SQL Programming – Level 2 Programming Project 04

# 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 WHERE clause to form the joins, that is do NOT use the FROM Clause to form any of these joins).***

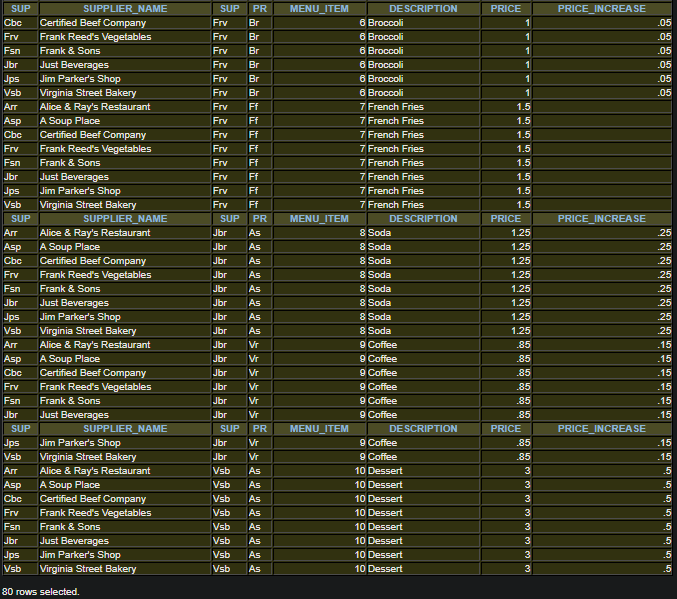
## Use the Oracle 9i server for questions 1 thru 12.

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

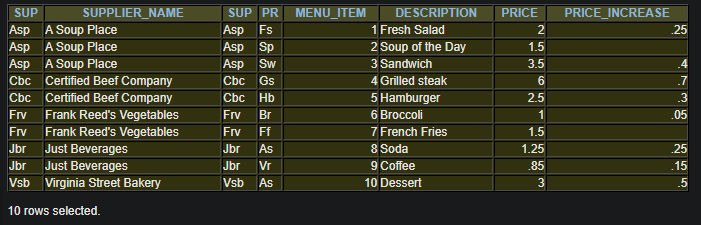




1. Equi join the suppliers table (L\_suppliers) with the foods (L\_foods) table. Show all columns. [10 rows]

SELECT \* FROM L\_suppliers, L\_foods

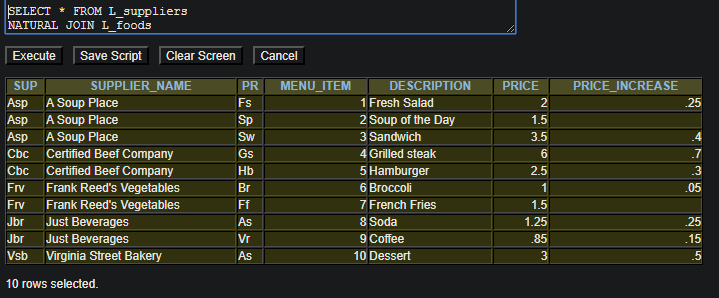
WHERE L\_suppliers.SUPPLIER\_ID = L\_foods.SUPPLIER\_ID



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,

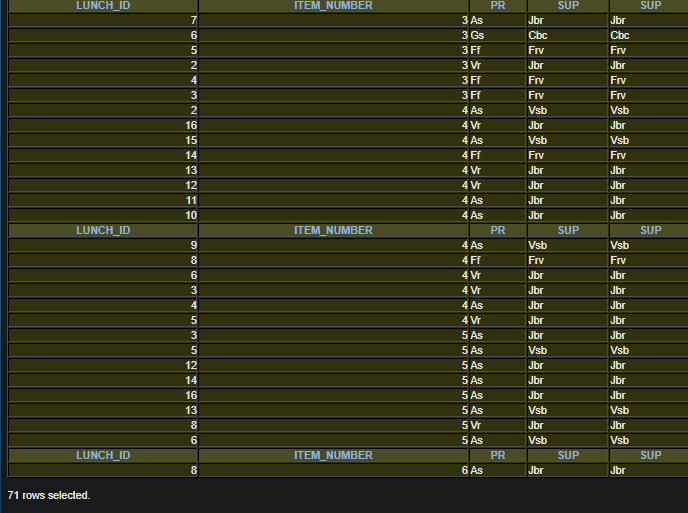
L\_suppliers.SUPPLIER\_ID

FROM L\_lunch\_items, L\_suppliers, L\_foods

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

AND 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

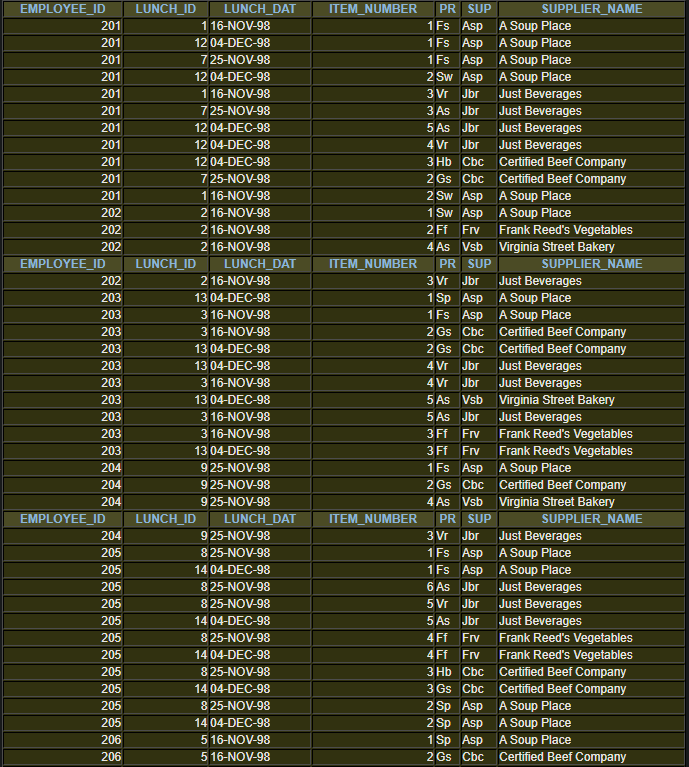
FROM L\_lunches, L\_lunch\_items, L\_suppliers, L\_foods

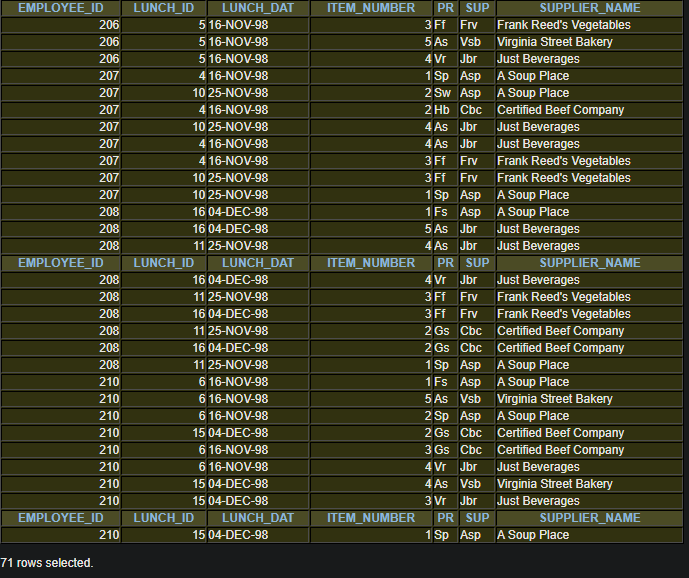
WHERE L\_lunches.LUNCH\_ID = L\_lunch\_items.LUNCH\_ID

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

AND 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, L\_lunches, L\_lunch\_items, L\_suppliers, L\_foods

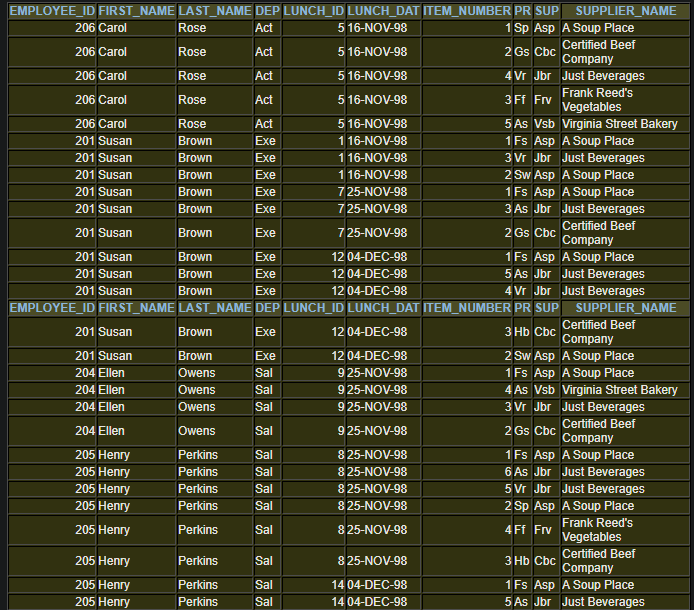
WHERE L\_employees.EMPLOYEE\_ID = L\_lunches.EMPLOYEE\_ID

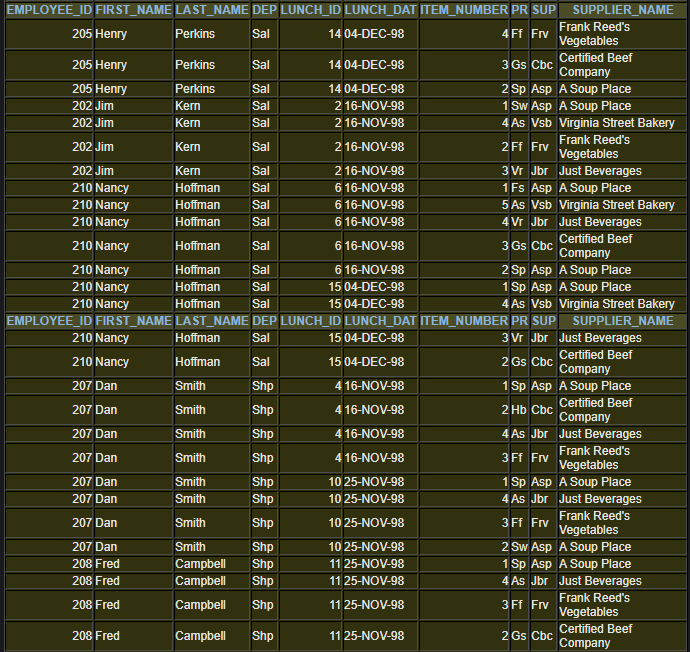
AND L\_lunches.LUNCH\_ID = L\_lunch\_items.LUNCH\_ID

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

AND 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]

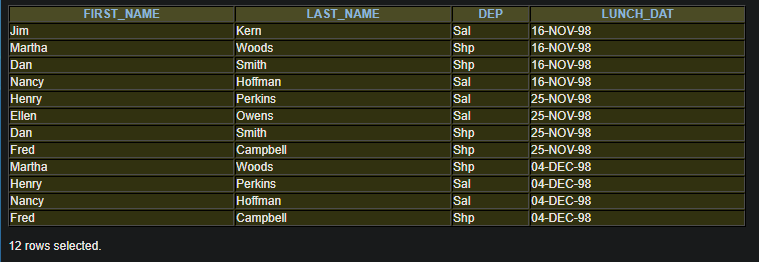
SELECT FIRST\_NAME, LAST\_NAME, DEPT\_CODE, LUNCH\_DATE

FROM L\_employees, L\_lunches

WHERE (DEPT\_CODE = 'Sal'

OR DEPT\_CODE = 'Shp')

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



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

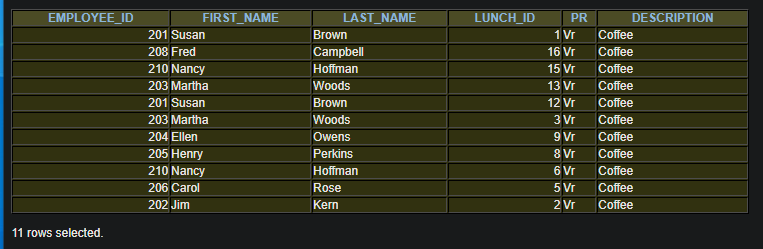
FROM L\_employees,L\_lunches, L\_lunch\_items,L\_foods

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

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

AND 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,

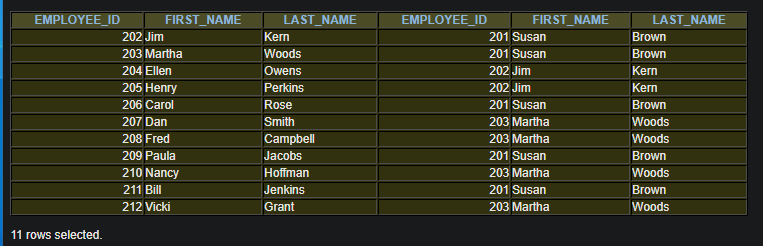
m.EMPLOYEE\_ID,

m.FIRST\_NAME,

m.LAST\_NAME

FROM L\_employees e, L\_employees m

WHERE 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,

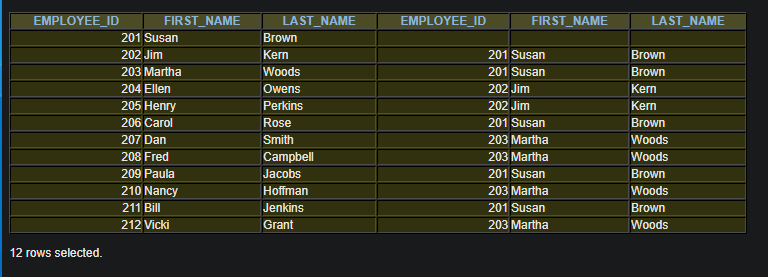
m.EMPLOYEE\_ID,

m.FIRST\_NAME,

m.LAST\_NAME

FROM L\_employees e, L\_employees m

WHERE 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,

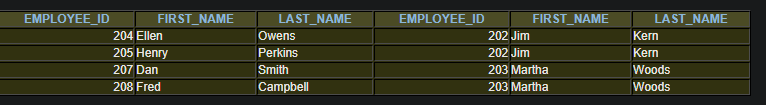
m.FIRST\_NAME,

m.LAST\_NAME

FROM L\_employees e, L\_employees m

WHERE 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,

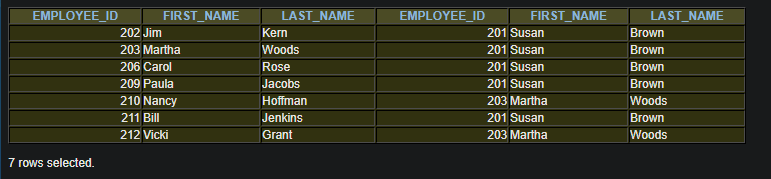
m.FIRST\_NAME,

m.LAST\_NAME

FROM L\_employees e, L\_employees m

WHERE 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]

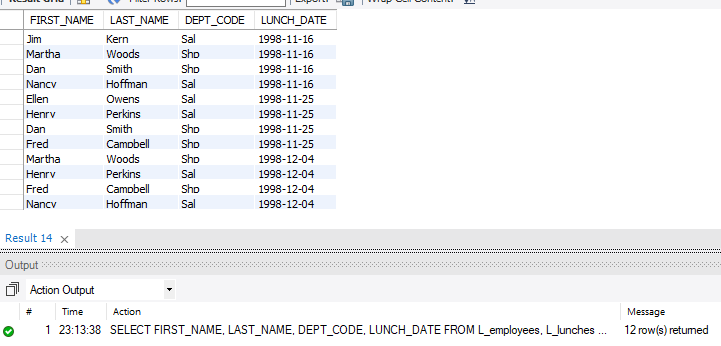
SELECT FIRST\_NAME, LAST\_NAME, DEPT\_CODE, LUNCH\_DATE

FROM L\_employees, L\_lunches

WHERE (DEPT\_CODE = 'Sal'

OR DEPT\_CODE = 'Shp')

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



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

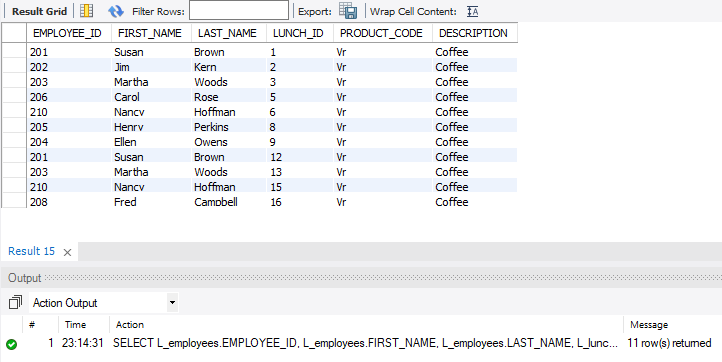
FROM L\_employees,L\_lunches, L\_lunch\_items,L\_foods

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

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

AND 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, L\_employees m

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

