**Week 5 Performance Assessment: Unions**

The following questions come from the “Check your understanding” examples in Chapter 15 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 each query containing your name. Your document should be named **W5\_PA\_Unions\_Lastname.docx**.

(15-1) Question 1:

1. Here is some information about the size of two tables:

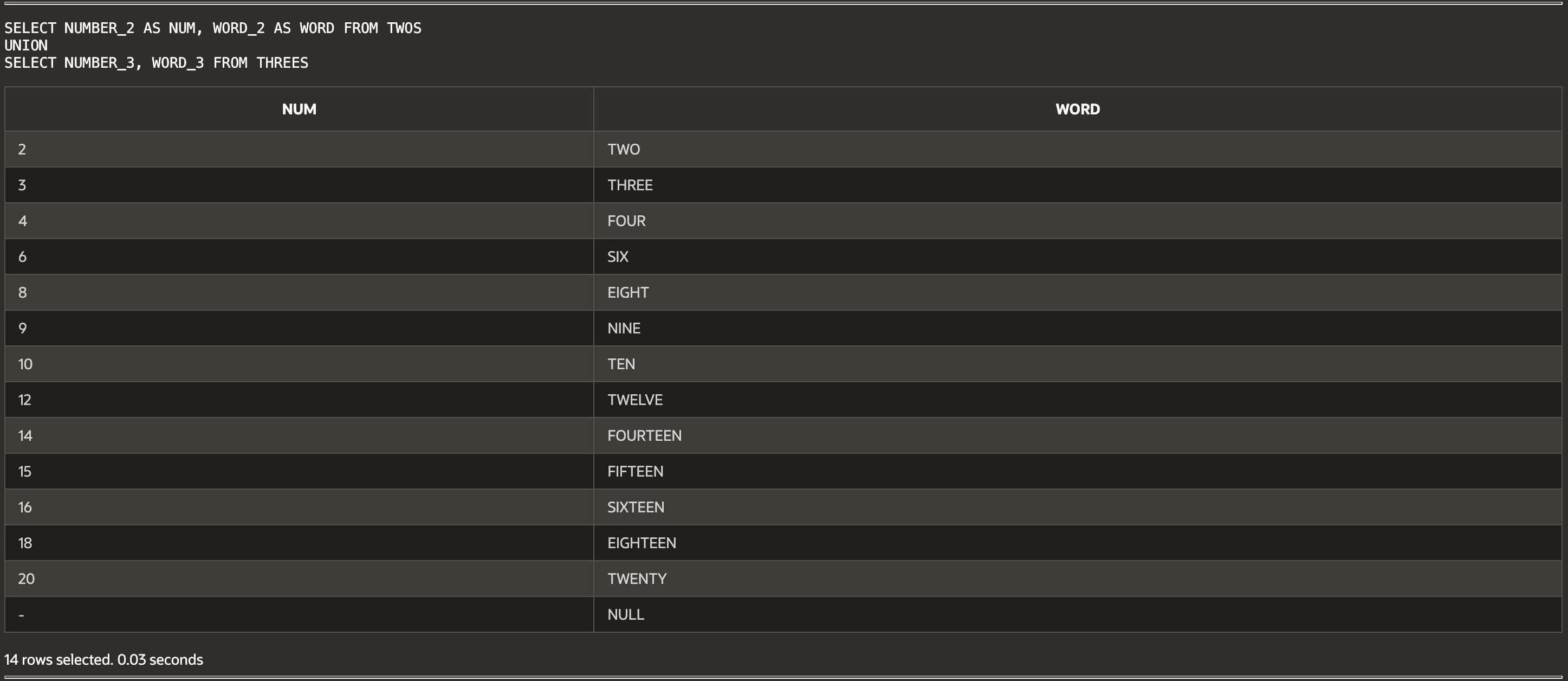
Table 1: 10 columns 100,000 rows

Table 2: 10 columns 50,000 rows

What is the maximum number of columns and rows of the union of these two tables? What is the maximum number of columns and rows of the inner join of the tables?   
**UNION:** Maximum columns = 10 (same structure), Maximum rows = 100,000 + 50,000 = 150,000

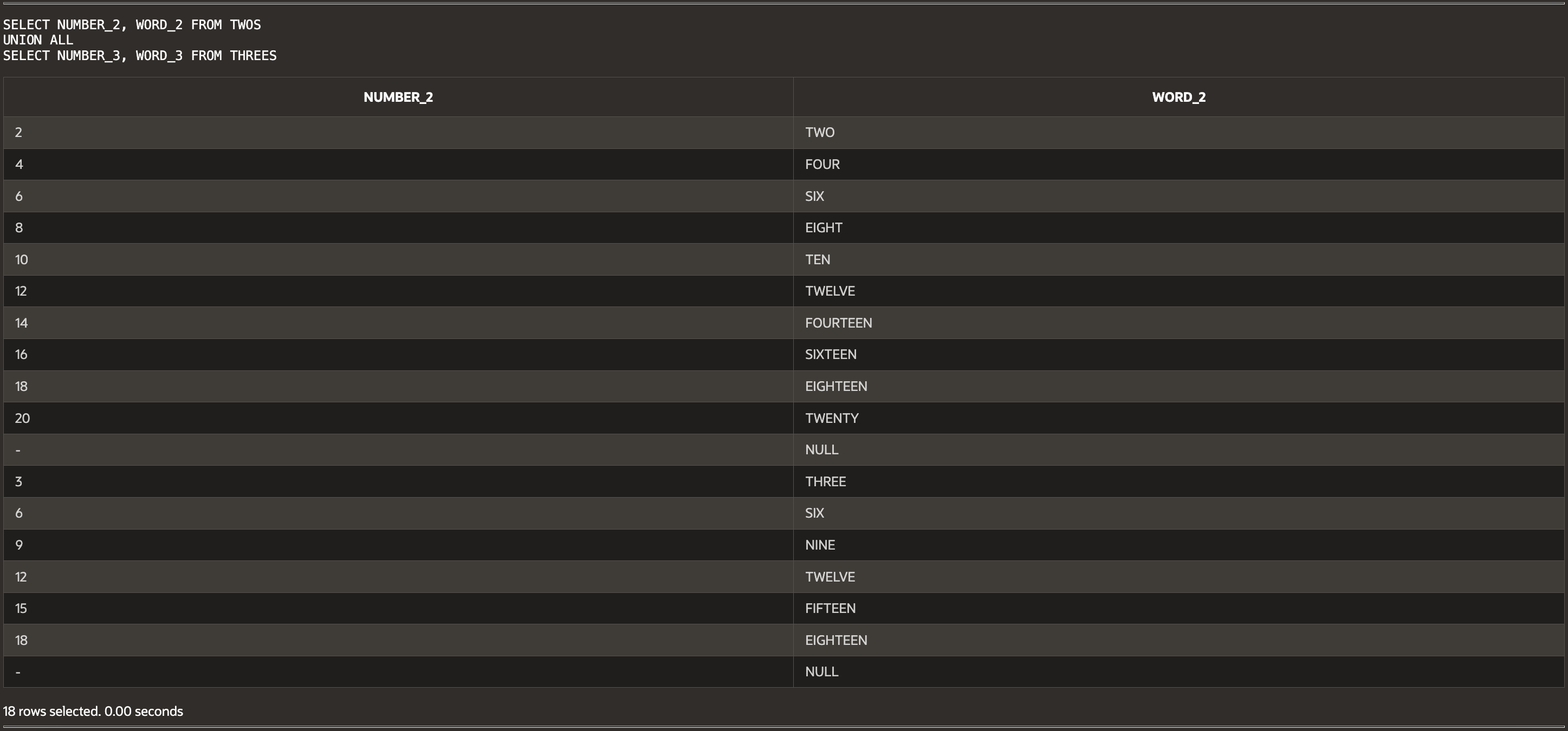
**INNER JOIN:** Maximum columns = 20 (all columns from both tables if no overlap), Maximum rows = at most 50,000 (assuming full match).

2. Write a *select* statement to form the *union* of the *twos* table and the threes table.



(15-2) Question 2:

Write a *select* statement to form a union of the *twos* table and the *threes* table. Use *union all*. How does this differ from using a regular *union*?

UNION removes duplicates, UNION ALL keeps all records including duplicates.

(15-3) Question 3:

1. What is wrong with this *select* statement?

*select number\_2*

*from twos*

*union*

*select number\_3,*

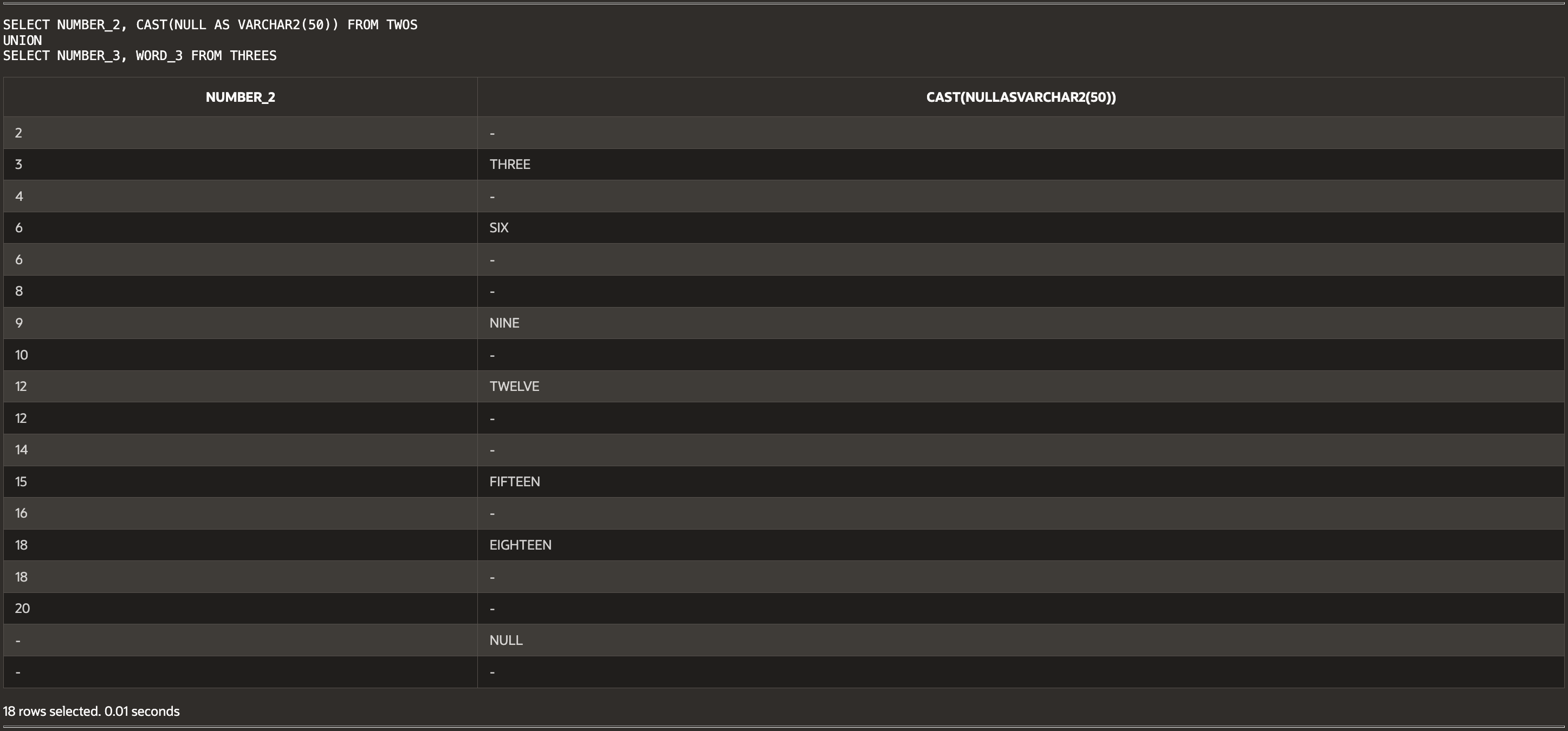
*word\_3*

*from threes;*

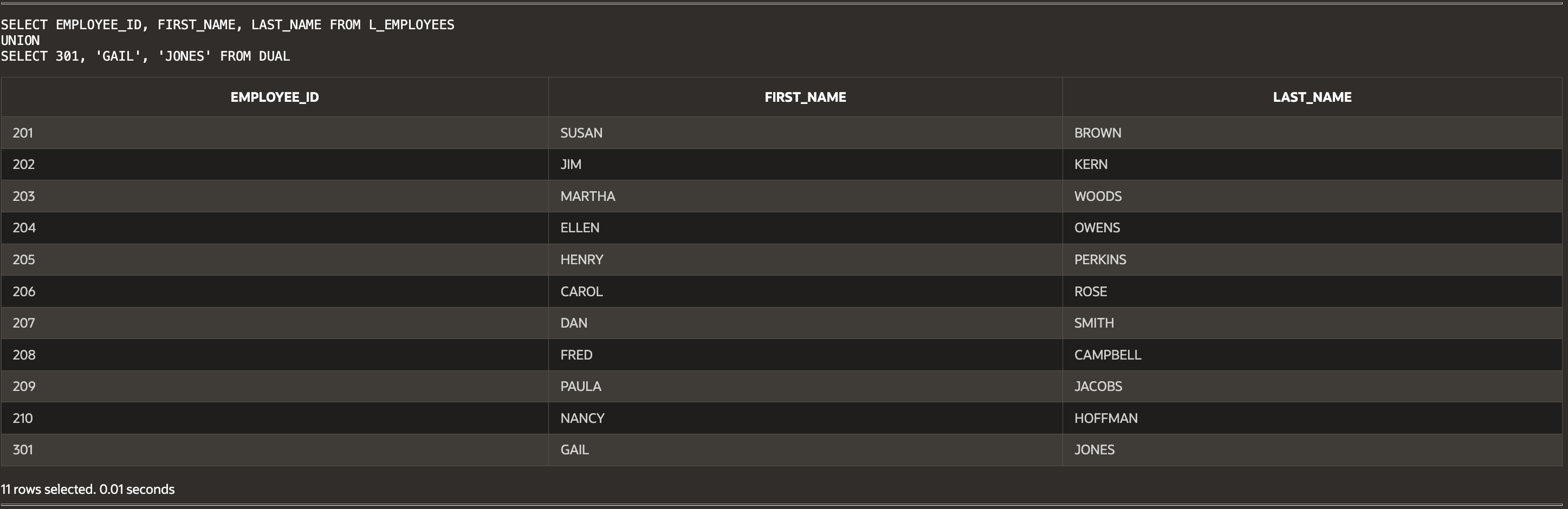
**What's wrong?** *The column count does not match.*

*SELECT NUMBER\_2 FROM TWOS UNION SELECT NUMBER\_3, WORD\_3 FROM THREES;*

**FIX:** *Ensure same number of columns in both selects*



2. Goal 1: Show that a *union* is similar to an *insert* statement in that it can add new data to the result table.



3. Goal 2: Show a *union* that uses more than two *select* statements. The following select statement shows the number of lunches that each employee will attend, but it does not account for Carol Rose or Paula Jacobs because they are not attending any lunches. Modify this statement to show that these two people will not attend any lunches.

*select a.first\_name,*

*a.last\_name,*

*count(b.lunch\_id) as number\_of\_lunches*

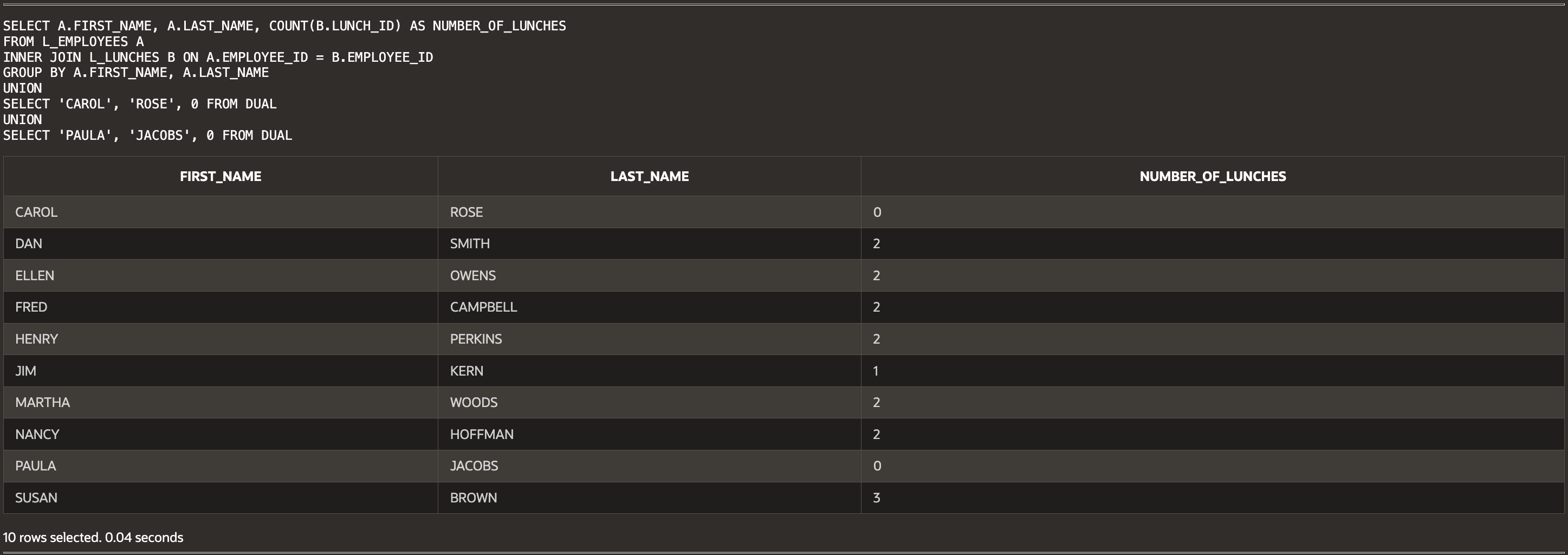
*from L\_EMPLOYEES a*

*inner join L\_LUNCHES b*

*on a.employee\_id = b.employee\_id*

*group by a.first\_name,*

*a.last\_name;*



(15-4) Question 4:

Modify the following *union*. Add an *order* by clause to it to sort the rows by the last name. Try all four methods. Which ones work?

*SELECT A.FIRST\_NAME, A.LAST\_NAME, COUNT(B.LUNCH\_ID) AS NUMBER\_OF\_LUNCHES*

*FROM L\_EMPLOYEES A*

*INNER JOIN L\_LUNCHES B ON A.EMPLOYEE\_ID = B.EMPLOYEE\_ID*

*GROUP BY A.FIRST\_NAME, A.LAST\_NAME*

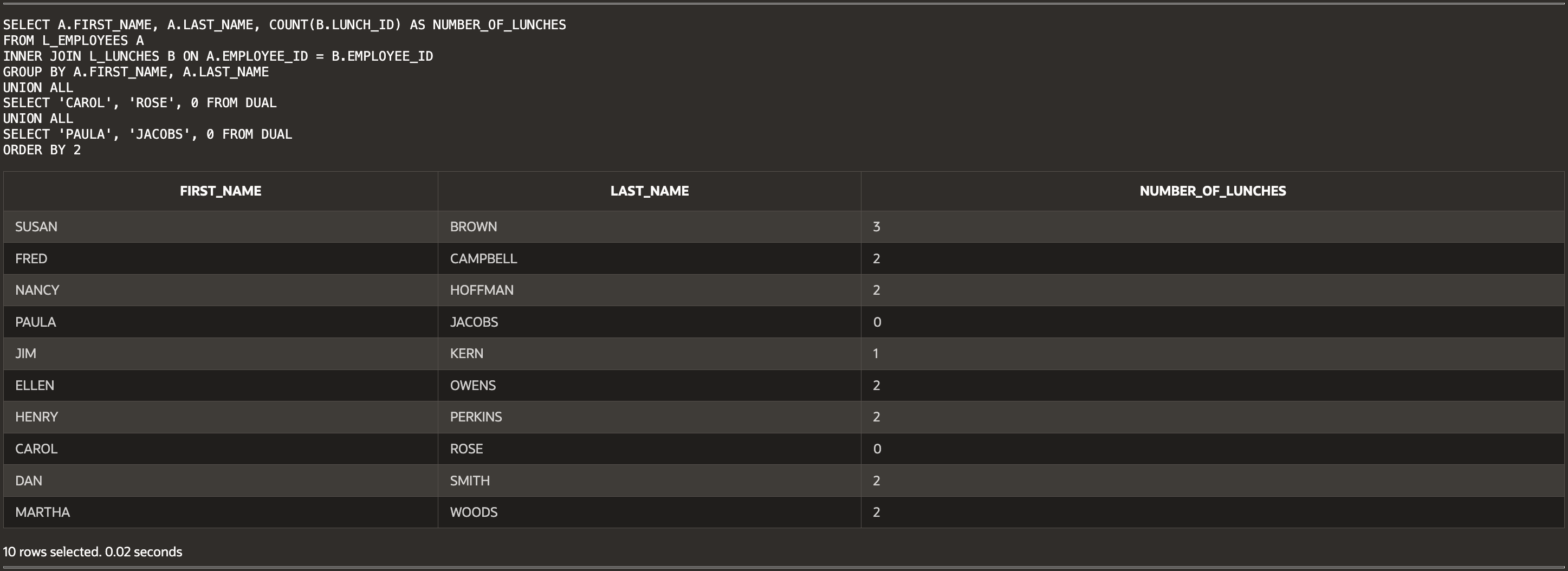
*UNION ALL*

*SELECT 'CAROL', 'ROSE', 0 FROM DUAL*

*UNION ALL*

*SELECT 'PAULA', 'JACOBS', 0 FROM DUAL*

*ORDER BY 2;*



(15-5) Question 5:

Goal: Show that a *union* can add new rows of data to a table. This is similar to what an *insert* statement does.

First, create a *select* statement that lists all the columns and rows of the *L\_EMPLOYEES* table and uses a *union all* to add the following new employee. Then save the result table as a new table called *sec1505\_employees*

Employee\_id: 301

First\_Name: Gail

Last\_Name: Jones

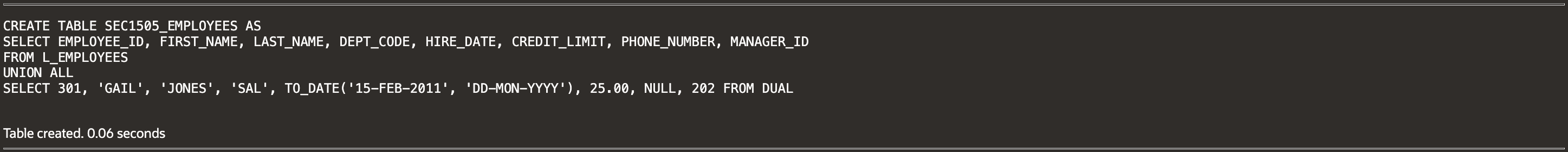
Dept\_code: Sal

Hire\_date: Feb 15, 2011

Credit\_limit: $25.00

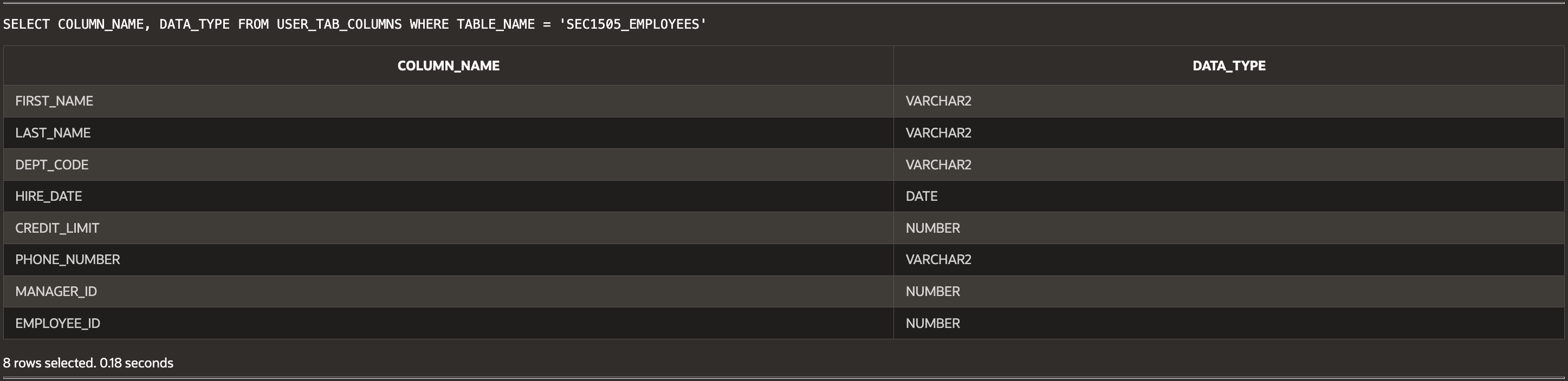
Phone\_number: (null)

Manager\_id: 202



(15-6) Question 6:

Run the code from this section. Use the methods of **section 7-12** to examine the datatypes of the columns of the beginning tables and of the new view created by the *union*. Have any of the datatypes changed in the process of forming the union?



(15-7) Question 7:

Modify the following select statement. Convert the datatypes of all the columns to text. (Actually, sometimes this code will work as it is and the conversion of the datatypes is done automatically for you behind the scenes.)

*select date\_1,*

*date\_1,*

*date\_1*

*from sec1507\_first*

*union*

*select number\_2,*

*word\_2,*

*date\_2*

*from sec1507\_second;*



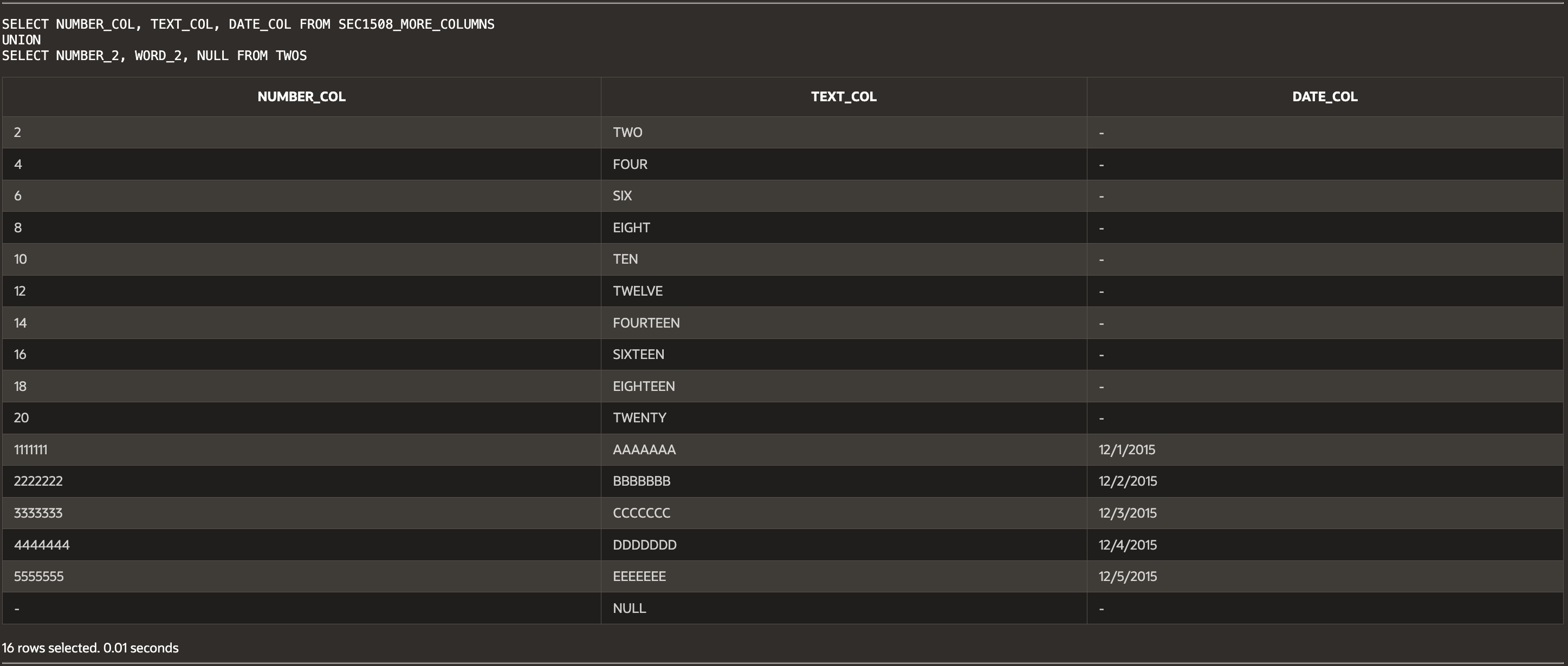
(15-8) Question 8:

Modify the following *select* statement to make it work. Add one more column to the second *select* statement. You can use either a null or a literal value.

SELECT NUMBER\_COL, TEXT\_COL, DATE\_COL FROM SEC1508\_MORE\_COLUMNS

UNION

SELECT NUMBER\_2, WORD\_2, NULL FROM TWOS;



(15-10) Question 9:

The following *select* statement creates a *union* of the *twos* table with the *threes* table. Add a new column to show the table from which each row comes.

*select number\_2,*

*word\_2*

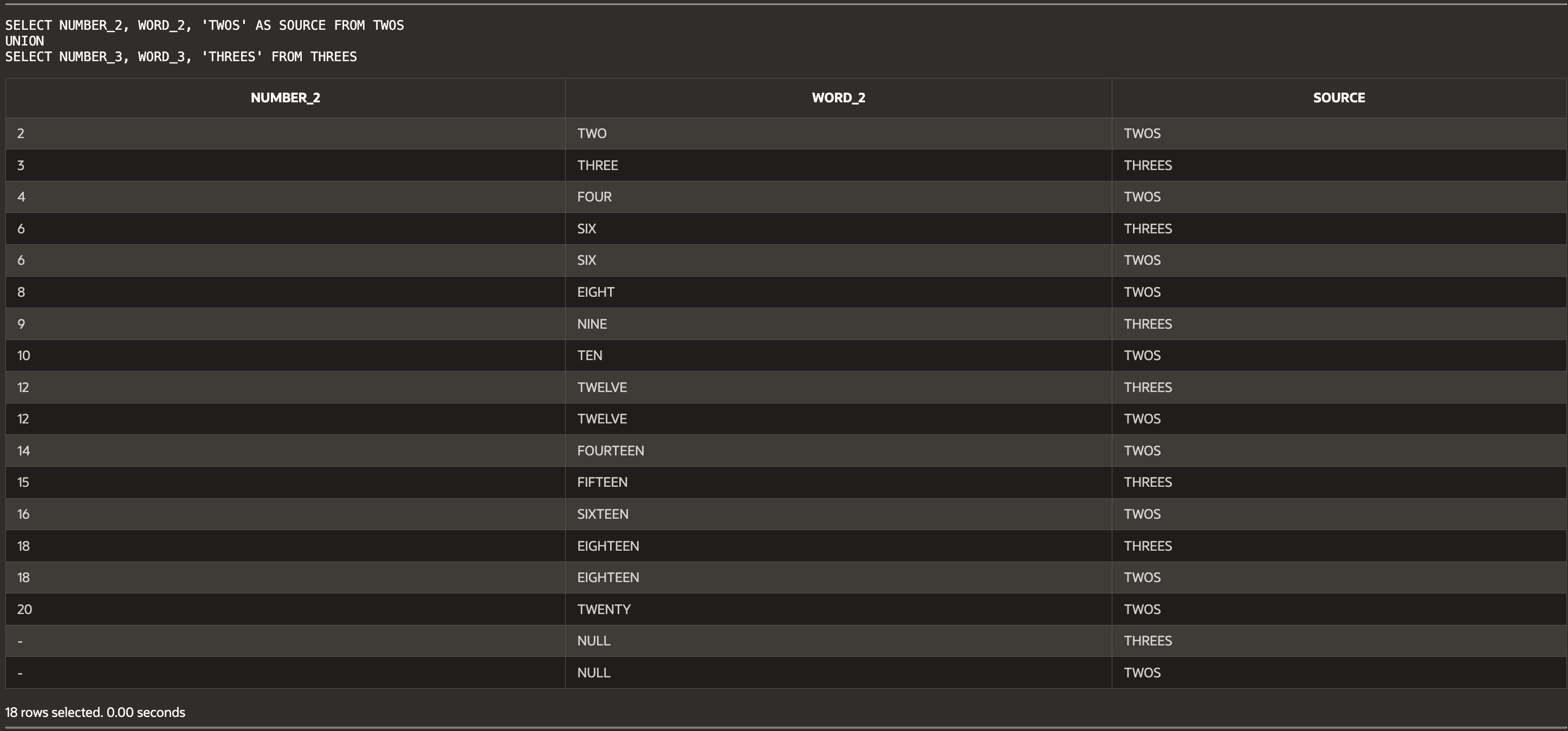
*from twos*

*union*

*select number\_3,*

*word\_3*

*from threes*;



**Script:**

-- Haley Archer

-- (15-1) Question 1:

-- Maximum number of columns and rows for UNION and INNER JOIN

-- Answer:

-- UNION: Maximum columns = 10 (same structure), Maximum rows = 100,000 + 50,000 = 150,000

-- INNER JOIN: Maximum columns = 20 (all columns from both tables if no overlap), Maximum rows = at most 50,000 (assuming full match).

-- (15-1) Question 2:

-- Form a UNION of the twos table and the threes table.

SELECT NUMBER\_2 AS NUM, WORD\_2 AS WORD FROM TWOS

UNION

SELECT NUMBER\_3, WORD\_3 FROM THREES;

-- (15-2) Question 1:

-- Form a UNION ALL of the twos and threes tables.

SELECT NUMBER\_2, WORD\_2 FROM TWOS

UNION ALL

SELECT NUMBER\_3, WORD\_3 FROM THREES;

-- UNION removes duplicates, UNION ALL keeps all records including duplicates.

-- (15-3) Question 1:

-- What's wrong? The column count does not match.

-- SELECT NUMBER\_2 FROM TWOS UNION SELECT NUMBER\_3, WORD\_3 FROM THREES;

-- FIX: Ensure same number of columns in both selects

SELECT NUMBER\_2, CAST(NULL AS VARCHAR2(50)) FROM TWOS

UNION

SELECT NUMBER\_3, WORD\_3 FROM THREES;

-- (15-3) Goal 1:

-- Show that a union adds new data like an insert.

SELECT EMPLOYEE\_ID, FIRST\_NAME, LAST\_NAME FROM L\_EMPLOYEES

UNION

SELECT 301, 'GAIL', 'JONES' FROM DUAL;

-- (15-3) Goal 2:

-- Modify the statement to show Carol Rose and Paula Jacobs with zero lunches.

SELECT A.FIRST\_NAME, A.LAST\_NAME, COUNT(B.LUNCH\_ID) AS NUMBER\_OF\_LUNCHES

FROM L\_EMPLOYEES A

INNER JOIN L\_LUNCHES B ON A.EMPLOYEE\_ID = B.EMPLOYEE\_ID

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

UNION

SELECT 'CAROL', 'ROSE', 0 FROM DUAL

UNION

SELECT 'PAULA', 'JACOBS', 0 FROM DUAL;

-- (15-4) Question 4:

-- Add ORDER BY clause to sort by last name using different methods

SELECT A.FIRST\_NAME, A.LAST\_NAME, COUNT(B.LUNCH\_ID) AS NUMBER\_OF\_LUNCHES

FROM L\_EMPLOYEES A

INNER JOIN L\_LUNCHES B ON A.EMPLOYEE\_ID = B.EMPLOYEE\_ID

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

UNION ALL

SELECT 'CAROL', 'ROSE', 0 FROM DUAL

UNION ALL

SELECT 'PAULA', 'JACOBS', 0 FROM DUAL

ORDER BY 2;

-- (15-5) Question 5:

-- Insert new employee into a new table using UNION ALL

CREATE TABLE SEC1505\_EMPLOYEES AS

SELECT EMPLOYEE\_ID, FIRST\_NAME, LAST\_NAME, DEPT\_CODE, HIRE\_DATE, CREDIT\_LIMIT, PHONE\_NUMBER, MANAGER\_ID

FROM L\_EMPLOYEES

UNION ALL

SELECT 301, 'GAIL', 'JONES', 'SAL', TO\_DATE('15-FEB-2011', 'DD-MON-YYYY'), 25.00, NULL, 202 FROM DUAL;

-- (15-6) Question 6:

-- Examine datatypes of original tables and new view.

SELECT COLUMN\_NAME, DATA\_TYPE FROM USER\_TAB\_COLUMNS WHERE TABLE\_NAME = 'SEC1505\_EMPLOYEES';

-- (15-7) Question 7:

-- Convert all datatypes to text

SELECT TO\_CHAR(DATE\_1), TO\_CHAR(DATE\_1), TO\_CHAR(DATE\_1) FROM SEC1507\_FIRST

UNION

SELECT TO\_CHAR(NUMBER\_2), WORD\_2, TO\_CHAR(DATE\_2) FROM SEC1507\_SECOND;

-- (15-8) Question 8:

-- Ensure both SELECTs have the same column count.

SELECT NUMBER\_COL, TEXT\_COL, DATE\_COL FROM SEC1508\_MORE\_COLUMNS

UNION

SELECT NUMBER\_2, WORD\_2, NULL FROM TWOS;

-- (15-10) Question 9:

-- Show the source table for each row in the UNION.

SELECT NUMBER\_2, WORD\_2, 'TWOS' AS SOURCE FROM TWOS

UNION

SELECT NUMBER\_3, WORD\_3, 'THREES' FROM THREES;