Instituto Politécnico Nacional

Escuela Superior de Cómputo

Bases de Datos

Práctica no. 5:

Vistas

Profesor: Euler Hernández Contreras

Alumno: Calva Hernández José Manuel

Grupo: 2CM12

## Índice

[Índice 2](#_Toc496805731)

[Marco Teórico 3](#_Toc496805732)

[Instrucciones 5](#_Toc496805733)

[Screenshots 9](#_Toc496805734)

[Conclusiones 17](#_Toc496805735)

[Referencias 17](#_Toc496805736)

## Marco Teórico

Una vista en terminología SQL es una tabla que deriva de otras tablas. Esas otras tablas pueden ser tablas base o vistas definidas anteriormente. Una vista no existe necesariamente en formato físico; está considerada como una tabla virtual, en oposición a las tablas base, cuyas tuplas están realmente almacenadas en la base de datos. Esto limita las posibles operaciones de actualización que pueden aplicarse a las vistas, pero no ofrecen limitación alguna al consultar una vista. Una vista es una forma de especificar una tabla a la que nos referimos con frecuencia, aunque no exista físicamente.

Otro importante beneficio de las vistas es la simplificación de tareas. Se pueden formular muchas consultas con mayor facilidad si se usa una vista en lugar de las tablas base. Sin una vista, un enunciado SELECT puede comprender dos, tres o más tablas y requerir de la agrupación en caso de necesitar datos resumidos. Con una vista, el enunciado SELECT sólo puede referirse a una vista sin enlaces ni agrupaciones.

Las vistas ofrecen también un flexible nivel de seguridad. La restricción de acceso con vistas es más flexible que las restricciones para columnas y tablas, porque una vista es cualquier parte derivada de una base de datos. Los datos que no están en la vista permanecen ocultos para el usuario.

La única desventaja de las vistas puede ser el desempeño. El uso de algunas vistas complejas puede comprender una reducción significativa del desempeño, en comparación con el uso directo de las tablas base. La baja en el desempeño puede variar dependiendo del DBMS.

Definir una vista no es más difícil que escribir una consulta. SQL ofrece el enunciado CREATE VIEW, en el que se debe especificar el nombre de la vista, y el enunciado SELECT subyacente.

Definición de una vista con tabla sencilla

CREATE VIEW IS\_View AS

SELECT \* FROM Student

WHERE StdMajor = 'IS'

Definición de una vista con tabla múltiple

CREATE VIEW MS\_View AS

SELECT OfferNo, Offering.CourseNo, CrsUnits, OffTerm,

OffYear, Offering.FacSSN, FacFirstName,

FacLastName, OffTime, OffDays

FROM Faculty, Course, Offering

WHERE FacDept = 'MS'

AND Faculty.FacSSN = Offering.FacSSN

AND Offering.CourseNo = Course.CourseNo

Consulta con el uso de una vista de varias tablas

SELECT OfferNo, CourseNo, FacFirstName, FacLastName,

OffTime, OffDays

FROM MS\_View

WHERE OffTerm = 'SPRING' AND OffYear = 2006

Consulta con el uso de una vista agrupada

SELECT OfferNo, CourseNo, Instructor, NumStudents

FROM Enrollment\_View

WHERE Term = 'SPRING' AND Year = 2006

AND CourseNo LIKE 'IS%'

Una actualización de vista es factible cuando sólo una actualización posible en las relaciones base puede lograr el efecto de actualización deseado en la vista. Siempre que una actualización en la vista pueda mapearse a más de una actualización en las relaciones base subyacentes, deberemos tener un determinado procedimiento para elegir la actualización deseada. Algunos investigadores han desarrollado métodos para elegir la actualización más probable, mientras que otros prefieren dejar que sea el usuario quien elija el mapeado de actualización deseado durante la definición de la vista.

En resumen, podemos hacer estas observaciones:

* Una vista con una sola tabla de definición es actualizable si los atributos de la vista contienen la clave principal de la relación base, así como todos los atributos con la restricción NOT NULL que no tienen valores predeterminados especificados.
* Las vistas definidas sobre varias tablas y que utilizan concatenaciones normalmente no son actualizables.
* Las vistas definidas utilizando funciones de agrupamiento y agregación no son actualizables.

Una vez que se ha definido la vista puede utilizarse en enunciados SELECT. Sólo tiene que usar el nombre de la vista en la cláusula FROM y las columnas en otras partes del enunciado. Es posible agregar otras condiciones y seleccionar un subconjunto de las columnas.

## Instrucciones

1.- Cargar elektra.tt

create database elektra;

use elektra;

source ...

2.- Creación de las siguientes vistas...

a) Nombre de la sucursal y el estado donde se ubica.

create view v1 as

Select nombre as SUCURSAL, estado as ESTADO

from tienda

order by 2;

b) Nombre del producto y su categoría.

create view v2 as

Select p.nombre as PRODUCTO, c.nombre as CATEGORIA

from producto p, subcategoria s, categoria c

where p.idsubcategoria=s.idsubcategoria

and s.idcategoria=c.idcategoria

order by 2, 1;

c) Nombre de la categoría y su subcategoría.

create view v3 as

Select c.nombre as CATEGORIA, s.nombre as SUBCATEGORIA

from categoria c, subcategoria s

where c.idcategoria=s.idcategoria

order by 1, 2;

d) Nombre completo del cliente y la sucursal donde fue dado de alta.

create view v4 as

Select c.nombre as NOMBRE, c.apPaterno, c.apMaterno, t.nombre as SUCURSAL

from cliente c, pago p, tienda t

where c.idcliente=p.idcliente

and p.idtienda=t.idtienda

order by 4, 2;

e) Nombre del producto y su subcategoría.

create view v5 as

Select p.nombre as PRODUCTO, s.nombre as SUBCATEGORIA

from producto p, subcategoria s

where p.idsubcategoria=s.idsubcategoria

order by 2, 1;

f) Nombre del cliente y su monto de crédito.

create view v6 as

Select c.nombre as NOMBRE, c.appaterno, c.apmaterno, cr.monto as CREDITO

from cliente c, pago p, credito cr

where c.idcliente=p.idcliente

and p.idcredito=cr.idcredito

order by 2, 3;

g) Nombre del cliente y fechas de pago

create view v7 as

Select c.nombre as NOMBRE, c.appaterno, c.apmaterno, p.fechapago as FECHA

from cliente c, pago p

where c.idcliente=p.idcliente

order by 2, 3;

h) Nombre del producto y precio unitario.

create view v8 as

Select nombre as PRODUCTO, precioUnitario as PRECIO

from producto

order by 1;

i) Nombre del producto y su marca.

create view v9 as

Select nombre as PRODUCTO, marca as MARCA

from producto

order by 1;

j) Nombre del cliente, email y su género.

create view v10 as

Select nombre, appaterno, apmaterno, email, sexo

from cliente

order by 2, 3;

k) Nombre del cliente y su salario.

create view v11 as

Select nombre, appaterno, apmaterno, salario

from cliente

order by 2, 3;

3.- Resolver las siguientes consultas empleando las vistas generadas anteriormente

a) Clientes existentes en las sucursales en el estado de Colima.

Select v4.\*, v1.estado

from v4, v1

where v4.sucursal=v1.sucursal

and v1.estado like "Colima%"

order by v1.sucursal;

b) Sucursales donde hay mujeres.

Select v4.\*, v10.sexo

from v10, v4

where v4.nombre=v10.nombre

and v4.appaterno=v10.appaterno

and v4.apmaterno=v10.apmaterno

and v10.sexo like "%F%"

order by v4.appaterno, v4.apmaterno;

c) Clientes que ganan entre $6,000 a $6,500, incluir sus sucursales.

Select v11.\*, v4.sucursal

from v11, v4

where v11.nombre=v4.nombre

and v11.appaterno=v4.appaterno

and v11.apmaterno=v4.apmaterno

and v11.salario between "6000" and "6500"

order by v11.salario, v11.appaterno;

d) Productos pertenecientes a la categoría DEPORTE.

Select \*

from v2

where categoria like "%deporte%"

order by producto;

e) Clientes que realizaron pagos el día 12 marzo 2010, incluir sucursal.

Select v7.\*, v4.sucursal

from v7, v4

where v7.nombre=v4.nombre

and v7.appaterno=v4.appaterno

and v7.apmaterno=v4.apmaterno

and v7.fecha="2010-03-12"

order by v7.appaterno, v7.apmaterno;

-----------------------------------------

drop view \_\_\_\_\_\_\_\_\_\_; // nombre vista

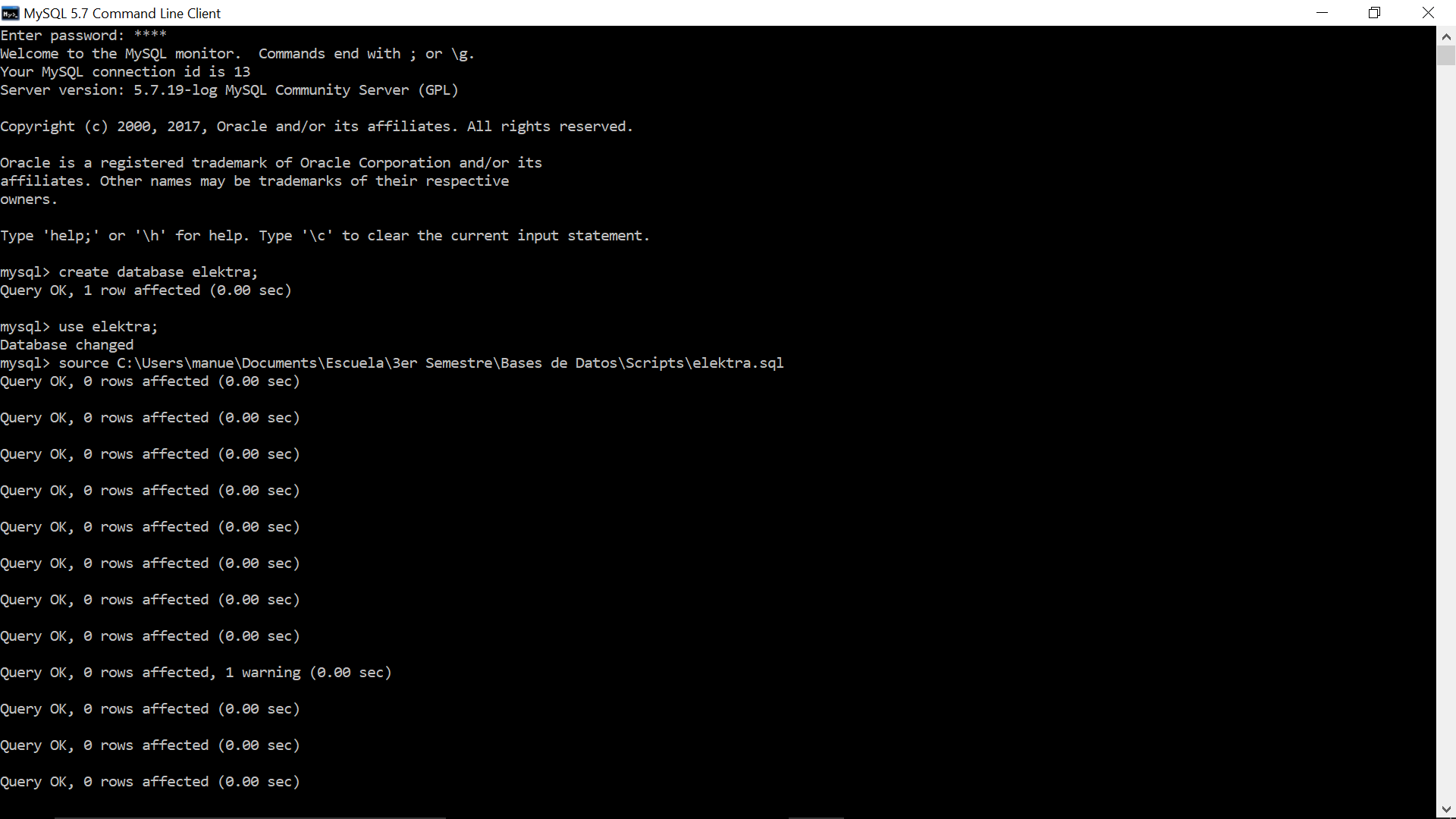
show tables; // nombre vista

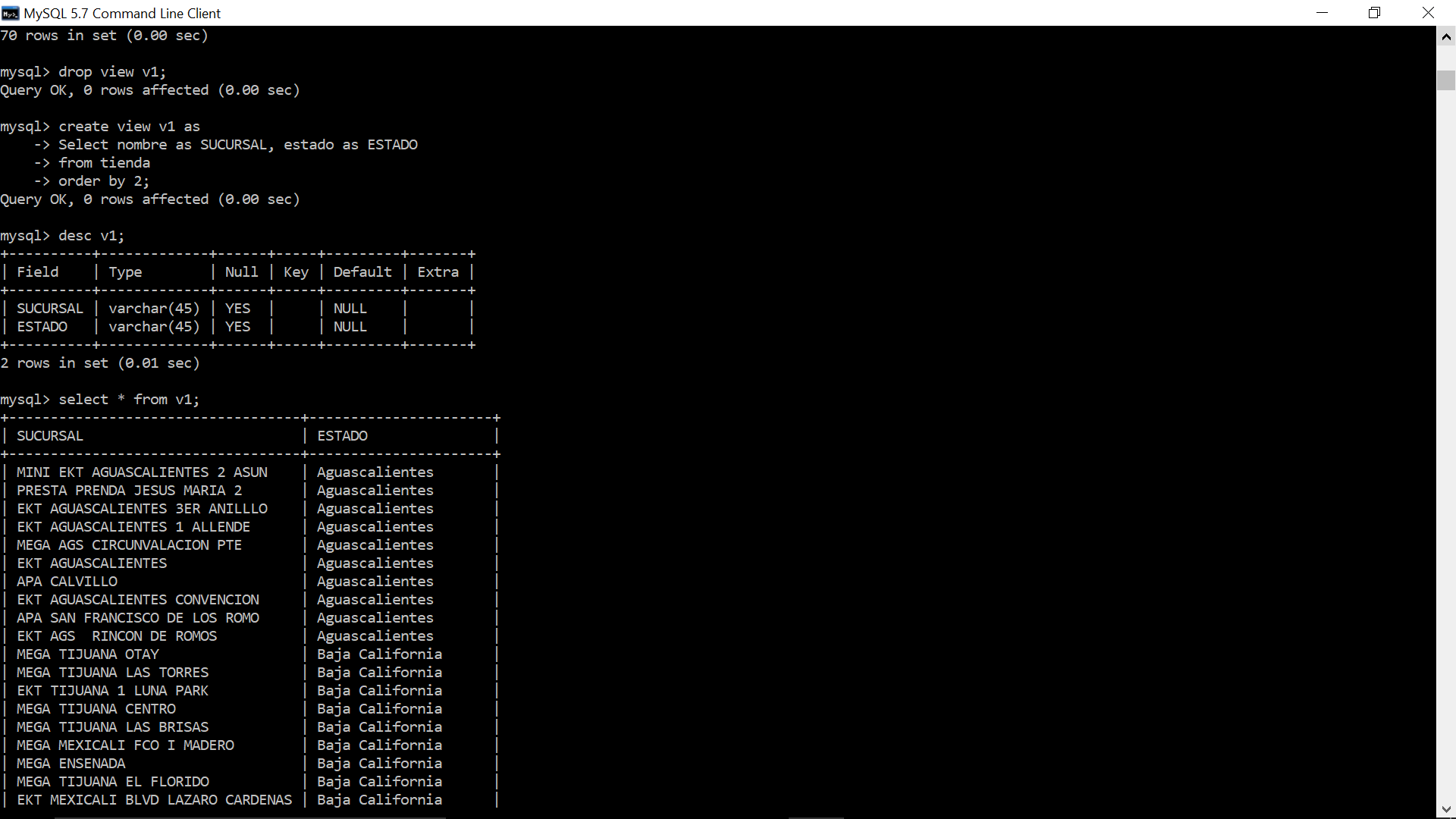
desc \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_; // metadato vista

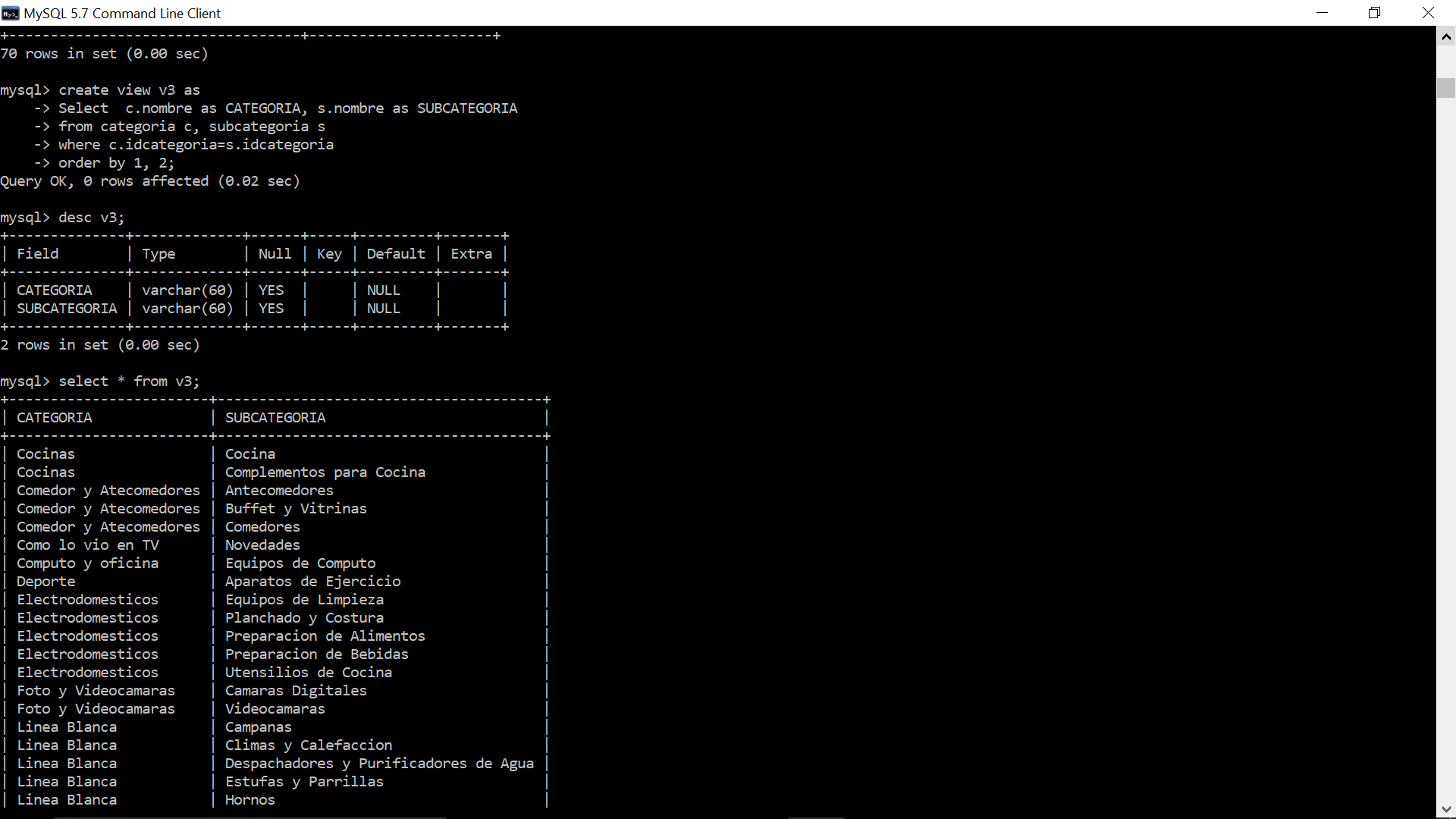
create view \_\_\_\_\_\_\_\_\_\_\_\_\_ as … sentencia SQL … ;

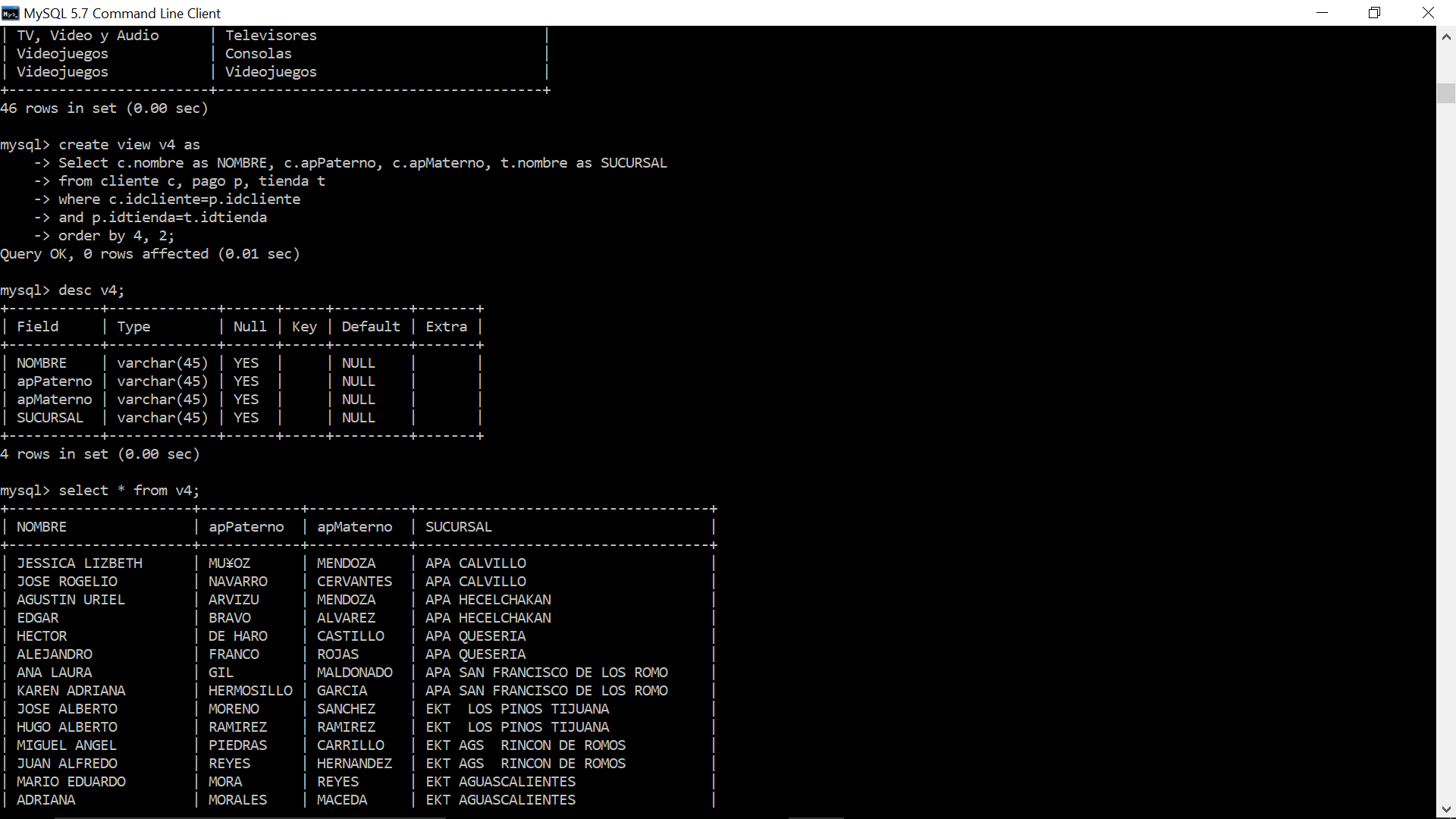
Select \* from \_\_\_\_\_\_\_\_\_\_\_\_\_;

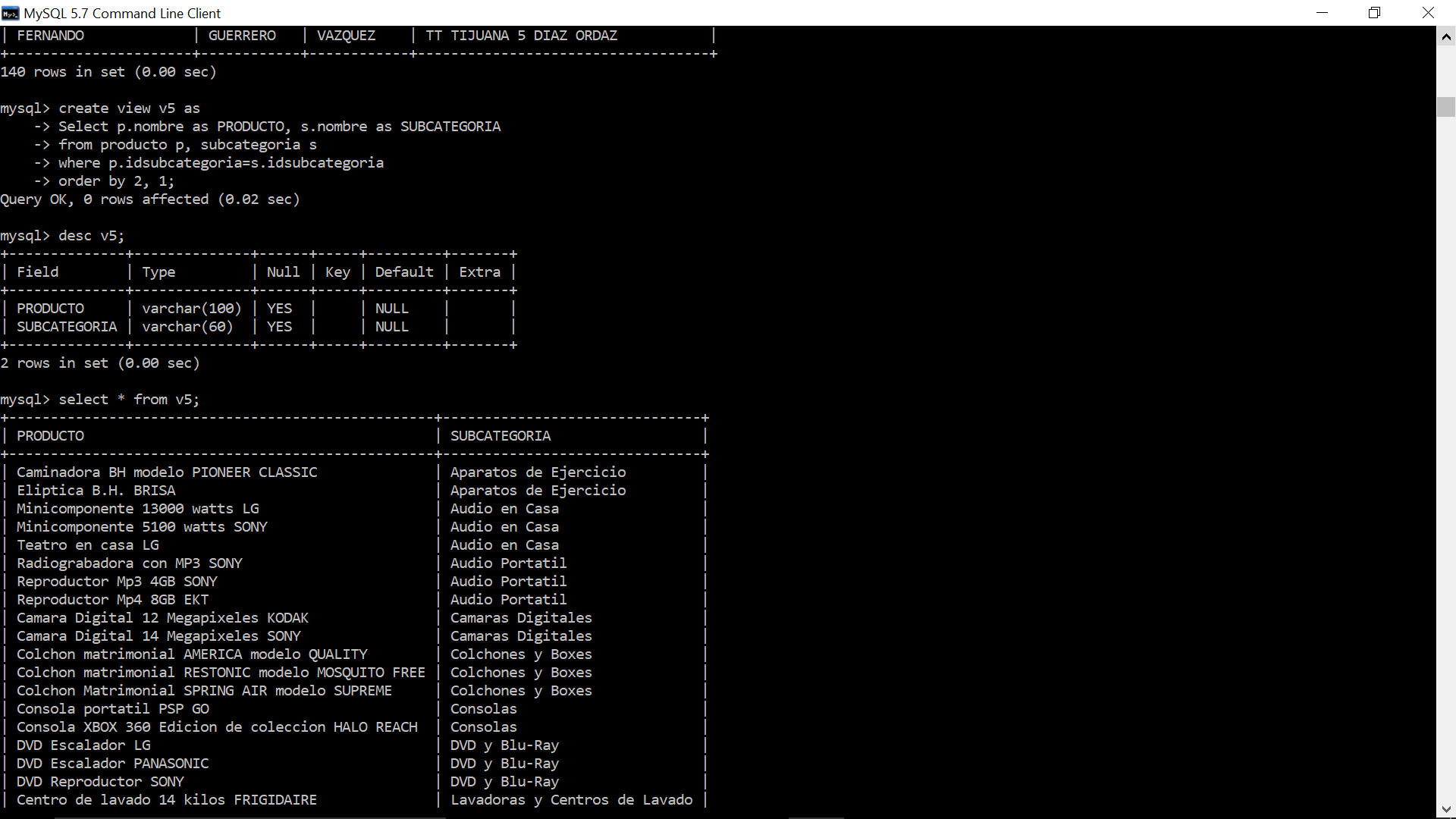
## Screenshots

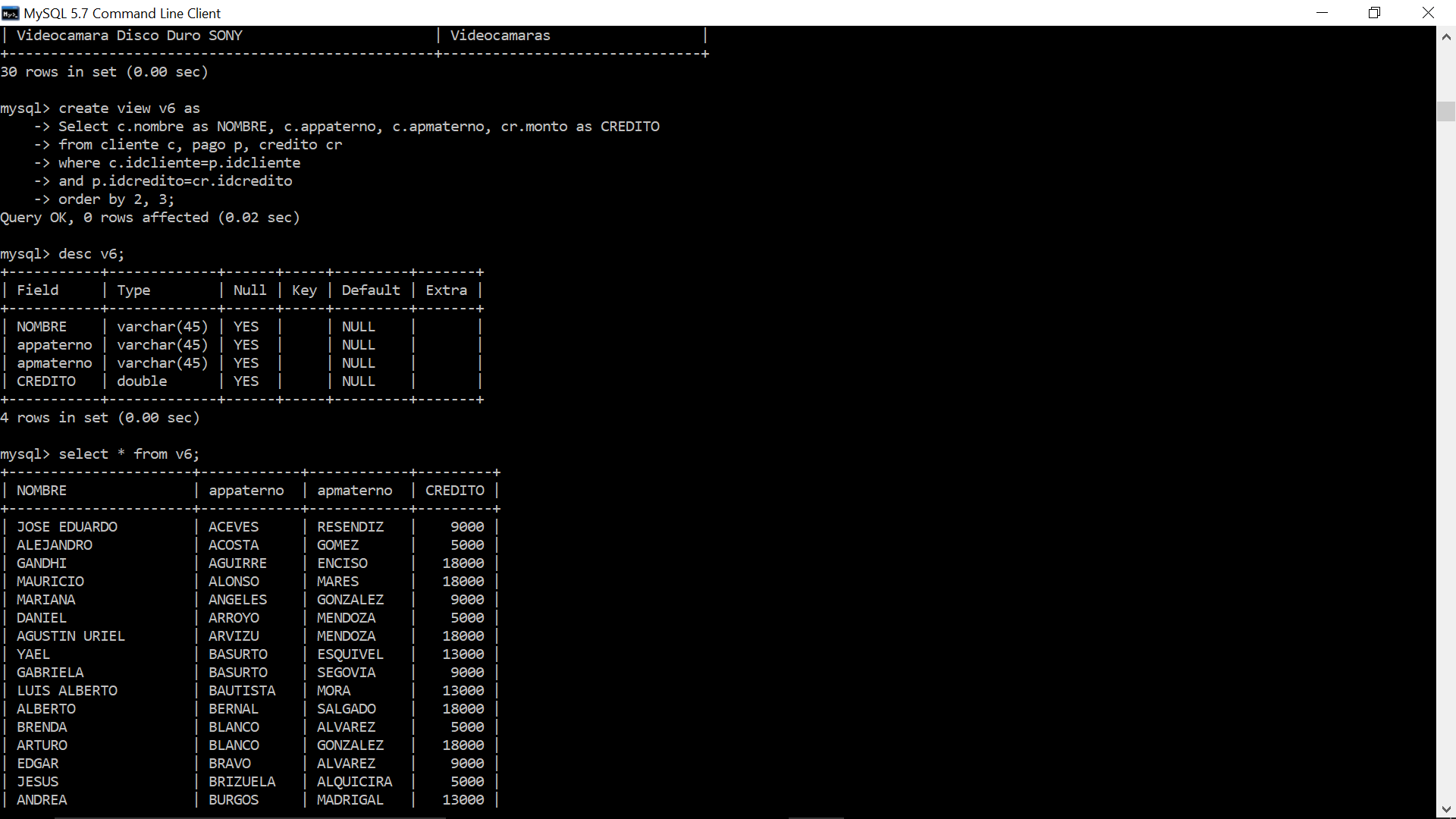


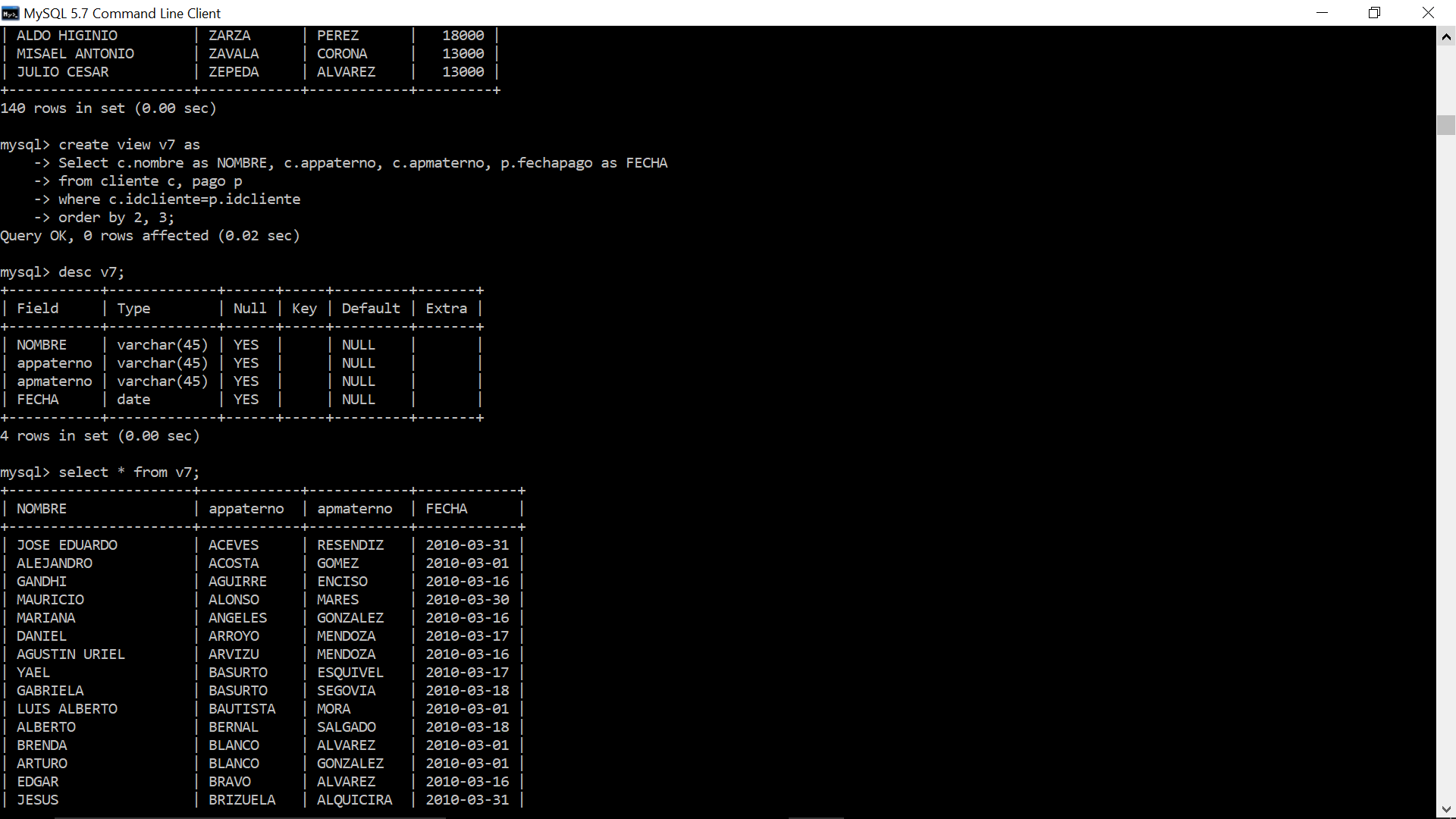


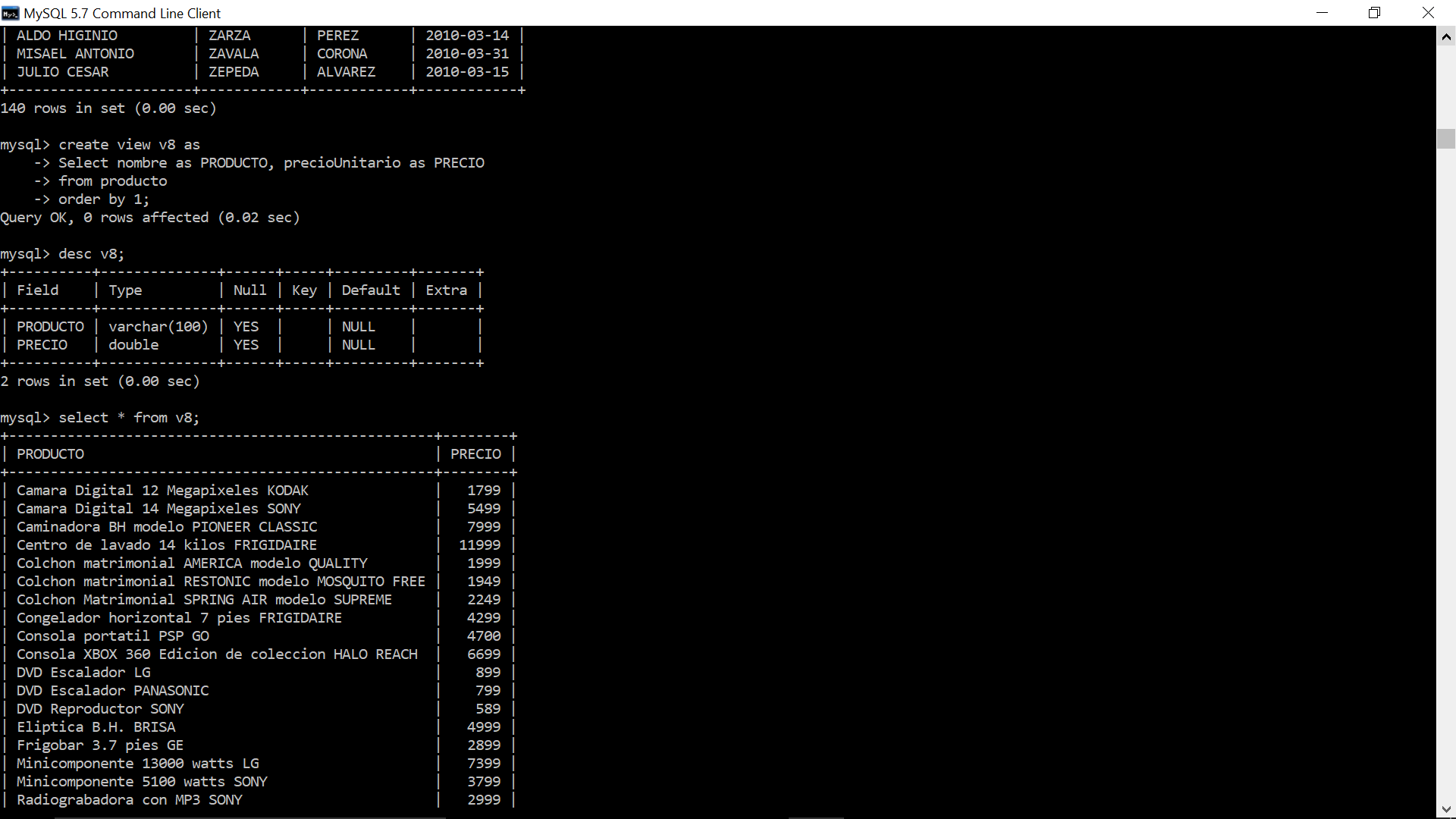


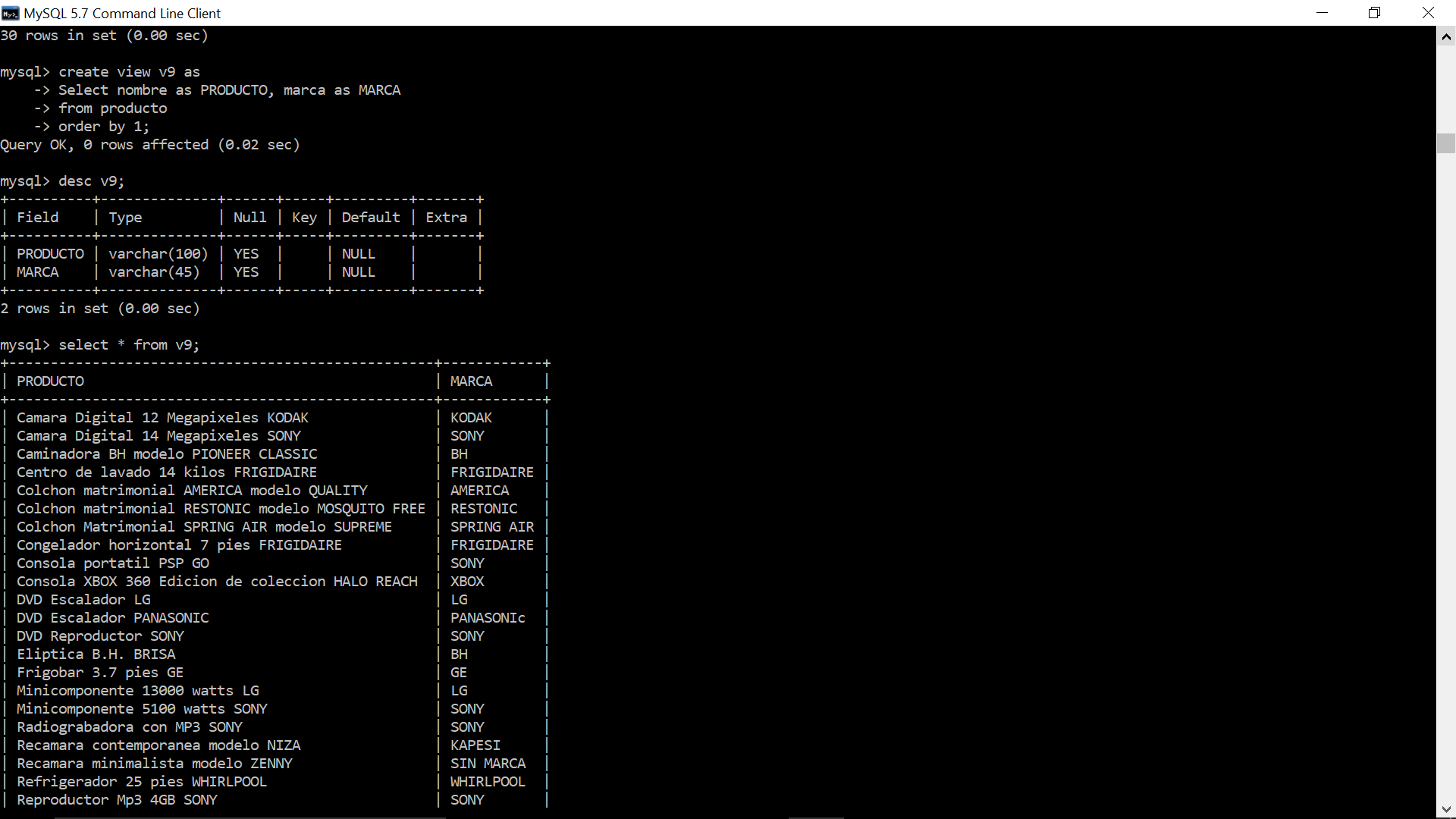


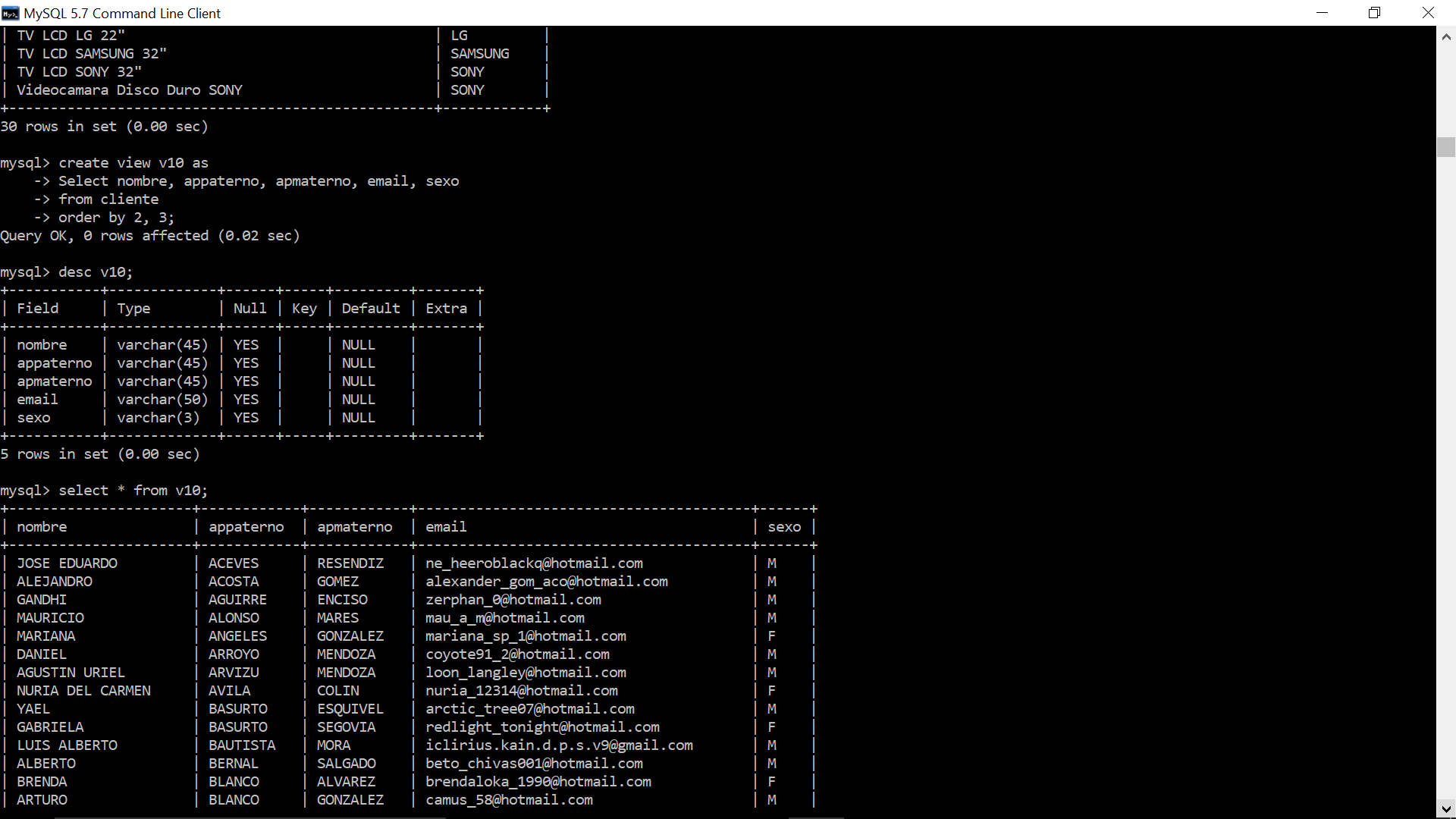


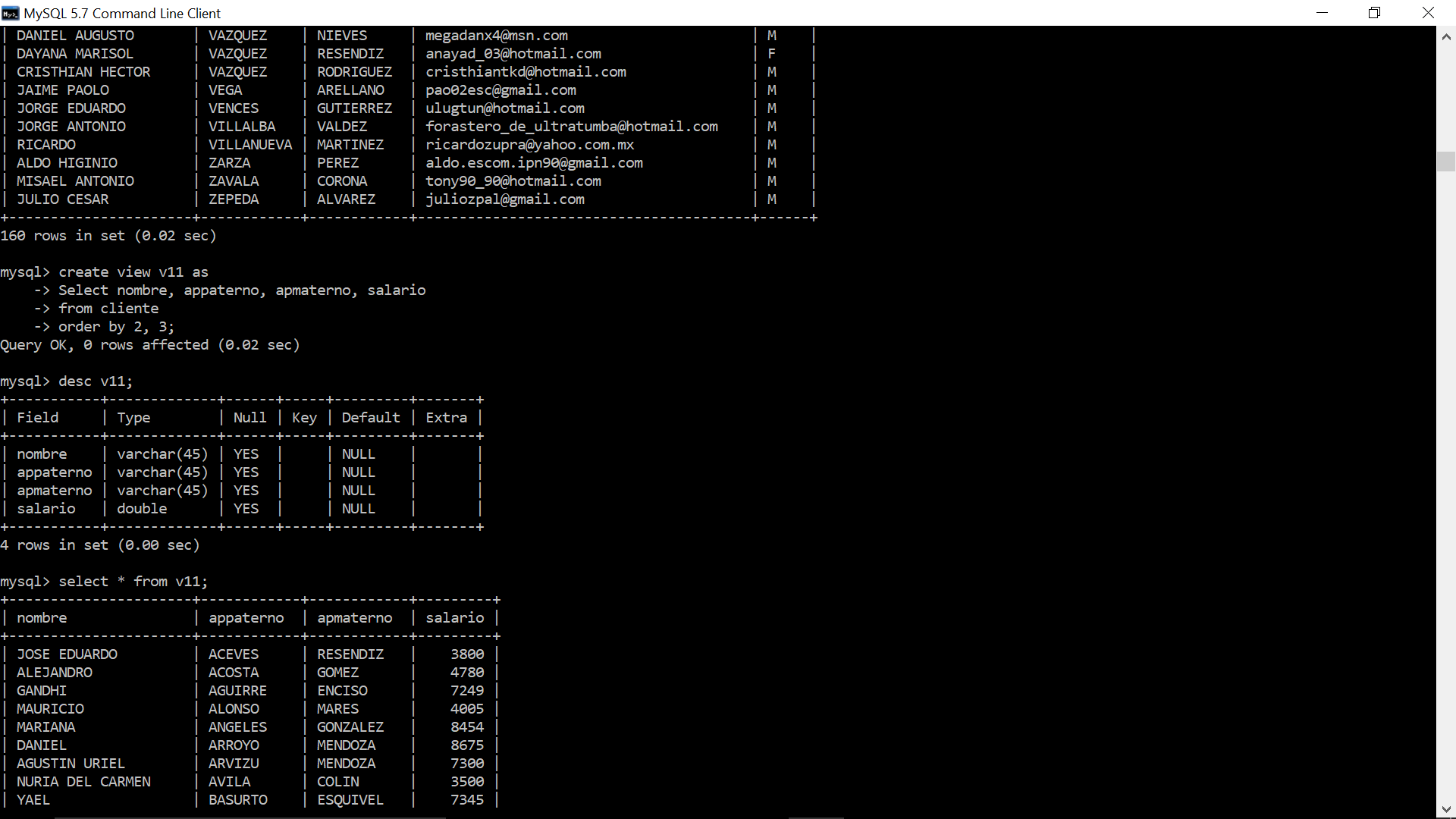


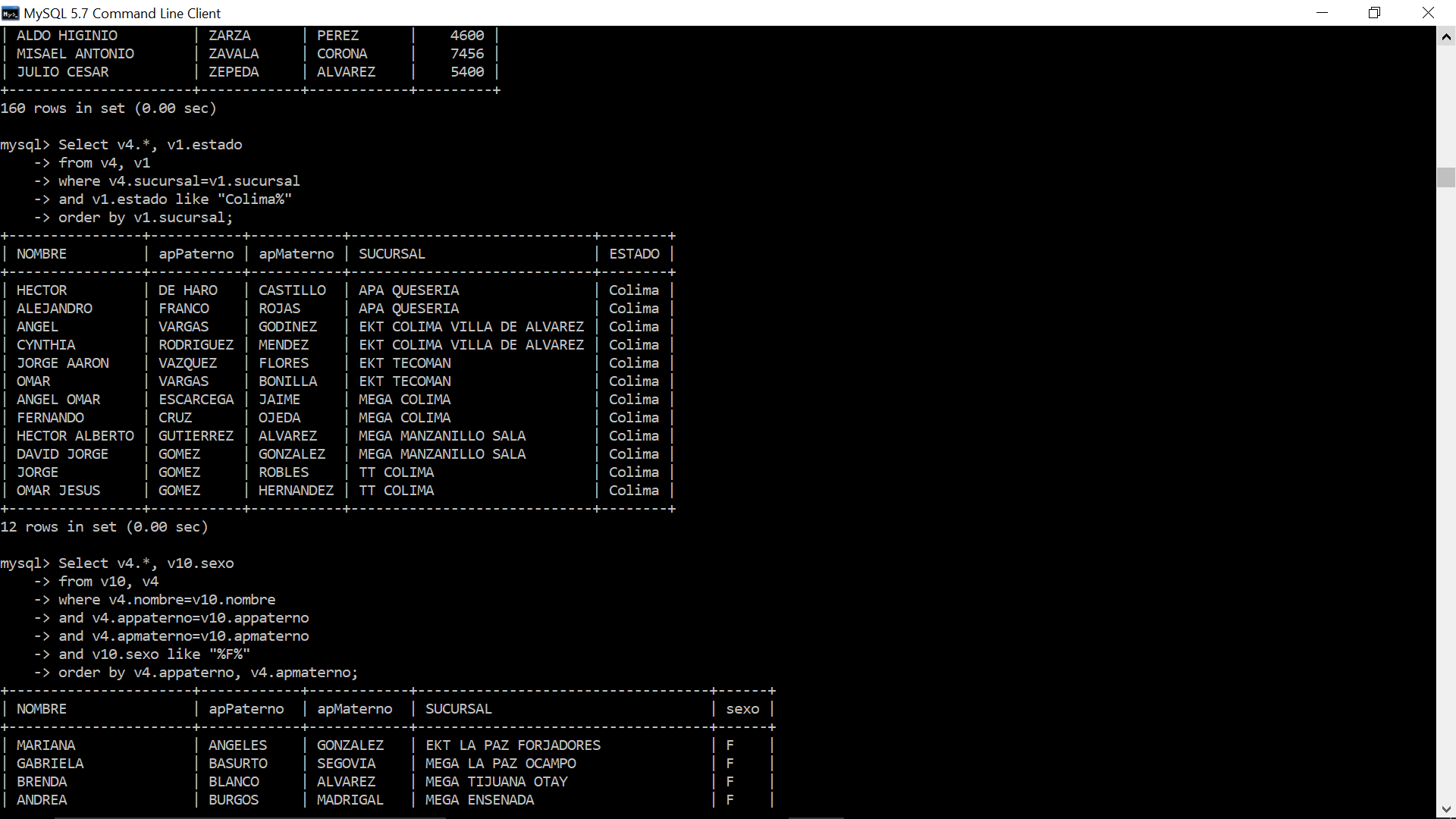


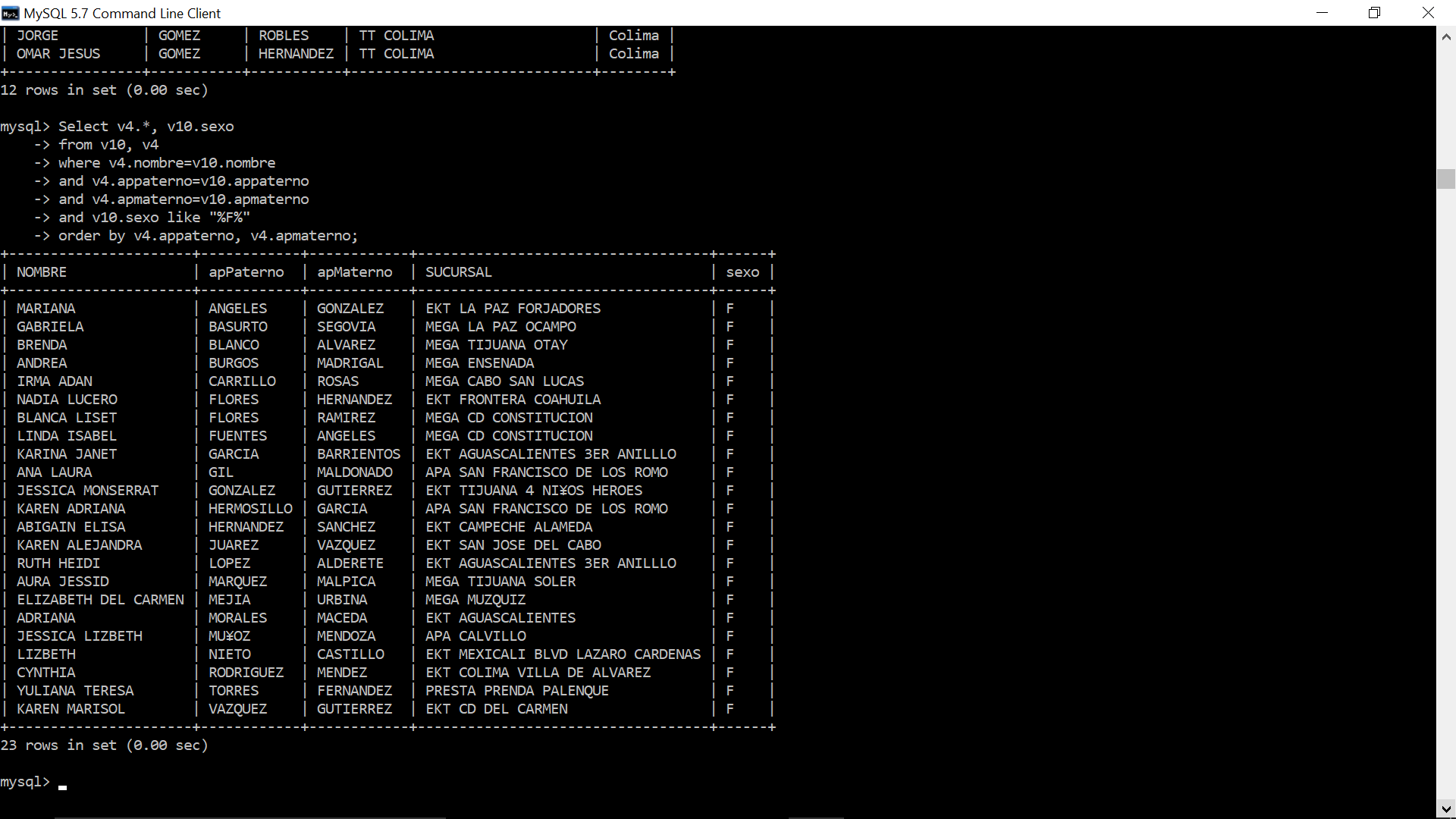


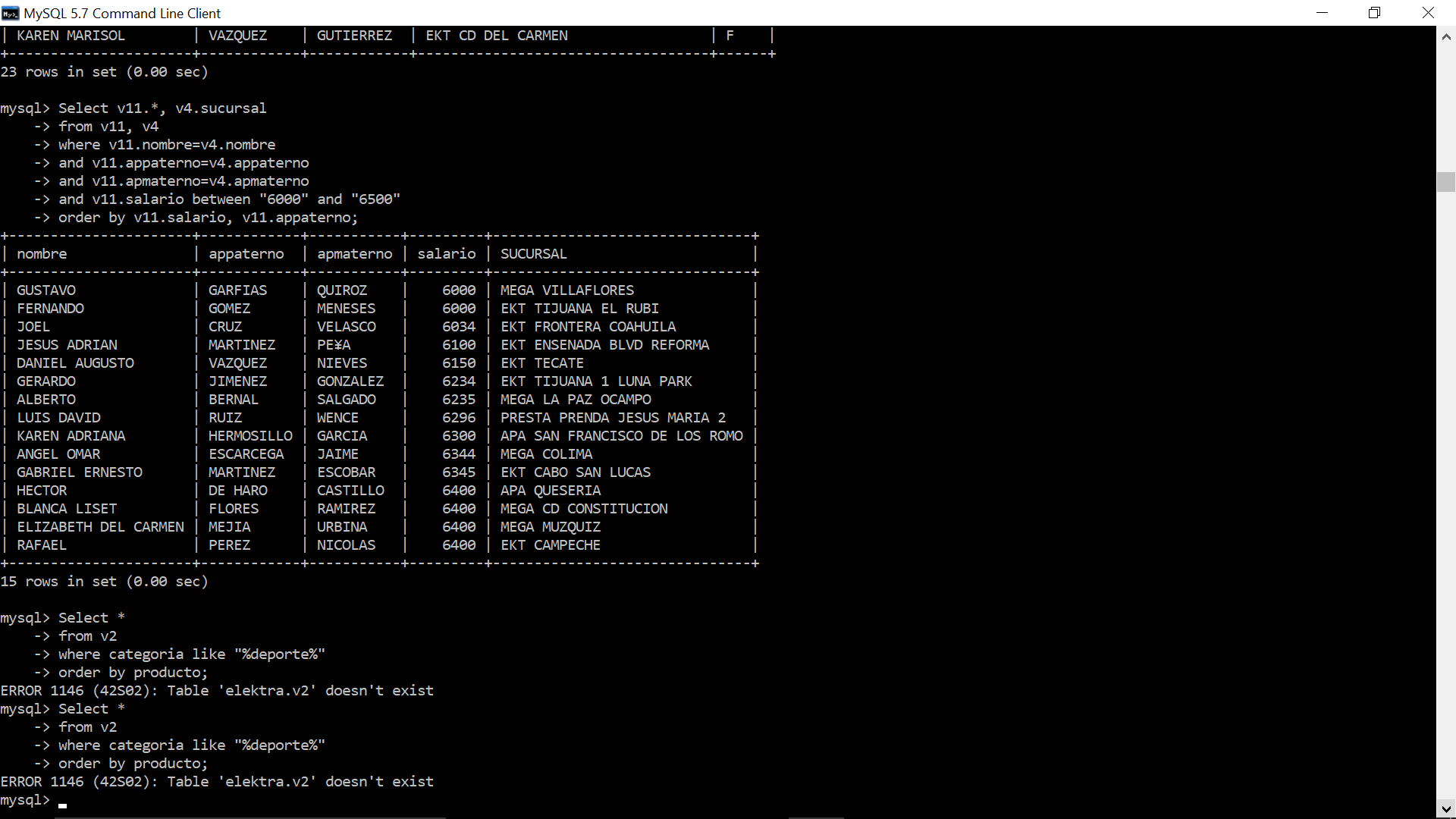


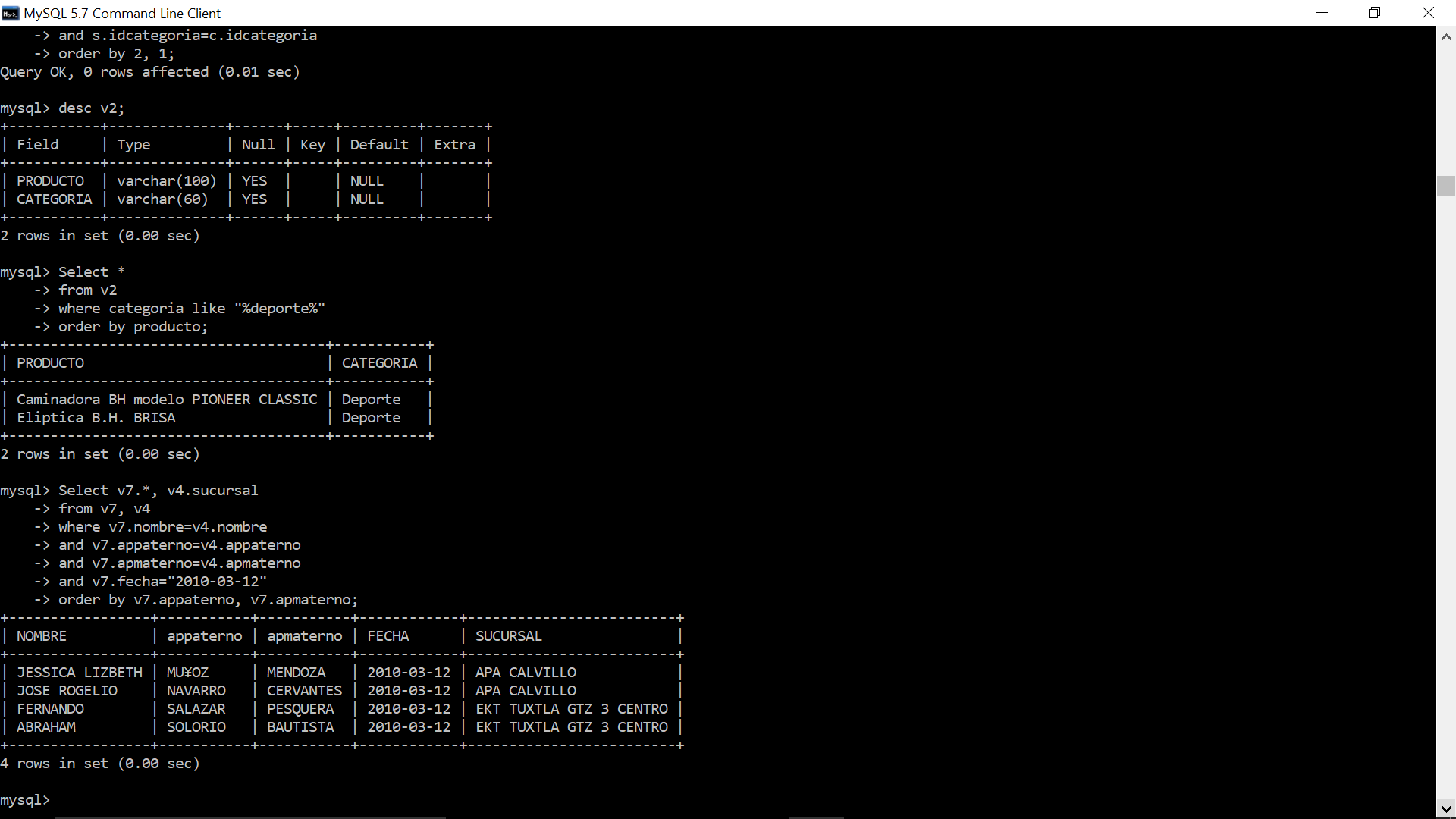


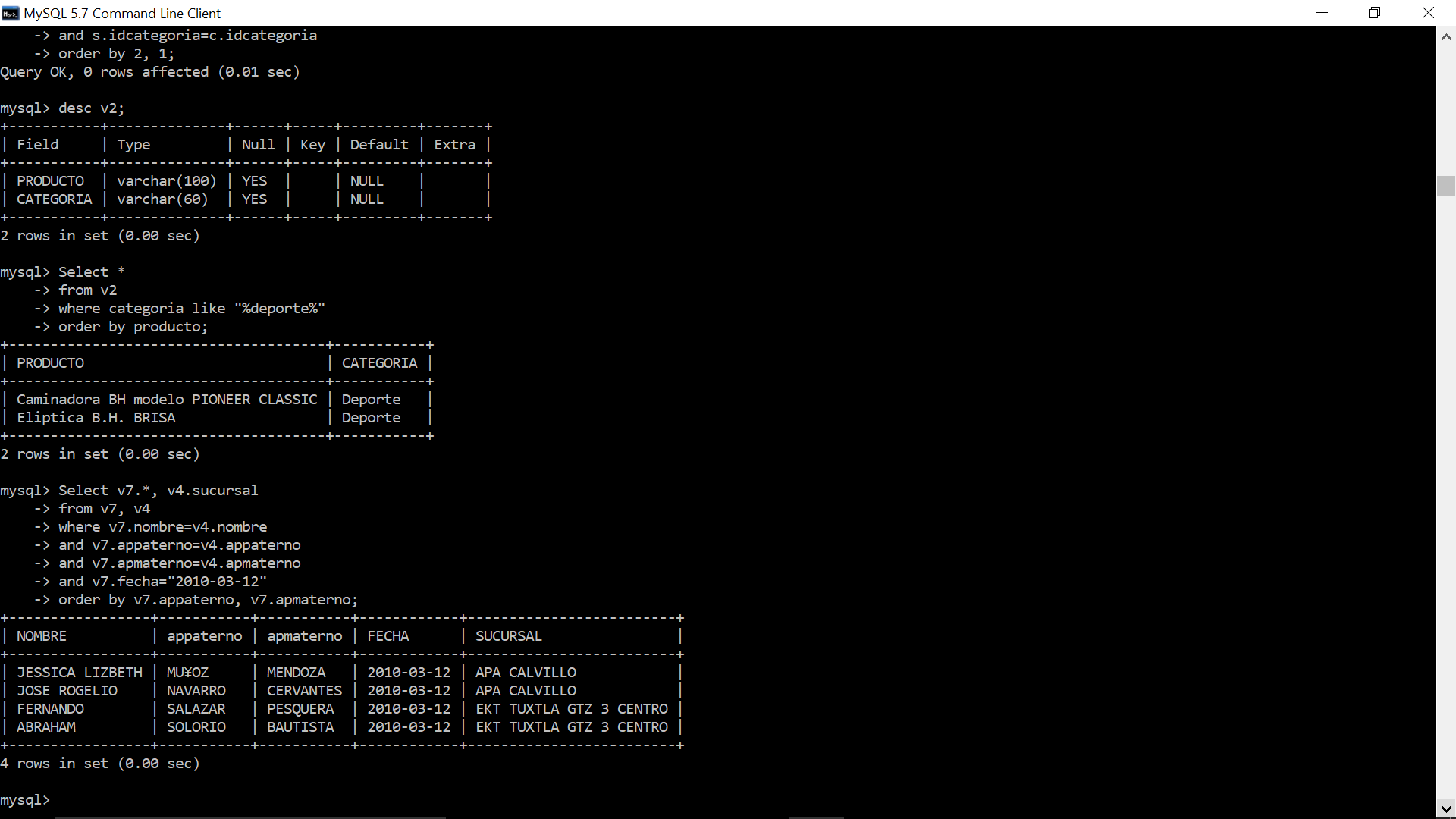












## Conclusiones

La práctica concluyó sin mayor dificultad, fue bastante sencilla ya que aprendimos que las vistas es una forma de interrelacionar tablas sin riesgo de modificar la estructura interna de la base de datos, a su vez, podemos usarla como una tabla para hacer consultas. Sin embargo, es importante tener claro como definimos cada vista, ya que el no saber el nombre de una, o los campos internos si los renombramos, o bien, si desconocemos que colocamos en cada una; puede resultar más problemática dado que las consultas sobre éstas se complican.

Las vistas además nos permiten un rápido acceso a datos concretos de interés, o bien, para hacer más accesibles estos a los usuarios, sin embargo, también puede consumir un mayor rendimiento de la máquina dado que tienen un tiempo base de lo que sería la consulta normal, más un tiempo en el que tardan en presentarla en pantalla, lo que se debe de tener en cuenta a la hora de decidir usar vistas.

## 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.