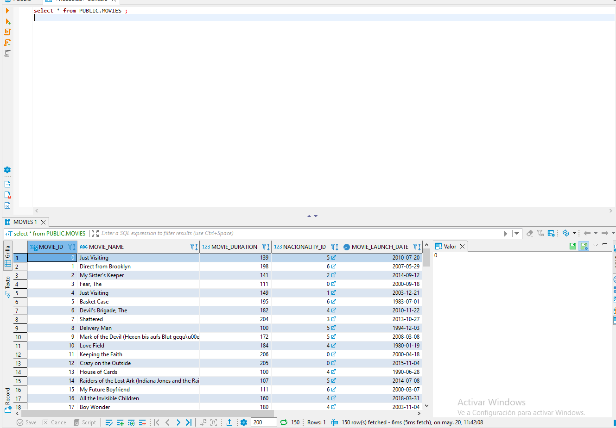
## Devuelve todas las películas​​​​​​​.Selección sin restricciones

La sentencia **SELECT** es la única que permite consultar información de forma que pude devolver un registro completo de datos o un solo valor .Se debe considerar que es la única función que permite obtener cualquier tipo de información desde las tablas de una base de datos.

El símbolo **\*** representa todos los registros (filas) así como todas las columnas de una determina tabla.

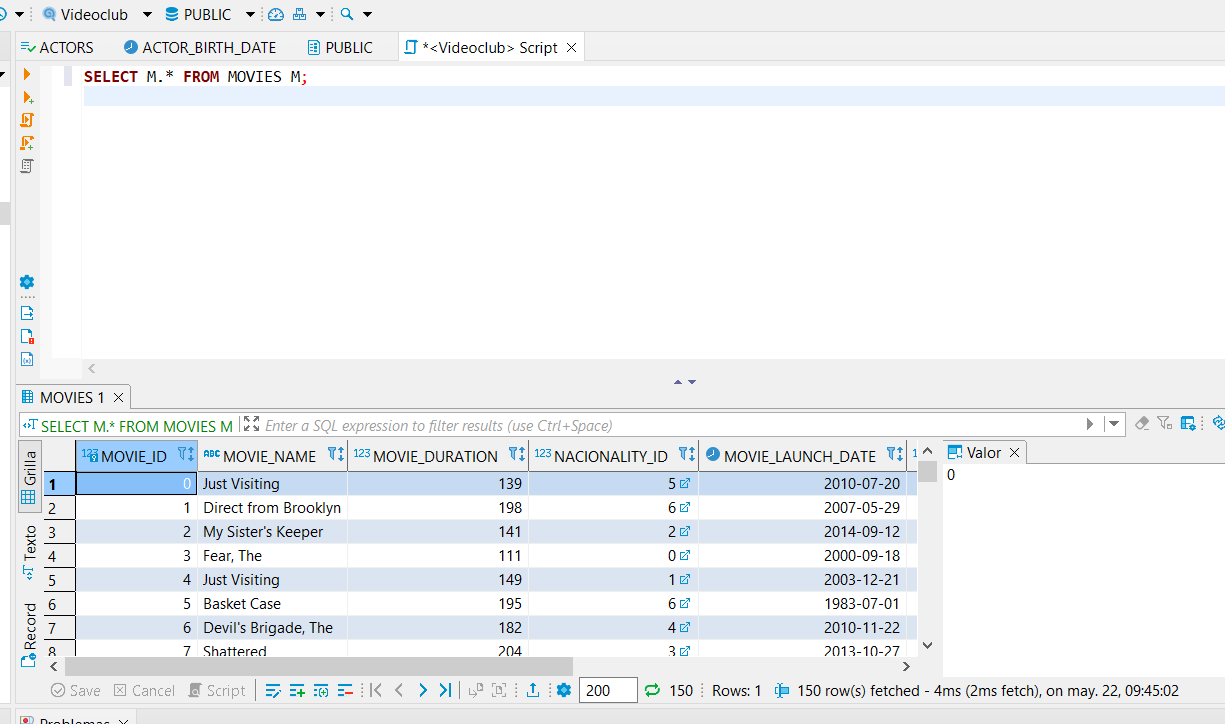
### Caso1 :Listar todos los registros de una tabla

**select** \* **from** PUBLIC.MOVIES ;



### Caso 2: USO DE ALIAS DE TABLA.

**SELECT** M.\* **FROM** MOVIES M;



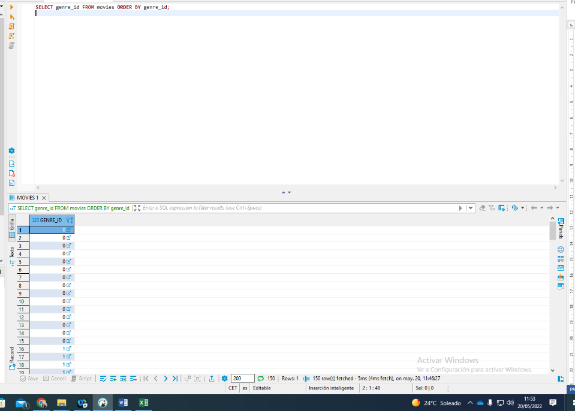
## Devuelve todos los géneros de las películas, selección de datos limitando las columnas de una tabla:

Este tipo de consulta especifica las columnas que se deben mostrar del total de las presentes en una tabla mediante el nombre de la columna de la tabla antes del **FROM**.

### Géneros ordenados.

**ORDER BY** , este tipo de consulta permite ordenar de forma ascendente (defecto) o descendentes los valores de una determinada columna o columnas de una tabla. En nuestro caso ordenamos por genero .

**SELECT** genre\_id **FROM** movies **ORDER** **BY** genre\_id;

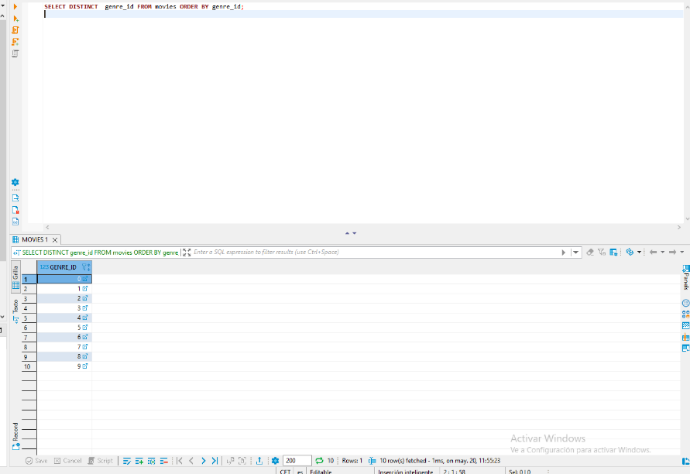
****

### Géneros ordenados sin repetición. Restringir los datos repetidos.

La sintaxis SELECT DISTINCT nombre\_columna FROM nombre\_tabla;

#### Solo los códigos de género.

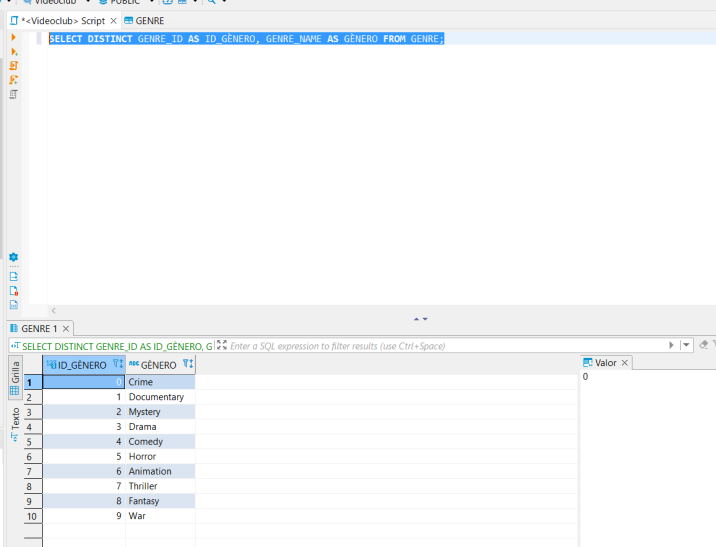
**SELECT** **DISTINCT** genre\_id **FROM** movies **ORDER** **BY** genre\_id;

****

### Códigos de género con su nombre y editando los encabezados.

Mediante le uso de as lo logramos:

**SELECT** **DISTINCT** GENRE\_ID **AS** ID\_GÈNERO, GENRE\_NAME **AS** GÈNERO **FROM** GENRE;

****

### Géneros ordenados por id y género, nombre de película.

**ORDER BY** , este tipo de consulta permite ordenar de forma asdecende (defecto) o descendentes los valores de una deerminada columna o columnas de una tabla.En nustro caso ordenamos primero por genero y luego por película (parece lo más lógico dentro de un videoclub).

**INNER JOIN**, permite mostrar la información cruzada entre dos o más tablas para diferentes procesos. Todo proviene de llave foránea de la tabla, ya que es la que permite acceder a l otra tabla y así combinarla.

**SELECT** MOVIES.MOVIE\_ID **AS** ID\_Película,

MOVIES.MOVIE\_NAME **as** Nombre\_Películas ,

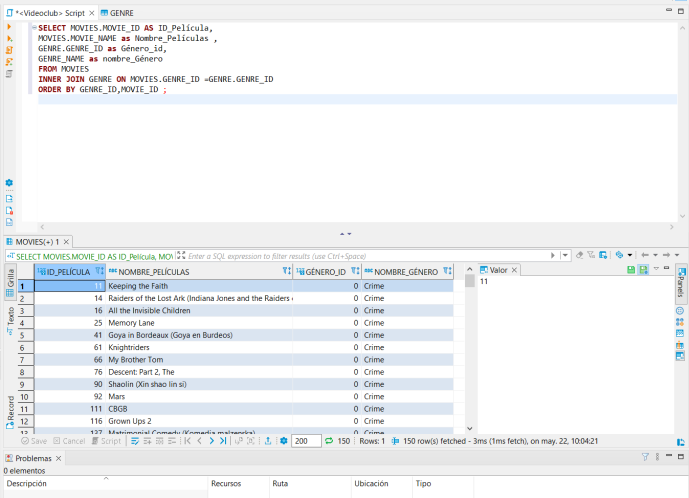
GENRE.GENRE\_ID **as** Género\_id,

GENRE\_NAME **as** nombre\_Género

**FROM** MOVIES

**INNER** **JOIN** GENRE **ON** MOVIES.GENRE\_ID =GENRE.GENRE\_ID

**ORDER** **BY** GENRE\_ID,MOVIE\_ID ;

****

## Devuelve el nombre de todas las películas y el nombre del estudio que las ha realizado.

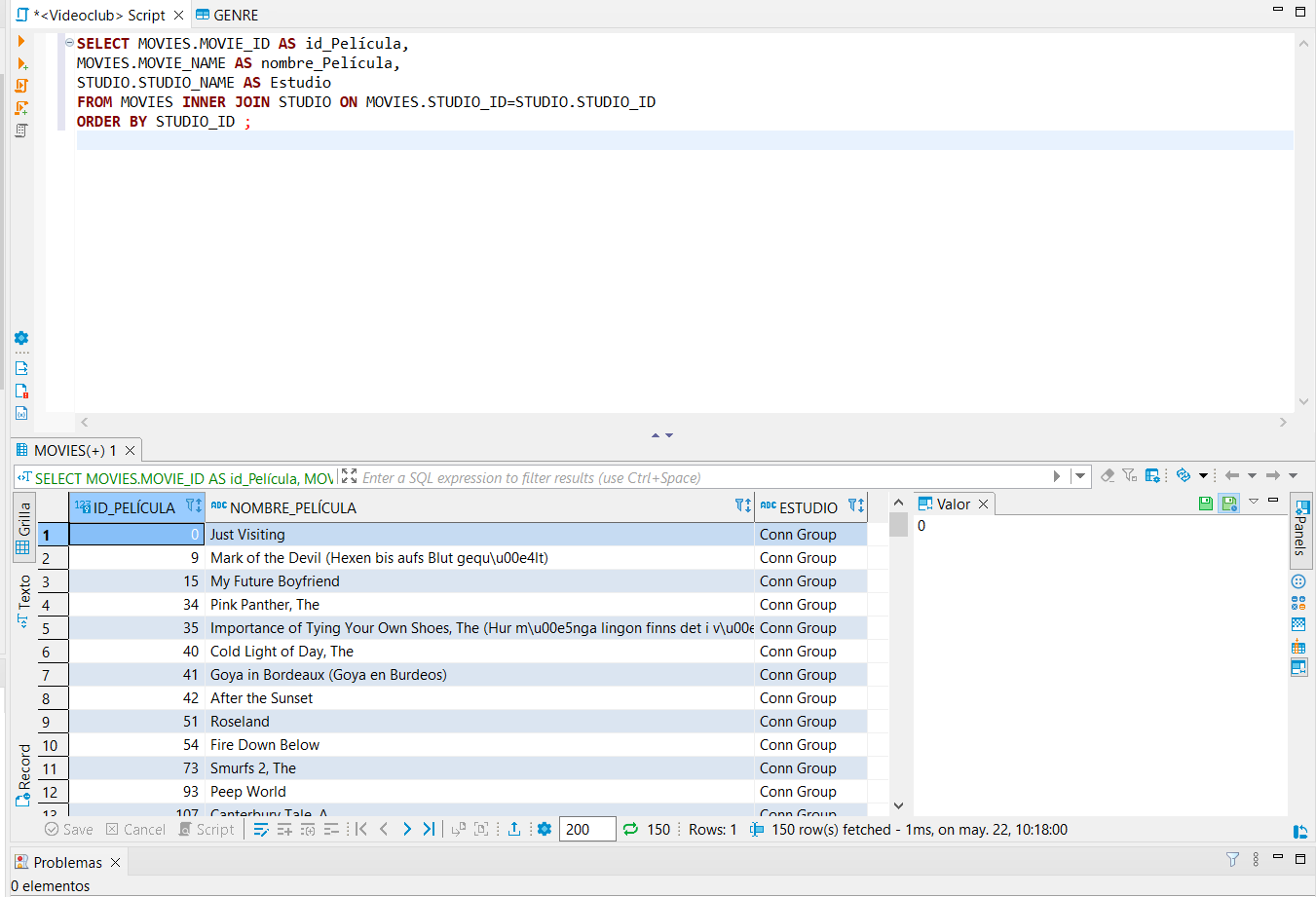
**SELECT** MOVIES.MOVIE\_ID **AS** id\_Película,

MOVIES.MOVIE\_NAME **AS** nombre\_Película,

STUDIO.STUDIO\_NAME **AS** Estudio

**FROM** MOVIES **INNER** **JOIN** STUDIO **ON** MOVIES.STUDIO\_ID=STUDIO.STUDIO\_ID

**ORDER** **BY** STUDIO\_ID ;



## Devuelve el nombre y la edad de todos los directores menores o iguales de 50 años.

### Sin filtrar por edad , primero calculamos la edad de todos (sin contar años bisiestos).

Utilizamos la función [DATEDIFF](https://www.w3schools.com/sql/func_sqlserver_datediff.asp)

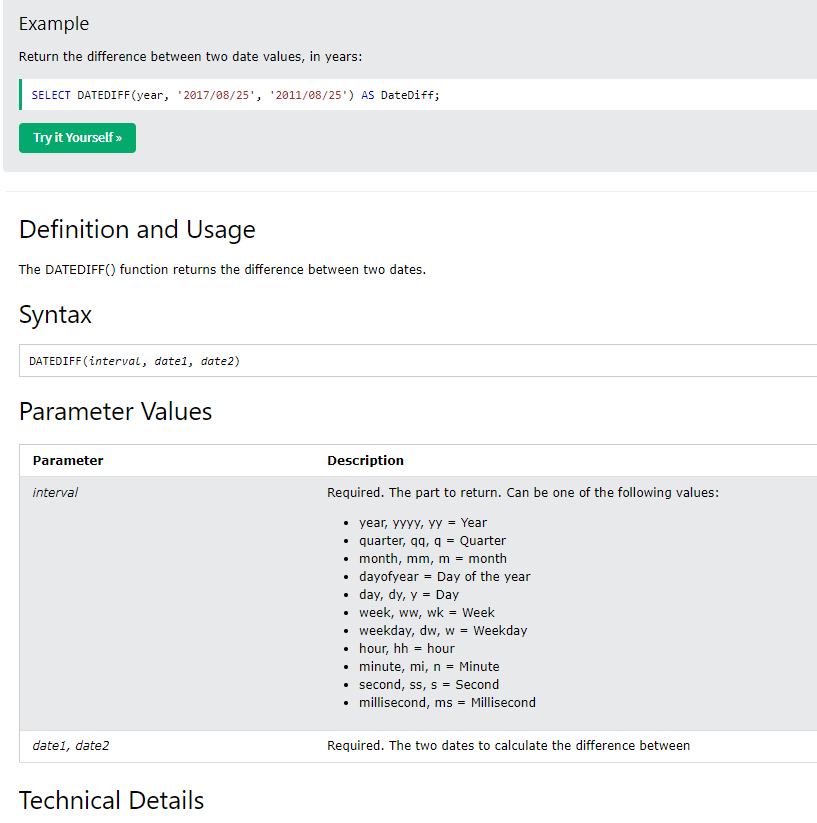


Ilustración 1 DATEDIFF()

Utilizamos la función [FLOOR](https://www.w3schools.com/sql/func_sqlserver_floor.asp)

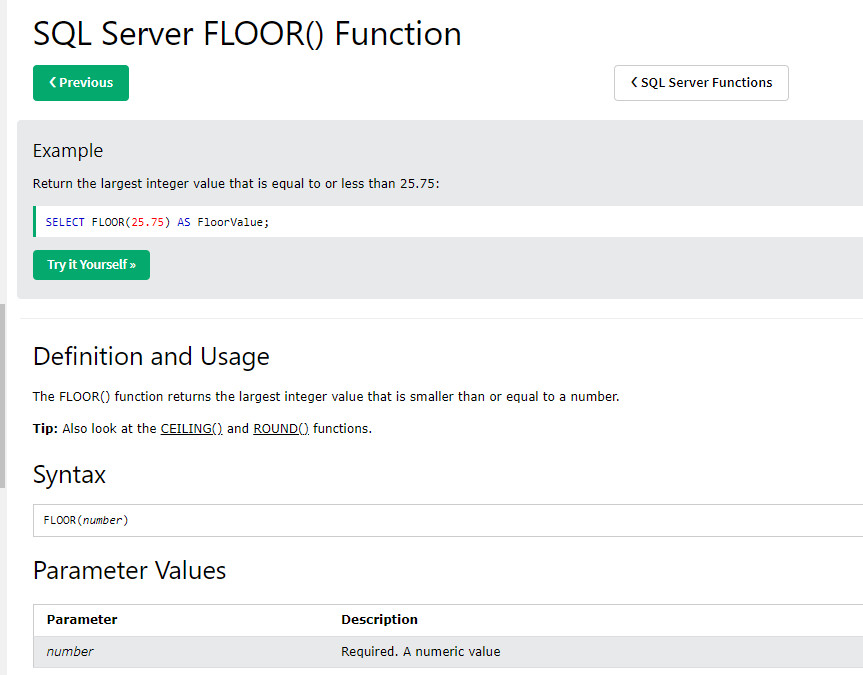
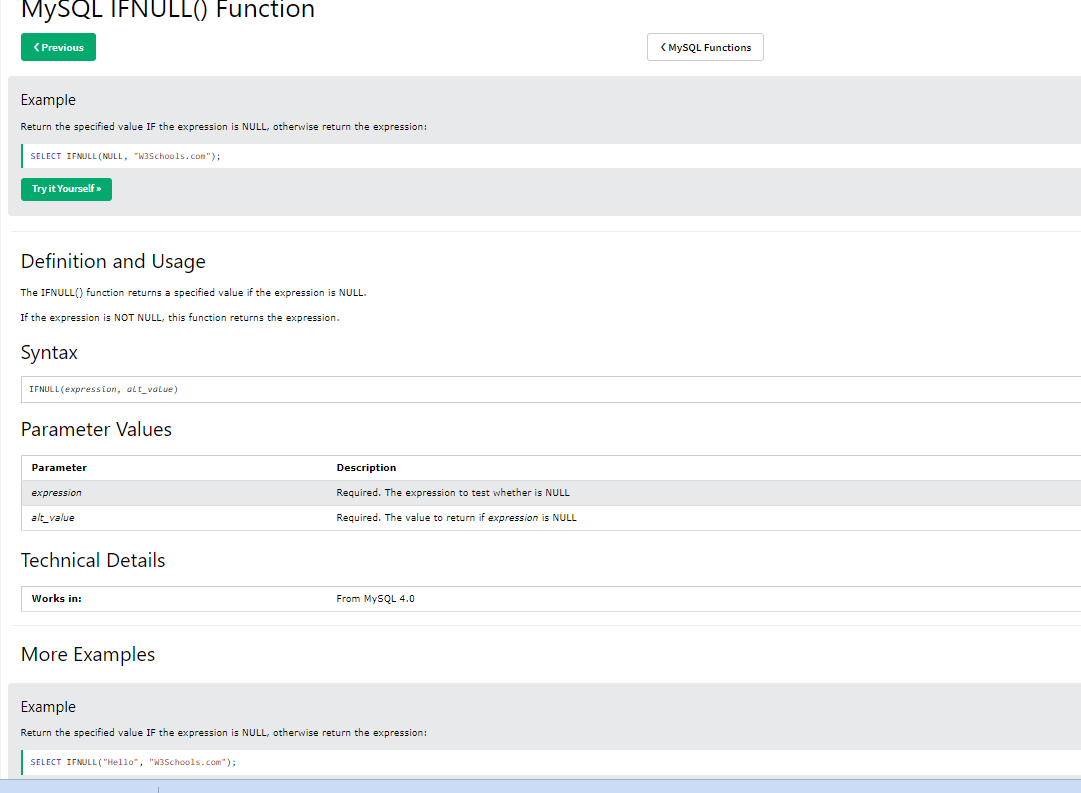
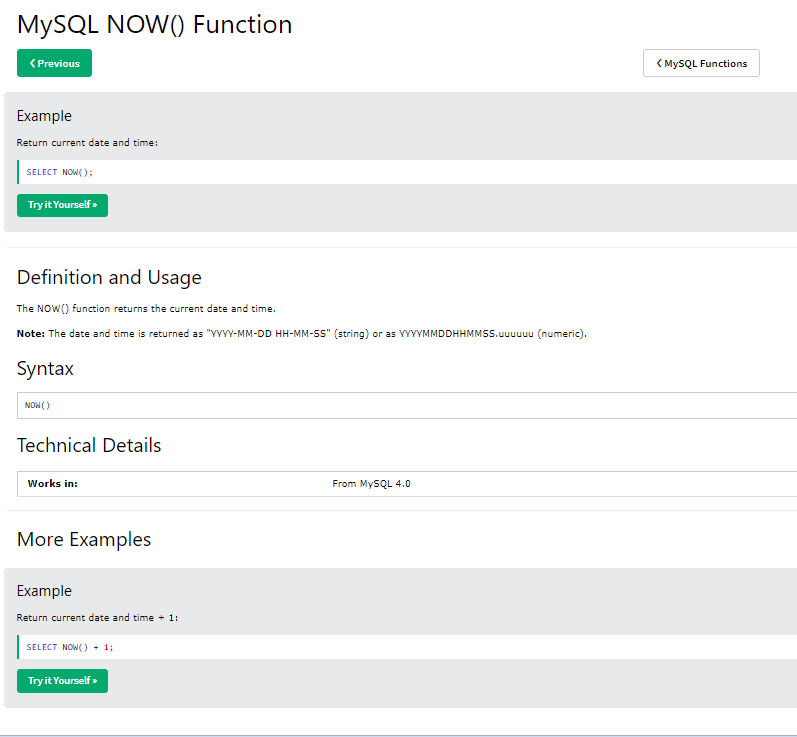


Ilustración 2 FLOOR().

Utilizamos la función [IFNULL()](https://www.w3schools.com/sql/func_mysql_ifnull.asp)



Utilizamos la función [NOW().](https://www.w3schools.com/sql/func_mysql_now.asp)



**SELECT** ACTORS.ACTOR\_ID **AS** Id\_Actor,

ACTORS.ACTOR\_NAME **AS** nombre\_Actor,

**FLOOR**(**DATEDIFF**(**IFNULL**(ACTORS.ACTOR\_DEAD\_DATE,**NOW**()),ACTORS.ACTOR\_BIRTH\_DATE)/365) **AS** edad

**FROM** ACTORS ;

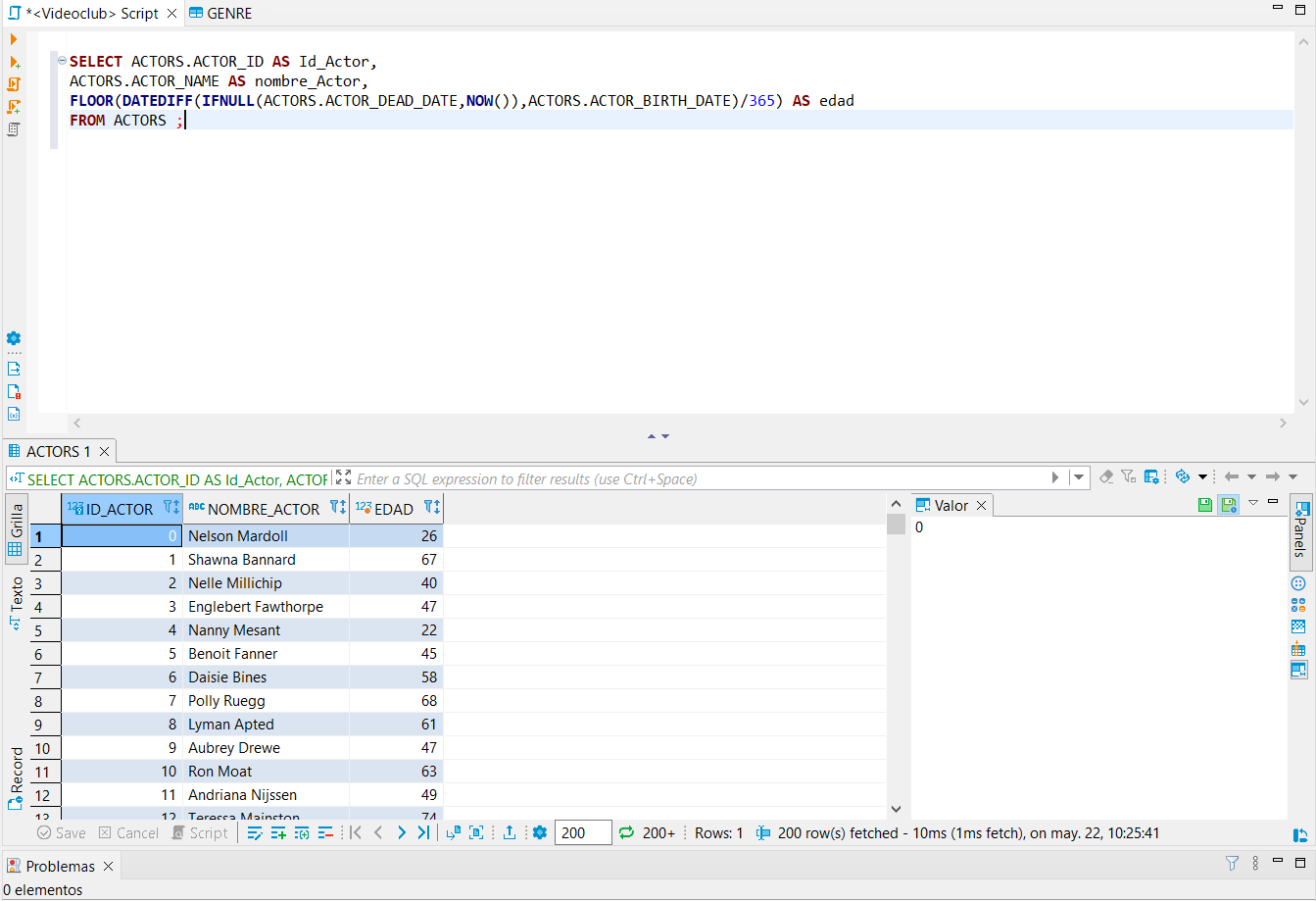


Ilustración 3 SIN FILTRAR POR EDAD

### Filtrando por menores de 50 años.

Añadimos una sentencia [WHERE.](https://www.w3schools.com/sql/sql_where.asp)



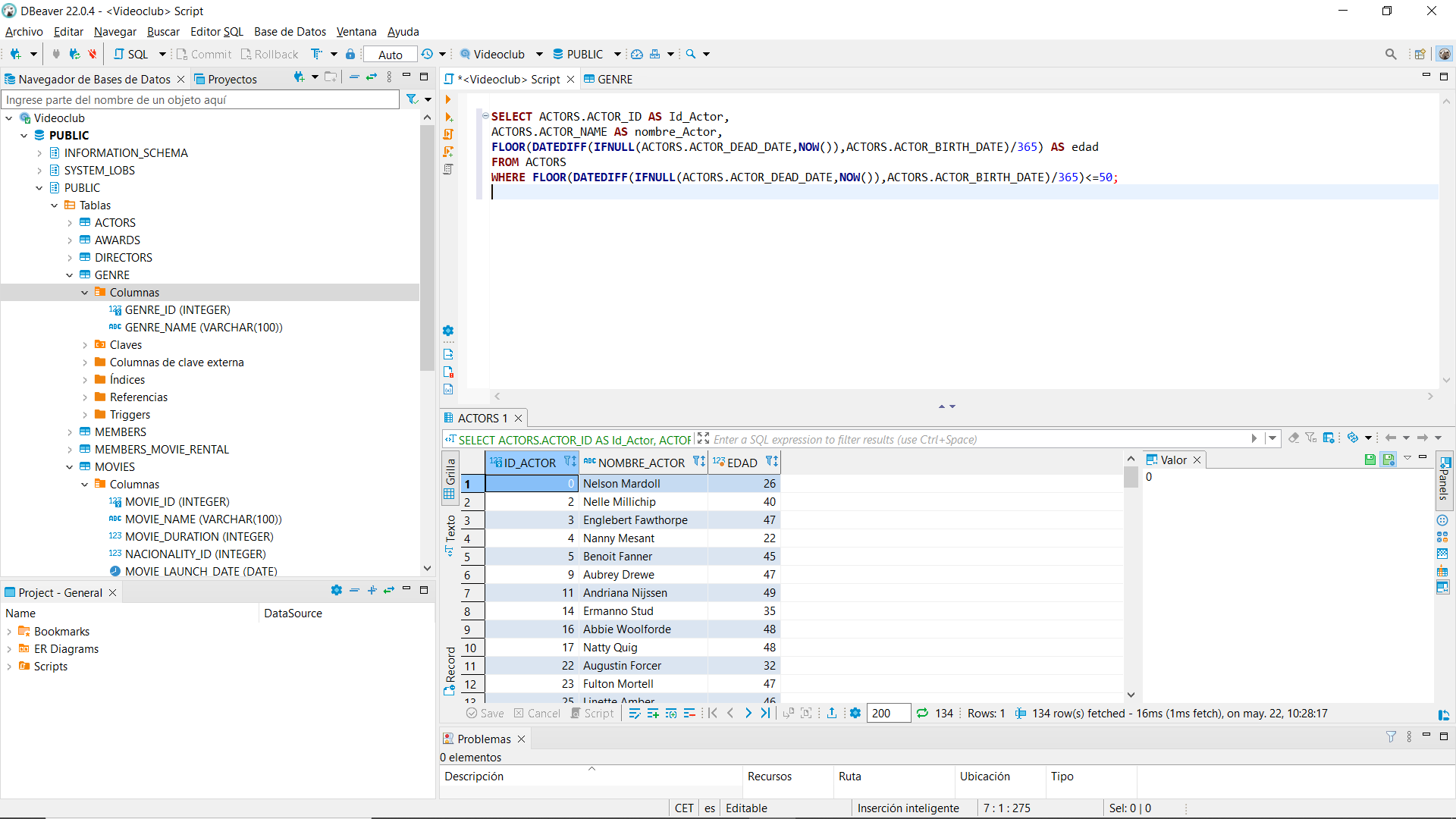
**SELECT** ACTORS.ACTOR\_ID **AS** Id\_Actor,

ACTORS.ACTOR\_NAME **AS** nombre\_Actor,

**FLOOR**(**DATEDIFF**(**IFNULL**(ACTORS.ACTOR\_DEAD\_DATE,**NOW**()),ACTORS.ACTOR\_BIRTH\_DATE)/365) **AS** edad

**FROM** ACTORS

**WHERE** **FLOOR**(**DATEDIFF**(**IFNULL**(ACTORS.ACTOR\_DEAD\_DATE,**NOW**()),ACTORS.ACTOR\_BIRTH\_DATE)/365)<=50;

****

## Devuelve el nombre y la edad de todos los actores menores de 50 años que hayan fallecido.

Usamos el comando [IS NOT NULL](https://www.w3schools.com/sql/sql_ref_is_not_null.asp) , que nos sirve para evaluar los campos no vacios, es decir aquellos que tengan fecha de muerte, claramente están muertos.

**SELECT** ACTORS.ACTOR\_ID **AS** Id\_Actor,

ACTORS.ACTOR\_NAME **AS** nombre\_Actor,

**Dateadd(year,50,** ACTORS.ACTOR\_BIRTH\_DATE**)** **AS** cumple50

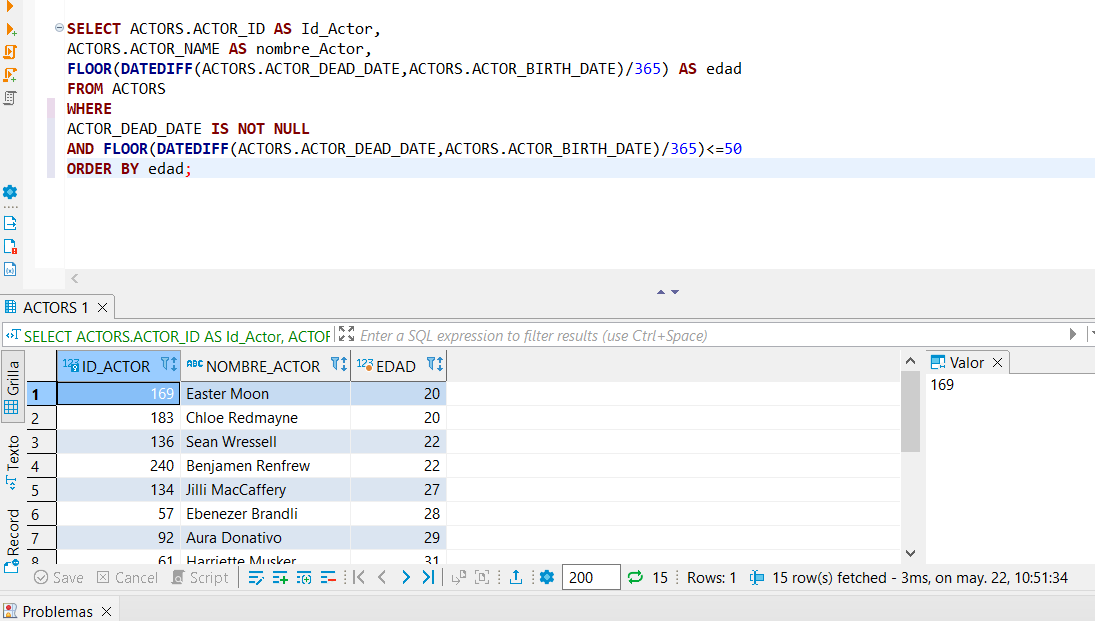
**FROM** ACTORS

**WHERE**

ACTOR\_DEAD\_DATE **IS** **NOT** **NULL**

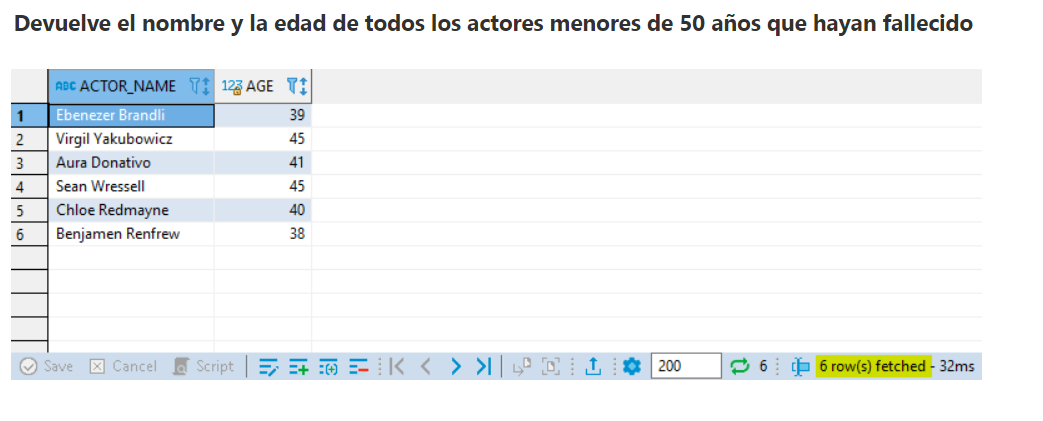
**AND** **Dateadd(year,50,** ACTORS.ACTOR\_BIRTH\_DATE**)>**ACTOR\_DEAD\_DATE

**ORDER** **BY** edad;





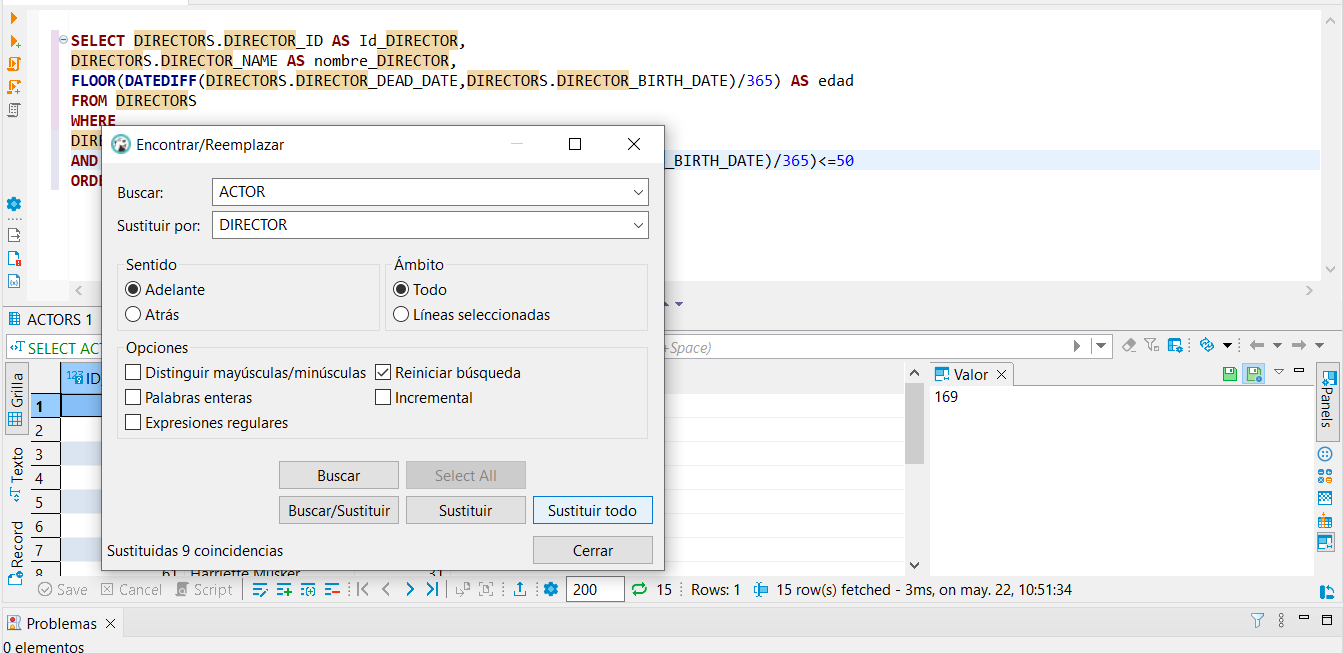
Compruebo 1-> id=170; 1984-1963=21 pero como ese año no llego a su cumpleaños (-1) =muerto a los 20.(vaya putada)



No coincide con la captura de campus dual donde solo 6 están muertos y con menos de 50 años, resultado igual que Carlos y cia.

## Devuelve el nombre de todos los directores menores o iguales de 40 años que estén vivos.

Basándonos en la consulta anterior mediante buscar y remplazar ctrl+F reemplazamos ACTORS por DIRECTORS (tabla) ACTOR por DIRECTOR (campos).



**SELECT** DIRECTORS.DIRECTOR\_ID **AS** Id\_DIRECTOR,

DIRECTORS.DIRECTOR\_NAME **AS** nombre\_DIRECTOR,

**FLOOR**(**DATEDIFF**(**NOW**(),DIRECTORS.DIRECTOR\_BIRTH\_DATE)/365) **AS** edad

**FROM** DIRECTORS

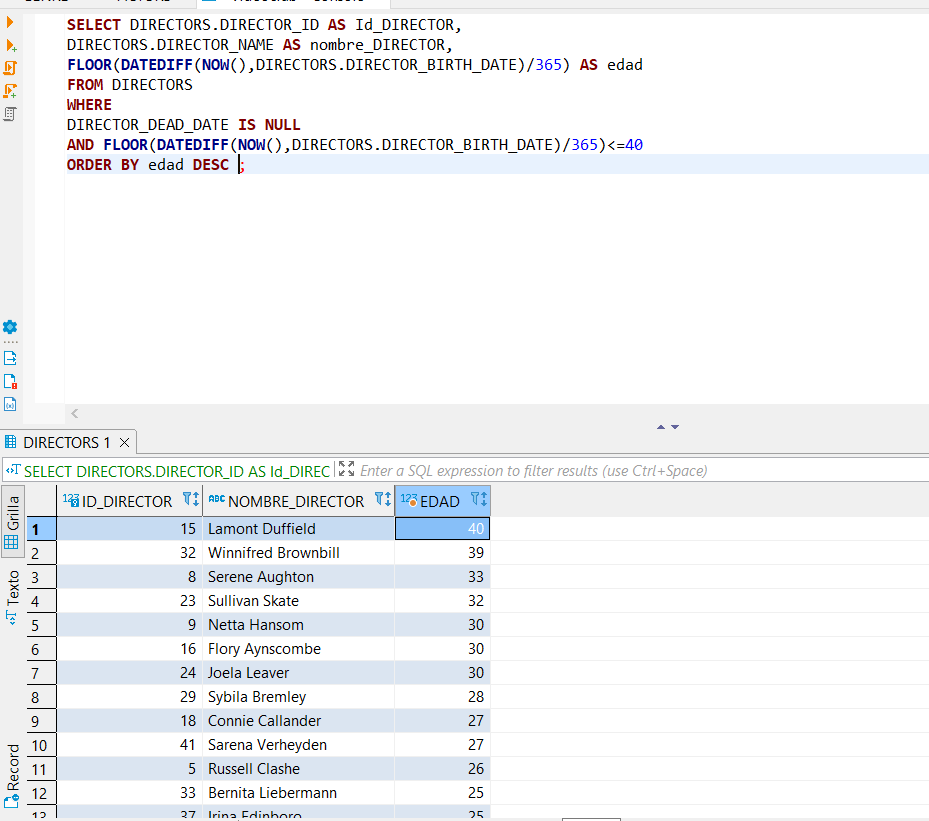
**WHERE**

DIRECTOR\_DEAD\_DATE **IS** **NULL**

**AND** **FLOOR**(**DATEDIFF**(**NOW**(),DIRECTORS.DIRECTOR\_BIRTH\_DATE)/365)<=40

**ORDER** **BY** edad;

Como debemos buscar los vivos , la fecha de muerte debe estar vacía (is null) , calculamos la edad con NOW() en lugar de la fecha de la muerte y mostramos los menores o iguales de ese calculo a 40 años , por último ordenamos por edad y así poder hacer una comprobación por limite( los de 40 años), 16 directores vivos menores de 40 años (campus dual 17).





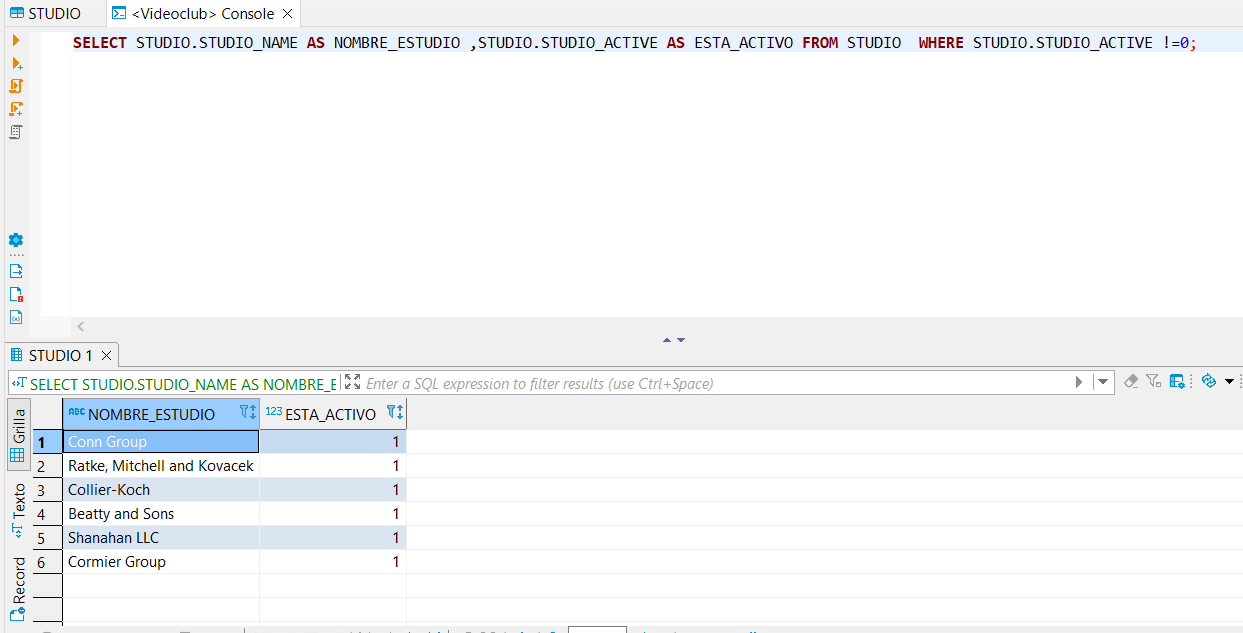
2022-1981=41 pero como aun no es agosto no llegó su cumpleaños -1 =40 años vivo.

La única diferencia con el campus dual que encuentra 17 concordancias y ahora a día de hoy son 16 es que Etti Byron ya cumplió en enero de 2022 ,41 años:



## Devuelve la lista de todos los estudios de grabación que estén activos.

**SELECT** STUDIO.STUDIO\_NAME **AS** NOMBRE\_ESTUDIO ,STUDIO.STUDIO\_ACTIVE **AS** ESTA\_ACTIVO **FROM** STUDIO **WHERE** STUDIO.STUDIO\_ACTIVE !=0;



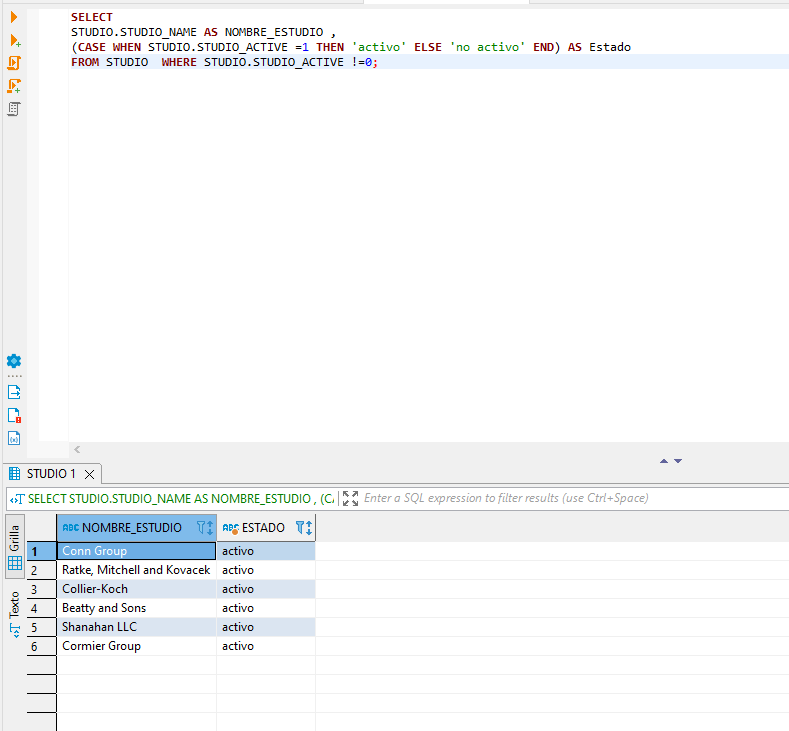
Se podría cambiar 1 por activo en el select???

**SELECT**

STUDIO.STUDIO\_NAME **AS** NOMBRE\_ESTUDIO ,

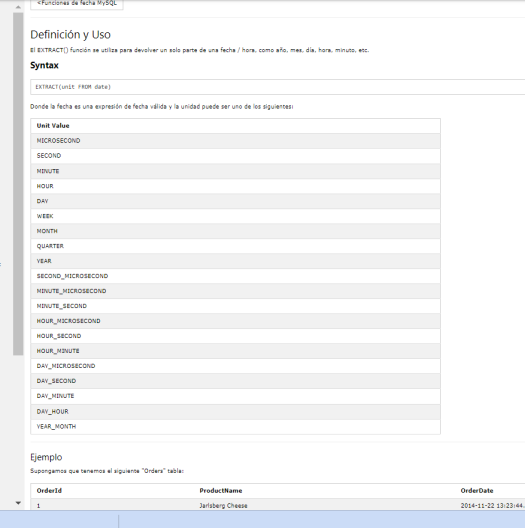
(**CASE** **WHEN** STUDIO.STUDIO\_ACTIVE =1 **THEN** 'activo' **ELSE** 'no activo' **END**) **AS** Estado

**FROM** STUDIO **WHERE** STUDIO.STUDIO\_ACTIVE !=0;



## Devuelve el nombre y el año de todas las películas que han sido producidas por un estudio que actualmente no esté activo

Utilizamos la función [Extract(unit from date)](http://www.w3bai.com/es/sql/func_extract.html).



**SELECT** M.MOVIE\_NAME **AS** Nombe\_Pelicula,

**EXTRACT** (**YEAR** **from** M.MOVIE\_LAUNCH\_DATE) **AS** Fecha\_Estreno,

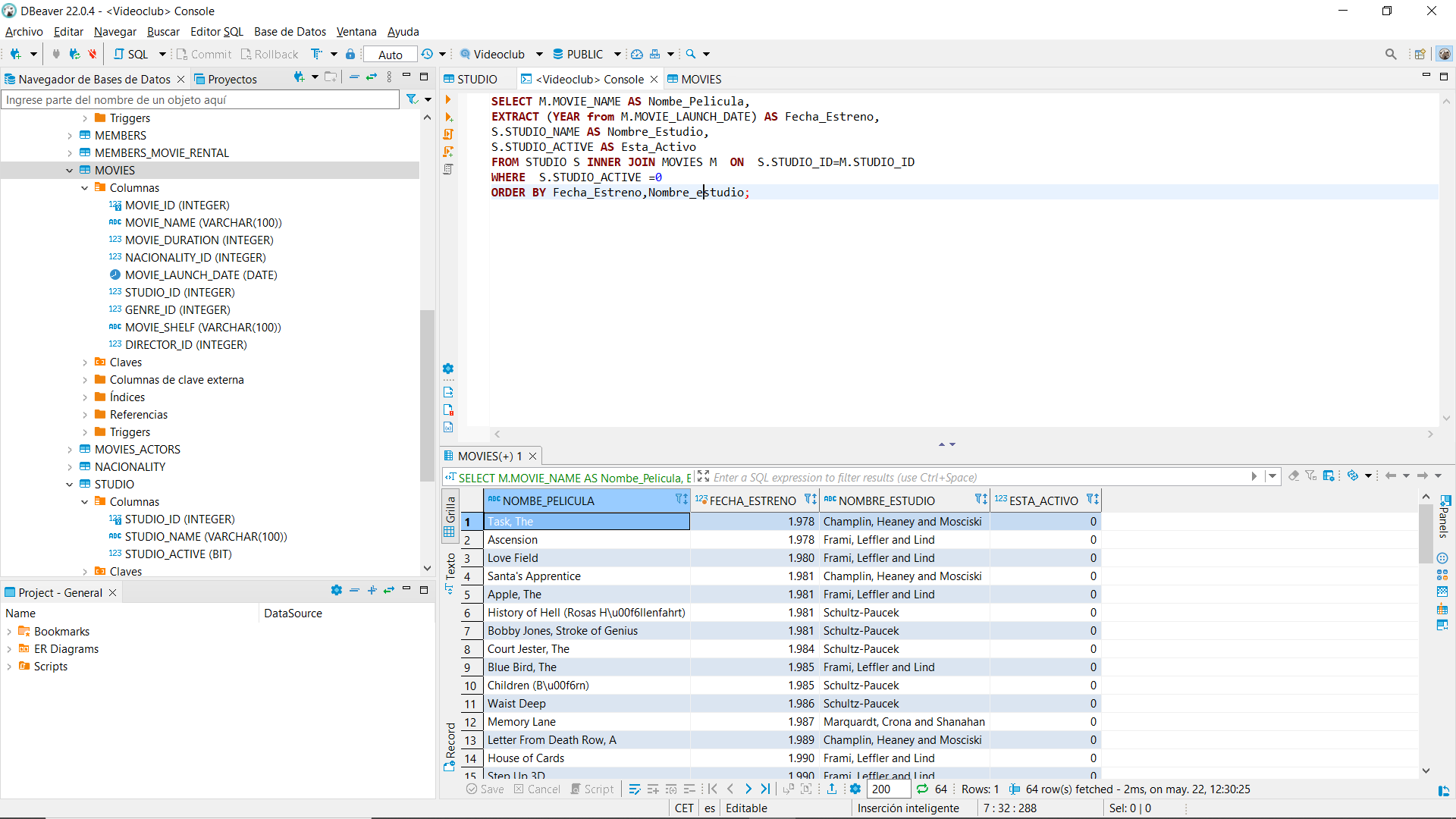
S.STUDIO\_NAME **AS** Nombre\_Estudio,

S.STUDIO\_ACTIVE **AS** Esta\_Activo

**FROM** STUDIO S **INNER** **JOIN** MOVIES M **ON** S.STUDIO\_ID=M.STUDIO\_ID

**WHERE** S.STUDIO\_ACTIVE =0

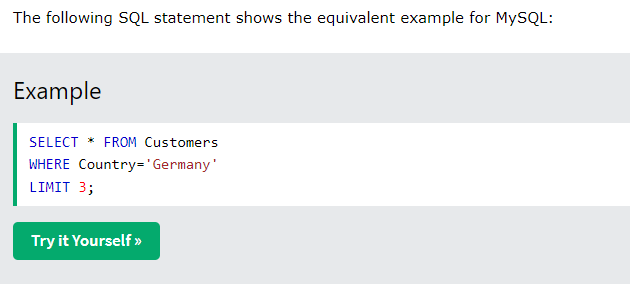
**ORDER** **BY** Fecha\_Estreno,Nombre\_estudio;



## Devuelve una lista de los 20 últimos miembros en anotarse al videoclub.

### Usando la sentencia Limit.

Usamos [LIMIT](https://www.w3schools.com/sql/sql_top.asp).



Además de los nombres comprobamos que sea así separando las fecha en día, mes y año.

**SELECT** MEMBERS.MEMBER\_NAME **AS** Nombre,

**EXTRACT**(**DAY** **FROM** MEMBERS.MEMBER\_DISCHARGE\_DATE) **AS** Dia,

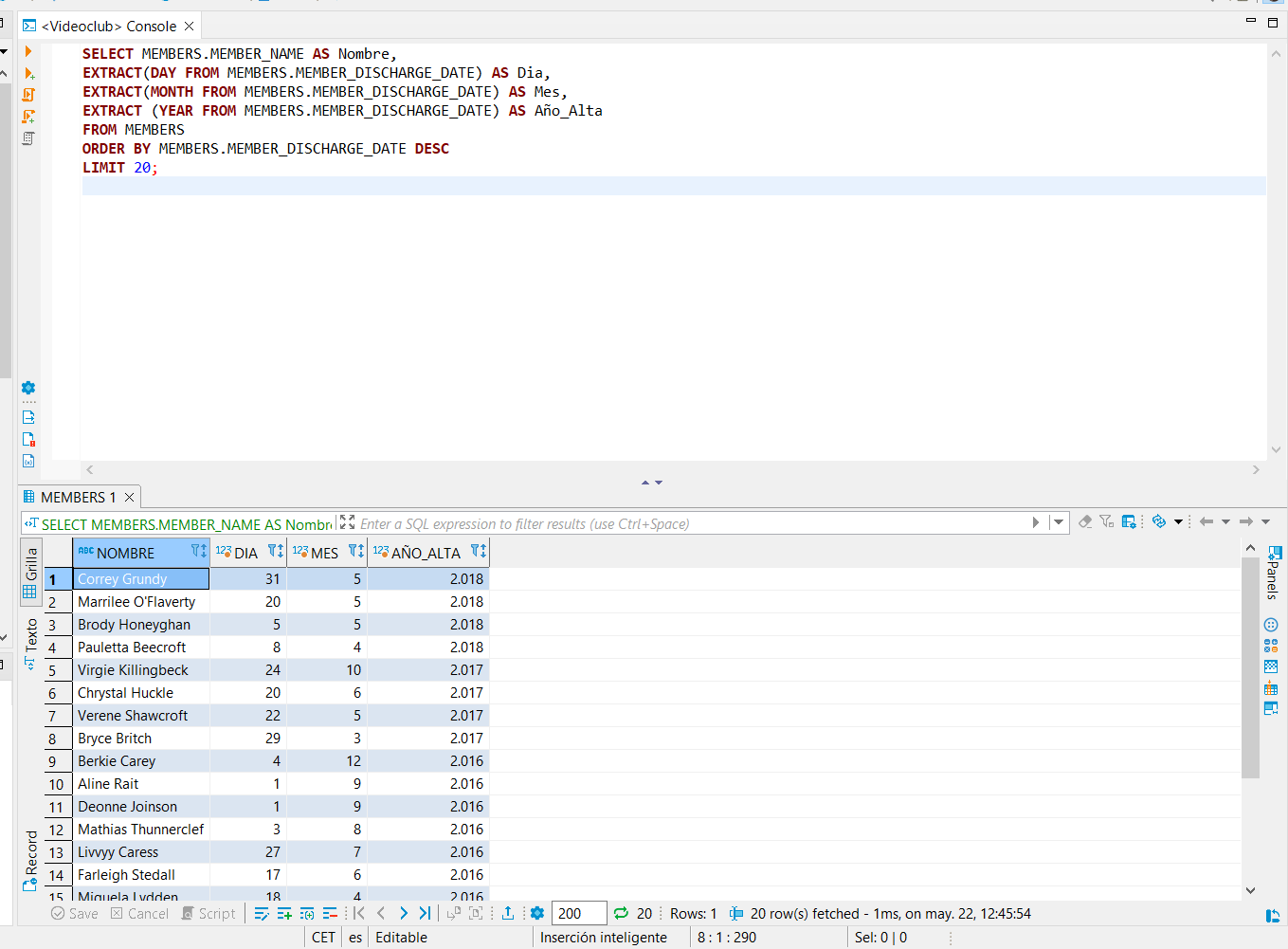
**EXTRACT**(**MONTH** **FROM** MEMBERS.MEMBER\_DISCHARGE\_DATE) **AS** Mes,

**EXTRACT** (**YEAR** **FROM** MEMBERS.MEMBER\_DISCHARGE\_DATE) **AS** Año\_Alta

**FROM** MEMBERS

**ORDER** **BY** MEMBERS.MEMBER\_DISCHARGE\_DATE **DESC**

**LIMIT** 20;



### Utilizando TOP.

Utilizando la clausula [TOP](https://www.w3schools.com/sql/sql_top.asp) esta aconsejada para tablas largas

**SELECT** TOP 20 MEMBERS.MEMBER\_NAME **AS** Nombre,

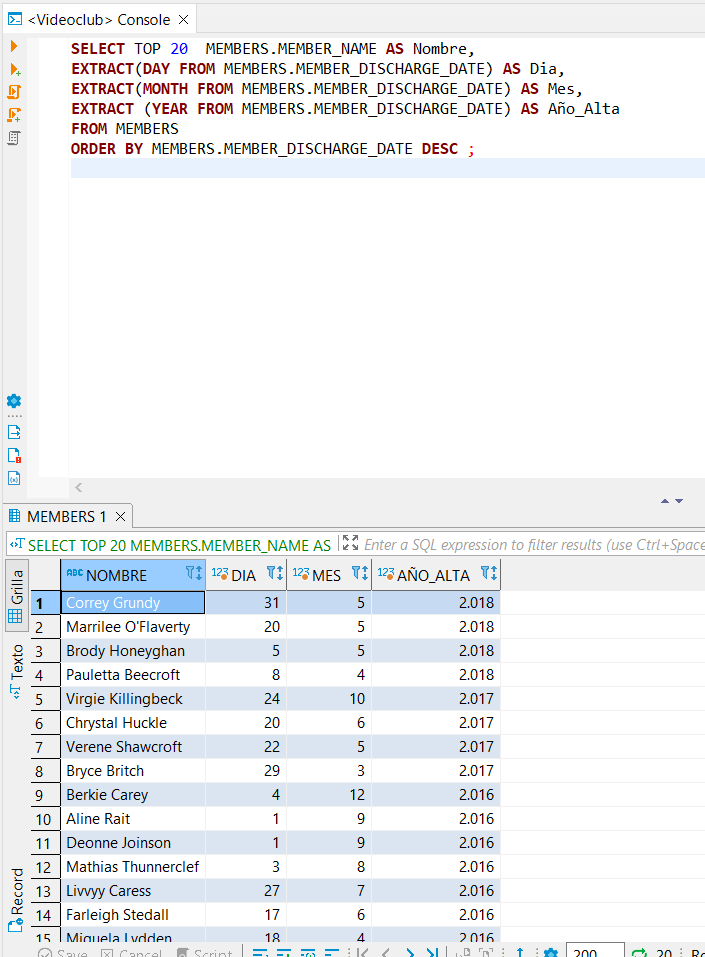
**EXTRACT**(**DAY** **FROM** MEMBERS.MEMBER\_DISCHARGE\_DATE) **AS** Dia,

**EXTRACT**(**MONTH** **FROM** MEMBERS.MEMBER\_DISCHARGE\_DATE) **AS** Mes,

**EXTRACT** (**YEAR** **FROM** MEMBERS.MEMBER\_DISCHARGE\_DATE) **AS** Año\_Alta

**FROM** MEMBERS

**ORDER** **BY** MEMBERS.MEMBER\_DISCHARGE\_DATE **DESC** ;



¿??? Que diferencia TOP LIMIT???

Permite Rango usándose en select:

La cláusula "limit" se usa para restringir los registros que se retornan en una consulta "select".

Recibe 1 ó 2 argumentos numéricos enteros positivos; el primero indica el número del primer registro a retornar, el segundo, el número máximo de registros a retornar. El número de registro inicial es 0 (no 1).

Si el segundo argumento supera la cantidad de registros de la tabla, se limita hasta el último registro.

**SELECT** **LIMIT** 1,20

MEMBERS.MEMBER\_NAME **AS** Nombre,

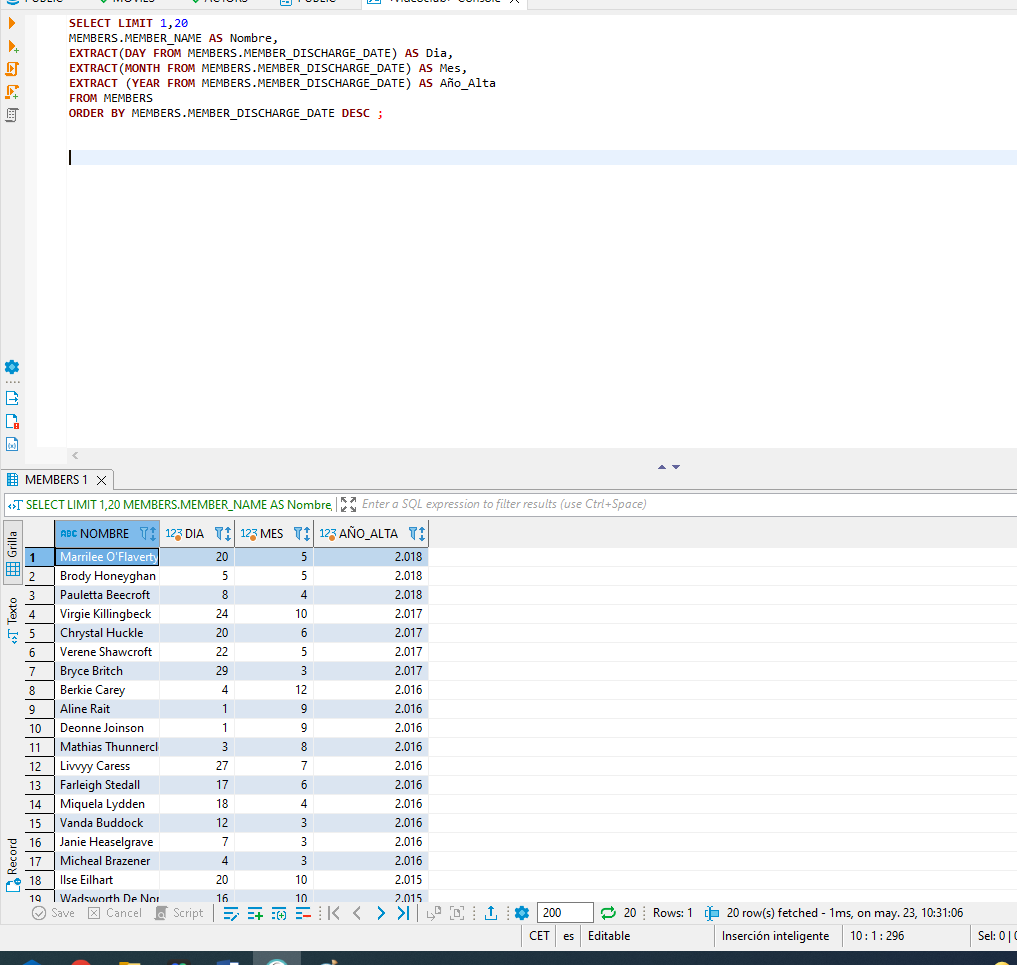
**EXTRACT**(**DAY** **FROM** MEMBERS.MEMBER\_DISCHARGE\_DATE) **AS** Dia,

**EXTRACT**(**MONTH** **FROM** MEMBERS.MEMBER\_DISCHARGE\_DATE) **AS** Mes,

**EXTRACT** (**YEAR** **FROM** MEMBERS.MEMBER\_DISCHARGE\_DATE) **AS** Año\_Alta

**FROM** MEMBERS

**ORDER** **BY** MEMBERS.MEMBER\_DISCHARGE\_DATE **DESC** ;



## Devuelve una lista de las últimas 10 películas que se han alquilado.

**SELECT** TOP 10

M.MOVIE\_NAME **AS** Nombe\_Pelicula,

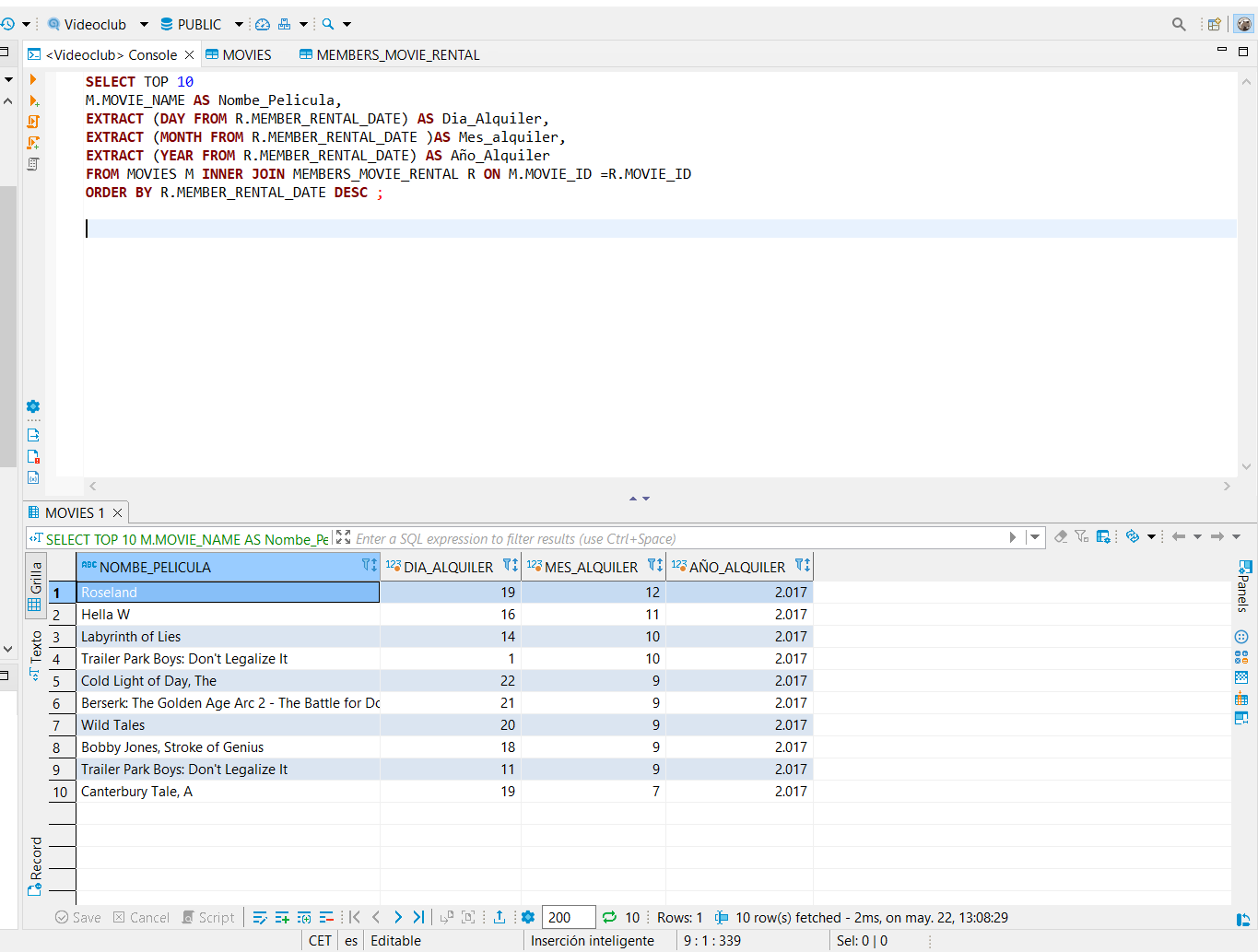
**EXTRACT** (**DAY** **FROM** R.MEMBER\_RENTAL\_DATE) **AS** Dia\_Alquiler,

**EXTRACT** (**MONTH** **FROM** R.MEMBER\_RENTAL\_DATE )**AS** Mes\_alquiler,

**EXTRACT** (**YEAR** **FROM** R.MEMBER\_RENTAL\_DATE) **AS** Año\_Alquiler

**FROM** MOVIES M **INNER** **JOIN** MEMBERS\_MOVIE\_RENTAL R **ON** M.MOVIE\_ID =R.MOVIE\_ID

**ORDER** **BY** R.MEMBER\_RENTAL\_DATE **DESC** ;



**SELECT**

M.MOVIE\_NAME **AS** Nombe\_Pelicula,

**EXTRACT** (**DAY** **FROM** R.MEMBER\_RENTAL\_DATE) **AS** Dia\_Alquiler,

**EXTRACT** (**MONTH** **FROM** R.MEMBER\_RENTAL\_DATE )**AS** Mes\_alquiler,

**EXTRACT** (**YEAR** **FROM** R.MEMBER\_RENTAL\_DATE) **AS** Año\_Alquiler

**FROM** MOVIES M **INNER** **JOIN** MEMBERS\_MOVIE\_RENTAL R **ON** M.MOVIE\_ID =R.MOVIE\_ID

**ORDER** **BY** R.MEMBER\_RENTAL\_DATE **DESC**

**LIMIT** 10;

**Produce la misma salida.**

## Indica cual es el nombre del director del que más películas se han alquilado

**Mirar opción distintc**

**SELECT**

D.DIRECTOR\_NAME **AS** Nombre\_Director,

**COUNT**(D.DIRECTOR\_Id) **AS** Alquileres

**FROM** MOVIES M

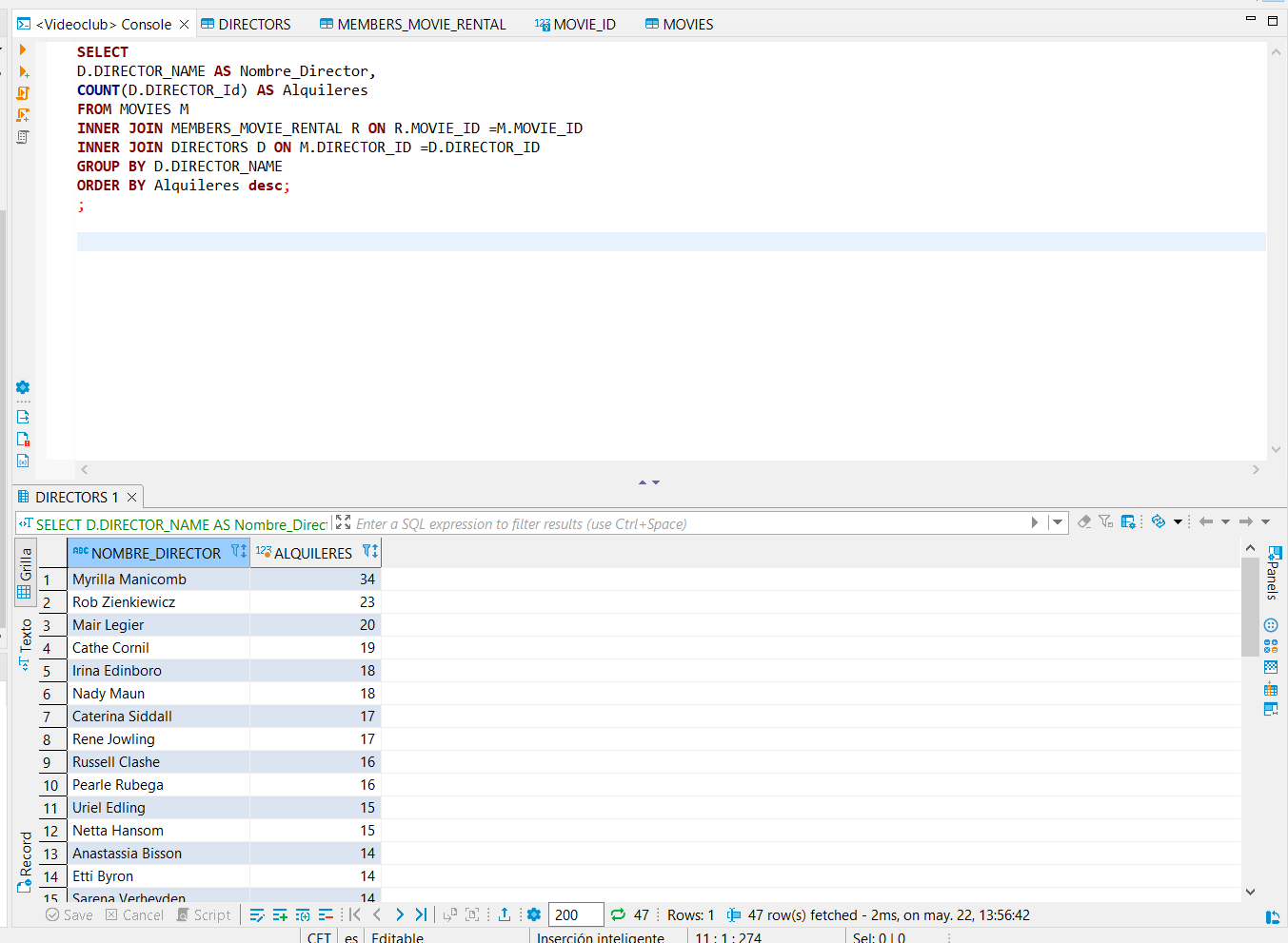
**INNER** **JOIN** MEMBERS\_MOVIE\_RENTAL R **ON** R.MOVIE\_ID =M.MOVIE\_ID

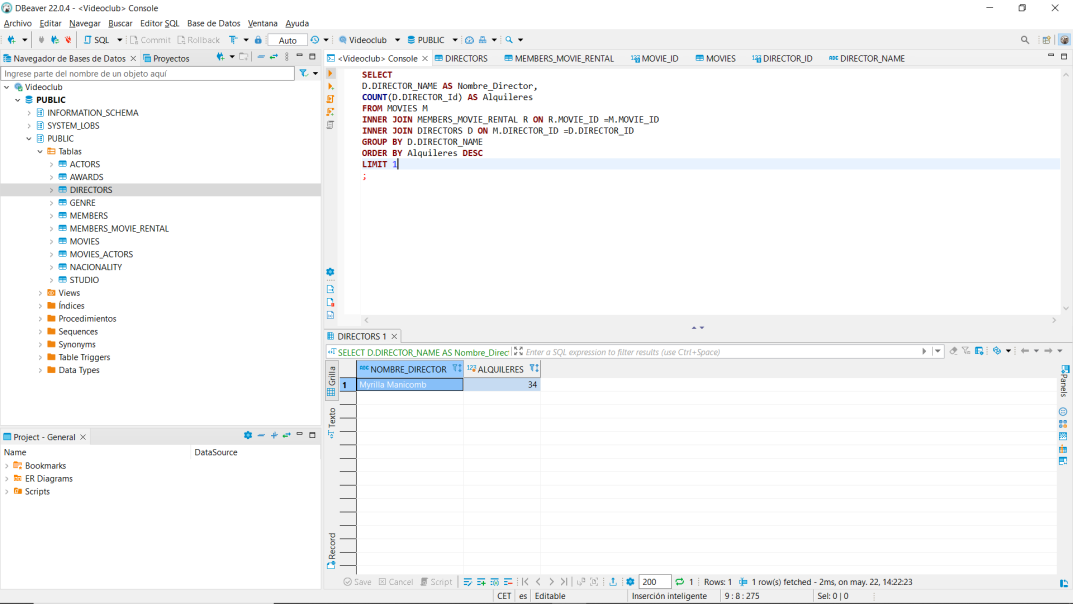
**INNER** **JOIN** DIRECTORS D **ON** M.DIRECTOR\_ID =D.DIRECTOR\_ID

**GROUP** **BY** D.DIRECTOR\_NAME

**ORDER** **BY** Alquileres **desc**;

Añadiendo Limit 1 , ¿???No coincide con el campus Se comprueba con Excel

****

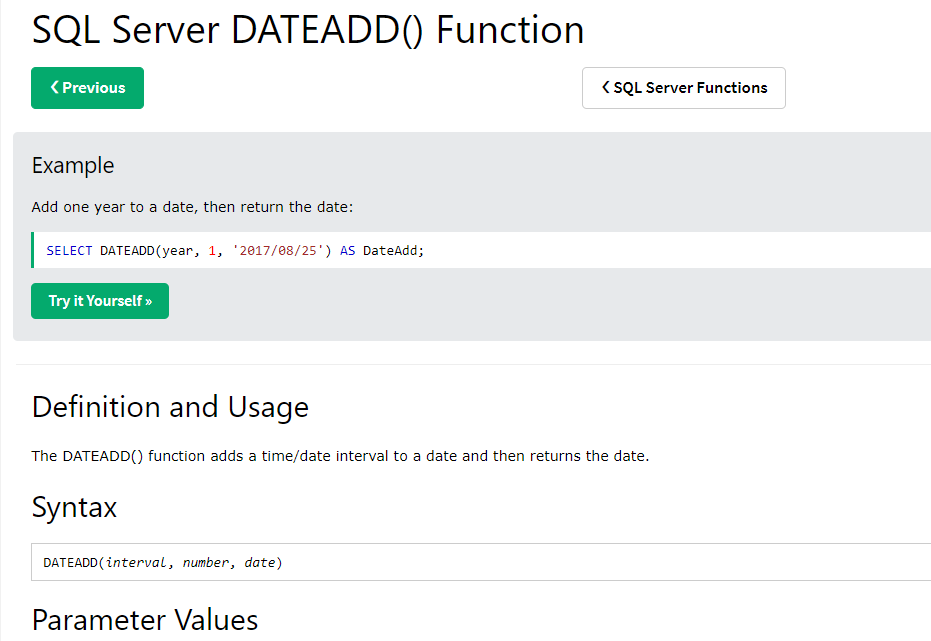
****

## Indica cuantos premios han ganado cada uno de los estudios con las películas que han creado.

No entiendo la tabla de premios

## Indica cuántas películas ha realizado cada director antes de cumplir 41 años.

Utilizamos la función [DATEADD](https://www.w3schools.com/sql/func_sqlserver_dateadd.asp) para calcular la fecha a los 41 cumpleaños, no comprobamos si está muerto por consistencia de Base de datos.



**SELECT**

D.DIRECTOR\_NAME **AS** Nombre\_Director,

--DATEADD(YEAR,41,D.DIRECTOR\_BIRTH\_DATE) AS Fecha\_41\_Cumpleaños,

**COUNT**(M.MOVIE\_NAME) **AS** n\_Peliculas

--M.MOVIE\_LAUNCH\_DATE AS Lanzamiento

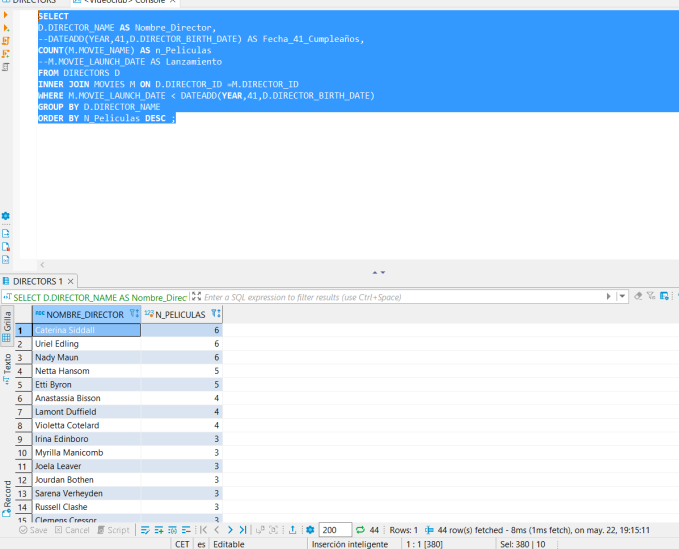
**FROM** DIRECTORS D

**INNER** **JOIN** MOVIES M **ON** D.DIRECTOR\_ID =M.DIRECTOR\_ID

**WHERE** M.MOVIE\_LAUNCH\_DATE < DATEADD(**YEAR**,41,D.DIRECTOR\_BIRTH\_DATE)

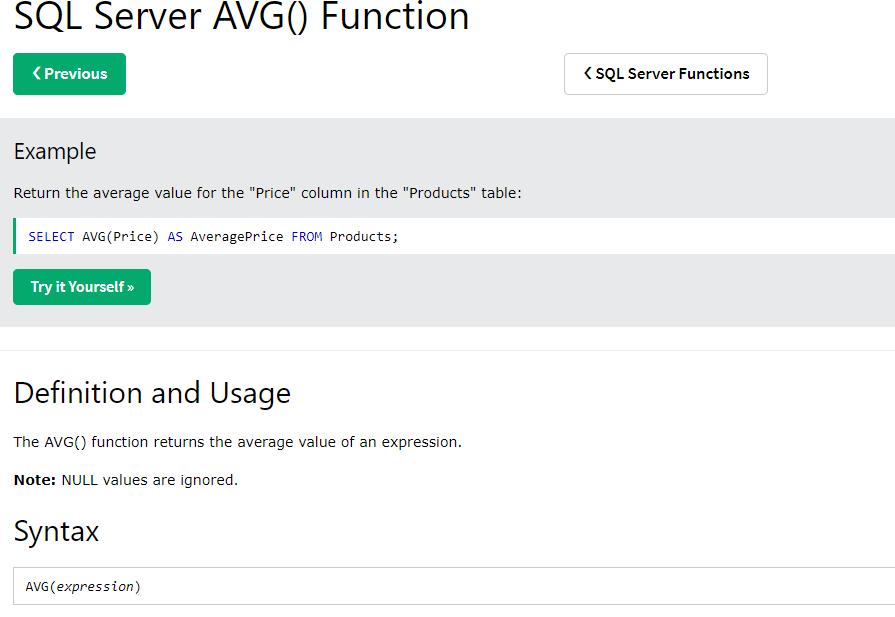
**GROUP** **BY** D.DIRECTOR\_NAME

**ORDER** **BY** N\_Peliculas **DESC** ;



## Indica la edad media de los directores vivos.

La función [AVG](https://www.w3schools.com/sql/func_sqlserver_avg.asp) calcula la media de los datos .



**SELECT**

**AVG**(**DATEDIFF**( **now**(),DIRECTORS.DIRECTOR\_BIRTH\_DATE)/365) **AS** media\_Edad\_Vivos

**FROM** DIRECTORS

**WHERE** DIRECTOR\_DEAD\_DATE **IS** **NULL**

;

****

Tiene sentido con el resultado bootcamp 46 el año pasado

## Indica la edad media de los actores que han fallecido.

Solo cambiamos la condición de where por IS NOT NULL (si la fecha de muerte no es nula viene a decir que están muertos). 6 muertos

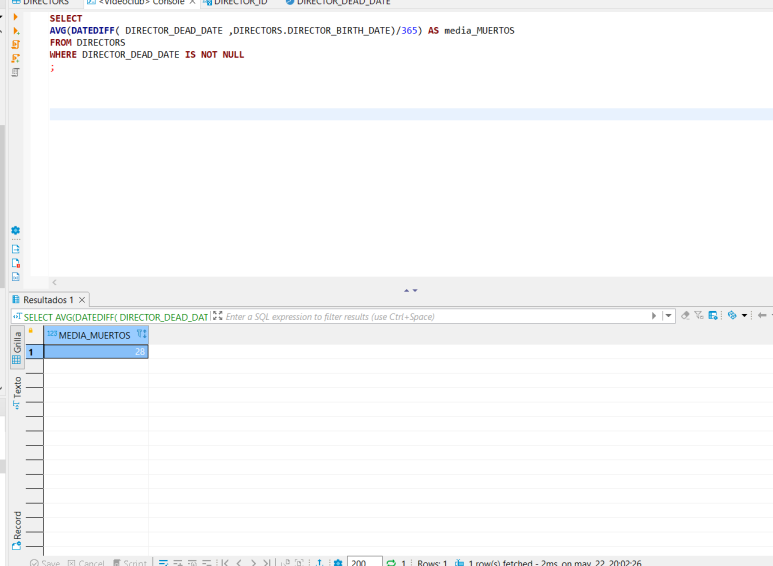
**SELECT**

**AVG**(**DATEDIFF**( DIRECTOR\_DEAD\_DATE ,DIRECTORS.DIRECTOR\_BIRTH\_DATE)/365) **AS** media\_MUERTOS

**FROM** DIRECTORS

**WHERE** DIRECTOR\_DEAD\_DATE **IS** **NOT** **NULL**

;



No coincide con el BOOTCAM 32 SON Actores no directores

### Comprobación Manual

Resultado 44 vivos con edad media 47 y 6 muertos con edad media de 28

La media ponderada será (44x47+6x28)/50=(2068+168)años/50 actores=44,72 aprox 45

### Comprobación Manual

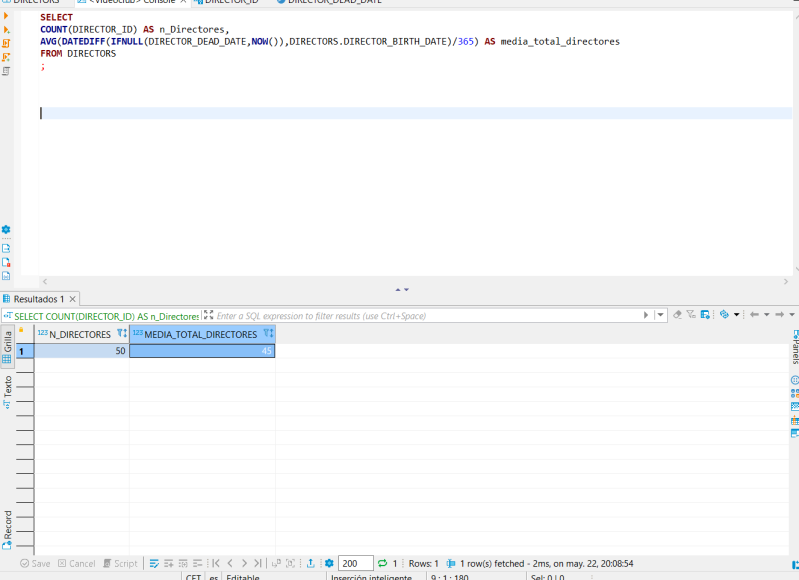
**SELECT**

**COUNT**(DIRECTOR\_ID) **AS** n\_Directores,

**AVG**(**DATEDIFF**(**IFNULL**(DIRECTOR\_DEAD\_DATE,**NOW**()),DIRECTORS.DIRECTOR\_BIRTH\_DATE)/365) **AS** media\_total\_directores

**FROM** DIRECTORS

;



## Indica cuál es el género favorito de cada uno de los directores cuando dirigen una película

Falta filtra por la primera fila de cada director.

GROUP\_CONCAT(GROUPID.GENRE\_NAME) AS GENRE\_NAME

**SELECT**

D.DIRECTOR\_NAME **AS** NOMBRE,

G.GENRE\_NAME **AS** NOMBRE\_GENERO,

**COUNT**(G.GENRE\_NAME )

**FROM** DIRECTORS D

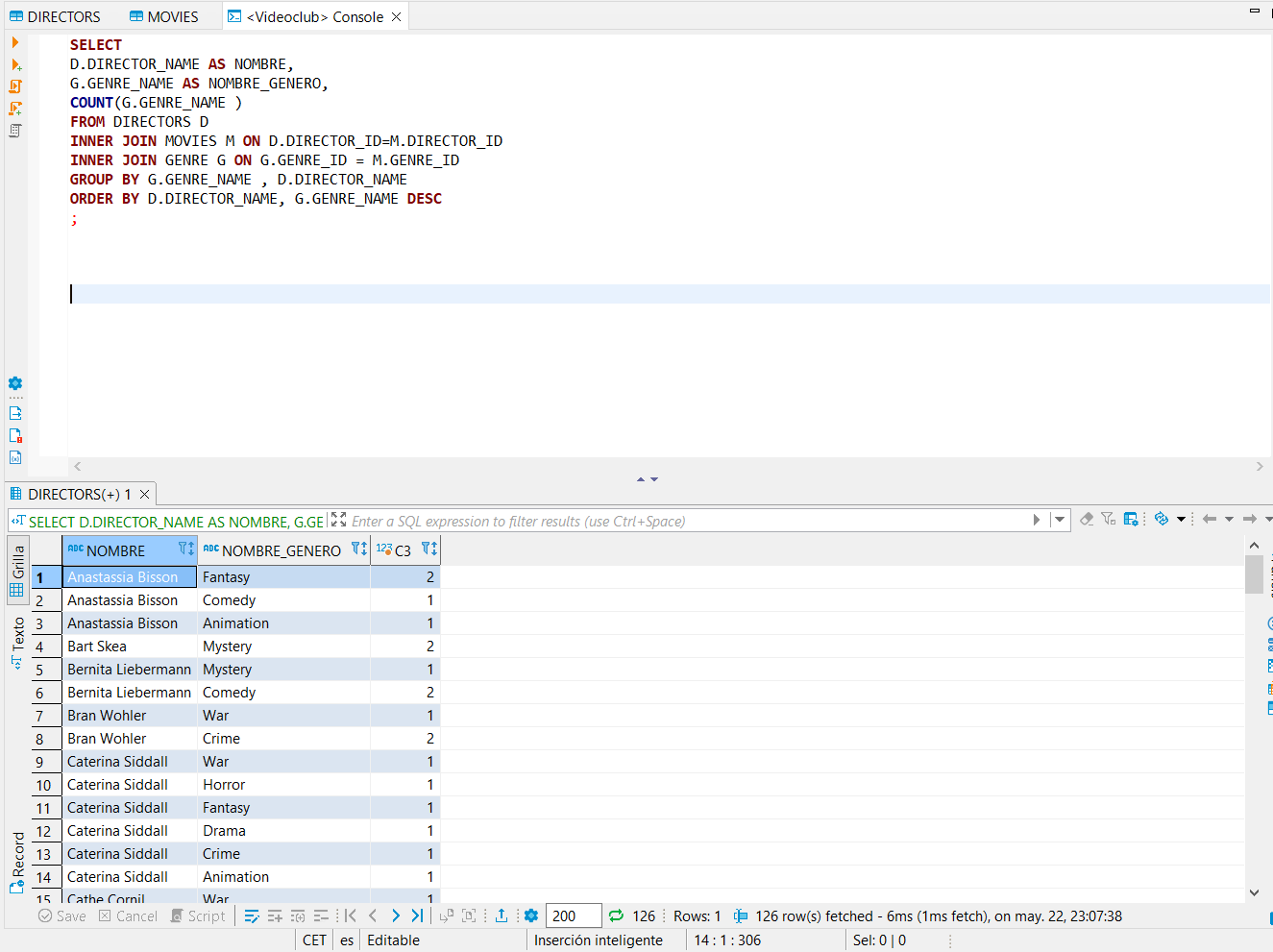
**INNER** **JOIN** MOVIES M **ON** D.DIRECTOR\_ID=M.DIRECTOR\_ID

**INNER** **JOIN** GENRE G **ON** G.GENRE\_ID = M.GENRE\_ID

**GROUP** **BY** G.GENRE\_NAME , D.DIRECTOR\_NAME

**ORDER** **BY** D.DIRECTOR\_ID, G.GENRE\_NAME **DESC**

;

****

SELECT col1, col2, ..., colN

GROUP\_CONCAT ( [DISTINCT] col\_name1

[ORDER BY clause] [SEPARATOR str\_val] )

FROM table\_name GROUP BY col\_name2;

**SELECT**

DIRECTORES\_GENERO.DIRECTOR\_ID **AS** D\_ID,

GROUP\_CONCAT(DIRECTORES\_GENERO.GENERO)

**FROM** (**SELECT**

D.DIRECTOR\_ID ,

G.GENRE\_NAME **AS** GENERO,

**COUNT**(G.GENRE\_NAME) **AS** VECES

**FROM** DIRECTORS D

**INNER** **JOIN** MOVIES M **ON** D.DIRECTOR\_ID=M.DIRECTOR\_ID

**INNER** **JOIN** GENRE G **ON** G.GENRE\_ID = M.GENRE\_ID

**GROUP** **BY** G.GENRE\_NAME , D.DIRECTOR\_ID

**ORDER** **BY** D.DIRECTOR\_ID, G.GENRE\_NAME **DESC**

) DIRECTORES\_GENERO **LEFT** **JOIN** (**SELECT**

DIRECTOR\_ID **AS** DIRECTOR\_ID,

**MAX**(N\_PELIS) **AS** MAX\_PELI

**FROM** (

**SELECT**

D.DIRECTOR\_ID ,

G.GENRE\_ID **AS** GENERO,

**COUNT**(M.MOVIE\_ID) **AS** N\_PELIS

**FROM** DIRECTORS D

**INNER** **JOIN** MOVIES M **ON** D.DIRECTOR\_ID=M.DIRECTOR\_ID

**INNER** **JOIN** GENRE G **ON** G.GENRE\_ID = M.GENRE\_ID

**GROUP** **BY** G.GENRE\_ID ,D.DIRECTOR\_ID )

**GROUP** **BY** DIRECTOR\_ID

)MAX\_GENERO\_DIR **ON** DIRECTORES\_GENERO.DIRECTOR\_ID=MAX\_GENERO\_DIR.DIRECTOR\_ID **AND** DIRECTORES\_GENERO.VECES=MAX\_GENERO\_DIR.MAX\_PELI

**GROUP** **BY** DIRECTORES\_GENERO.DIRECTOR\_ID ;

## Indica cuál es la nacionalidad favorita de cada uno de los estudios en la producción de las películas

## Indica cuál es la media de duración de las películas de cada director.

**SELECT** D.DIRECTOR\_NAME **AS** nombre\_Director,

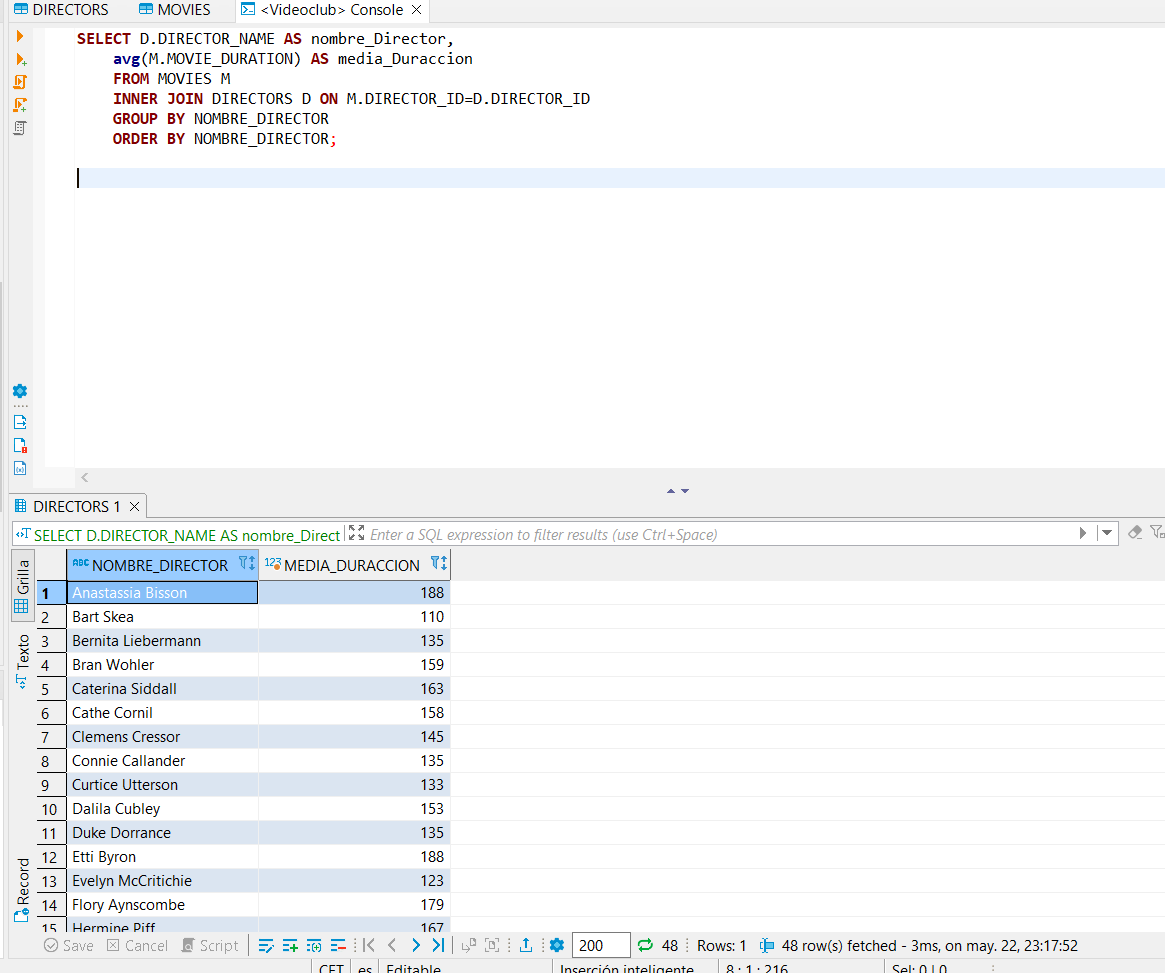
**avg**(M.MOVIE\_DURATION) **AS** media\_Duraccion

**FROM** MOVIES M

**INNER** **JOIN** DIRECTORS D **ON** M.DIRECTOR\_ID=D.DIRECTOR\_ID

**GROUP** **BY** NOMBRE\_DIRECTOR

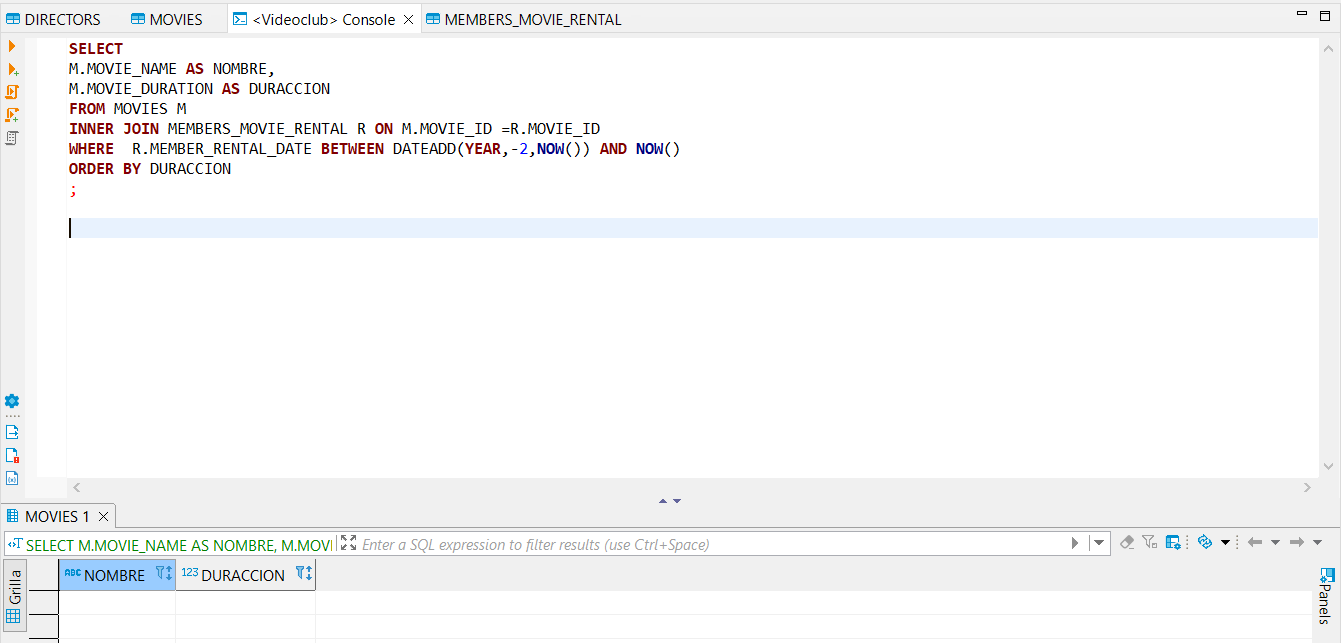
**ORDER** **BY** NOMBRE\_DIRECTOR;



## Indica cuál es la el nombre y la duración mínima de las películas que han sido alquiladas en los últimos 2 años por los miembros del videoclub

En los últimos dos años el resultado es 0.

La clausula BETWEEN debe estar ordenada de menor a mayor fecha



Encontramos que el último año con resultados es hace 5 años (2017)

**SELECT**

M.MOVIE\_NAME **AS** NOMBRE,

M.MOVIE\_DURATION **AS** DURACCION

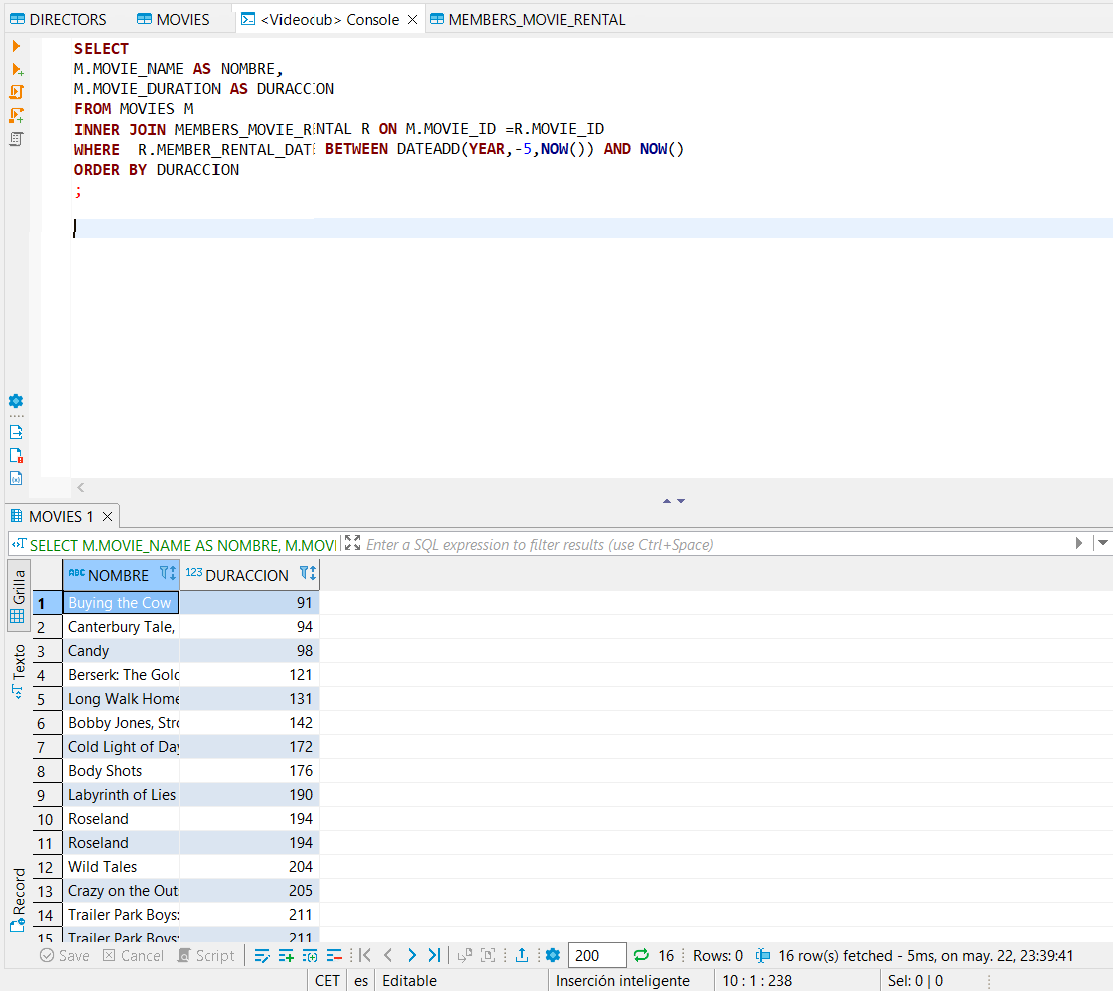
**FROM** MOVIES M

**INNER** **JOIN** MEMBERS\_MOVIE\_RENTAL R **ON** M.MOVIE\_ID =R.MOVIE\_ID

**WHERE** R.MEMBER\_RENTAL\_DATE **BETWEEN** DATEADD(**YEAR**,-5,**NOW**()) **AND** **NOW**()

**ORDER** **BY** DURACCION

;



Otra manera seria con la fecha literal:

**SELECT**

M.MOVIE\_NAME **AS** NOMBRE,

M.MOVIE\_DURATION **AS** DURACCION

**FROM** MOVIES M

**INNER** **JOIN** MEMBERS\_MOVIE\_RENTAL R **ON** M.MOVIE\_ID =R.MOVIE\_ID

**WHERE** R.MEMBER\_RENTAL\_DATE >='2017-01-01'

**ORDER** **BY** DURACCION

;

## Indica cuál fue la primera película que alquilaron los miembros del videoclub cuyos teléfonos tengan como último dígito el ID de alguna nacionalidad

## Indica el número de premios a los que estuvo nominado un actor, pero que no ha conseguido (Si una película está nominada a un premio, su actor también lo está).

**SELECT**

A.ACTOR\_NAME **AS** Actor\_Nombre,

--M.MOVIE\_NAME AS NOMBRE\_PELICULA,

**SUM**(P.AWARD\_NOMINATION) **AS** SUMA\_NOMINACIONES

**FROM** MOVIES M

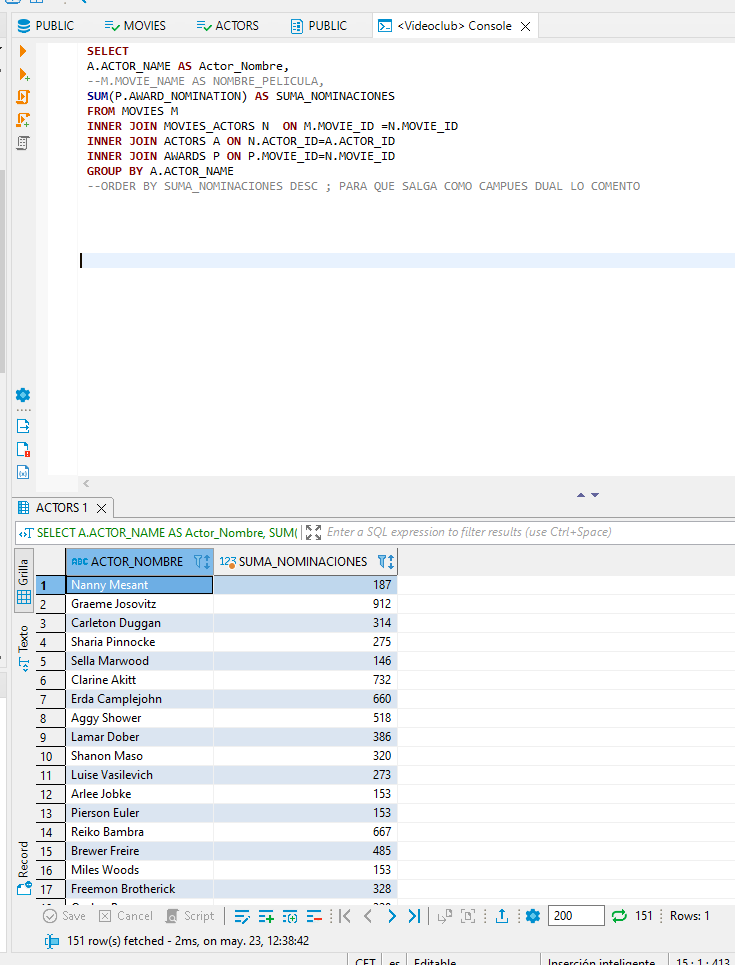
**INNER** **JOIN** MOVIES\_ACTORS N **ON** M.MOVIE\_ID =N.MOVIE\_ID

**INNER** **JOIN** ACTORS A **ON** N.ACTOR\_ID=A.ACTOR\_ID

**INNER** **JOIN** AWARDS P **ON** P.MOVIE\_ID=N.MOVIE\_ID

**GROUP** **BY** A.ACTOR\_NAME

--ORDER BY SUMA\_NOMINACIONES DESC ; PARA QUE SALGA COMO CAMPUES DUAL LO COMENTO



## Indica cuantos actores y directores hicieron películas para los estudios no activos.

**SELECT**

**COUNT**(**DISTINCT** A.ACTOR\_NAME) **AS** N\_ACTORES,

**COUNT**(**DISTINCT** D.DIRECTOR\_NAME) **AS** N\_DIRECTORES

**FROM** MOVIES M

**INNER** **JOIN** MOVIES\_ACTORS N **ON** M.MOVIE\_ID=N.MOVIE\_ID

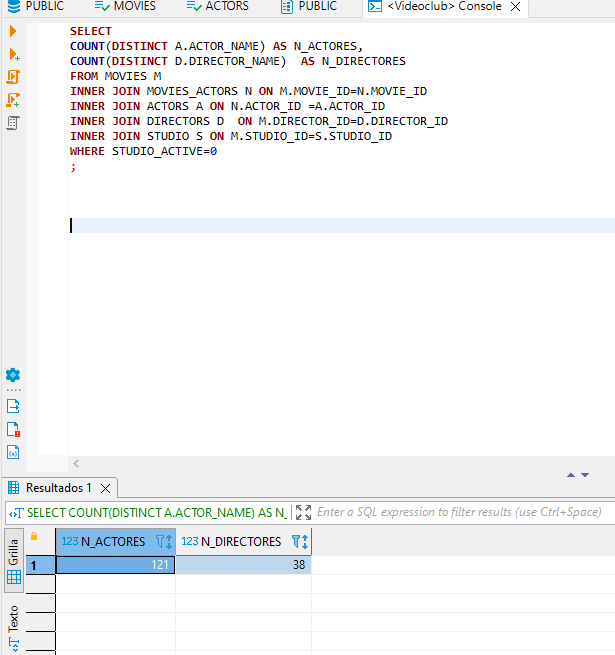
**INNER** **JOIN** ACTORS A **ON** N.ACTOR\_ID =A.ACTOR\_ID

**INNER** **JOIN** DIRECTORS D **ON** M.DIRECTOR\_ID=D.DIRECTOR\_ID

**INNER** **JOIN** STUDIO S **ON** M.STUDIO\_ID=S.STUDIO\_ID

**WHERE** STUDIO\_ACTIVE=0

;



Más sencillo compañera de curso

**SELECT**

**COUNT**(**DISTINCT** DIRECTOR\_ID) **AS** DIRECTORS,

**COUNT**(**DISTINCT** ACTOR\_ID) **AS** ACTORS

**FROM** STUDIO

**LEFT** **JOIN** MOVIES **USING**(STUDIO\_ID)

**JOIN** MOVIES\_ACTORS **USING**(MOVIE\_ID)

**WHERE** STUDIO\_ACTIVE=0

;

## Indica el nombre, ciudad, y teléfono de todos los miembros del videoclub que hayan alquilado películas que hayan sido nominadas a más de 150 premios y ganaran menos de 50.

Subselect o having

## Indica el número de películas que hayan hecho los directores durante las décadas de los 60, 70 y 80 que contengan la palabra "The" en cualquier parte del título.

SELECT \* FROM movies WHERE upper(movie\_name) LIKE upper('%The%')

Like %the%

## Indica si hay alguna coincidencia de nacimiento de ciudad (y si las hay, indicarlas) entre los miembros del videoclub y los directores.

## Comprueba si hay errores en la BD entre las películas y directores (un director muerto en el 76 no puede dirigir una película en el 88)

## Usando como condición la sentencia anterior, modifica la fecha de defunción a un año más tarde del estreno de la película (mediante sentencia SQL)

MERGE INTO PUBLIC.DIRECTORS D USING ( SELECT D.DIRECTOR\_ID, MAX(DATEADD(YEAR, 1, M.MOVIE\_LAUNCH\_DATE)) AS DIRECTOR\_DEAD\_DATE FROM PUBLIC.MOVIES M INNER JOIN PUBLIC.DIRECTORS D ON M.DIRECTOR\_ID = D.DIRECTOR\_ID WHERE D.DIRECTOR\_DEAD\_DATE IS NOT NULL AND D.DIRECTOR\_DEAD\_DATE < M.MOVIE\_LAUNCH\_DATE GROUP BY D.DIRECTOR\_ID) AS L(DIRECTOR\_ID, DIRECTOR\_DEAD\_DATE) ON D.DIRECTOR\_ID = L.DIRECTOR\_ID WHEN MATCHED THEN UPDATE SET D.DIRECTOR\_DEAD\_DATE = L.DIRECTOR\_DEAD\_DATE