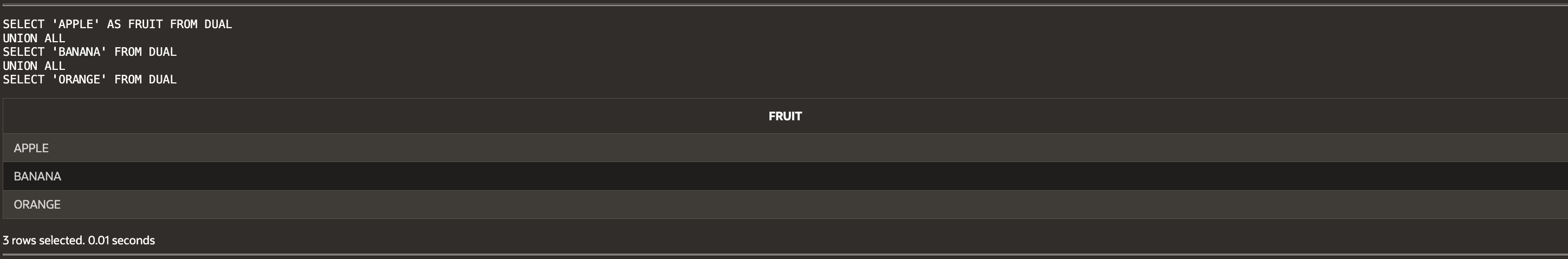
**Week 5 Guided Practice: Unions**

The following questions come from the Task examples of Chapter 15 in your textbook.

After you are finished, please submit a Microsoft Word file that contains screenshots the SQL queries, the output, and please put a comment line in each query with your name. Your document should be named **W5\_GP\_Unions\_Lastname.docx**.

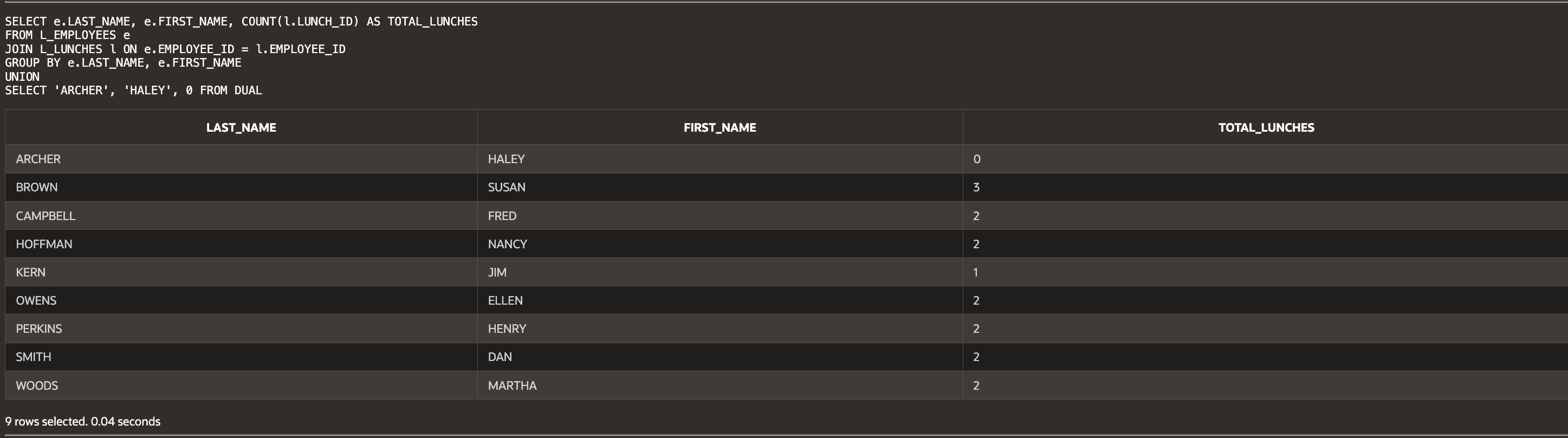
(15-2) Question 1:

Show an example of a *union all*.



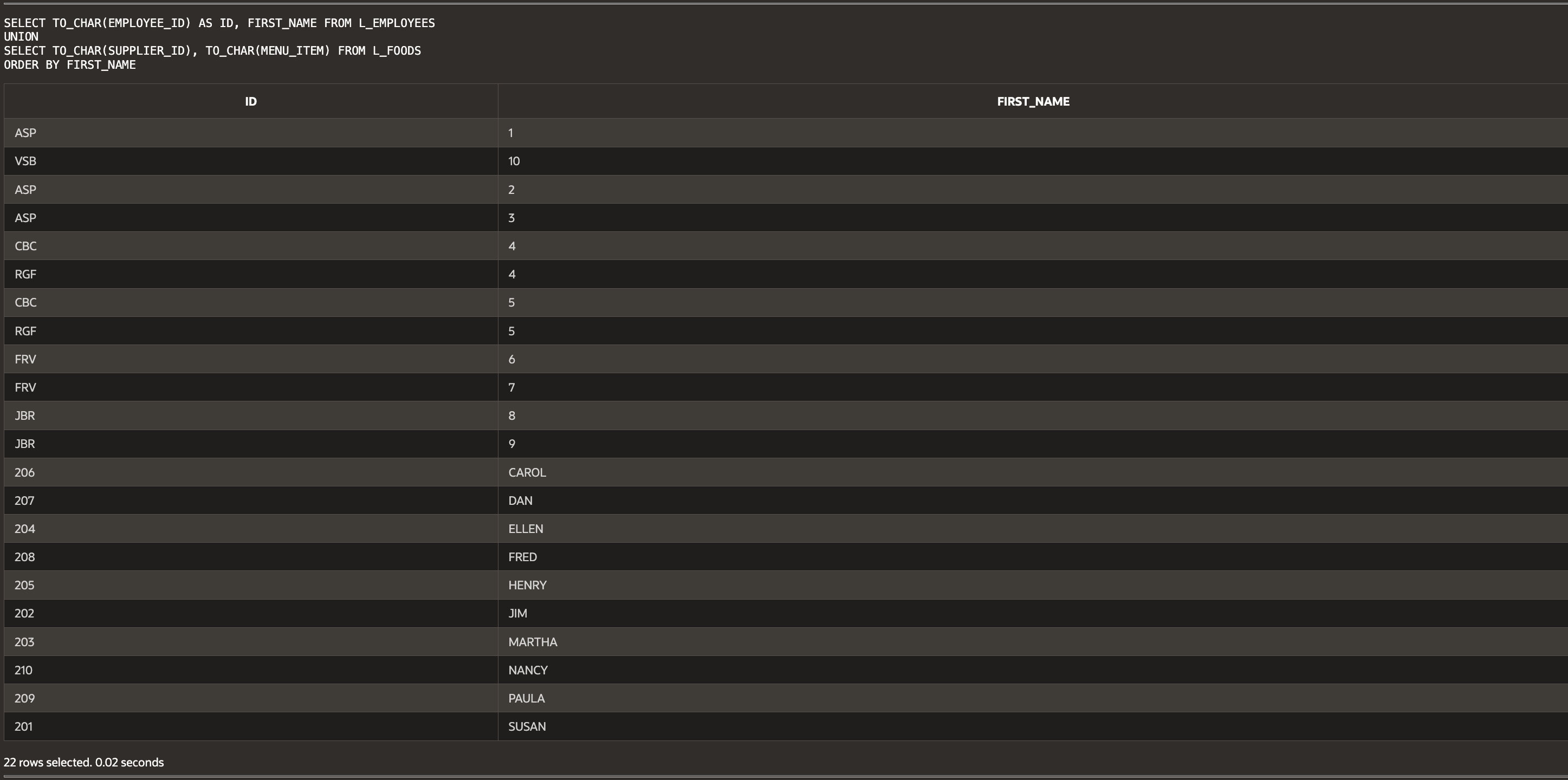
(15-3) Question 2:

From the *L\_LUNCHES* table, count the number of lunches each employee will attend. Get the last name and first name of each employee from the *L\_EMPLOYEES* table using an inner join. Do not try to include employees who are not attending any lunches. Use a *union* to include a row showing that you will not attend any of the lunches.



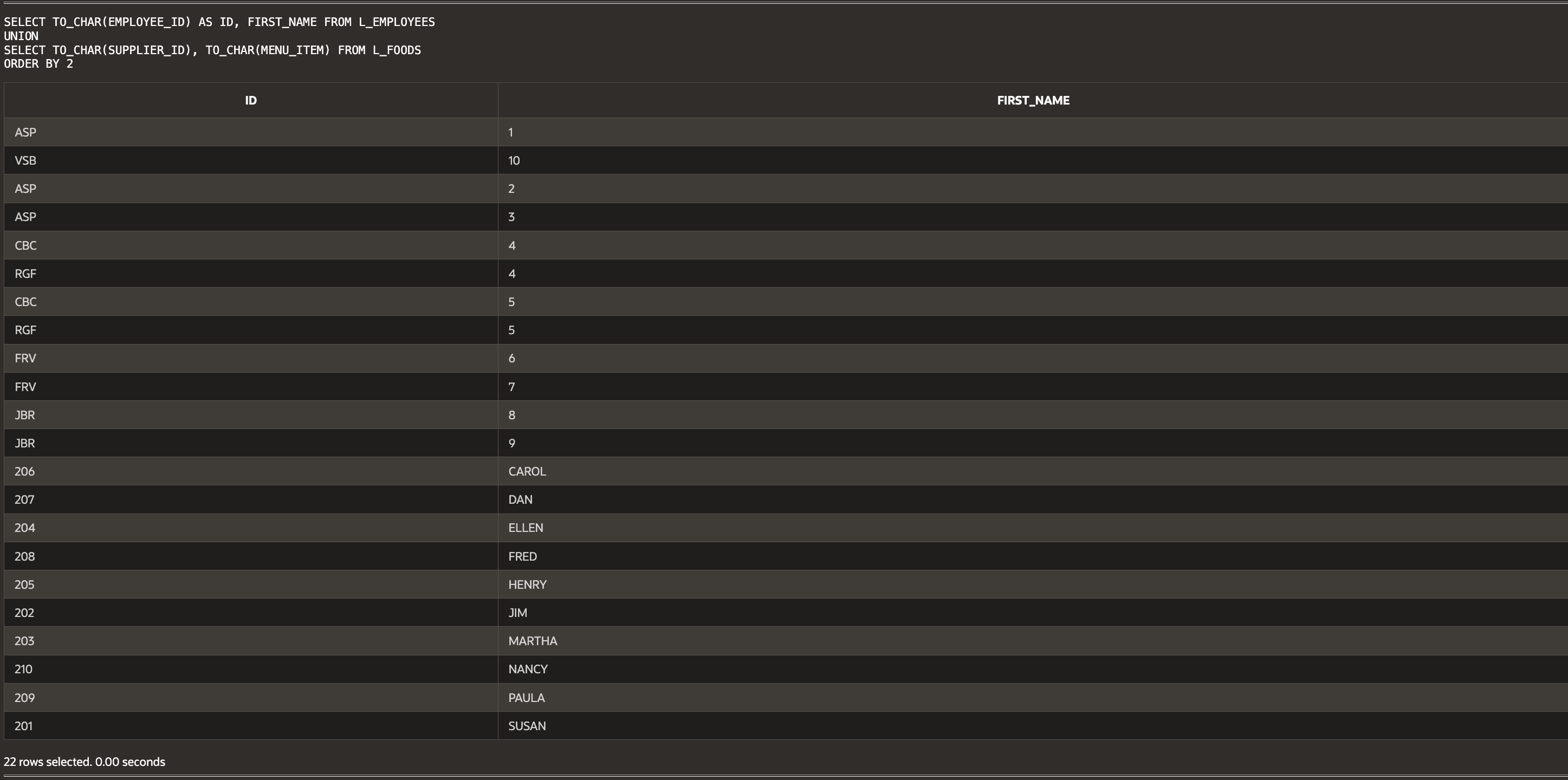
(15-4) Question 3:

Show a *union* that uses a column name in its *order by* clause.



(15-4) Question 4:

Show a *union* that uses a column position number in its *order by* clause.



(15-5) Question 5:

Create a view in Oracle that is defined using a *union*. Show that you can also include an *order* by clause.



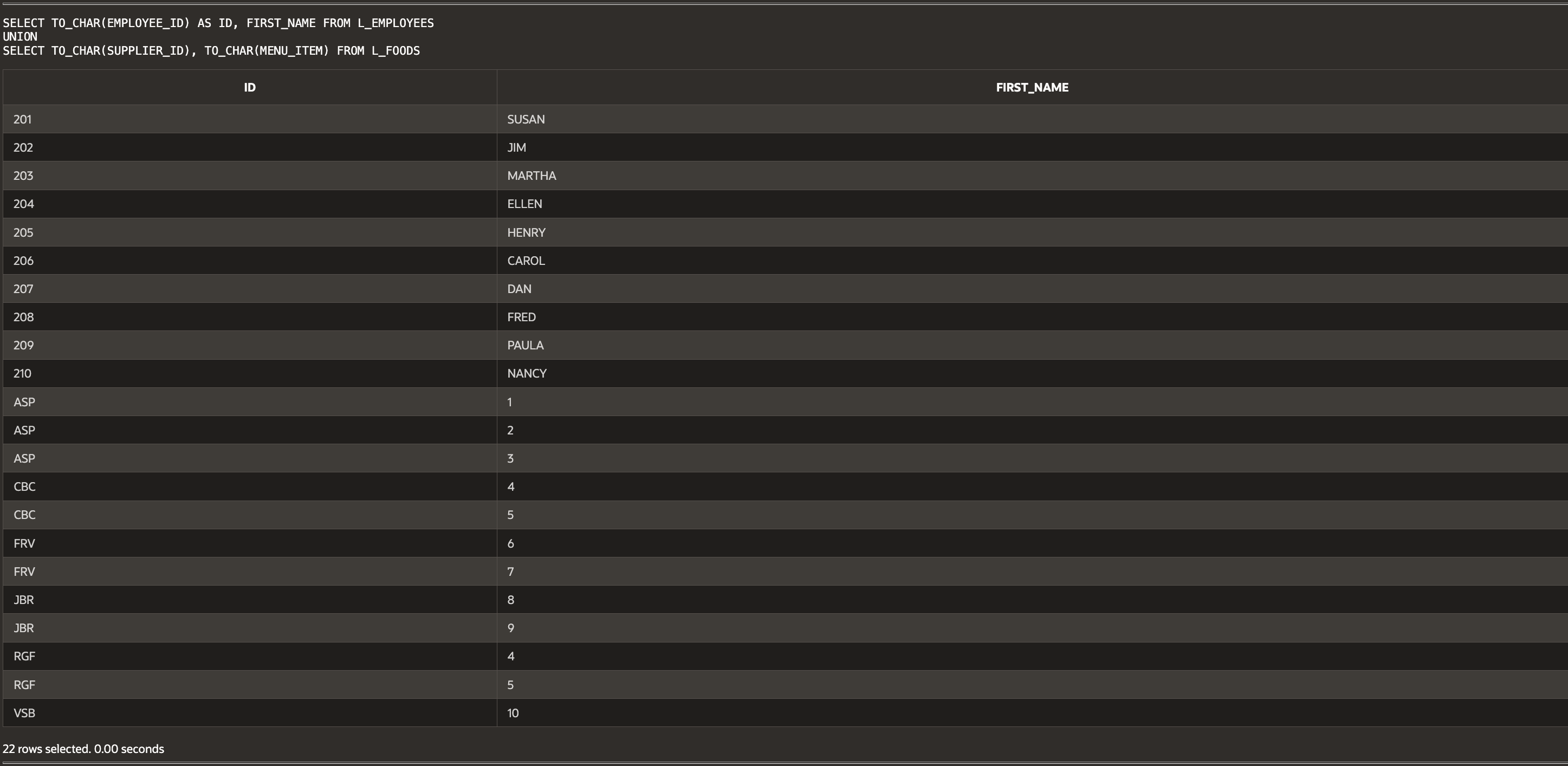
(15-5) Question 6:

Create a table in Oracle that is defined using a *union*. Show that you can also include an *order* by clause.



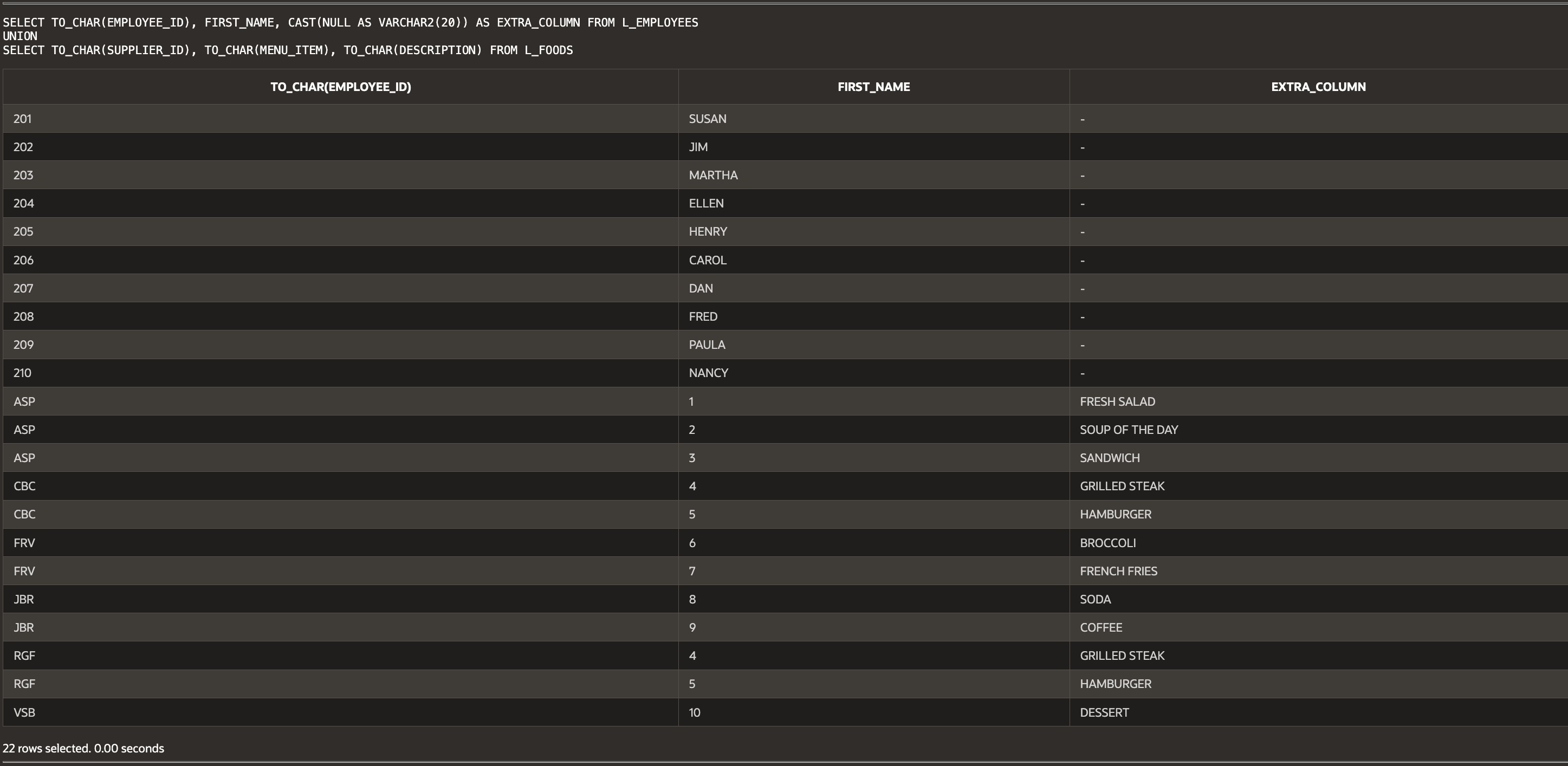
(15-7) Question 7:

Show how to use datatype conversion functions in a union to make every column into a text column.



(15-8) Question 8:

Show how to form a union of two tables that have different numbers of columns.



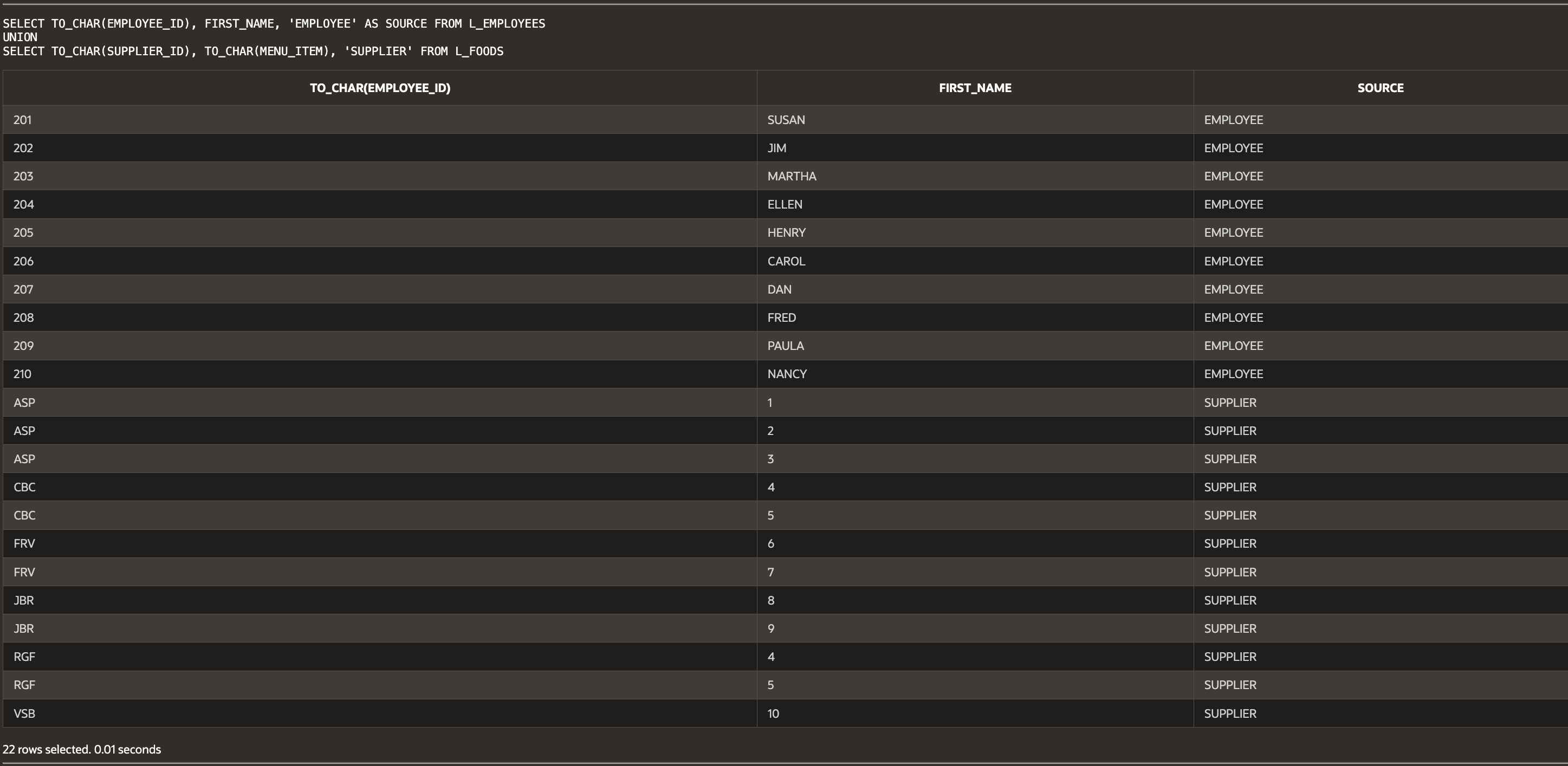
(15-9) Question 9:

Test whether two tables are identical. We already know that these tables have the same number of columns and that the datatypes of those columns are compatible.



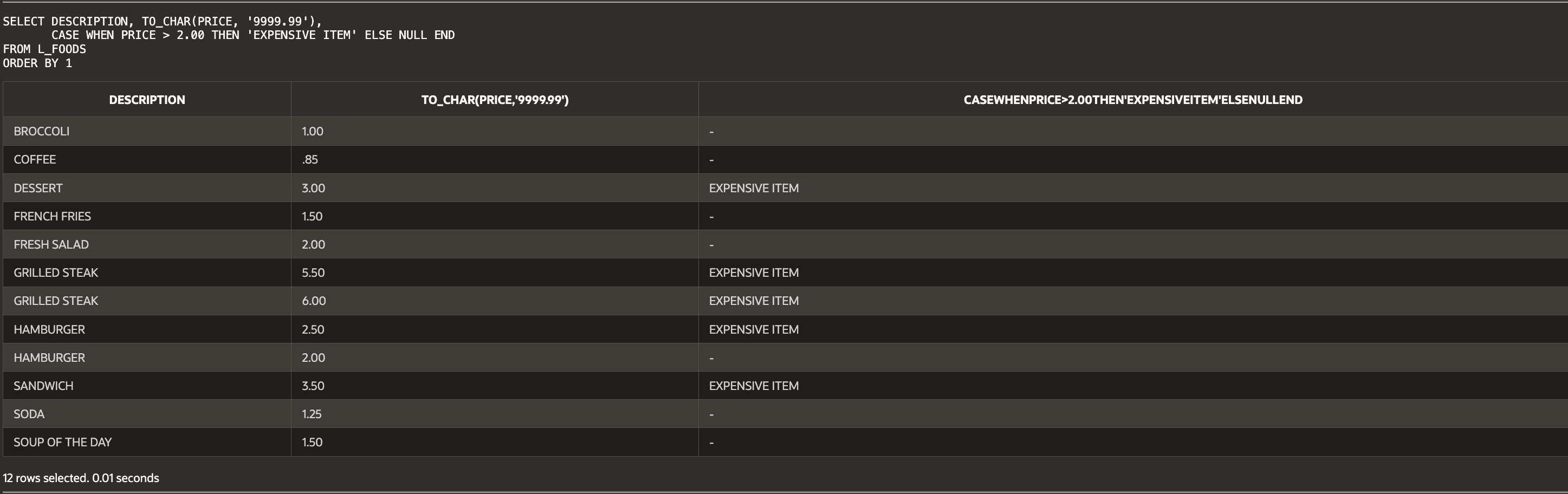
(15-10) Question 10:

Show a *select* statement that uses a *union* with literals to identify the source of each row.



(15-11) Question 11:

List the foods and their prices. Add the message “expensive item” to the foods that cost more than $2.00. List the foods in alphabetical order.



**Script:**

-- Haley Archer

-- (15-2) Question 1:

-- Show an example of a UNION ALL.

SELECT 'APPLE' AS FRUIT FROM DUAL

UNION ALL

SELECT 'BANANA' FROM DUAL

UNION ALL

SELECT 'ORANGE' FROM DUAL;

-- (15-3) Question 2:

-- Count the number of lunches each employee will attend.

-- Get last name & first name using INNER JOIN.

-- Use UNION to include a row showing that you will not attend any lunches.

SELECT e.LAST\_NAME, e.FIRST\_NAME, COUNT(l.LUNCH\_ID) AS TOTAL\_LUNCHES

FROM L\_EMPLOYEES e

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

GROUP BY e.LAST\_NAME, e.FIRST\_NAME

UNION

SELECT 'ARCHER', 'HALEY', TO\_NUMBER(0) FROM DUAL;

-- (15-4) Question 3:

-- Show a UNION that uses a column name in ORDER BY.

SELECT TO\_CHAR(EMPLOYEE\_ID) AS ID, FIRST\_NAME FROM L\_EMPLOYEES

UNION

SELECT TO\_CHAR(SUPPLIER\_ID), TO\_CHAR(MENU\_ITEM) FROM L\_FOODS

ORDER BY FIRST\_NAME;

-- (15-4) Question 4:

-- Show a UNION that uses a column position in ORDER BY.

SELECT TO\_CHAR(EMPLOYEE\_ID) AS ID, FIRST\_NAME FROM L\_EMPLOYEES

UNION

SELECT TO\_CHAR(SUPPLIER\_ID), TO\_CHAR(MENU\_ITEM) FROM L\_FOODS

ORDER BY 2;

-- (15-5) Question 5:

-- Create a VIEW defined using a UNION with ORDER BY.

CREATE OR REPLACE VIEW EMPLOYEES\_AND\_SUPPLIERS AS

SELECT TO\_CHAR(EMPLOYEE\_ID) AS ID, FIRST\_NAME AS NAME FROM L\_EMPLOYEES

UNION

SELECT TO\_CHAR(SUPPLIER\_ID), TO\_CHAR(MENU\_ITEM) FROM L\_FOODS;

-- (15-5) Question 6:

-- Create a TABLE defined using a UNION with ORDER BY.

CREATE TABLE EMPLOYEE\_SUPPLIER\_TABLE AS

SELECT TO\_CHAR(EMPLOYEE\_ID) AS ID, FIRST\_NAME AS NAME FROM L\_EMPLOYEES

UNION

SELECT TO\_CHAR(SUPPLIER\_ID), TO\_CHAR(MENU\_ITEM) FROM L\_FOODS;

-- (15-7) Question 7:

-- Use datatype conversion functions in a UNION.

SELECT TO\_CHAR(EMPLOYEE\_ID) AS ID, FIRST\_NAME FROM L\_EMPLOYEES

UNION

SELECT TO\_CHAR(SUPPLIER\_ID), TO\_CHAR(MENU\_ITEM) FROM L\_FOODS;

-- (15-8) Question 8:

-- UNION of two tables with different numbers of columns.

SELECT TO\_CHAR(EMPLOYEE\_ID), FIRST\_NAME, CAST(NULL AS VARCHAR2(20)) AS EXTRA\_COLUMN FROM L\_EMPLOYEES

UNION

SELECT TO\_CHAR(SUPPLIER\_ID), TO\_CHAR(MENU\_ITEM), TO\_CHAR(DESCRIPTION) FROM L\_FOODS;

-- (15-9) Question 9:

-- Test if two tables are identical.

SELECT 'MATCH' AS RESULT FROM DUAL

WHERE EXISTS (

SELECT TO\_CHAR(EMPLOYEE\_ID), FIRST\_NAME FROM L\_EMPLOYEES

INTERSECT

SELECT TO\_CHAR(EMPLOYEE\_ID), FIRST\_NAME FROM L\_EMPLOYEES

)

UNION

SELECT 'NO MATCH' FROM DUAL

WHERE NOT EXISTS (

SELECT TO\_CHAR(EMPLOYEE\_ID), FIRST\_NAME FROM L\_EMPLOYEES

INTERSECT

SELECT TO\_CHAR(EMPLOYEE\_ID), FIRST\_NAME FROM L\_EMPLOYEES

);

-- (15-10) Question 10:

-- UNION with literals to identify the source of each row.

SELECT TO\_CHAR(EMPLOYEE\_ID), FIRST\_NAME, 'EMPLOYEE' AS SOURCE FROM L\_EMPLOYEES

UNION

SELECT TO\_CHAR(SUPPLIER\_ID), TO\_CHAR(MENU\_ITEM), 'SUPPLIER' FROM L\_FOODS;

-- (15-11) Question 11:

-- List foods and prices, marking "EXPENSIVE ITEM" for prices > $2.

SELECT DESCRIPTION, TO\_CHAR(PRICE, '9999.99'),

CASE WHEN PRICE > 2.00 THEN 'EXPENSIVE ITEM' ELSE NULL END

FROM L\_FOODS

ORDER BY 1;