**BBDD Actividad 5**

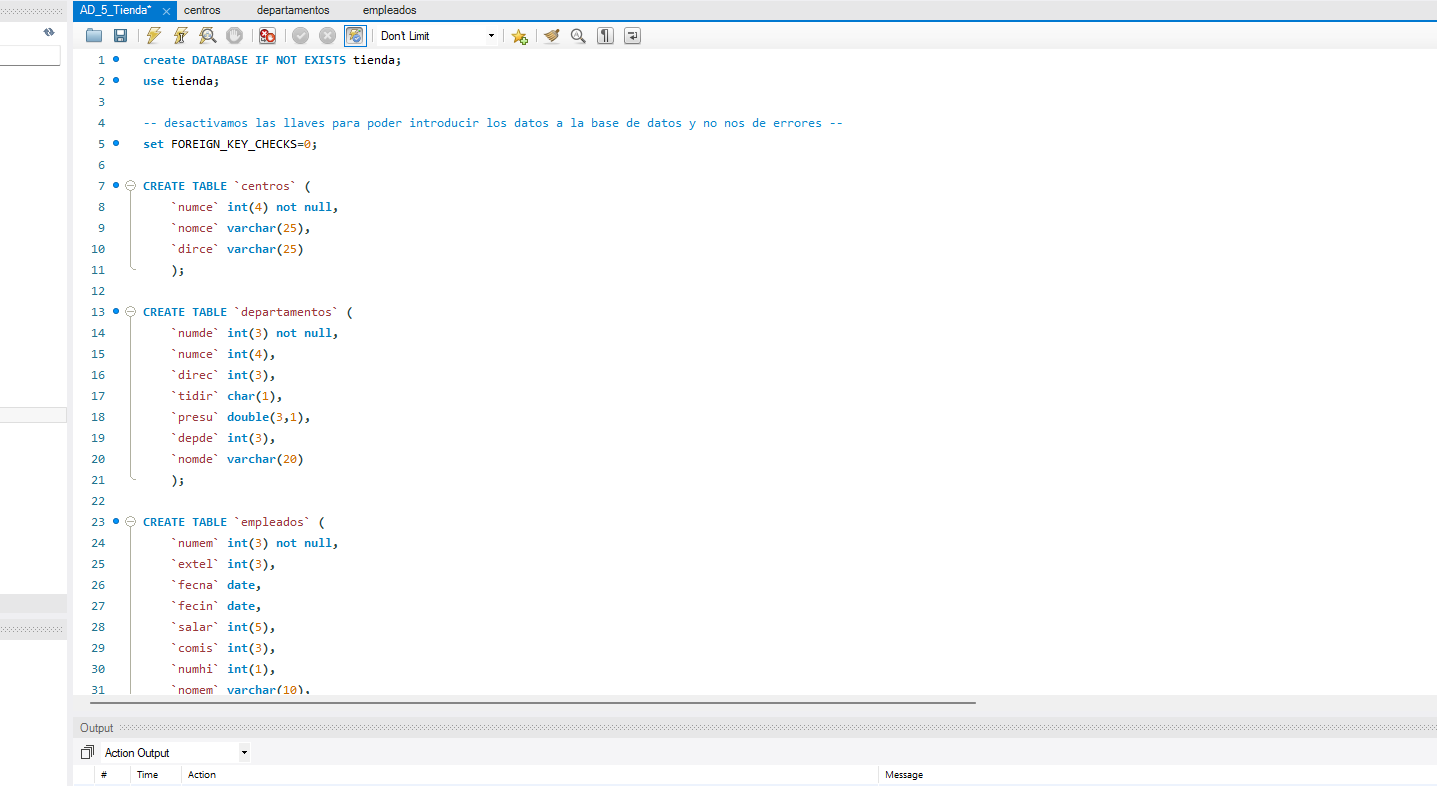
**SENTENCIAS SQL**

**AVANZADAS Y VISTAS**

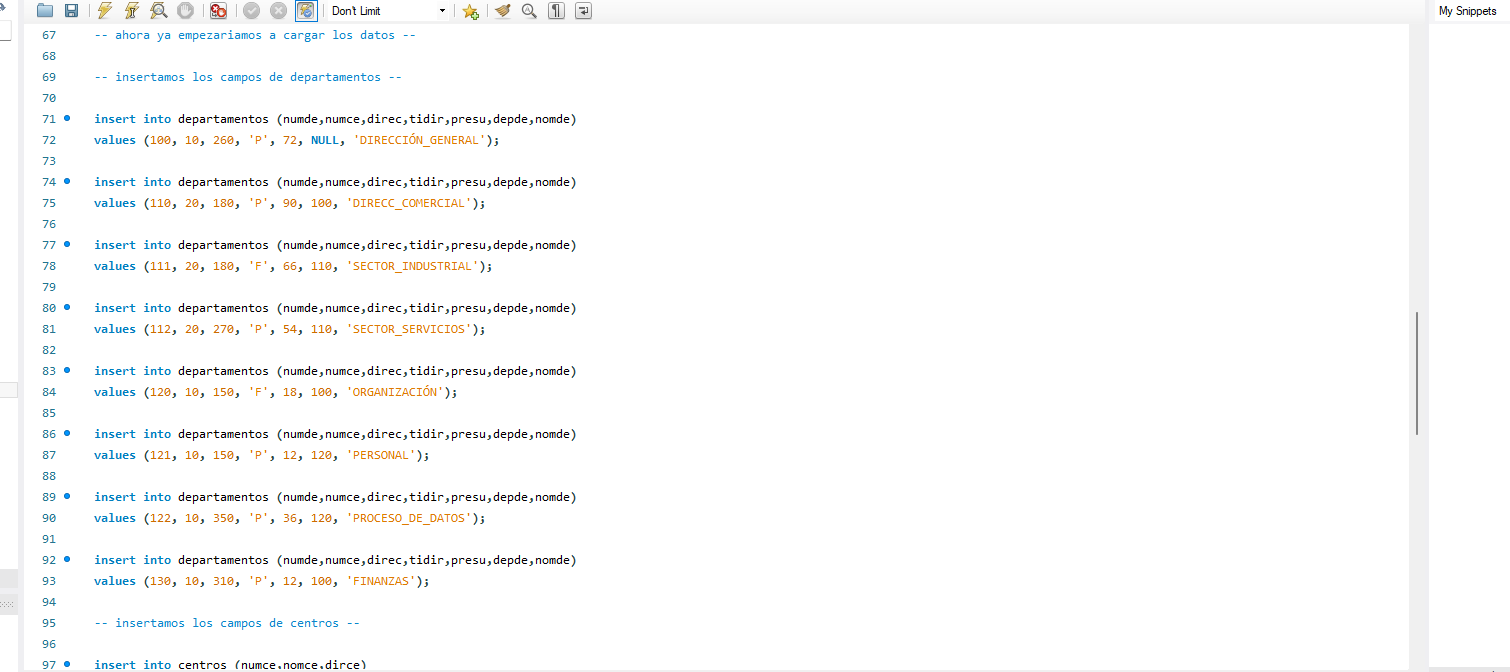
* **GRUPO 13:**
  + Gerard Perujo Buxeda
  + María Isabel Martín Simal
  + Noelia Villahermosa García

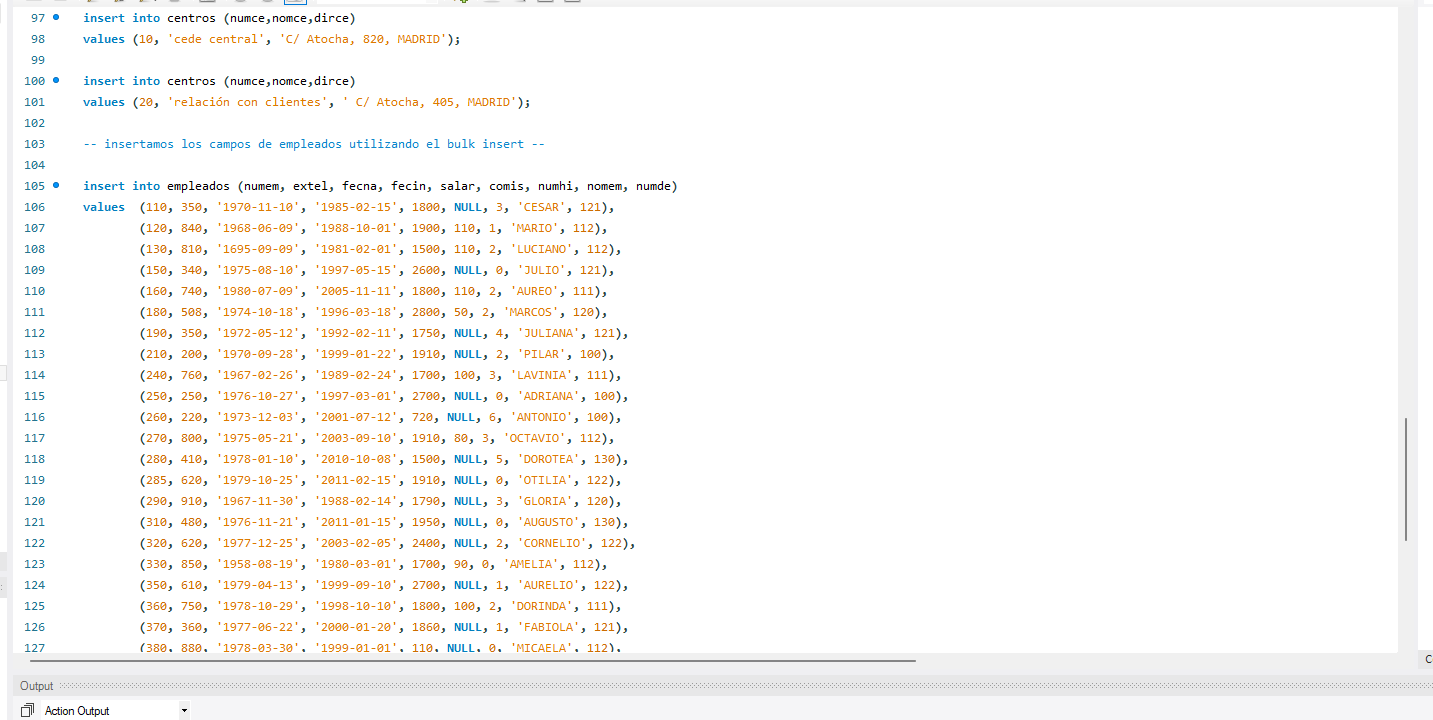
**Creación Base de Datos: Gerard Perujo**

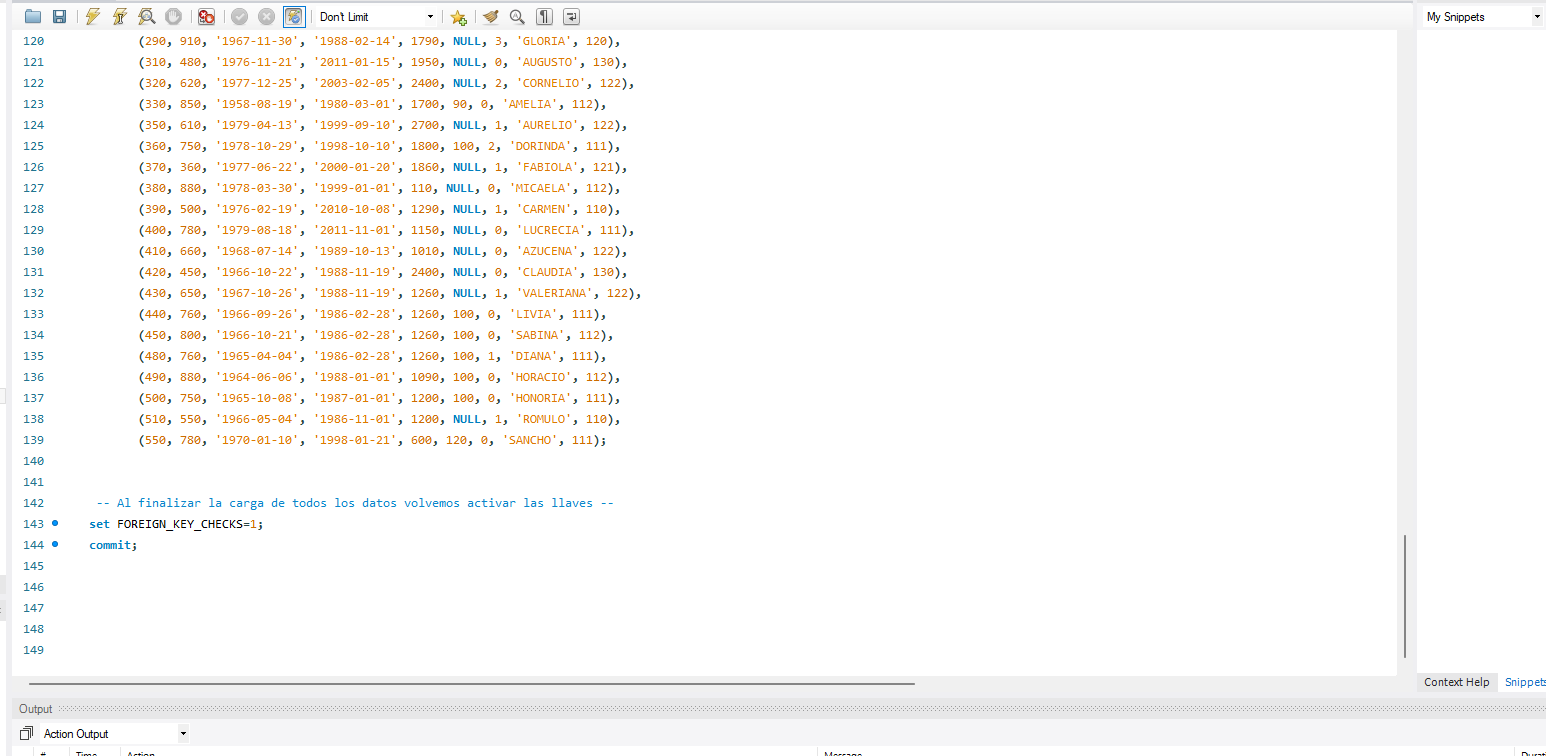
1. Creamos la base de datos de Tienda. Al insertar el primer registro si no se desactivan las llaves da error, ya que no se pueden introducir datos con las llaves activadas. Es necesario desactivarlas antes de introducir datos.











**SENTENCIAS**

**Queries 1 al 6: Gerard Perujo**

1. Para cada departamento con presupuesto inferior a 35.000 €, hallar el nombre del Centro donde está ubicado y el máximo salario de sus empleados (si dicho máximo excede de 1.500 €). Clasificar

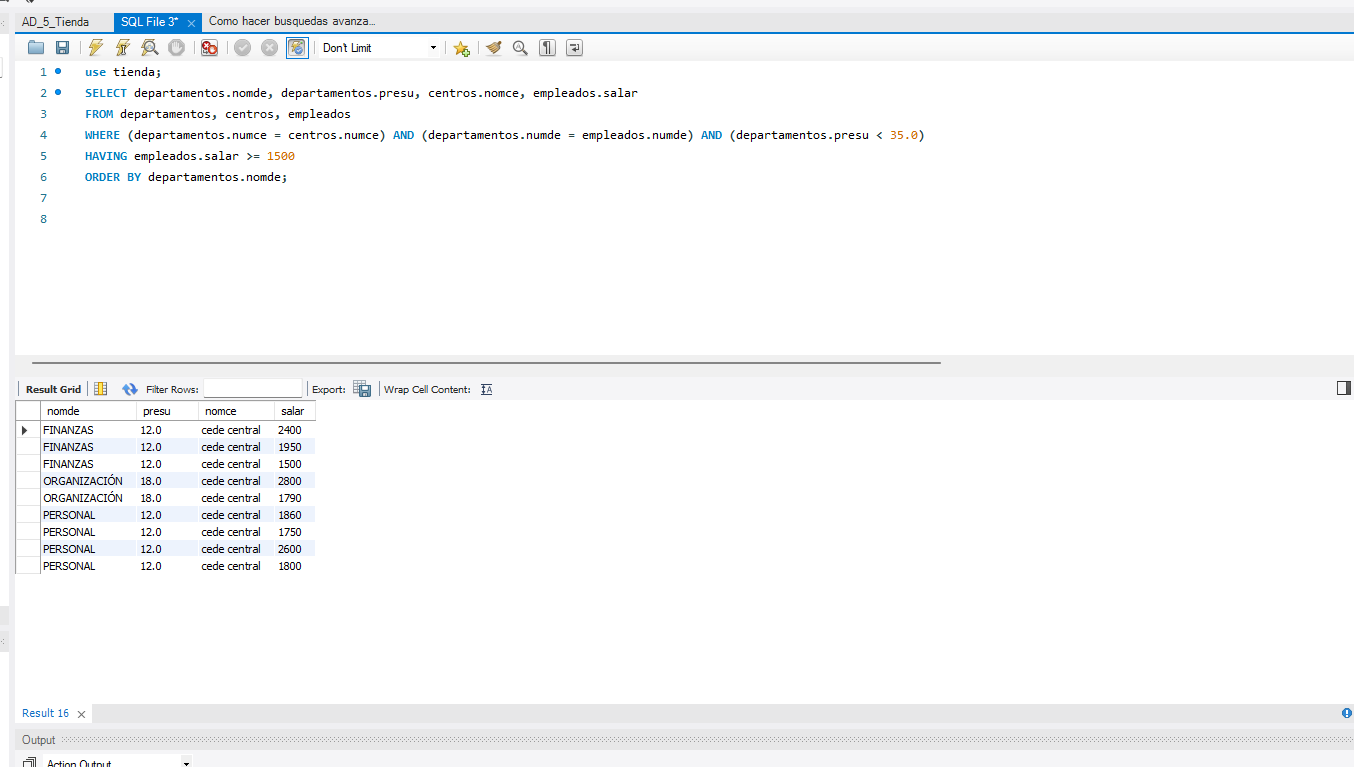
SELECT departamentos.nomde, departamentos.presu, centros.nomce, empleados.salar

FROM departamentos, centros, empleados

WHERE (departamentos.numce = centros.numce) AND (departamentos.numde = empleados.numde) AND (departamentos.presu < 35.0)

HAVING empleados.salar >= 1500

ORDER BY departamentos.nomde;



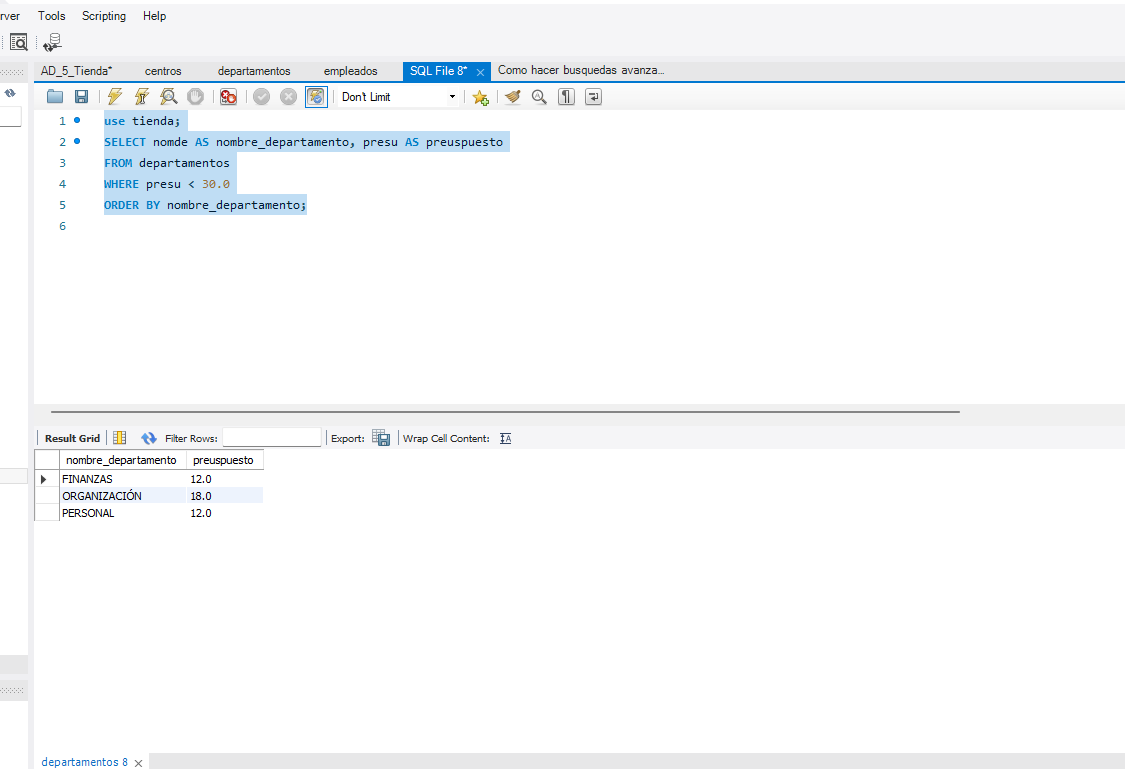
1. Hallar por orden alfabético los nombres de los departamentos que dependen de los que tienen un presupuesto inferior a 30.000 €. También queremos conocer el nombre del departamento del que dependen y su presupuesto.

SELECT nomde AS nombre\_departamento, presu AS preuspuesto

FROM departamentos

WHERE presu < 30.0

ORDER BY nombre\_departamento;



1. Obtener los nombres y los salarios medios de los departamentos cuyo salario medio supera al salario medio de la empresa.

SELECT departamentos.nomde, AVG(empleados.salar)

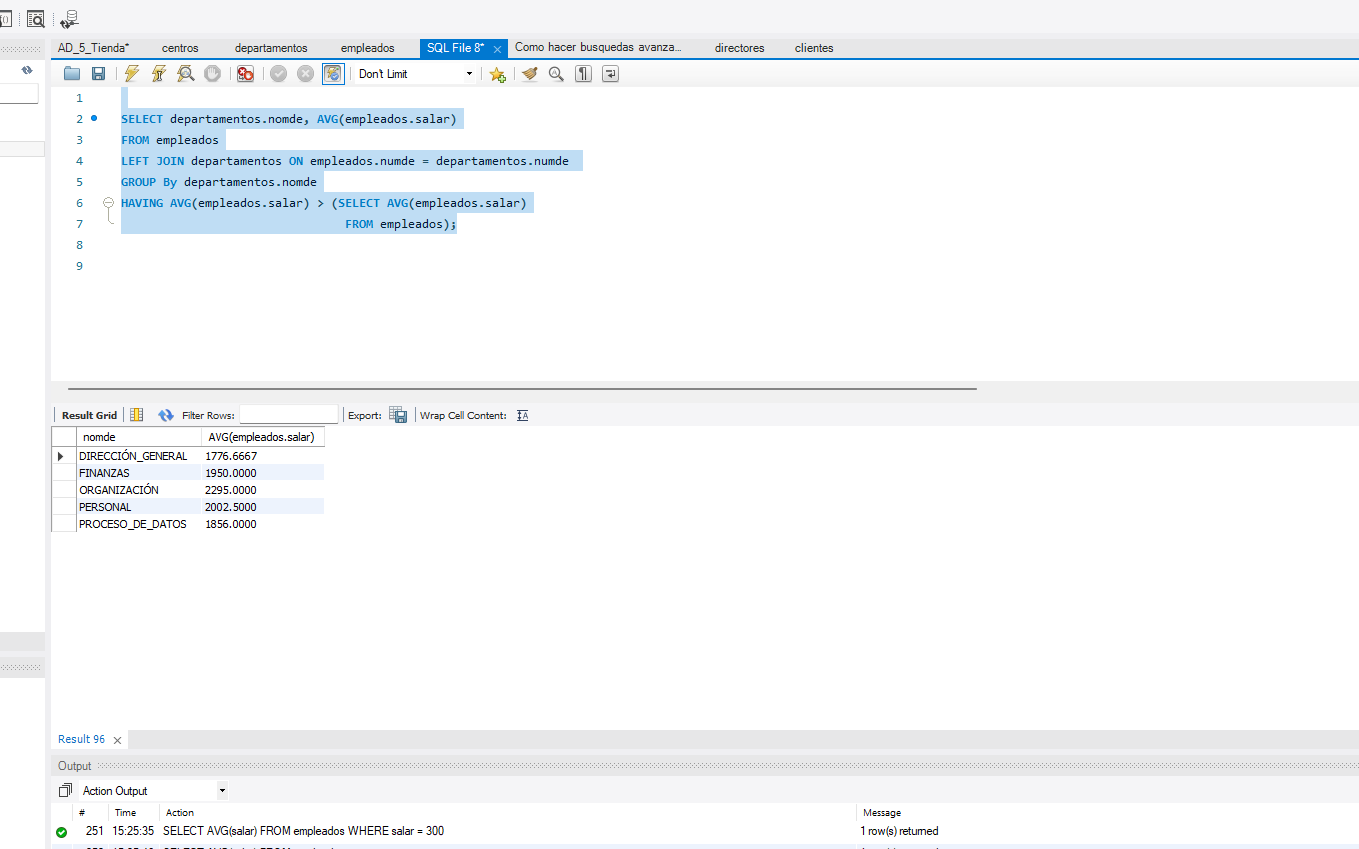
FROM empleados

LEFT JOIN departamentos ON empleados.numde = departamentos.numde

GROUP By departamentos.nomde

HAVING AVG(empleados.salar) > (SELECT AVG(empleados.salar)

FROM empleados);



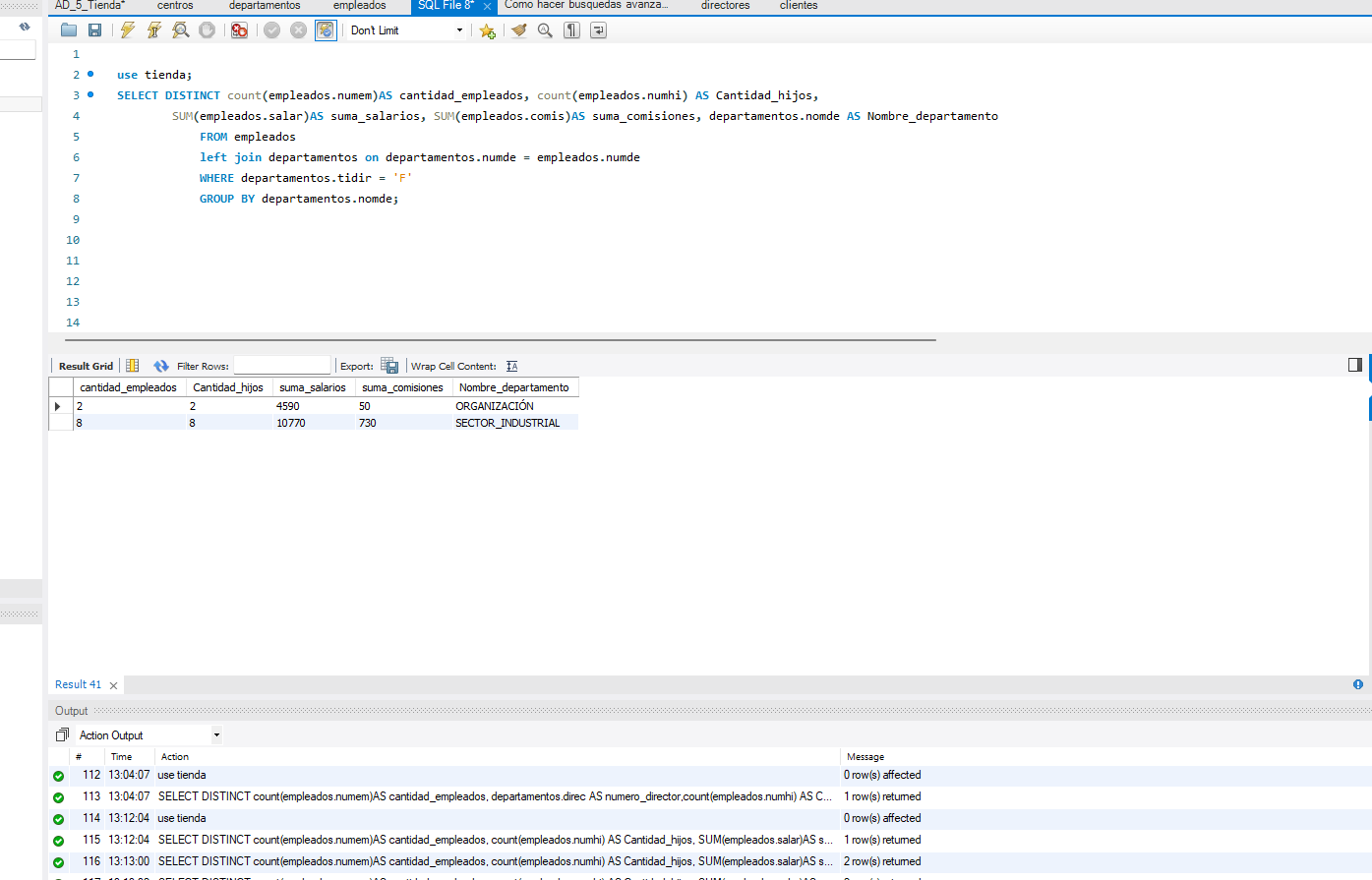
1. Para los departamentos cuyo director lo sea en funciones, hallar el número de empleados y la suma de sus salarios, comisiones y número de hijos.

SELECT departamentos.nomde, count(empleados.numem) as numero\_empleados, Sum(empleados.salar) as suma\_salario, SUM(empleados.comis) as suma\_comision, empleados.numhi

FROM departamentos, empleados

WHERE departamentos.numde = empleados.numde AND departamentos.tidir = 'F'

GROUP BY departamentos.nomde;



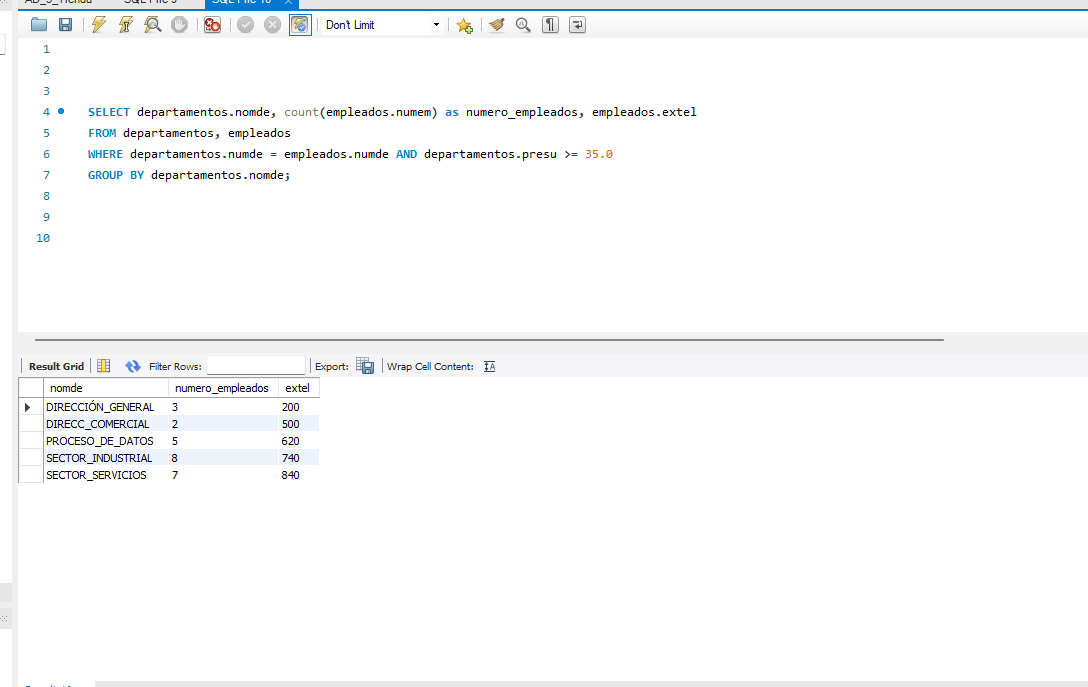
1. Para los departamentos cuyo presupuesto anual supera los 35.000 €, hallar cuantos empleados hay por cada extensión telefónica.

SELECT departamentos.nomde, count(empleados.numem) as numero\_empleados, empleados.extel

FROM departamentos, empleados

WHERE departamentos.numde = empleados.numde AND departamentos.presu >= 35.0

GROUP BY departamentos.nomde;



1. Hallar por orden alfabético los nombres de los empleados y su número de hijos para aquellos que son directores en funciones.

SELECT nomem, numhi

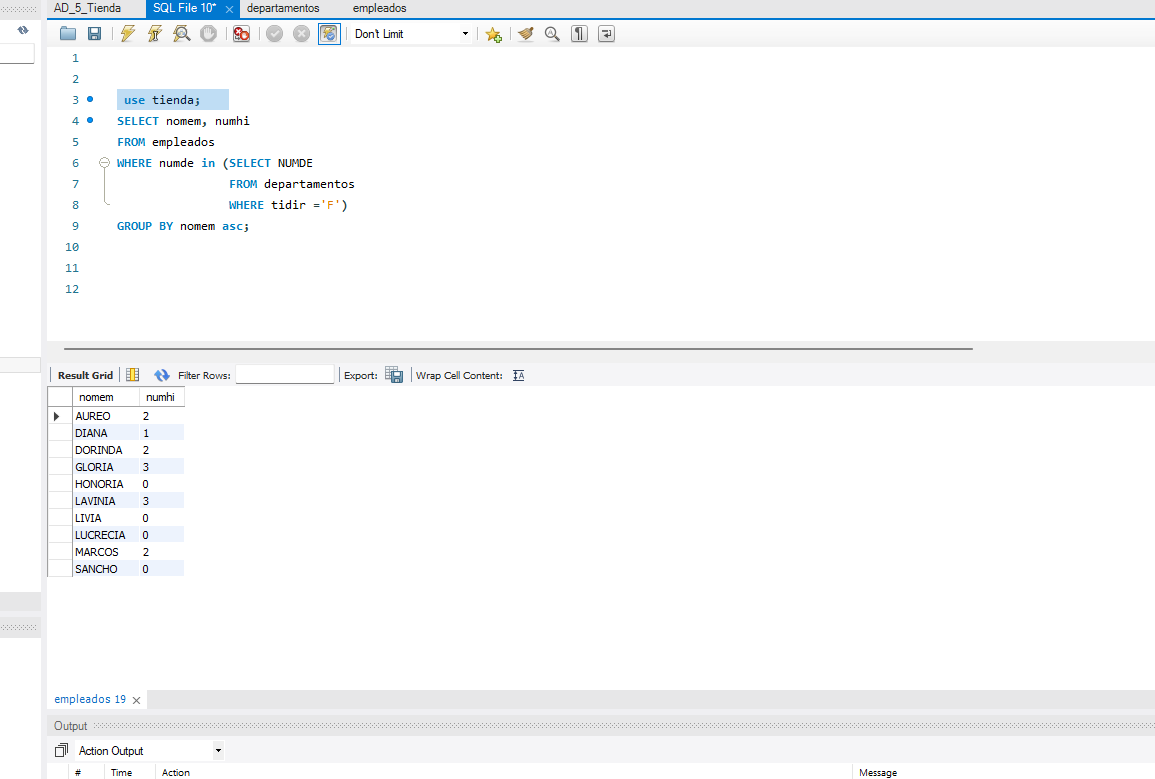
FROM empleados

WHERE numde in (SELECT numde

FROM departamentos

WHERE tidir ='F')

GROUP BY nomem asc;



**QUERIES 7 AL 14: Noelia Villahermosa**

1. Hallar si hay algún departamento (suponemos que sería de reciente creación) que aún no tenga empleados asignados ni director en propiedad.

SELECT departamentos.nomde AS NOMBRE\_DEPARTAMENTO, empleados.nomem AS NOMBRE\_EMPL, departamentos.tidir AS TIPO\_DIRECTOR

FROM departamentos

LEFT JOIN empleados ON departamentos.numde = empleados.numde

WHERE empleados.numde IS NULL AND departamentos.tidir <> 'F';

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

.

.

1. Añadir un nuevo departamento de nombre NUEVO y con director en funciones.

SET FOREIGN\_KEY\_CHECKS = 0;

SET AUTOCOMMIT = 0;

INSERT INTO departamentos (numde, numce, direc, tidir, presu, depde, nomde)

VALUES (140, 10, 160, ‘F’, 72, 100, ‘NUEVO’);

SET FOREIGN\_KEY\_CHECKS = 1;

ROLLBACK;

SELECT \*  
FROM departamentos;

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

1. Añadir un nuevo empleado de nombre NORBERTO y sin departamento asignado. Inventar el resto de los datos.

SET FOREIGN\_KEY\_CHECKS = 0;

SET AUTOCOMMIT = 0;

INSERT INTO empleados (numem, extel, fecna, fecin, salar, comis, numhi, nomem, numde)

VALUES (570, 926, ‘1998-02-15’, ‘2019-02-15’, 1900, NULL, 4, ‘NORBERTO’, NULL);

SET FOREIGN\_KEY\_CHECKS = 1;

ROLLBACK;

SELECT \*

FROM empleados;

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

9B. Como por error he creado dos empleados NORBERTO, borro de la tabla el empleado NORBERTO con fecha de nacimiento 0000-00-00

SET FOREIGN\_KEY\_CHECKS = 0;

DELETE FROM empleados

WHERE numen = 560;

SET FOREIGN\_KEY\_CHECKS = 1;

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

1. Muestra los departamentos que no tienen empleados.

SELECT departamentos.numde AS DEPARTAMENTO, empleados.numde AS EMPLEADOS

FROM departamentos

LEFT JOIN empleados ON departamentos.numde = empleados.numde

WHERE empleados.numde IS NULL;

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

1. Muestra los nombres de departamentos que no tienen empleados haciendo uso la combinación externa LEFT JOIN. Muestra una segunda columna con los nombres de empleados para asegurarnos que realmente está a NULL.

SELECT departamentos.numde AS DEPARTAMENTO, departamentos.nomde AS NOMBRE\_DEPARTAMENTO, empleados.numde AS ID\_EMPLEADO, empleados.nomem AS NOMBRE\_EMPLEADO

FROM departamentos

LEFT JOIN empleados ON departamentos.numde = empleados.numde

WHERE empleados.numde IS NULL;

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

1. Muestra los nombres de departamentos que no tienen empleados haciendo uso la combinación externa RIGH JOIN. Muestra una segunda columna con los nombres de empleados para asegurarnos que realmente está a NULL.

SELECT departamentos.numde AS NUM\_DEPARTAMENTO, departamentos.nomde AS NOMBRE\_DEPARTAMENTO, empleados.numde AS NUM\_EMPLEADO, empleados.nomem AS NOMBRE\_EMPLEADO

FROM empleados

RIGHT JOIN departamentos ON departamentos.numde = empleados.numde

WHERE empleados.numde IS NULL;

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

1. Muestra los nombres de empleados que no tienen departamento haciendo uso la combinación externa LEFT JOIN. Muestra una segunda columna con los nombres de departamentos para asegurarnos que realmente está a NULL.

SELECT departamentos.numde AS NUM\_DEPARTAMENTO, departamentos.nomde AS NOMBRE\_DEPARTAMENTO, empleados.nomem AS NOMBRE\_EMPLEADO

FROM empleados

LEFT JOIN departamentos ON empleados.numde = departamentos.numde

WHERE departamentos.numde IS NULL;

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

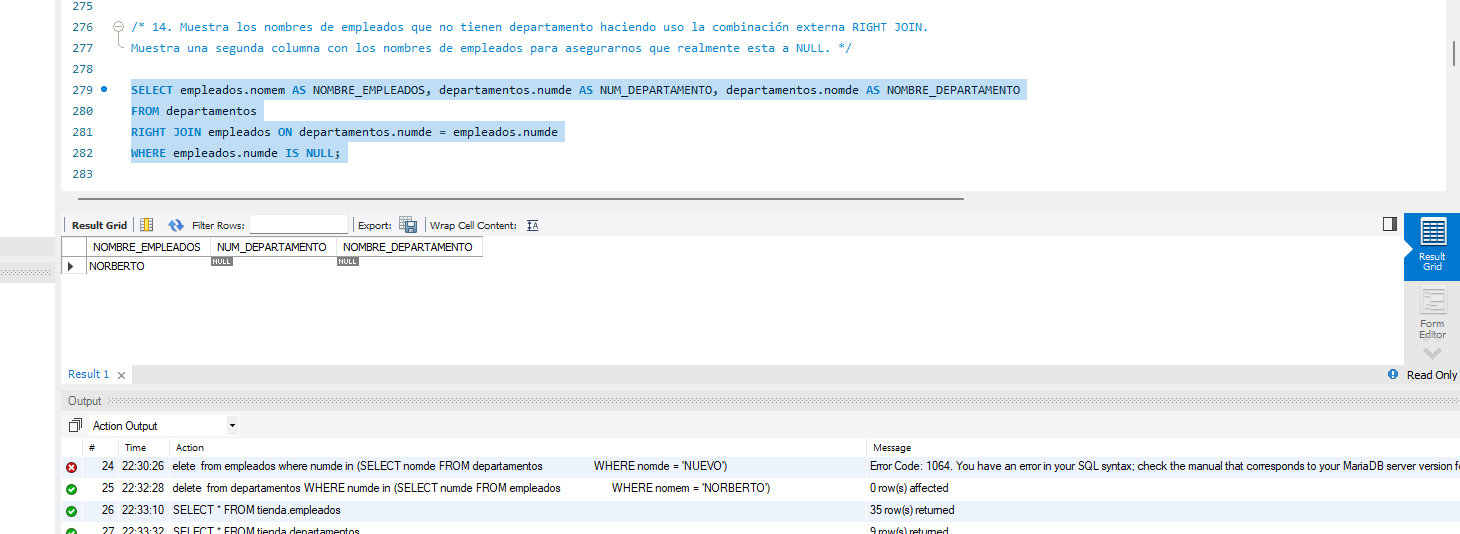
1. Muestra los nombres de empleados que no tienen departamento haciendo uso la combinación externa RIGHT JOIN. Muestra una segunda columna con los nombres de empleados para asegurarnos que realmente está a NULL.

SELECT empleados.nomem AS NOMBRE\_EMPLEADOS, departamentos.numde AS NUM\_DEPARTAMENTO, departamentos.nomde AS NOMBRE\_DEPARTAMENTO

FROM departamentos

RIGHT JOIN empleados ON departamentos.numde = empleados.numde

WHERE empleados.numde IS NULL;



**Queries 15 al 20: María Isabel Martín**

1. Muestra los departamentos que no tienen empleados y los empleados que no tiene departamento haciendo uso la combinación externa FULL JOIN.

/\*

Por problemas de compatibilidad, en muchos casos MySQL no acepta el full join, cuyo esquema es el siguiente:

SELECT columns

FROM table1

FULL [OUTER] JOIN table2

ON table1.column = table2.column;

Para solucionarlo, interesa aplicar la unión de un left join con un right join, como se va a desarrollar a continuación:

\*/

SELECT \*

FROM departamentos LEFT JOIN empleados

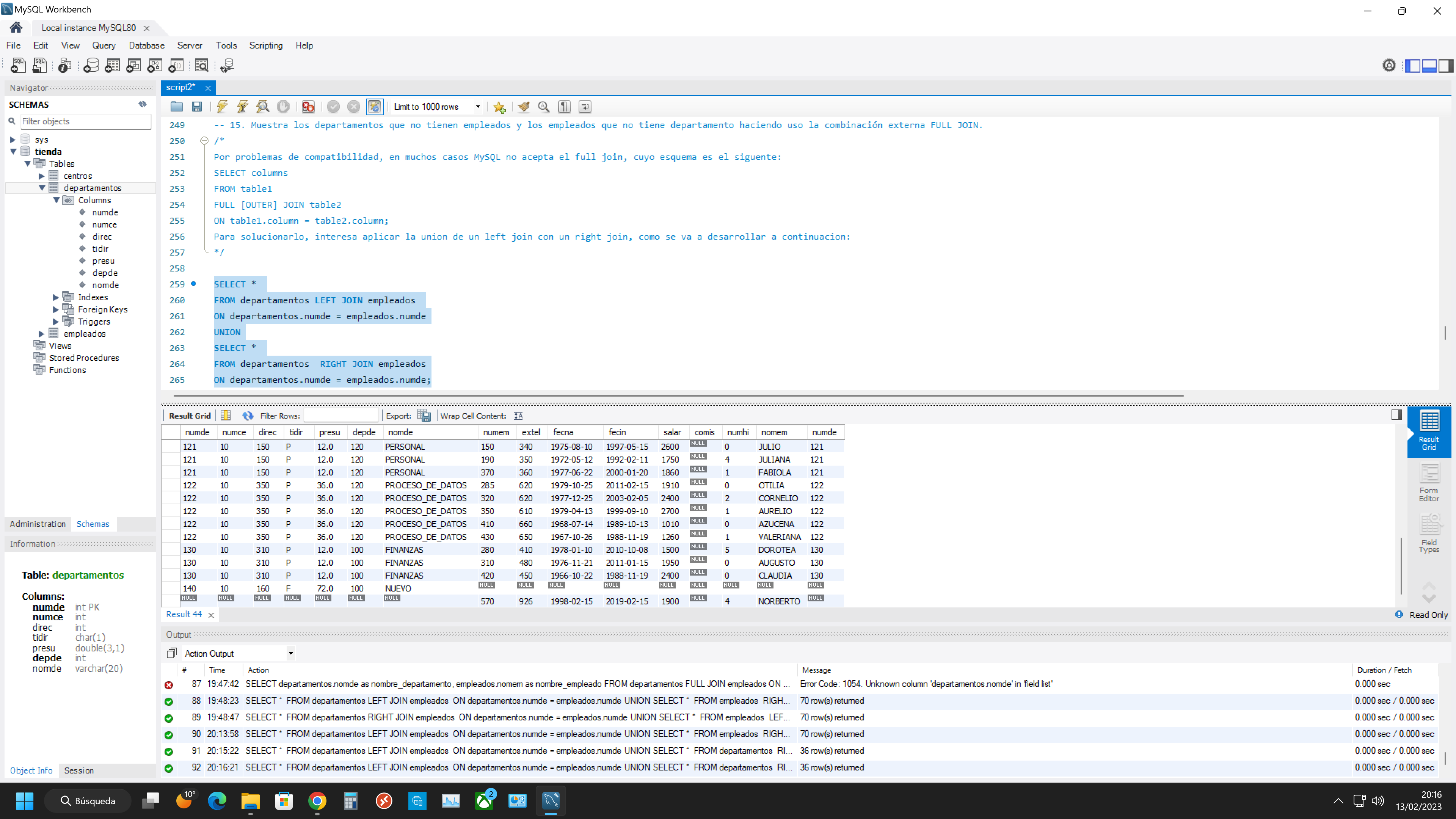
ON departamentos.numde = empleados.numde

UNION

SELECT \*

FROM departamentos RIGHT JOIN empleados

partamentos.numde = empleados.numde;



1. Muestra los empleados y sus respectivos departamentos haciendo uso de la combinación interna INNER JOIN. ¿Aparecen el departamento NUEVO y el empleado NORBERTO?¿Por qué?

SELECT departamentos.nomde AS nombre\_departamento, empleados.nomem AS nombre\_empleado

FROM empleados

INNER JOIN departamentos

ON departamentos.numde = empleados.numde;

/\*

No aparecen ni el departamento NUEVO ni NORBERTO ya que INNER JOIN a la hora de mostrar por pantalla el resultado de la ejecución suele excluir filas en las que hay datos nulos en alguna columna correspondiente.

Los valores nulos no contarían con el mismo tratamiento de igualdad que el resto de datos.

\*/



1. Realiza la misma consulta anterior donde se cumpla la condición que NUMDE está a NULL. ¿Aparece algún resultado?¿Por qué?

SELECT departamentos.nomde AS nombre\_departamento, empleados.nomem AS nombre\_empleado

FROM empleados

INNER JOIN departamentos

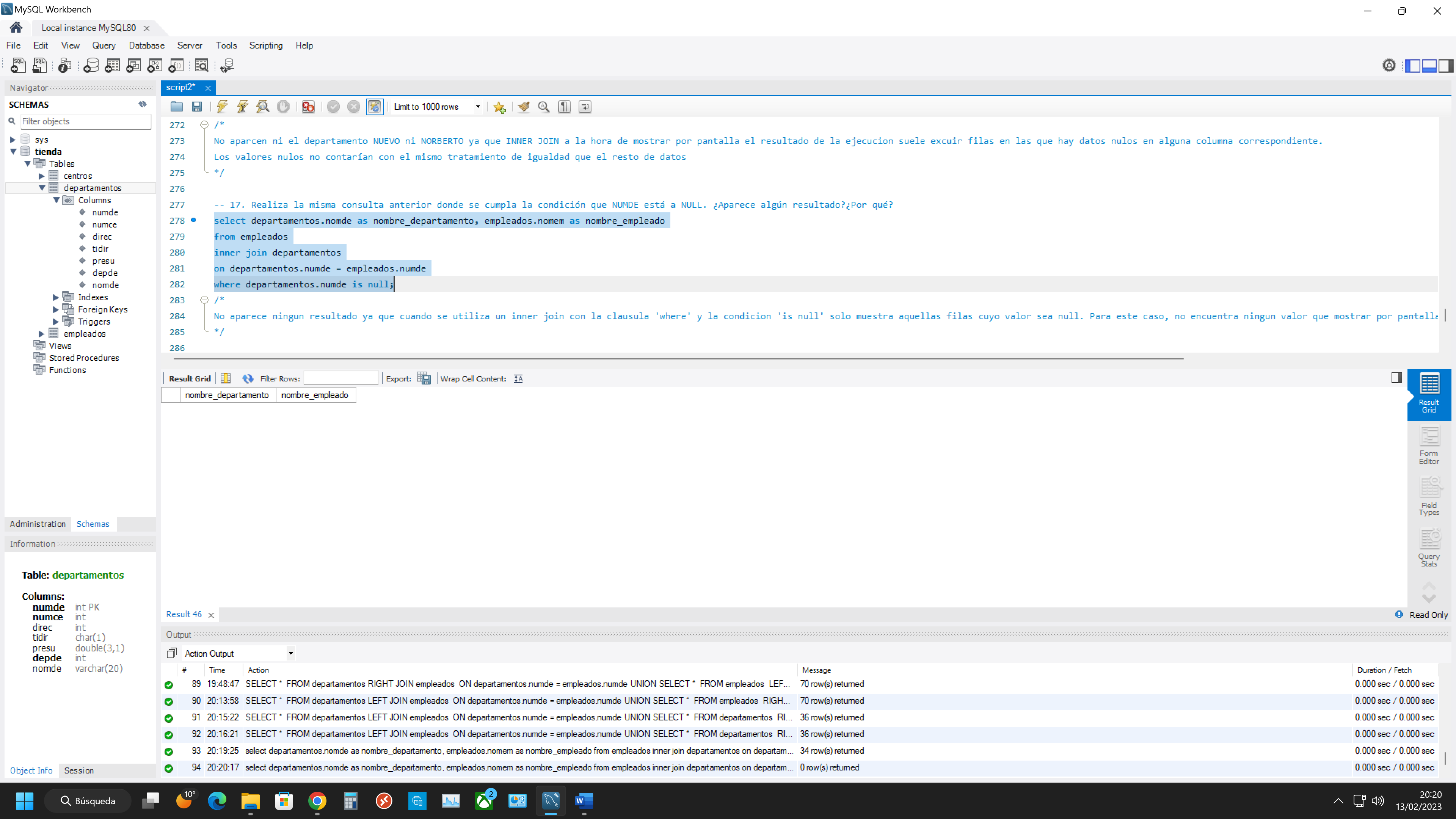
ON departamentos.numde = empleados.numde

WHERE departamentos.numde is null;

/\*

No aparece ningún resultado ya que cuando se utiliza un inner join con la cláusula 'where' y la condición 'is null' solo muestra aquellas filas cuyo valor sea null. Para este caso, no encuentra ningún valor que mostrar por pantalla.

\*/

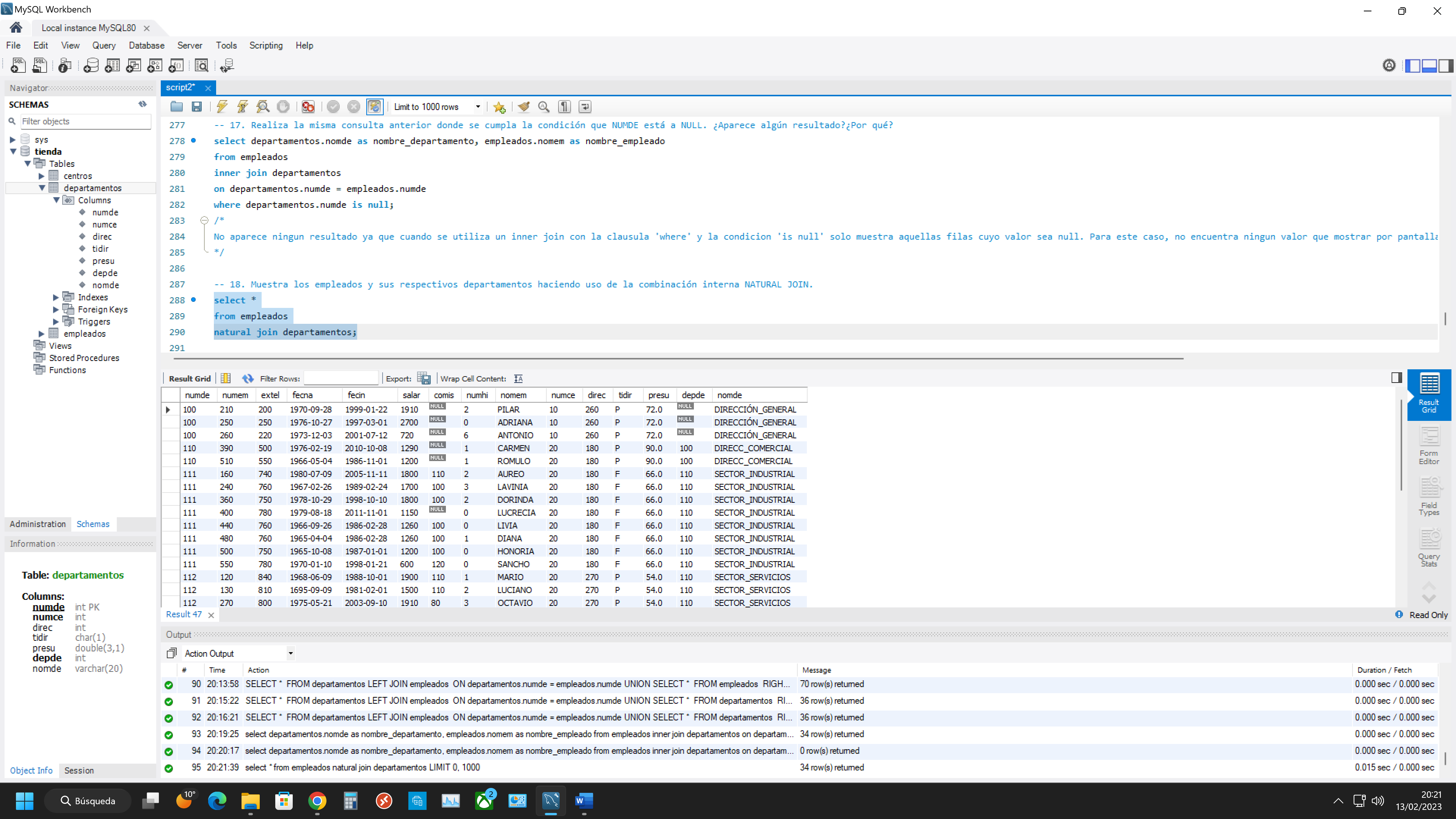


1. Muestra los empleados y sus respectivos departamentos haciendo uso de la combinación interna NATURAL JOIN.

SELECT \*

FROM empleados

NATURAL JOIN departamentos;



Con el natural join se consigue que en una misma tabla resultante se incorporen todas las columnas de las tablas escogidas. En el caso de que alguna se repite solo se muestra una. En este caso se cuentan 15 columnas.

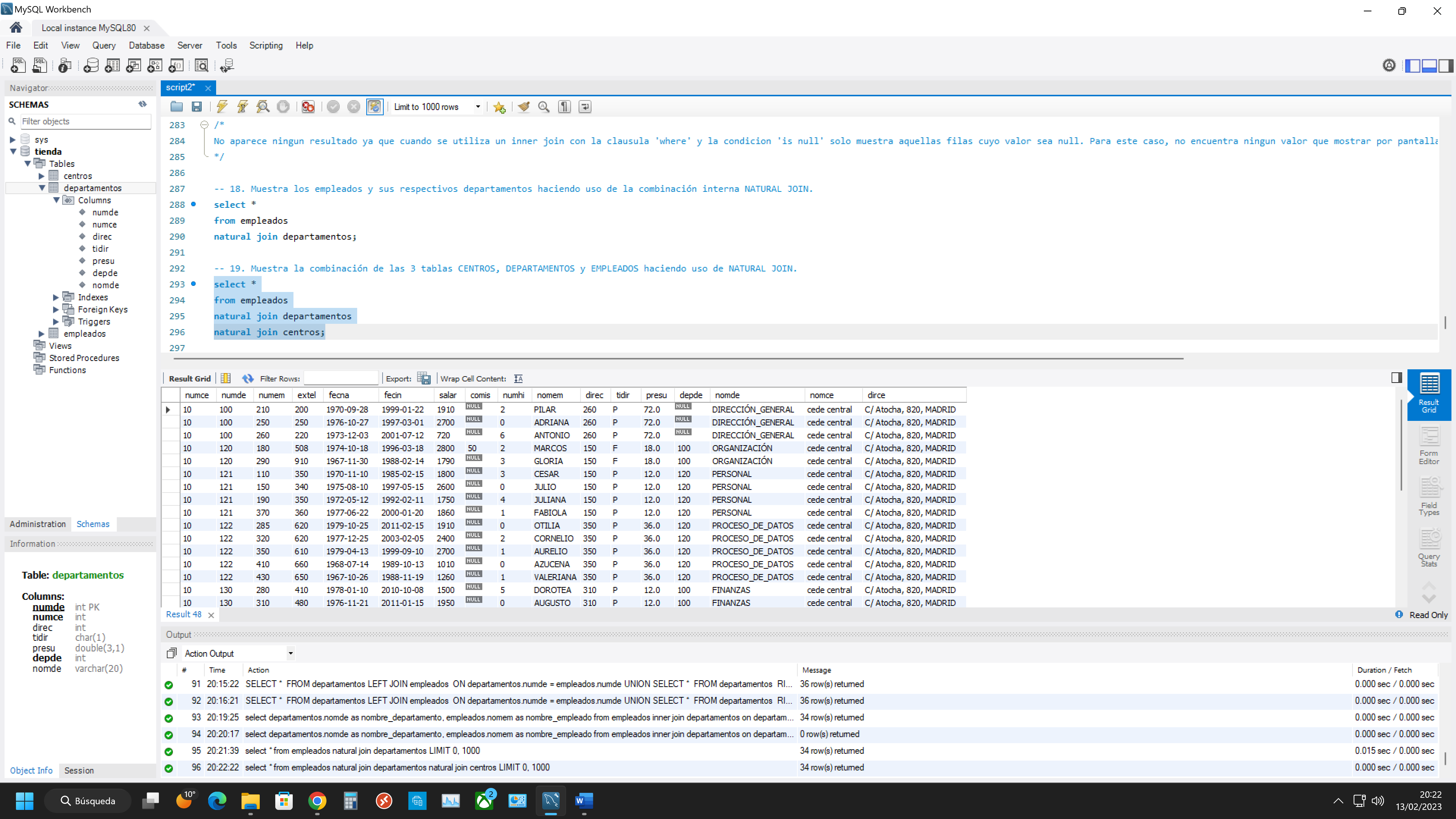
1. Muestra la combinación de las 3 tablas CENTROS, DEPARTAMENTOS y EMPLEADOS haciendo uso de NATURAL JOIN.

SELECT\*

FROM empleados

NATURAL JOIN departamentos

NATURAL JOIN centros;



Para este natural join se añade también la tabla ‘centros’. La consecuencia es que la tabla resultante es la misma que en la query anterior pero se añaden dos columnas más que figuran solo en la tabla ‘centros’.

1. Borra los registros dados de alta para el departamento NUEVO y el empleado introducido en el apartado anterior.

SET FOREIGN\_KEY\_CHECKS = 0;

DELETE

FROM empleados

WHERE nomem = 'NORBERTO';

SET FOREIGN\_KEY\_CHECKS = 1;

SET FOREIGN\_KEY\_CHECKS = 0;

DELETE

FROM departamentos

WHERE nomde = 'NUEVO';

SET FOREIGN\_KEY\_CHECKS = 1;

SELECT \*

FROM empleados, departamentos

WHERE empleados.numde = departamentos.numde

GROUP BY departamentos.nomde;

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Tras realizar los ‘delete’ se comprueba con select que, efectivamente, se han eliminado esos registros.