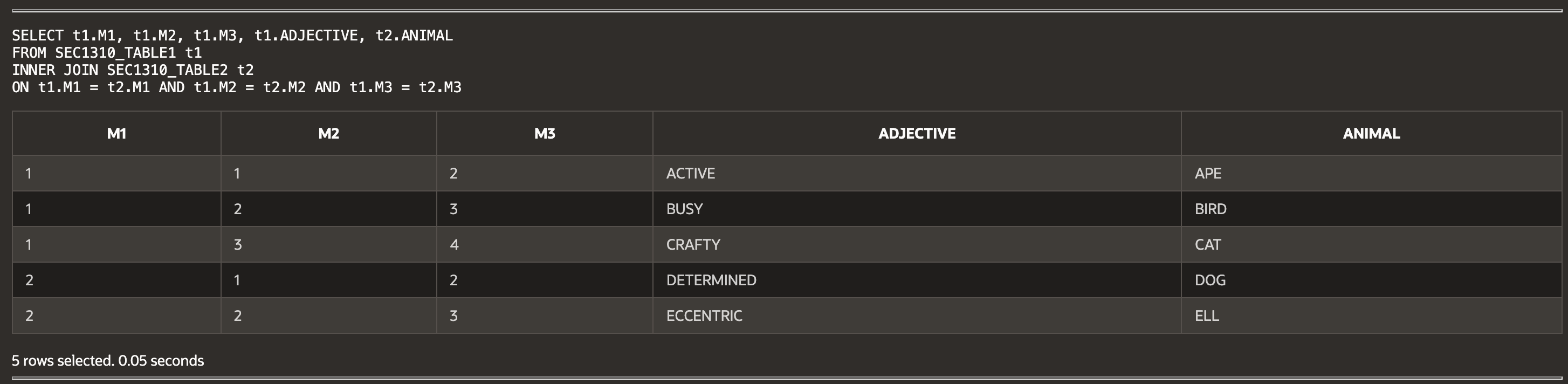
**Week 4 Performance Assessment: Inner Joins Part 2**

The following questions come from the “Check your understanding” examples of each section of chapter 13 in your textbook.

After you are finished, please submit a Microsoft Word file that contains screenshots of the SQL queries and the output, along with a comment in the query containing your name. Your document should be named **W4\_PA\_InnerJoins2\_Lastname.docx**.

(13-10) Question 1:

Write a select statement to create the inner join of tables *sec1310\_table1* and *sec1310\_table2*. The join condition should say that the first three columns of these tables are equal.



(13-11) Question 2:

A large dictionary has four volumes. Table *sec1311\_dictionary* shows the range of words that are in each volume. Table *sec1311\_words* contains some words that are in the dictionary. Write SQL to determine which volume of the dictionary contains each of these words.



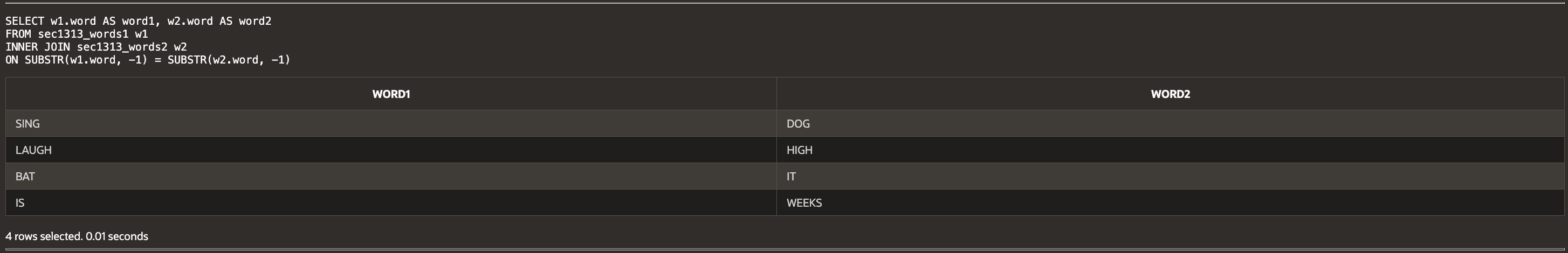
(13-12) Question 3:

List all the letters that are greater than S. Use the *alphabet* table. Put these letters in order.



(13-13) Question 4:

Tables *sec1313\_words1* and *sec1313\_words2* contains words. Join these tables together when the words end in the same letter.



(13-14) Question 5:

Change this SQL, writing the join condition in the *from* clause.

*select a.student\_name,*

*a.test\_score,*

*b.letter\_grade*

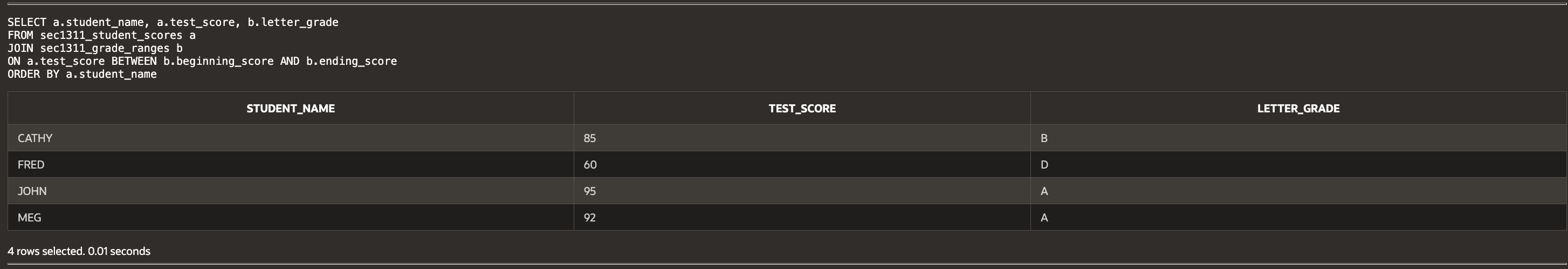
*from sec1311\_student\_scores a,*

*sec1311\_grade\_ranges b*

*where a.test\_score between b.beginning\_score*

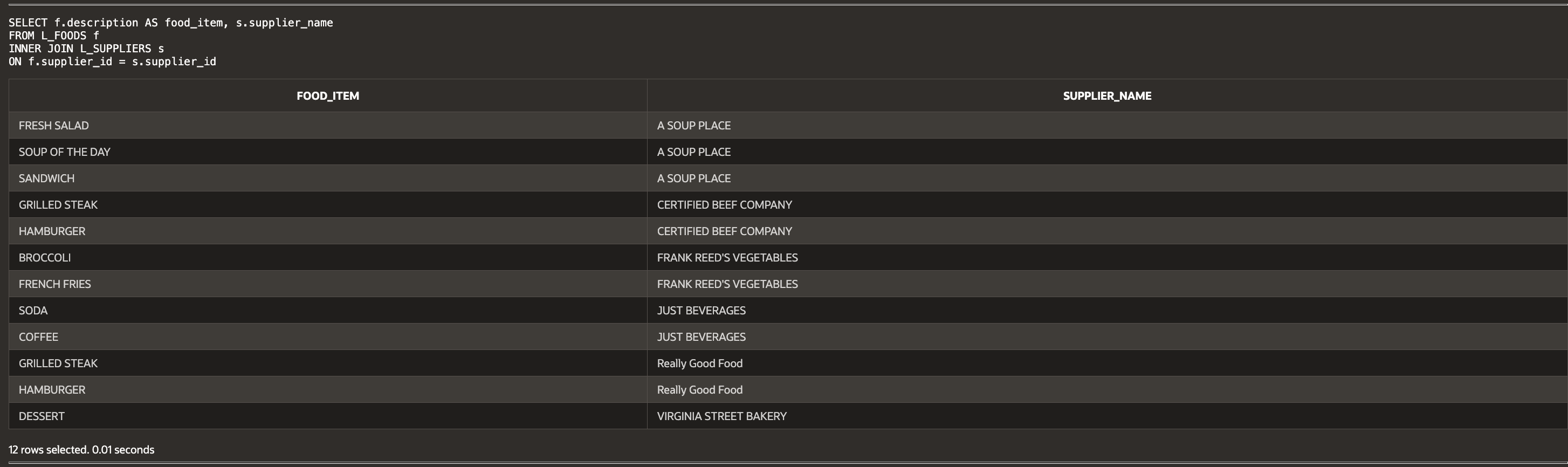
*and b.ending\_score*

*order by a.student\_name;*



(13-15) Question 6:

Write a *select* statement to list all the foods on the lunch menu and show the full name of the supplier of each food.



(13-16) Question 7:

Demonstrate that a *select* statement can be separated into two parts: The first part joins the tables and creates a new table, and the second part restricts the amount of data that is shown. For the following *select* statement, write two SQL statements to separate these two steps.

*select a.description,*

*b.supplier\_name*

*from L\_FOODS a,*

*L\_SUPPLIERS b*

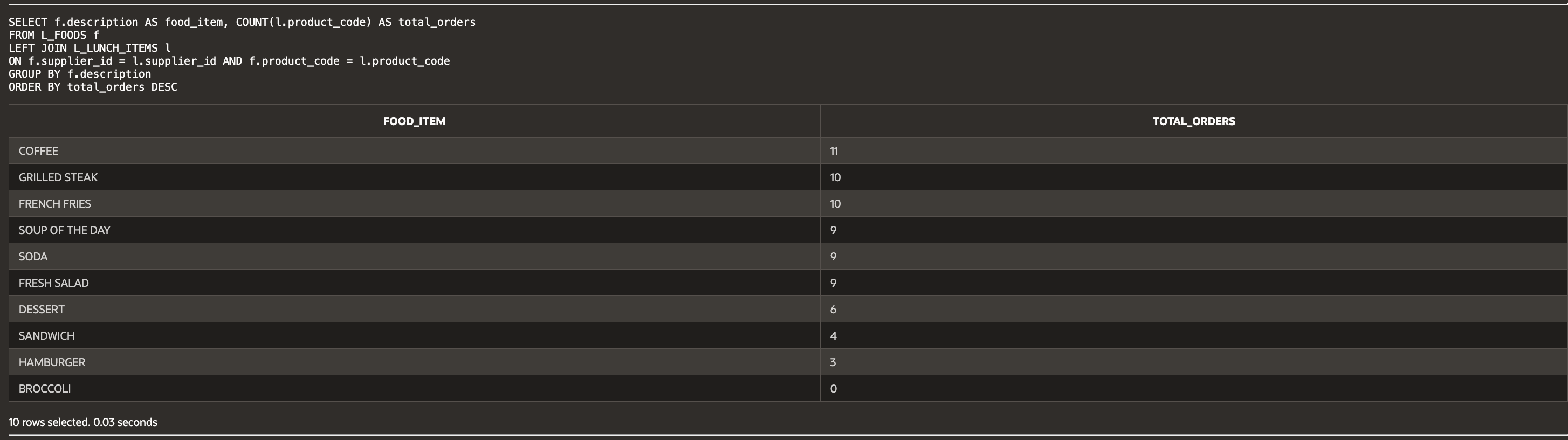
*where a.supplier\_id = b.supplier\_id*

*order* by *a.description;*



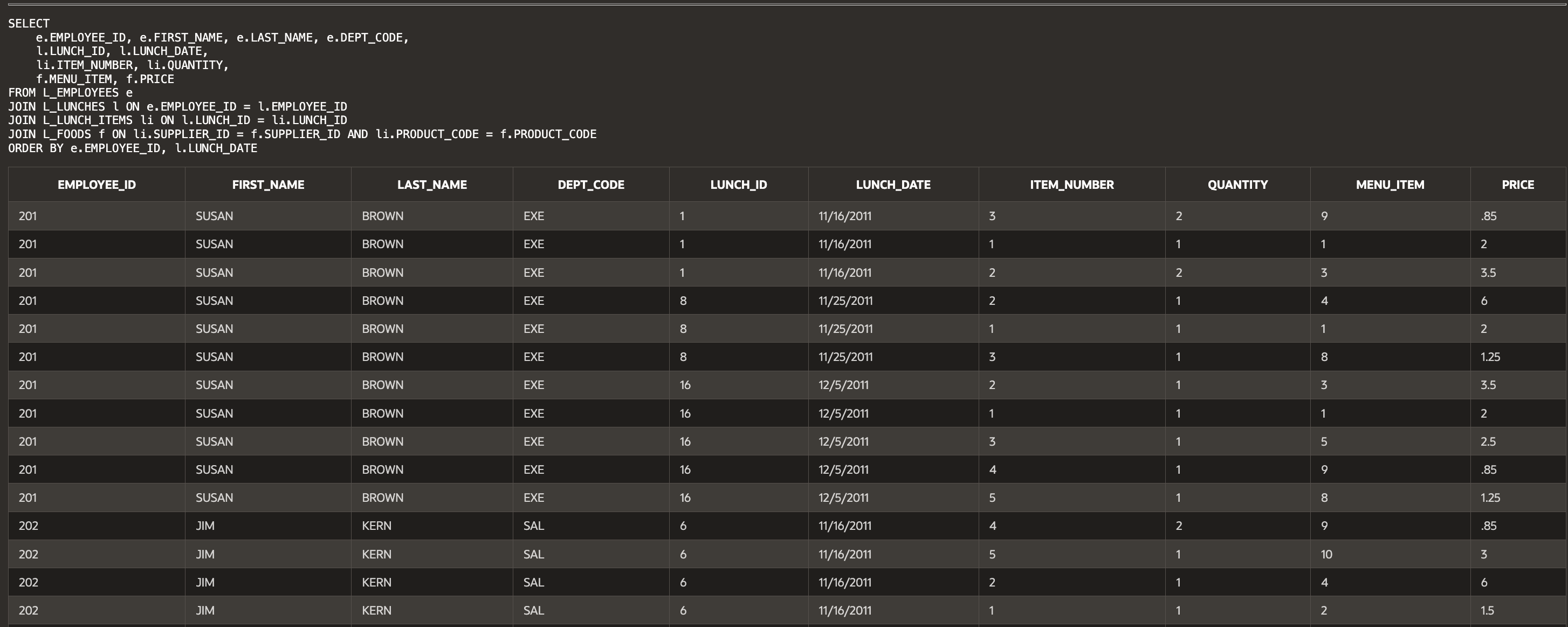
(13-17) Question 8:

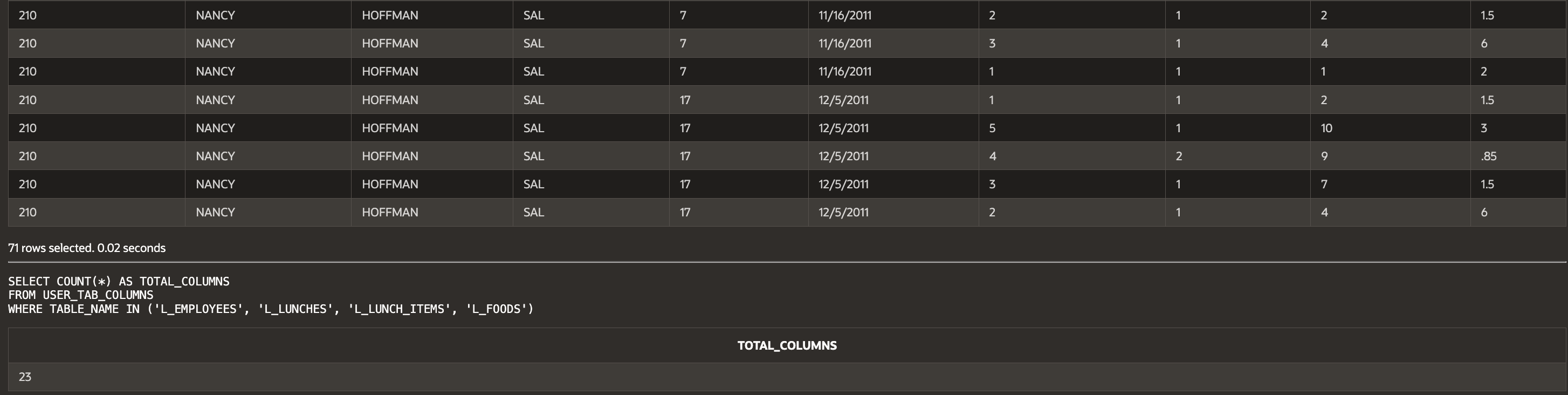
List all the foods on the menu and the total number of orders for each food item. (Note that broccoli does not show up in the result because no one has ordered it.)



(13-19) Question 9:

Join all the tables of the *lunches* database together (Anything table starting with *L\_*). Show all the columns of each table. To do this, modify the *select* statement in this section and add the three other tables to it. How many rows and columns are in this table?

...



**Script:**

-- Haley Archer

-- (13-10) Question 1:

-- Write a select statement to create the inner join of tables sec1310\_table1 and sec1310\_table2.

-- The join condition should say that the first three columns of these tables are equal.

SELECT t1.M1, t1.M2, t1.M3, t1.ADJECTIVE, t2.ANIMAL

FROM SEC1310\_TABLE1 t1

INNER JOIN SEC1310\_TABLE2 t2

ON t1.M1 = t2.M1 AND t1.M2 = t2.M2 AND t1.M3 = t2.M3;

-- (13-11) Question 2:

-- A large dictionary has four volumes. Table sec1311\_dictionary shows the range of words that are in each volume.

-- Table sec1311\_words contains some words that are in the dictionary. Write SQL to determine which volume of the dictionary contains each of these words.

SELECT w.WORD, d.DICTIONARY\_VOLUME

FROM SEC1311\_WORDS w

INNER JOIN SEC1311\_DICTIONARY d

ON w.WORD BETWEEN d.BEGINNING\_LETTER AND d.ENDING\_LETTER;

-- (13-12) Question 3:

-- List all the letters that are greater than S. Use the alphabet table. Put these letters in order.

SELECT letter

FROM alphabet

WHERE letter > 'S'

ORDER BY letter;

-- (13-13) Question 4:

-- Tables sec1313\_words1 and sec1313\_words2 contain words.

-- Join these tables together when the words end in the same letter.

SELECT w1.word AS word1, w2.word AS word2

FROM sec1313\_words1 w1

INNER JOIN sec1313\_words2 w2

ON SUBSTR(w1.word, -1) = SUBSTR(w2.word, -1);

-- (13-14) Question 5:

-- Change this SQL, writing the join condition in the from clause.

SELECT a.student\_name, a.test\_score, b.letter\_grade

FROM sec1311\_student\_scores a

JOIN sec1311\_grade\_ranges b

ON a.test\_score BETWEEN b.beginning\_score AND b.ending\_score

ORDER BY a.student\_name;

-- (13-15) Question 6:

-- Write a select statement to list all the foods on the lunch menu and show the full name of the supplier of each food.

SELECT f.description AS food\_item, s.supplier\_name

FROM L\_FOODS f

INNER JOIN L\_SUPPLIERS s

ON f.supplier\_id = s.supplier\_id;

-- (13-16) Question 7:

-- Demonstrate that a select statement can be separated into two parts:

-- The first part joins the tables and creates a new table, and the second part restricts the amount of data that is shown.

-- Step 1: Create the joined table

CREATE VIEW food\_suppliers AS

SELECT f.description, s.supplier\_name

FROM L\_FOODS f

INNER JOIN L\_SUPPLIERS s

ON f.supplier\_id = s.supplier\_id;

-- Step 2: Restrict the data shown

SELECT \* FROM food\_suppliers

WHERE description LIKE '%SALAD%';

-- (13-17) Question 8:

-- List all the foods on the menu and the total number of orders for each food item.

SELECT f.description AS food\_item, COUNT(l.product\_code) AS total\_orders

FROM L\_FOODS f

LEFT JOIN L\_LUNCH\_ITEMS l

ON f.supplier\_id = l.supplier\_id AND f.product\_code = l.product\_code

GROUP BY f.description

ORDER BY total\_orders DESC;

-- (13-19) Question 9:

-- Join all the tables of the lunches database together (Anything table starting with L\_).

-- Show only the most relevant columns to avoid excessive duplication.

SELECT

e.EMPLOYEE\_ID, e.FIRST\_NAME, e.LAST\_NAME, e.DEPT\_CODE,

l.LUNCH\_ID, l.LUNCH\_DATE,

li.ITEM\_NUMBER, li.QUANTITY,

f.MENU\_ITEM, f.PRICE

FROM L\_EMPLOYEES e

JOIN L\_LUNCHES l ON e.EMPLOYEE\_ID = l.EMPLOYEE\_ID

JOIN L\_LUNCH\_ITEMS li ON l.LUNCH\_ID = li.LUNCH\_ID

JOIN L\_FOODS f ON li.SUPPLIER\_ID = f.SUPPLIER\_ID AND li.PRODUCT\_CODE = f.PRODUCT\_CODE

ORDER BY e.EMPLOYEE\_ID, l.LUNCH\_DATE;

-- Count the total number of columns

SELECT COUNT(\*) AS TOTAL\_COLUMNS

FROM USER\_TAB\_COLUMNS

WHERE TABLE\_NAME IN ('L\_EMPLOYEES', 'L\_LUNCHES', 'L\_LUNCH\_ITEMS', 'L\_FOODS');