Instituto Politécnico Nacional

Escuela Superior de Cómputo

Bases de Datos

Práctica no. 4:

Consultas múltiples (Join)

Profesor: Euler Hernández Contreras

Alumno: Calva Hernández José Manuel

Grupo: 2CM12

## Índice

[Índice 2](#_Toc496732528)

[Marco Teórico 3](#_Toc496732529)

[Instrucciones 6](#_Toc496732530)

[Screenshots 10](#_Toc496732531)

[Conclusiones 15](#_Toc496732532)

[Referencias 15](#_Toc496732533)

## Marco Teórico

**JOIN**

El concepto de tabla concatenada (o relación concatenada) se incorporó a SQL para poder especificar una tabla como resultado de una operación de concatenación en la cláusula from de una consulta. Esta estructura es más fácil que mezclar todas las condiciones de selección y concatenación en la cláusula WHERE. Por ejemplo, considere la consulta siguiente, que recupera el nombre y la dirección de los empleados que trabajan para el departamento 'Investigación'. Puede ser más fácil especificar primero la concatenación de las relaciones EMPLEADO y DEPARTAMENTO, Y después seleccionar las tuplas y los atributos deseados. Esto se puede escribir en SQL como:

SELECT Nombre, Apellid01, Dirección

FROM (EMPLEADO JOIN DEPARTAMENTO ON Dno=NumeroDpto)

WHERE NombreDpto='lnvestigación';

La cláusula FROM contiene una sola tabla concatenada. Los atributos de dicha tabla son todos los atributos de la primera tabla, EMPLEADO, seguidos por todos los atributos de la segunda tabla, DEPARTAMENTO. El concepto de tabla concatenada también permite especificar tipos diferentes de concatenación, como NATURAL JOIN y varios tipos de OUTER JOIN. En una concatenación natural (NATURAL JOIN) sobre las relaciones R y S, no se especifica condición de concatenación alguna; se crea una condición EQUIJOIN implícita para cada par de atributos con el mismo nombre que en R y S. Cada par de atributos se incluye una sola vez en la relación.

La EQUIJOIN (equicombinación) de tablas. Para formar la equicombinación, entonces, comience con dos tablas que tengan una o más columnas en común. Compare cada tupla de la primera con cada tupla de la segunda y elija sólo aquellas concatenaciones en las que los valores en las columnas comunes sean iguales.

Usted puede notar que, por definición, en una equicombinación, siempre se tienen al menos dos columnas idénticas. Puesto que parece innecesario incluir la columna repetida, se le puede eliminar y definir una NATURAL JOIN (combinación natural) como una equicombinación en la que se elimina la columna repetida. Ésta es la forma más común de la operación JOIN (combinación).

Si los nombres de los atributos de concatenación no coinciden con los de las relaciones base, es posible renombrarlos para que coincidan, y después aplicar NATURAL JOIN. En este caso, se puede utilizar la estructura AS para renombrar una relación y todos sus atributos en la cláusula FROM. Es lo que se ilustra en la siguiente consulta, donde la relación DEPARTAMENTO se renombra como DEPT y sus atributos se renombran como NombreDpto, Dno (para que coincida con el nombre del atributo de concatenación deseado Dno de EMPLEADO), DniDelDirector y FechalnDirector. La condición de concatenación implicada en esta NATURAL JOIN es EMPLEADO.Dno=DEPT.Dno porque es el único par de atributos con el mismo nombre después de haber renombrado:

SELECT Nombre, Apellid01, Dirección

FROM (EMPLEADO NATURAL JOIN

(DEPARTAMENTO AS DEPT (NombreDpto, Dno, DniDelDirector, FechalnDirector)))

WHERE NombreDpto='lnvestigación';

El tipo predeterminado de concatenación en una tabla concatenada es una concatenación interna, en la que una tupla se incluye en el resultado si en la otra relación existe una tupla coincidente. Por ejemplo, en la consulta siguiente, en el resultado sólo se incluyen los empleados que tienen un supervisor; se excluyen las tuplas EMPLEADO cuyo valor para SuperDni es NULL. Si se necesita que se incluyan todos los empleados, debe utilizarse explícitamente una OUTER JOIN. En SQL, esto se manipula especificando explícitamente la OUTER JOIN en una tabla concatenada, como se ilustra:

SELECT E.Apellid01 AS NombreDeEmpleado,

S.Apellid01 AS NombreDeSupervisor

FROM (EMPLEADO AS E LEFT OUTER JOIN EMPLEADO AS S

ON E.SuperDni=S.Dni);

Se pueden definir muchos otros tipos de operadores combinación. Una variación es la SEMIJOIN (semicombinación) de dos tablas. Si A y B son tablas, entonces la semicombinación izquierda (left-semijoin) A |x B se encuentra al tomar la combinación natural de A y B y luego proyectar el resultado en los atributos de A.

Otro tipo de operación combinación es la OUTERJOIN (combinación exterior). Esta operación es una extensión de una operación THETA JOIN, una EQUIJOIN o una NATURAL JOIN. Cuando se forma cualquiera de estas combinaciones, cualquier tupla de una de las tablas originales para las que no hay coincidencia en la segunda tabla no entra al resultado.

Una variación de la equicombinación exterior mostrada es una LEFT-OUTER-EQUIJOIN (equicombinación-exterior-izquierda), lo que significa que sólo las filas sin coincidencia de la primera tabla (izquierda) aparecen en el resultado. En una RIGHT-OUTER-EQUIJOIN (equicombinación-exterior-derecha) se incluyen las filas sin coincidencia de la segunda tabla (derecha). También se puede definir la combinación theta exterior general, la combinación theta exterior izquierda y la combinación theta exterior derecha en una forma similar.

La combinación natural exterior es similar a la equicombinación exterior, excepto que se eliminan las columnas repetidas, como es usual para una combinación natural. Si las filas combinadas tienen un valor no nulo igual para una columna repetida, se usa dicho valor en la columna común. Si ambas tienen valores nulos, se usa uno nulo, y si una tiene un valor nulo y la otra no, se usa el valor no nulo en el resultado.

**IN**

Algunas consultas requieren obtener valores existentes en la base de datos para usarlos después en una condición de comparación. Dichas consultas se pueden formular convenientemente mediante consultas anidadas, que son bloques select-from-where completos dentro de la cláusula WHERE de otra consulta. Esa otra consulta es la que se conoce como consulta externa, donde se introduce el operador de comparación IN, que compara un valor v con un conjunto (o multiconjunto) de valores V y se evalúa como TRUE si ves uno de los elementos de V.

En la consulta externa, utilizamos OR para recuperar una tupla PROYECTO si el valor NUMPROYECTO de esa tupla se encuentra en el resultado de cualquier consulta anidada. Si una consulta anidada devuelve un solo atributo y una sola tupla, el resultado de la consulta será un solo valor (escalar). En estos casos, está permitido utilizar = en lugar de IN como operador de comparación. En general, la consulta anidada devolverá una tabla (relación), que es un conjunto o un multiconjunto de tuplas. SQL permite el uso de tuplas de valores en las comparaciones colocándolas entre paréntesis.

**GROUP BY**

En muchos casos queremos aplicar las funciones agregadas a subgrupos de tuplas de una relación, estando los subgrupos basados en algunos valores de atributo. En estos casos, tenemos que dividir la relación en subconjuntos no solapados (o grupos) de tuplas. Cada grupo (partición) estará compuesto por las tuplas que tienen el mismo valor para algún(os) atributo(s), denominado(s) atributo(s) de agrupamiento. Después podemos aplicar la función independientemente a cada -grupo. SQL tiene una cláusula GROUP BY para este propósito. Esta cláusula especifica los atributos de agrupamiento, que también deben aparecer en la cláusula SELECT, por lo que el valor resultante de aplicar la función de agregación a un grupo de rupias aparece junto con el valor de los atributos de agrupamiento.

## Instrucciones

Cargar el script

create database tt;

use tt;

source ...

Resolver las siguientes consultas

1. Concoer el no de tt, de aquellos tts que ha dirigido el dr.

Select d.noTT

from dirige d, profesor p

where d.idProf=p.idProf

and p.nombre like "Andr\_s%"

and p.apPaterno like "Orti%"

order by 1;

2. Cuántos tts ha dirigido Dr. Flavio.

Select COUNT(\*)

from dirige d, profesor p

where d.idProf=p.idProf

and p.nombre like "Flavio%"

and p.apPaterno like "S\_nc%"

order by 1;

Select COUNT(\*)

from dirige d, profesor p

where d.idProf=p.idProf

and p.nombre like "Idalia%"

and p.apPaterno like "Maldon%"

order by 1;

3. Cuál es el título de los tts de la consulta anterior.

Select t.titulo

from dirige d, profesor p, tt t

where d.idProf=p.idProf

and d.nott=t.nott

and p.nombre like "Flavio%"

and p.apPaterno like "S\_nc%"

order by 1;

Select t.titulo

from dirige d, profesor p, tt t

where d.idProf=p.idProf

and d.nott=t.nott

and p.nombre like "Ignacio%"

and p.apPaterno like "R\_os%"

order by 1;

4. Qué número de tt tienen aquellos tts que se han presentado en el año 2008.

Select noTT, fecha

from presentacion

where fecha between "2008-01-01" and "2008-12-31"

order by 2, 1;

5. Mostrar el tipo de tt que ha dirigido Euler.

select t.nott, x.tipo

from tt t, dirige d, presentacion x, profesor p

where x.nott=t.nott

and t.nott=d.nott

and d.idProf=p.idProf

and p.nombre like "Euler%"

and p.apPaterno like "Hern%"

order by 2, 1;

6. Qué grado de estudios tienen los profesores que se apellidan García.

Select p.\*, ge.descripcion

from profesor p, gradoProf gp, gradoEstudios ge

where p.idProf=gp.idProf

and gp.idGrado=ge.idGrado

and (p.apPaterno like "Garc\_a%" or p.apMaterno like "Garc\_a%")

order by p.apPaterno;

7. Qué profesores han hecho estudios en la UNAM.

Select p.\*, gp.institucion

from profesor p, gradoProf gp

where p.idProf=gp.idProf

and gp.institucion like "UNAM%"

order by p.apPaterno;

8. Mostrar el no. de tt y el tipo, además del dictamen de aquellos tts, donde la Dra. Lorena Chavarria ha sido sinodal.

Select t.nott, x.tipo

from presentacion x, tt t, sinodalia s, profesor p

where x.nott=t.nott

and t.nott=s.nott

and (s.s1=p.idprof

or s.s2=p.idprof

or s.s3=p.idprof)

and p.nombre like "Euler%"

and p.apPaterno like "Hern%"

order by 2, 1;

9. Mostrar el no. de tt y la fecha de presentación de aquellos tts que incluyen la palabra "redes neuronales".

Select p.nott, p.fecha, t.titulo

from presentacion p, tt t

where p.nott=t.nott

and t.titulo like "%redes neuronales%"

order by 2, 1;

10. Mostrar el no. de tt y el nombre de los directores que han dirigido tts remediales.

Select t.nott, x.tipo, p.nombre, p.apPaterno, p.apMaterno

from presentacion x, tt t, dirige d, profesor p

where x.nott=t.nott

and t.nott=d.nott

and d.idProf=p.idProf

and x.tipo like "%R%"

order by p.apPaterno, 1;

11. Mostrar la cédulo profesional y la institución de aquellos profesores que tienen grado de maestría

Select gp.cedula, ge.descripcion, gp.institucion

from gradoProf gp, gradoEstudios ge

where gp.idGrado=ge.idGrado

and ge.descripcion like "%maestr\_a%"

order by 1;

12. Mostrar el no. de tt, la calificación de los sinodales, donde el revisor ha sido la Dra. Fabiola Ocampo

Select t.nott, x.califSinodales, p.nombre, p.apPaterno

from presentacion x, tt t, sinodalia s, profesor p

where x.nott=t.nott

and t.nott=s.nott

and s.revisor=p.idProf

and p.nombre like "Fabiola%"

and p.apPaterno like "Ocampo%"

order by 2, 1;

13. Mostrar los sinodales que han tenido los siguientes tts 2000-0209, 06-1-0174

Select p.nombre, p.apPaterno, p.apMaterno, s.noTT

from profesor p, sinodalia s

where (s.s1=p.idprof

or s.s2=p.idprof

or s.s3=p.idprof)

and s.nott in("2000-0209","06-1-0174")

order by s.noTT, p.apPaterno;

Select p.nombre, p.apPaterno, p.apMaterno, s.noTT

from profesor p, sinodalia s

where s.revisor=p.idProf

and s.nott in("09-2-0277")

order by s.noTT, p.apPaterno;

14. Quién fue el revisor del tt que ha dirigido Idalia Maldonado

Select d.nott, p.\*

from dirige d, profesor p, sinodalia s

where s.revisor=p.idprof

and (Select idProf from profesor where nombre like "Euler%" and appaterno like "Hern%") = d.idprof

order by 1;

15. Cuál es la fecha de presentación de aquellos tts que ha sido sinodal el profesor Ulises Vélez

Select x.fecha, t.nott, p.nombre, p.apPaterno

from presentacion x, tt t, sinodalia s, profesor p

where x.nott=t.nott

and t.nott=s.nott

and (s.s1=p.idProf

or s.s2=p.idProf

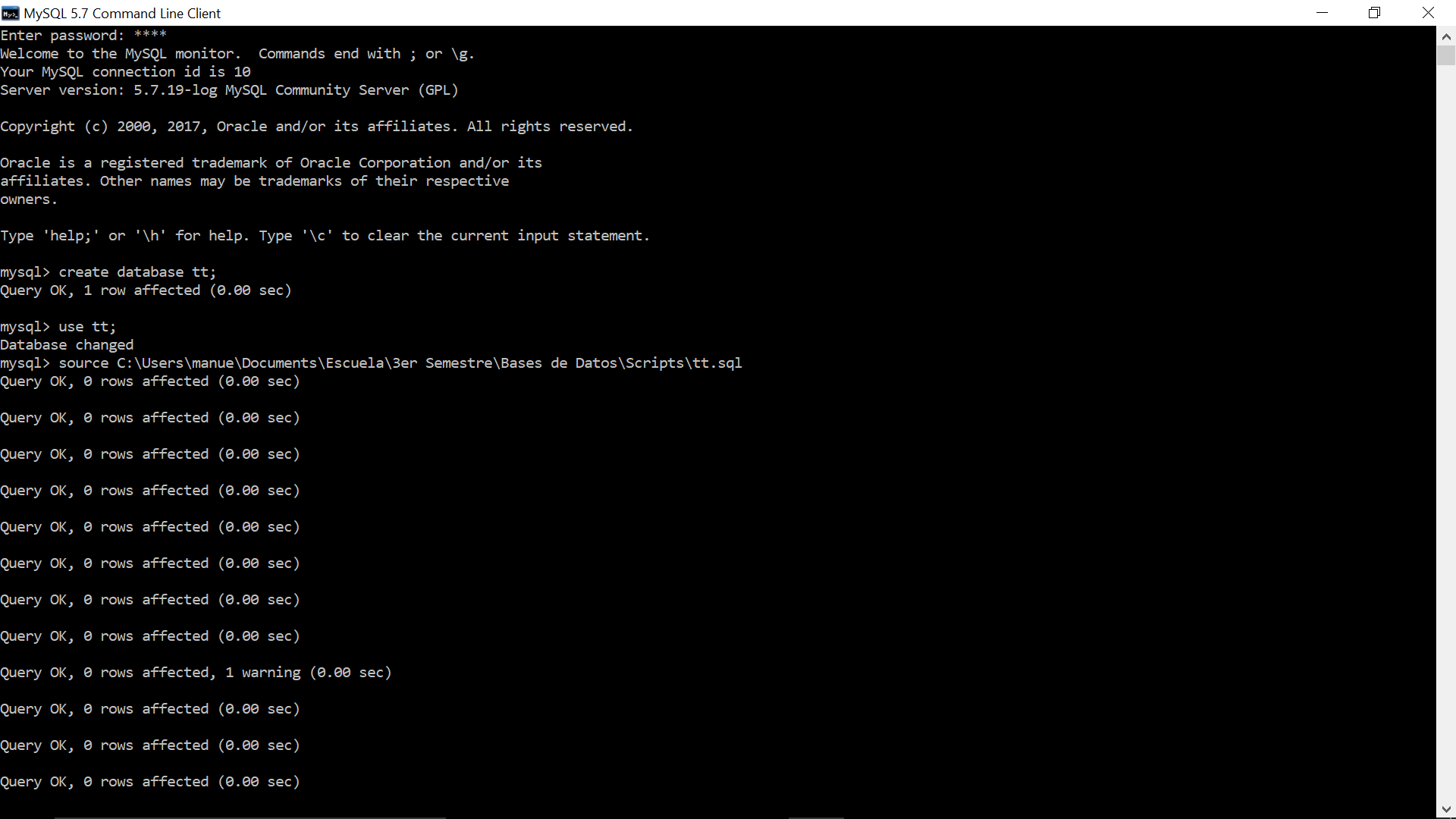
or s.s3=p.idProf)

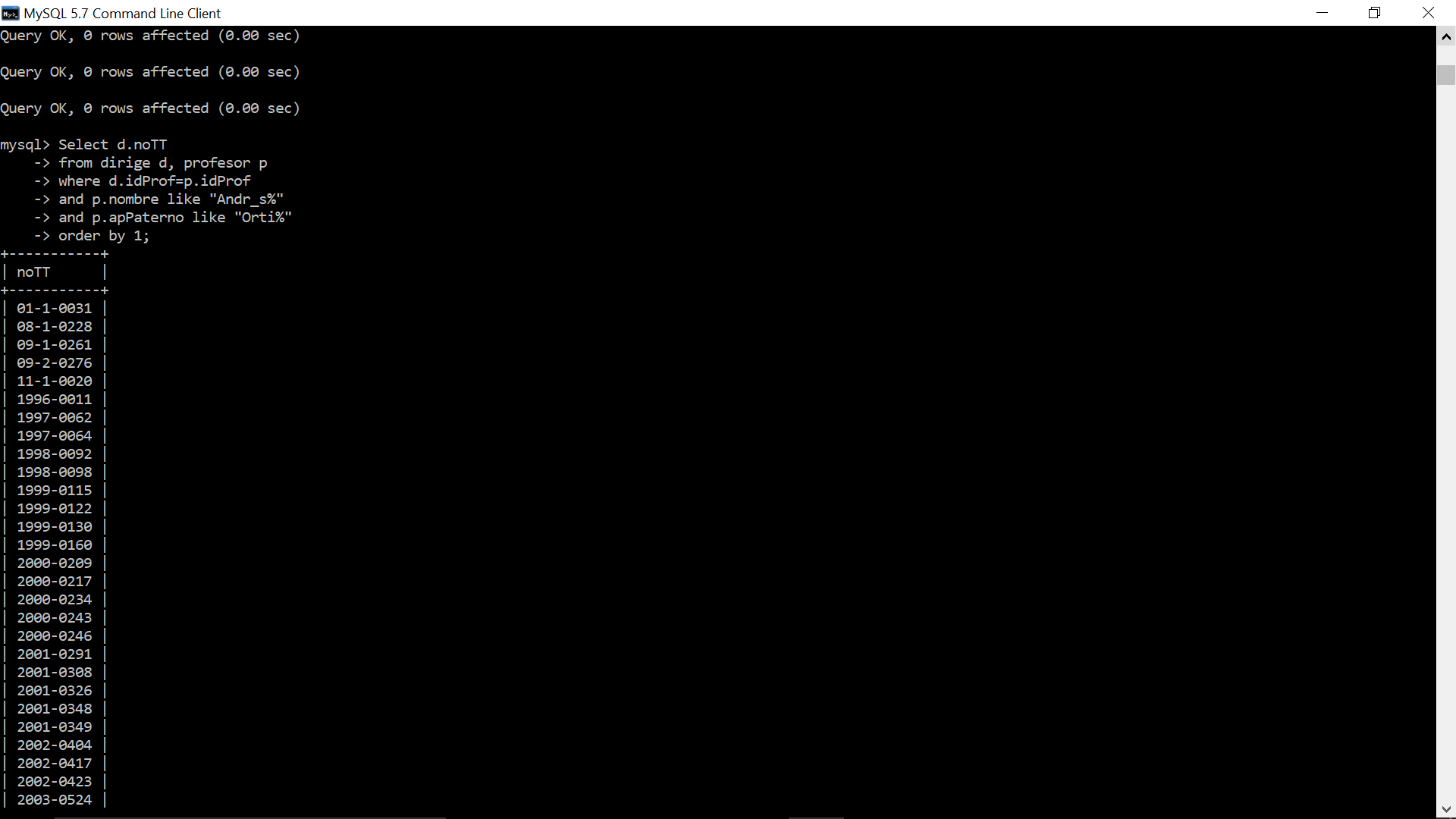
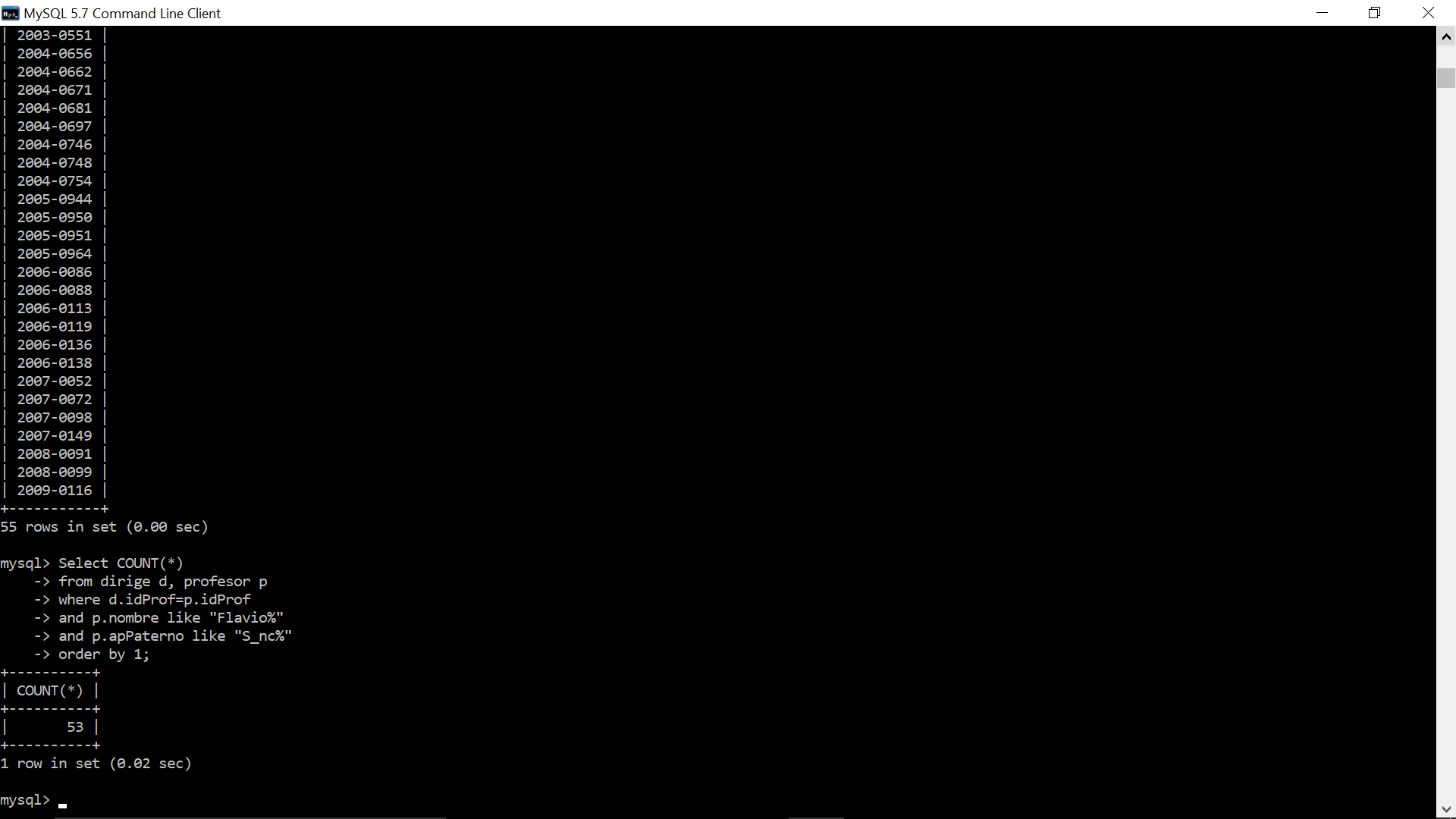
and p.nombre like "Ulises%"

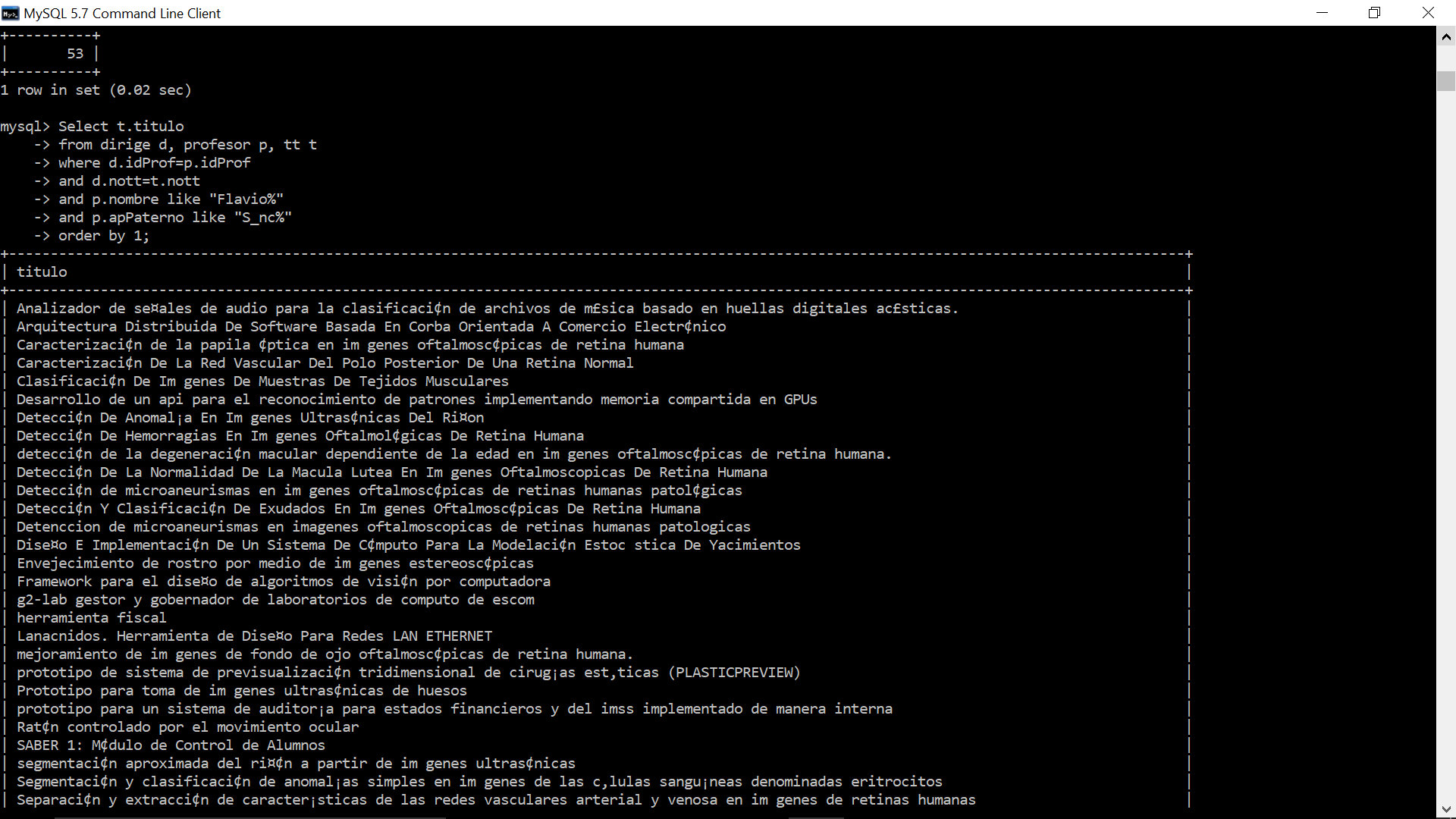
and p.apPaterno like "V\_lez%"

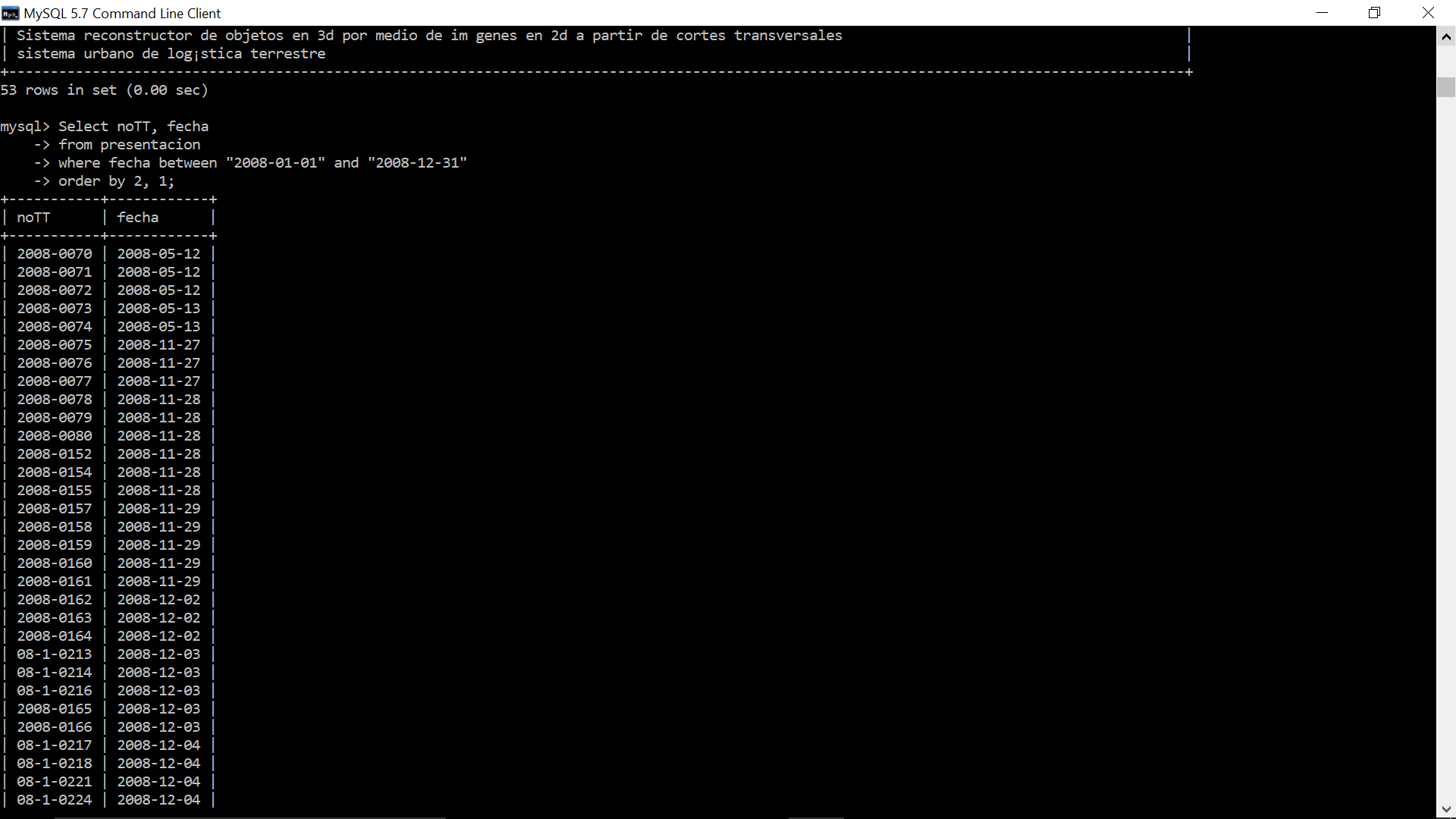
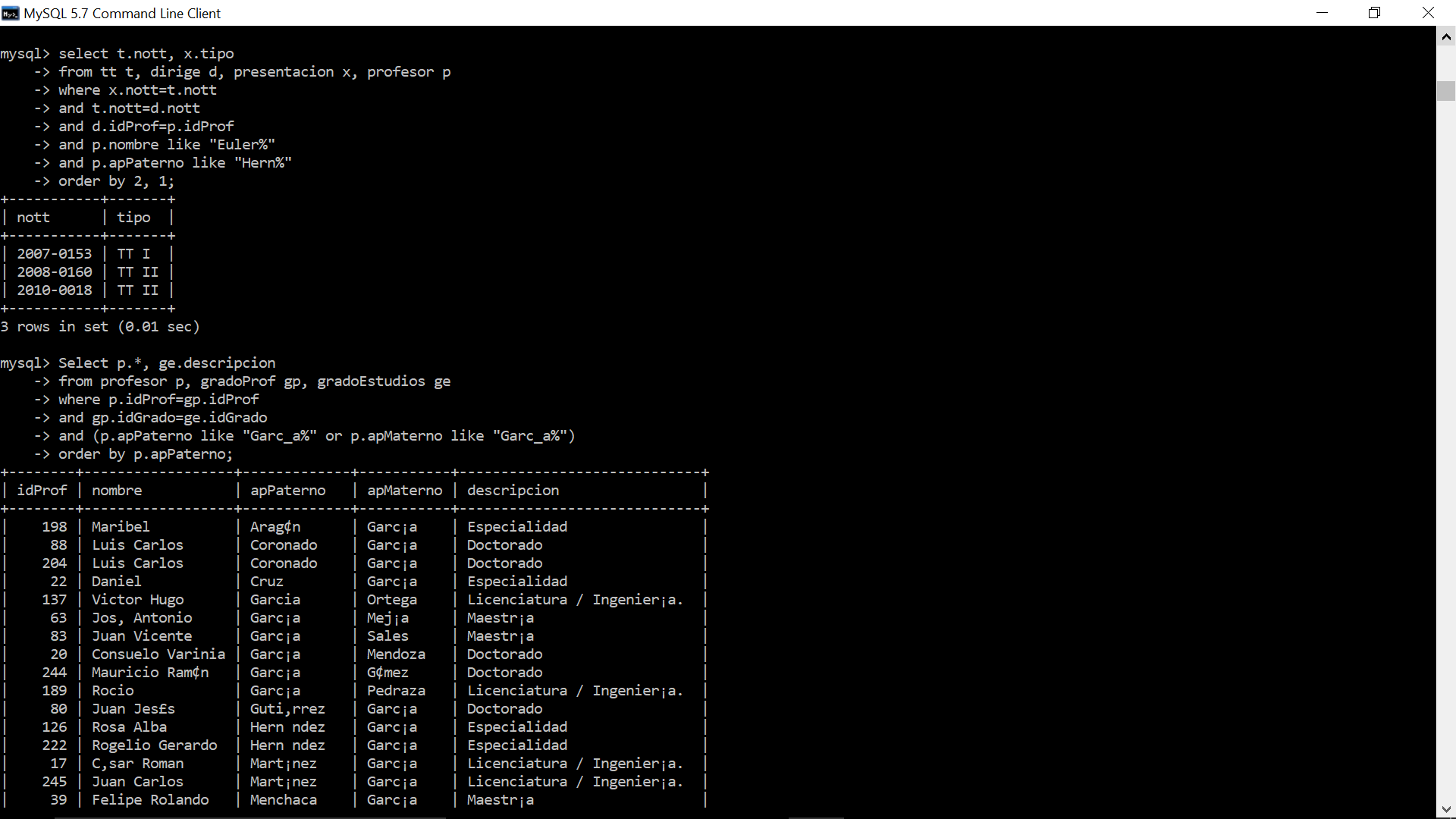
order by 2, 1;

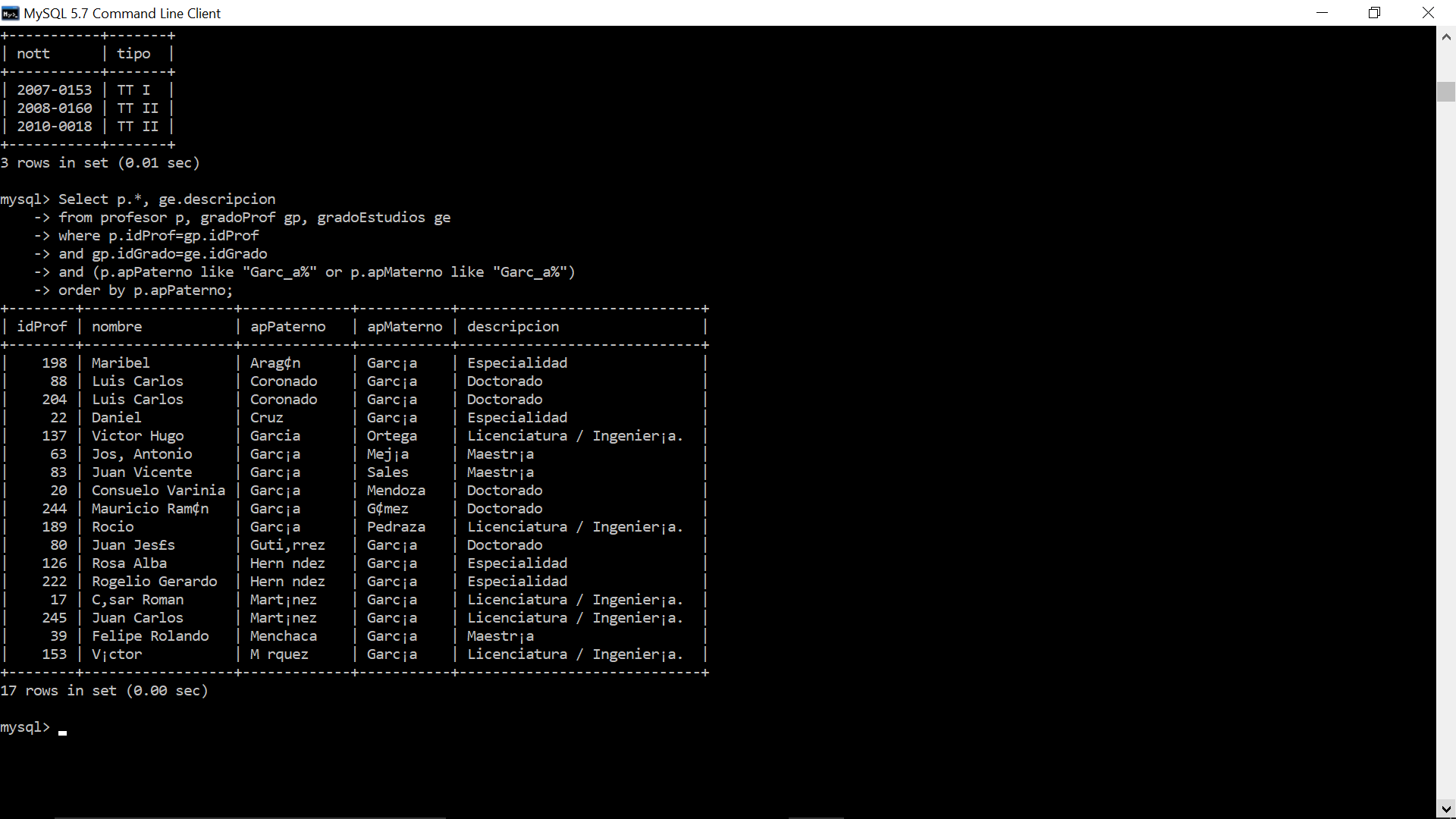
## Screenshots

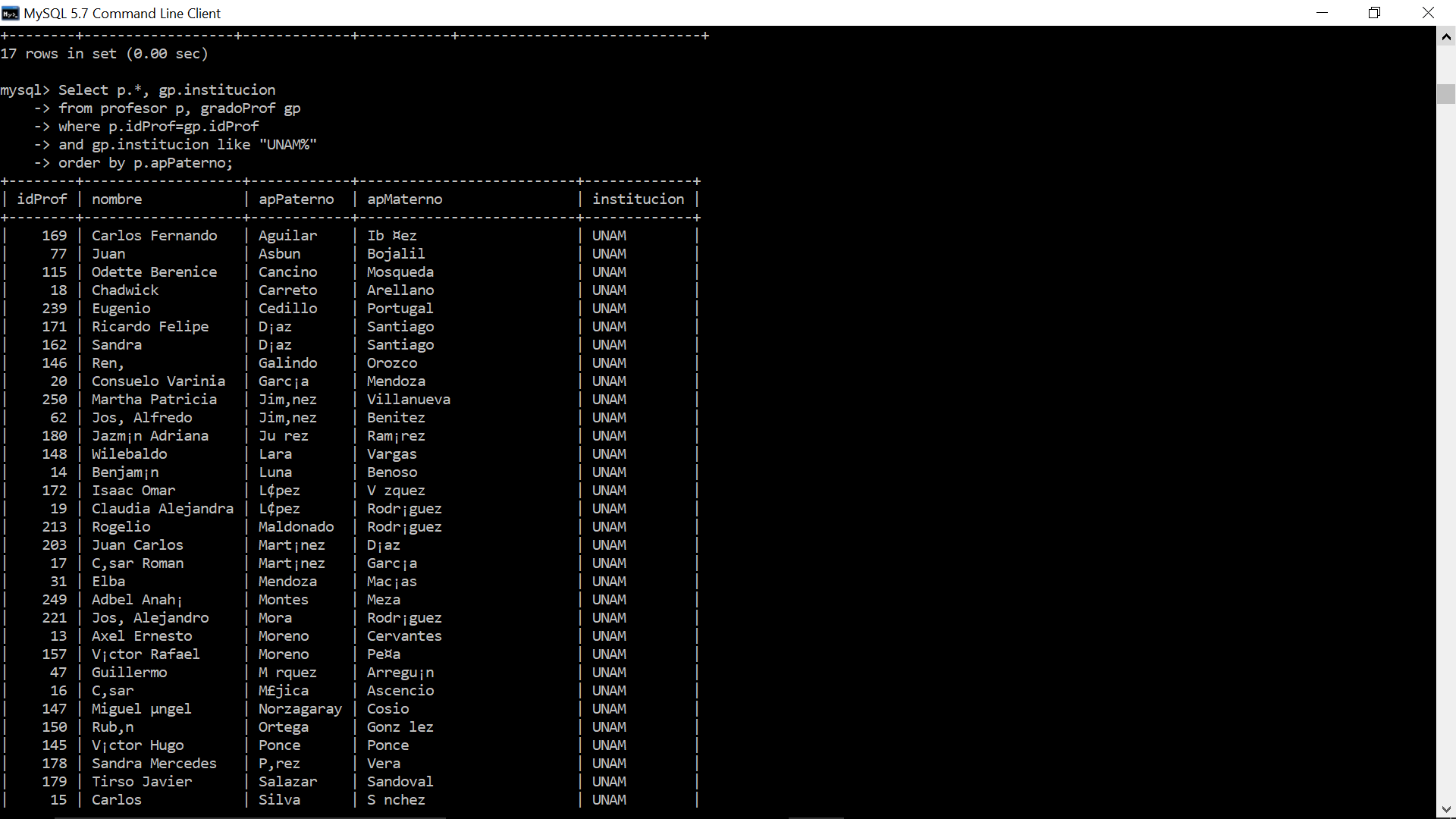


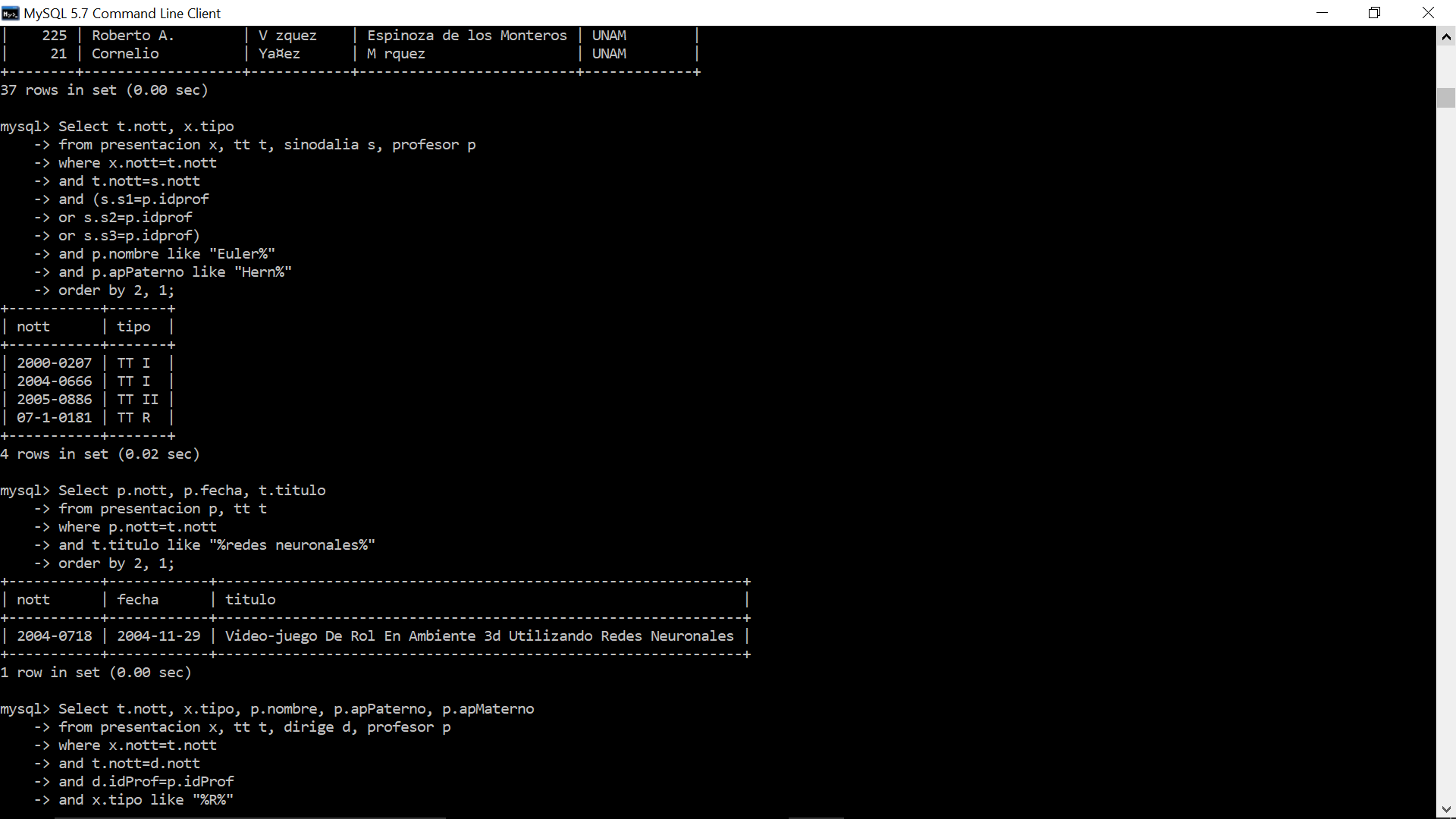
 

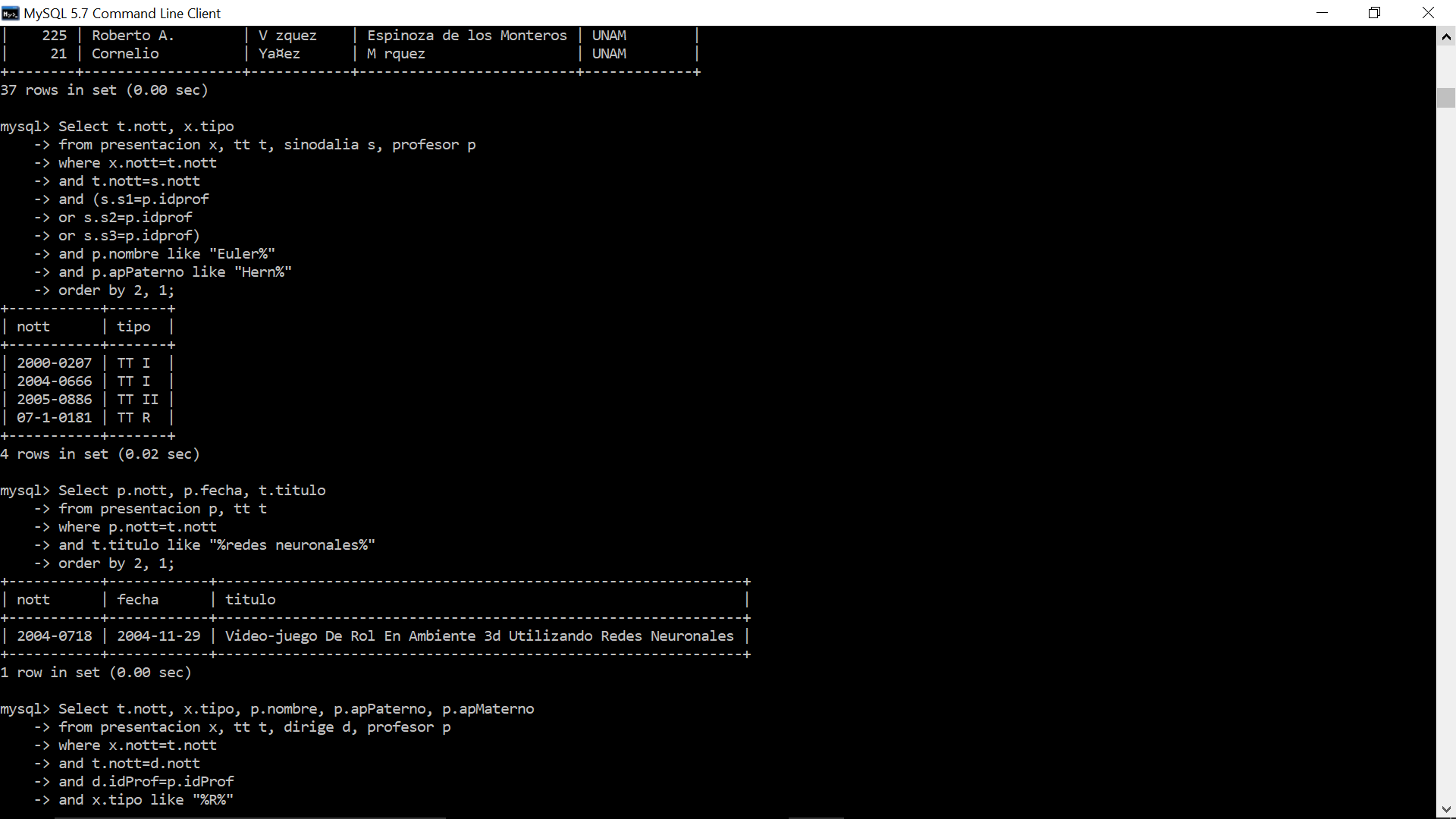


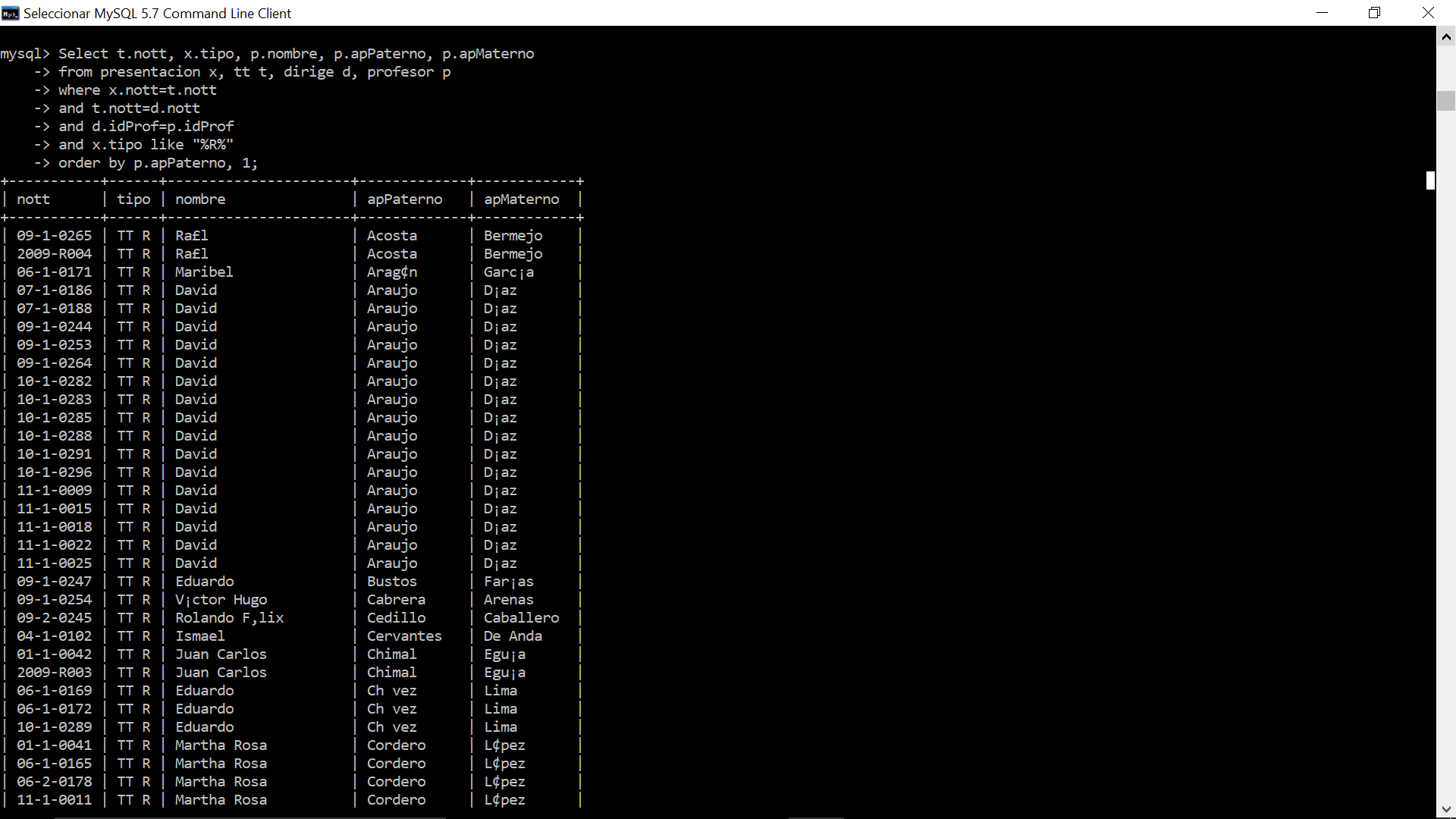
 

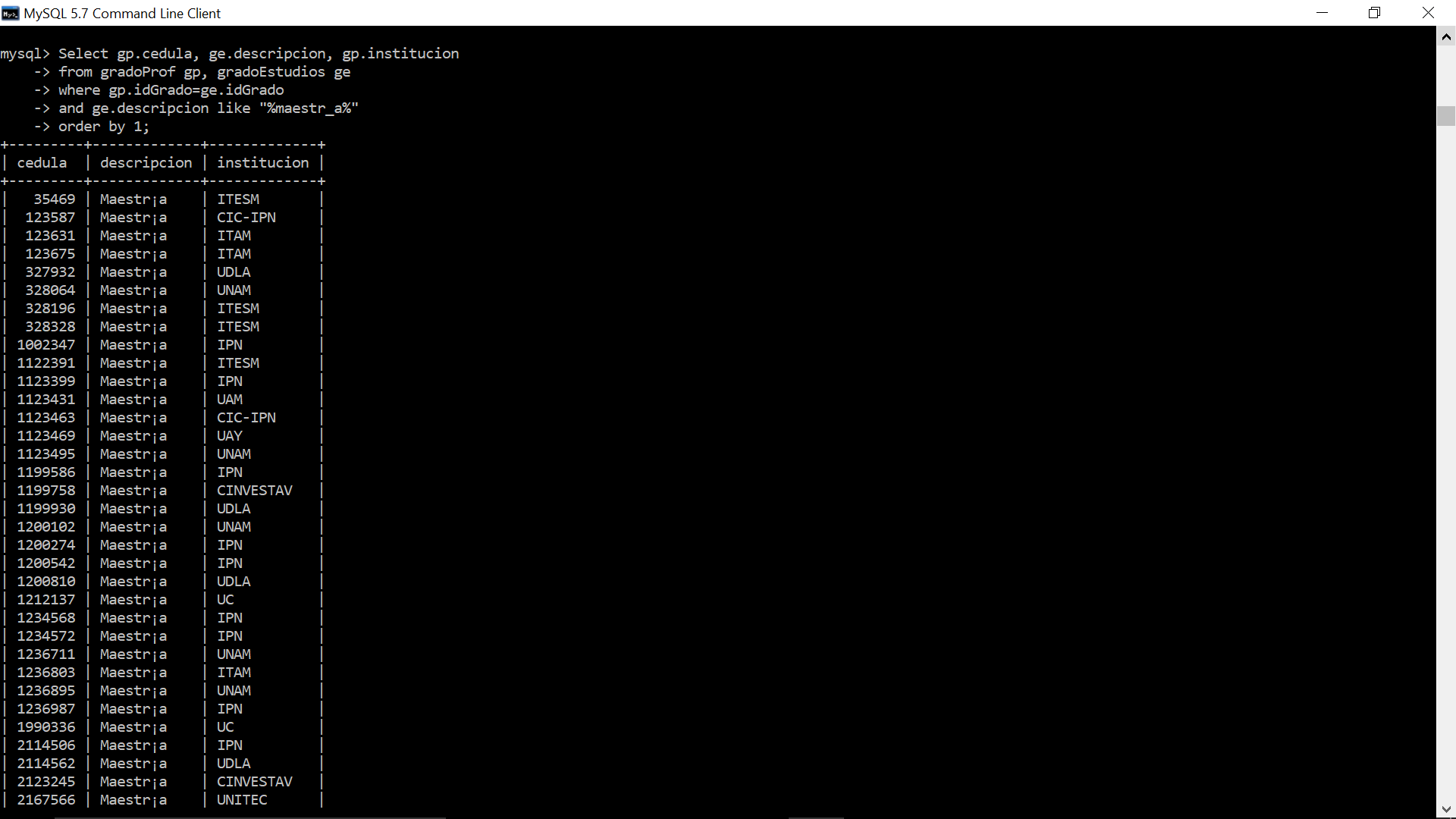
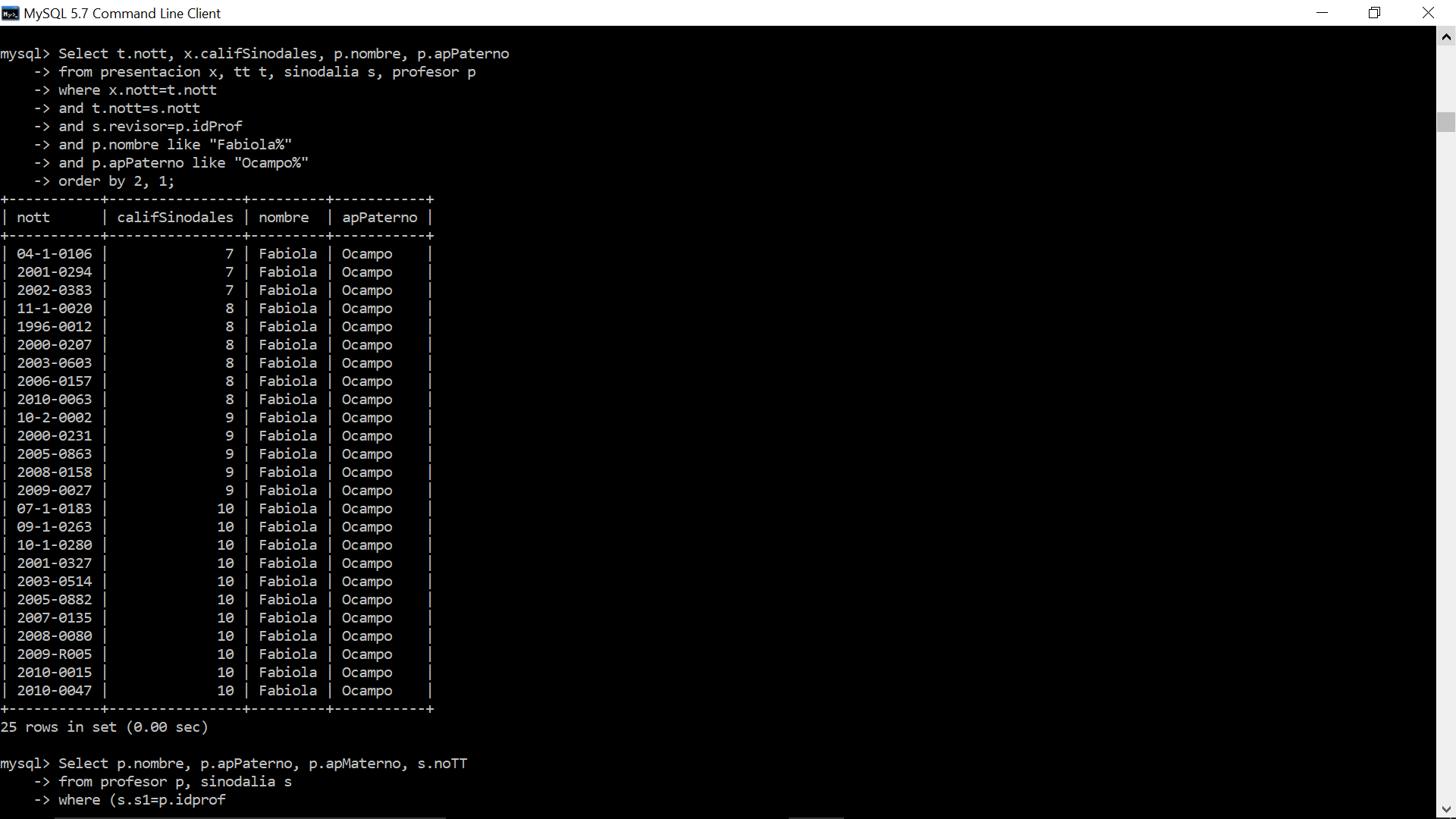


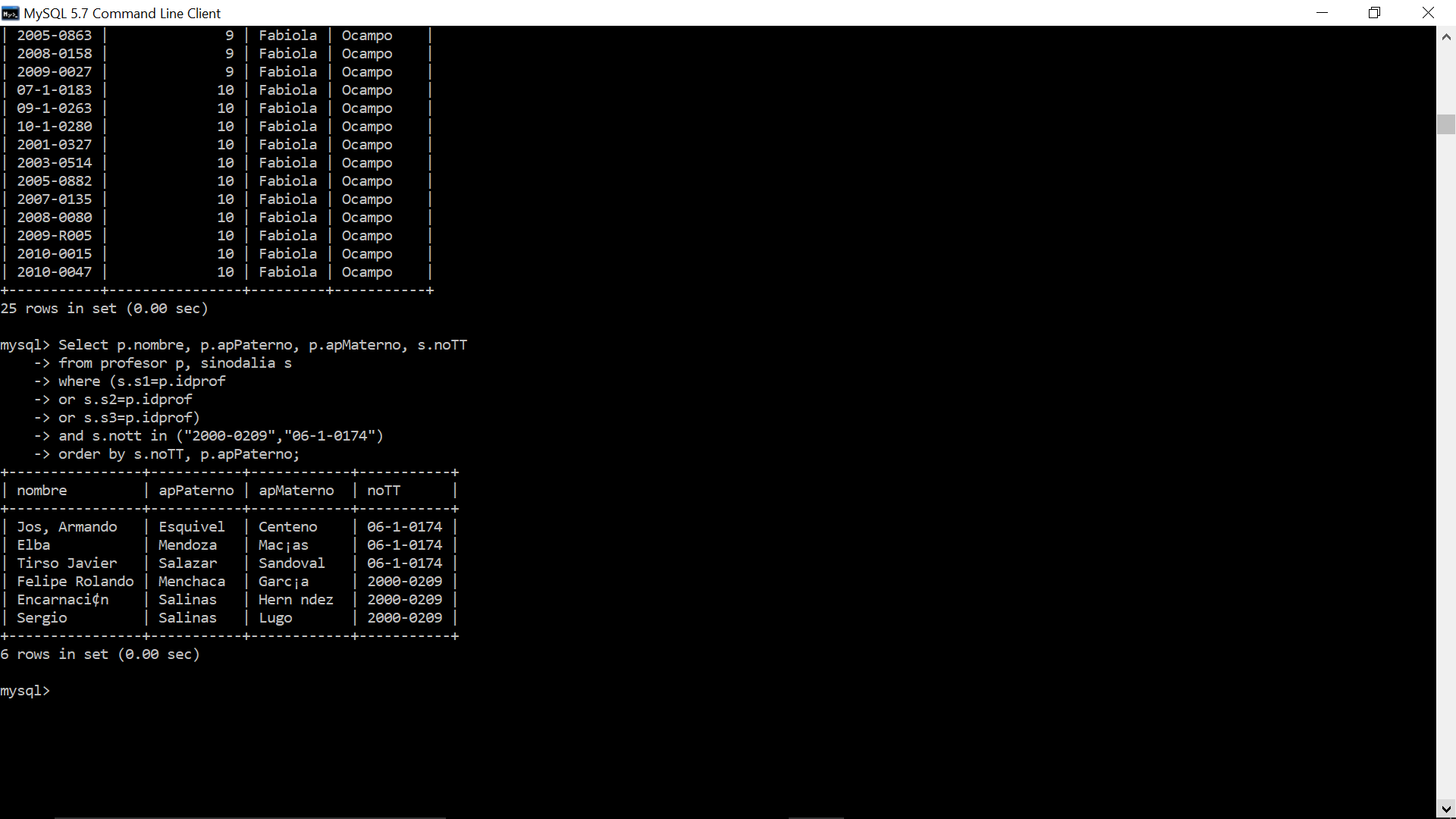


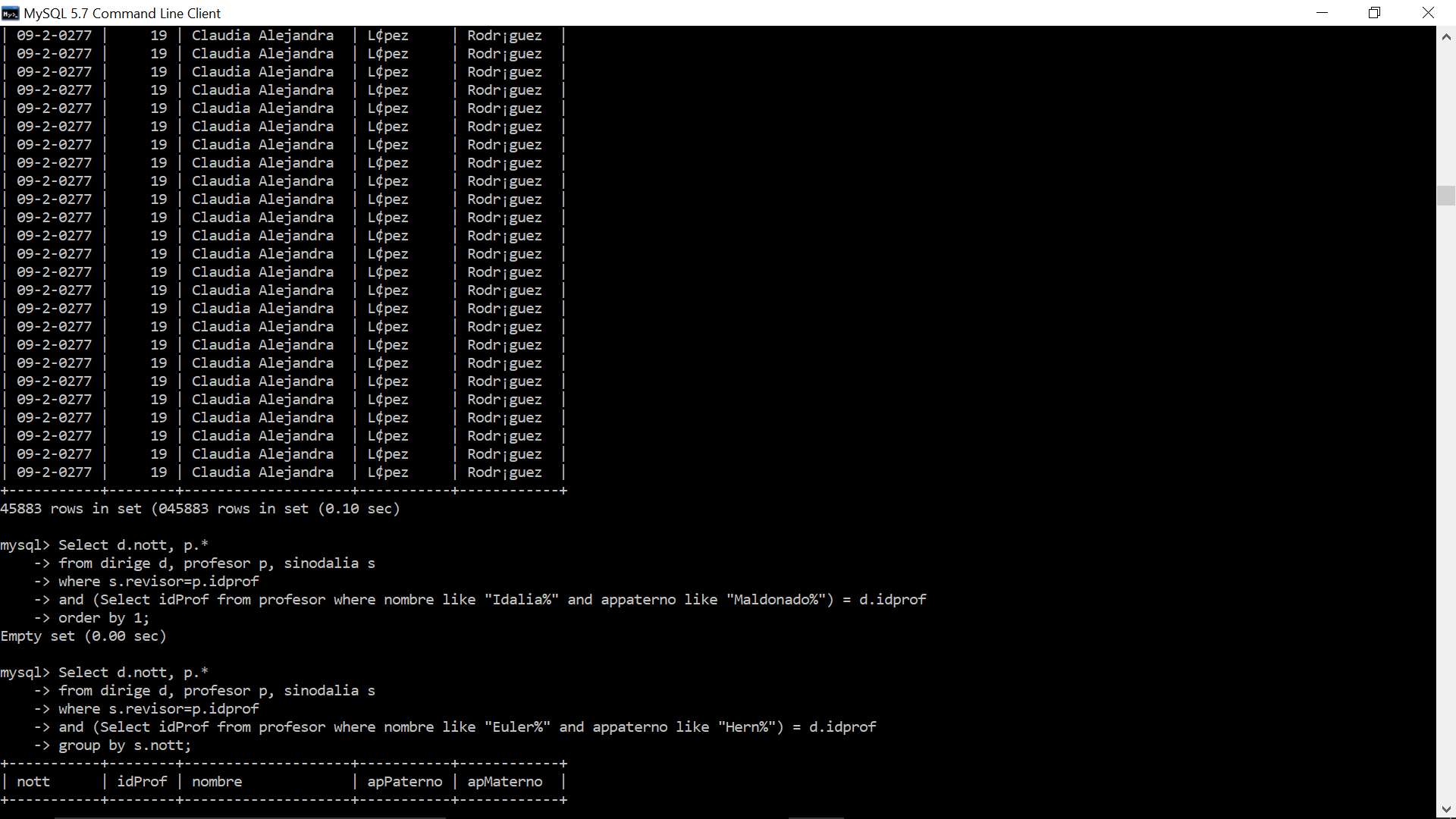


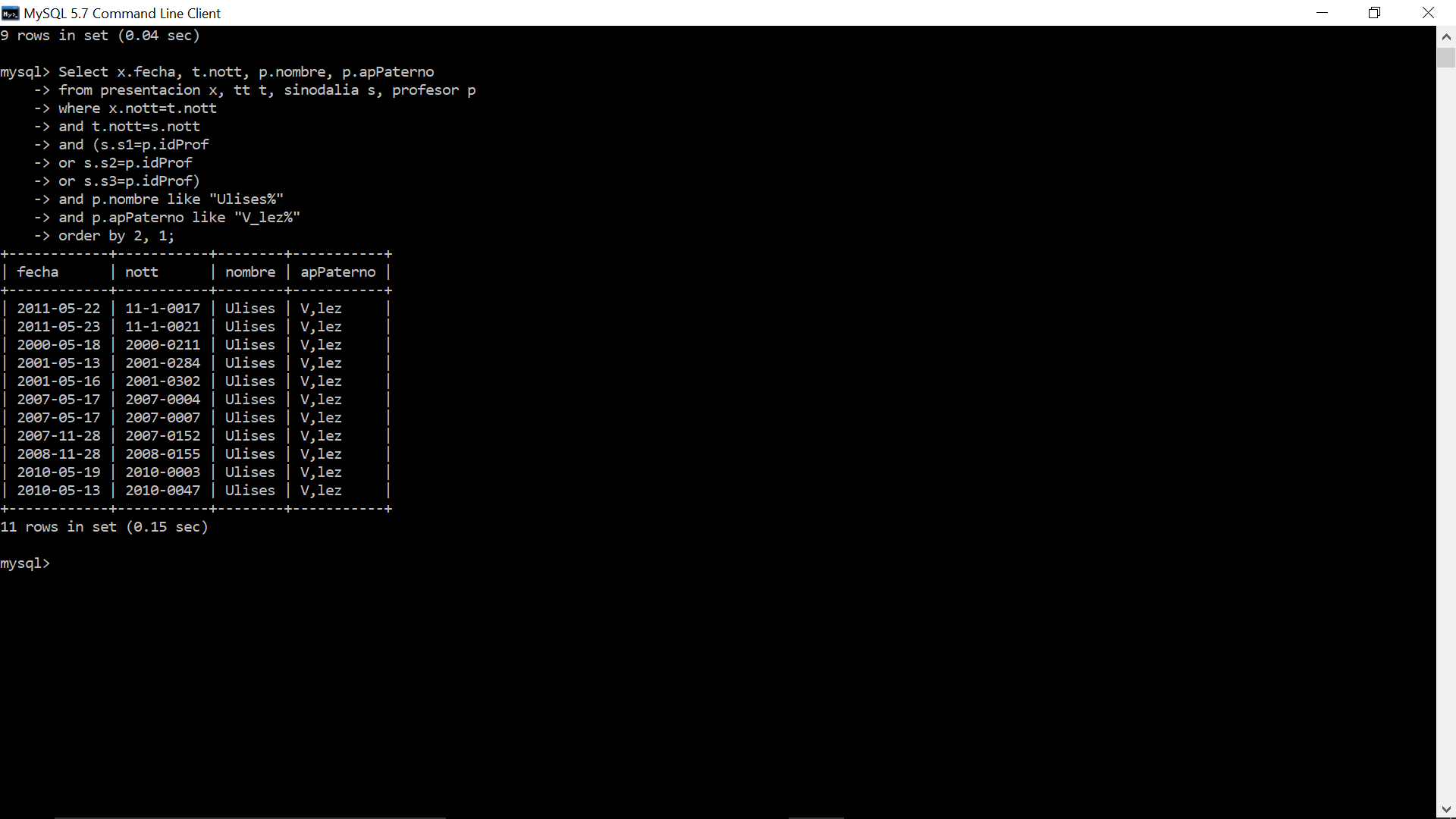










## Conclusiones

Esta práctica fue una continuación de consultas, comenzamos a ampliar la complejidad de las mismas para poder entender mejor cómo es posible relacionar las tablas en una base de datos, y para ello hicimos uso de nuevos operadores como es el JOIN. Éste nos ahorra el tener que concatenar las tablas por medio del ‘ = ’, y sus variaciones nos ayudan a reducir los campos de búsquedas en las respectivas tablas.

Por otra parte, también añadimos nuevos elementos a la hora de implementar las restricciones, ya que el operador IN viene a hacer un trabajo similar al OR, pero de una manera más sencilla, e inclusive más escalable en cierto sentido.

Para concluir, dimos un acercamiento a lo que es la proyección sobre las bases de datos, lo cual se basa en realizar la consulta evitando las repeticiones, o dicho de una mejor forma, la redundancia. Esto se logra por medio del operador GROUP BY, que es el más eficiente en dicha tarea, sin embargo, también mencionamos el operador DISTINCT que viene a realizar el mismo trabajo, pero con un costo mayor para la computadora.

La práctica no tuvo mayores complicaciones prácticas, ya que en general fue bastante sencilla al ser simplemente una ampliación de las pasadas.

## Referencias

Ramez, E., & Navathe, S. (2000). *Sistemas de Bases de Datos: Conceptos Fundamentales* (1st ed.). México: Pearson Educación.

Ricardo, C., Campos Olguín, V., & Enríquez Brito, J. (2010). *Bases de datos*. México: Mcgraw-Hill Interamericana.

Mannino, M. (2007). *Administración de Bases de Datos. Diseño y desarrollo de aplicaciones* (3rd ed.). México: McGraw-Hill Interamericana.