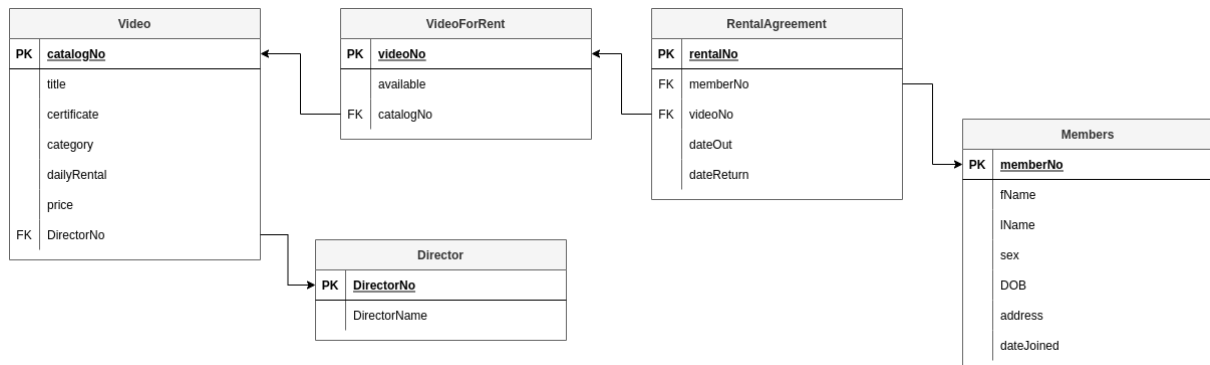


## Esquema relacional - VideoClub



## DDL de Tablas

```
CREATE TABLE Director
(
  DirectorNo      CHAR(5) NOT NULL,
  DirectorName    VARCHAR(25),
  PRIMARY KEY     (directorNo)
);

CREATE TABLE Video
(catalogNo        INT NOT NULL,
title             VARCHAR(30),
certificate        VARCHAR(10),
category          VARCHAR(10),
dailyRental       DECIMAL(15,4),
price             DECIMAL(15,4),
DirectorNo        CHAR(5),
PRIMARY KEY       (catalogNo),
FOREIGN KEY       (directorNo) REFERENCES Director (directorNo)
);

CREATE TABLE VideoForRent
(videoNo          INT NOT NULL,
available         INT,
catalogNo         INT,
PRIMARY KEY       (videoNo),
FOREIGN KEY       (catalogNo) REFERENCES Video (catalogNo)
);
```

```

CREATE TABLE Members
(memberNo          INT NOT NULL,
 fName            VARCHAR(15),
 lName            VARCHAR(15),
 sex              CHAR(1) DEFAULT 'M' CHECK (sex in ('M','F')),
 DOB              timestamp,
 address          VARCHAR(50),
 dateJoined       timestamp,
 PRIMARY KEY      (memberNo)
);

CREATE TABLE RentalAgreement
(rentalNo          INT NOT NULL,
 memberNo         INT,
 videoNo          INT,
 dateOut          timestamp DEFAULT NOW(),
 dateReturn       timestamp,
 PRIMARY KEY      (rentalNo),
 FOREIGN KEY      (videoNo) REFERENCES VideoForRent(videoNo),
 FOREIGN KEY      (memberNo) REFERENCES Members(memberNo)
);

INSERT INTO Director VALUES ('D0078', 'Stephen Herek');

INSERT INTO Director VALUES ('D1001', 'Roger Spottiswoode');

INSERT INTO Director VALUES ('D3765', 'Jane Emmerick');

INSERT INTO Director VALUES ('D4576', 'John Woo');

INSERT INTO Director VALUES ('D5743', 'Michael Bay');

INSERT INTO Director VALUES ('D7834', 'Sally Nichols');

INSERT INTO Video VALUES (207132, 'Tomorrow Never
Dies','12','Action',5,21.99,'D1001');

INSERT INTO Video VALUES (330553,
'Face/Off','12','Thriller',5,31.99,'D4576');

INSERT INTO Video VALUES (445624, 'The
Rock','18','Action',4,29.99,'D5743');

INSERT INTO Video VALUES (634817, 'Independence
Day','PG','Sci-Fi',4.5,32.99,'D3765');

```

```

INSERT INTO Video VALUES (781132, '101
Dalmatians','U','Children',4,18.5,'D0078');

INSERT INTO Video VALUES (902355, 'Primary
Colors','U','Comedy',4.5,14.5,'D7834');

INSERT INTO VideoForRent VALUES (178643, 0,634817);

INSERT INTO VideoForRent VALUES (199004, 1,207132);

INSERT INTO VideoForRent VALUES (200900, 1,330553);

INSERT INTO VideoForRent VALUES (210087, 1,902355);

INSERT INTO VideoForRent VALUES (243431, 1,634817);

INSERT INTO VideoForRent VALUES (245456, 1,207132);

INSERT INTO VideoForRent VALUES (245457, 1,207132);

INSERT INTO VideoForRent VALUES (317411, 1,781132);

INSERT INTO Members (memberNo, fname, lname, sex, DOB,
address,dateJoined) VALUES (1, 'Karen', 'Parker','F','1972-2-22','22
Greenway Drive, Glasgow, G12 8DS','1991-1-6');

INSERT INTO Members (memberNo, fname, lname, sex, DOB,
address,dateJoined) VALUES (2, 'Gillian', 'Peters','F','1954-03-09','89
Redmond Road, Glasgow, G11 9YR','1995-4-19');

INSERT INTO Members (memberNo, fname, lname, sex, DOB,
address,dateJoined) VALUES (3, 'Bob', 'Adams','M','1974-11-15','57
Littleway Road, Glasgow, G3 6DS','1998-1-6');

INSERT INTO Members (memberNo, fname, lname, sex, DOB,
address,dateJoined) VALUES (4, 'Don', 'Nelson','M','1951-12-12','123
Suffolk Lane,Glasgow, G15 1RC','2000-4-7');

INSERT INTO Members (memberNo, fname, lname, sex, DOB,
address,dateJoined) VALUES (5, 'Lorna', 'Smith','F','1972-1-1','Flat 5A
London Road, Glasgow, G3','1998-1-11');

INSERT INTO Members (memberNo, fname, lname, sex, DOB,
address,dateJoined) VALUES (6, 'Dan', 'White','M','1960-5-5','200 Great
Western Road, Glasgow, G11 9JJ','2001-5-5');

```

```
INSERT INTO RentalAgreement (rentalNo, memberNo, videoNo, dateOut,
dateReturn) VALUES (1, 1, 178643, '2000-2-5', '2000-2-7');

INSERT INTO RentalAgreement (rentalNo, memberNo, videoNo, dateOut,
dateReturn) VALUES (2, 1, 199004, '2000-2-5', '2000-2-7');

INSERT INTO RentalAgreement (rentalNo, memberNo, videoNo, dateOut,
dateReturn) VALUES (3, 1, 245456, '2000-2-4', '2000-2-6');

INSERT INTO RentalAgreement (rentalNo, memberNo, videoNo, dateOut,
dateReturn) VALUES (4, 1, 243431, '2000-2-4', '2000-2-6');

INSERT INTO RentalAgreement (rentalNo, memberNo, videoNo, dateOut,
dateReturn) VALUES (5, 3, 199004, '1999-11-11', '1999-11-12');

INSERT INTO RentalAgreement (rentalNo, memberNo, videoNo, dateOut,
dateReturn) VALUES (6, 2, 245456, '1999-11-11', '1999-11-13');

INSERT INTO RentalAgreement (rentalNo, memberNo, videoNo, dateOut,
dateReturn) VALUES (7, 4, 178643, '1999-11-14', '1999-11-16');

INSERT INTO RentalAgreement (rentalNo, memberNo, videoNo, dateOut,
dateReturn) VALUES (8, 3, 243431, '1999-11-11', '1999-11-13');

INSERT INTO RentalAgreement (rentalNo, memberNo, videoNo, dateOut,
dateReturn) VALUES (9, 1, 245457, '1999-11-12', '1999-11-14');

INSERT INTO RentalAgreement (rentalNo, memberNo, videoNo, dateOut,
dateReturn) VALUES (10, 3, 199004, '2000-8-10', NULL);

INSERT INTO RentalAgreement (rentalNo, memberNo, videoNo, dateOut,
dateReturn) VALUES (11, 2, 199004, '2000-8-17', '2000-8-20');

INSERT INTO RentalAgreement (rentalNo, memberNo, videoNo, dateOut,
dateReturn) VALUES (12, 6, 178643, '2004-2-5', '2004-2-7');

INSERT INTO RentalAgreement (rentalNo, memberNo, videoNo, dateOut,
dateReturn) VALUES (13, 6, 317411, '2004-2-5', '2004-2-7');

INSERT INTO RentalAgreement (rentalNo, memberNo, videoNo, dateOut,
dateReturn) VALUES (14, 4, 317411, '2003-2-5', '2003-2-7');

INSERT INTO RentalAgreement (rentalNo, memberNo, videoNo, dateOut,
dateReturn) VALUES (15, 1, 317411, '2002-2-5', '2002-2-7');
```

## Bloque 1: Generación de salida

Dadas las siguientes tablas, se debe indicar el resultado de aplicar las consultas indicadas:

Tabla1

A	B	D
a1	blue	d1
a2	blue	d2
a3	black	d3
a4	red	null

Tabla2

A	C
a1	100
a3	200

### Consulta 1)

```
select
    t1.A, t1.B, t2.C
from Tabla1 t1
left join Tabla2 t2 on t1.A = t2.A;
```

	A	B	C
1	a1	blue	100
2	a3	black	200
3	a2	blue	<null>
4	a4	red	<null>

### Consulta 2)

```
select
    t1.B, t1.A
from Tabla1 t1
group by t1.B;
```

	B	A
1	black	a3
2	blue	a1
3	red	a4

### Consulta 3)

```
select
    t1.B, count(t1.D) as conteo
from Tabla1 t1
group by t1.B;
```

	B	conteo
1	black	1
2	blue	2
3	red	0

### Consulta 4)

```
select
    A, C
from Tabla2
order by C desc;
```

	A	C
1	a3	200
2	a1	100

## Bloque 2: Análisis de video club

Análisis de videoclub, en el ejercicio se planteó un escenario y se realizaron solicitudes de información.

### Escenario:

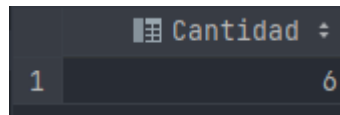
En una tienda de renta de videos se acaba de incorporar un nuevo gerente y requiere cierta información para conocer el lugar y su operación.

Se debe entregar:

- Consulta
- Resultados de la consulta, tomar una captura de la pantalla con los resultados.

**Solicitud 1)** Para iniciar a conocer el inventario solicita la cantidad de películas diferentes.  
**Nota:** revisar en el diagrama, que una película puede tener varios videos (en stock) asociados.

```
SELECT COUNT(DISTINCT title) as Cantidad  
FROM video;
```



Cantidad
6

**Solicitud 2)** Lista los títulos de películas ordenados por nombre de película:

```
SELECT title  
FROM video  
ORDER BY title;
```



title
1 101 Dalmatians
2 Face/Off
3 Independence Day
4 Primary Colors
5 The Rock
6 Tomorrow Never Dies

Ejemplo de salida:

Película
Avengers
Dr. Strange
El señor de los anillos: La comunidad del anillo
La liga de la justicia

**Solicitud 3)** Desea conocer el inventario actual. Por lo que requiere el listado de películas y cantidad de videos por cada película, ordenado por cantidad de videos de forma descendiente.

Ejemplo de salida:

Película	Cantidad de videos
Avengers	5
El señor de los anillos: La comunidad del anillo	3
Dr. Strange	2
La liga de la justicia	0

```
SELECT title, SUM(IF(video.catalogNo = videoforrent.catalogNo, 1, 0)) as Cantidad
FROM video, videoforrent
GROUP BY title
ORDER BY Cantidad DESC;
```

	title	Cantidad
1	Tomorrow Never Dies	3
2	Independence Day	2
3	101 Dalmatians	1
4	Primary Colors	1
5	Face/Off	1
6	The Rock	0

```
# OW WITHOUT 0s
SELECT title, COUNT(*) as Cantidad
FROM video, videoforrent
WHERE video.catalogNo = videoforrent.catalogNo
GROUP BY title
ORDER BY Cantidad DESC;
```

	title	Cantidad
1	Tomorrow Never Dies	3
2	Independence Day	2
3	Primary Colors	1
4	Face/Off	1
5	101 Dalmatians	1



**Solicitud 4)** Desea conocer el comportamiento de las rentas, para lo que quiere un listado de las películas y el número de veces que se ha rentado cada película, el listado debe estar ordenado de mayor a menor por la cantidad de rentas.

Ejemplo de salida:

Title	Cantidad de rentas
Avengers	15
Dr. Strage	7
El señor de los anillos: La comunidad del anillo	3
La lista de la justicia	0

```
SELECT title, SUM(IF(video.catalogNo = videoforrent.catalogNo AND
rentalagreement.videoNo = videoforrent.videoNo, 1, 0)) as Cantidad
FROM video,
      rentalagreement,
      videoforrent
GROUP BY title
ORDER BY Cantidad DESC;
```

	title	Cantidad
1	Tomorrow Never Dies	7
2	Independence Day	5
3	101 Dalmatians	3
4	Primary Colors	0
5	Face/Off	0
6	The Rock	0

```
# OW WITHOUT 0s
SELECT title, COUNT(*) as Cantidad
FROM video,
      rentalagreement,
      videoforrent
WHERE video.catalogNo = videoforrent.catalogNo AND
      rentalagreement.videoNo = videoforrent.videoNo
GROUP BY title
ORDER BY Cantidad DESC;
```

	title	Cantidad
1	Tomorrow Never Dies	7
2	Independence Day	5
3	101 Dalmatians	3

**Solicitud 5)** Desea conocer quienes son los mejores clientes, para lo que solicita un listado de clientes: nombre, apellido, género, fecha de nacimiento de los clientes y el número de rentas realizadas. El listado debe estar ordenado por rentas realizadas de mayor a menor. Los primeros en la lista le permitirán conocer quienes son los mejores clientes.

Ejemplo de salida:

Nombre	Apellido Paterno	Género	Fecha nacimiento	Número de rentas
Carlos Natanael	Lecona Valdespino	M	2001-01-01	6
Susana Paola	Martínez Calzada	F	2001-01-01	5
Guillermo Alejandro	Hernández Sosa	M	2001-01-01	5

```
SELECT fName as Nombre, LName as Apellido, sex as Genero, DOB as FechaNacimiento,
COUNT(rentalNo) as NumeroRentas
FROM members,
    rentalagreement
WHERE members.memberNo = rentalagreement.memberNo
GROUP by fName
ORDER BY NumeroRentas DESC;
```

	Nombre	Apellido	Genero	FechaNacimiento	NumeroRentas
1	Karen	Parker	F	1972-02-22 00:00:00	6
2	Bob	Adams	M	1974-11-15 00:00:00	3
3	Dan	White	M	1990-05-05 00:00:00	2
4	Gillian	Peters	F	1994-03-09 00:00:00	2
5	Don	Nelson	M	1991-12-12 00:00:00	2

**Solicitud 7)** Para propósitos de marketing quiere saber el nombre del director y la cantidad de mujeres y hombres que han rentado sus películas.

Ejemplo de salida:

Director	Total F	Total M
Joss Whedon	3	3
Sam Raimi	3	1
Zack Snyder	0	0

```
# USING SUM IF/CASE WHEN AS COUNTER
SELECT DirectorName, SUM(IF(sex = 'F', 1, 0)) as Female, SUM(IF(sex = 'M', 1, 0)) as Male
FROM director,
     video,
     videoforrent,
     rentalagreement,
     members
WHERE (director.DirectorNo = video.DirectorNo AND video.catalogNo =
      videoforrent.catalogNo AND
      videoforrent.videoNo = rentalagreement.videoNo AND rentalagreement.memberNo =
      members.memberNo)
GROUP BY DirectorName;
```

	DirectorName	Female	Male
1	Jane Emmerick	2	3
2	Roger Spottiswoode	5	2
3	Stephen Herek	1	2

```
# NATURAL JOIN
SELECT *
FROM (SELECT DirectorName, COUNT(*) as Male
      FROM director,
           video,
           videoforrent,
           rentalagreement,
           members
      WHERE (director.DirectorNo = video.DirectorNo AND video.catalogNo =
            videoforrent.catalogNo AND
            videoforrent.videoNo = rentalagreement.videoNo AND rentalagreement.memberNo
            = members.memberNo AND
            members.sex = 'M')
      GROUP BY DirectorName) as DNM
      NATURAL JOIN (SELECT DirectorName, COUNT(*) as Female
                    FROM director,
                         video,
                         videoforrent,
                         rentalagreement,
                         members
                    WHERE (director.DirectorNo = video.DirectorNo AND video.catalogNo
                    = videoforrent.catalogNo AND
                        videoforrent.videoNo = rentalagreement.videoNo AND
                        rentalagreement.memberNo = members.memberNo AND
                        members.sex = 'F')
                    GROUP BY DirectorName) as DNF;
```

	DirectorName	Male	Female
1	Jane Emmerick	3	2
2	Roger Spottiswoode	2	5
3	Stephen Herek	2	1

# OW WITH TWO TABLES CREATED; BUT NEEDS TO STORE

CREATE table if not exists Male

SELECT DirectorName, COUNT(\*) as Male

FROM director,  
video,  
videoforrent,  
rentalagreement,  
members

WHERE (director.DirectorNo = video.DirectorNo AND video.catalogNo =  
videoforrent.catalogNo AND

videoforrent.videoNo = rentalagreement.videoNo AND rentalagreement.memberNo =  
members.memberNo AND  
members.sex = 'M')

GROUP BY DirectorName;

CREATE table if not exists female

SELECT DirectorName, COUNT(\*) as Female

FROM director,  
video,  
videoforrent,  
rentalagreement,  
members

WHERE (director.DirectorNo = video.DirectorNo AND video.catalogNo =  
videoforrent.catalogNo AND

videoforrent.videoNo = rentalagreement.videoNo AND rentalagreement.memberNo =  
members.memberNo AND  
members.sex = 'F')

GROUP BY DirectorName;

# PUT TWO TABLES TOGETHER

SELECT \*

FROM male

NATURAL JOIN female;

# DROP TABLES

Drop table if exists male, female;

	DirectorName	Male	Female
1	Jane Emmerick	3	2
2	Roger Spottiswoode	2	5
3	Stephen Herek	2	1

## Bloque 3: Entendimiento de consultas

Describe la consulta dada, no parafrasear la consulta.

Ejemplo:

```
SELECT title, price
from video
where price > 10;
```

Descripción:

- Esta consulta crea una proyección de los títulos y precios de los videos que cuestan más de 10 dólares.

**Ejercicio 1:**

```
select
  fname, lname
from members
where datejoined > '1998-01-01'
order by datejoined desc;
```

Descripción:

- Esta consulta genera una proyección de los nombres y apellidos de los miembros que se unieron después del 1 de enero de 1998, ordenados por fecha de ingreso descendente.

**Ejercicio 2:**

```
select distinct
  d.directorname
  , v.title
from video v
left join videoforrent vf on v.catalogno = vf.catalogno
left join rentalagreement ra on vf.videonno = ra.videonno
left join members me on ra.memberno = me.memberno
left join director d on v.directorno = d.directorno
where me.lname = 'Parker';
```

Descripción:

- Esta consulta genera una proyección de los nombres de directores y títulos de videos, con un left join de la tabla videoforrent donde el número de catálogo del video sea igual en la tabla video y en la tabla videoforrent, un left join de la tabla rentalagreement donde el número de videos sea igual en la tabla videoforrent y en la tabla rentalagreement, un left join de la tabla members donde el número de miembro

sea igual en la tabla rentalagreement y en la tabla members, un left join de la tabla director, donde el número de directores sea igual en la tabla video y en la tabla director, y donde el apellido del miembro sea Parker.

### Ejercicio 3:

```
select title
from video
EXCEPT
select title from Video v, VideoForRent vr, RentalAgreement ra
where v.catalogNo = vr.catalogNo
and vr.videoNo = ra.videoNo;
```

#### Descripción:

- Esta consulta genera una proyección de los títulos de los videos que están en la tabla video, a excepción de los videos que están en la tabla video, la tabla videoforrent y la tabla rentalagreement donde el número de catálogo del video sea igual en la tabla video y en la tabla videoforrent, y el número de videos sea igual en la tabla videoforrent y en la tabla rentalagreement.