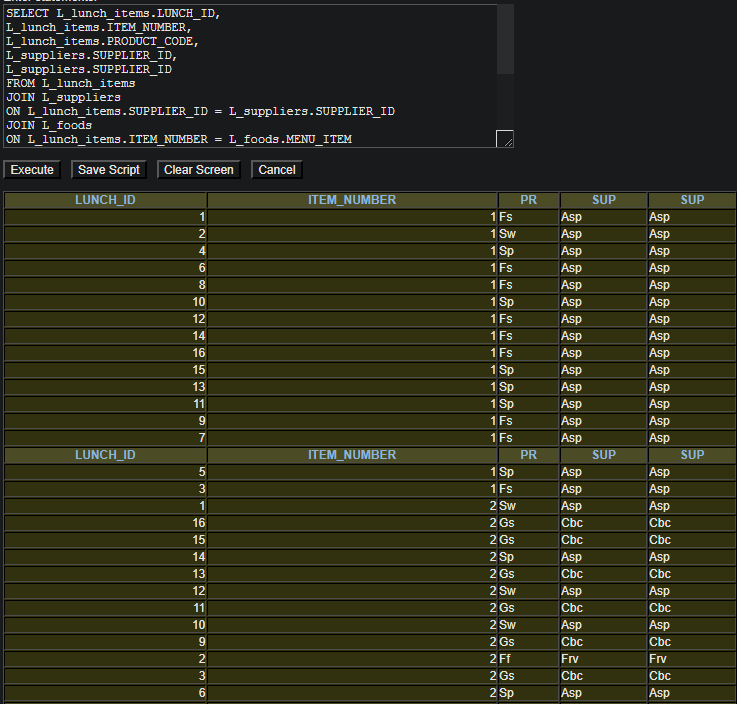
FROM L\_lunch\_items

JOIN L\_suppliers

ON L\_lunch\_items.SUPPLIER\_ID = L\_suppliers.SUPPLIER\_ID

JOIN L\_foods

ON L\_lunch\_items.ITEM\_NUMBER = L\_foods.MENU\_ITEM



1. Equi join the L\_lunches table with the L-lunch\_items table with the L\_suppliers table, with the L-foods table. Show these columns in this order: employee\_id, lunch\_id, lunch\_date, item\_number, product code, supplier\_id, supplier\_name. Arrange the rows in employee id order. [71 rows]

SELECT L\_lunches.EMPLOYEE\_ID,

L\_lunches.LUNCH\_ID,

L\_lunches.LUNCH\_DATE,

L\_lunch\_items.ITEM\_NUMBER,

L\_lunch\_items.PRODUCT\_CODE,

L\_suppliers.SUPPLIER\_ID,

L\_suppliers.SUPPLIER\_NAME

FROM L\_lunches

JOIN L\_lunch\_items

ON L\_lunches.LUNCH\_ID = L\_lunch\_items.LUNCH\_ID

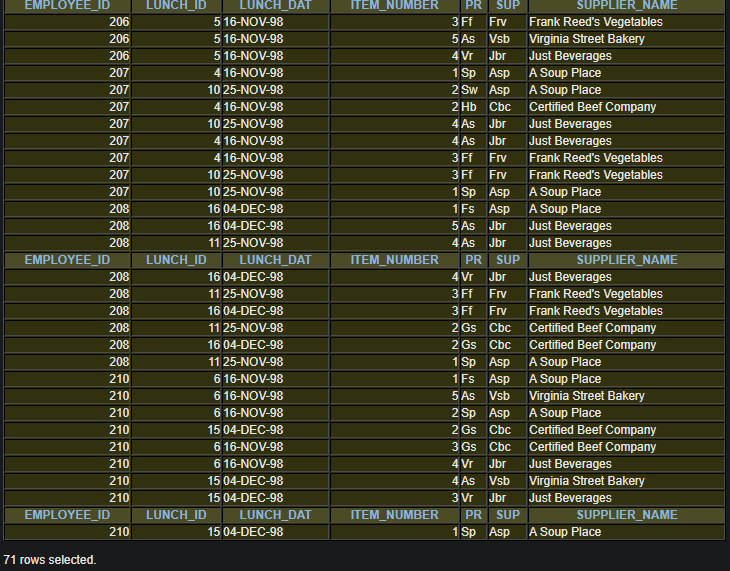
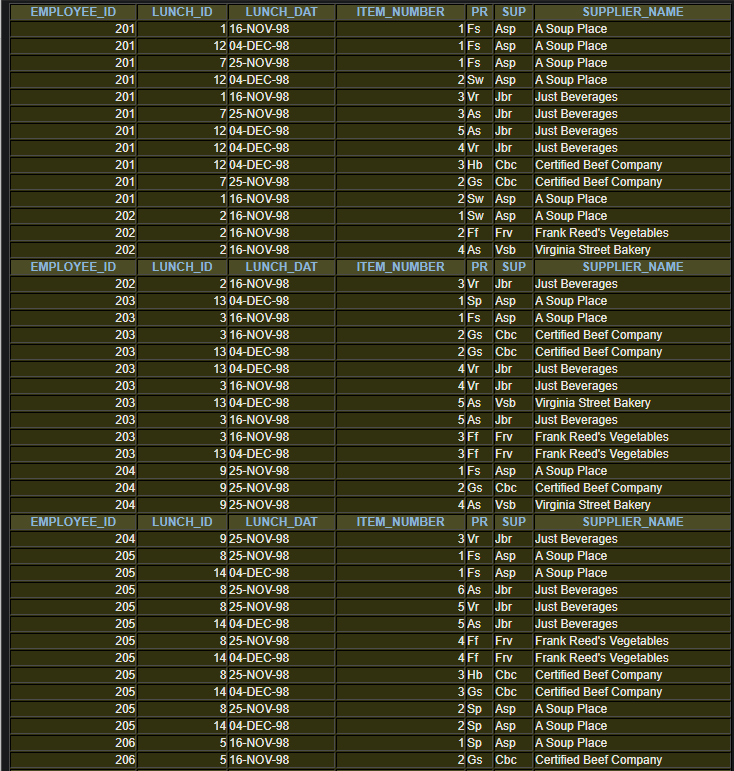
JOIN L\_suppliers

ON L\_lunch\_items.SUPPLIER\_ID = L\_suppliers.SUPPLIER\_ID

JOIN L\_foods

ON L\_lunch\_items.ITEM\_NUMBER = L\_foods.MENU\_ITEM

ORDER BY EMPLOYEE\_ID



1. Equi join the L-employees table with the L\_lunches table with the L-lunch\_items table with the L\_suppliers table, with the L-foods table. Show these columns in this order: employee\_id, employee\_name, dept\_code, lunch\_id, lunch\_date, item\_number, product code, supplier\_id, supplier\_name. Arrange the rows in dept\_code, employee name order, lunch\_date order. [71 rows]

SELECT L\_employees.EMPLOYEE\_ID,

L\_employees.FIRST\_NAME,

L\_employees.LAST\_NAME,

L\_employees.DEPT\_CODE,

L\_lunches.LUNCH\_ID,

L\_lunches.LUNCH\_DATE,

L\_lunch\_items.ITEM\_NUMBER,

L\_lunch\_items.PRODUCT\_CODE,

L\_suppliers.SUPPLIER\_ID,

L\_suppliers.SUPPLIER\_NAME

FROM L\_employees

JOIN L\_lunches

ON L\_employees.EMPLOYEE\_ID = L\_lunches.EMPLOYEE\_ID

JOIN L\_lunch\_items

ON L\_lunches.LUNCH\_ID = L\_lunch\_items.LUNCH\_ID

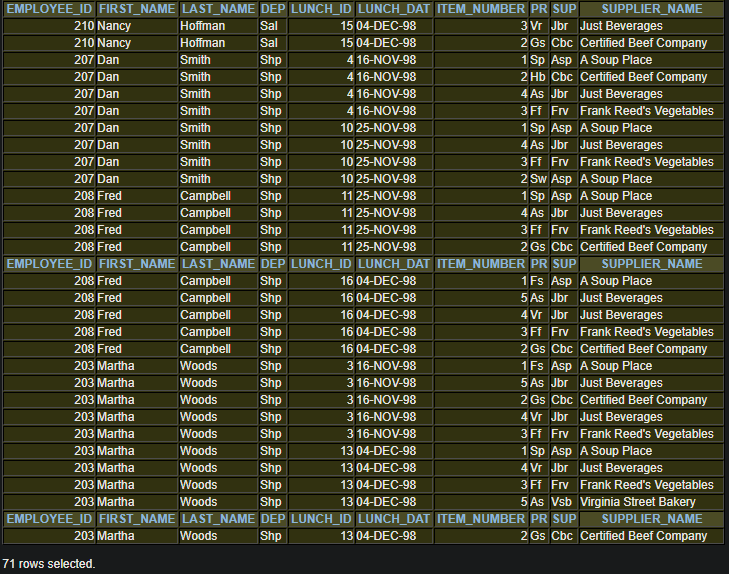
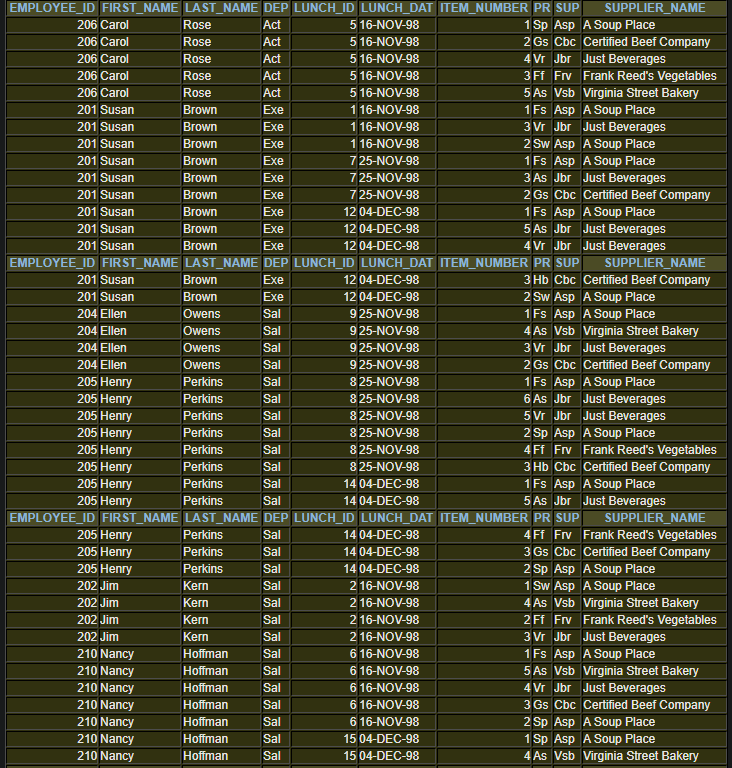
JOIN L\_suppliers

ON L\_lunch\_items.SUPPLIER\_ID = L\_suppliers.SUPPLIER\_ID

JOIN L\_foods

ON L\_lunch\_items.ITEM\_NUMBER = L\_foods.MENU\_ITEM

ORDER BY DEPT\_CODE, FIRST\_NAME, LAST\_NAME, LUNCH\_DATE



1. Which of our employees, who work in either the Sales or Shipping departments, dined in the cafeteria. Show the employee name, dept code, and each of the dates when they used the cafeteria. [12 rows]

SELECT FIRST\_NAME, LAST\_NAME, DEPT\_CODE, LUNCH\_DATE

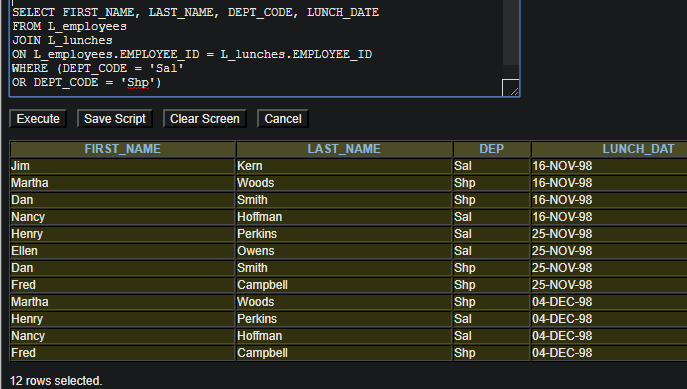
FROM L\_employees

JOIN L\_lunches

ON L\_employees.EMPLOYEE\_ID = L\_lunches.EMPLOYEE\_ID

AND (DEPT\_CODE = 'Sal'

OR DEPT\_CODE = 'Shp')



1. Which of our employees ordered coffee with their meal (or ordered coffee as a single item) from the cafeteria. Show the employee\_id, name, lunch\_id, product code and description. [11 rows]

SELECT L\_employees.EMPLOYEE\_ID,

L\_employees.FIRST\_NAME,

L\_employees.LAST\_NAME,

L\_lunches.LUNCH\_ID,

L\_lunch\_items.PRODUCT\_CODE,

L\_foods.DESCRIPTION

FROM L\_employees

JOIN L\_lunches

ON L\_employees.EMPLOYEE\_ID = L\_lunches.EMPLOYEE\_ID

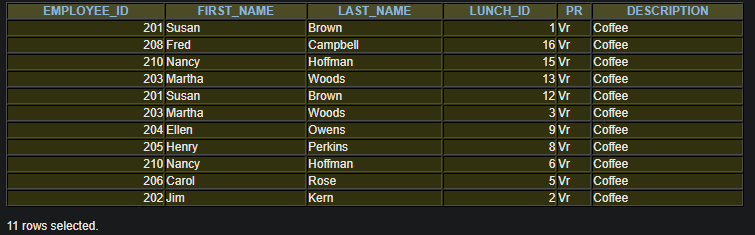
JOIN L\_lunch\_items

ON L\_lunches.LUNCH\_ID = L\_lunch\_items.LUNCH\_ID

JOIN L\_foods

ON L\_lunch\_items.PRODUCT\_CODE = L\_foods.PRODUCT\_CODE

AND L\_foods.PRODUCT\_CODE = 'Vr'



1. Equi join the l\_employees table with the l\_employees table. This is a recursive join that links employees with their managers. Show the workers employee\_id, first\_name, and last\_name, along with the manager’s employee\_id, first-name, and last\_name.[11 rows]

SELECT e.EMPLOYEE\_ID,

e.FIRST\_NAME,

e.LAST\_NAME,

m.EMPLOYEE\_ID,

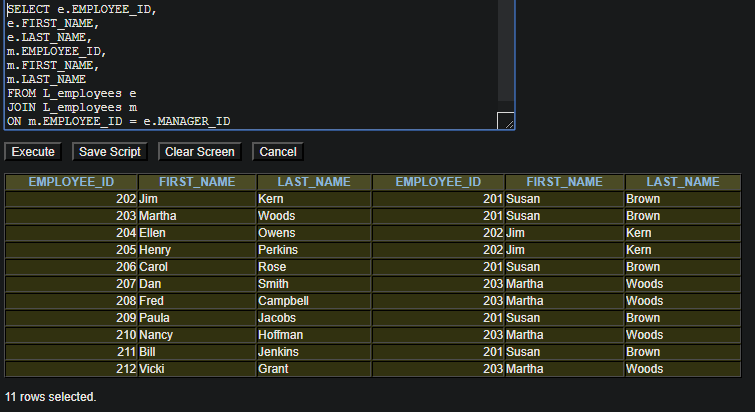
m.FIRST\_NAME,

m.LAST\_NAME

FROM L\_employees e

JOIN L\_employees m

ON m.EMPLOYEE\_ID = e.MANAGER\_ID



1. Same as the previous question, but be sure that every employee is included in the report. [12 rows]

SELECT e.EMPLOYEE\_ID,

e.FIRST\_NAME,

e.LAST\_NAME,

m.EMPLOYEE\_ID,

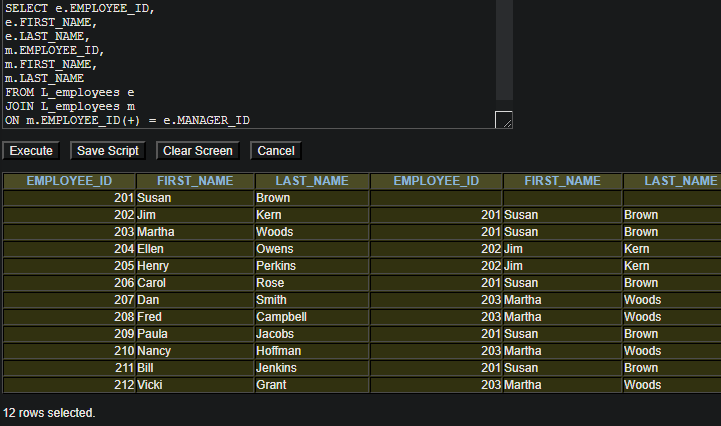
m.FIRST\_NAME,

m.LAST\_NAME

FROM L\_employees e

JOIN L\_employees m

ON m.EMPLOYEE\_ID(+) = e.MANAGER\_ID



1. Prepare a report showing employee id and name information, along with their manager’s id and name information, for those employees whose manager works in the same department as they do. [4 rows]

SELECT e.EMPLOYEE\_ID,

e.FIRST\_NAME,

e.LAST\_NAME,

m.EMPLOYEE\_ID,

m.FIRST\_NAME,

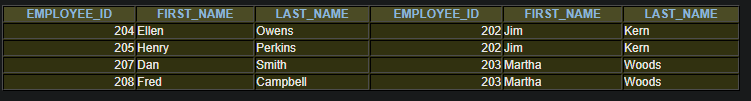
m.LAST\_NAME

FROM L\_employees e

JOIN L\_employees m

ON m.EMPLOYEE\_ID = e.MANAGER\_ID

AND e.DEPT\_CODE = m.DEPT\_CODE



1. Prepare a report showing employee id and name information, along with their manager’s id and name information, but only show this information for those employees whose manager does NOT work in the same department as they do. [7 rows]

SELECT e.EMPLOYEE\_ID,

e.FIRST\_NAME,

e.LAST\_NAME,

m.EMPLOYEE\_ID,

m.FIRST\_NAME,

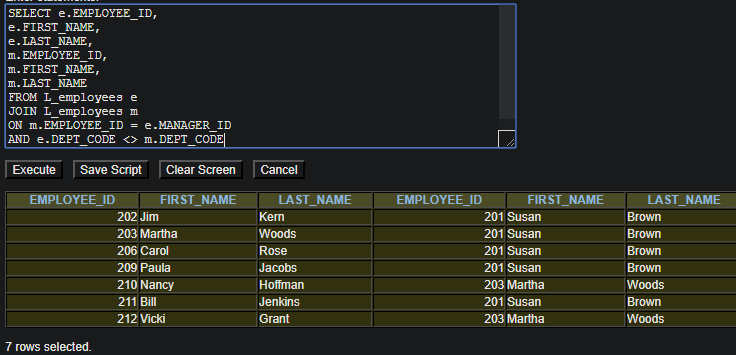
m.LAST\_NAME

FROM L\_employees e

JOIN L\_employees m

ON m.EMPLOYEE\_ID = e.MANAGER\_ID

AND e.DEPT\_CODE <> m.DEPT\_CODE



## Use the MySQL server for questions 13 thru 15.

1. Which of our employees, who work in either the Sales or Shipping departments, dined in the cafeteria. Show the employee name, dept code, and each of the dates when they used the cafeteria. [12 rows]

SELECT FIRST\_NAME, LAST\_NAME, DEPT\_CODE, LUNCH\_DATE

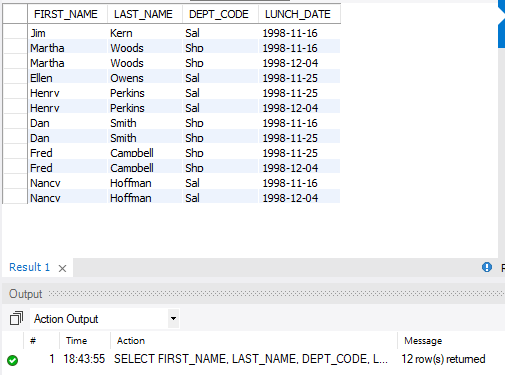
FROM L\_employees

JOIN L\_lunches

ON L\_employees.EMPLOYEE\_ID = L\_lunches.EMPLOYEE\_ID

AND (DEPT\_CODE = 'Sal'

OR DEPT\_CODE = 'Shp')



1. Which of our employees ordered coffee with their meal (or ordered coffee as a single item) from the cafeteria. Show the employee\_id, name, lunch\_id, product code and description. [11 rows]

SELECT L\_employees.EMPLOYEE\_ID,

L\_employees.FIRST\_NAME,

L\_employees.LAST\_NAME,

L\_lunches.LUNCH\_ID,

L\_lunch\_items.PRODUCT\_CODE,

L\_foods.DESCRIPTION

FROM L\_employees

JOIN L\_lunches

ON L\_employees.EMPLOYEE\_ID = L\_lunches.EMPLOYEE\_ID

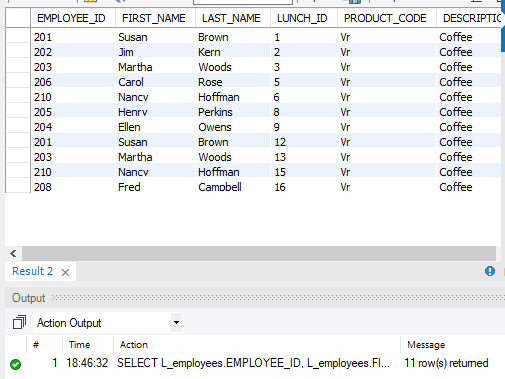
JOIN L\_lunch\_items

ON L\_lunches.LUNCH\_ID = L\_lunch\_items.LUNCH\_ID

JOIN L\_foods

ON L\_lunch\_items.PRODUCT\_CODE = L\_foods.PRODUCT\_CODE

AND L\_foods.PRODUCT\_CODE = 'Vr'



1. Equi join the l\_employees table with the l\_employees table. This is a recursive join that links employees with their managers. Show the workers employee\_id, first\_name, and last\_name, along with the manager’s employee\_id, first-name, and last\_name. [11 rows]

SELECT e.EMPLOYEE\_ID,

e.FIRST\_NAME,

e.LAST\_NAME,

m.EMPLOYEE\_ID,

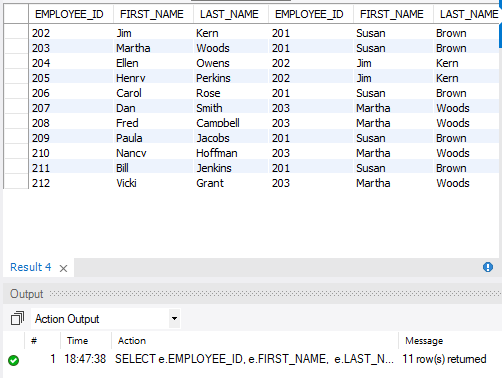
m.FIRST\_NAME,

m.LAST\_NAME

FROM L\_employees e

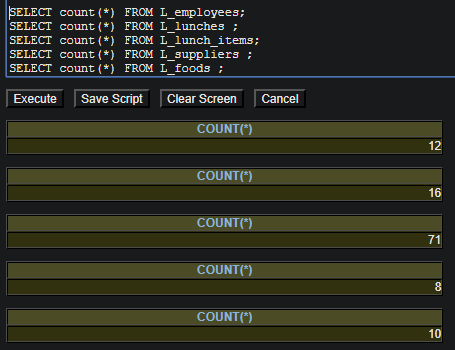
JOIN L\_employees m

ON m.EMPLOYEE\_ID = e.MANAGER\_ID



1. ***Analysis***. For question 6, the number of rows in the result set should be 71. What is the maximum number of rows we could see for a question like this? Why (explain your rationale)?

According to Module 9, Slide 23, We should no expect to see more rows in the result table than the maximum number of rows in the largest table. The largest table is the L\_lunch\_items table (71 rows).



1. ***Analysis***. Questions 9 and 10 involve a recursive relationship on the employees table. Why is the value for the number of rows returned different between these two result sets? Be sure to specifically address why there are fewer rows in the result set for question 9.

The results are different between question 9 and 10 because one is taking into consideration all rows, while the other is only looking at the condition statement.

Question 9 as 1 less row because the row for Susan Brown, does not meet the condition in the where statement, while question 10 uses the (+) operator to create a FULL OUTER JOIN, which insures all rows in each table are represented in the results table. (module 9, slide 50)