

MySQL

Repaso

```
SELECT id, titulo, fecha_de_estreno  
FROM pelicula  
WHERE fecha_de_estreno > '2000-01-01'  
ORDER BY titulo DESC  
LIMIT 5, 5
```

Repaso

```
SELECT pelicula.titulo, pelicula.fecha_de_estreno, genero.nombre  
FROM pelicula  
      INNER JOIN genero ON genero.id = pelicula.id_genero  
WHERE pelicula.fecha_de_estreno > '2000-01-01'
```

Repaso

```
SELECT
    pelicula.id,
    pelicula.titulo,
    pelicula.fecha_de_estreno,
    genero.nombre,
    actor_pelicula.id_actor,
    actor.nombre,
    actor.apellido
FROM pelicula
    INNER JOIN genero ON genero.id = pelicula.id_genero
    LEFT JOIN actor_pelicula ON actor_pelicula.id_pelicula = pelicula.id
        LEFT JOIN actor ON actor.id = actor_pelicula.id_actor
WHERE
    pelicula.id > 15
```

Repaso - Obtener los actores de las películas que digan “Harry Potter”

```
SELECT actor.nombre, actor.apellido
FROM actor
      INNER JOIN actor_pelicula ON actor.id = actor_pelicula.id_actor
      INNER JOIN pelicula ON pelicula.id = actor_pelicula.id_pelicula
WHERE
      pelicula.titulo LIKE '%Harry Potter%'
;
```

Repaso - Obtener los actores de las películas que digan “Harry Potter”

nombre	apellido
▶ Daniel	Radcliffe
Emma	Watson
Rupert	Grint
Daniel	Radcliffe
Emma	Watson
Rupert	Grint
Daniel	Radcliffe
Emma	Watson
Rupert	Grint
Helena	Bonham Carter

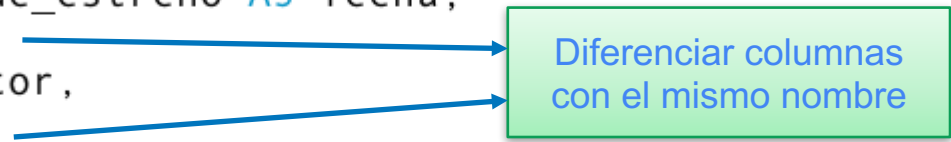
DISTINCT

```
SELECT DISTINCT actor.nombre, actor.apellido
FROM actor
      INNER JOIN actor_película ON actor.id = actor_película.id_actor
      INNER JOIN película ON película.id = actor_película.id_película
WHERE
      película.título LIKE '%Harry Potter%'
;
```

nombre	apellido
▶ Daniel	Radcliffe
Emma	Watson
Helena	Bonham Carter
Rupert	Grint

Alias de Tablas / Campos

```
SELECT
    p.id,
    p.titulo,
    p.fecha_de_estreno AS fecha,
    g.nombre,
    ap.id_actor,
    a.nombre,
    a.apellido
FROM pelicula AS p
    INNER JOIN genero AS g ON g.id = p.id_genero
    LEFT JOIN actor_pelicula AS ap ON ap.id_pelicula = p.id
        LEFT JOIN actor AS a ON a.id = ap.id_actor
WHERE
    p.id > 15
```



Diferenciar columnas con el mismo nombre

Práctica!

BETWEEN

```
SELECT p.id, p.titulo
FROM pelicula AS p
WHERE
    p.id >= 5
    AND p.id <= 10
;
```

BETWEEN

```
SELECT p.id, p.titulo  
FROM pelicula AS p  
WHERE  
    p.id BETWEEN 5 AND 10  
;
```

BETWEEN

```
SELECT p.*  
FROM pelicula AS p  
WHERE  
    fecha_de_estreno BETWEEN '2005-01-01' AND '2015-12-31'  
;
```

IN

```
SELECT p.id, p.titulo  
FROM pelicula AS p  
WHERE  
    p.id IN (5, 6, 7, 8, 9, 10)|  
;
```

Práctica!

Agrupación de datos

Permiten responder preguntas del tipo:

- ¿Cuántas películas tengo en total?
- ¿Cuántas películas tengo de determinado Género?
- ¿Qué rating tienen en promedio?
- ¿Cuál es la que dura más y cuál la que dura menos tiempo?

Funciones de agregación

COUNT

MIN

MAX

SUM

AVG

Funciones de agregación

```
SELECT COUNT(*)  
FROM pelicula;
```

```
SELECT COUNT(id)  
FROM pelicula;
```

```
SELECT COUNT(id) AS total  
FROM pelicula  
WHERE id_genero = 3;
```

Funciones de agregación

```
SELECT AVG(rating)
FROM pelicula;
```

```
SELECT SUM(duracion)
FROM pelicula;
```

```
SELECT MIN(rating)
FROM pelicula;
```

```
SELECT MAX(rating)
FROM pelicula;
```

Funciones de agregación

Ejercicio:

Mostrar la cantidad de películas por género

GROUP BY

```
SELECT id_genero, COUNT(id) AS total  
FROM pelicula  
GROUP BY id_genero;
```

```
SELECT g.nombre, COUNT(p.id) AS total  
FROM pelicula p  
    INNER JOIN genero g ON p.id_genero = g.id  
GROUP BY g.nombre;
```

Funciones de agregación

Ejercicio:

Mostrar los géneros y su cantidad de películas

GROUP BY

```
SELECT g.nombre, COUNT(p.id) AS total
FROM genero g
      INNER JOIN pelicula p ON p.id_genero = g.id
GROUP BY g.nombre;
```

```
SELECT g.nombre, COUNT(p.id) AS total
FROM genero g
      LEFT OUTER JOIN pelicula p ON p.id_genero = g.id
GROUP BY g.nombre;
```

Funciones de agregación – Mostrar los géneros que tengan al menos 3 películas

```
SELECT g.nombre, COUNT(p.id) AS total
FROM genero g
     LEFT OUTER JOIN pelicula p ON p.id_genero = g.id
GROUP BY g.nombre
ORDER BY total;
```

nombre	total
► Suspenso	0
Accion	0
Terror	0
Comedia	1
Documental	1
Aventuras	3
Drama	3
Infantiles	3
Animacion	4
Ciencia Ficción	5

HAVING

```
SELECT g.nombre, COUNT(p.id) AS total
FROM genero g
      LEFT OUTER JOIN pelicula p ON p.id_genero = g.id
GROUP BY g.nombre
HAVING total >= 3
ORDER BY total;
```

HAVING sería el WHERE de las funciones de agregación

Funciones Interesantes

```
SELECT
    id,
    titulo,
    fecha_de_estreno,
    DATE_FORMAT(fecha_de_estreno, '%W %M %Y') AS fecha_estreno_verbose
FROM pelicula
ORDER BY rating
```

Funciones Interesantes

```
SELECT
    id,
    REPLACE(titulo, 'Harry', 'Pedro') as titulo_manipulado
FROM pelicula
ORDER BY id
```

Funciones Interesantes

```
SELECT
    id,
    nombre,
    apellido,
    CONCAT(nombre, ' ', apellido) AS nombre_completo
FROM actor
ORDER BY id
```

Funciones Interesantes

```
SELECT
  id,
  titulo,
  rating,
  CASE
    WHEN rating < 4 THEN 'mala'
    WHEN rating < 6 THEN 'regular'
    WHEN rating < 8 THEN 'buena'
    WHEN rating < 9.5 THEN 'muy buena'
    ELSE 'excelente'
  END AS rating_cat
FROM pelicula
ORDER BY rating
```

Más funciones

Para ir investigando:

- Operadores
- Funciones de Control de Flujo
- Funciones para tipos Numérico, String, Date&Time
- ...

<http://dev.mysql.com/doc/refman/5.7/en/functions.html>

Práctica!