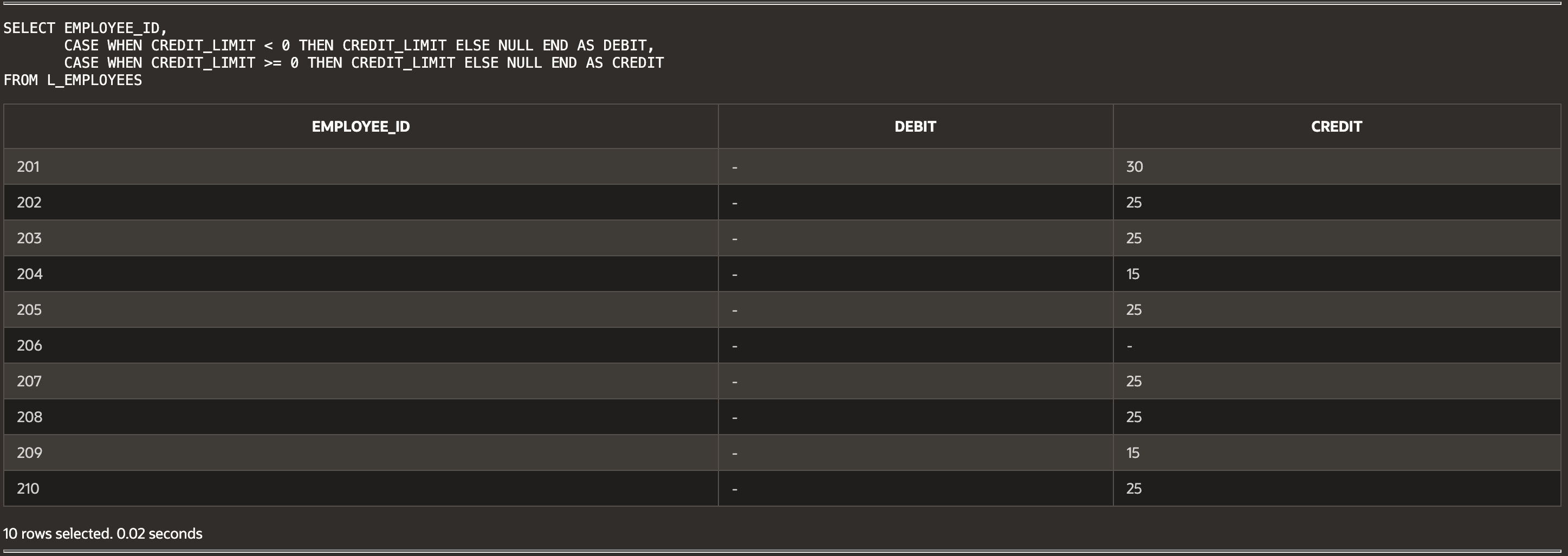
**Week 5 Guided Practice: Union All, Cross Joins and CrossTab Queries.**

The following questions come from the Task examples of each section of chapter 15 & 16 in your textbook.

After you are finished, please submit the Microsoft Word file that contains screenshots of the SQL script and the resulting tables. Your document should be named **W5\_GP\_Unions2\_Lastname.docx**.

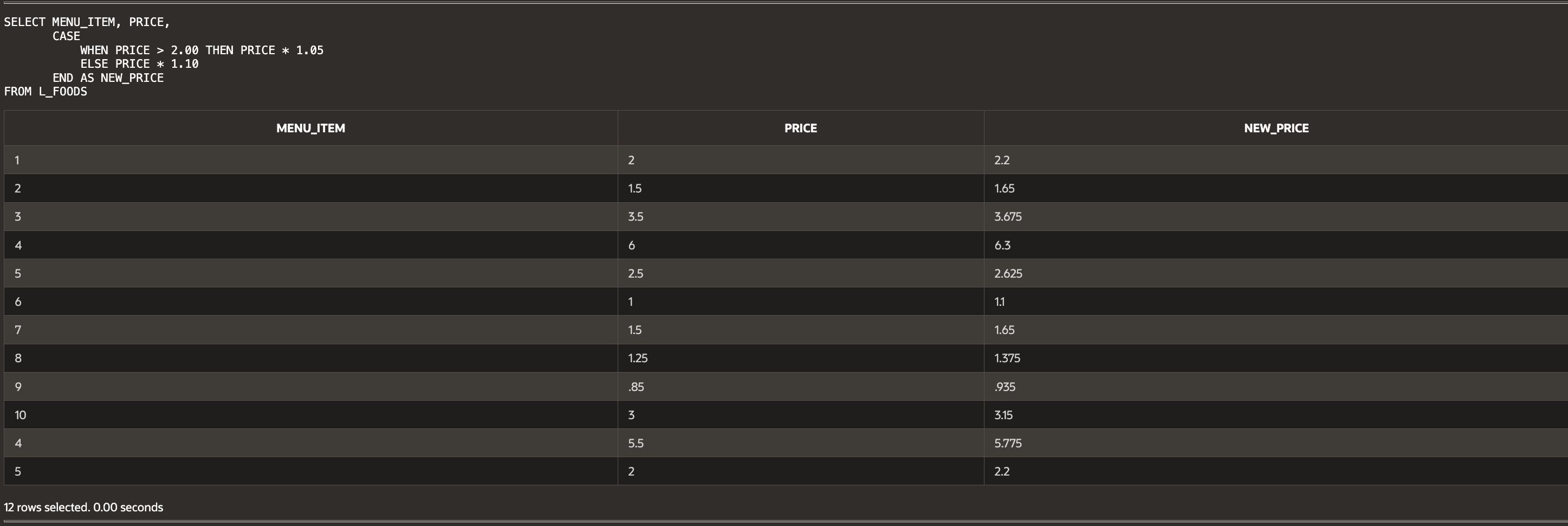
(15-12) Question 1:

Divide the cost column from the beginning table into two columns: *debits* and *credits*.



(15-13) Question 2:

Show how to make two different calculations, depending on the data in a row. Increase the price of all foods costing more than $2.00 by 5 percent. Increase the price of all other foods by 10 percent. Ignore the existing *price\_increase* column.



(15-14) Question 3:

List the letters from *'A'* to *'G'*. Do this as a *union* of seven tables.



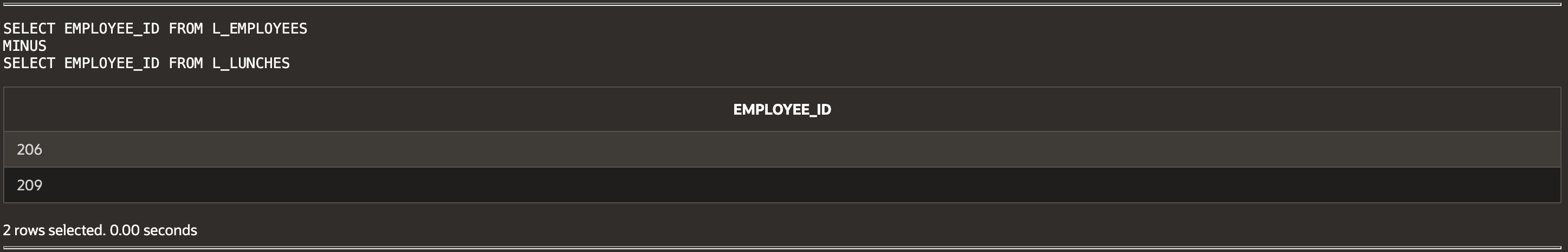
(15-15) Question 4:

Find the intersection of two tables. That is, find all the rows that occur in both tables.



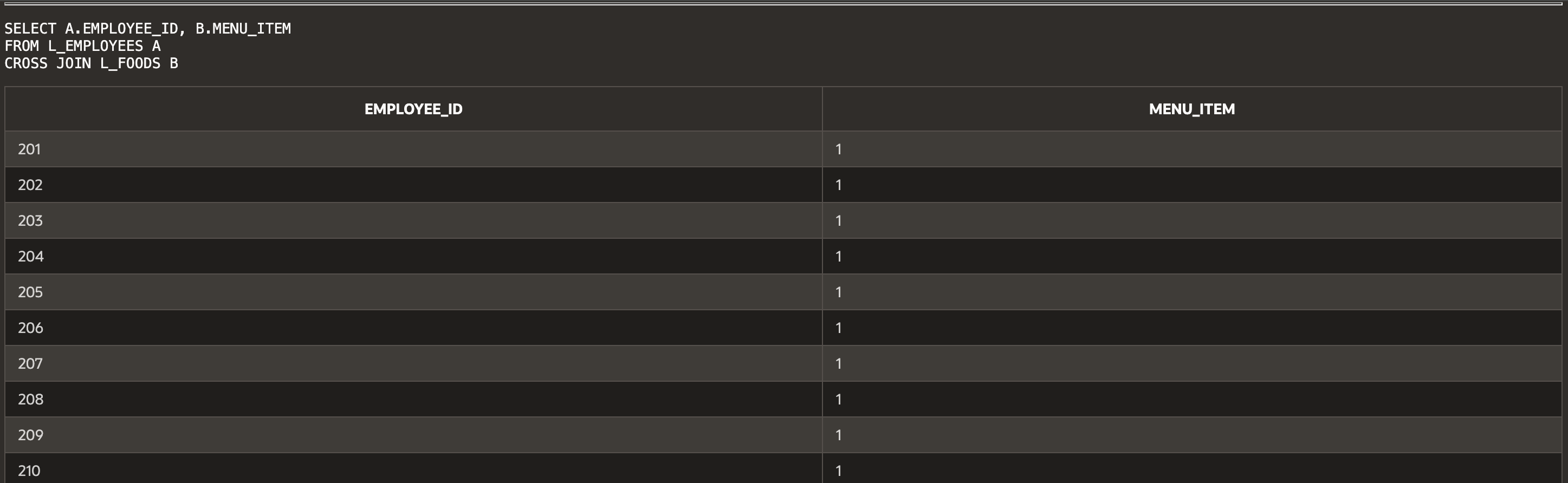
(15-16) Question 5:

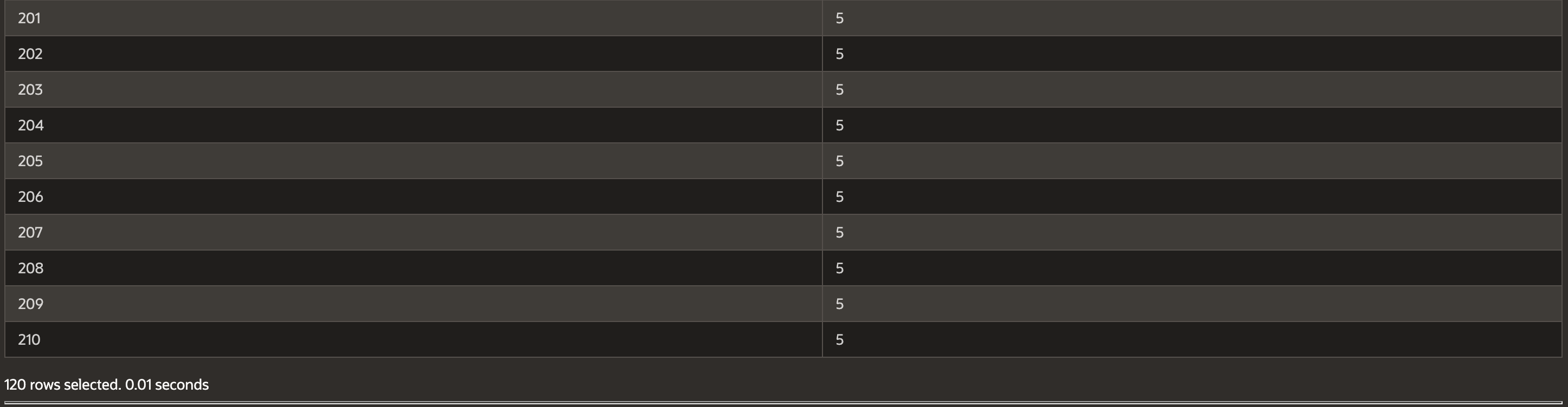
Find all the rows that are in one table and not in the other table. Do this both ways to find all the differences between the two tables.



(16-1) Question 6:

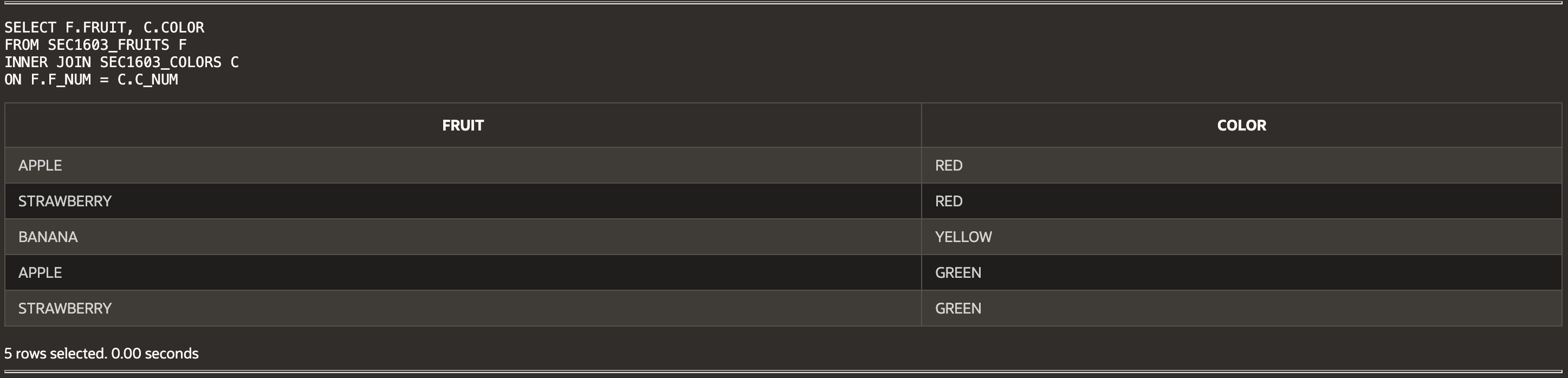
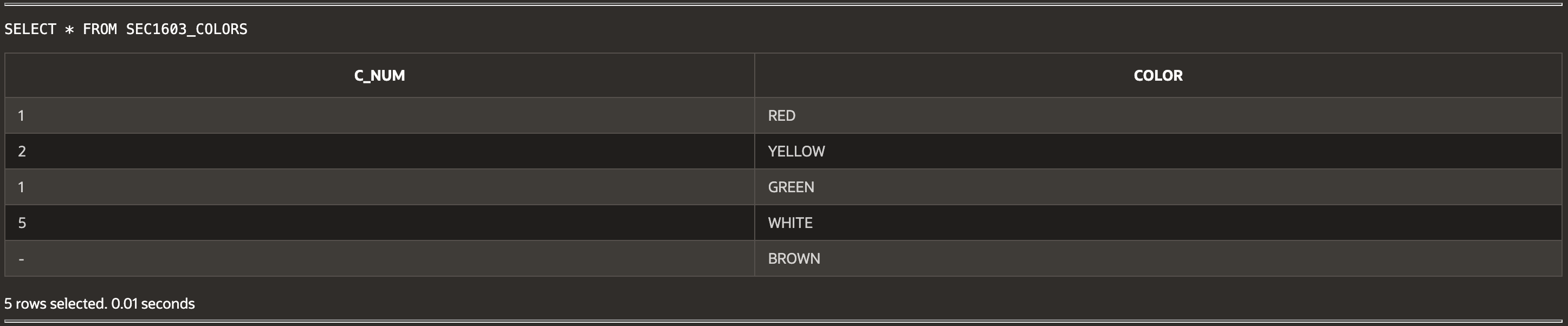
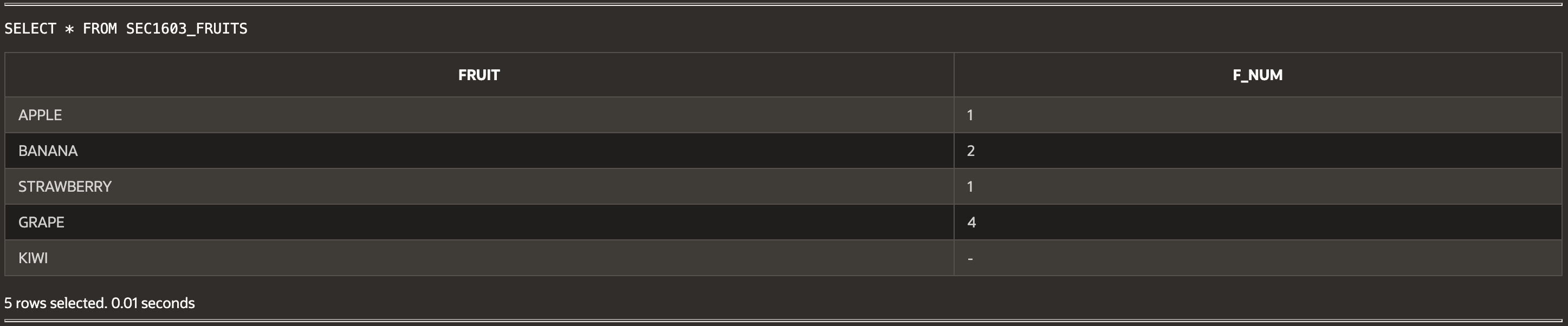
Show an example of a cross join.

...



(16-3) Question 7:

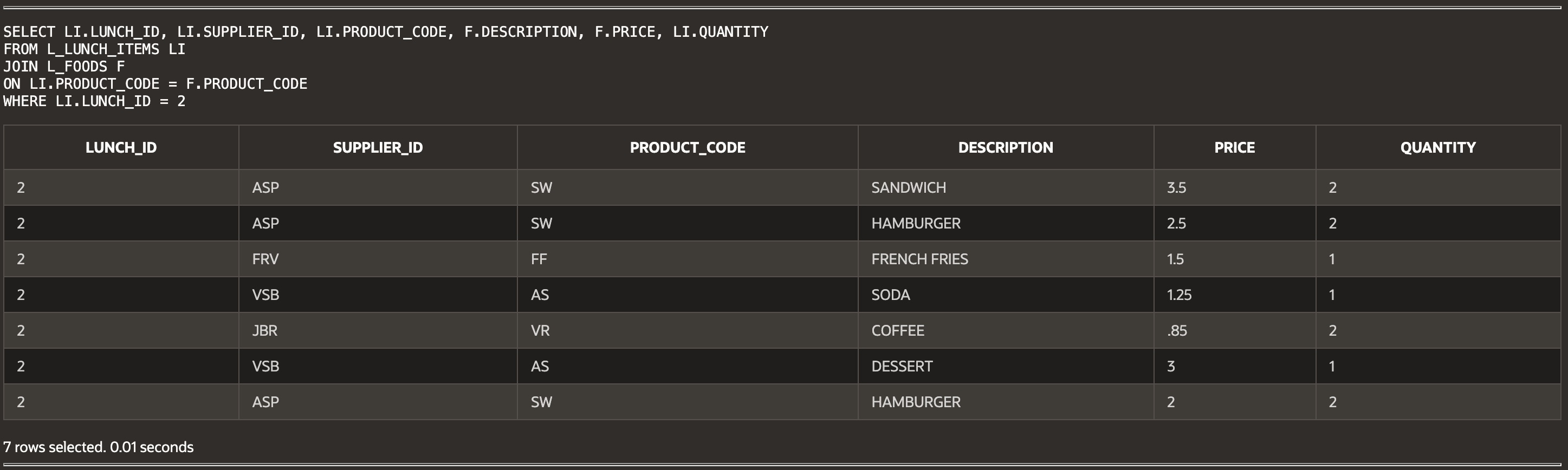
Show each step of the process to create an inner join of the *sec1603\_fruits* and *sec1603\_colors* tables. Use the join condition *f\_num = c\_num*.



(16-5) Question 8:

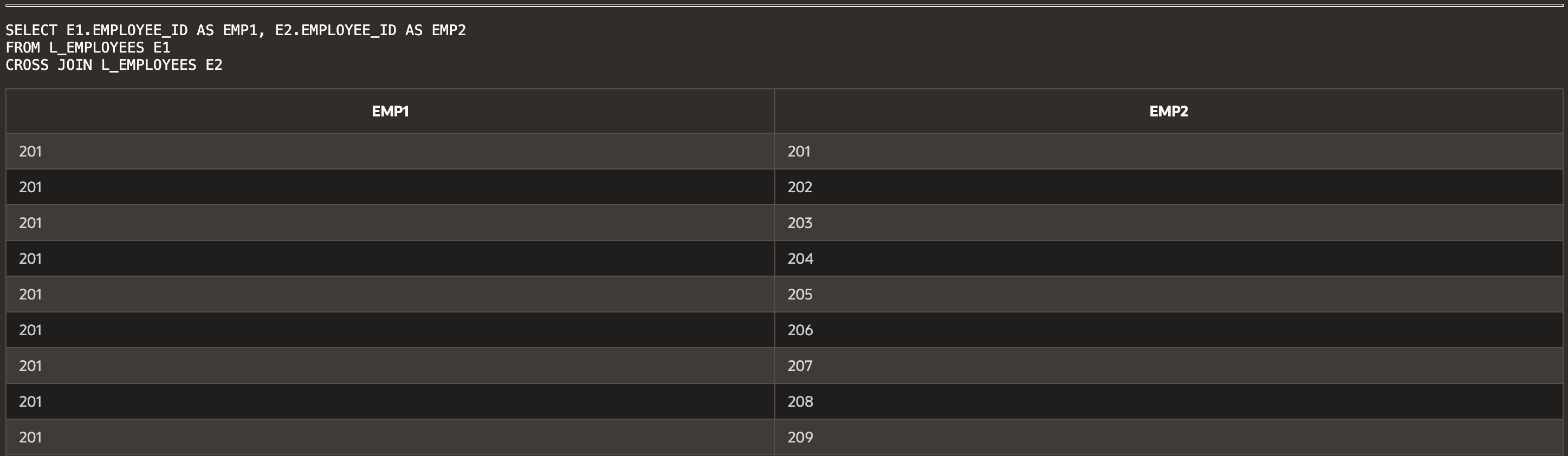
Show an example of SQL that contains an error. Leave one of the join conditions out of the *where* clause. Show how we might detect this error.

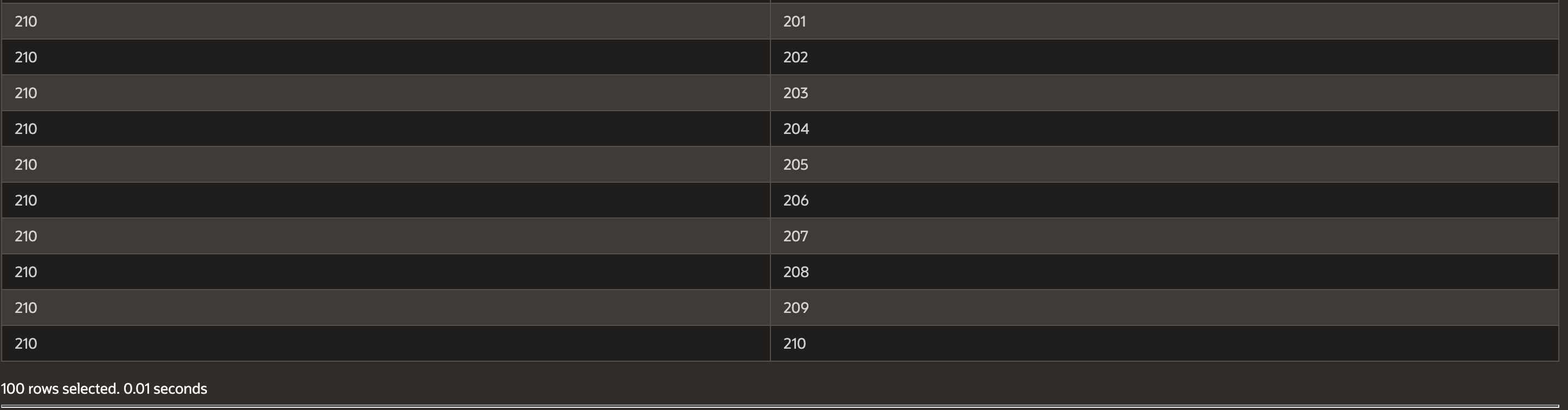
For lunch 2, list the *lunch\_id, supplier\_id, product\_code, description, price*, and *quantity* columns. Use the *L\_FOODS* table and the *L\_LUNCH\_ITEMS* table.



(16-9) Question 9:

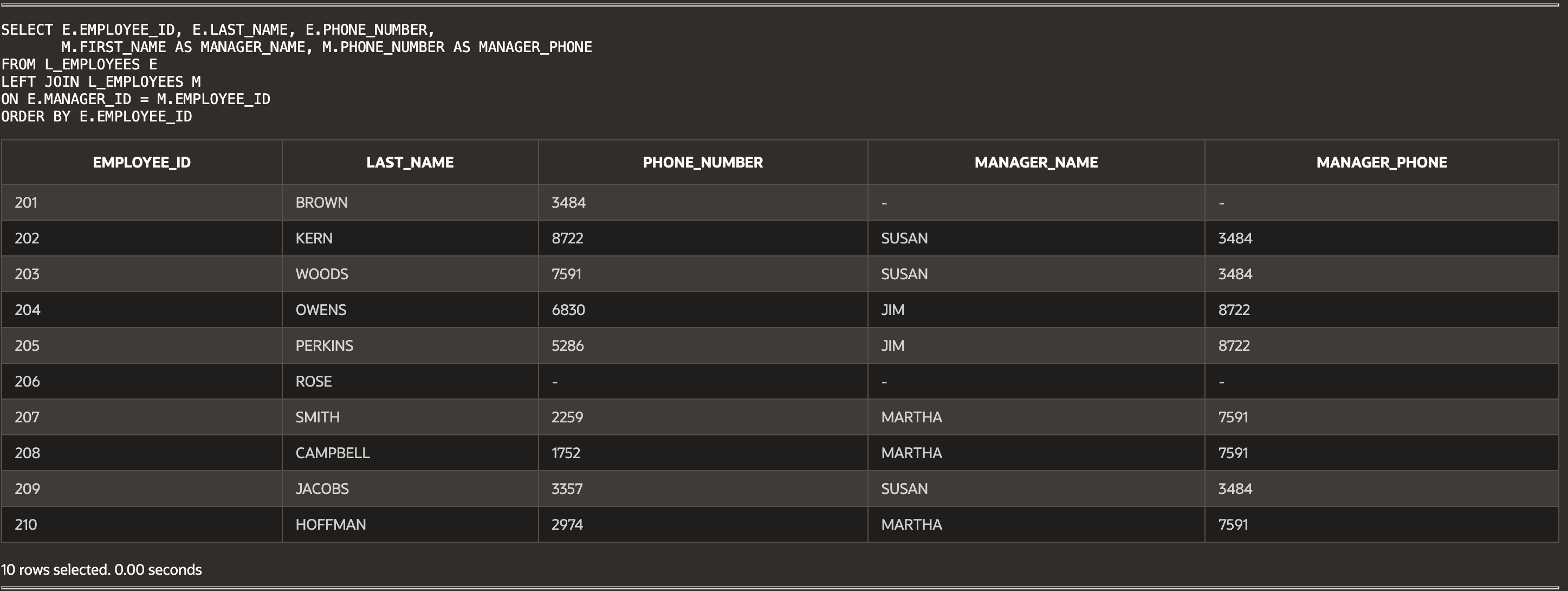
Form a cross join of a table with itself.

...



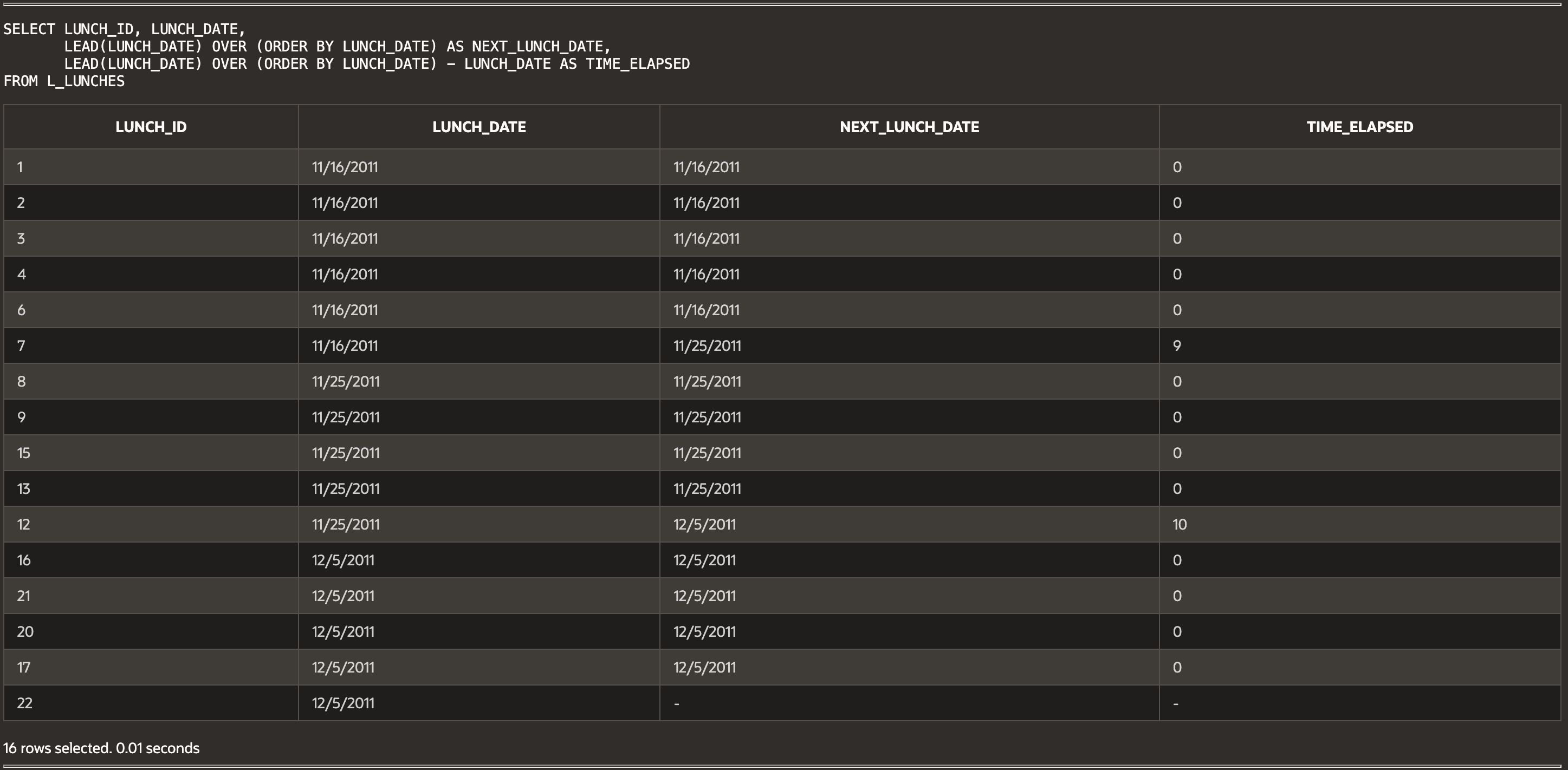
(16-10) Question 10:

From the *L\_EMPLOYEES* table, list the employee ID, last name, and phone number of each employee with the name and phone number of his or her manager. Include a row for each employee, even those who do not have a manager. Sort the rows by the *employee\_id* column.



(16-11) Question 11:

Find the amount of time that has elapsed between one event and the next event.



(16-12) Questions 12:

Create a table with all the numbers from 0 to 999. First, create a table of the numbers from 0 to 9. Then cross join it with itself.

We already have a table named *numbers\_0\_to\_9* containing all the digits. We want to leave that table alone. We create a new table with a slightly different name so I can show you how this all works.



**Script:**

-- Haley Archer

-- (15-12) Question 1:

-- Divide the cost column into two columns: debits and credits.

-- FIX: Since TRANSACTIONS\_TABLE does not exist, using L\_EMPLOYEES (CREDIT\_LIMIT as cost).

SELECT EMPLOYEE\_ID,

CASE WHEN CREDIT\_LIMIT < 0 THEN CREDIT\_LIMIT ELSE NULL END AS DEBIT,

CASE WHEN CREDIT\_LIMIT >= 0 THEN CREDIT\_LIMIT ELSE NULL END AS CREDIT

FROM L\_EMPLOYEES;

-- (15-13) Question 2:

-- Increase the price of all foods costing more than $2.00 by 5%.

-- Increase the price of all other foods by 10%.

SELECT MENU\_ITEM, PRICE,

CASE

WHEN PRICE > 2.00 THEN PRICE \* 1.05

ELSE PRICE \* 1.10

END AS NEW\_PRICE

FROM L\_FOODS;

-- (15-14) Question 3:

-- List the letters from 'A' to 'G' using a UNION.

SELECT 'A' AS LETTER FROM DUAL

UNION

SELECT 'B' FROM DUAL

UNION

SELECT 'C' FROM DUAL

UNION

SELECT 'D' FROM DUAL

UNION

SELECT 'E' FROM DUAL

UNION

SELECT 'F' FROM DUAL

UNION

SELECT 'G' FROM DUAL;

-- (15-15) Question 4:

-- Find the intersection of two tables (common rows in both tables).

-- FIX: TABLE\_A and TABLE\_B do not exist. Using L\_EMPLOYEES and L\_LUNCHES as examples.

SELECT EMPLOYEE\_ID FROM L\_EMPLOYEES

INTERSECT

SELECT EMPLOYEE\_ID FROM L\_LUNCHES;

-- (15-16) Question 5:

-- Find all the rows that are in one table and not in the other table.

-- Do this both ways to find all the differences.

SELECT EMPLOYEE\_ID FROM L\_EMPLOYEES

MINUS

SELECT EMPLOYEE\_ID FROM L\_LUNCHES;

SELECT EMPLOYEE\_ID FROM L\_LUNCHES

MINUS

SELECT EMPLOYEE\_ID FROM L\_EMPLOYEES;

-- (16-1) Question 6:

-- Show an example of a CROSS JOIN.

SELECT A.EMPLOYEE\_ID, B.MENU\_ITEM

FROM L\_EMPLOYEES A

CROSS JOIN L\_FOODS B;

-- (16-3) Question 7:

-- Create an INNER JOIN of sec1603\_fruits and sec1603\_colors.

-- Use the join condition f\_num = c\_num.

SELECT F.FRUIT, C.COLOR

FROM SEC1603\_FRUITS F

INNER JOIN SEC1603\_COLORS C

ON F.F\_NUM = C.C\_NUM;

-- (16-5) Question 8:

-- Show an SQL error by leaving out a join condition.

-- Using the L\_FOODS and L\_LUNCH\_ITEMS tables.

SELECT LI.LUNCH\_ID, LI.SUPPLIER\_ID, LI.PRODUCT\_CODE, F.DESCRIPTION, F.PRICE, LI.QUANTITY

FROM L\_LUNCH\_ITEMS LI

JOIN L\_FOODS F

ON LI.PRODUCT\_CODE = F.PRODUCT\_CODE

-- ERROR: Missing supplier join condition

WHERE LI.LUNCH\_ID = 2;

-- (16-9) Question 9:

-- Form a CROSS JOIN of a table with itself.

SELECT E1.EMPLOYEE\_ID AS EMP1, E2.EMPLOYEE\_ID AS EMP2

FROM L\_EMPLOYEES E1

CROSS JOIN L\_EMPLOYEES E2;

-- (16-10) Question 10:

-- From L\_EMPLOYEES, list employee details with their manager’s name and phone.

SELECT E.EMPLOYEE\_ID, E.LAST\_NAME, E.PHONE\_NUMBER,

M.FIRST\_NAME AS MANAGER\_NAME, M.PHONE\_NUMBER AS MANAGER\_PHONE

FROM L\_EMPLOYEES E

LEFT JOIN L\_EMPLOYEES M

ON E.MANAGER\_ID = M.EMPLOYEE\_ID

ORDER BY E.EMPLOYEE\_ID;

-- (16-11) Question 11:

-- Find the amount of time that has elapsed between one event and the next.

-- FIX: EVENTS\_TABLE does not exist, replaced with L\_LUNCHES (LUNCH\_DATE as event timestamp).

SELECT LUNCH\_ID, LUNCH\_DATE,

LEAD(LUNCH\_DATE) OVER (ORDER BY LUNCH\_DATE) AS NEXT\_LUNCH\_DATE,

LEAD(LUNCH\_DATE) OVER (ORDER BY LUNCH\_DATE) - LUNCH\_DATE AS TIME\_ELAPSED

FROM L\_LUNCHES;

-- (16-12) Question 12:

-- Create a table with all numbers from 0 to 999 using a cross join.

-- FIX: NUM column does not exist in NUMBERS\_0\_TO\_9, replaced with DIGIT.

CREATE TABLE NUMBERS\_0\_TO\_999 AS

SELECT (T1.DIGIT \* 10 + T2.DIGIT) AS NUMBER\_VALUE

FROM NUMBERS\_0\_TO\_9 T1

CROSS JOIN NUMBERS\_0\_TO\_9 T2;